INTELLIGENT SYSTEMS FOR DIGITAL PATTERN ANALYSIS AND DESIGN SUPPORT

QU HONG

PhD

The Hong Kong Polytechnic University

2023

The Hong Kong Polytechnic University

School of Fashion and Textiles

Intelligent Systems for Digital Pattern Analysis and Design Support

Qu Hong

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

August 2022

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

屈弘 (Signed)

QU Hong (Name of student)

ABSTRACT

Nowadays, consumers are empowered, trendy, and they are always looking for newness and excitement. To address this need, the demand for new products is growing at accelerated speed, and this can be applied to all types of consumer products, including fashion. Graphics and patterns play an essential role in fashion design and product development. Hundreds of decorative patterns are required to be produced within days while the tedious, manual and laborious creation process is not catching up well with the demand. Very few research has been done on elevating the decorative pattern design process and the existing Computer-Aid Design (CAD) systems are rarely used to create patterns in digital format. In the digital era, computers are now being used to generate designs, but the designs or artworks generated neither meet the aesthetic criteria nor are in the correct format for production or further editing.

Following the development pathway of artificial intelligence, the way we can build an intelligent pattern design system is by enabling computers to learn from design samples, of which useful pattern knowledge can be analyzed, extracted and processed to create new designs. By thoroughly reviewing the existing practices of design creation in the fashion industry, a new framework is proposed. Three intelligent systems are developed and experimentally verified in this study. They include the efficient repeated pattern detection system (R-system), the automatic design element extraction and vectorization system (E-system), and the vector-based digital pattern generation system (G-system). Each system serves a specific purpose. They can work either interpedently as standalone system or seamlessly work together as a whole for digital pattern creations.

The R-system is responsible for efficient detection of repeated patterns from input images. A repeated pattern is the smallest unit that can tile the overall image and represents the primary information of the image in a compact form. Repeated patterns appear in many kinds of visuals: from natural scenes, building and architectures, to designs such as textiles, product/graphic designs for packing or wallpapers. Detection of such repeated patterns also supports many downstream applications such as image retrieval and image synthesis. After reviewing different approaches, a hybrid method is proposed to keep a good balance among content diversity, topology regularity and the trade-off between robustness and speed. In particular, this study firstly leverages activations of a pre-trained Convolutional Neural Network (CNN) to predict coarse repeated pattern size options of the input image. Accurate repeated patterns of the input images are then obtained by template matching optimization. Experiments are conducted on a proposed dataset to demonstrate the superiority of our methods. The accuracy of our method is 0.673 which is 20% higher than the baseline method and the time cost is only its 11%.

The proposed R-system analyzes input images to detect repeated patterns generating unit region of image with repeated patterns or entire image of unrepeated image as outputs. Such outputs from the R-system can be inputted to the E-system for further design analysis. The E-system focuses on automatically identifying and extracting core design elements from input images. A design image can be considered as an arrangement of design elements filled with specific colors according to certain layout rules. This study vectorizes the extracted design elements and keeps outcomes compact. In particular, unsupervised segmentation is applied during core design element extraction to solve the problem of the lack of dataset for labeled designs. Next, a novel design element deconstruction method is proposed for vectorization based on color quantization. Extensive experiments on design images demonstrated the effectiveness of the proposed method. Our method extracts and vectorizes the core design elements of an image in around 13 seconds. The output vectorized design elements are more compact than common business software and are easier to reuse for new design generations. Furthermore, a vector-based design element dataset is built to support design generation.

The proposed R-system and E-system can analyze large volume of digital pattern samples, and extract useful design elements and rules such as colors and topology structures. With the learnt design rules, the G-system is proposed to support the creation of new vector-based patterns, meeting humans' need on aesthetics and are ready for production. The proposed G-system is for generation of new digital patterns that can be applied in textiles and other decorative graphic designs. To achieve this, this study generally classifies textile patterns into three main groups: stripe, check, and motif. New designs can be regarded as combinations of different geometric and color parameters. Hence parametric models are formulated by flexible combination of geometric and color parameters learnt from reference images. Implementation experiments prove the design outputs meet basic human aesthetic, and our system can support design work.

To summarize, a new REG framework is proposed in this study by integrating three intelligent systems for analyzing and supporting pattern design making. It promotes the use of artificial intelligence for design generation that meets the needs of human aesthetic requirements and caters to the speedy fashion product development cycles.

Keywords: Fashion Computer-aid Design; Digital Pattern Analysis; Design Generation; Vectorization; Intelligent Systems; Human Aesthetics

PUBLICATIONS

1. Qu, H., Zhou, Y., Chau, K. P., & Mok, P. Y., Repeated pattern extraction with knowledge-based attention and semantic embeddings, 14th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, 2020.
2. Qu, H., Chau, K. P., & Mok, P. Y., Detection of Repeated Patterns with CNN Activations and Similarity Matching, 1st ITC-KSCT Joint Symposium, 2022.
3. Qu, H., Chau, K. P., & Mok, P. Y., Design Elements Extraction Based on Unsupervised Segmentation and Compact Vectorization, 16th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, 2022.
4. Qu, H., Zhou, Y., Chau, K. P., & Mok, P. Y., Efficient and Effective Detection of Repeated Pattern from Fronto-Parallel Images with Unknown Visual Contents, Under Review, Engineering Applications of Artificial Intelligence, 2023.
5. Qu, H., Zhou, Y., Chau, K. P., & Mok, P. Y., Graphic Pattern Deconstruction for Reuse: Automatic Design Elements Extraction and Vectorization, to be submitted to Computer-Aided Design, 2023.
6. Qu, H., Zhou, Y., Chau, K. P., & Mok, P. Y., Editable Graphic Pattern Generation for Textile, Working paper, 2023.

Acknowledgements

I want to show sincere appreciation to my chief supervisor, Dr. P. Y. Mok, who gave me the opportunity to start all over again, which was like giving a drowning person a lifebuoy. Over the past few years, she has supported me, given me freedom to do things I am interested in, and shown me unending tolerance and patience. She is a very kind, wise, and responsible lady, strict with herself yet very understanding of those around her. In contrast, I realized how I can be extreme and foolish sometimes. Maugham said a person could be the pursuit of the highest ideal is self-perfection. Life is long, and I think I always have a chance, and yet still a long way to go, to emulate her. All in all, it is my honor to be her student; without her, I cannot finish my research. My sincere appreciation also goes to the professors who helped me get into the sacred sanctuary of the academic world, Prof. Kinor Jiang, Prof. Lilly Li, and my co-supervisor, Dr. K. P. Chau.

I want to thank the Hong Kong government and PolyU for working so hard in keeping the society and university function normally throughout the pandemic and making sure my work and life were almost unaffected. Furthermore, I was treated humanely in the quarantine camp during my COVID-19 infection.

I also wish to thank my parents. Not just for giving me life, but for their love and support all the way through. Previously, I didn't think love is unconditional. I now believe it.

They gave me the best they can, though we frequently misunderstood each another and struggled with expressing our feelings. And I hope that in the future, we will still have plenty of chances to treat each another nicely.

I also want to thank my friends and colleagues for their support and patience. Special thanks go out to Dr. Gaby Shi, Dr. Xie Ningrong, Dr. Ding Yujuan, Dr. Chao Xujiang, Dr. Wang Chen, Ms. Zhang Xin, Ms. He Honghong, Ms. Zhou Yangping, Ms. Ding Yueming, Mr. Peng Jihua, and Mr. Cao Yu for their invaluable assistance with my research and life. Sometimes their simple words can solve my great confusion. Special thanks to my friend, Ms. Li Runze, I very much appreciate her constant support and encouragement. Special thanks to Dr. Zhou Yanghong, for leading me into the field of scientific research. Special thanks to my friend and roommate, Dr. Li Ziqi, for her company, understanding, selfless sharing, and support for the past four years. The same go out to the unnamed friends. The list is too long to be listed here.

Special greetings to Vivian, the inspiration for the research. And hope everything goes well.

Table of Contents

List of Figures

List of Tables

CHAPTER 1. Introduction

1.1 Background

The global fashion industry is essential to our day-to-day life and indispensable to the world’s economy. Although the world’s economy have suffered from nearly two years of disruption, the fashion industry still accounts for 2% of the global economy (FashionUnited, 2022). Consumers are provided with homogeneous products at low prices in sufficient quantity through mass production led by technological advancement. Meanwhile, the technology revolution has also been shaping *empowered* consumers who are closely connected to the internet, keep sharing fashion information, and are more complex, trendy, and hard to satisfy (Blázquez, 2014; Xu, 2015). The expectations of consumers today are far higher than they used to be. They frequently choose fashion products with added features to better express their personalities and distinguish themselves from the crowd (Teunissen *et al.*, 2015). To meet consumers’ demands, fashion companies have segmented the market to manufacture *consumer-focused* products and look for active methods to appeal to consumers to their brands (Camargo *et al.*, 2020; Ko *et al.*, 2007; Quinn *et al.*, 2007). Moreover, due to the seasonal characteristic of fashion products, the development of consumer-focused fashion products becomes a dynamic process with high seasonal demand and a tight development cycle.

Examining the process of fashion product development is crucial for creating consumer-focused products. Although product development for new products is a comprehensive process, the generic development process begins with designing. Followed by modeling and prototyping, which aims to create the sample products to be displayed at fashion fairs, detail engineering and material sourcing come next, while the final process is manufacturing and distribution (Bandinelli *et al.*, 2013; Spahiu *et al.*, 2014).

In order to efficiently respond to the changing demand for fashion products, digital technology is applied to almost every section along the fashion value chain. For example, Enterprise Resource Planning (ERP) systems are applied to manage daily business activities, for instance, accounting, procurement, project management, and risk management and compliance (Westrup *et al.*, 2000). Supply Chain Management (SCM) systems focus on managing the flow of goods, data and finances, from acquiring raw materials to delivering the product to its final destination (Oracle, 2022). Customer Relationship (CRM) systems gather, link, and analyze all collected customer data, including contact details and communications with business partners (ApparelMagic, 2022). Computer-aided design (CAD) systems support different design operations, including garment patternmaking, fashion technical drawing, 3D garment simulation, and decorative pattern creation in digital formats (Tabraz, 2017; Xu *et al.*, 2016).

As discussed early, design is the first step of the product journey, one of the most important process in the product development cycle. The process of design covers many types of design. Apart from the shape or silhouette, fashion design also covers decorative elements like decorative patterns (Albert *et al.*, 2004; Seivewright, 2012). Decorative patterns are essential components for numerous fashion products because of they can improve uniqueness of the product by incorporating expressive means of visual design (Fogg, 2006; Seo *et al.*, 2007). Decorative pattern design is frequently linked to textile design. However, it also can be applied directly to fashion products using a range of techniques such as placement printing or patch embroidery.

Early decorative patterns were designed by manual sketching, making them difficult to edit and preventing them from being used for mass production (Studd, 2002). To overcome these limitations, decorative patterns in digital editable formats emerge. This kind of digital patterns are usually in *vector* format, also named as Scalable Vector Graphics (SVG). They are made up of geometrical primitives and thus provide a number of benefits, including minimal file size, and ease to edit or scale to any size without a loss in quality or details (Dominici *et al.*, 2020; Lu *et al.*, 2014; Selinger, 2003).

Although digital patterns are essential components of garments, the development of digital patterns is often separated from fashion garment development (Lu *et al.*, 2017).

To better assist digital pattern development, it is necessary to get a thorough understanding of its development process. Studd (2002) divided the process into five generic phases based on a study with digital pattern designers involving different fashion companies; see Figure 1-1 for an illustration of the 5-phase development process of digital pattern. In the first phase, design project planning should be conducted, considering the trends of the specific product category, to decide the number of designs, colorways, due time, etc. Next, in the phase of research and analysis, the information about the products and markets are collected and analyzed. The third phase is design synthesis, including concept development, design creation, design solutions, and colorway development. Following which is design selection, consisting of design presentation, selection, sampling, and final selection. This phase of processes requires repeated communication between all participants. The last phase is production. Comparing to design synthesis, the phases for research and repeated communication are more time-consuming in the digital pattern design process. Thus, scholars have been researching on how to extract useful design information from existing designs and re-use them to support new design generation. Computer aided design (CAD) plays an important role is this area.

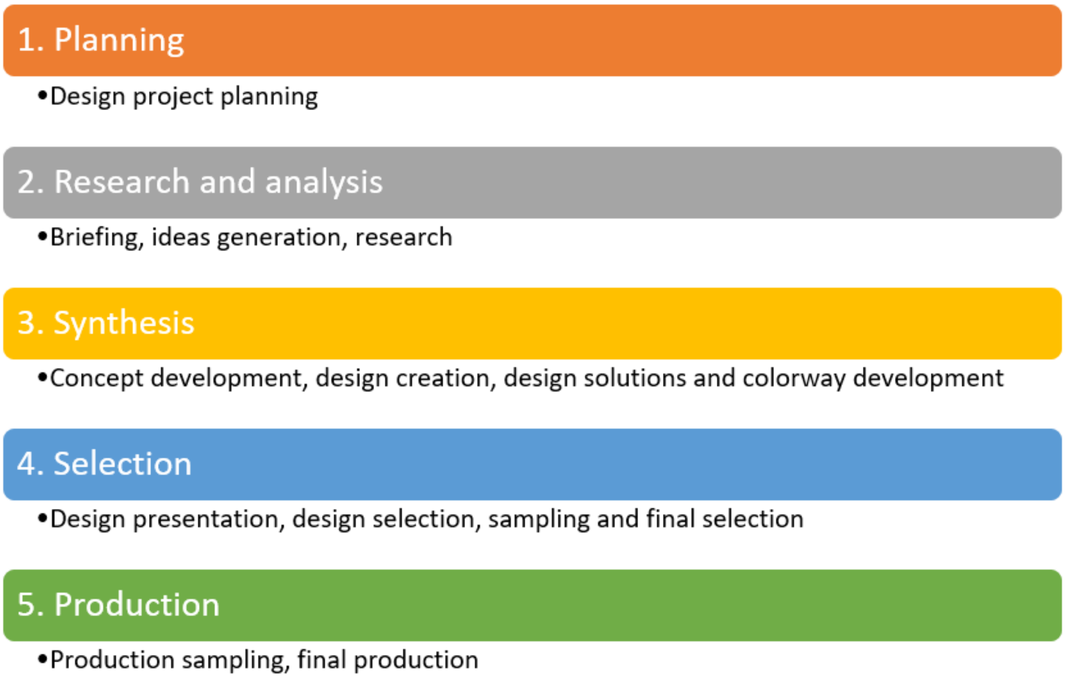

1. Planning
•Design project planning
2. Research and analysis
•Briefing, ideas generation, research
3. Synthesis
•Concept development, design creation, design solutions and colorway development
4. Selection
•Design presentation, design selection, sampling and final selection
5. Production
•Production sampling, final production

Figure 1-1 The generic process for digital pattern design (Studd, 2002)

CAD is now essential for creating digital patterns in the fashion industry (Bandinelli *et al.*, 2013). The existing CAD software, such as Adobe Illustrator and CorelDraw, streamline design work to a certain extent; they provide a convenient and efficient way to constantly improve design efficiency. In the current CAD interfaces, designers can create or modify editable *design elements* to form various digital patterns conveniently in different *colorways*. They also can tile small-size patterns (also known as *repeated patterns*) to construct large-size patterns that satisfy production needs. However, these CAD interfaces do not essentially optimize the entire design process, and computers are only used to replace the traditional pen and paper in the process of pattern design. In a word, common CAD software are just digital creation facilities replacing freehand tools. Designers still need to spend a lot of time capturing customer preferences and collecting inspiration for creation. Due to increasing stress/demand on fashion product development, designers long for better CAD systems that can provide professional support or suggestions basing on industry needs or fashion product development knowledge.

Other than commercial software, some researchers also proposed new methods or systems to support design creation. For example, some researchers have utilized the close connection between mathematics and art to create particular digital patterns in vector formats, such as marbling and fractal art design (Liu *et al.*, 2008; Lu *et al.*, 2014; Wang *et al.*, 2019); some output examples are presented in Figure 1-2. Since these patterns exist in vector format, they can be easily to modified to meet production requirements. However, each method is only for creating a particular type of digital pattern and hence limits its application prospects. Artificial intelligence (AI), deep learning in particular, has recently been making significant advancements in solving image understanding and computer vision problems. For example, image classification (Liu *et al.*, 2015), image segmentation (Kaur *et al.*, 2014), and object detection (Zou *et al.*, 2019). There are some novel methods leveraging deep learning technology to generate new designs and artworks in *raster* image formats. However, these methods can neither guarantee the outputs can be considered aesthetically pleasing to people nor can they be further edited to meet design or production requirements (more information is described in section 2.4).

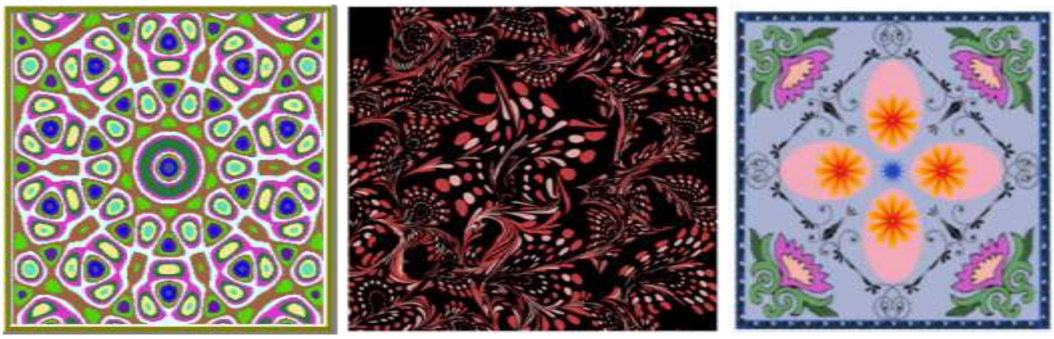


Figure 1-2 Examples of the generation results from Liu *et al.* (2008), Lu *et al.* (2014) and Wang *et al.* (2019), respectively

1.2 Research Problems and Objectives

As mentioned previously, the demand for digital patterns is increasing, adding more pressure on designers. Little work has been devoted to improve the decorative pattern design process and the existing CAD systems are rarely used to create patterns in digital format. Moreover, the generated designs or artworks by some research works can neither meet human aesthetic requirements nor are in the correct format for production or further editing. To overcome these limitations and provide valuable assistance, the outputs of the proposed system should preferably be editable. Furthermore, the output patterns should satisfy human aesthetics as much as possible so that designers can get useful results and their workload can be reduced.

It is known that majority of new designs are generated from editing the existing ones. This implies that the existing designs contain much useful information, including design elements, color combination rules, elements layout rules and so forth. In order to build intelligent design support systems, this study follows the development pathway of artificial intelligence to mimic how human designers utilize their creativity. Computers should learn from design samples first and then attempt to automatically generate design outputs that are pleasing to human aesthetics. To achieve such a goal, this study should tackle the following problems:

a) How could useful design information be analyzed and extracted from existing ones?
b) How could new designs that are pleasing to human aesthetics be generated by leveraging the obtained design information?
c) How could outputs be made easily editable to meet industry requirements?

By thoroughly reviewing the existing practices of design creation in the fashion industry, many digital patterns are found to be created by tiling smaller units called repeated patterns; and repeated patterns are composed of repetitive design elements in several colors. Repeated patterns can be found everywhere, for example, natural scenery, buildings and architectures, and design-related works, especially textiles. The repeated patterns represent the primary information of an image in a compact form. Therefore, detecting repeated patterns is beneficial for design understanding and analysis, especially the layout of design elements.

Considering digital patterns with several design elements, this study plans to go one step further to analyze the design at the design element level. This study proposes to identify and extract design elements inside existing designs. Simultaneously, in order to ensure the extracted design elements can be easily used for further design generation, it is plan to vectorize them and keep the outcomes compact.

Through analyzing the existing designs, a wealth of useful design information is obtained, including editable design elements and design rules such as colors and topology structures to support further design generation. Advanced artificial intelligence technology has brought new inspiration to solve computer vision problems and accelerated the development of CAD packages for digital pattern design. Therefore, this study seeks to employ artificial intelligence techniques to solve the stated problems and to fulfill the following research objectives:

i. To establish a systematic understanding on digital pattern design, especially layout rules and color combination rules;

ii. To comprehensively analyze pattern designers' workflow in the fashion industry and establish the structure of this study;

iii. To construct a digital pattern image dataset and classify them according to the layout, aiming to provide enough design resources for the study;

iv. To propose an efficient repeated pattern detection method for digital pattern analysis in repeated pattern level;

v. To propose a novel method for design elements extraction and vectorization from *unknown* digital pattern images for design analysis in design element level;

vi. To build a vector-based design element dataset for subsequent design generation;

vii. To extract color palettes from the given reference images according to requirements;

viii. To build a design-knowledge based digital pattern generation system covering general pattern types and ensure the outputs are editable as well as aligned with human aesthetics.

1.3 Framework Overview and Thesis Organization

Before establishing the overall framework and conducting experiments, this study should have a systematic understanding of digital pattern design, including its main components, common CAD software, and industry workflow. This is mainly explained in CHAPTER 1 and CHAPTER 2, and fulfilling objectives (i) and (ii).

According to the defined research objectives, this study proposes a REG framework composed of three intelligent systems: (1) efficient repeated pattern detection system (R-system), (2) design elements extraction and vectorization system (E-system), and (3) vector-based digital pattern generation system (G-system). The overview of the proposed framework is described in CHAPTER 3.

The R-system, realizing research objectives (iii) and (iv), is reported in CHAPTER 4. To achieve a good balance among content diversity, topology regularity and the trade-off between robustness and speed, the R-system firstly leverages activations of a pre-trained Convolutional Neural Network (CNN) to predict coarse repeated pattern size options of the input image. Accurate repeated patterns of the input images are then obtained by an optimization method based on template matching. Extensive experiments are conducted on the proposed dataset to demonstrate the superiority of the proposed R-system in repeated pattern detection. In addition, the outputs from the R-system can be input for the subsequent E-system for further design analysis.

The E-system, realizing research objectives (v) and (vi), is reported in CHAPTER 5. Vectorization is a process of converting raster graphics into vector graphics, which is common in design work and helpful for design editing and creation. Digital patterns are always composed of an arbitrary number of repetitive design elements. Those design elements with distinctly colored regions and sharp boundaries are particularly well suited for vectorization. Thus, the design elements should be extracted beforehand. Since the lack of datasets with labeled design elements, unsupervised segmentation is applied during core design element extraction. Next, a novel design element deconstruction method is proposed for vectorization based on color quantization to keep outcomes compact. Extensive experiments on collected design images demonstrate the effectiveness of the proposed E-system. Furthermore, a vector-based design element dataset is built using the proposed E-system from design images downloadable from internet, and these vector-based design elements can support new design generation in next phase.

The G-system, realizing research objectives (vii) and (viii), is reported in CHAPTER 6. In this study, digital patterns are classified into stripe, check, and motif patterns. As known, majority of new designs originate from modification of existing design samples. The previous two systems help us obtain useful design information from a large volume of digital pattern samples at different levels, including vector-based design elements and design rules such as colors and topology structures. The G-system designs a parametric model for each type of pattern from geometric and color perspectives, and flexible combinations of the obtained/generated vector-based design elements and color information are allowed. Implementation results show that the outcomes meet basic human aesthetics, and hence our method is able to support the design work.

In CHAPTER 7, the research work of the whole study is concluded, highlighting the contributions and key findings. Recommendations for future work are given. An illustration of thesis organization is shown in Figure 1-3.

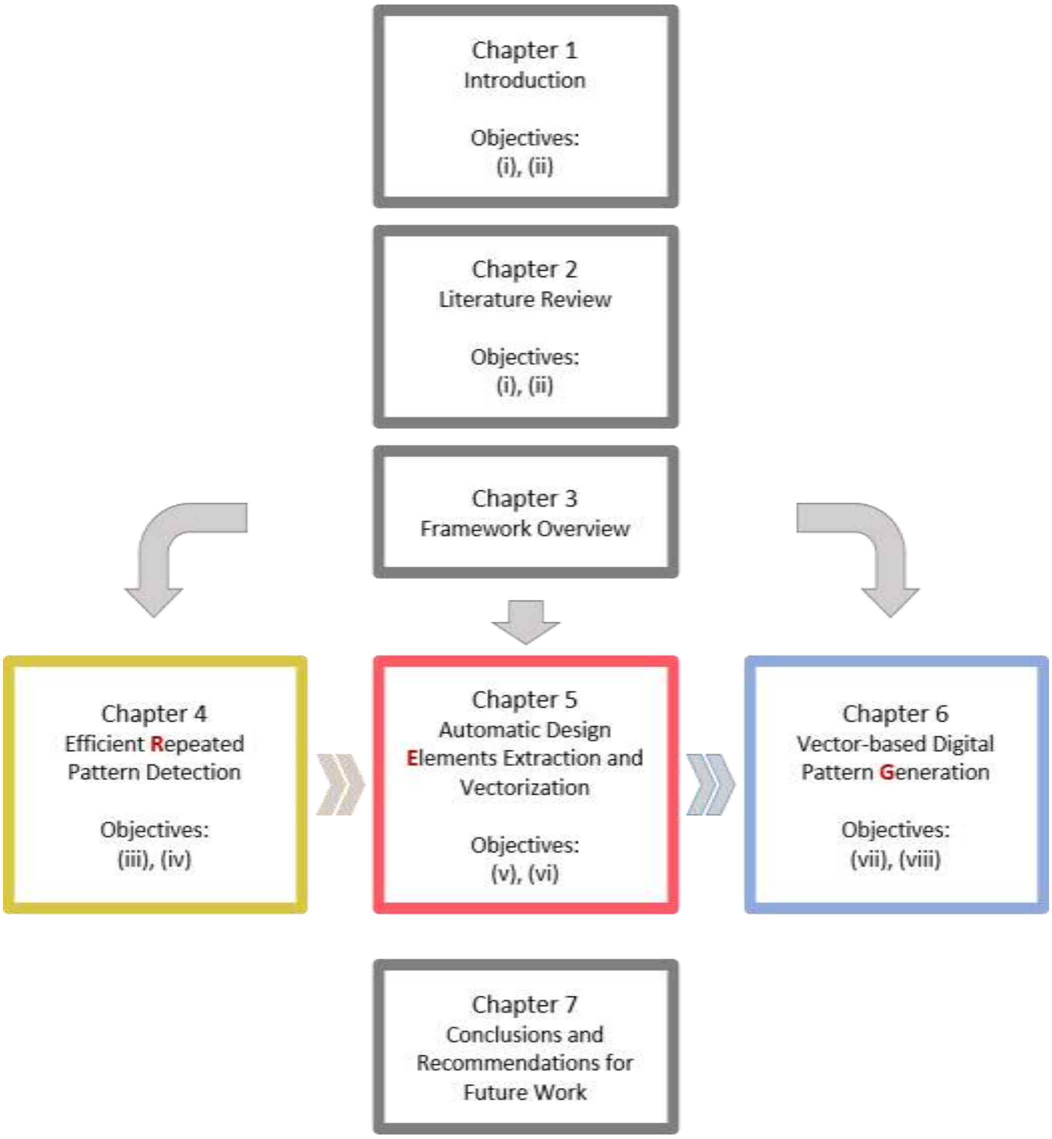


Figure 1-3 Thesis organization

1.4 Significance of the Study

This study is significant to both academia and the fashion industry. With digital pattern analysis and design support, this study proposes a novel REG framework consisting of three intelligent systems to resolve specific research problems. For academia, this study indicates three innovative solutions to tackle the stated problems. For the fashion industry, the results from this thesis are applicable to assist and improve the processes for creating digital patterns, which will improve the development of fashion products.

1.4.1 Significance to academia

This study proposed a novel hybrid method to effectively predict accurate repeated pattern sizes of unknown images. Instead of solely relying on local image features or CNN activations, the study first leverages selected CNN activations to predict the coarse repeated pattern size of every unknown image and then optimizes the size by the autocorrelation-based template matching method. The results of image boundary detection are used to screen CNN activations. Experiments on the proposed dataset with 841 images demonstrate our approach achieves the best trade-off between detection accuracy and efficiency than related works. In particular, the accuracy of our method is 0.673 which is 20% higher than the baseline method and the time cost is only its 11%. At the same time, the robustness of the proposed method is proved by experimenting with fronto-planar real-world images. Moreover, the ability to extract feature information using pre-trained CNNs is discussed and analysed in detail.

The study proposed a novel framework to extract and vectorize design elements inside unknown design images for the first time. The framework marks the first effort to concentrate on extracting design elements using unsupervised segmentation. Particularly, this method uses the feature extraction ability of CNN and joint learning to continuously optimize the final extraction results, overcoming the limitation of lacking of labeled design datasets. Extensive experiments demonstrate that the proposed method outperforms the traditional image segmentation and object detection methods. Additionally, a novel design element deconstruction method is proposed for vectorization based on color quantization: when combined with the classic vectorization approach for black-on-white images which huge number of meaningless vector paths are eliminated, the framework can produce compact vectorization results for color images. Experiments have shown that our method can effectively process one image in 13 seconds, and the output vectorized design elements are compact and good for further editing. Moreover, a vector-based design element dataset is built to support design generation.

This study proposed the first vector-based digital pattern generation system, covering stripe, check, and motif patterns. Compared with pixel-based image generation systems, the outputs of the proposed G-system have many advantages, including scalability, compatibility, simplicity of editing, and minimal file size. Moreover, the variety of patterns generated in the fashion domain is by far the richest. Apart from that, the generated results are controllable and ensured of compliance with design knowledge. Thus, the outputs can be further edited, supporting design process which is iterative nature. Extensive experiments demonstrate the outputs align with basic human aesthetics and further prove the approach is capable of assisting fashion design work.

What's more, the proposed REG framework promotes the use of artificial intelligence for design generation and eases the designers' or developers' burden in fashion product developments.

1.4.2 Significance to the industry

The demand and stress on product development keep rising day by day. Designers and even the whole industry are yearning for better CAD technologies to assist them in reducing design workload and providing more inspiration. The proposed framework is able to assist design work in providing different levels of results and producing vector-based digital patterns that align with basic human aesthetics from analysing design samples. In particular, the outputs of the R-system and E-system consist of repeated pattern level and design element level results. These results are beneficial for understanding a design and simplifying the design process. Moreover, the G-system is designed to quickly create new patterns that meet higher aesthetic requirements by simply tweaking and modifying vector-based digital elements. Furthermore, these systems can provide customers, who are mostly inexperienced with design (non-professional users), a tool for expressing their ideas for design. It encourages user involvement in the design process, adding value to the fashion products and improving sales.

The rest of this thesis is organized as follows: In CHAPTER 2, related works are reviewed, including previous works on computer-aided fashion design, brief review of computer vision, brief account of convolutional neural network, and generative tasks vs design generation. In CHAPTER 4, the efficient repeated pattern detection system is described. In CHAPTER 5, a system of design elements extraction and vectorization will be explained. In CHAPTER 6, a vector-based digital pattern generation system covering multiple pattern types and following general design rules will be explained. The experimental results will be presented in corresponding chapters. Finally, CHAPTER 7 concludes the research work of this study and discusses future work.

CHAPTER 2. Literature Review

This study aims to propose and develop intelligent systems for digital pattern analysis and design support, improving digital pattern design efficiency and enabling non-professional users to take part in the design process. Based on the research objectives defined in CHAPTER 1, this chapter reviews the related work as follows. First, the development of computer-aided design systems, supporting different functions of the fashion product development process, are reviewed at the beginning. The field of study on how computer understand the contents of an image is called computer vision. Since all design samples to be analyzed are in form of raster images, the basic concepts and approaches of key tasks of computer vision are also reviewed, forming the foundation of research work proposed in this study. Apart from reviewing traditional image features and typical computer vision tasks, advanced methods in computer vision are all based on deep neural networks, therefore the standard operations and typical architecture of CNN are introduced. Subsequently, the state-of-the-art generative adversarial network (GAN) methods are reviewed and the gaps for design generation research are discussed. The related research status on digital patterns and the current limitations are summarized in the end.

2.1 Computer-aided Fashion Design

CAD systems have been widely applied in fashion product design because they largely improve the efficiency to meet the pressing demands on consumer-focused product development (Bertolotti *et al.*, 2004; Sayem *et al.*, 2010). CAD systems are used to support the following typical functions or tasks in the design process: digital pattern design, garment style design, and garment pattern making.

2.1.1 Digital pattern design

In digital pattern design, CAD systems include two-dimensional (2D) graphics software, such as Adobe Illustrator (Adobe, 2022) and CorelDRAW (Bouton, 2017), or software that have been customized for the fashion industry like Kaledo Style (Kaledo, n.d.) and TexDesign (Koppermann, n.d.). These software are known for their guaranteed speed and accuracy. The digital formats (mainly in vector formats) of design outputs from these software have a number of advantages, including the designs are scalable, reusable, and can be easily adjusted to fit any size requirements for production. Unfortunately, these software only work as a substitute for pencil and paper rather than creating/generating new designs to reduce the workload of designers.

Some research studies have been reported enabling computers to generate editable new designs by incorporating existing design rules. For example, by investigating uniform stochastic web and repeated patterns, Liu *et al.* (2008) formulated a number of mathematical models for textile pattern generation. They applied the principle of repeated pattern evolution to textile pattern design and transformed previously abstract mathematical information into visual textile patterns. Lu *et al.* (2014) proposed the first and the only evolutionary marbling textile design system using mathematical marbling functions. It applied productive, deductive, and inductive philosophical models of design reasoning to depict the design process in forms of artificial evolution. The output marble patterns fulfill the textile industry requirements, in vector format, with various repeat arrangement and in different colorways. Wang *et al.* (2019) analyzed the basic principle of fractal artwork generation and the requirements of graphics. On this basis, they generated flower and geometric art graphics and then applied them to digital fashion pattern design. These methods encourage creativity in digital pattern design and accelerate fashion product development (examples are shown in Figure 2-1). However, the above methods can only produce patterns of a particular style, and the results are not always aligned with human aesthetics.

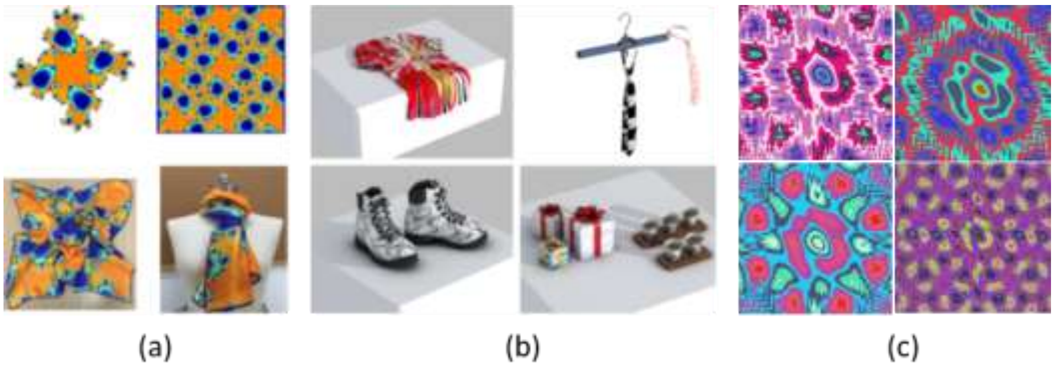
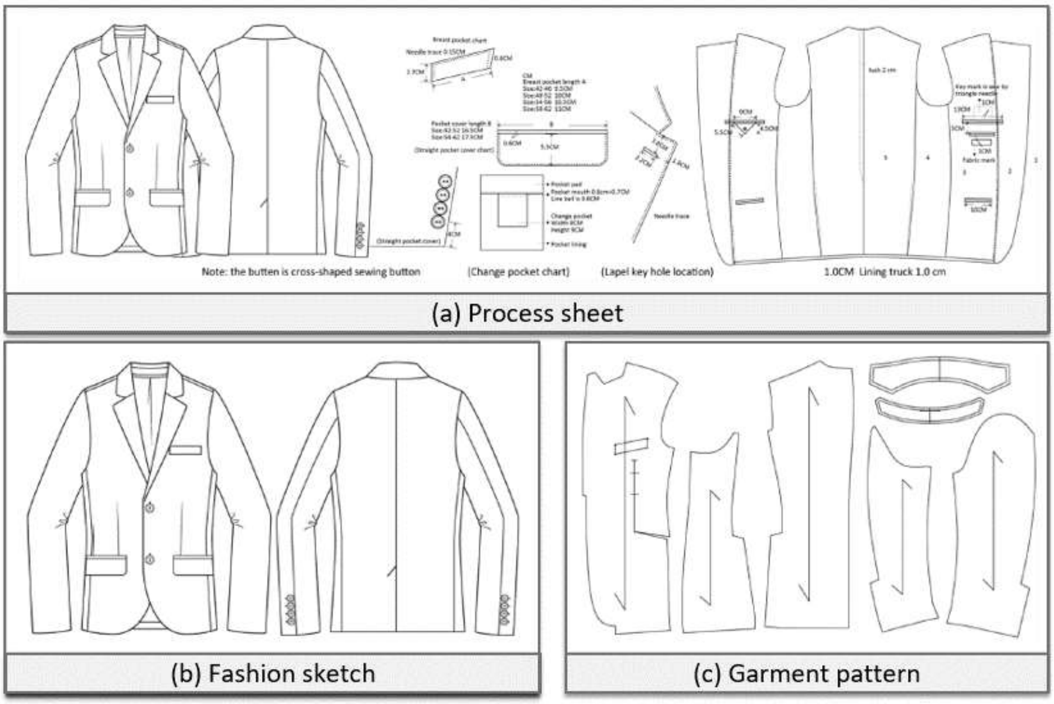


Figure 2-1 Examples of the outputs from the mentioned research works. (a) from Wang *et al.* (2019), (b) from Lu *et al.* (2014), and (c) from Liu *et al.* (2008)

2.1.2 Garment style design


Note: the butten is cross-shaped sewing button
(Change pocket chart)
(Lapel key hole location)
1.0CM Lining truck 1.0 cm

(a) Process sheet

(b) Fashion sketch

(c) Garment pattern

Figure 2-2 Fashion process sheet, fashion sketch and garment pattern (Liu *et al.*, 2019)

Garment style design is usually expressed utilizing *fashion sketches* (see Figure 2-2(b)) in garment development process. However, not much work has been done on automatic creation of fashion sketches. Commercial CAD software being available for sketch drawing is similar to that for digital patterns, i.e. mainly Adobe Illustrator and CorelDRAW are used. Fashion sketches are highly technical, and it still relies on tedious and manual process to develop every new style. Regardless drawing manually with pencil and paper or using design software, the design process of garment styles is skill-based and it takes years or months of training to manage the skills of style design.

In recent years, empowered customers have a stronger desire to take part in the creation process of fashion products. To this end, a few interactive CAD systems were proposed and developed, following similar approach of first identifying components or constituent parts of fashion products and then combining different parts together to form new designs. For instance, Ogata *et al.* (2007) proposed a jacket design support system: through deconstructing the jacket into parts i.e., a body, a collar, a sleeve, pockets, material and color of a jacket and then applying the Interactive Genetic Algorithms (IGA), users' preferences are allowed to be reflected in garment design. Users can review different jacket design options created by the system and select their preferred ones. The system then provides genetic algorithm (GA) operations of selection, crossover and mutations, to generate new jacket designs based on user judgment (see Figure 2-3(a)). Satisfactory jacket designs are obtained by repeating the above steps. Mok *et al.* (2013) developed a customized fashion skirt and dress design system for customers to create design sketches in a user-friendly interface (see Figure 2-3(b)). In particular, they describe a skirt from three levels: silhouette, key style elements and design details. Each level has a number of options. The system allows options from different levels to be freely combined to generate new skirt or dress designs. In addition, Liu *et al.* (2019) developed a system (Figure 2-3(c)) to assist the creation of jean designs, outputting fashion sketches with corresponding sewing patterns based on six parametric inputs that reflect the shape of human body and design requirements of users. The above-mentioned systems are able to help non-professional users to create or customize their fashion designs/sketches of specified categories in a user-friendly way. Their idea of breaking down fashion sketches into constituent parts and reorganizing them into new designs inspired the research about digital pattern design in this study.

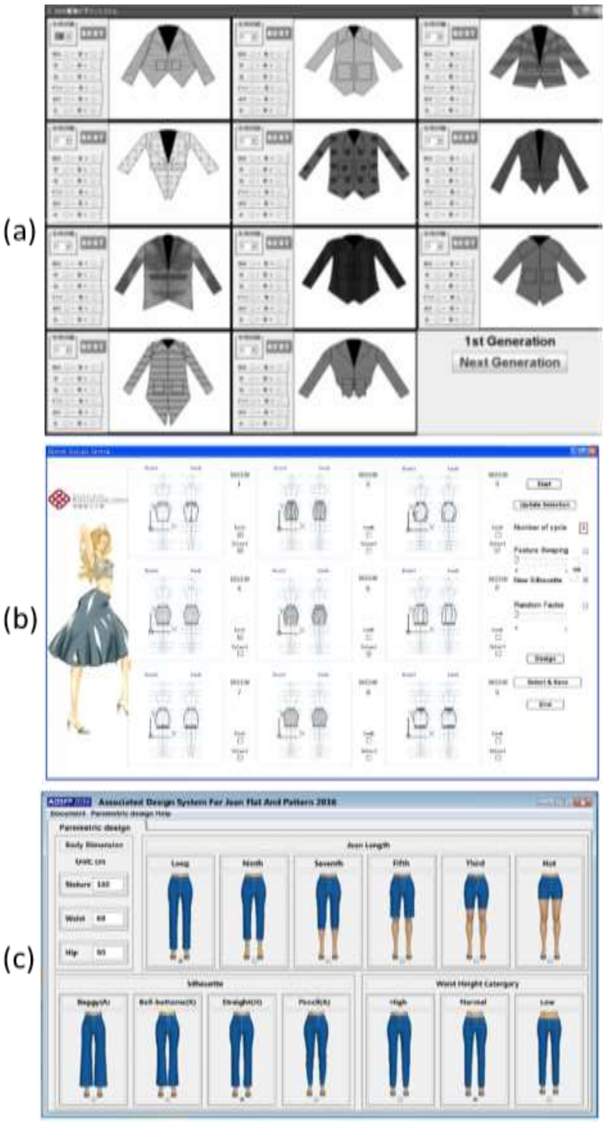


Figure 2-3 Examples of garment style CAD systems. (a) User interface of Ogata *et al.* (2007), (b) user interface of Mok *et al.* (2013), and (c) user interface of Liu *et al.* (2019)

2.1.3 Garment pattern making

Garment patternmaking is also called garment construction design (see Figure 2-2(c)). Traditionally, according to a fashion sketch, pattern makers create and draw on paper garment patterns, which are the blueprint on the basis of which fabric is cut (Aldrich, 2013; Liu *et al.*, 2018), and these pattern pieces are sewn together to form a finished 3D garment, which has the exact styles/design of the fashion sketch. The work of patternmaking is therefore vitally important, linking from fashion design to garment making. A range of commercial tools are developed, such as Optitex PDS (Optitex, n.d.), CLO (CLO, 2022), and Gerber AccuMark (Lectra, n.d.) to support garment pattern design process.

A great deal of CAD software and research work have been focused specifically on virtually sewing 2D clothing patterns on 3D mannequins to assess the appearance and fit of the finished 3D garments (an example is shown in Figure 2-4), aiming at shortening the sample making and fitting cycles. This is known as 3D clothing simulation (Li *et al.*, 2017; Liu *et al.*, 2019; Meng *et al.*, 2012). These works of garment pattern design follow a 2D-to-3D approach. Another area of garment patternmaking CAD studies focus on designing clothing directly in 3D space and then flattening 3D garments to 2D patterns (Bang *et al.*, 2021; Liu *et al.*, 2018). This approach mimics the 3D modeling and draping in fashion design to obtain 2D patterns, following a 3D-to-2D approach. Nevertheless, flattening complex-shaped 3D clothes is technically difficult to be realized in computer simulation, due to the folds and gathers produced are non-developable surfaces (Huang *et al.*, 2012). Regardless 2D and 3D-based CAD systems, they are designed for experienced users, such as pattern makers or computer professionals who have a thorough understanding of 3D modeling. The outputs of these systems are editable clothing patterns often in dxf formats.

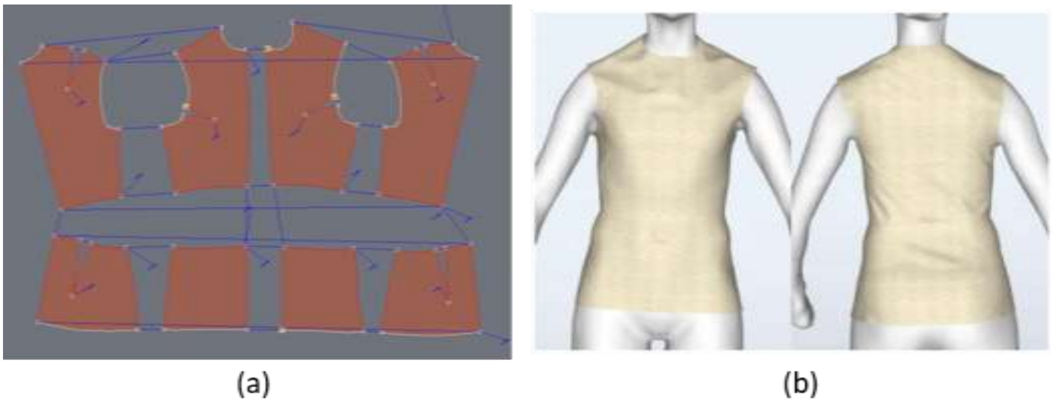


Figure 2-4 Example of 2D-to-3D virtual try-on. (a) 2D clothing patterns of a top garment; (b) Patterns are sewn on a 3D virtual model (Mosleh *et al.*, 2021)

Usually, garment pattern making and digital patterns/fabric design are two separate processes. When using existing digital patterns/fabric for garment production, professionals need to pay extra attention to the placement of the garment pattern when cutting the fabric to ensure that the digital pattern is matched at the seams of the garment (Vilumsone-Nemes *et al.*, 2020). Lu *et al.* (2017) proposed a novel system that combines the design processes of digital patterns/fabric design and garment designs. In this system, fashion designs are completed first to obtain a distortion-free 3D garment model, on top designers can paint their digital patterns, and the 2D clothing pattern pieces are obtained with matched textures that are ready for digital printing and garment production. An example is illustrated in Figure 2-5, the textures of the garment model, a close-fit top, are designed by 3D painting. The outcomes of this method are ensured of textural continuity in pattern and garment pieces, and fabric utilization is high saving the cost for matching fabric textures and arrangement for production. Additionally, this work demonstrates the potential of direct application of digital pattern design in garment production.

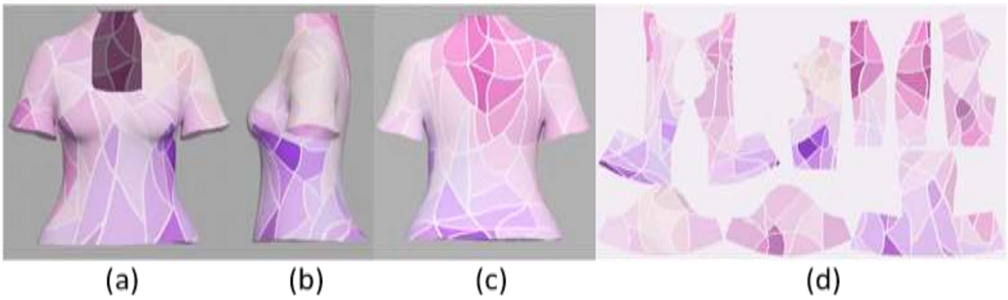


Figure 2-5 Finished 3D painting of a blouse model. (a–c) Front, side, and back views of the blouse, and (d) the final sewing pattern pieces (Lu *et al.,* 2017)

2.2 Brief Review of Computer Vision

For presentation and communication purposes, most CAD systems output designs in digital format of raster images. On the other hand, there are thousands of existing fashion images shared over the internet and social media. Nowadays, people enjoy recording interesting scenes or designs using input devices like cameras, mobiles or scanners, and all these visuals are recorded as digital images. Therefore, digital images are the most common means to record designs. The research objective of current study is to develop intelligent systems that are able to generate new designs by learning from previous examples (design images). This objective indeed aligns with that of computer vision domain, which also aims to enable computers to understand and deduce meaning from input images. The techniques and methods of computer vision will be investigated and utilized to develop intelligent design systems, and therefore field of computer vision study is briefly reviewed here.

2.2.1 Images features

In computer vision, a feature is a piece of information that is important for completing the required computations for a specific application. The feature notion is quite generic, the choice of features in a particular computer vision system may be significantly influenced by the particular application (Nixon *et al.*, 2019). Features may be shapes defined in terms of curves or boundaries between different image regions, properties of such a region, or patterns of an object in an image that help to identify it. Therefore, extracting good visual feature is important for representing an image compactly. The three primitive visual features are color, texture, and shape (Hiremath *et al.*, 2007; Srivastava *et al.*, 2015).

2.2.1.1 Color features

Color can convey emotional symbols in graphic design and have a significant impact on people's psychological activities. Therefore, the use of colors is important in design work, such as packaging design and website design (Hembree, 2006; Hurley *et al.*, 2017; Shan, 2018). In fashion product development, color helps enhance the attractiveness of the design without adjusting the shape or silhouette (Khajeh *et al.*, 2016). For example, relatively dark colors or those in harmony with the environment can be used to cover physical defects.

The nature of an image on a digital screen is pixel values representing different colors. Color feature is one of the most vital, reliable, and widely used image features in computer vision (Wang *et al.*, 2005). Color features are utilized to support computer vision tasks of image retrieval, segmentation, and classification (Chai *et al.*, 2000). Color, as an identifiable feature, is computed in terms of different color spaces, and color feature is independent of view and resolution (Swain *et al.*, 1991). Common color spaces include RGB, CIE, L*a*b*, and HSV (Chen *et al.*, 2008). In RGB color space, each pixel in an image comprises three color channels known as RGB components (Dutta *et al.*, 2009). Each of these components has a value ranging from 0 to 255. Although RGB is widely used standard, it is not close to human perceptions. The HSV (Hue, saturation, value) color space is proved to have better results in computer vision tasks, such as image segmentation, than RGB color space. It is capable of emphasizing human visual perception in hues and can be easily inverted or transformed from RGB (Chen *et al.*, 2008).

Based on the chosen color space, color features like color moments (CM) and color histogram, can be extracted. CM is color feature in the simplest form, and this feature describes colors by color mean (u_k), standard deviation (σ_k) and skewness (γ_k), which are calculated as follows:

$$u_k = \frac{1}{N}\sum_{j=1}^{N} f_k(x,y) \qquad (2\text{-}1)$$

$$\sigma_k = \sqrt{\left(\frac{1}{N}\sum_{j=1}^{N}(f_k(x,y) - u_k)^2\right)} \qquad (2\text{-}2)$$

$$\gamma_k = \sqrt[3]{\left(\frac{1}{N}\sum_{j=1}^{N}(f_k(x,y) - u_k)^3\right)} \qquad (2\text{-}3)$$

where N is the total number of pixels in an image and $f_k(x,y)$ is the value of the k-th color channel of the image pixel located at the (x,y) coordinate.

Color histogram is a way to represent the distribution of colors within an image, and each histogram represents a color by grouping pixel of similar values into a ‘bin’ along a color space (Hafner *et al.*, 1995). They count similar pixels and store them: the Y-axis represents the number of pixels, and the X-axis represents a fixed color range (Mohamad *et al.*, 2010; Stricker *et al.*, 1995). The formulas are shown below:

$$h[l] = n_l \qquad (2\text{-}4)$$

$$N = \sum_{l=1}^{L} h[l] \tag{2-5}$$

where n_l represents a color in a discrete color space, $h[l]$ represents the number of pixels of color n_l and N is the total number of pixels in an image. When an image is represented by a color histogram $H(N)$, this color feature is abstracted into a long vector for easy calculation.

The above two color features are often used for different computer visions tasks. For example, color histograms are frequently utilized for image retrieval and similarity comparison, since they are invariant to translation and alter only slightly under various viewing angles, scales, and even in the presence of occlusions (Hafner *et al.*, 1995; Stricker *et al.*, 1995). For comparison of similarity between images, the common methods include histogram intersection and counting the distance between CMs (Mohamad *et al.*, 2010). The histogram intersection algorithm was proposed as "Color Indexing" (Swain *et al.*, 1991). The similarity between two images is represented by the histogram intersection distance (Meskaldji *et al.*, 2009; Mohamad *et al.*, 2010; Srivastava *et al.*, 2015). Given the histogram H of the input image and the histogram H' of the image for comparison, each histogram containing L bins, the intersection is defined as:

$$\sum_{l=1}^{L} min(H_l, H_l') \tag{2-6}$$

In equation (2-6), the min function takes two histograms as inputs and returns the smallest one. The result of the intersection is the number of pixels from the compared image that have corresponding pixels of the same colors in the input image. The result obtained by equation (2-6) is often normalized by dividing the number of pixels in the model histogram as follows:

$$H_{norm}(H,H') = \frac{\sum_{l=1}^{L} \min(H_l, H_l')}{\sum_{l=1}^{L} H_l'} \qquad (2\text{-}7)$$

Although the color feature is ubiquitous and effective in computer vision, the image colors appear enormous and chaotic in the color space, which causes trouble during accurate extraction. To address this problem, researchers proposed *color reduction* scheme by clustering color features/pixels within an image. Srivastava *et al.* (2015) used K-mean clustering algorithm to reduce the number of colors. In their work, the image contains *k* cluster centers and each pixel in the image whose value will be replaced with the closest cluster center. Similarly, Nikolaou *et al.* (2009) clustered pixels in an image into a number of colors by the Mean-Shift algorithm. Unfortunately, the speed of this method is very slow. Papamarkos *et al.* (2002) developed a color reduction scheme using a tree clustering procedure. These color reduction methods are often used as a pre-processing step to improve the performance of the models (Dong *et al.*, 2005).

2.2.1.2 Texture features

Texture is a feature that separates images into areas of interest as well as classifying those separated areas. Texture features are being used in a vast number of applications and computer vision tasks, for example, image classification (Humeau-Heurtier, 2019), image segmentation (du Buf *et al.*, 1990), image synthesis, or pattern recognition (Tan *et al.*, 2010).

In addition to providing information on the spatial arrangement of colors or intensities of an image, texture also characterizes the spatial distribution of the intensity levels of an image (Wirth, 2004). Most texture features are not influenced by translation, rotation, affine and perspective transforms, and they can distinguish certain information that the color features would have them ignored. To vividly explain the texture features, Figure 2-6 shows an example of three different images with the same intensity distribution but with different textures.

There are many ways to extract texture features. Humeau-Heurtier (2019) has categorized texture feature extraction methods into seven classes. Among them, the representative methods include the Gabor filtering approach, the Grey Level Co-occurrence Matrix (GLCM) statistical approach, and Local Binary Pattern (LBP) structural approach.

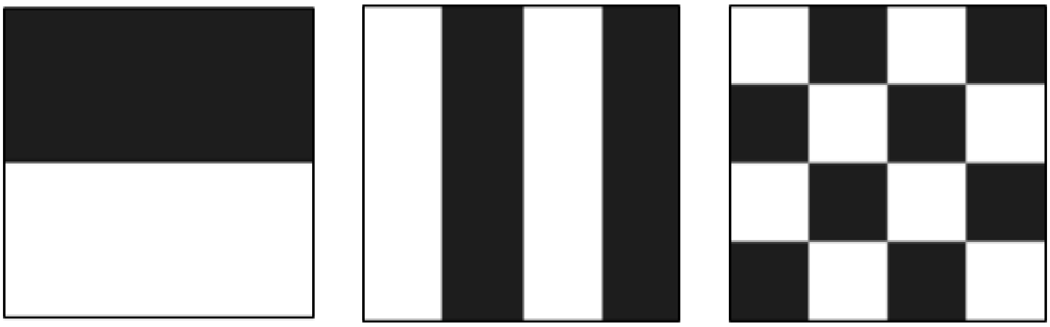


Figure 2-6 Examples of different textures: Each image above has a 50%-50% distribution of black and white pixels but of different textures (Wirth, 2004)

2.2.1.3 Shape features

Shape descriptor is a set of numbers, representing a given shape feature and quantifying the shape in a way that make sense to people (or task-specific requirements). It can then be applied to classify, recognize, align and retrieve images. Efficient shape descriptors should represent the complete content of the shape as much as possible, be compactly stored and easy for comparison. Usually, shape descriptors are stored as vectors. Previously, they are broadly classified as the contour-based methods using the shape boundary points and region-based methods using shape interior points (Zhang *et al.*, 2004). Yang *et al.* (2008) gave a good summary of many new techniques being proposed in recent years, as shown in Figure 2-7.

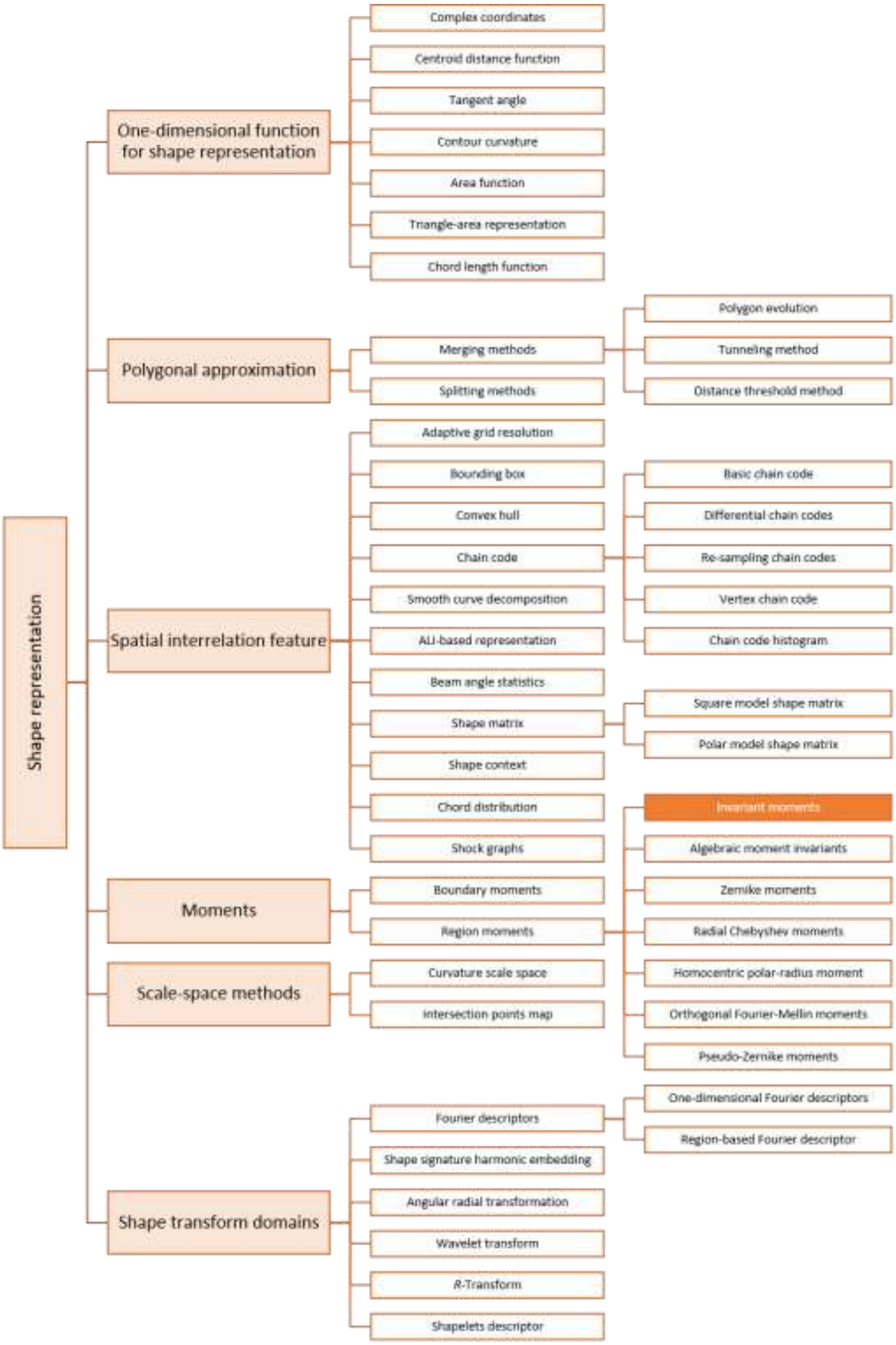


Figure 2-7 An overview of shape description techniques (Yang *et al.*, 2008)

Wavelet transform, Fourier shape description, and invariant moments are commonly used in many computer vision tasks. Invariant moments, also known as geometric moment invariants, will be used in CHAPTER 5 for design element extraction. Invariant moments are particular forms of moments. A moment function m_{pq} of an image intensity function $f(x, y)$ can be defined as follows:

$$m_{pq} = \sum_{x}\sum_{y} x^p y^q f(x, y) \tag{2-8}$$

where $x^p y^q$ is known as the moment weighting kernel or the basis set, and $p, q = 0,1,2 \cdots$.

Geometric central moments are invariant to translation, they are defined as:

$$\eta_{pq} = \sum_{x}\sum_{y} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \qquad p, q = 0,1,2 \cdots \tag{2-9}$$

where $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$, and $p, q = 0,1,2 \cdots$.

With this standard function representations, Hu (1962) proposed seven invariant moments based on geometric central moments, which are invariant to rotation, scaling, and translation, and Table 2-1 shows the detail function definitions. Although invariant moments are computationally simple, they are sensitive to noise, especially for higher-order moments; and are unstable to have a good shape representation when the image size is large (Celebi *et al.*, 2005; Yang *et al.*, 2008).

Table 2-1 Formulas of invariant moments

1.	$\phi_1 = \eta_{20} + \eta_{02}$
2.	$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4{\eta_{11}}^2$
3.	$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$
4.	$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$
5.	$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$ $+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$
6.	$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$ $+ 4{\eta_{11}}^2(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$
7.	$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2]$ $+ (3\eta_{12} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

2.2.1.4 Feature descriptors

Image features cover both global and local features. Global features consider the image as a whole, whereas local features describe the image patches (a small group of pixels) (Kabbai *et al.*, 2019). Local feature descriptors are widely employed in computer vision tasks, such as image retrieval and image alignment since they have low computation costs and are robust (Dewan *et al.*, 2020; Karami *et al.*, 2017; Ke *et al.*, 2004). Famous local feature descriptors include Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) (Bay *et al.*, 2006), Features from Accelerated Segment Test (FAST) (Rosten *et al.*, 2006), Histogram of Oriented Gradients (HoG) (Dalal *et al.*, 2005), Binary Robust Independent Elementary Features (BRIEF) (Calonder *et al.*, 2010), and Oriented FAST and Rotated BRIEF (ORB) (Rublee *et al.*, 2011).

The powerful SIFT descriptor was first presented by Ng *et al.* (2003) for local interest points extraction and is well-known for its robustness to object rotation and scale variations. It combined the difference-of-Gaussian (DoG) detector and histogram of gradient (HoG) descriptors. The SIFT algorithm consists of four main steps. Firstly, it uses DoG to build an image pyramid space and then detects potential interest points under the scale space. Secondly, candidate interest points are localized to sub-pixel accuracy and eliminated if found to be unstable. Thirdly, it assigns an orientation by the peak of gradient distribution of neighbour pixels to each interest point, making it rotation invariant. The assigned orientations, scale, and location for each interest point enable SIFT to create a canonical view of the interest points that is resistant to similarity transforms. Finally, a local image descriptor for each interest point is built based on its local neighbourhood's image gradient. For evaluation of similarity between two images using SIFT features, this is done by matching their corresponding interest points.

Traditional image features are well-established, transparent, and very general, and they can be applied to any image. In addition to general features, hand-designed feature extractors are often developed for specific computer visions tasks, and the design of such hand-crafted features require a considerable amount of engineering skills and domain expertise.

2.2.2 Image classification

Image classification is a major computer vision task, and it assigns one or multiple labels to an image based on the image content (Akata *et al.*, 2013). The labels are usually numerical values that indicate the category of the image content, such as “bottle” or “cup”; an example is shown in Figure 2-8(a). Image classification is important as it also supports other computer vision tasks, such as recognizing objects, detecting objects and tracking objects in real-time (Bolon-Canedo *et al.*, 2020). Given an image can be described as a set of image features, images of the same class have similar content and feature properties. In the early image recognition systems, researchers used handcrafted features for classification, such as the aforementioned scale-invariant feature transform (SIFT) (Ng *et al.*, 2003) and histogram of oriented gradients (HoG) (Dalal *et al.*, 2005). For processing images efficiently, feature extraction is common technique. They are usually focused on the extraction of relevant image properties which may include shape, color, texture, spatial information, etc. (Bolon-Canedo *et al.*, 2020). The extracted features significantly affect the performance of the classification systems in specific applications. Therefore, definition of effective image features is the focus of the research efforts, and researchers carefully consider the characteristics of images in a specific class to design better image features, thereby improving system performance (Al-Saffar *et al.*, 2017).

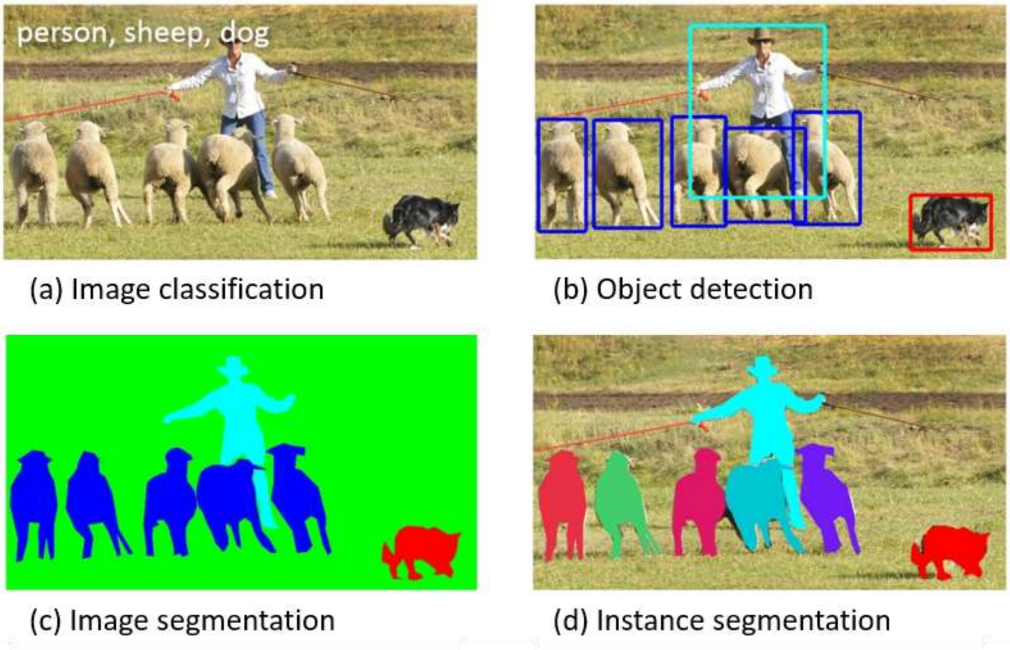


Figure 2-8 Different computer vision tasks for understanding image content (Lin *et al.*, 2014)

In the past decade, deep neural networks have received much research attentions because they are good at extracting features as well as establishing relationships between the extracted features. Deep neural networks combine feature extraction and model optimization in one system, and the development of these deep neural network is also called deep learning because they enable computers to *learn* from data to discover the relationships/rules for specific task. Deep learning can be classified as supervised learning, unsupervised learning and reinforcement learning. In the case of image classification, if the network models are trained to learn from image data with known class labels, this is called supervised learning. Supervised image classification is one of the earliest computer vision applications of deep learning, and it was most influential because supervised image classification has astounding performance suppressing traditional methods with very large margins or even humans. For supervised image classification models, convolution neural networks (CNN) were commonly used, which will be described in Section 2.3.

2.2.3 Image segmentation

Image segmentation is the process of dividing an image into different regions that are coherent according to certain criteria (Cheng *et al.*, 2001; Egmont-Petersen *et al.*, 2002). Initially, researchers never attempted to make it understand what the segmented regions represent; numerous image segmentation methods leveraging low-level image features have been developed. These methods are classical methods and could be further classified into thresholding (Bhargavi *et al.*, 2014), edge-based (Narkhede, 2013), region-based, clustering-based (Dong *et al.*, 2005; Naik *et al.*, 2014; Naz *et al.*, 2010), and watershed-based (Bleau *et al.*, 2000). The other approach is the learning-based methods. The hybrid techniques refer to those when classical methods are combined with other specific tools. Figure 2-9 shows the classification of image segmentation methods.

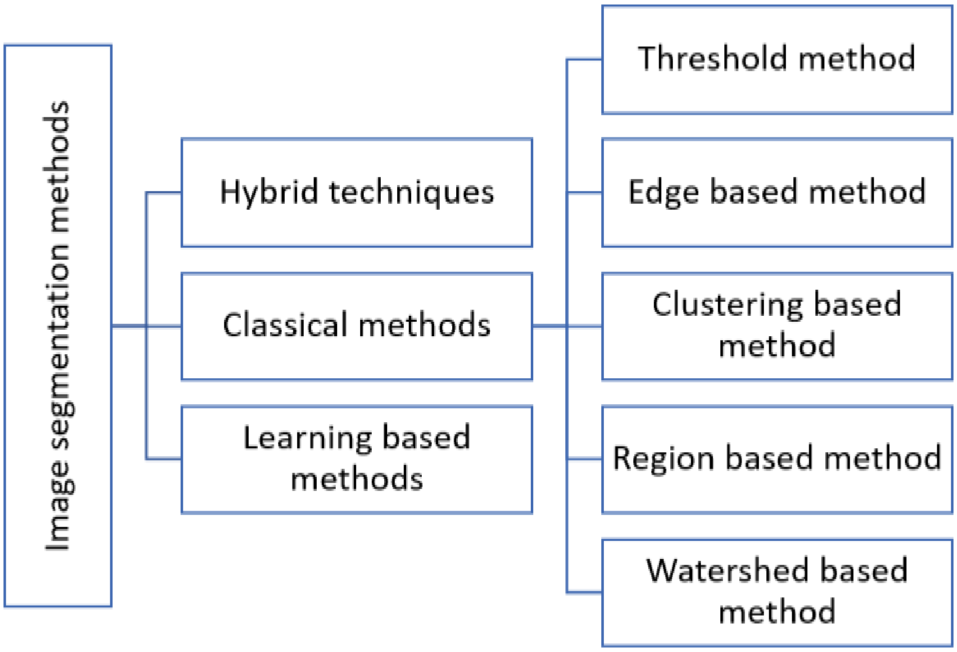


Figure 2-9 Classification of image segmentation methods (Yuheng *et al.*, 2017)

Felzenszwalb *et al.* (2004) proposed an efficient graph-based segmentation method, which is the state-of-the-art method following traditional approach. Their method is often applied to various computing tasks today as well as in this study. It will be used in the E-system (described in CHAPTER 5) for unsupervised segmentation. Their method first defines a predicate, D, for evaluating whether or not there is evidence for a boundary between two regions. The predicate is based on comparing the dissimilarity between adjacent elements within each of the two components, to the dissimilarity between elements along the boundary. Then, the internal difference of a component $C \subseteq V$ is defined as:

$$Int(C) = \max_{e \in MST(C,E)} w(e) \qquad (2\text{-}10)$$

representing the largest weight in the minimum spanning tree of the component,

$MST(C, E)$. The difference between the two components $C_1, C_2 \subseteq V$ to be the minimum weight edge connecting the two components:

$$Dif(C_1, C_2) = {}_{v_i \in C_1, v_j \in C_2, v_i, v_j \in E}^{min} w\left(\left(v_i, v_j\right)\right) \tag{2-11}$$

The segmentation region is predicted only when there is evidence for a boundary between a pair of components. It is found by determining if the difference between the components, $Dif(C_1, C_2)$, is large relative to the internal difference within at least one of the components, $Int(C_1)$ and $Int(C_2)$. To apply this method, a threshold function is defined,

$$\tau(C) = k/|C| \tag{2-12}$$

where $|C|$ is the size of C and k is a constant parameter used to control the degree of the difference between components. The final predicate method is,

$$D(C_1, C_2) = \begin{cases} true & ifDif(C_1, C_2) > MInt(C_1, C_2) \\ false & otherwise \end{cases} \tag{2-13}$$

where the minimum internal difference $MInt$, is defined as, $MInt(C_1, C_2) = min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2))$.

Deep learning has consistently outperformed conventional methods in computer vision field, and it is now the norm for tasks like semantic segmentation. Semantic segmentation is the process of assigning each pixel to a particular label (see Figure 2-8(c)). Typical jobs are Fully Convolutional Network (FCN) proposed by Long *et al.* (2015) and U-Net proposed by Ronneberger *et al.* (2015). However, semantic segmentation cannot differentiate different instances of the same object. For example, there are many sheep in the image, as seen in Figure 2-8(c), and semantic segmentation would assign the same label to every pixel of these sheep. Instance segmentation differs from semantic segmentation and it assigns a unique label to every instance of a particular object in the image. Seen in Figure 2-8(d), sheep are assigned with different labels.

2.2.4 Object detection

The process of finding instances of specific classes of visual objects (such as people, animals, or cars) in digital images is known as object detection and is an important computer vision task (Zhao *et al.*, 2019; Zou *et al.*, 2019) (see Figure 2-8(b)). Early works are built on handcrafted features and could not identify objects with subtle semantic labels. The recent explosive surge in the usage of deep learning techniques has given object identification new life, resulting in amazing advancements and drawing unheard-of attention to it, making it a research hotspot. Object detection has now been widely applied in many real-world applications, such as robot vision, autonomous driving, and video surveillance (Zou *et al.*, 2019). Best-performing object detection techniques are supervised learning-based and typically learned from real-world images, like YOOLOv4 (Bochkovskiy *et al.*, 2020), Mask r-cnn (He *et al.*, 2017). In contrast, few works have focused on the extraction of meaningful objects in digital pattern images.

2.3 Brief account of convolutional neural network

Deep learning develops computational models with numerous processing layers to learn data representations at various levels of abstraction (LeCun *et al.*, 2015). In recent years, computer vision has achieved remarkable improvements based on deep learning technologies, especially on convolutional neural network (CNN). Convolution neural network is a multi-layer artificial neural network specially designed to handle images, that are mathematically represented as two-dimensional input data. Each layer in the network is composed of multiple two-dimensional planes, and each plane consists of multiple independent convolution filters (also named as kernels). For instance, a color image is made up of three 2D arrays, containing the pixel intensities for the three color channels. See Figure 2-10 for an example of pixel intensities array.

Inspired by the biological neural network, CNNs use a weight-sharing network structure, which can be adjusted by changing the depth and breadth of the network (Pak *et al.*, 2017).

The first deep CNN architecture is called LeNet, proposed by (LeCun *et al.*, 1998), including all CNN essentials of convolutional layers, subsampling layers, and fully connected layers (Ding, 2021). When it was first proposed, it did not gain much attention due to the lack of powerful hardware and labeled data (Wang *et al.*, 2019). As computer hardware advanced and the amount of data available grew in the subsequent years, AlexNet (Krizhevsky *et al.*, 2012) was proposed and won the ILSVRC-2012 image classification competition with astounding results in 2012. Thanks to its success, researchers showed growing interests in CNNs thereafter. Many CNNs such as Alexnet (Krizhevsky *et al.*, 2012), VGG (Simonyan *et al.*, 2014), GoogleNet (Szegedy *et al.*, 2015), and ResNet (He *et al.*, 2016) have been developed and used as building blocks for solving various practical problems.

Over the years, a trend is observed that the networks are getting deeper and deeper and with more complex structures. Nevertheless, the basic components of these CNNs are similar, mainly include convolutional layer, activation function, pooling layer, and fully connected layer (Liu *et al.*, 2015; Simonyan *et al.*, 2014; Xu *et al.*, 2019). The architecture of a typical CNN is shown in Figure 2-12.

The convolutional layer contains a set of *filters,* and each filter is associated with a set of weights. Their function is to perform a convolution operation between these filters and the input arrays; specifically, an element-wise product between each element of the filter and the input array is calculated at each location of the array and summed to obtain the output value in the corresponding position of the output array to create *feature maps*. An illustration is shown in Figure 2-10. A typical size for filters is 3 × 3, sometimes 5 × 5 or 7 × 7. In the described convolution operation, a filter would pass more times in the middle part of the image than along the edges. The resulting features will have more knowledge about the middle part of the image than the edges, and some important features at the edges may be lost. To address this issue, researchers proposed padding. Typically zero padding will be used, where rows and columns of zeros are added to each side of the input array, keeping the same in-plane dimension in the convolution operation. The distance between two successive convolution operations is called a stride, the common choice of a stride is 1. In modern CNN architectures, padding and stride are used to control or retain the in-plane dimensions of input array and output feature (see Figure 2-10). The output feature maps of convolution operations will pass through a non-linearity function, also named *activation function* (Mechelli *et al.*, 2019), and typical activation function is Rectified Linear Unit (ReLU).

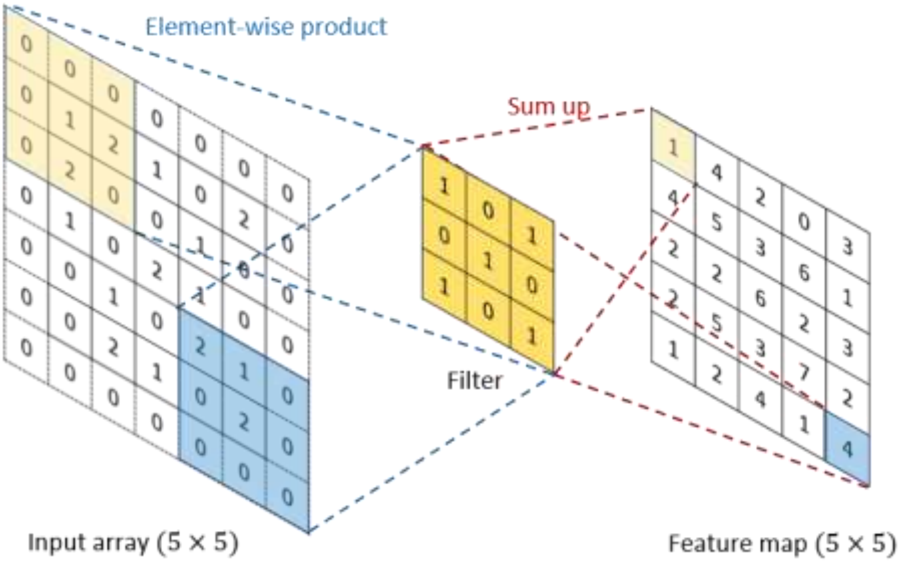


Figure 2-10 A convolution operation with zero padding to keep the in-plane dimensions. It should be noted that the output feature map maintains the input dimension

of 5 × 5. The kernel size and stride in this illustration are set to 3 × 3 and 1, respectively

A pooling layer reduces the in-plane dimension of the feature maps in order to introduce a translation invariance to small shifts and distortions, and decrease the number of subsequent learnable parameters (Yamashita *et al.*, 2018). Therefore, pooling layers are viewed as downsampling operation. Max pooling and average pooling are typical operations of pooling layers (Mechelli *et al.*, 2019). Max pooling is a mathematical operation that takes the largest value from a portion of the array of a specific size, whereas average pooling is a mathematical operation that takes the average value of a portion of the array of a specific size. Figure 2-11 shows an example of max pooling operation and average pooling with a 2 × 2 filter from 4 × 4 array input. An alternative technique to perform downsampling is setting a stride of convolution operation larger than 1. It is noteworthy that there is no learnable parameter in any of the pooling layers, whereas filter size, stride, and padding are hyperparameters.

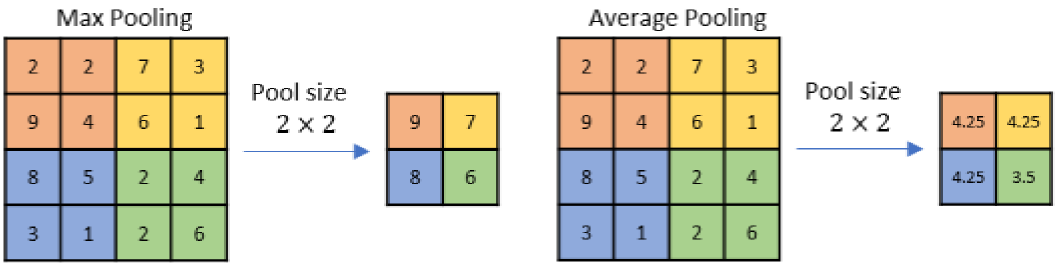


Figure 2-11 Illustration of Max Pooling and Average Pooling

Fully connected layers, also known as dense layers, can flatten the input feature maps of convolution or pooling layer into a one-dimensional (1D) array of numbers (or vector) through a linear transformation by a set of learnable weights (Yamashita *et al.*, 2018). In image classification, the final fully connected layer typically has the number of output nodes the same as the number of classes, and each node value representing the predicted probability (score) of each class. The final predicted class is that with the highest score value.

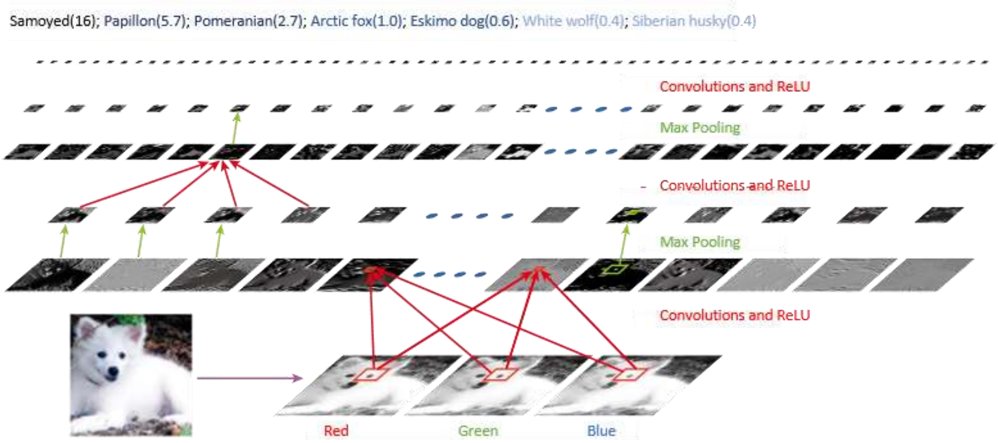

Samoyed(16); Papillon(5.7); Pomeranian(2.7); Arctic fox(1.0); Eskimo dog(0.6); White wolf(0.4); Siberian husky(0.4)
Convolutions and ReLU
Max Pooling
Convolutions and ReLU
Max Pooling
Convolutions and ReLU
Red
Green
Blue

Figure 2-12 A typical CNN structure applied to a classification task. The input image is split into three arrays, representing RGB pixel intensities. Each rectangular image is a feature map corresponding to a specific convolution filter. Input image is processed bottom up, where feature information from the lower layers is the input of the upper layers. Finally, a score is computed for each image class (LeCun *et al.*, 2015)

In CNNs, the output of the lower layer becomes the input for the higher layer, and thus higher layers capture more semantic information. Because of this hierarchical structure of CNN, a trained CNN can be viewed as a hierarchical feature extractor (Long *et al.*, 2014; Yang *et al.*, 2015). CNN is used to extract the deep features of the images for repeated pattern detection in the development of R-system (this will be given in CHAPTER 4).

2.4 Generative Tasks vs Design Generation

Similar to typical art and design work like graphic design, digital pattern design have the same main components, namely visual elements, layout, color palette, and text (Wilson, 2001; Zhao, 2020). The key difference between digital pattern design and graphic design is that digital pattern design must take into consideration different rules and constraints in the design process, including material properties, production equipment and the cost budget. For instance, screen printing has a limit on the number of colors, and the cost of printing rises as the number of colors increases.

With the popularity of deep learning, there are increasing interests in generating artworks or design images (Liu *et al.*, 2016; Zhao *et al.*, 2018), most notably using generative adversarial network (GAN). GANs, firstly introduced by Goodfellow (2020), are deep learning-based methods designed to perform generative tasks. GANs are characterized by two networks -- Generator network generates pixels images and Discriminator network determine whether the image comes from the generator or real data distribution. GANs have produced some astonishing results in generative tasks like style transfer, makeup transfer or face synthesis or aging, yet there are also known limitations of existing GANs for design generation. First, most GAN work are devoted to exploring graphic layouts (Guo *et al.*, 2021; Li *et al.*, 2019). Second, the results from these GANs are neither stable or predictable. Third, the outputs from such systems are *raster* images, which are difficult to further edited and cannot meet the production requirements (some examples are shown in Figure 2-13).

Figure 2-13 Examples of artwork generated by GAN (Cetinic *et al.,* 2022)

When comparing with generative tasks, the existing GAN research may not align well with the philosophy or be in compliance with the requirements for design. Design creation is the process of looking for specific characteristics in a design proposal to deliver the intended service based on prior information and certain broad presuppositions (Lu *et al.*, 2014). Design process is typically iterative. In this study, other than GANs, a new approach of design generation is proposed that aims to produce new editable designs from existing design images, aligning with the human aesthetics and follow standard design rules.

2.5 Chapter Summary

This chapter reviews the related research work of this study, including computer-aided fashion design, how computers understand an image, CNN's formulation and architectures, and deep learning-based design generation techniques.

Digital pattern generation receives very little attention in the current computer-aided design systems. Existing technologies are not able to generate new designs simultaneously meeting two requirements of aesthetically pleasing to humans on one hand and in an editable format on the other hand. As reviewed, pattern designs mainly are composed of design elements, layout, text, and color. It is possible to extract design elements and color composition from existing design images, reuse them according to the standard design rules to create new, aesthetically pleasing digital patterns that are in vector formats. This is not only the research direction of this study but also in line with the expectations and needs of the fashion and textile industry.

Deep learning is very effective in solving problems with large number of training data sets. Due to its success in identifying complex structures in high-dimensional data, it can be used in various fields of science, business, and engineering. However, features learnt from a deep neural network are specific to the training dataset. For example, images/data from the same training data set tend to have the same feature space and the same data distribution. When the test data has a different data distribution than that of the training data, the prediction model will have a degraded performance. It is believed that deep learning models are very much data driven, will not have a comparable performance for cross domain applications. Moreover, in addition to the mandatory requirements for high-performance equipment, deep learning methods have other limitations, e.g. low interpretability of results. In contrast, traditional computer vision methods offer advantages of full transparency and universality.

It is obvious that digital pattern images and real-world images have very different characteristics. Directly applying deep learning methods trained in real-world images will not achieve the expected results. In this study, new methods will be proposed integrating traditional computer vision methods and deep learning methods, taking advantage of both approaches.

CHAPTER 3. Framework Overview

As discussed in both Chapters 1 and 2, analyzing existing designs and extracting the design information builds a foundation for computers to generate new designs that align with basic human aesthetics and meet with production requirements. In this study, a new framework, denoted as REG framework, is proposed, on which a total of three intelligent systems are developed to *analyze* existing digital patterns images and to *support* new design generation.

As shown in Figure 3-1, on the REG framework, both R-system and E-system are used for design images analysis, aiming to mining useful information from existing design images, namely repeated patterns detection and design elements extraction. The G-system is used to support the design generation, in which methods are also developed to extract color information and layout rules from reference images, applying them flexibly to generate editable new designs. The three intelligent systems are related yet can operate independently This chapter first outlines this entire framework and each system, as well as the relationship between these three intelligent systems. The detailed method and the rationale of each system, the experimental verification as well as how it compares with other existing methods in computer vision are given in later Chapters 4-6.

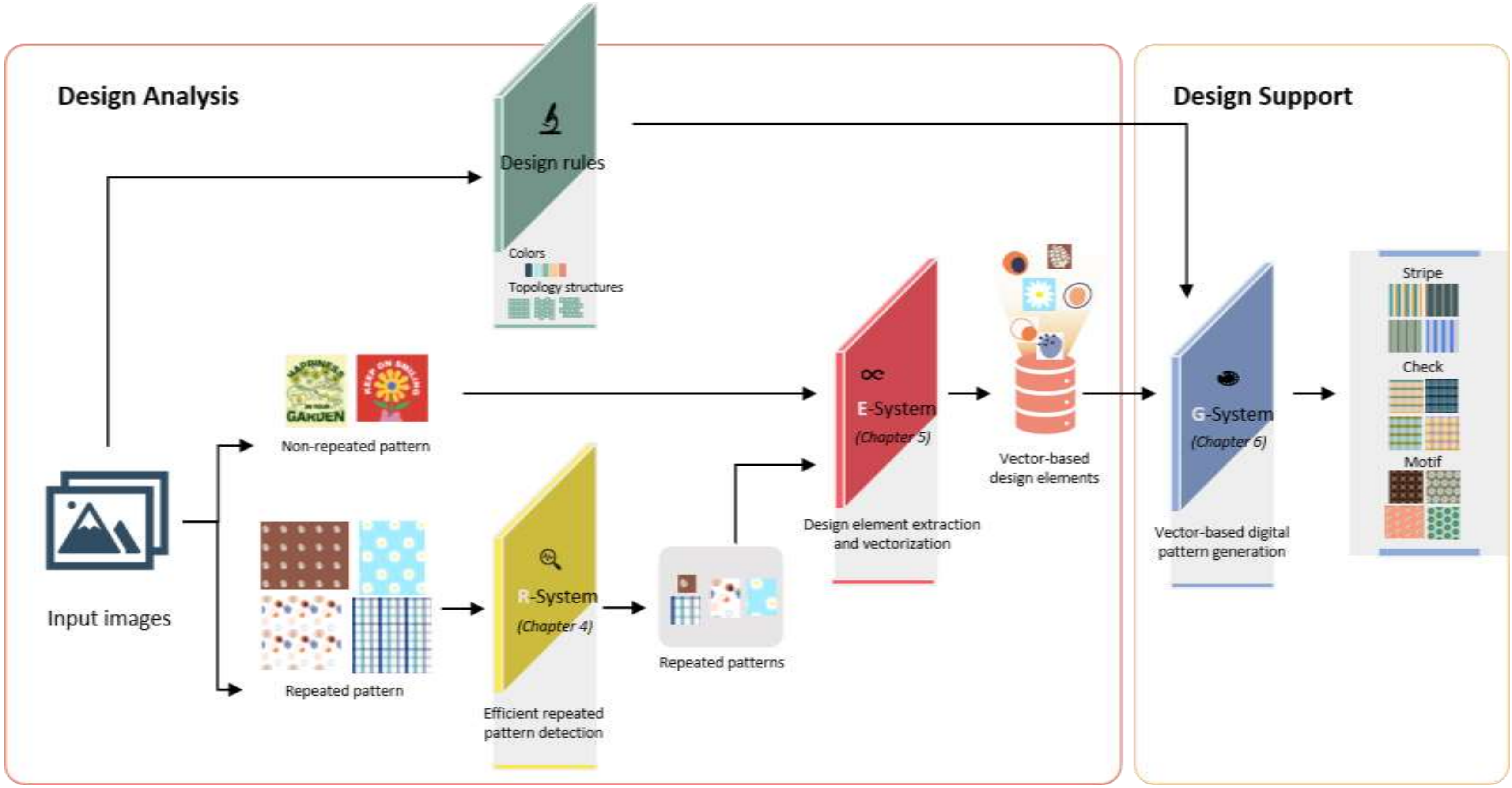


Figure 3-1 Framework overview of this study

3.1 R-System for Repeated Pattern Detection

Based on the discussion in CHAPTER 1 and CHAPTER 2, designers create digital patterns by designing elements in vector formats which can be easily edited into different shapes and colors. By applying these small design elements using layout rules, more complex and interesting pattern designs are generated. This procedure aligns with human aesthetic pursuit, considering color harmony, symmetric shape, layout regularity, etc. By mimicking the creative operation of human designers, the proposed system should be able to assist the analysis of existing design, e.g. detecting repeats and layout rules.

A repeated pattern is the smallest unit that can be used to form a large digital pattern. It contains the core information of the complete pattern. Therefore, when exiting digital pattern images are input to the framework, such repeated patterns, if present, must be retrieved for design understanding and information extraction. Other benefits of detecting repeated patterns include to use representative content of the design images for further process, such as image retrieval and classification.

Repeated patterns are common found in design images and natural scenery. The detection of repeated pattern is yet a challenging topic in computer vision, as there are many different types of images and their repeated patterns vary in terms of content diversity and topology regularity. By reviewing the literature on repeated pattern detection, a new system called R-system is proposed to balance robustness and efficiency of the system in detecting repeated patterns focusing on digital pattern images.

Repeated patterns on digital pattern images can be abstracted as a set of repetitive image features that are consistent in location. Repeated pattern detection is equivalent to finding the repetition rules through modeling the respective image features. CNN-based methods are proposed to ensure the R-system is robust and manage diversified content taking advantage of the strength of CNN of being able to learn from a large volume of examples. As reviewed in Section 2.3, a CNN's main components are convolutional layers, activation functions, and pooling layers. Particularly, at various levels of scale, the filters of various convolutional layers acquire various semantic properties. Higher layer learning contains more semantic information than lower layer learning. On the basis of that, a pre-trained CNN can be viewed as a multi-layer image feature extractor that takes into account shallow image features and high-semantic image features. The extracted CNN features will be used to detect repeated patterns. Since large number of features are extracted and training of prediction model based on the extracted features is thus time-consuming and computationally expensive.

Comparatively, traditional autocorrelation-based method is efficient to train given a set of training data. The autocorrelation-based methods use a given *template* patch and autocorrelation among repeated patterns to construct a model to predict the size of the repeated pattern. The drawback is they are sensitive to image noise and the initially defined *template* patch, resulting in unstable results.

Based on the above discussion, the proposed R-system consists of two main modules that use CNN features and template matching to better balance detection accuracy and efficiency in repeated pattern detection from unknown images. Specifically, the first module uses CNN features to predict a coarse repeated pattern size. When an image is fed into a CNN, the filters inside each convolutional layer will produce feature maps that react to different semantic and scale levels. The *activation peaks* on the selected feature maps reflect the location information of the repeated patterns. Therefore, the size and locations of the repeated patterns can be then predicted by fusing and examining the distribution of the collected activation peaks from different feature maps. The running speed of a CNN feature-based method is therefore proportional to the number of selected filters. To speed up the processing, this study selects filters from a pre-trained CNN network by leveraging the boundary detection results of the input images. To efficiently optimize the predicted repeated pattern size, an optimization module based on template matching with a self-adaption similarity threshold is proposed in the second module. The output varies for different inputs. If the input images are with repeated patterns, the R-system will output single unit of repeat, otherwise, it will output the entire image of unrepeating image for next phase of image analysis. The output from R-system can be further processed in next phase for design analysis or supporting design.

3.2 E-System for Design Element Extraction and Vectorization

Raster image and vector image are two common file formats for any visual design, especially in digital patterns, which serve different purposes of design (Lu *et al.*, 2014). Raster image, made of pixels, is a format mainly used for design presentation in digital means, for instance, on websites. It is difficult to edit and the quality of the image is determined by the amount of pixels used to represent the content within a standard size/area of image (resolution). Vector images are often known as scalable vector graphics (SVG), which are resolution independent, are easy to edit or scale to any size without loss in quality or details. Design is not a one-off service, designs must be iteratively fine-tuned and adjusted, so most design systems will output vector formats of design to fit for production needs (Dominici *et al.*, 2020). Vectorization is a process of converting a raster image to a vector one, which is useful for design information extraction from existing designs and making sure they are editable for further use. Moreover, since digital patterns consist of multiple distinctly colored regions, they are inherently suitable for vectorization. Therefore, the vectorized output is a must for the proposed system, even for the overall framework for design analysis and support.

Common digital patterns used in the fashion industry, also the target subject of this study, are often composed of multiple repetitive or non-repetitive design-related objects, also known as design elements. According to the analysis in the Introduction, reorganizing design resources by combining design rules allows computers to mimic human designers to create digital patterns. Therefore, the proposed system should also analyze and deconstruct design images into reusable design elements, and record which as design resources for future use.

In order to achieve fully automatic processing from extraction to vectorization of the core design elements from an unknown image, a novel E-system is proposed, consisting of two main modules for the extraction and vectorization of design elements, respectively. It’s worth noting that the E-system could work independently to process any digital pattern images, while if the input image contains repeated patterns, the R-system can be a preprocessing step to trim down the input to a patch of only 2×2 repeated pattern in order to reduce computation cost.

There are great differences between design images and real-world images in terms of content and style, leading to some popular image processing techniques cannot be directly applied, such as supervised object detection and image segmentation. These techniques or systems are data-driven and even data-hungry, need to be trained on a large dataset of labeled images. Nevertheless, there is no publicly accessible dataset with images of labeled design elements. Different from the other domains, digital pattern images in the fashion domain could be more diverse and difficult to label. Considering the uniqueness of this task and the cost of labeling a large dataset, this study proposes the use of unsupervised segmentation for design element extraction, solving the problem of without a labeled dataset.

Upon the foreground and background content of an image are isolated by unsupervised segmentation, the resulting background mask is utilized to help locate its foreground design elements. In the first module, an unsupervised segmentation method is used to separate the foreground and background contents. Particularly, a Fully Convolutional Network (FCN) with three convolutional components is proposed. Over-segmentation algorithm (Felzenszwalb *et al.*, 2004) (discussed in Section 2.2.3) is used to produce initial labels and initialize the network parameters by Xavier initialization (Glorot *et al.*, 2010). The network is trained by iteratively comparing predicted labels of over-segmentation algorithm and that from the network. After comparing the area of the minimum bounding rectangle of each segmentation mask, the background mask is obtained. Next, the selected background mask is used to extract all the design elements. Since there may be duplicate elements among these extracted design elements, the duplicate design elements are removed by comparing their pHash values (Zauner, 2010). Moreover, the proposed method for repeated pattern detection trims images without losing their main information, simplifying the extracted design elements.

Image vectorization requires deconstructing the image into a number of intermediate representations, then converting each intermediate representation into a corresponding vector path with an enclosed region/area. Due to the lack of research on image deconstruction, image segmentation is the go-to solution for image vectorization. Each vector path represents a specific color region and is composed of geometry and color parameters. In this study, a novel image deconstruction method is proposed leveraging color information of the input image. In the second module, a design element is deconstructed into several intermediate representations and the corresponding color information is obtained simultaneously. In particular, in the design element deconstruction method, firstly color quantization is leveraged to get the number of colors and then the k-means clustering algorithm is used for color reduction and path simplification. Next, the Potrace (Selinger, 2003) algorithm is applied to generate the vector paths for the intermediate representations. Furthermore, if too many tiny vector paths are used to represent the final vectorization results, making the result hard to further edit or reuse, a simplification mechanism is proposed that the vector paths with very small enclosed area are eliminated or regrouped before vectorization process. By doing this, we can ensure that the vectorization results are compact and easier to use.

3.3 G-System for Vector-based Digital Pattern Generation

Computer aided design systems were developed to generate specific type/style of patterns, such as fractal art and marbling patterns (introduced in Section 2.1), but they are not for typical pattern designs. In general, digital patterns can be roughly categorized into stripe, check, and motif patterns in the industry. For better design support and to fulfill industry requirements, the proposed G-system aims to generate vector-based digital patterns covering the above types of patterns. To generate digital patterns satisfy basic human aesthetics, the G-system mimics human creative process and utilizes the design resources extracted from the R and E systems. Specifically, the inputs to the G-system are reference images for color and topology layout reference as well as vector-based design elements. The system utilizes industrial design principles and flexibly applies design elements to generate new patterns by formulating parametric models of different digital patterns.

The proposed G-system consists of three parametric models to generate particular types of patterns. As vector images contain geometric and color parameters, the proposed method accomplishes design generation from these two aspects. The first parametric model is used to generate stripe patterns. A stripe pattern can be considered as a set of parallel bars with varied width, and each bar is filled with a different color. A parametric model for stripe pattern of known square area will define both geometric and color parameters, namely the width of the parallel bars and the color of each bar. The color parameter of a stripe pattern is extracted from a given reference image in the proposed G-system.

The second parametric model is developed for check pattern generation. A check pattern is defined as a combination of two orthogonal stripe patterns; therefore, the system generates check patterns by combining two stripe patterns whose color combinations follow the same reference image. Relationship between the number of colors and the cost in textile production are considered in this operation.

The third parametric model aims for motif pattern generation. Motif patterns for textile use usually consist of several repetitive design elements that are arranged in a unit area and then compose for more complex design by applying different design layout rules. In order to generate motif patterns that satisfy basic human aesthetics, parametric model incorporating golden ratio is developed to automatically layout design elements in various scale, orientation and location to form representative design unit -- motif pattern, which are further tiled to form various repeats based on industrial design rules, generating vector-based digital motif pattern designs ready for production.

3.4 Chapter Summary

This chapter presents digital pattern analysis and design support framework composed of three independent and interrelated intelligent systems. The methods for the three systems apply both deep learning and traditional computer vision techniques to address the specific research task. Specifically, the R-system and E-system are proposed for analyzing digital pattern images at the repeated pattern level and design element level. The G-system is proposed to generate vector-based digital patterns that meet basic human aesthetics for design support.

CHAPTER 4. Efficient Repeated Pattern Detection

4.1 Background Introduction

As discussed in the Introduction, fashion products are important to help wearers express their personalities and attitudes towards life. Incorporating appropriate digital patterns in fashion products can add values to the products and thus increase sales. Digital patterns can be divided into repeated and non-repeated patterns according to their contents (as shown in Figure 4-1). In terms of fashion product development, the *repeated patterns* are often applied in most fabric designs while *non-repeated patterns* are placed and applied to certain areas of the clothing, e.g. in center front of the clothing or logo designs. Specifically, repeated patterns can be defined as an image's minimum representative unit (or tile) and can replicate its original appearance when tiled (Lin *et al.*, 2006; Rodriguez-Pardo *et al.*, 2019). They represent the primary information of the image in a compact form and thus can provide important geometric or semantic cues for many algorithms (Lettry *et al.*, 2017; Li *et al.*, 2020; Park *et al.*, 2009).

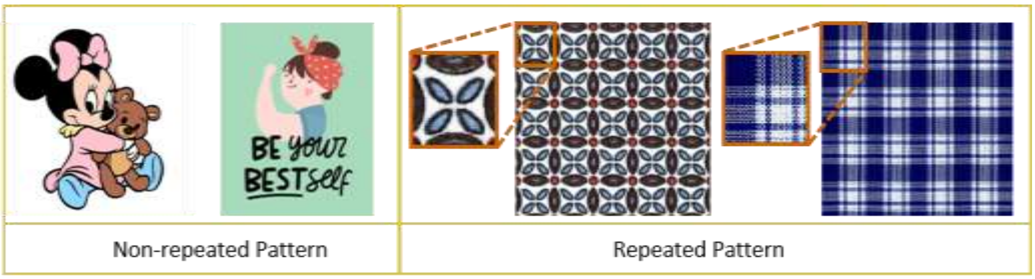


Figure 4-1 Classification of digital patterns. Digital patterns can be classified into repeated patterns and non-repeated patterns; examples are shown above, and the repeated patterns are zoomed in orange boxes

Detecting repeated patterns in a single image contributes to analyzing digital patterns by distilling their primary information. Moreover, it benefits other fashion industry tasks, such as textile image classification and retrieval (Lettry *et al.*, 2017; Li *et al.*, 2020; Park *et al.*, 2009), low-cost textile information storage (Kuo *et al.*, 2017), and textile image synthesis (Rodriguez-Pardo *et al.*, 2019). Other than fashion applications, detection of repeated pattern is very important in computer vision and they can support various applications in other domains, such as place recognition (Torii *et al.*, 2013), city image retrieval (Doubek *et al.*, 2010), and photo-realistic 3D reconstruction (Liu *et al.*, 2019) See Figure 4-2 for various repeated patterns in various domains of image.

Although it is almost human instinct to detect repeated patterns, it is challenging for computers because of the diversity of image types and complex variations caused by perspective, lighting, occlusion, projection, deformation, and other factors. Different repeated patterns can be characterized in terms of *content diversity* and *topology regularity*, to which new algorithms or methods must address for the effective detections. As shown in Figure 4-2, the horizontal and vertical axes represent topology regularity and content diversity, and these two-dimensional varieties form a matrix or coordination system for different types of repeated patterns. Images located at different positions within the coordinate system represent repeated patterns with different levels of variety in terms of content and topology structure.

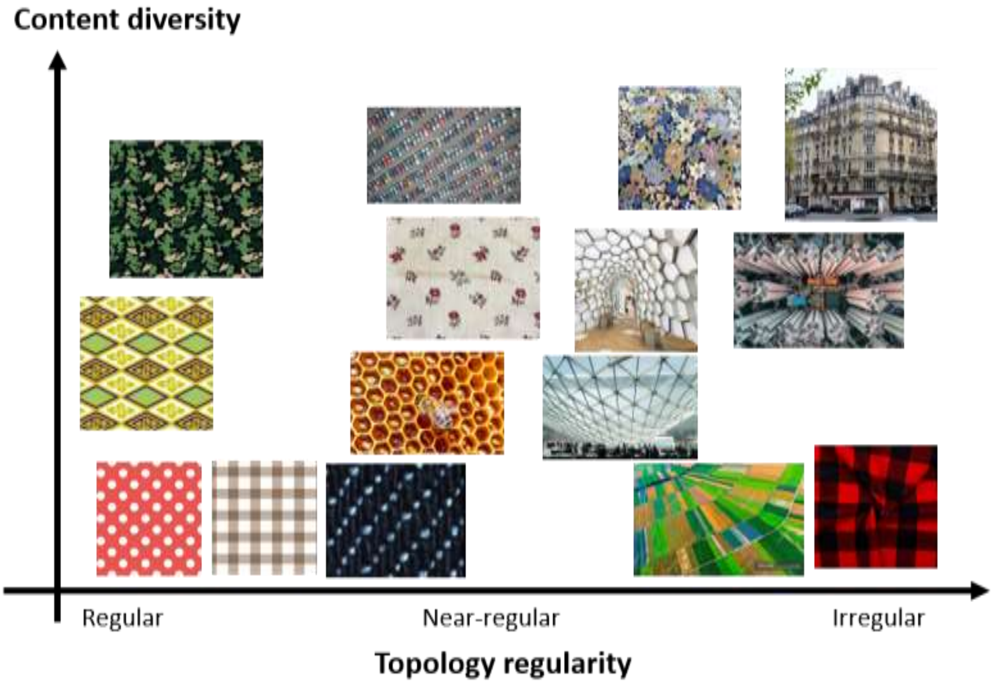


Figure 4-2 Variation of repeated patterns in terms of content diversity and topology regularity

Many methods have been proposed to handle different types of repeated patterns. Certain methods may output the size of repeated pattern on the image, while other methods find the key points of the repeated pattern as output, which will be further used in image retrieval, identification and classification. Any method for repeated patterns detection should include (1) feature extraction and processing and (2) model development and optimization. In particular, to handle the detection of repeated patterns that are exactly the same in content, some methods use the autocorrelation functions to construct prediction models to accurately estimate/optimize the repeated pattern size (Tao *et al.*, 2017; Zhang *et al.*, 2015). Although these methods are efficient, they are only applicable to regular and noise-free repeated patterns. On the other hand, other studies have been devoted to detect repeated patterns with significant variations caused by occlusion, projection, and distortion. These methods have demonstrated that after abstracting an image into a set of local features, the repeated patterns in the image can be detected by recognising the repetition rule of features (Liu *et al.*, 2004; Schindler *et al.*, 2008). Nevertheless, these methods may suffer the common drawbacks of traditional features, which are ‘handcrafted’ with low content adaptability. Recently, Convolutional Neural Networks (CNNs) have shown great performance in many computer vision tasks, such as image classification (Simonyan *et al.*, 2014), object detection (Zhang *et al.*, 2021), and semantic segmentation (Vitale *et al.*, 2016). As discussed in CHAPTER 2, CNNs are good at obtain hierarchical feature information of images, which means not only the low-level information like corners and lines, but also high-level information with semantic meaning can be extracted (Long *et al.*, 2014; Malik *et al.*, 2021). Thus, many research works view trained CNNs as feature extractors to substitute local feature descriptors, including repeated pattern detection (Lettry *et al.*, 2017; Rodriguez-Pardo *et al.*, 2019). These methods predict the repeated patterns' position and/or size by processing deep image features. Although these methods improve the robustness, being able to handle more diverse content, they are computational expensive, with low efficiency, especially for large image size and high-resolution images.

As discussed, repeated patterns are present in many domains of image, including architecture, natural scenery, and other real-world images. Design images like digital pattern designs obviously have very different characteristics than real-world images. The types of repeated patterns in digital pattern design images are also different, although digital patterns have very diverse and complex content, but they tend to have relatively regular topology. According to the special characteristics of pattern design images, this study proposes a novel R-system that combines the use of CNN features and an autocorrelation function named template matching (Brunelli, 2009) for repeated pattern detection from digital pattern sample images. The R-system firstly leverages CNN features to predict a coarse repeated pattern size option of the input image. Specifically, a CNN filter selection method combined with boundary prediction is proposed to select the filters with meaningful information, which is beneficial in reducing computational time. Then, a specially designed optimization method based on template matching is proposed for efficient identification of repeat pattern sizes, considering the input image characteristics. To verify the effectiveness of the proposed R-system, a dataset of digital patterns design images with manually defined repeated pattern ground truth is developed. Extensive experiments have been conducted on this proposed dataset to show the effectiveness of each component of our proposed system and our proposed system has also shown competitive advantages on both accuracy and efficiency, in comparison to other state-of-the-art methods

The rest of the chapter is organized as follows. Section 4.2 introduces related works on detecting repeated patterns and explains the inspiration for our system. Section 4.3 describes our system in detail. Section 4.4 presents the evaluation results of our system, showing better performance on both accuracy and efficiency over related works. Section 4.5 discusses the results on manufactured environmental images and the impact of various CNN models and boundary detection methods. And in section 4.6, we conclude.

4.2 Related Work

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

4.2.1 Model development and optimization

When an image consists of repeated patterns with a regular layout and very similar content, the relationship between repeated patterns (namely the regularity of the layout) can be analysed and predicted by *autocorrelation*. Since the representations from autocorrelation analysis show the topology regularity/pattern of the input image, by matching the analysis results of a target images and a template patch, the size of the repeated pattern can be predicted. Many works use autocorrelation functions in regular repeated pattern detection (Doubek *et al.*, 2010; Lin *et al.*, 1997; Neupane *et al.*, 2019; Tao *et al.*, 2017). Tao *et al.* (2017) used Fourier transform to build a similarity space to identify repeated patterns in textile images. Some researchers leveraged template matching (Brunelli, 2009) for repeated pattern detection. For example, Qayum *et al.* (2017) 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 were also presented in the work of Neupane *et al.* (2019) and Lin *et al.* (1997). Although these methods work very efficiently for images of any intensity with strong or weak features, they need a given template patch in advance (Cai *et al.*, 2011; Kuo *et al.*, 2008). The size deviation between the given template patch and the repeated pattern will affect the accuracy of the final results. In addition, their methods are poor in robustness which means they have a low tolerance for noise and thus cannot achieve good detection results on repeated patterns with variations, e.g. by occlusion or other forms of deformation.

4.2.2 Feature extraction and processing

4.2.2.1 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 (Cai *et al.*, 2011; Conners *et al.*, 1980; Pinho *et al.*, 2011; Tuytelaars *et al.*, 2003). Some algorithms use SIFT (Lindeberg, 2012) as feature descriptors (Liu *et al.*, 2013; Liu *et al.*, 2004; Pritts *et al.*, 2014). However, these methods are only applicable for images with distinct corner/edge features and similar degrees of variation. Furthermore, some feature descriptors are hand-crafted and designed based on professional knowledge of images in specific fields (Guo *et al.*, 2019). Therefore, despite the fact that local feature-based approaches are widely used in 2D 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 (Liu *et al.*, 2018).

4.2.2.2 CNN feature-based

As discussed in Section 2.3, a trained CNN contains a set of convolutional layers with fixed weights that can capture hierarchical information of the image. A convolutional *filter* in the convolutional layer generates different activations of the input image/pre-layer output and encodes the spatial information of discriminative regions into a *feature map* (Ahmad *et al.*, 2018; Zhou *et al.*, 2016). 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 (Malik *et al.*, 2021). An example was given in Figure 2-10 on page

46 that feature values after convolution operations can be viewed as activation scores. 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 (Long *et al.*, 2014; Yang *et al.*, 2015). Therefore, in many tasks, a trained CNN can be treated as a more robust feature descriptor and replace local image feature descriptors, such as SIFT or HoG. For example, El Amin *et al.* (2016) used CNN features for change detection in satellite images; Xiang *et al.* (2019) leveraged CNN features to help retrieve similar fabric image.

The potential of a CNN-feature-based approach in repeated pattern detection was presented by Lettry *et al.* (2017). The authors demonstrated that repeated patterns' key points in 2d images have spatial consistency with certain activations within feature maps; hence, extracting activations can assist in repeated pattern detection. Moreover, the authors proved that CNN features can perform better than local feature-based techniques. Nevertheless, their method took most of the activations in each feature map into consideration for the final model optimization, which is computational expensive and result in poor efficiency. In a related study, Rodriguez-Pardo *et al.* (2019) reduced the amount of computation by selecting filters that are active more than others and thus reducing the number of feature maps processed as well as activations. A similar idea of data reduction was used in landscape image analysis: Malik *et al.* (2021) used principal component analysis (PCA) to reduce the CNN feature maps to a compact representation that best encodes patterns in a given landscape for landscape similarity analysis.

To achieve a good trade-off between accuracy and efficiency for repeated pattern detection in an unknown digital pattern image, the R-system is established by combining CNN features processing and autocorrelation-based optimization.

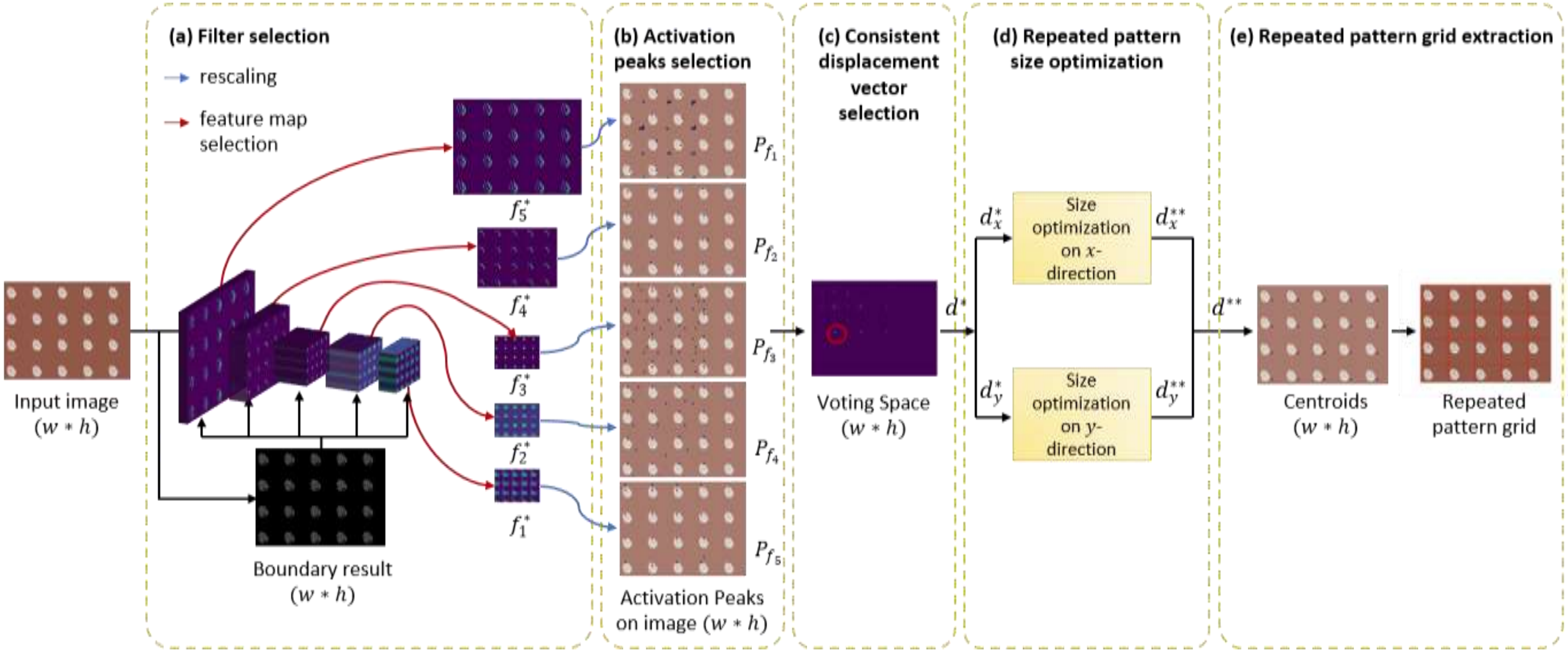

(a) Filter selection
rescaling
feature map selection
f_5^*
f_4^*
f_3^*
f_2^*
f_1^*
Input image $(w * h)$
Boundary result $(w * h)$
(b) Activation peaks selection
P_{f_1}
P_{f_2}
P_{f_3}
P_{f_4}
P_{f_5}
Activation Peaks on image $(w * h)$
(c) Consistent displacement vector selection
Voting Space $(w * h)$
d^*
(d) Repeated pattern size optimization
d_x^*
Size optimization on x-direction
d_x^{**}
d_y^*
Size optimization on y-direction
d_y^{**}
d^{**}
(e) Repeated pattern grid extraction
Centroids $(w * h)$
Repeated pattern grid

Figure 4-3 Illustration of the R-system for efficient repeated pattern detection

4.3 Method

The overall pipeline of the R-system consists of five parts and is illustrated in Figure 4-3. CNN has shown a great capability for representation learning and feature extraction, and the kernels (i.e. filters) of different convolutional layers capture spatial information of discriminative regions at varying scales and encode them in feature maps. When the image contains repeated patterns passed over a pre-trained CNN, activations of the feature maps are in line with the repetition structure. In order to efficiently use CNN features to predict the coarse repeated pattern size, a filter selection method leveraging boundary detection results is proposed in section 4.3.1, as first step/part of the R-system. Then, an activation peaks extraction method is described in section 4.3.2 as the second step. Following which, a consistent displacement vector selection method is used to fuse the extracted activation peaks for repeated pattern size prediction, which is described in section 4.3.3. After that, an optimization algorithm is presented in section 4.3.4, which optimizes the repeated pattern size in the x and y directions, respectively. Lastly, in section 4.3.5, a repeated pattern grid extraction method using the selected filters is explained. As shown, the inputs to the R-system are any unknown digital pattern design images, and the outputs from the system are the corresponding pattern grid of the repeat, if present. The detail steps of the proposed R-system are described in the following subsections.

4.3.1 Filter selection

In every image representation scheme, selecting the appropriate features is essential. By removing irrelevant features, it enables feature dimension reduction and increases the effectiveness of feature extraction (Li *et al.*, 2019; Xie *et al.*, 2018). The feature maps generated by CNN filters reflect repeated patterns' location information in CNN feature space. In a pre-trained CNN, feature maps from a deep convolutional layer are capable of representing a huge variety of information, based on the type of data it is trained on (e.g. ImageNet) (Ahmad *et al.*, 2018). For example, the classic AlexNet has five convolutional layers, which correspondingly have 96, 256, 384, 384, and 256 number of filters within the layers (Krizhevsky *et al.*, 2012); when an image passes, it will generate 1376 (=96+256+384+384+256) feature maps. When all these 1376 feature maps are analyzed in the model optimization for detection of repeated patterns, the analysis is computationally expensive, but yet may not be necessary because not all features maps carry equally important information for the said task of repeated pattern detection. Feature selection techniques are therefore used to determine an appropriate subset of feature maps to adequately represent digital pattern images. This appropriate subset of feature maps needs to be both concise and take into account hierarchical information.

In this study, boundary information is proposed to use for filter selection. This is inspired by the fact that boundary detection extracts object boundaries and visually salient edges that are persevering the gist of the image and ignore weakly related information (He *et al.*, 2019). Moreover, the output from a boundary detection result is binary, in form of a two-dimensional $[0,1]$ matrix, which is easy for feature computation. In the first step of filter selection of the R-system, image boundary detection results are leveraged to select feature maps and their corresponding filters in this study.

Given a pre-trained classification CNN, the convolutional filters in the CNN is denoted as $F = \{F_n | n \in N\}$, $f_n \in F_n$, where N is a set of convolutional layers, F_n is the set of convolutional filters in the n-th layer, f_n is the resulting feature map of the n-th layer. Taking AlexNet as an example, which has 5 convolutional layers, thus N=5 and F_1 is the set 96 kernels (filters) in the first layer, F_2 is the set of 256 filters in the second layer, and so forth.

When an input image I with size $w \times h$ is passed through the CNN, the convolutional filters would generate activated feature maps. If the input image I has repeated patterns, the activations should separate at regular distances showing the topology layout of repeated patterns. Each convolutional layer of a deep CNN has many filters, and there is a filter response the highest among all of them since it captures the most significant features following the repetition structure. The detected boundary result of the input image is denoted as B and is resized to be the same size as the feature maps f_n of the corresponding convolutional layer as B_n. The resized B_n is then multiplied with the feature map $f_n(I)$ from the n-th layer and sum over the activated values of all pixels as follows:

$$f_n^B = \sum_{i,j} B_n \cdot f_n\,(I) \tag{4-1}$$

where (i, j) represent the pixel. The highest response convolutional filter with regular activated feature map f_n^* of the n-th layer can be obtained by:

$$f_n^* = \underset{n \in N}{\operatorname{argmax}}(f_n^B) \tag{4-2}$$

The classic Canny edge detector is used in the proposed pipeline (Canny, 1986) for its efficient speed. An example of feature maps of the selected filters is shown in (a) of Figure 4-3. A comparative study of different boundary/edge detectors for generating the filter B will be given later experimental Section 4.5.3.

4.3.2 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 $\{f_n^* | n = 1,2, \dots, N\}$, the set of activation peaks P_{f_n} is found by extracting the local maxima from each feature map $f_n^*(I)$. An algorithm is proposed for the extraction of activation peaks, which is outlined in Algorithm 4-1. Assume a given distance threshold is $dist$ pixels, P_{f_n} can be computed within the region of $2 \times dist + 1$. Considering the different resolutions of the selected feature maps, $dist$ is determined based on the required number of activation peaks so as to guarantee the extraction of sufficient peaks from each feature map. Let k be the required number of activation peaks, one of the inputs of the algorithm. If the number of generated peaks is less than k, then $dist = dist - 1$, and a loop is recursively executed until the number of generated peaks is more than k. Moreover, the coordinates of the extracted activation peaks would be mapped from feature space to the input image by proportion. An example is shown in Figure 4-3(b).

Algorithm 4-1 Activation peaks selection

```
Input: the filter $f_n^*$
        the feature map $f_n^*(I)$
        the required number of activation peaks $k$
        the distance threshold is $dist$
Output: extracted activation peaks $P_{f_n}$
1.  $P_{f_n} = peak\ local\ max(f_n^*(I), k)$
2.  if length$(P_{f_n}) < k$ then
3.      $k = k - 1$
4.      if $k > 0$ then
5.          $P_{f_n} = peak\ local\ max(f_n^*(I), dist)$
6.      else
7.          $P_{f_n} = [\,]$
8.          return $P_{f_n}$
9.  else
10.     return $P_{f_n}$
```

4.3.3 Consistent displacement vector selection

The most frequent distance between activation peaks in both image and feature space is the most probable size of the repeated pattern. For each activated feature map f_n^*, a set of *displacement vectors* D_{f_n} is obtained by computing the distance between each pair of peaks

$$D_{f_n} = \left\{ d_{f_n}^{i,j} = \left| p^i - p^j \right|, \forall p^i, p^j \in P_{f_n}, i \neq j \right\}, \tag{4-3}$$

where |.| denotes the element-wise absolute difference, displacement of the two element. Since each displacement vector $d_{f_n}^{i,j} \in D_{f_n}$ represents a candidate size of the repeated pattern, all the displacement vectors are used to estimate the initial repeated pattern size by voting in a Hough-like voting space $V: \mathbb{R}^n \to \mathbb{R}^2$.

To obtain the continuous distribution V of displacement, each sub-distribution $V_{n,i,j}$ of displacement vector is modelled using a 2-dimensional Gaussian distribution with mean $d_{f_n}^{i,j}$ and standard deviation δ_n and fuse the activation peaks across multiple layers. Particularly, δ_n is related to the width of each feature map $w_{f_n^*(I)}$. Moreover, to normalize the energy of each vector in each convolutional filter f_n^*, the vote is weighted by the number of displacement vectors $\left| D_{f_n} \right|$ across all layers and filters as follows:

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

where

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

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

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

When the input images are digital pattern design images, these images tend to have high topology regularity, the resulting *size* of the repeated patterns can be represented as the most consistent displacement vector d^*, which is defined as the maxima on the voting space along both the *x*- and *y*-axes:

$$d^* = \left(\underset{x}{argmax}\, V_{x,0}\, , \underset{y}{argmax}\, V_{0,y}\right) \tag{4-5}$$

Figure 4-4 shows some examples of the voting space V, where the highlighted spots show the most consistent displacement vectors.

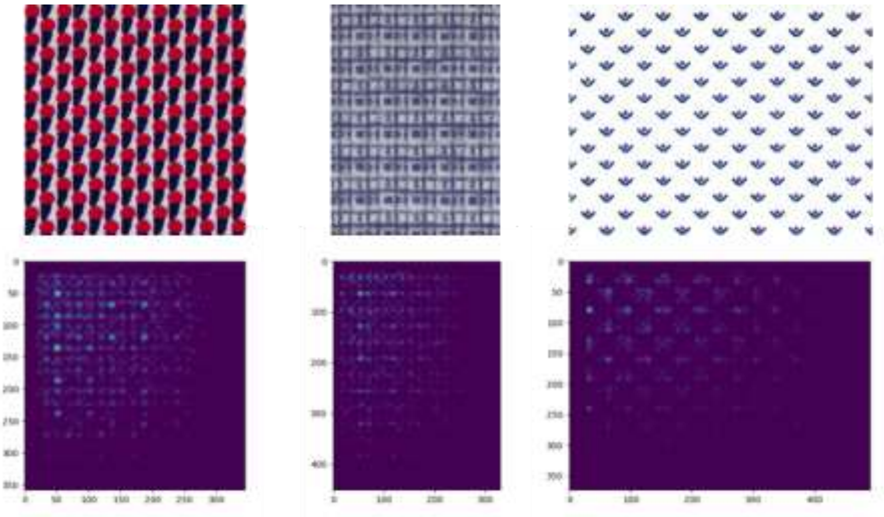


Figure 4-4 Illustration of voting for consistent displacement vector. Top: Images with repeated patterns. Bottom: Corresponding displacement vector maps. The light spots correspond to a higher frequency of displacement vectors

4.3.4 Repeated pattern size optimization

Template matching is a method for searching and finding the location of a given template patch in a larger image (Brunelli, 2009). As reviewed in Section 4.2.1, there are studies using autocorrelation function for repeated pattern detection in textile images. In view of the digital pattern design images characteristic, which have relative regular or near-regular topology but very diverse content, an algorithm integrating autocorrection together with template matching is also developed for optimizing repeated pattern size in this study.

After obtaining the initial repeated pattern size d^*, the region from the left-upper point $o = (0,0)$ is cropped with size d^* as the template patch T. The matching process overlays T onto every possible area within the image of size $(w * h)$, and computes a pairwise correlation $cor_{(x,y)}$ that indicates the similarity degree between template T and that particular area, denoted as $S_{(x,y)}$. The correlation is calculated by:

$$cor_{(x,y)} = \frac{\sum_{x,y}\left(T_{(x,y)} \cdot S_{(x,y)}\right)}{\sqrt{\sum_{x,y} {T_{(x,y)}}^2 \cdot \sum_{x,y} {S_{(x,y)}}^2}} \tag{4-6}$$

Area $S_{(x,y)}$. is searching within a region defined as $x\epsilon[1, w - d_x^*]$, $y\epsilon\left[1, h - d_y^*\right]$.

A larger value of $cor_{(x,y)}$ in Eq (4-6) represents a higher level of similarity between the template patch and the overlapped area on point (x, y). Since the image can be understood as being continuously duplicated by one repeated pattern, the repetition structures in the image can be obtained by finding all areas identical to the template patch. It should be noted that for efficient calculation, the search is not applied to the whole image but to cropped image $I_{ox} \leftarrow I\left[\frac{1}{3}d_x^*: w, 0: d_y^*\right]$ $I_{oy} \leftarrow I\left[0: d_x^*, \frac{1}{3}d_y^*: h\right]$.

In order to refine d_x^*, T is moved over the cropped image I_{ox} pixel by pixel to search for matched areas along the x direction, and the correlation values in cor^x calculated by Eq (4-6) are recorded. Considering that input digital pattern design images may have diverse content, it is evaluated whether the area $S_{x,0}$ is matched with T by setting a dynamic threshold

$$sim_x = \varepsilon * \ max(cor^x) \tag{4-7}$$

When the first matched area is found at the x^*-th pixel on x direction, i.e., $cor^{x^*} >$

sim_x, the repeated pattern size in x direction is refined as

$$d_x^{**} = x^* + \frac{1}{3} d_x^* \tag{4-8}$$

An illustration is shown in Figure 4-5(a). The drawn repeated pattern grids before and after using the optimization algorithm are presented in Figure 4-5(b) and (c). It can be seen that the detected accuracy is significantly improved after using the optimization algorithm. Moreover, to avoid too small repeat sizes being detected, the adjacent areas both with high cor^x but less than 2-pixel distance are prohibited. If the distance Δd between the adjacent area $S_{w,0}$ and $S_{w+\Delta d,0}$ is smaller than two, then $d_x^{**} = d_x^{**} + \Delta d$, this operation is repeated until the distance $\Delta d \geq 2$. The pixel value of 2 can be determined based on the smallest size of design element. The optimization process on the y direction is the same with the operation in x direction.

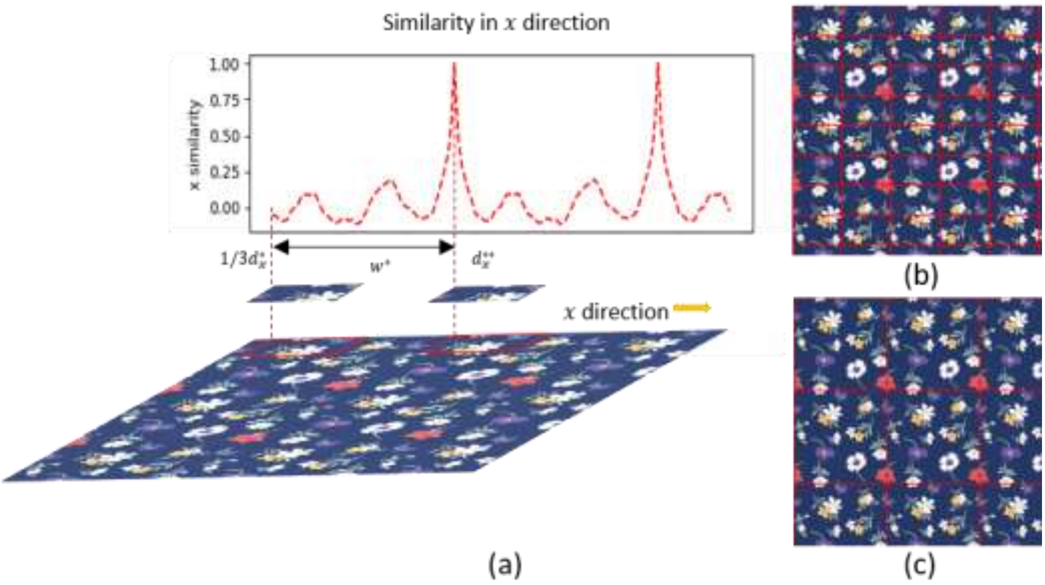


Figure 4-5 (a) Illustration of how the optimization algorithm work in x direction, the correlation value changes with the movement of the template patch and meets our requirements at the point d_x^{**} ; (b) Repeated pattern grid before optimization; and (c) Repeated pattern grid after optimization

4.3.5 Repeated pattern grid extraction

The output from the proposed R-system is grid showing the location and size of the repeated pattern presented on the input images. The refined pattern sizes d^{**} obtained from the previous optimisation step are used to decide the final grid, an a comparative study with the state-of-the-art method (Lettry *et al.*, 2017) will later be given in Section 4.4.

For comparison later, final output of repeated pattern detection by Lettry *et al.* (2017) is listed here. Displacement vectors $d_{fn}^{i,j} \in D_{fn}$ from the selected filters are used to decide the centroids of the repeated pattern by voting. To do so, the displacement vectors are gathered to fit in a module space:

$$M: R^2 \to [0, d_x^{**}] * [0, d_y^{**}] \tag{4-9}$$

$$M(v) \to (v_x \; mod \; d_x^{**}, v_y \; mod \; d_y^{**}) \tag{4-10}$$

where v_x and v_y are the length of Hough-like voting space V in the x and y directions.

Different from Lettry *et al.* (2017), a weighted method is used to decide the final grid using displacement vectors $d_{f_n}^{i,j}$ and refined pattern sizes d^{**}. In other words, to avoid $d^{i,j}$ that is very different from d^{**}, the given $d^{i,j}$ is further screened by $D_{f_n}^{*} = \{d_{f_n}^{i,j} \in D_{f_n}: ||d_{f_n}^{i,j} - d^{**}|| < 5\sigma\}$.

After that, for the reserved $d_{f_n}^{i,j} \in D_{f_n}^{*}$, each $d_{f_n}^{i,j}$ is exponentially weighed according to how consistent they are with the repeated pattern size d^{**}. The weights are then normalized using the number of vectors in each filter $|D_{f_n}^{*}|$ as follows:

$$W_{i,j,f_n^*} = \frac{1}{|D_{f_n}^{*}| + \delta} \cdot exp\left(-\frac{\left\|d_{f_n}^{i,j} - d^{**}\right\|^2}{2\sigma^2}\right) \tag{4-11}$$

where the parameter δ is set to reduce the impact caused by the number imbalance of $D_{f_n}^{*}$ in each filter and δ=0.8 in the experiment, same setting for comparative study with previous work (Lettry *et al.*, 2017; Rodriguez-Pardo *et al.*, 2019).

Since only one filter is selected from each convolutional layer, to avoid biasing certain layers over the others, the σ is set to 1 to treat each layer equally. Then the centroid $c^* = (c_x, c_y)$ is chosen to minimize the weighted average distance of the displacement vectors by the below formula:

$$c^* = \underset{c}{\operatorname{argmin}} \sum_{\substack{n \epsilon N \\ f_n^* \epsilon F_n \\ d_{f_n}^{i,j} \epsilon D_{f_n}^*}} W_{i,j,f_n^*} \cdot \left\| M\left(d_{f_n}^{i,j} - c\right) - d^{**}/2 \right\| \tag{4-12}$$

Finally, an elastic repeated pattern grid is formed based on the identified centroid and optimized repeated sizes d^{**}; a few example results are shown in Figure 4-6.

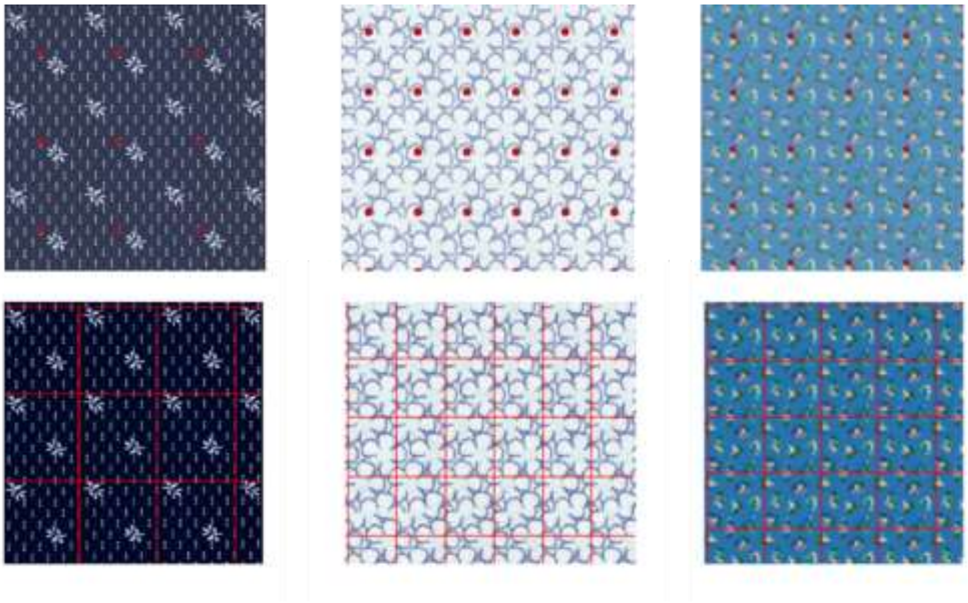


Figure 4-6 Examples of centroids and elastic repeated pattern grids. Top: Images with Centroids. Bottom: Images with elastic repeated pattern grids

4.4 Experiments

4.4.1 Dataset and evaluation metric

There is no publicly available dataset of digital pattern design images that can be used for quantitative analysis. To support the relevant study and evaluate our approach, a repeated pattern detection (RPD) dataset was constructed in this study; and the dataset has 841 images, including 774 scanned fabric images as well as 67 computer-generated design images. These images have rich texture of diverse styles, with regular or and 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 direction. Although the RPD is not huge comparing to other deep learning based model, it was not used for training but for testing. The detection of repeated patterns by R-system is evaluated on every single image, similar to unsupervised learning or self-supervised learning that the optimization is done online based on every single image. Dataset in such size is sufficient for testing in terms of accuracy, robustness and efficiency.

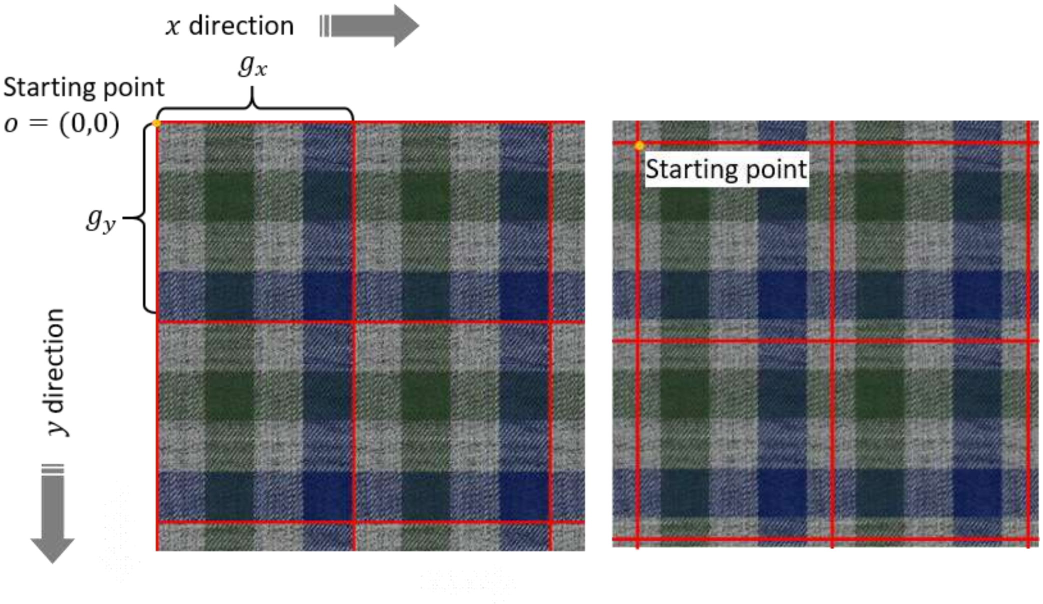


Figure 4-7 Example of ground truth image annotation. g_x and g_y is the ground truth in x and y directions, reapectively. Repeated pattern grid can be found from any point in the same image

Since the output of the current method is represented in form of pattern grid, the repeated pattern size of the grid, instead of location to a specific pixel position, can be used to evaluate the accuracy of repeat pattern detection. This is because a repeated pattern may be found at any place in one image (as shown in Figure 4-7). Thus, repeated pattern grids were drawn from point $o = (0,0)$ for the qualitative evaluation.

To evaluate the accuracy of each approach, all repeated patterns were assumed to start from the same point and the intersection over union (IoU) is calculated to measure quantitively the accuracy of the detected repeated pattern. The IoU (Rezatofighi *et al.*, 2019) is computed as follows:

$$IoU = \frac{min(d_x^{**}, g_x) \cdot min(d_y^{**}, g_y)}{d_x^{**} \cdot d_y^{**} + g_x \cdot g_y - min(d_x^{*}, g_x) \cdot min(d_y^{*}, g_y)} \tag{4-13}$$

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.

4.4.2 Experimental setting

The proposed repeated pattern detection method was developed on Pytorch framework, and Alexnet (Krizhevsky *et al.*, 2012) trained on ImageNet 2012 was selected as the pre-trained CNN. The number of activation peaks of Algorithm 4-1 on page 80 was set as $k = 20$; and the parameter ε from the dynamic threshold Equation (4-7) in the repeated pattern size optimization method was set to 0.9.

4.4.3 Baseline and related work

To objectively evaluate the effectiveness of the proposed method, it was compared with four related methods. Lettry's (2017) method was the first one using 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’s (2019) scheme proposed to select CNN filters of the AlexNet by abandoning CNN 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 4-8. The left is their input image, with uneven yarns and non-uniform color variations, and the right is the regularized image (Rodriguez-Pardo *et al.*, 2019). In the proposed method, pre-processing was necessary and therefore not included. Furthermore, their method returns multiple repeated pattern sizes, instead of an optimised pattern size for each image, the size that is closest to the ground truth is chosen as the final result.

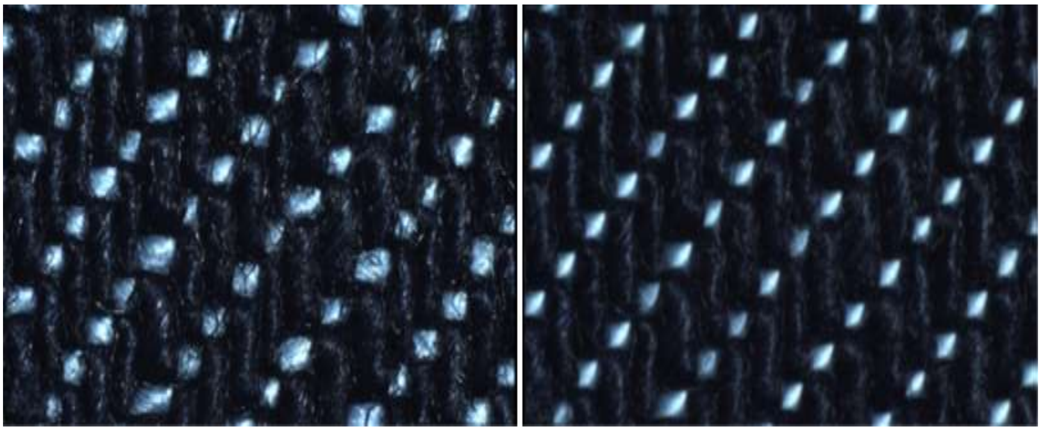


Figure 4-8 Example of pre-processing for image regularization (Rodriguez-Pardo *et al.*, 2019)

Apart from comparing with deep learning based methods, the proposed method was compared with traditional methods, including SIFT (Lindeberg, 2012) for feature extraction. SIFT descriptor is a classic local feature descriptor. To detect the repeated pattern sizes using SIFT extracted, the same voting method (described in Section 4.3.3

and Eq (4-5)) was used.

Another traditional method chosen to compare was Neupane *et al.* (2019), an autocorrelation-based algorithm that did not require definition of template patches in advance. This method copied the input image and cyclic-rotated it by one pixel each time; meanwhile, the cyclic-rotated direction of the repeated pattern was obtained by comparing and minimising bitwise XOR differences between the input image and the cyclic-rotated one. By performing these operations along the *x* and *y* directions of the image, the repeated pattern size was eventually determined.

4.4.4 Experiments results

The proposed R-system is tested on the RPD dataset, and both accuracy, efficiency and robustness are evaluated in comparison to both state-of-the-art (SOTA) traditional and deep learning-based methods. For accuracy evaluation, both qualitative and quantitative analyses were conducted.

4.4.4.1 Qualitative evaluation

For qualitative evaluation, the repeated pattern grids generated by the proposed R-system and other SOTA methods are presented in Figure 4-9 and Figure 4-10, respectively, for real fabric scanned images and computer generated design images. Fabric images are typical digital pattern images with repeated patterns; most of them have rich texture information (see Figure 4-9). With the proposed method, repeated patterns were perfectly detected on these images, especially fabric images with dense patterns (columns 2 and 3 in Figure 4-9). The proposed R-system also performs well on fabric images with multiple design elements (columns 5 and 6 in Figure 4-9). Moreover, the image in column 7 was with many small wrinkles and shadows, and an accurate detection result was obtained without pre-processing. It demonstrated the robustness of our work.

Lettry *et al.* (2017) (row (c) in Figure 4-9) leveraged almost all 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, 6, and 7 in Figure 4-9), they are indeed repetitive patterns that can be tiled to recover the input images. And their detection results are more accurate than those of Rodriguez-Pardo *et al.* (2019) (row (b) in Figure 4-9) and the approach leveraging SIFT (row (d) in Figure 4-9). The main difference between Rodriguez-Pardo's and Lettry's work is that Rodriguez-Pardo introduced filter selection while Lettry had used all filter results. In result, about 1/3 less filters were used in Rodriguez-Pardo's model than that in Lettry's, and comparatively less accurate detection results. SIFT is a local image feature descriptor that does not consider global or heretical image information, and the results are less satisfactory in most of the example images. On the other hand, Neupane *et al.* (2019) had the worst result among all. Neupane *et al.* (2019) was not suitable for repeated pattern detection on real fabric images, as this approach was successfully demonstrated on computer-generated images only. It also demonstrates the superiority of CNN-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 and real-fabric scanned images (see Figure 4-10). Comparatively, the performance of Lettry's, Rodriguez-Pardo's, and SIFT-based methods degraded for images with large number of small and similar design elements, namely motif patterns (see columns 2 to 4 in Figure 4-10), but reasonably good for check designs (columns 5-7). The detection results of Neupane's method in Figure 4-10 again are unsatisfactory, because their method can detecte relatively simple patterns and the output repeat sizes are the smallest. For computer generated design images, the proposed R-system again outperformed all other methods.

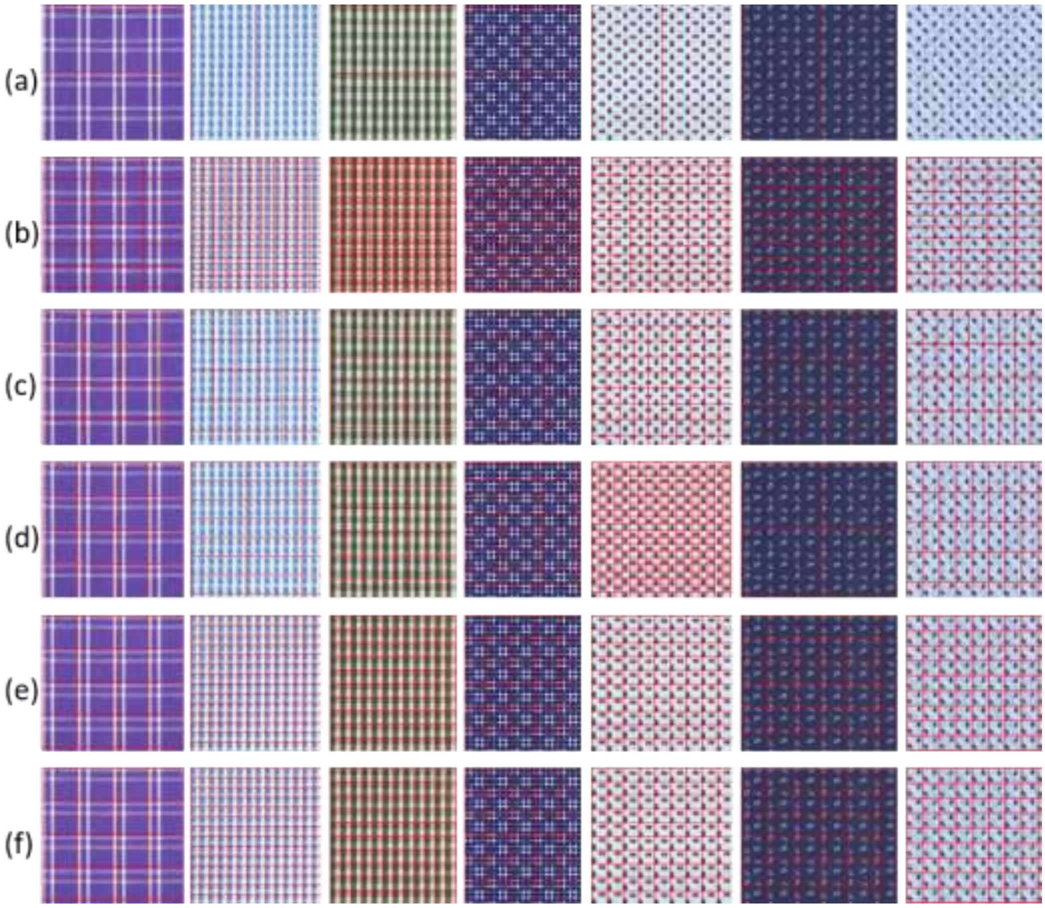


Figure 4-9 Qualitative comparison for processing real fabric images (along different columns) from results of (a) Neupane's, (b) Rodriguez-Pardo's, (c) Lettry's, (d) SIFT's, (e) the ground truth, and (f) the present method of R-system, respectively

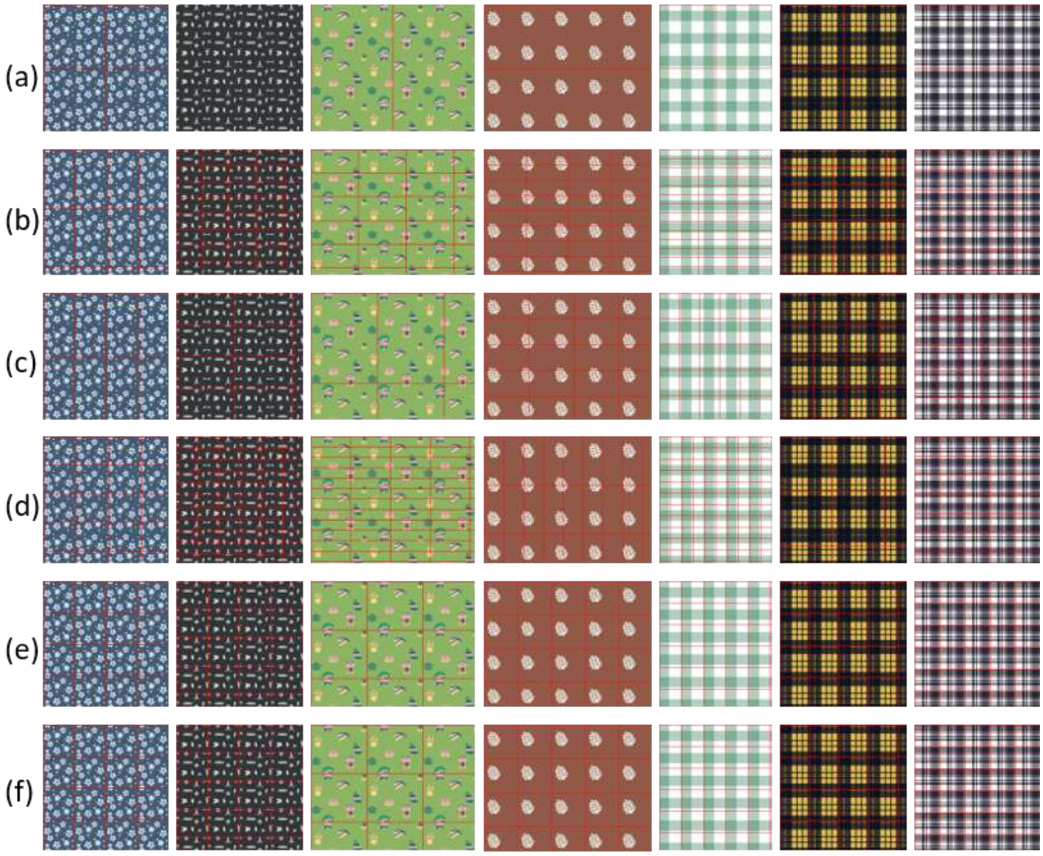


Figure 4-10 Qualitative comparison on computer-generated images (along different columns) from results of (a) Neupane's, (b) Rodriguez-Pardo's, (c) Lettry's, (d) SIFT's, (e) the ground truth, and (f) the present method of R-system, respectively

4.4.4.2 Quantitative evaluation

A quantitative evaluation using the RPD dataset and computed time cost and resulting detection accuracies of various methods are compared in Table 4-1. Again, in the table, quantitative analysis was separated into two subsets of data: computer-generated images and real fabric scanned images.

Table 4-1 Summary of quantitative results. Accuracy and time cost over the different subsets of our dataset

Image type	Neupane *et al.* (2019)		SIFT-based approach		Rodriguez-Pardo *et al.* (2019)		Lettry *et al.* (2017)		The current work	
	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)
Computer-generated	0.291	0.3	0.308	65.4	0.379	55.0	0.713	731.3	**0.844**	**46.3**
Fabric	0.116	0.2	0.335	61.0	0.409	49.0	0.543	335.8	**0.658**	**39.4**
All images	0.130	0.2	0.333	61.3	0.407	49.4	0.557	366.7	**0.673**	**40.0**

Computer config: CPU Intel i7-6700k, GPU GeForce GTX TITAN X, Memory 16GB.

The overall accuracy of the R-system is the highest (0.673 on average) among all methods, and so is the subsets' performance: the average accuracy of real fabric scanned images is 0.658, and 0.844 for computer-generated images. Lettry's average accuracy of fabric and computer-generated images are 0.543 and 0.713, respectively, which are both higher than the corresponding results of Rodriguez-Pardo. Especially in the computer-generated category, Lettry's result is nearly twice better than Rodriguez-Pardo's. These demonstrate the influence of the amount of selected filters on the accuracy of the method; the fewer filter used, the lower the accuracy; and it also explains the reason for adding an optimization process. Moreover, Neupane and SIFT-based methods perform poorly quantitatively on both fabric and computer-generated images. Neupane's work has the worst performance among all methods. The experimental results have proved the outstanding design of the R-system, which performs best on both fabric and computer-generated images.

The time cost of R-system is 40s on average per image of the same input size, which is

1/9 of Lettry's time cost, about 10s less than Rodriguez-Pardo's, and about 20s less than the SIFT-based approach. Although Neupane's work takes the shortest time, its accuracy is the lowest. Therefore, the experiments have proven that the R-system achieves a good balance between accuracy and efficiency.

Table 4-2 Summary of quantitative results of our work with (w/) or without (w/o) applying optimization algorithm and other feature-based approaches

Image type	SIFT-based approach		The current work w/ filter selection and w/ optimization		The current work w/ filter selection but w/o optimization		Rodriguez-Pardo *et al.* (2019)		Lettry *et al.* (2017)	
	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)
Computer-generated	0.308	65.4	0.844	46.3	0.322	46.2	0.379	55.0	0.713	731.3
Fabric	0.335	61.0	0.658	39.4	0.337	39.2	0.409	49.0	0.543	335.8
All images	0.333	61.3	0.673	**40.0**	0.336	**40.0**	0.407	49.4	0.557	366.7

Computer config: CPU Intel i7-6700k, GPU GeForce GTX TITAN X, Memory 16GB.

To examine the effectiveness of the module for filter selection (described in Section 4.3.1) to optimization algorithm (Section 4.3.4) of the R-system, an ablation study was conducted and reported in Table 4-2, in terms of accuracy and time cost with and without adopting the corresponding module/algorithm. As shown in Table 4-2, the accuracy of the SIFT-based approach is lower than any CNN-based one, including the result of the R-system before optimization. It proves that the selected filters still have the ability to extract useful features comparable to local feature descriptors. In other words, only a small number of selected filters from different convolutional layers can effectively extract features for the task of repeated pattern detection.

Moreover, the accuracy of the R-system without optimization module is similar to that of Rodriguez-Pardo but slight less time cost. It means that the proposed filter selection method in Section 4.3.1 using boundary results can substantially lower the computational cost but maintain reasonable features for repeat detection. As shown in Table 4-2, the time cost before and after optimization is almost the same (both are 40 seconds), and it demonstrates that the proposed optimization algorithm dramatically improves detection accuracy is without increasing the time consumption.

Table 4-3 Summary of each step's time cost (s) over all the CNN-based methods

Method	Filter Selection	Activation peaks extraction	Consistent displacement vector selection	Repeated pattern size optimization	Total
Rodriguez-Pardo et al.	0.1	0.0	51.4	*NaN*	51.6
Lettry et al.	0.1	36.1	327.1	*NaN*	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 16GB.

For further analysis of the efficiency of various modules of the R-system, the per-step time costs of the R-system and other CNN-based methods are summarized in Table 4-3. Consistent displacement vector selection is the most time-consuming step of all the above CNN filter-based methods. The consistent displacement vector selection uses a 2-dimensional Gaussian distribution to fuse the activation peaks across different layers. Displacement vector number and running time are positively correlated (explained in section 4.3.3). In other words, the more filters used in the method, the more displacement vectors will be created and thus the more time it takes for the voting process. As a result, the number of filters is negatively correlate to the final runtime. Consequently, by applying a filter selection algorithm, Rodriguez-Pardo's work and the current R-system have more efficiency results than that of Lettry's. Table 4-3 has proved that the accuracy before or after the applied optimization algorithm would not change obviously as long as the approach selects the same number of filters for the whole pipeline.

4.5 Discussion

4.5.1 Experiments on other type of images (non-pattern design ones)

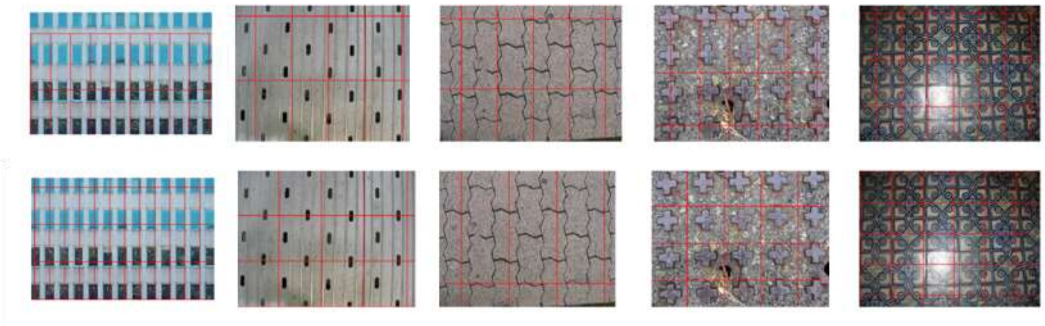


Figure 4-11 Detection results of environmental images. Top: the detection results from Lettry *et al.* (2017). Bottom: the results from our approach

Lettry *et al.* (2017) was the first proposal of using pre-trained CNN to assist repeated pattern detection on images, and they applied and tested their method on various real world images including images other than design images, e.g. environmental image that are fronto-planar. With this as the baseline method, the proposed R-system with various modules to better address our target image type -- for digital pattern design images. As illustrated in Figure 4-2, repeated pattern images are different in terms of content diversity and topological regularity. It is interesting to explore the proposed R-system can also work on images other than digital pattern design ones. The system was tested on some manufactured environmental images, some images of the testing dataset of Lettry *et al.* (2017). Figure 4-11 shows some results from Lettry and our work. It demonstrates that although the proposed R-system uses fewer filters, similar detection results as theirs are achieved for non-pattern design images. Surprisingly, the R-system performs better on the $5th$ image that has a very strong illumination difference. It certifies that our method could achieve satisfactory results on fronto-planar manufactured environmental images with significant varied content. As a future work, it will explore different system configuration for serve other types of image, which may vary in the dimension of content and topology regularity.

4.5.2 Impact of using different CNN models

AlexNet was used in the current proposal as CNN baseline for feature extraction. Experiments were also conducted on the RPD dataset, replacing the pre-trained Alexnet network with two other CNN models commonly used for feature extraction: the VGG16 network (Simonyan *et al.*, 2014) and the Resnet50 network (He *et al.*, 2016). 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 Resent50 has five convolutional stages, and each contains one or several convolutional layers. Similarly, when the Resnet50 network was used, one filter from the last layer of each convolutional stage was selected. Overall, our work selected five filters from each pre-trained CNN covering various scales of information, and the left parts of the whole pipeline remain unchanged. The quantitative results are compared in Table 4-4.

Table 4-4 Summary of quantitative results of using different pre-trained CNNs

Image type	Alexnet		VGG16		Resnet50	
	Accuracy	Time (s)	Accuracy	Time (s)	Accuracy	Time (s)
Computer-generated	0.844	46.3	0.880	455.4	0.860	232.2
Fabric	0.658	39.4	0.653	96.3	0.655	72.1
All images	**0.673**	40.0	**0.664**	113.5	**0.675**	85.7

Computer config: CPU Intel i7-6700k, GPU GeForce GTX TITAN X, Memory 16GB.

Table 4-4 shows that it will increase time cost if more complex models are used. Although the deeper CNN models perform better at feature extraction in the conventional sense, considering the balance of accuracy and computation cost, pre-trained AlexNet can provide a rich enough features for repeated pattern detection. It can be future work to explore best way of feature extraction in different CNNs for repeated pattern detection.

4.5.3 Impact of different boundary detection methods

In the proposed R-system, Canny edge detector was used to provide binary boundary results for filter selection. Experiments were conducted to compare the effectiveness of other boundary/edge detector, namely a learning-based method named BDCN (He *et al.*, 2020) for filter selection and feature extraction. BDCN showed better performance on edge detection than Canny on the *BSDS500* dataset (He *et al.*, 2020). From Table 4-5 the overall accuracy of using Canny and BDCN are the same. However, the method using Canny takes an average of 5 seconds less than the one using BDCN. Therefore, the Canny edge detection is chosen in the proposed filter selection algorithm.

Table 4-5 Summary of quantitative results of using different boundary detection methods

Image type	Alexnet + Canny		Alexnet + BDCN	
	Accuracy	Overall time (s)	Accuracy	Overall time (s)
Computer-generated	0.844	46.3	0.876	86.8
Fabric	0.658	39.4	0.656	41.0
All images	**0.673**	**40.0**	**0.673**	**44.7**

Computer config: CPU Intel i7-6700k, GPU GeForce GTX TITAN X, Memory 16GB.

4.6 Chapter Summary

In this chapter, a novel approach for repeated pattern detection on unknown images with digital patterns (R-system) has been proposed. Unlike previous methods only focus on image feature extraction or prediction model optimization, the proposed R-system combines CNN filters and template matching to achieve the best balance in accuracy and efficiency. The R-system have introduced a CNN filter selection method leveraging boundary detection results and an optimization approach based on template matching. A dataset (RPD) containing real fabric scanned images and computer-generated images was built. The experimental results based on the RPD dataset have demonstrated that the R-system achieves a good balance between accuracy and efficiency. Moreover, the R-system works not only on digital pattern design images but also gets promising results on fronto-planar manufactured environmental images.

In addition, experiments were conducted to assess the effectiveness of each module/component in the proposed R-system. Specifically, the impacts of using different CNN models and boundary detection methods have been discussed. By replacing the existing AlexNet with two deeper CNN models, VGG16 and ResNet50, it has been found that the final accuracy will not be significantly affected when the number of filters selected is fixed. However, the deeper CNN will take a longer running time. Therefore, to achieve satisfactory accuracy but lower time costs, the AlexNet is chosen for the task. Furthermore, by replacing the existing boundary detection method, the Canny edge detector, with a deep learning-based boundary detector, little improvement on overall accuracy is resulted but higher time cost. Therefore, the Canny edge detector is recommended for boundary detection.

CHAPTER 5. Automatic Design Elements Extraction and Vectorization

5.1 Background Introduction

Digital pattern design is an important visual communication tool in the modern world, encompassing everything from printed fabrics to book covers to website design, and is especially important for fashion product development (Ambrose *et al.*, 2019; Li *et al.*, 2019). Digital pattern designs composed of various design elements presented on either two-dimensional surfaces or three-dimensional forms (Demir *et al.*, 2021; Guerrero *et al.*, 2016). Design principles are used in the composition of pattern designs, and these design principles mainly include the geometric arrangement of design elements, also named layout, satisfying human aesthetics (Zhang *et al.*, 2020). Before arranging different layouts of design contents as digital patterns, artists should first prepare sufficient contents, namely different design elements. Design elements can be lines, bands, points, regular geometry as well as abstract shapes created by artists. Common digital pattern designs for textiles are classified into stripe, check, and motif patterns in this study, and a detailed explanation will be given in later CHAPTER 6 on the classification. This chapter is focused on research design element extraction from digital motif patterns. An example of a typical motif digital pattern consisting of multiple repeating or non-repeating design elements regularly layout over a background layer is shown in Figure 5-1(a).

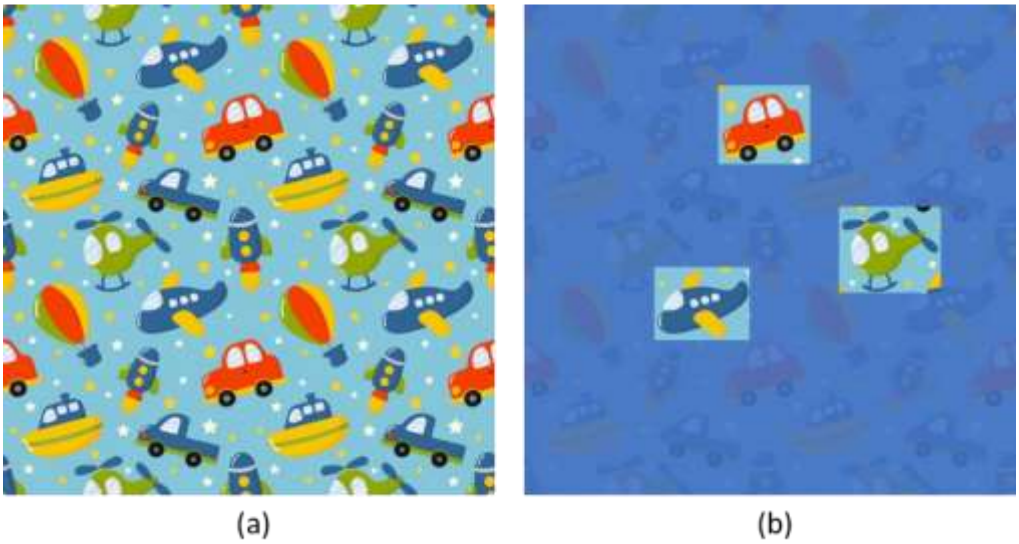


Figure 5-1 (a) A typical digital pattern image that often appears around us, it contains several colorful design elements and a light blue background layer; (b) examples of design elements within (a)

Since digital pattern is powerful means for communication, creating digital patterns is indeed the desire for both professional designers and non-professional users (Shen *et al.*, 2021). However, creating a novel design from scratch is time-consuming and tedious (Kovacs *et al.*, 2018; O'Donovan *et al.*, 2015). An alternative and efficient way is to synthesize new designs from existing ones. Many works focus on exploring the layout of digital pattern design (Guerrero *et al.*, 2016; Li *et al.*, 2019; O'Donovan *et al.*, 2015), while research on extracting design elements has been given little attention. It would be useful to extract core design elements automatically and efficiently from existing design images and reuse them in new designs, and this indeed align with the common practice in the design professional.

Raster images and vector images are two standard image formats serving different purposes for digital pattern design. Raster images use rectangular grids of colored pixels to represent content, which is a format mainly for display on the screen (Inglis *et al.*, 2012). However, they cannot be edited for further reuse, and the content inside will be distorted after scaling. In contrast, vector images consist of several *vector paths*. Each vector path is mathematically defined by its control points and filled with specific colors, representing a certain region when the vector image is visualized (Favreau *et al.*, 2017). Vector images thus have several advantages, such as better geometric editability, resolution independence, and smaller memory volume (Bergen *et al.*, 2012; Dominici *et al.*, 2020; Hoshyari *et al.*, 2018). Image vectorization, also known as image tracing, is the process of converting raster images into vector images (Inglis *et al.*, 2012). Due to artist-drawn design elements containing distinct colored regions separated by sharp boundaries, they are perfect for vectorization.

Previous works proposed for vectorizing digital pattern images mainly contain a single design element. Many of them focus on how to generate vector paths whose shapes correspond to human shape perception (Dominici *et al.*, 2020; Hoshyari *et al.*, 2018). However, these methods act on the pre-processed intermediate representations rather than directly on the input image itself. What it means by pre-processed intermediate representation refers to separated single-color regions, also named geometric primitives, an example is shown in later Figure 5-3(b). Other methods and commercial software, for example, Adobe (2022) and VectorMagic (2021), that directly act on input images, generally create vector images containing a large number of short paths that faithfully represent the images in great details, but the number of paths is too large to edit further and the paths are not meaningfully connected like those manually drew by designers using design software. These meaningless paths usually have a very small area and must be removed manually before subsequent use (see Figure 5-3(c)). Digital pattern image vectorization is not purely for generating vector paths that can reappear the target image precisely. Rather, a vector image in the design is used as a visual inspiration for a new, geometric approximation of that image. Therefore, compact vector images that retain the main content are more in line with design needs.

In CHAPTER 4, an efficient repeated pattern detection method has been proposed. Since the existing fashion decorative patterns or graphic design images that can be used as design resources are often composed of repeated patterns, the repeated pattern detection method can reduce computational cost of the subsequent operations by keeping the small units and core content. Therefore, for images with repeated patterns, twice of the repeated pattern, as shown in Figure 5-2(b), are input to the current image analysis step.

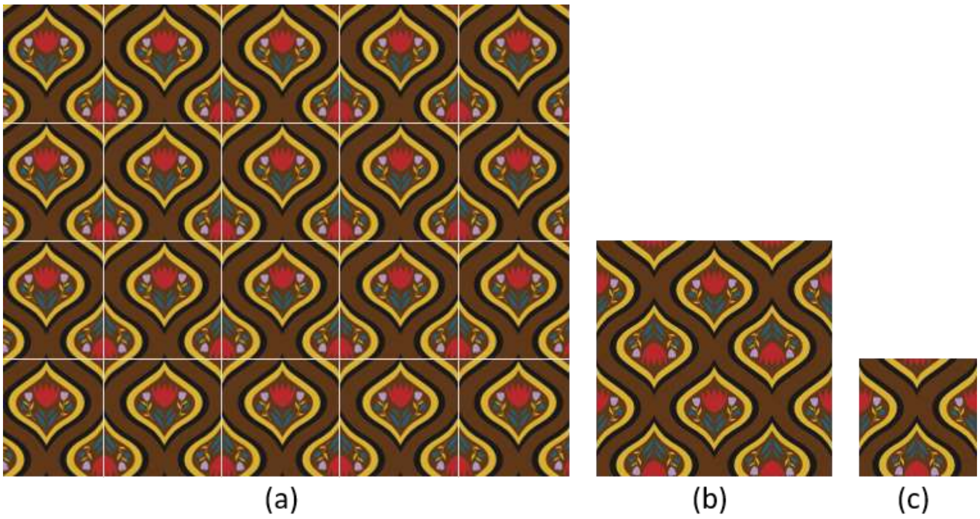


Figure 5-2 (a) A typical image of digital patterns with repeated patterns; (b) The image is cropped to twice the size of its repeated pattern; (c) Repeated pattern of digital pattern in (a). Comparing (b) with (c), the design element in (c) is separated into four independent parts, while (b) contains at least one complete design element

In this chapter, the focus is on automatically extracting core design elements from an *unknown* image and compactly vectorizing them for further processing. An intelligent system, denoted as E-system, is proposed and developed for this task. The input images to the E-system are assumed to have the following characteristics: (1) The image contains one or several repeating or non-repeating design elements, and such elements are composed of multiple closed regions in different colors. (2) There is separation between design elements in the image. In other words, design elements do not overlap with each other on the image. (3) The image has clear background and foreground content, and design elements are regarded as foreground content and the color of the background is uniform.

Design elements can be considered as objects in the foreground of the design image, so the task of extracting design elements from the design image is more like an object detection task in which detecting each design element and marking it in a bounding box or a supervised segmentation task which separates each design element with a corresponding mask. The best-performing methods for each of these two tasks of object detection and segmentation are deep learning based. However, to the best of our knowledge, there is no publicly available dataset on design images with labeled design elements, while building a labeled dataset on design elements can be very costly. Considering that unsupervised segmentation, being one of the segmentation tasks, can effectively separate an image's foreground contents from its background, an unsupervised segmentation method is thus leveraged to extract foreground design elements from graphic design images. After separation of foreground and background, a color quantization method was proposed to deconstruct each design element into different colored components or regions. Lastly, the design elements are vectorized one by one. Before the detail discussion on the method for design element extraction and vectorization, the related work in the literature is first discussed.

5.2 Related Work

The proposed E-system is capable of performing end-to-end automatic extraction and vectorization of core design elements in single images, and there is no reported work in the literature working on the same task. Although there is no exact single system similar to the proposed E-system, it is closely related to several domains of study in computer vision and graphics, including object detection, image segmentation, region and image vectorization.

5.2.1 Design elements detection

Digital pattern design images in the present study are composed of several foreground design elements on a uniform-colored background layer. Very few reported works focus on a similar task as the current proposed E-system, namely detecting design elements from a single image. The most similar work found in the literature was by Cheng *et al.* (2010), who used boundary information to detect and extract repeated objects in an interactive manner that users were required to help localize possible object regions with scribbles. Different from Cheng *et al.* (2010), the goal of this study is to extract design elements *automatically*. Design elements can be thought of as regularly laid foreground objects on a uniform color image background. Design elements extraction should consider the content and locations that can be linked to object detection or image segmentation problems.

As discussed in Sections 2.2.3 and 2.2.4 of CHAPTER 2, both object detection and image segmentation are image processing tasks. Object detection is a task to localize instances of particular object classes in an image (Inoue *et al.*, 2018; Zou *et al.*, 2019), while image segmentation aims to partition an image into multiple segments and regions of interest, representing different objects (Vitale *et al.*, 2016). The state-of-the-art methods for both object detection and image segmentation are based on supervised learning; and a typical example includes segmenting natural scene images, which pixels are with labels of *known* categories, like road, mountain, sky, and sea. Nevertheless, the supervised semantic segmentation approach may not be applicable in the current task of extracting design elements from single images for twofold reasons. Firstly, there is no dataset of labeled design images. Design images seldom have pixel-level semantic labels and their contents are very different from real-world images like natural scenes and object images. Secondly, the diversity of design images makes it difficult, if not impossible, to define semantics for all design elements and build a labeled dataset large enough for training purposes.

Alternatively, other image segmentation approaches have utilized traditional image features to separate foreground contents from the image background, based on graph theory (Felzenszwalb *et al.*, 2004) or multivariate image analysis (Comaniciu *et al.*, 2002). These methods have a known drawback of over-segmentation (Zhang *et al.*, 2009), namely generating a large number of regions and meaningless shapes, which have to be further processed and regrouped, such as by clustering techniques. Nevertheless, clustering itself is a challenging research topic in image processing, and its effectiveness are largely dependent on the content of the target images.

In the family of deep learning, unsupervised segmentation has received a great deal of attention in recent years, since it has taken advantage of the powerful CNNs to classify pixels with similar features to the same label and continuously optimize the segmentation results and CNN parameters through backpropagation (Kanezaki, 2018; Kim *et al.*, 2020). Such methods are especially good at separating the foreground and background of the image. Moreover, they do not require large labeled datasets. In other words, unsupervised segmentation is effective to segment the contents of *unknown* images into the foreground and the background. Therefore, unsupervised segmentation is adopted to assist foreground design element extraction in the proposed E-system.

5.2.2 Design elements similarity

A complex digital pattern design can be created by manipulating a few design elements with simple transformations, such as translation, rotation and scaling (Ambrose *et al.*, 2019). Taking reference to the above design principle, the second step of design image understanding is by identifying and isolating all core design elements present in a design image, and such design elements vary in size, shape and orientation because of the aforementioned transformation used in the design creation process. Once identified, these design elements can be reused by modification and transformation to come up with new digital pattern designs. Nevertheless, identification and localization of design elements present in a single design image can be, to some extent, comparable to object detection, one of the typical tasks of computer vision.

Determining the similarity between instances (design elements) present on a single image is itself a challenging task in computer vision, and the traditional approaches are again based on image features. Nevertheless, there is no single feature can work well in all scenarios, neither for powerful image descriptors like scale-invariant feature transforms (SIFT) (Ng *et al.*, 2003), because design elements have distinct image features, such as corner, edge, and contour. Some researchers use CNNs' powerful feature extraction ability to compare the similarity between images (Appalaraju *et al.*, 2017; Shimoda *et al.*, 2017). A CNN can be viewed as a function $\mathcal{F}$ that transforms each input image I into a set of vectors x, based on a set of parameters ϑ, $x = \mathcal{F}(I, \vartheta)$. The parameters ϑ contains all the fixed weights and biases for the convolutional and inner product layers. However, in such approach, CNNs are used for compare similarity between entire images and standard input image size is set to 224×224 px. In current task of identifying similar regions (design elements) within single input images, direct application of CNNs based method is suggested, the extracted design elements are small in sizes, not meeting the CNN's input requirements, and it is too costly to train CNNs can cover so diverse design elements.

In this study, perceptual hashing (pHash) is used for comparing design element similarity. The pHash is a *fingerprinting* algorithm encoding the result of the discrete cosine transformation that produces a fingerprint of the image at a fast speed (Monga *et al.*, 2006). For example, Hua *et al.* (2019) proposed using the pHash method to create a similarity index of cultural artifacts; Kumar *et al.* (2021) used pHash to compare the content similarity to protect copyright of multimedia; and Zannettou *et al.* (2018) used pHash to extract representative feature vectors from *meme* images. The method works for images of any intensity and resolution, not just ones with strong features. When the pHash values of two images are “close” to one another, it means that the image contents are similar. Hence, by comparing pHash values of two design elements, the similarity between the two can be efficiently assessed.

5.2.3 Design elements vectorization

Image vectorization, also known as image tracing, is the process of converting a raster image into a vector image. It is important to note that there is no single ‘ground truth’ result for vectorization because the same raster image can be described by many possible vector paths as long as all the traced paths truly represent the input raster image. In other words, the same region of pixels can be represented by hundreds of low-resolution closed paths or a single path clustering all pixels with the same visual content (RGB value).

Learning-based image vectorization methods are still restricted to fonts and simple icon domains (Lopes *et al.*, 2019; Reddy *et al.*, 2021), where the outputs are well-defined.

Most classic methods use closed Bezier curves (vector paths) to describe intermediate representations generated by boundary detection or image segmentation. For instance, many related works rely on mean-shift clustering (Comaniciu *et al.*, 2002). Since raster images are composed of matrix of pixels, they naturally have jagged edges (see Figure 5-3(a)) and thus easily generate far too large number of small paths due to over-segmentation. Consequently, the output vector paths are sometimes redundant and large in number due to inaccurate segmentation or lack of curve priors (Adobe, 2022; Inkscape, 2020; Kerautret *et al.*, 2019; Yang *et al.*, 2015; Zhang *et al.*, 2009). Even with some algorithms devoted to addressing these problems, there is still no single good solution. Recently, Dominici *et al.* (2020) and Hoshyari *et al.* (2018) proposed approaches to generate vector images consistent with human expectations. However, their methods assume the input image is pre-processed (deconstructed) into multiple color intermediate representations. In addition, these methods are computationally expensive with long processing time.

The goal of this study is to directly vectorize design elements extracted from *unknown* images ***without*** human intervention. High fidelity, efficiency and compactness are key factors to consider, it is proposed to obtain compact vectorized results by optimizing outputs from the classic Potrace (Selinger, 2003).

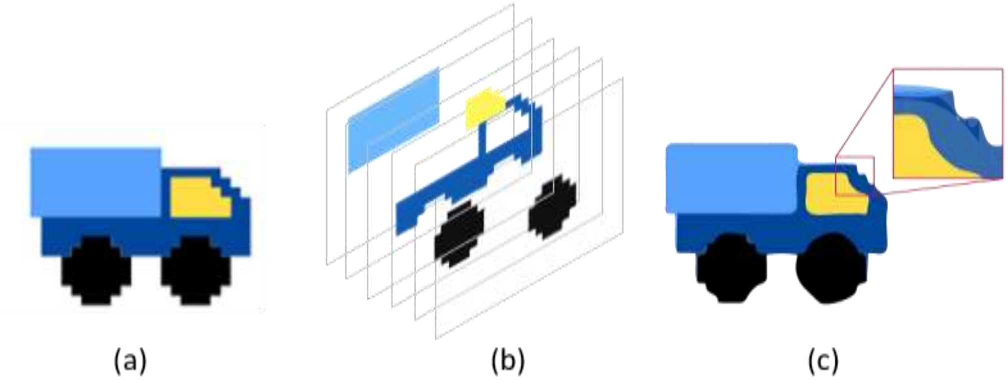


Figure 5-3 (a) The input raster image; (b) intermediate representations of (a), each intermediate representation will be described by a path in the following vectorization algorithm; (c) Adobe (2022) vector result of (a), which contains some tiny and meaningless paths that must be removed before reuse (highlighted in zoomed-in insets)

Moreover, to get a compact vectorized output, the number of intermediate representations should be made as small as possible. Towards simultaneously obtaining the corresponding color information of each intermediate representation, the color information is used to deconstruct the image. Traditional works rely on mean-shift clustering (Comaniciu *et al.*, 2002), which is slow to process and prone to over-segmentation errors. The simplest way to computationally describe the color information of a design element is by means of its color palette, which contains the representative colors within an image, just like a painter's palette. The color palette can easily be obtained from a color histogram (Bradski, 2000), as well as the number of colors (Delon *et al.*, 2005; Srivastava *et al.*, 2015). K-means clustering (Likas *et al.*, 2003) can efficiently deconstruct an image into k regions, however, the region number

k should be provided in the algorithm. An adaptive design element deconstruction method is therefore proposed based on k-means and setting k to the number of colors.

5.3 Method

The E-system, as illustrated in Figure 5-4, is proposed to vectorize the core design elements within an *unknown* digital pattern design image, which covers two main stages of image processing. Each stage of processing includes a sequence of steps, for instance, for core design element extraction in stage one, a CNN is developed and trained for unsupervised segmentation to extract the image foreground from its background; then core design elements are selected through similarity comparison based on the perceptual hashing algorithm (pHash). Next, in the second phase of core design element vectorization, each identified core design element in stage one is deconstructed into several intermediate representations by color quantization and the k-means algorithm (Section 5.3.2.1). After that, each intermediate representation is vectorized by Potrace (Selinger, 2003) and removes the meaningless vector paths. Finally, several compact vectorized design elements will be obtained after synthesizing the vector paths ordered by area (Section 5.3.2.2). The two-stage process with detailed operations are explained in the following subsections.

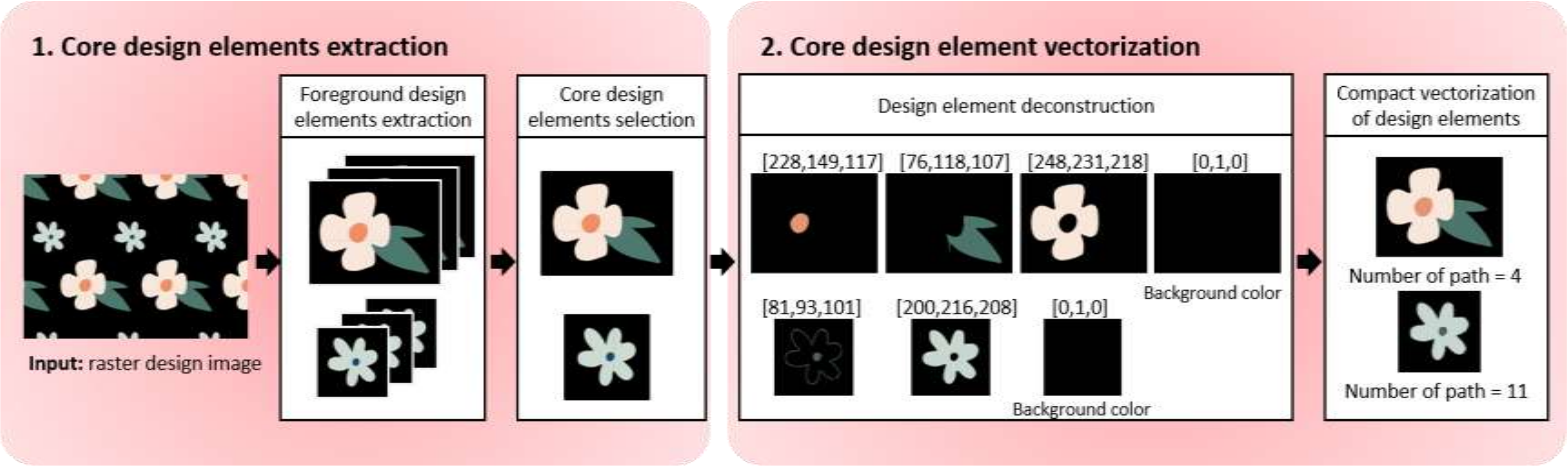

1. Core design elements extraction
Input: raster design image
Foreground design elements extraction
Core design elements selection
2. Core design element vectorization
Design element deconstruction
[228,149,117]
[76,118,107]
[248,231,218]
[0,1,0]
Background color
[81,93,101]
[200,216,208]
[0,1,0]
Background color
Compact vectorization of design elements
Number of path = 4
Number of path = 11

Figure 5-4 Illustration of the E-system for core design elements extraction and compact vectorization

5.3.1 Core design elements extraction

5.3.1.1 Foreground design element extraction

In order to separate an image into an arbitrary number of foreground design elements and a background layer, it leverages an unsupervised segmentation approach in current study. Unsupervised segmentation (Kanezaki, 2018; Kim *et al.*, 2020) can be considered as assigning pixels of an *unknown* image to cluster labels based on pixels' feature representations similarity (for instance, color and texture) and pixels' spatial distance. Considering the excellent feature extraction ability of CNN, a CNN is adopted to extract pixel-level feature representations and classify pixels with similar features into the same cluster labels. Subsequently, the cluster results are iteratively refined by a joint-learning back propagation approach. Finally, the pixels are assigned to an arbitrary number of cluster labels $\{c_n\}_{n=1}^{q'}$, pixels with the same cluster label compose a segmentation mask which means the image is labeled by q' segmentation masks. A large number of cluster labels indicate over-segmentation, whereas a small number of cluster labels indicate under-segmentation. Since over-segmentation is unavoidable, the independent closed areas (also named connected regions) in the background mask are utilized to extract the design elements of the foreground in the proposed E-system.

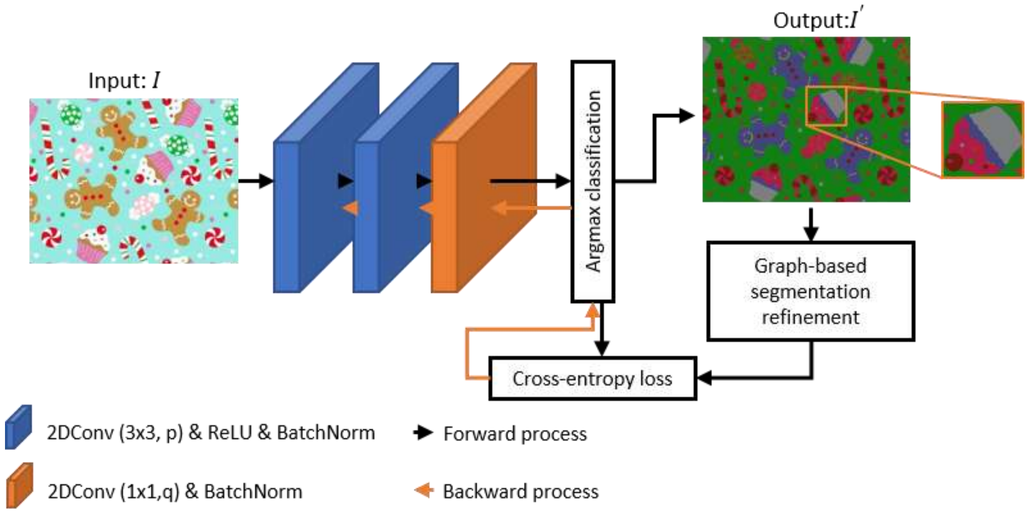

Input: I
Output: I'
Argmax classification
Graph-based segmentation refinement
Cross-entropy loss
2DConv (3x3, p) & ReLU & BatchNorm
2DConv (1x1,q) & BatchNorm
Forward process
Backward process

Figure 5-5 Illustration of training the unsupervised segmentation algorithm

CNN architecture and forward propagation

Given an input RGB image $I = \{v_n \in \mathbb{R}^3\}_{n=1}^N$, where N denotes the number of pixels, and each pixel value v_n is normalized to $[0, 1]$. A CNN with two convolutional components is constructed (see Figure 5-5). Each convolutional component comprises a convolutional layer that has p convolutional kernels of region size 3×3, a ReLU activation function, and a batch normalization function (BatchNorm). The parameters $\{W_a\}$ of convolutional components are initialized with Xavier initialization (Glorot *et al.*, 2010). The output is a p-dimensional feature map $\{x_n\}$ from $\{v_n\}$.

Next, a response map $\{y_n = W_c x_n\}_{n=1}^N$ is obtained after a linear classifier, a convolutional layer with q convolutional kernels of region size 1×1, where $W_c \in \mathbb{R}^{q \times p}$. The response map is normalized to $\{y_n'\}_{n=1}^N$ of zero mean and unit variance. The motivation behind the normalization function is to give each y_n' nearly even chance in the following operations aiming to avoid an under-segmentation failure ($q' = 1$). Subsequently, the $\{y'_n\}$ are assigned into q clusters. The ith cluster of the final responses can be represented as:

$$C_i = \{y'_n \in \mathbb{R}^q | y'_{n,i} \geq y'_{n,j}, \forall j\} \tag{5-1}$$

where $y'_{n,i}$ denotes the ith element of y'_n. It is equivalent to assigning pixels to the nearest point among q representative points, which are placed at an infinite distance on the respective axis in q-dimensional space. Note that C_i can be $\emptyset$, and therefore the number of final unique cluster labels q' is arbitrary from 1 to q. The above process is the forward propagation for self-training the network. See Figure 5-5 for an illustration.

Learning network by joint-learning back propagation approach

There is no clue of how many segmentation masks will be generated for an image in unsupervised segmentation. The segmentation results depend on the image content. However, in image segmentation, it is reasonable for the clusters of pixels to be spatially continuous. To consider the pixel's location, a pre-segmentation algorithm is added. Different from Superpixel used in Kanezaki (2018), a Graph-based Segmentation (GS) (Felzenszwalb *et al.*, 2004) (Section 2.2.3 for more details) is chosen for faster convergence and robustness. Therefore, when the image $I = \{v_n \in \mathbb{R}^3\}_{n=1}^{N}$ is input, T segments are simultaneously extracted from GS, defined as $\{GS_t\}_{t=1}^{T}$, and all of pixels in GS_t are forced to be assigned the same cluster label. T is always bigger than q, which means pixels in different segments will be aggregated into a new cluster label in the learning process. More specifically, Let $|c_n|_{n \in GS_t}$ be the number of pixels in GS_t that belong to the c_nth cluster, the cluster label with the maximum number of pixels, c_{max}, is selected, where $|c_{max}|_{n \in GS_t} \geq |c_n|_{n \in GS_t}$ for all $c_n \in \{1, \dots, q\}$. The cluster labels are then replaced by c_{max} for $v_n \in GS_t$. This process is the graph-based segmentation optimization (see Figure 5-5), and the output will be used in the next iteration.

The essence of unsupervised segmentation is equivalent to interactively solving the following two sub-problems: the first one is using the network with fixed parameters to predict the cluster labels, while the second one is updating network parameters using a backward gradient descent process. In this study, stochastic gradient descent with momentum is leveraged, and specifically, the cross-entropy loss between the network responses $\left\{y_n^{'}\right\}_{n=1}^{N}$ is calculated and the refined cluster labels $\left\{c_n^{'}\right\}$ is leveraged to update the parameters of convolutional components $\{W_a\}$ as well as the classifier $\{W_c\}$. Through iteratively implementing this forward-backward process M times, the final prediction of cluster labels $\{c_n\}_{n=1}^{q'}$ and a group of image pixels in each cluster/segment $S = \{s_n\}_{n=1}^{q'}$ are obtained. The segmentation result I' with q' segments (each represented as a segment mask) is the final output of the process. Figure 5-5 gives an example of how the unsupervised segmentation algorithm works.

Although the proposed unsupervised segmentation method endeavors to prevent under-segmentation failure, the segmentation result is unavoidably over-segmented, meaning a complete design element will be assigned with multiple cluster labels (see an example as highlighted in the zoomed-in inset of Figure 5-5). The next task of core design element extraction is to integrate design element fragments with different cluster labels into a complete one, and this task is itself computationally challenging. To do so, the background mask is first selected, which can provide location clues for the identification of foreground design elements (see Figure 5-6).

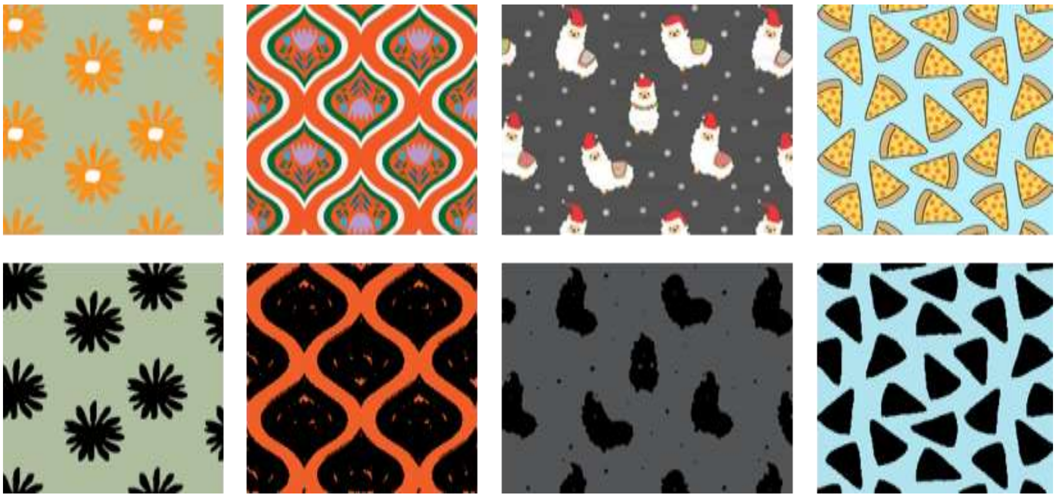


Figure 5-6 Examples of selecting background mask. Top: the input design images. Bottom: their corresponding background masks

The input image is separated into several segment masks by the unsupervised segmentation algorithm, which each mask has no clear semantic label. In order to identify the background mask, the minimum Bounding Rectangle (minBoundingRect) of each segment mask in S is computed. When design elements do not overlap each other within the image, the mask with the largest bounding rectangle area can be identified as the background mask s_{bk} as follows:

$$s_{bk} = max(minBoundingRect(S)) \quad (5\text{-}2)$$

The foreground, namely the design elements, are the connected regions of *the inverse* of the background mask. By discarding some small connected and incomplete regions, foreground design elements along each connected region are cropped out from the input image one by one, according to the segment mask. To avoid losing information, an area slightly larger than the minimum area defined on the connected region of inverse mask is cropped out (Figure 5-7(c)). Consequently, a set of design elements are obtained $E = \{e_1, \cdots, e_n\}$. Simultaneously, the background color is also obtained by counting the average RGB value of pixels on the background region.

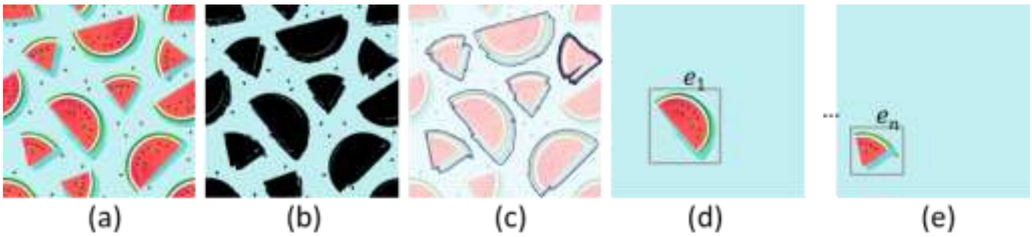


Figure 5-7 (a) The input design image; (b) the background mask; (c) illustration of foreground design elements extraction, red lines in (c) show the outline of connected regions, and the regions circled in blue are the actual cropping areas; (d-e) extracted design elements, grey boxes outline regions in (d) and (e) are examples of the extracted design elements

5.3.1.2 Core design elements selection

A design image consists of multiple repeated design elements, only the core distinctive design elements are selected. The core elements are selected among set E, based on the inter-element similarity assessment. The pHash algorithm (Monga *et al.*, 2006) is selected for the feature similarity assessment since it encodes the global content of a design element and is not sensitive to image size. pHash generates a feature vector of 64 elements that describe an image, computed from the Discrete Cosine Transform among the different frequency domains of the image. Therefore, similar design elements are filtered by comparing their pHash values.

Given the design elements $E = \{e_1, \cdots, e_n\}$, firstly, their corresponding pHash value is calculated $P = \{p_1, \cdots, p_n\}$. Then the design element with the largest area is selected, removed from E and added to the final design element list E' (Initially E' is empty). Then this design element is compared with all the elements by calculating the pHash values distances ($disP$) of this design element with the rest design elements in the set E. If the distance is closer than the threshold τ, those elements from E are removed. Next, the design element with the largest area is taken from the remaining in E , removed from E and added to E'. Once again, the pHash value distance with all the design elements in E is calculated and the design elements with a closer distance than the threshold is eliminated. This process is repeated until there are no more design elements left in E. Finally, the core design elements set E' is obtained. The pseudo-code is shown in Algorithm 5-1.

Algorithm 5-1 The pseudo-code of the algorithm for core design element selection

```
Input: The design elements: $E = \{e_1, \cdots, e_n\}$,
       The corresponding pHash value: $P = \{p_1, \cdots, p_n\}$,
       The pHash value distance threshold: $\tau$
begin
    $E' \leftarrow \{\}$
    while $E \neq empty$ do
        $m \leftarrow argmax\ (area(E))$
        $E' \leftarrow E' \cup e_m; E \leftarrow E - e_m$
        for $e_i$ in $E$ do
            if $disP(p_m, p_i)\ \leq \tau$ then
                $E \leftarrow E - e_i; P \leftarrow P - p_i$
            end
        end
    end
    return $E'$
end
```

5.3.2 Core design element vectorization

With the method proposed in section 5.3.1, it is possible to get multiple raster design elements from existing design images. Since vectorization is important for reuse of design resources in design process, many methods have been proposed aiming to faithfully restore all the details in the original image. Nevertheless, existing vectorization methods are not satisfactory because they inevitably generate too many meaningless vector paths. This study is endeavor to produce vector-based design elements in compact structures, and this is done by combining multiple vector paths. Each vector path is a mathematical representation of an intermediate representation. In previous works, the intermediate representations of an image generated by traditional segmentation techniques were easily over-segmented. To achieve compact vectorization results, a novel design element deconstruction method is proposed.

5.3.2.1 Design element deconstruction

To deconstruct design element to a compact set of intermediate representations, the color information is utilized. Firstly, the colored design element is converted into grayscale, $\{e_i\}$ to $\{e_i'\}$, and then its color histogram H is calculated. The number of histogram bars is set to a and the number of bars whose values are larger than a threshold ϵ are used as k. Thereafter, k-means is leveraged to deconstruct the design element into k intermediate representations $\{r_i\}_{i=1}^{k}$, as well as predict their color values. An illustration of this method is shown in Figure 5-8.

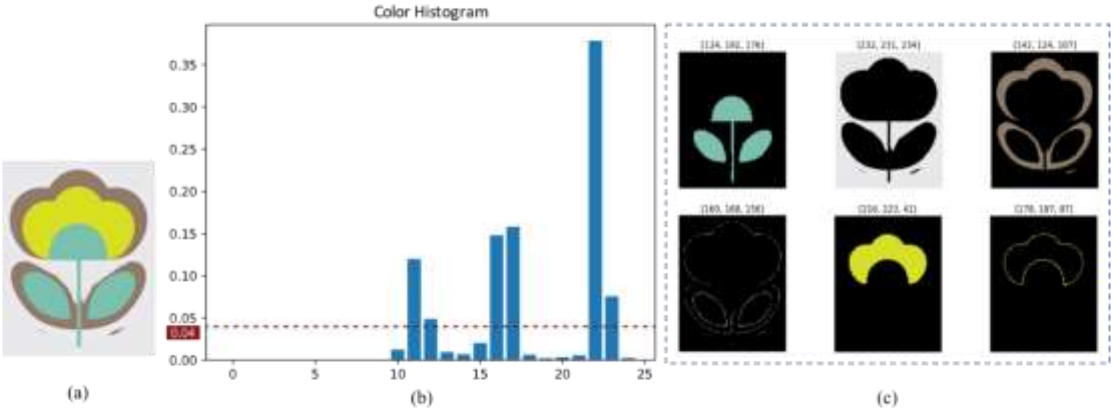


Figure 5-8 An example of color separation. (a) The input design element image; (b) the color histogram, and in this image *k* equals 6; (c) intermediate representations with their corresponding color value

5.3.2.2 Compact vectorization of design elements

Each color region in a color vector graphic involves geometric and color parameters. The geometric parameters of each intermediate representation is obtained by the Potrace algorithm (Selinger, 2003). After combining the color information of each path, multiple independent vector paths can be obtained (see Figure 5-9). A vector graphic is a multi-layer composition of many vector paths. When the path set is obtained, firstly all the paths are ordered by area to avoid occlusion between paths during visualization. Since the area of vector paths cannot be calculated directly, each is converted to a raster image for calculation. Meanwhile, the vector path whose area is smaller than 3% of the image is defined as a meaningless path and will be removed. Ultimately, the reserve paths are synthesized in order of area from small to large and a background layer is added at the bottom that is a rectangular filling with the background color. Examples are shown in Figure 5-9.

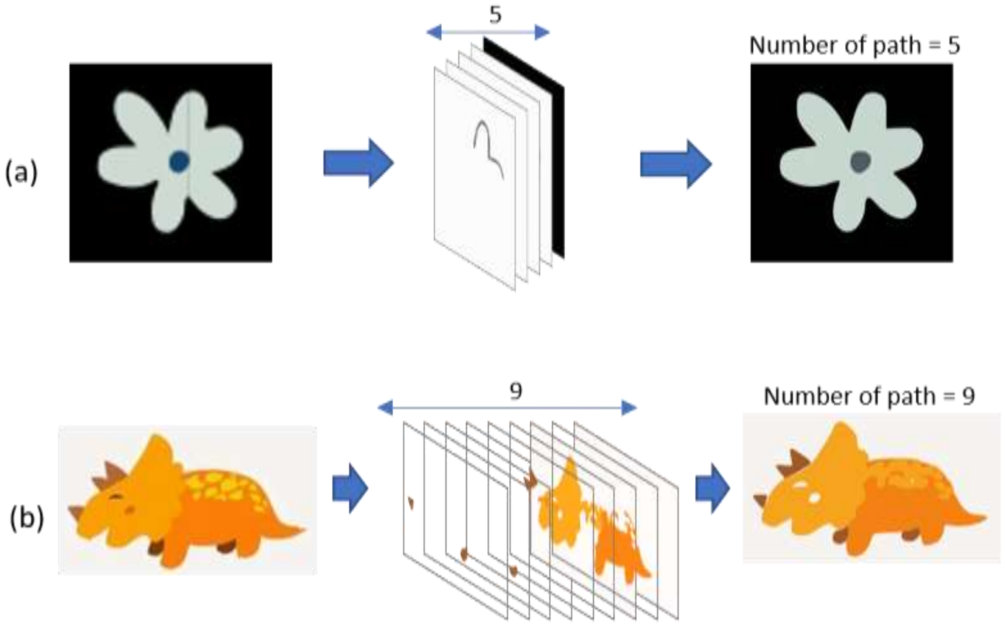


Figure 5-9 Examples of vector graphic synthesis. The vectorized result of design element (a) contains five vector paths, the black path is its background layer; and the vectorized design element (b) contains nine vector paths. Although our vectorized results lose some detail, our results remain the main content and contain a minimal number of vectorized paths

5.4 Experiment and Discussion

This section introduces the conducted experiments on design images to evaluate the effectiveness of the proposed E-system. All the experiments are conducted on the equipment with CPU Intel i7-6700k, Memory of 16GB, and GPU GeForce GTX TITAN X.

5.4.1 Experiments of core design elements extraction

5.4.1.1 Dataset and evaluate metric

To evaluate the effect of core design elements extraction from design images, the proposed method is tested on a dataset of 114 design images across a wide range of sizes, from 75 × 75 to 1024 × 1024 pixels (see Figure 5-10(a)). Each image has one or more design elements random layout on the image (examples are shown in Figure 5-10(b)).

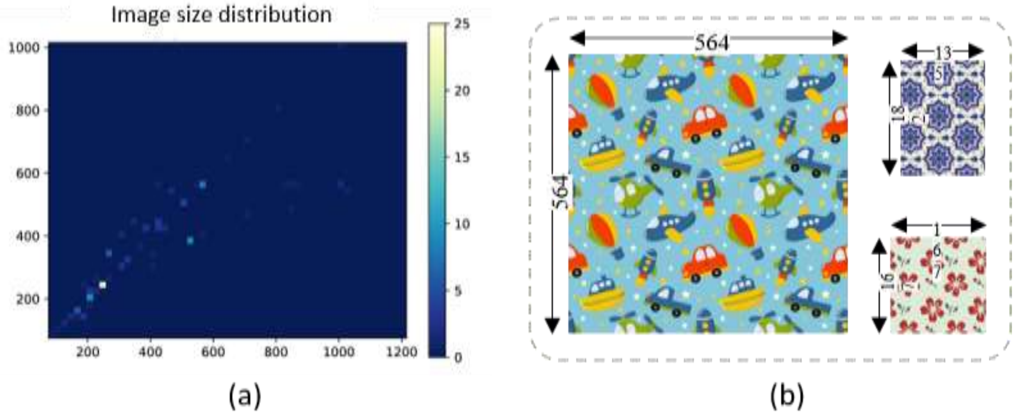


Figure 5-10 (a) Image size distribution of the test dataset; (b) examples of test images

It is worth emphasizing that the purpose of the proposed E-system is to extract the core design elements contained in the existing design images, not to extract all the design elements in the image. Therefore, images that do not extract at least one design element are defined as failure cases. The success detection rate r is calculated as follows:

$$r = \frac{|test| - |failure|}{|test|} \times 100\% \tag{5-3}$$

where $|test|$ represents the number of all the test images, $|failure|$ represents the number of failure cases.

5.4.1.2 *Parameter setting*

It is set that $p = q = 100$ for CNN architecture in foreground design elements extraction, and the detailed CNN configurations are outlined in Table 5-1. The algorithm can be equivalent to iteratively optimizing the CNN parameters and the segmentation results. The goal is to find a complete background mask efficiently. To balance the segmentation results and the time required, the best result from $M = 1,2,3,\dots,50$ is chosen for iteration on the whole test dataset. Finally, setting $M = 11$ is found for each image to achieve the best balance. Figure 5-11 reflects the relationship between the number of iterations and the success detection rate. In addition, in the case of GS, parameters are set to $\sigma = 0.5$, $k' = 32$, and the threshold of core design elements selection is set to $\tau = 0.85$.

Table 5-1 CNN configurations. The rows in turn represent the convolution components

	Kernel size	*Dim*	*Stride*	*Padding*	*Activation*
1st	3×3	100	1	1	ReLU & BatchNorm
2nd	3×3	100	1	1	ReLU & BatchNorm
classifier	1×1	100	1	1	BatchNorm

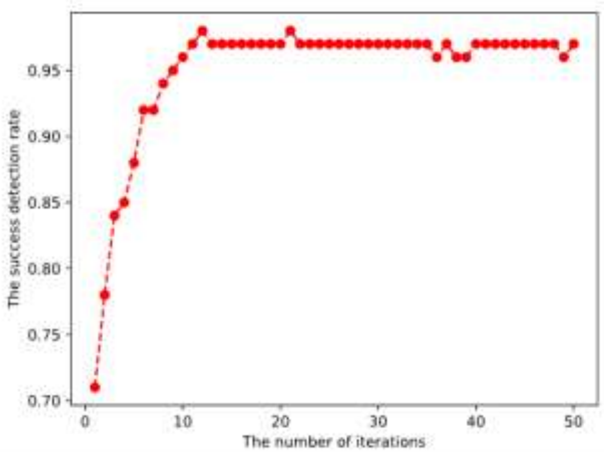


Figure 5-11 The curve of the success detection rate and the number of iterations

5.4.1.3 Performance

The proposed E-system successfully extracted 684 design elements from 109 images, and the total time cost is 1107 seconds, with an average time of 10 seconds per image. Figure 5-12 shows parts of the result of the proposed method to prove that our proposed method is able to extract core design elements from an unknown image.

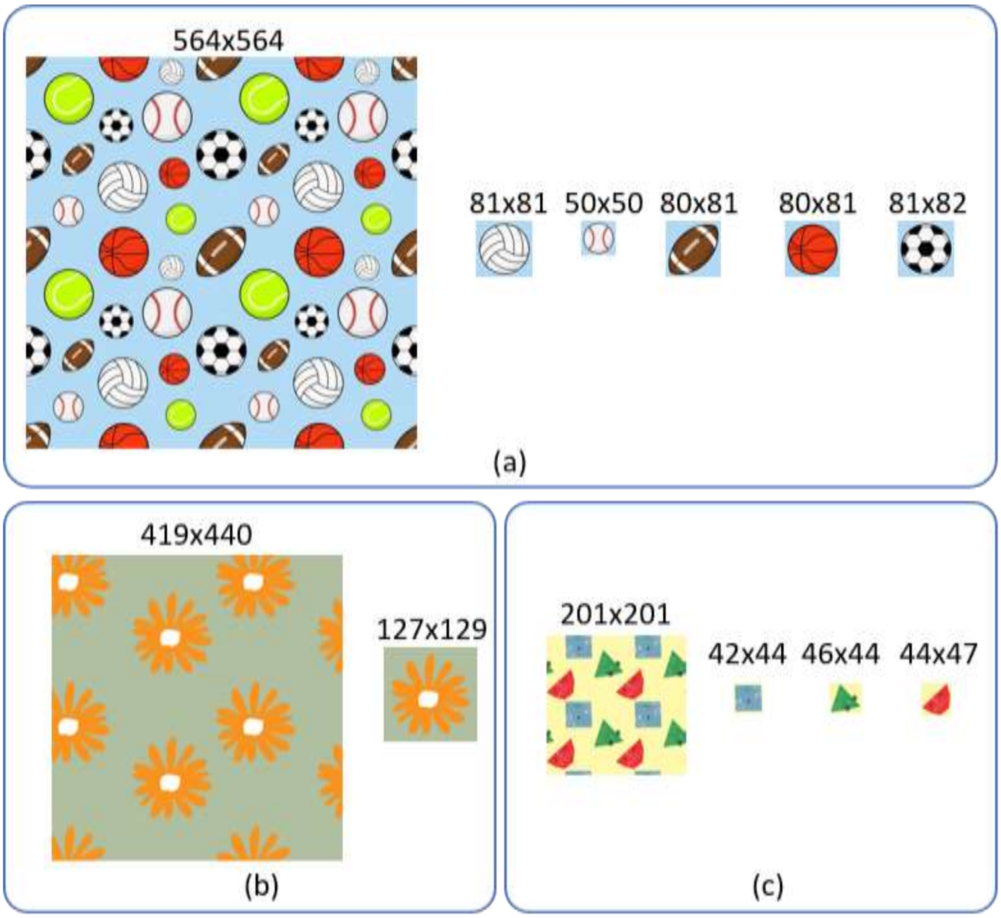


Figure 5-12 Examples of the output of the core design elements extraction, with the image size information

To further demonstrate the effectiveness of each module of the E-system, a set of comparative experiments was conducted. First, mean-shift clustering (Comaniciu *et al.*, 2002), another unsupervised segmentation method, was chosen to compare with the proposed CNN-based unsupervised segmentation method. Secondly, with reference to the work of Kanezaki (2018), the GS method was replaced with the superpixel method (Achanta *et al.*, 2012) as a comparison. For the Superpixel method, the number of Superpixel/segments *num_s* was set as 10000, 1000, and 100, respectively.

Table 5-2 Comparison of the computation time and success detection rate for design elements extraction among related works

	Mean-Shift	The proposed E-system	Method w/ Superpixel		
			num_s: 10000	num_s: 1000	num_s: 100
r	85%	96%	90%	87%	73%
time cost/s	62	10	17	10	8

Computer config: CPU Intel i7-6700k, GPU GeForce GTX TITAN X, Memory 16GB

Table 5-2 shows comparative results of the computation time and success detection rate for design elements extraction. From the table, the proposed E-system outperforms all other methods with the highest success detection rate of 96%; the element extraction time is 10s on average. The success detection rate of the methods with Superpixel is below 90%, which means some images that can be processed by the present method but failed when using Superpixel, demonstrating the advantage of the E-system. Specifically, setting a fixed number of segments for all test images of different sizes and uncertain content would result in some images being under-segmented and thus not available for extracting design elements. Accordingly, it is found that the number of segments influences the success detection rate of Superpixel method. The larger the number of segments, the higher the success detection rate. However, it takes a longer time for the network to converge and iterate the larger number of segments, therefore the time cost also increases accordingly. Furthermore, the success detection rate of mean-shift is about 10% less than the proposed one, while the time cost is about six times of the proposed one. It also proves the advantage of the proposed method on design image deconstruction.

5.4.2 Experiments of design elements vectorization

For design element deconstruction, the number of color histogram bars a is set to 25, the threshold $\epsilon = 0.04$ for the whole experiment. Extensive experiments were conducted on the output of the previous core design elements extraction. It took around 1924 seconds to vectorize 684 design elements, with an average time cost of each design element less than 3 seconds which is impossible to do manually with such efficiency.

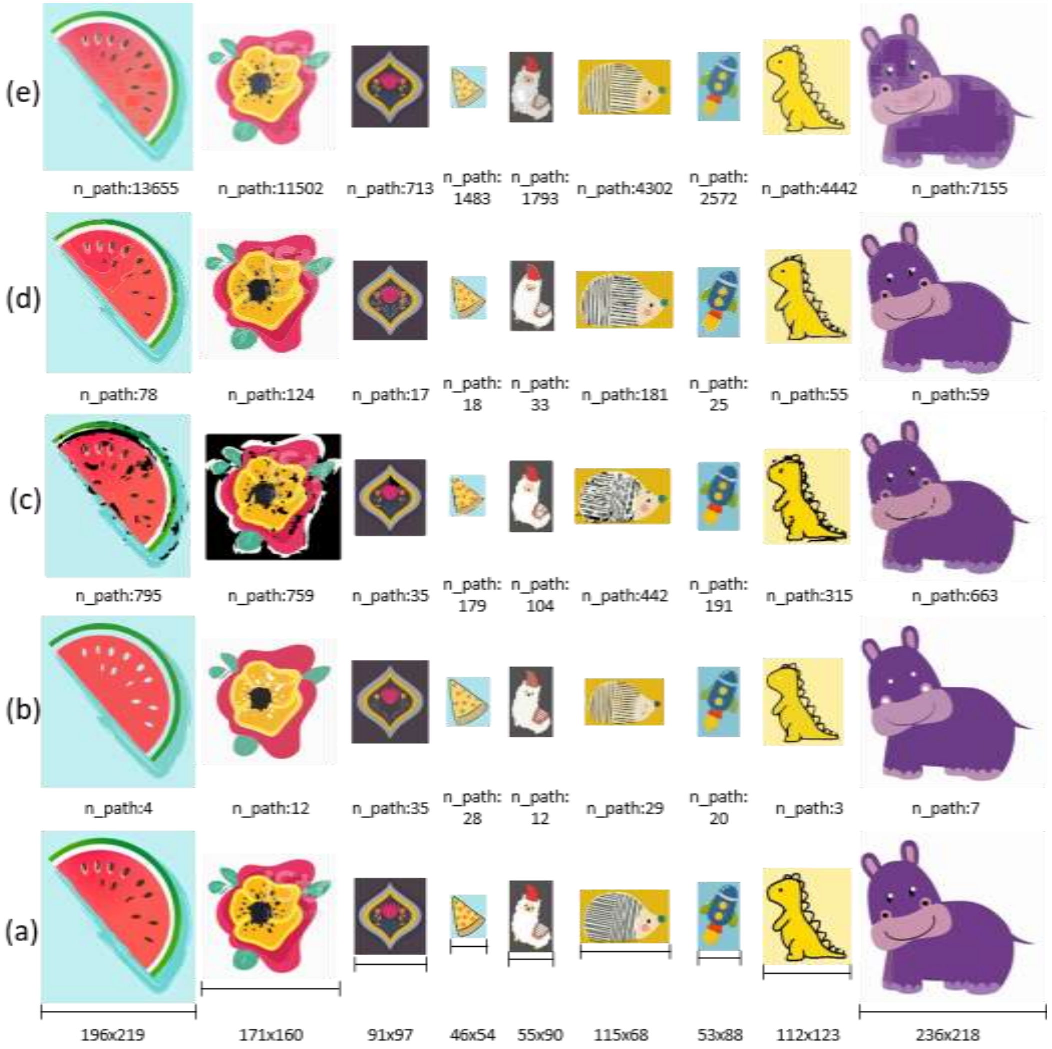


Figure 5-13 Comparison of our method across inputs of various sizes, (a) input raster design elements with their size information, (b) results of the proposed E-system, (c) results of Adobe (2022), (d) results of VectorMagic (2021), (e) results of Kerautret et al. (2019)

To demonstrate the effectiveness of the proposed approach in terms of compactness, comparative experiments were conducted. Figure 5-13 shows some results of design element vectorization with the number of vector paths from related works. The results were compared with Kerautret *et al.* (2019), who proposed a *combinatorial variational model* that merges geometry theory into a classical total variation model. As seen in

Figure 5-13(e), although the fidelity of their results was high, the files had too many vectorized paths to be simply editable. In addition, the proposed work also compared with Adobe (2022) and VectorMagic (2021), Figure 5-13(c) and (d) show the respective results. VectorMagic is a commercial software specialized for image vectorization, and Adobe Illustrator is a more generic vector-based design software. It can be seen that the results of VectorMagic (Figure 5-13(d)) are better than those of Adobe (Figure 5-13(c)) in terms of fidelity. But their results were less concise compared to the proposed method. The results obtained by the present method (Figure 5-13(b)) contain a minimum of vectorized paths while retaining the main content of each design element. Although their results lose some details, they can be easily edited to generate new graphs, which will be discussed in Section 5.4.4.

5.4.3 Limitation and discussion

Since the goal of the E-system is to extract and vectorize core and distinctive design elements from raster graphic design images, the images without extracted design elements are defined as failure cases. Across all the images, the proposed method failed in only five image inputs, and these images are classified into two categories. Examples of the first category are shown in Figure 5-14(a) and (b). The background mask of each image is separated into several non-adjacent parts. Therefore, the proposed method fails to find the corresponding background mask and locate the foreground design elements. In particular, in Figure 5-14(a), when white flowers and dark green leaves are considered as the foreground design elements, it is found that its background mask (in light green) is separated into several parts. It cannot be identified and used to locate the foreground design elements. The second category is shown in Figure 5-14(c). Although the design information is successfully segmented, the design elements are next to each other, and the proposed method cannot identify the background mask and the corresponding design elements. Nevertheless, it can be seen from Figure 5-14 that the proposed method can separate the inside design element without clear semantic information. In the future, it will research on directly transforming segmentation masks into vector regions without generating intermediate representations.

Furthermore, since the proposed method seeks to generate the most compact vectorized representations for design elements, paths with very small area would be defined as meaningless and be removed (described in Section 5.3.2.2). It causes the outputs to lose some image detail, such as the 'eye' of the yellow dragon in the 8th column of Figure 5-13. It is also one of the limitations of current method, and this will be improved in future work.

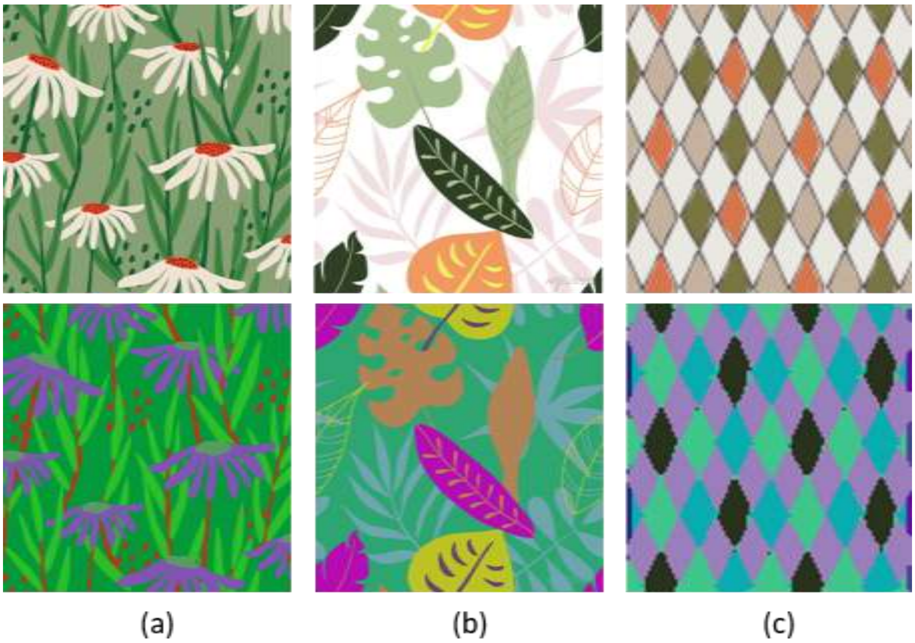


Figure 5-14 Failure cases. The top row is the input images, and the bottom row is the obtained segmentation masks

5.4.4 Application of the generated graphic images

In order to simulate the design element creation process, the obtained vectorized design elements were processed by replacing the fill colors with random colors and slightly changing the elements' shape. Repeat is one of the layout rules in graphic design (Ambrose *et al.*, 2019). Design elements can be tiled into larger patterns that can be applied in different scenes. Therefore, new designs are generated by repeating the generated design elements. The generated graphic design images are applied to some 3D models, see example of a silk scarf and a cup in Figure 5-15. Design generation, which could take hours to do manually, can now be performed on computers automatically within seconds. The experimental results illustrate that if combined with standard design rules, it is possible to generate new designs that conform to the human aesthetics using the proposed method and thus prove that the proposed method has the potential to assist design generation.

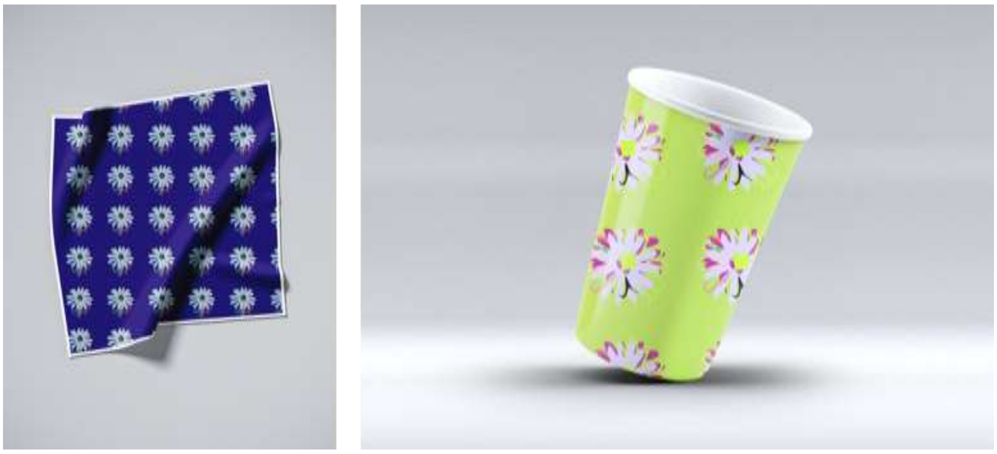


Figure 5-15 Computer-generated graphic design images used in a silk scarf and a cup. The 3D objects were processed in Adobe Photoshop

5.5 Chapter Summary

In this chapter, a novel method to extract and vectorize core design elements from an unknown raster input design image is proposed. To extract its design elements, unsupervised segmentation is leveraged to separate the foreground design elements with the background. Then, core design elements are selected by comparing the similarity among foreground design elements based on pHash values. After obtaining the core design elements, they are deconstructed into several colored intermediate representations. Subsequently, intermediate representations are vectorized to vector paths. After removing the vector paths with very small areas, the remaining vector paths are integrated into a new compact vector path by ascending order of area from small to large. The experimental results have demonstrated that the proposed system can efficiently extract core design elements and vectorize them into compact vector curves. Moreover, to demonstrate that the proposed system can assist design generation for professionals and non-professional users, the obtained vector-based design elements are used to generate new graphic designs and applied to 3D models.

A vector element dataset with 684 design elements is developed using the proposed E-system. The application proves that the proposed system is suitable for design reuse and re-creation. The next chapter will introduce how to generate human aesthetics aligned digital patterns, leveraging the vector-based design element dataset and standard design rules.

CHAPTER 6. Vector-based Digital Pattern Generation

6.1 Background Introduction

As described in the Introduction of CHAPTER 1, the digital pattern is an essential part of fashion design, and the pressure for its development is increasing both in terms of speed and quantity, so designers are eager for intelligent systems that may assist them to design rapidly and effectively (Wang *et al.*, 2019). Digital patterns can be understood as a combination of multiple colorful design elements (*fonts* and *icons* can be treated as particular design elements) on a two-dimensional plane. Because design process is typically iterative, the output designs must allow sequential design modifications/improvements to fulfill the specific design requirements in terms of pattern types and color preferences during the design process until confirmation (Briggs-Goode *et al.*, 2011; Studd, 2002). For instance, in the fashion industry, digital patterns should always be scaled and tiled to fit the requirements of the machinery and recolored to develop numerous *colorways* for a category of products and modified to new patterns. Therefore, digital patterns recorded in vector formats are popular in the industry since they are simple to modify or scale to any size without losing quality or details (more information on vector images was presented in CHAPTER 5). In addition, because new designs are usually based on a secondary development of existing ones, by editing or adjusting semi-finished patterns that conforms to human aesthetics. Therefore, a digital pattern design support system in line with the industry expectations should fulfill the below two requirements: (1) the outputs conform to basic human aesthetics, (2) the outputs are vector-based. However, to the best of our knowledge, no such system currently exists. The mainstream works on design generation techniques are at pixel level, and a few focus on content-specific vector-based pattern generation, such as marble patterns (details are described in CHAPTER 2). To properly assist digital pattern design generation and fill the research gap, this study proposes a G-system aiming to generate vector-based digital patterns that conform to the basic human aesthetics.

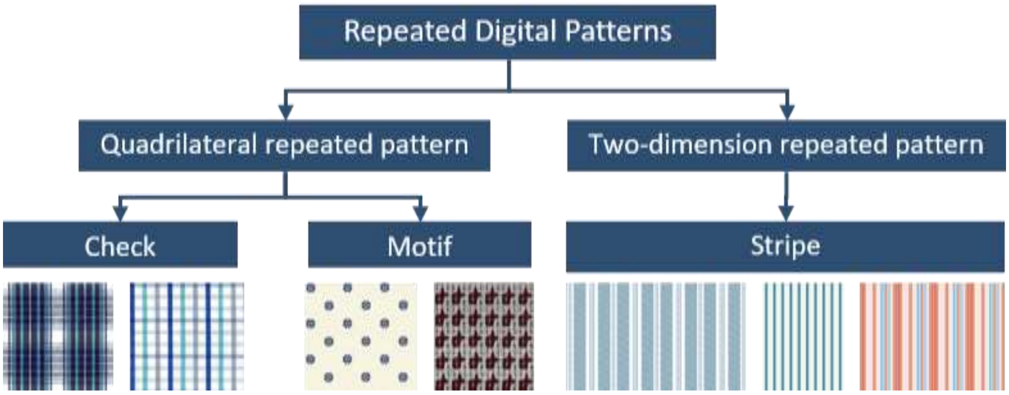
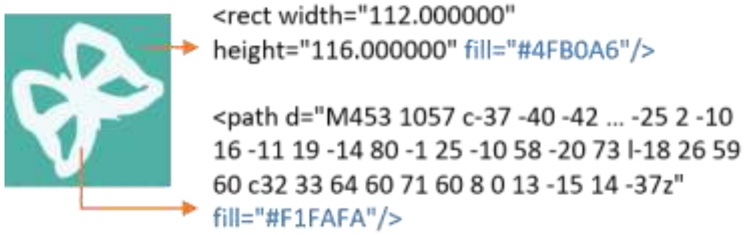

Repeated Digital Patterns
Quadrilateral repeated pattern
Two-dimension repeated pattern
Check
Motif
Stripe

Figure 6-1 Classification of repeated decorative digital patterns with illustrations

Digital patterns are always made up of multiple repeated patterns and come in a wide range of shapes and styles, such as geometric patterns (Lu *et al.*, 2014); traditional patterns (Yin *et al.*, 2020); pixel-art patterns (Kopf *et al.*, 2011); fractal graphics (Field *et al.*, 2009; Wang *et al.*, 2019) and so on. To cover common pattern classes, in this study, the decorative-use repeated digital patterns are classified into *two-dimension* (also known as *stripe*), and *quadrilateral* repeated patterns by layout, and then the quadrilateral repeated patterns can be further roughly divided into the *motif* and *check* class according to the pattern content, as illustrated in Figure 6-1. It is known that digital patterns are combinations of various colorful design elements that follow specific layout rules. On the other hand, vector-based digital patterns compose of several vector paths representing closed color regions. Each vector path contains geometry and color parameters, as shown in Figure 6-2. The geometric parameters represent a series of numbers describing the location and shape of a region enclosed by a vector path, and the color parameter represents the filled color of the region. In order to generate patterns meeting human aesthetic requirements and covering the most of common patterns, the proposed G-system will work on generating three main classes of patterns in vector formats, including stripe, check, and motif, from both vector-based geometric design generation and design colorization perspectives.



```
<rect width="112.000000"
height="116.000000" fill="#4FB0A6"/>

<path d="M453 1057 c-37 -40 -42 ... -25 2 -10
16 -11 19 -14 80 -1 25 -10 58 -20 73 l-18 26 59
60 c32 33 64 60 71 60 8 0 13 -15 14 -37z"
fill="#F1FAFA"/>
```

Figure 6-2 Illustration of a vector-based design element. The design element contains two vector paths, each path has geometric and color parameters

The G-system proposes three parametric models for stripe, check, and motif pattern generation. Each parametric model first generates vector-based geometric design information without considering the color; subsequently, it will colorize the generated patterns according to the reference image. Specifically, stripe and check patterns are essentially combinations of colorful slender rectangles (denoted as bands in this chapter). A stripe pattern is made up of continuous, uninterrupted color bands that run in one direction, while a check pattern consists of two orthogonal stripe patterns. As a result, the system splits a square into multiple parallel bands and convert these bands into vector paths to generate geometric information for stripe patterns. Likewise, the system orthogonalizes two stripe patterns to create a vector-based check pattern. Subsequently, these patterns will be repeated into larger ones to meet size requirements. The motif patterns contain several distinctive design elements, and the arrangement of design elements is varied. Design elements can be extracted and created using the method described in CHAPTER 5. Using the proposed R-system and E-system in CHAPTER 4 and CHAPTER 5, useful information can be analyzed from existing design images, from which a vector-based design element dataset was developed. In this chapter, a G-system is proposed to generate various geometric designs.

Color plays an important role in digital pattern design and is one of the most important visual cues in human perception (Sartori *et al.*, 2015; Shan, 2018). A good color combination helps improve the attractiveness of the created designs. Existing research has put efforts into exploring related topics such as color harmony and complex color combination rules. It is believed that extracting color combination rules from approved images is a easy and reasonable way to create design pleasing to users. Hence, once the geometric information of a pattern is obtained, the G-system will colorize it with colors according to the color palettes extracted from a reference image shared by user. Moreover, Pantone standard color codes are widely applied in the fashion industry to facilitate color communication, and designers often need to match desired colors with Pantone codes for production purpose (Kuo *et al.*, 2008; Zhou *et al.*, 2019). To better support design, the proposed G-system will simultaneously provide the recommended Pantone codes of the generated digital pattern. The implementation results and discussion of the proposed system demonstrate that the G-system can be applied in diverse applications and is able to support both professional designers and non-professional users in creating aesthetic pleasing digital patterns.

The rest of the chapter is structured as follows. Section 6.2 introduces related works on design layout generation and color palette extraction; section 6.3 explains the three parametric models separately; and section 6.4 presents the implementation results and discussion. Finally, conclusion is given in section 6.6.

6.2 Relate Work

The proposed system generates digital patterns taking reference to layout design and color combination. Therefore, this section presents the related work for generating various designs based on layout rules and extraction of colour information from input images.

6.2.1 Design layout generation

Researchers have worked on automating design generation by applying various layout rules, mostly in graphic design (Guo *et al.*, 2021; O'Donovan *et al.*, 2015). However, there is little work focusing on automatic digital pattern layouts. Majority of the existing studies of digital patterns pay attention to using the theoretical foundation of symmetry groups which is a mature mathematical theory for analyzing periodic patterns (Liu *et al.*, 2004). The symmetry group parameterizes the operation of rotation, reflection, translation, and scaling of the design elements and arranges the design elements' to different positions on the pattern. One of the early work was a system developed by Alexander in 1975 generating 17 symmetry patterns on 2D flat surfaces (Albert *et al.*, 2004). Valor *et al.* (2003) collated the 2D symmetry groups' rules in detail and generated some textile patterns (see Figure 6-3). Due to the position of design elements in the symmetry groups being close and the generated pattern style tend to being unitary, it is more suitable for using simple polygons to generate new patterns. Instead, the inputs of the proposed G-system are design elements with arbitrary styles and sizes.

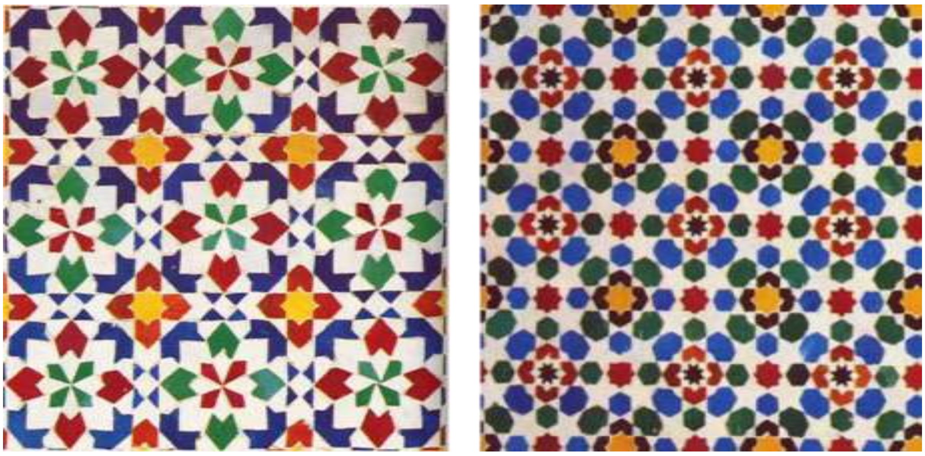


Figure 6-3 Illustrating tile design images using 2D symmetry groups by Valor *et al.* (2003)

6.2.2 Color palette extraction

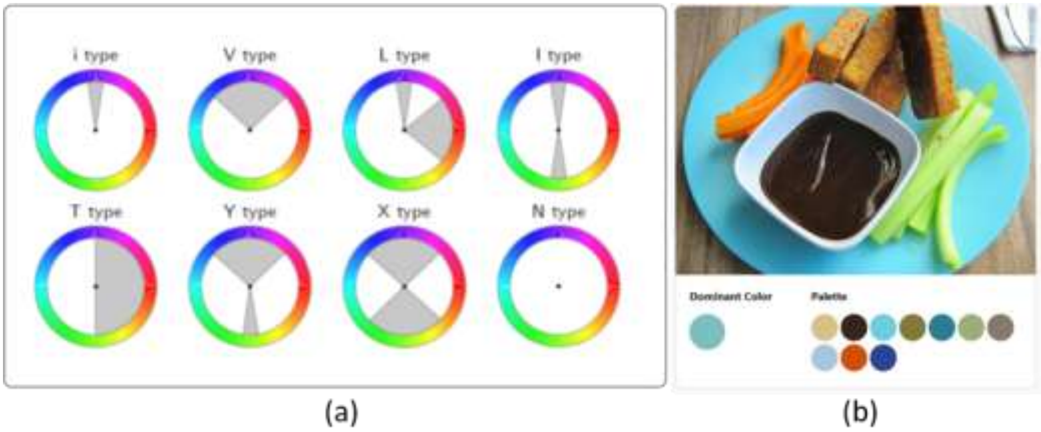


Figure 6-4 (a) Color harmony schemes on the hue wheel. A collection of colors that fall into the gray areas is considered to be harmonic (Cohen-Or *et al.*, 2006); (b) an example of an output color palette from ColorThief (Dhakar, 2020)

Color conveys emotional or aesthetic information to impact people's psychological activities (Bylinskii *et al.*, 2017; Hurley *et al.*, 2017) and influences fashion product sales (Jing *et al.*, 2022; Palmer *et al.*, 2010). For the sake of design analysis, a number of methods are devoted to exploring how colors work together, especially color harmony. Color harmonies can be defined as two or more colors brought together to produce a satisfying affective response (Cohen-Or *et al.*, 2006). Itten *et al.* (1961) introduced a kind of color wheel in which they described the color harmony theory based on the relative positions of the hues on the color wheel. On the ground of their theories, Matsuda (1995) and Cohen-Or *et al.* (2006) proposed more complex color schemes, defined by combining several types of hue and tone distributions (see Figure 6-4(a)). These schemes were used in a wide range of works, such as webpage design (Nazar *et al.*, 2017) and packaging design (Hurley *et al.*, 2017). The design of digital patterns covers a wide range of content, and more complex color schemes are needed. However, a new color scheme can be costly to develop and hard to guarantee its effectiveness. Therefore, some researchers turn to research on extracting corresponding *color palettes* from existing images or approved designs, using which to guide colorizing new designs (Cao *et al.*, 2017; Jalal *et al.*, 2015).

The color palette, like a painter's palette, defines the representative colors within an image. In essence, it is the statistical and quantitative processing of pixel values in the image. Some studies used the color histogram to extract color palettes defined in various color spaces, with HSV and L*a*b* being the most widely used ones (Delon *et al.*, 2005; Meskaldji *et al.*, 2009). Moreover, some works leverage the median cut algorithm to extract the corresponding color palette (Chen, 2008). In addition, other researchers treat color palette extraction as a disguised clustering problem (Gijsenij *et al.*, 2022; Lai *et al.*, 2020; Lertrusdachakul *et al.*, 2019), in which K-means is the most commonly used clustering algorithm (Jing *et al.*, 2022). Each pixel is assigned a cluster based on which cluster centroid is closest to the pixel in a k-dimensional space. At the same time, k-means clustering aims to find the set of k centroids so that the difference between individual pixel values within a cluster is minimized. For example, a commonly used tool called ColorThief (Dhakar, 2020) based on a clustering algorithm extracts the domain colors from the input image (see Figure 6-4(b)). Although clustering methods need to specify the number of colors in advance, the output is promising.

6.3 Method

A vector-based fashion digital pattern generation system (G-system) is proposed in this study. The proposed system simulates industrial workflow in order to generate vector-based digital patterns that align with the basic human aesthetics. The proposed G-system consists of two main modules: 1) vector-based geometric design generation, including parameterized layout rules and parametric models for each class of patterns, and 2) design colorization, including color palette extraction and digital pattern colorization, as illustrated in Figure 6-5.

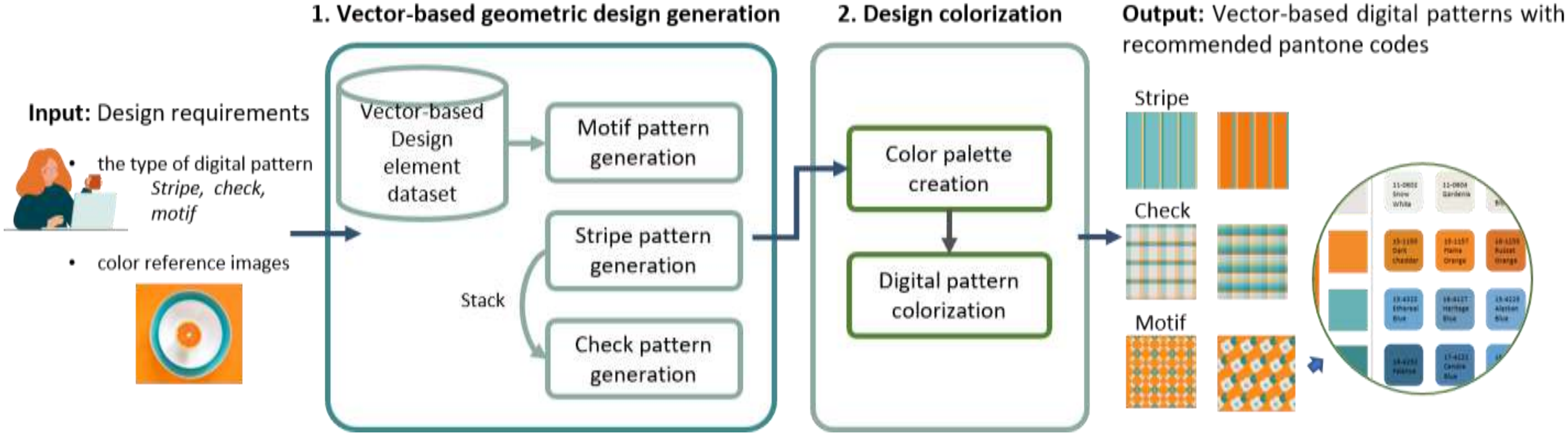


Figure 6-5 Illustration of the proposed vector-based digital pattern generation system (G-system)

6.3.1 Vector-based geometric design generation

The geometric information of the vector-based digital patterns includes the morphological and positional parameters of the design elements, where the positional parameters are calculated according to the given layout rules. In addition to generating design elements' information for stripe and check patterns, to ensure that the patterns conform to basic human aesthetics, the system parameterizes the classical repeat structures commonly used in textile design and proposes a novel layout rule for motif patterns based on the golden ratio.

6.3.1.1 Repeat structures

In the fashion industry, a digital pattern covering large continued area can be generated by repeating several small-sized patterns, also called repeated patterns (a more detailed analysis of repeated patterns was given in CHAPTER 4). Repeat structures are the most common layout rules. Basic repeat structures are summarized into four types: straight repeat, half drop, tile (or brick) repeat, and repeat mirrored vertically and horizontally (Phillips *et al.*, 1993; Wilson, 2001). An illustration of them is shown in Figure 6-6.

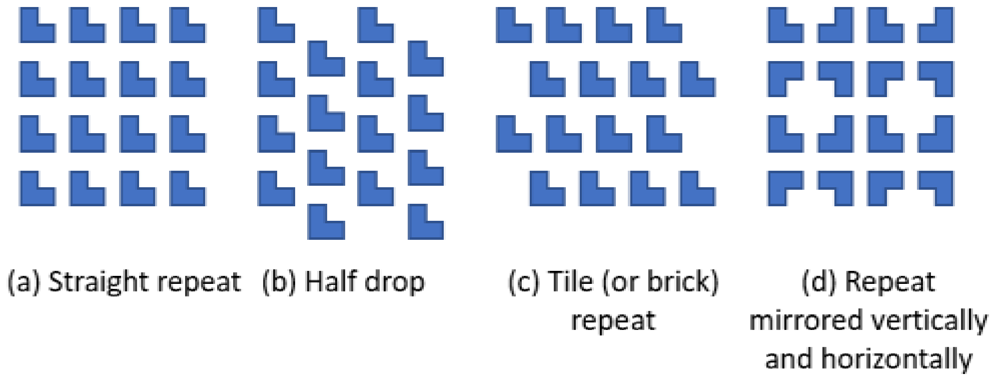

(a) Straight repeat
(b) Half drop
(c) Tile (or brick) repeat
(d) Repeat mirrored vertically and horizontally

Figure 6-6 Illustrations of repeat structures

In particular, the straight repeat structure is the pattern repeating directly above and below in straight lines (see Figure 6-6(a)). The half drop repeat structure is based on the straight repeat structure; the odd/even pattern verticals slide halfway down in the vertical direction (see Figure 6-6(b)). Similarly, the tile repeat structure (see Figure 6-6(c)) is the odd/even pattern columns slide halfway right in the horizontal direction. The pattern columns/verticals can also slide across other amounts. For repeat mirrored vertically and horizontally structure, the first repeated pattern is mirrored horizontally; after that, the repeated pattern and the mirrored one are then mirrored vertically to form a compound repeated pattern that is then repeated in straight lines (see Figure 6-6(d)).

6.3.1.2 Motif pattern generation

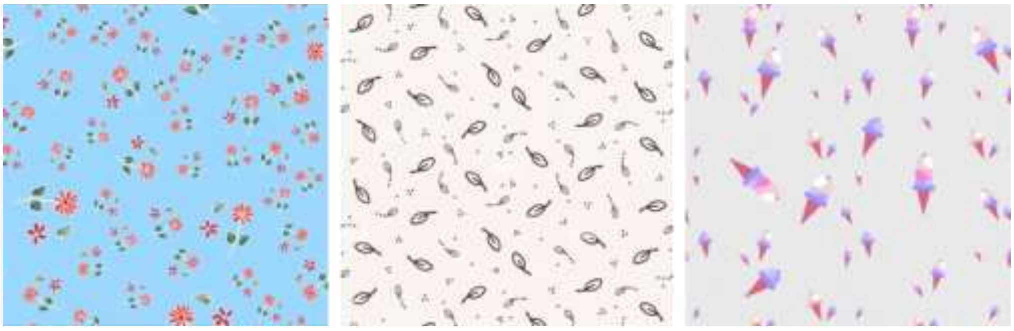


Figure 6-7 Examples of typical fashion digital patterns. The images are downloaded online

Motif pattern generation is the core part of the proposed system. A motif pattern contains one or several design elements, such as animal, human, leaf, flower, fruit, or geometric shapes (Tsetimtheo, 2020). Creating new design elements is crucial for motif pattern design, and this can be routine work for designers. In practice, most vector-based design elements are created by tracing existing raster design elements into vector formats and adjusting them manually (Yang *et al.*, 2015). For automatically generating motif patterns, the G-system uses design elements from a vector-based design element dataset built by the E-system (in CHAPTER 5).

Variable of motif pattern generation is the positional layout of design elements, which are subjectively arranged by designers based on their experience and design knowledge. To ensure that the generated patterns are aesthetically pleasing, the G-system adopts the classic repeat rules that are explained in section 6.3.1.1 to guide the positions of design elements. Moreover, it is found that arranging single elements according to some aesthetic rules is able to create patterns that appear complex (see Figure 6-7). Art and mathematics are found to have a profound connection (Wang *et al.*, 2019), in which the application of golden ratio is often identified in the layout of many artworks (Avramović *et al.*, 2013). The golden ratio is capable of creating aesthetically pleasing forms, and the golden ratio is defined as:

$$\vartheta = \frac{w}{h} = \frac{1+\sqrt{5}}{2} \approx 1.618 \tag{6-1}$$

where w and h represent the lengths of two line segments, respectively.

In this study, the golden ratio is used to create layout rules, *golden squares*, for design elements in order to enrich the generated motif patterns. As shown in Figure 6-8(a), the blue points illustrate the positions of design elements. The pink-golden square (zoomed in Figure 6-8(b)) is the smallest unit within the golden squares. The shortest line of the pink-golden square is defined as a, which is used to define the size of the design element, calculated as:

$$a = \frac{max(w_e, h_e)}{1.618^2} \tag{6-2}$$

where w_e and h_e are the width and height of the design element, respectively. The yellow golden square is 1.618 times of the pink golden square, and the green square is 1.618^2 times of the pink golden square, and so do the internal parts inside the square, a deconstruction illustration is shown in Figure 6-8(c). Consequently, a relationship is established between different sizes of the design element and their arrangement within the golden square.

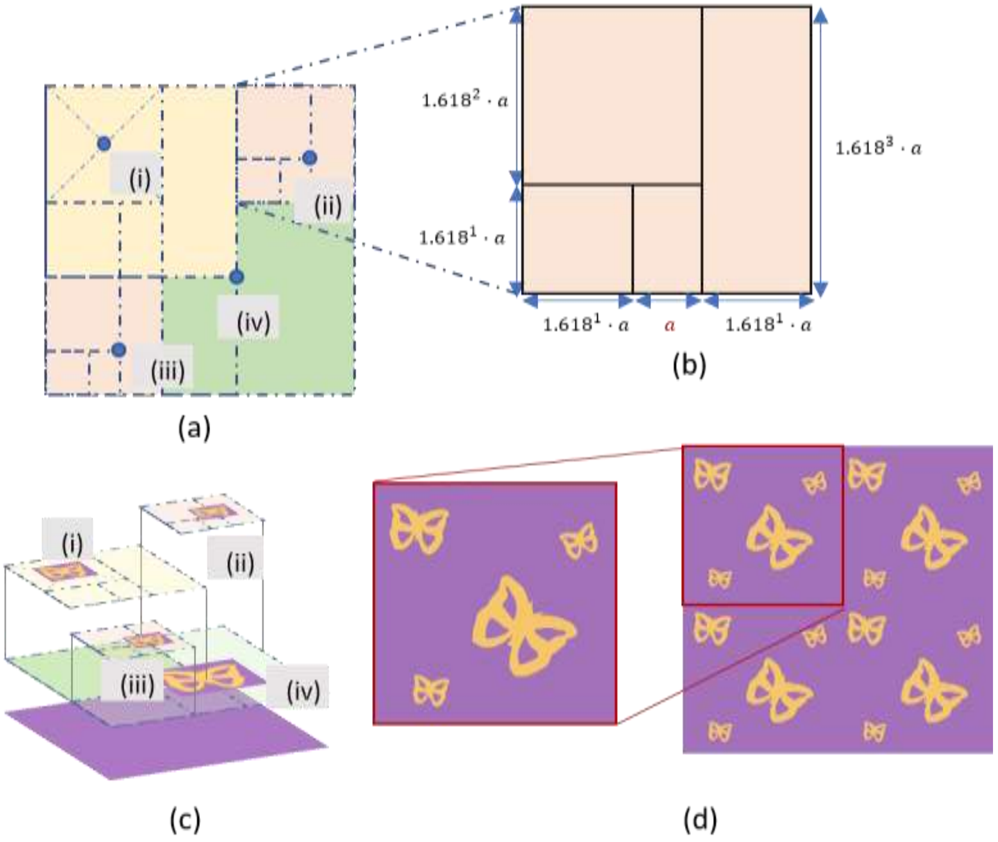


Figure 6-8 The proposed golden squares; (a) and (b) explain the proposed golden squares layout rule; (c) gives an example of design elements' positions based on the golden squares structure; (d) shows the generated motif pattern and its repeated pattern

The size and position of each design element are adjusted to ensure the generated repeated patterns to be in line with human aesthetics. As illustrated in Figure 6-8(c), the design element in position (i), whose width is w_e, height is h_e and the rotation angle is

0°; both design elements in positions (ii) and (iii) are $(\frac{w_e}{1.618} \times \frac{h_e}{1.618})$ in size, and the rotation angles are 330° and 0°, respectively; and the design element in (iv), whose size is $(1.618 \cdot w_e \times 1.618 \cdot h_e)$ and the rotation angle is 30°. The generated motif pattern is tiled following the straight repeat structure (Figure 6-8(a)) to generate a larger one. An example of the generated motif pattern is shown in Figure 6-8(d), where its repeated pattern is in zoomed-in insets. Figure 6-9 shows the motif patterns generated by the G-system using the same design element combined with different layout rules. More examples will be presented in section 6.4.

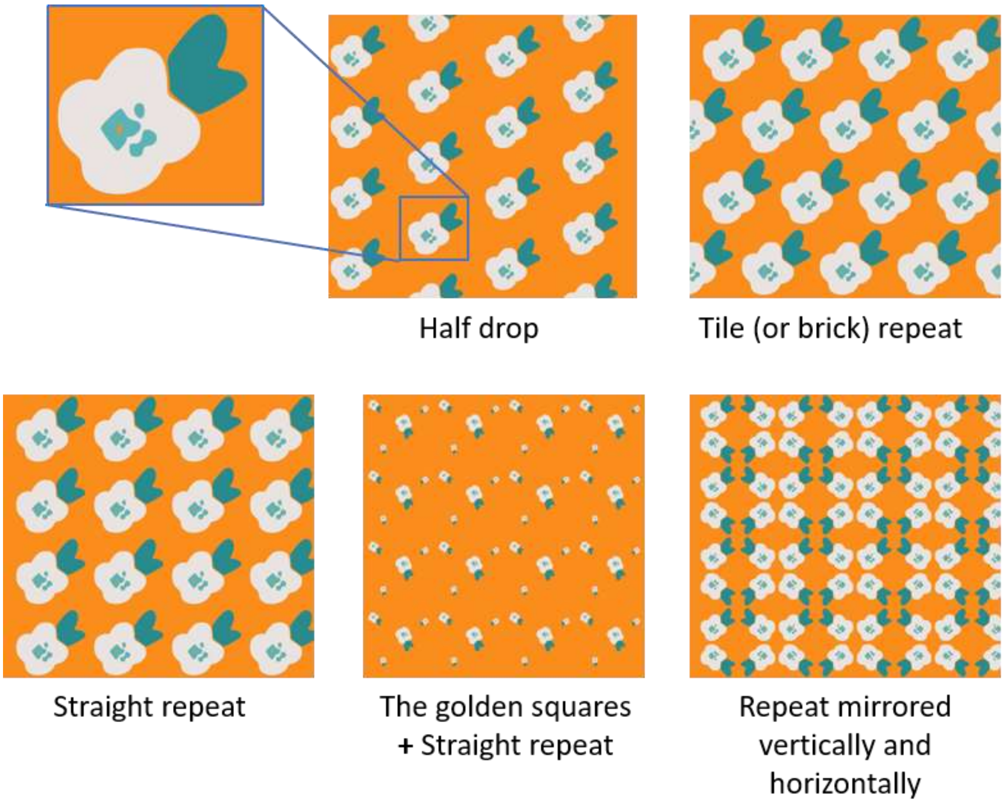

Half drop
Tile (or brick) repeat
Straight repeat
The golden squares + Straight repeat
Repeat mirrored vertically and horizontally

Figure 6-9 Examples of the generated motif patterns using the same design elements under different repeat structures

6.3.1.3 Stripe pattern generation

Stripe pattern is one of the oldest patterns used in textile materials or fabrics, which has been used since ancient times (Hampshire *et al.*, 2006). A stripe pattern is made up of continuous, uninterrupted lines or bands that run either horizontally, vertically, or diagonally (usually at 45°) (Wilson, 2001). Figure 6-10 shows an illustration of different stripe patterns. Hence, the width of the stripes, their distribution, scale, color, and color combinations, as well as how stripes are organized and arranged, can determine the form and appearance of striped patterns.

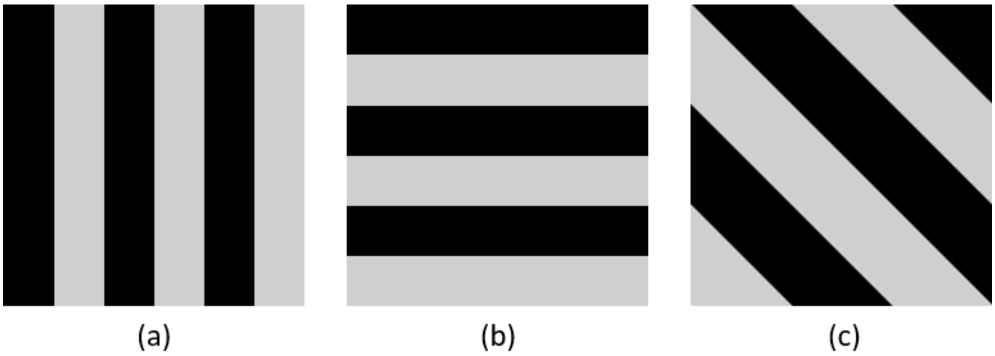


Figure 6-10 Illustration of different stripe patterns, (a) a horizontal stripe pattern, (b) a vertical stripe pattern, and (c) a diagonal stripe pattern at a 45°

Since the stripe pattern repeats only in one direction, only the length of the pattern in the repeat direction needs to be taken into account. Assume that the size of a *horizontally* repeated stripe pattern is $S \times S$, the number of bands is N_s and the width of each band is s_i, so that

$$S = \sum_{i=0}^{N_s} s_i \tag{6-3}$$

The band width s_i is generated by a uniform distributed function, $random(lb, ub)$ between a lower bound, lb, and an upper bound, ub, of the band width. Notably, the width of each band is not unlimited. The first band width can be calculated as follows:

$$s_0 = random\left(b, \frac{S}{N_s}\right) \tag{6-4}$$

where b is the minimum band width. The rest bands s_i, where $i \geq 1$, are obtained by:

$$s_i = random\left(b, S - \sum_{k=0}^{i-1} s_k\right) \tag{6-5}$$

The generated stripe pattern is able to tile into a larger one by the straight repeat layout rule (Figure 6-10(a)). Moreover, vertical and diagonal stripe patterns can be obtained by rotating a given horizontal stripe pattern, Γ, by an angle of 90° and 45°, respectively. The formulation is defined as follows:

$$\Gamma' = \Gamma \cdot \begin{bmatrix} cos(\alpha) & -sin(\alpha) & 0 \\ sin(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6-6}$$

where α represents the rotation angle, when $\alpha = 90$, the resulting pattern Γ' is a vertical stripe pattern; and when $\alpha = 45$, Γ' is a diagonal stripe pattern.

6.3.1.4 Check pattern generation

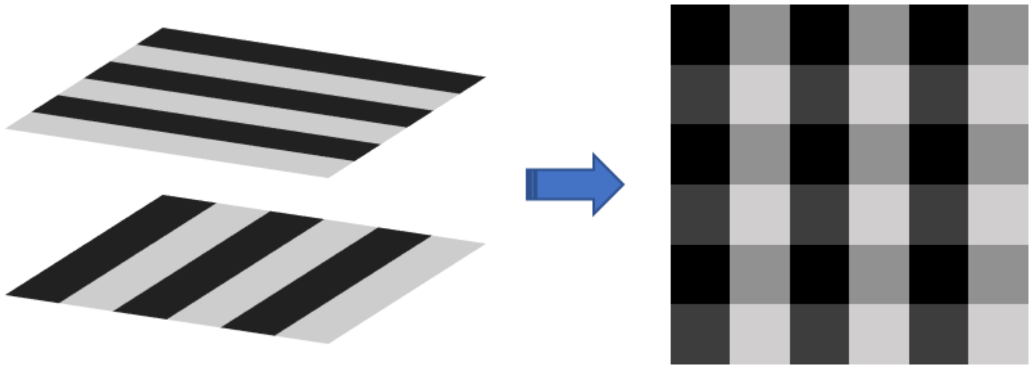


Figure 6-11 Illustration of check pattern generation

Another class of patterns is check pattern, which could be described as vertical and horizontal stripes crossing over one another (Hann, 2019). Therefore, a check pattern can be generated by stacking two orthogonal stripe patterns (see Figure 6-11). A horizontal stripe pattern Γ_x of size $S \times S$ can be obtained by the above parametric model, and the number of bands is N_x^c, which are often represented in different colors. Similarly, a vertical stripe pattern Γ_y of size $S \times S$ can be obtained with N_y^c colors. A check pattern C can be generated by:

$$C = \Gamma_x \oplus \Gamma_y \qquad (6\text{-}7)$$

where $\oplus$ represents the stack operation. To create a visual effect of a check pattern, the color opacity of the upper stripe pattern, Γ_x, has been adjusted to 50%. Similarly, the generated check pattern is able to tile into a larger one by the straight repeat layout rule (Figure 6-6(a)).

6.3.2 Design colorization

6.3.2.1 Color palette creation

As explained in section 6.1, each vector-based pattern involves geometric and color parameters. The method described in section 6.3.1 provides the geometric information of each vector-based digital pattern. Subsequently, the G-system colorizes the generated patterns according to a given reference image. Therefore, the system first needs to analyze the number of colors required for the pattern. For check and stripe patterns, the number of vector paths in their repeated patterns equals the number of colors required, i.e. the number of colors in a stripe pattern N_s, the number of colors in the horizontal direction of a check pattern N_c^w and the number of colors in vertical direction N_c^h. For motif patterns, the number of colors of the entire pattern is the same as the number of colors of the design element is defined as N_m.

To simulate the designer's operation for coloring patterns with color palettes, the system extracts the corresponding color palette from reference images to guide the colorizing operation. As discussed in section 6.2.2, k-means clustering is one of the common methods for color palette creation. The G-system leverages the k-means and sets the k to be the number of colors of the pattern. A color palette is extracted from a given reference image by k-means in L*a*b*color space. Specifically, the color palette with colors sorted by the color proportion (the ratio of the area of a color to the whole image) from large to small is recorded as a set $\Theta' = \{(c'_k, cr'_k) | k \in [1, N_m]\}$, where c'_k is the color value and cr'_k is its color proportion.

Moreover, the system collected a total of 1275 Pantone codes covering most colors used in the fashion industry to provide Pantone codes for the generated patterns. Cosine similarity is leveraged to measure the similarity between the desired color and the collected Pantone values. The similarity score sim is calculated using:

$$sim = \frac{\sum_{i=1}^{n} M_i \times P_i}{\sqrt{\sum_{i=1}^{n} {M_i}^2} \times \sqrt{\sum_{i=1}^{n} {N_i}^2}} \tag{6-8}$$

where M_i is the RGB value of the desired color and P_i is the RGB value of the Pantone code. The system will provide the Pantone codes with higher similarity values to the desired color for user reference. Examples of extracted color palettes are shown in Figure 6-12.

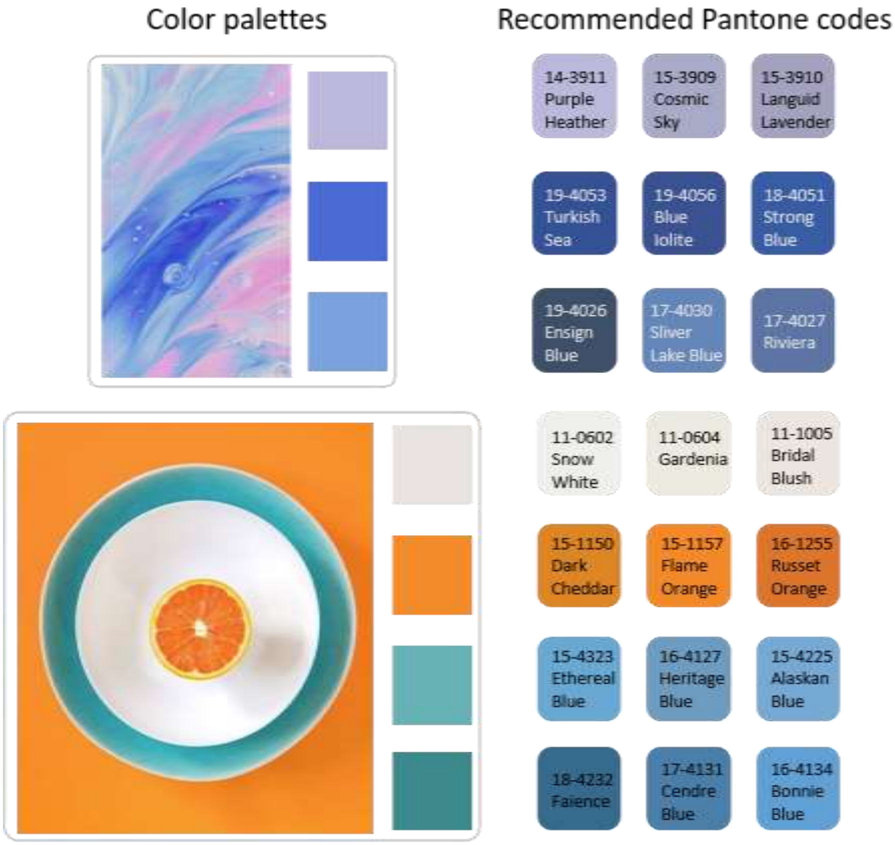


Figure 6-12 Examples of created color palettes with recommended Pantone codes

6.3.2.2 Digital pattern colorization

The G-system colorizes the generated patterns according to the extracted color palettes. For stripe patterns, the system fills each vector path with a color chosen from the palette, and it tends to use more color options defined in the palette. Considering a check pattern is composed of two stripe patterns, the system colorizes the two orthogonal stripe patterns based on the same color palette. This constraint can effectively limit the number of colors in a check pattern to match the corporate restrictions on color. Moreover, the system takes into account the color proportion of the applied design element when colorizing the motif pattern to ensure visual aesthetics. Specifically, the system first sorts the colors of the design element according to the color proportion from large to small and records them into a set $O = \{(c_j', cr_j')|j \in [1, N_m]\}$, where c_j' is the color value, cr_j' is its color proportion and N_m is the number of colors. This set O represents the color information of the entire pattern, too. By setting the number of colors in the motif pattern the same as the number of colors in the extracted color palette $\Theta' = \{(c_k', cr_k')|k \in [1, N_m]\}$, the system replaces the color of the target image by colors in the palette one by one, which means replace c_j' to c_k', to obtain a recolored motif pattern.

6.4 Implementation Results and Discussion

After inputting the color reference images and defining the required pattern category in the G-system, the outputs are vector-based digital patterns, including stripe, check, and motif patterns. Table 6-1 reports the overall performance of vector-based digital pattern generation. A motif pattern with an average size of $1714 \times 1713\ pix$ can be generated in around 0.7 seconds, and a stripe and a check pattern with $400 \times 400\ pix$ in size require about 8 and 6.5 seconds, respectively. Notably, the design generation was done on a standard PC with an Intel i7-6700k CPU with 16GB of memory without a GPU. It is believed that it will go faster with better hardware configuration.

Table 6-1 The average pattern size and computation time cost of the patterns generated by category

	Stripe	Check	Motif
Pattern size /pix	400×400	400×400	1714×1713
Time cost /s	8.0	6.5	0.7

Computer config: CPU Intel i7-6700k, Memory 16GB

For design colorization, 12 images were collected from the Internet as references. Meanwhile, recommended Pantone codes are provided for design support. The proposed system adopts cosine similarity to recommend Pantone code for each desired color. However, finding the exact matching Pantone code of the desired color is still a difficult problem for computers and is often operated by humans in practice. Therefore, the proposed method provides three recommended Pantone codes for recommendation. Two examples of created color palettes and their recommended Pantone codes are presented in Figure 6-12. Next, the generation effect of each type of pattern will be discussed separately.

Stripe pattern: Fashion digital pattern design rules are highly related to extensive knowledge of material properties and fabrication methods. Due to the limitations of production equipment and cost, prints and jacquards should not use too many colors (Wilson, 2001). Therefore, the number of stripe bands' colors is defined as $N_s \in [2,10]$ and the output size is $400 \times 400\, pix$ for the subsequent experiments. Figure 6-13 shows some generated stripe patterns that follow different color reference images as well as different color numbers (in rows (a) and (b), whose color reference images are on the left). At the same time, the algorithm used can generate different colorways for patterns with the same geometric information (see Figure 6-13(c)).

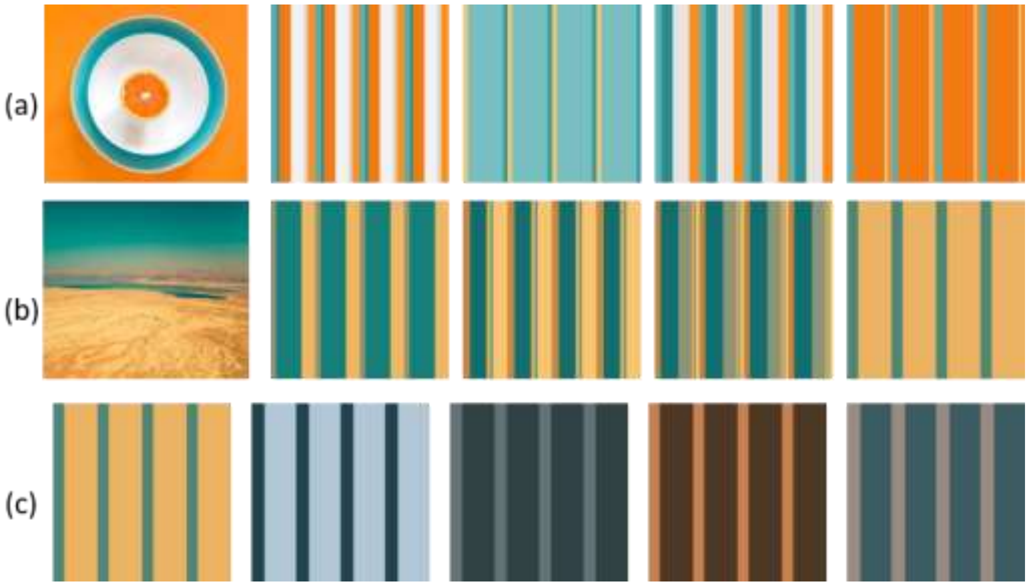


Figure 6-13 Stripe patterns generated by the proposed system. (a-b) Stripe patterns colorized following the same reference image, (c) stripe pattern in several colorways

Check pattern: In this study, the check pattern is considered to be composed of two orthogonal stripes. Figure 6-14 presents parts of the generated outcomes, all with a size of $400 \times 400\, pix$, the color palette of the two orthogonal stripes forming a check pattern is set to be derived from the same color reference image. In this way, uncoordinated color matching and even excessive color numbers can be avoided in the check pattern. Examples are displayed in Figure 6-14. Row (a) and (b) show check patterns following two different color reference images, respectively, and row (c) presents several colorways of a check pattern with the same geometric information.

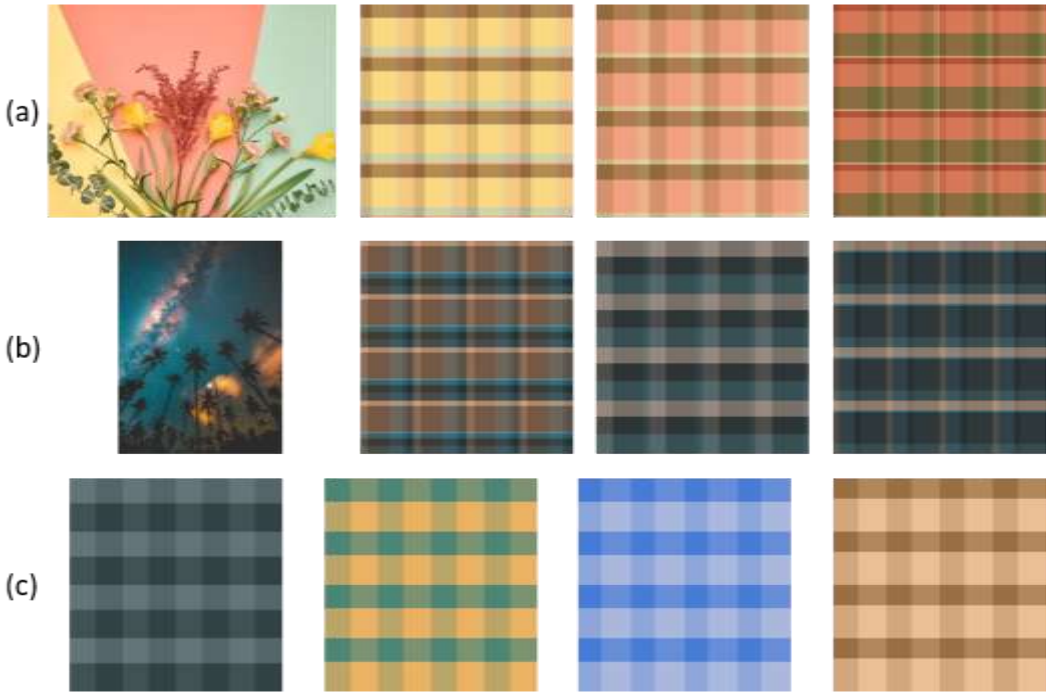


Figure 6-14 Check patterns generated by the proposed system. (a-b) Check patterns colorized following the same reference image, (c) check pattern in different colorways

Motif pattern: Through the previous chapters, a dataset with 684 vector-based design elements is built. New motif patterns are created by arranging the elements from the dataset following the parameterized layout rules, particularly straight repeat, repeat mirrored vertically and horizontally, a half-drop or tile repeat, and customized golden squares. Figure 6-15 presents some generated motif patterns by the proposed G-system.

Each column in (a) represents a set of motif patterns colorized following the same reference image, and (b) shows a generated motif pattern with several colorways.

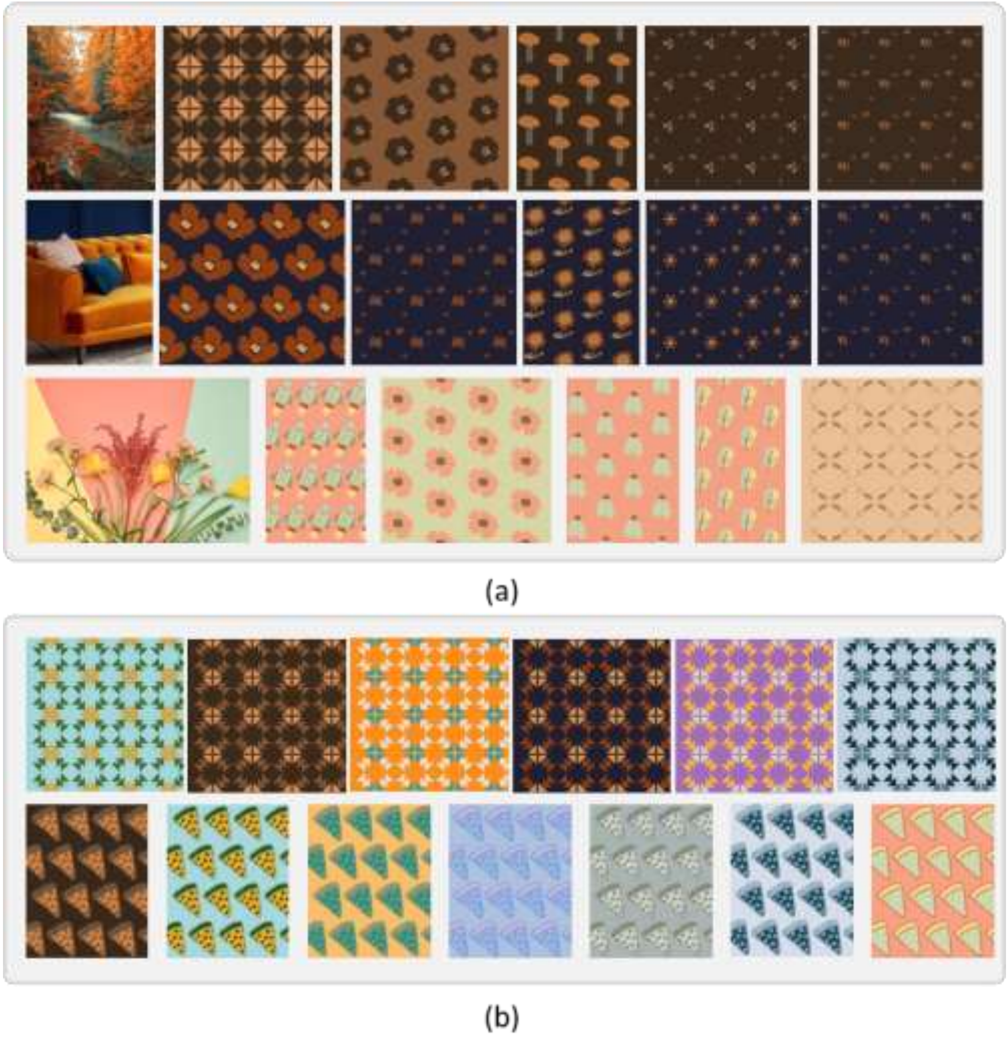


Figure 6-15 Motif patterns generated by the proposed system. (a) Motif patterns colorized following the same reference image, (b) generated motif patterns in several colorways

To further demonstrated that the system can assist fashion digital patterns design, the generated patterns are applied on some 3D models (see Figure 6-16).

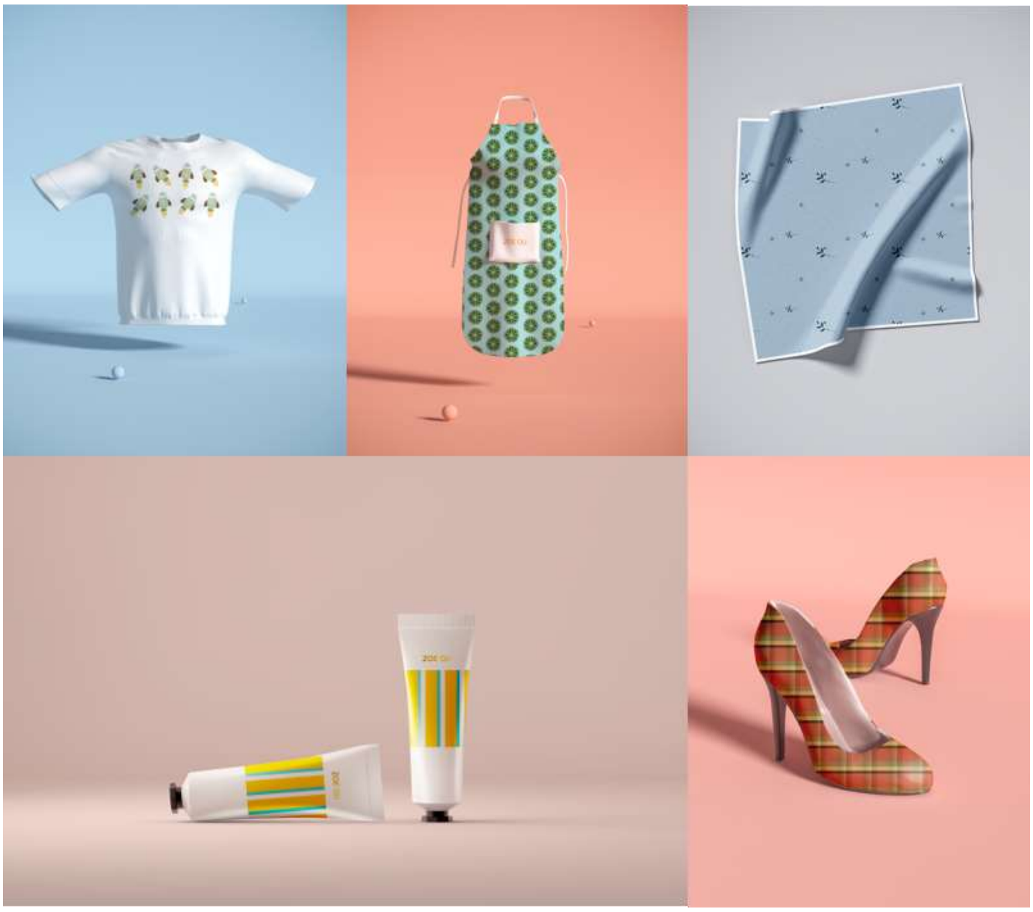


Figure 6-16 Generated digital patterns used in fashion products, including a T-shirt, an apron, a scarf, a hand cream packaging, and a pair of high-heel shoes

6.5 System Evaluation by Questionnaire Survey

6.5.1 Experimental evaluation and questionnaire design

The proposed G-system is aimed for digital pattern design support. As outlined in the background introduction (section 6.1) that a digital pattern design support system in line with the industry expectations should generate designs (i.e. the system outputs) conforming to basic human aesthetics and the outputs should be vector-based, easily to be re-used and re-edited and ready for production. To evaluate the proposed G-system, a questionnaire survey was conducted, in which targeted users with certain work/study experience in fashion industry were invited to evaluate the system, in terms of system output quality and system performance. Each subject participated in the evaluation individually. In the questionnaire survey, each subject was shown a short video explaining how the system operates. Next, subjects were shown sequentially system outputs – three groups of generated digital patterns including 13 stripe patterns, 10 check patterns, and 28 motif patterns. Subjects were then asked to complete a survey to evaluate the system, and the questionnaire used for the survey study is given in Appendix 1.

This questionnaire consists of two sections: section A evaluates the system outputs and performance and section B collects demographics of the participants. In section A, subjects were asked to give subjective assessment on their agreements or disagreements on 11 statements, in which the first nine statements are used to evaluate whether the outcome stripe, check and motif patterns conform to basic human aesthetics and whether the outcomes have desired quality in design-related works and be ready for production and their degree of novelty. The remaining two statements are used to give overall assessment regarding the system’s efficiency and convenience, and usefulness of supporting digital pattern design. All these 11 statements are evaluated in Likert scale of 1-5, “1= Strongly disagree” and “5= Strong agree". In addition, an open question is used to solicit participants’ suggestion(s) on the proposed system. Section B includes 5 questions on demographics information of the participants, including gender, age, work experience, and locations and product categories of their companies.

6.5.2 Survey results

A total of 44 participants working/studying in the fashion-related industry were recruited by convenient sampling method, and the questionnaires were distributed electronically by email. At the end, 30 qualified questionnaires were collected, representing a response rate of 68.2%.

Two out of the 30 participants are males and the remaining 28 are females. Among these participants, 83.3% of them have more than three years of working experience (see Figure 6-17(a)), and 80.0% of them are aged between 25-30 years old, and 10% each were in 30-35 years old group and 35-40 years old group (see Figure 6-17(b)). In addition, their companies are all based in Asia, and most of them work is in womenswear sectors, and some work in materials and textiles sectors of the fashion and textile supply chain. This implies that these results of the study represent the view of the fashion and textile industry, because all participants have related work experience and specialized knowledge.

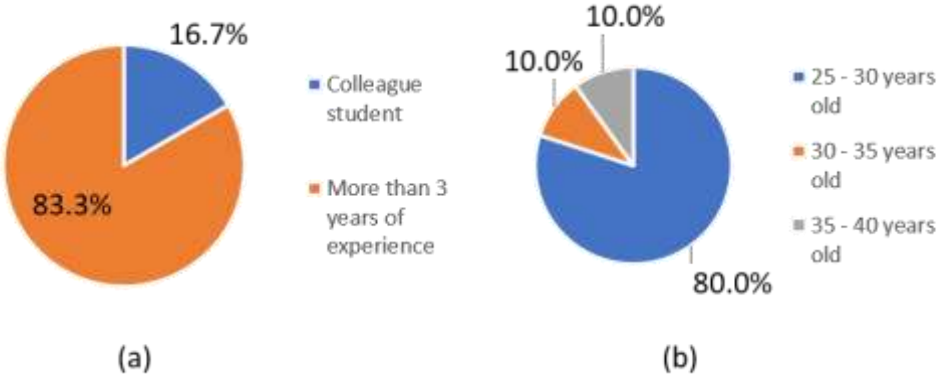


Figure 6-17 (a) Distribution of work experience of participants; (b) distribution of age groups of participants

Table 6-2 summarises the survey results in average score, standard deviation, and percentages in agreement, neural and disagreement. The results of the survey are very exciting. In terms of aesthetic values of the generated patterns, the average score is 4.5 with a standard deviation of 0.7, indicating that the system can generate digital patterns, meeting aesthetic requestments of humans. Comparatively, the class of generated check patterns has the highest mean score and the largest percentage of agreement (96.7%), but there was one participant (representing 3.3%) disagreed that the generated check patterns are aesthetic pleasing. This is absent in the cases of stripe and motif patterns; even thought the average score of these two classes are lower than that of check patterns. Another aspect to assessment the quality of system outputs is that whether or not generated patterns meet the general requirements for production, i.e. comparable and compatible with most commercial software. The average score is 4.4 with a standard deviation of 0.7, demonstrating that the outcomes meet the general requirements for production. Specifically, 90.0% of participants agreed or strongly agreed that the stripe, check, and motif patterns are in line with industry requirements, which are much larger than those who disagreed and had neutral opinion. Again, there is no disagreement on the system's output readiness for production except one participant (3.3%) viewed the generated check patterns were not ready for production or incomparable to commercial software. An intelligent system should have the ability to create new and innovative designs. Therefore, the subjects were asked to rate their agreement or disagreement on the generated patterns are novel and compliance with industrial regulations on design creation. The average score for design novelty is lowest among the three assessment areas, namely 4.2 with a standard deviation of 1.0. Although the score is slightly lower compared with the above two aspects, majority of the participants still agreed that the generated patterns are novel, with 83.3% of participants agreed with the novelty of stripe and motif patterns, and 76.7% agreed with check patterns.

Table 6-2 Assessments of the G-system (5 means = 'strongly agree' whereas 1 means = 'strongly disagree)

	1	*2*	*3*	*4*	*5*	*Average score*	*Standard deviation*	*Disagree (1-2) (%)*	*Neutral (3) (%)*	*Agree (4-5) (%)*
Generated *Stripe* patterns are in general with aesthetic values	0	0	4	11	15	4.4	0.7	0.0	13.3	86.7
Generated *Check* patterns are in general with aesthetic values	0	1	0	11	18	4.5	0.7	3.3	0.0	96.7
Generated *Motif* patterns are in general with aesthetic values	0	0	3	8	19	4.5	0.7	0.0	10.0	90.0
Average across classes of aesthetic values						**4.5**	**0.7**			
Generated *Stripe* patterns meet the general requirements for production	0	0	3	10	17	4.5	0.7	0.0	10.0	90.0
Generated *Check* patterns meet the general requirements for production	0	1	2	10	17	4.4	0.8	3.3	6.7	90.0
Generated *Motif* patterns meet the general requirements for production	0	0	3	11	16	4.4	0.7	0.0	10.0	90.0
Average across classes on meeting general requirements for production						**4.4**	**0.7**			
Generated *Stripe* patterns are novel	0	3	4	10	13	4.1	1.0	10.0	13.3	76.7
Generated *Check* patterns are novel	0	1	4	9	16	4.3	0.8	3.3	13.3	83.3
Generated *Motif* patterns are novel	2	1	2	11	14	4.1	1.1	10.0	6.7	83.3
Average across classes in terms of design novelty						**4.2**	**1.0**			
The system is efficient and convenient to use.	0	0	2	13	15	**4.4**	**0.6**	0.0	6.7	93.3
The system is useful to support digital pattern design.	0	1	0	9	20	**4.6**	**0.7**	3.3	0.0	96.7

Apart from the effectiveness and quality of the generated design outputs, the survey study also covered other system performance of the proposed G-system, including efficiency and user-friendiness as well as usefulness in supporting designs. The results demonstrate the G-system is efficient and convenient to use, with an average score of 4.4 and a standard deviation of 0.6. This is indeed a very high score, and 93.3% of the participants gave positive feedback. In terms of the proposed system being useful for design support, the average score is 4.6, with a standard deviation of 0.7. It is especially promising that 97% of the participants highly agreed on this. These results offer compelling proof that the G-system can assist in the creation of digital pattern and has a wide variety of potential applications.

A few qualitative feedbacks were received among the 30 participants. For example, some suggestions on interactively adding or removing design elements or manually swap the sequence of colorways, and improving the color consistency between the outcomes and the reference images.

6.6 Chapter Summary

In this chapter, a vector-based digital pattern generation intelligent system (G-system) has been developed for design support. The G-system can automatically handle stripe, check, and motif pattern generation. By adopting the standard design guidelines and the learned design resources from CHAPTER 4 and CHAPTER 5, the generated digital patterns align with human aesthetics. Other than inputting design requirements and reference images, the proposed system does not require any additional action from the user. Particularly, the G-system first generates vector-based geometric information for each class of patterns. This phase includes four sub-phases: 1) parameterizing the classic repeat structures, 2) proposing a novel layout rule based on the golden ratio for motif patterns, 3) combing the vector-based design elements with the parameterized layout rules, and 4) generating geometric information for stripe and check patterns and repeating them according to sizes requirement. On the ground of the generated vector-based geometric information of digital patterns, the G-system extracts color palettes from reference images and then colorizes the patterns following the extracted color palettes. Furthermore, the Pantone codes of the desired colors are recommended for production purpose. The outcomes of the proposed G-system are vector-based, which is different from the previous works and more in line with the fashion industry requirements. To visualize the application effect, this study simulated the generated digital patterns on some 3D models. A survey study with fashion practitioners shows that the generated results satisfy basic human aesthetics, assist design creation that are ready for production.

This study is an extension and exploration of computer-aided design techniques for vector-based digital pattern generation. Despite its exploratory nature, several questions still remain to be answered. First, the speed of generating stripe and check patterns still need to be accelerated. Secondly, the motif patterns currently generated can only use one design element, which causes the generated patterns are still not rich enough. These questions will be addressed in future work.

CHAPTER 7. Conclusions and Suggestions for Future Research

This thesis works on establishing a REG framework consisting of three intelligent systems for digital pattern analysis and design support: R-system, E-system, and G-system, a brief structure review of the REG framework is shown in Figure 7-1. It focuses on three research tasks: repeated pattern detection, design elements extraction and vectorization, and vector-based digital pattern generation. Particularly, the R-system focuses on repeated pattern detection and E-system focuses on design element extraction and vectorization. These two systems are mainly used to analyze existing designs and from which to extract their useful design information. Moreover, the G-system is proposed for supporting design work by automatically generating vector-based digital patterns. To provide appropriate design support that conforms to the basic human aesthetic and meets the industry's requirements, the G-system leverages the design information extracted from the R-system and E-system. Specific conclusions, the limitation of this thesis, and the potential future directions are given in this chapter.

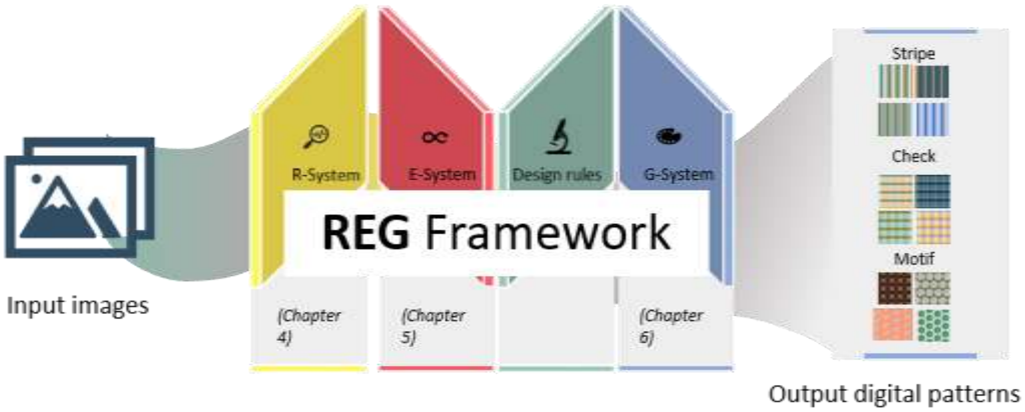


Figure 7-1 Brief structure review of the REG framework

7.1 Conclusions

The study aims to develop methods and systems to automatically analyze digital pattern images, and from which to support the creation of new digital patterns, conforming to human aesthetic and production requirements. A new framework consists of three intelligent systems have been proposed and developed in this study, according to eight defined research objectives. The key findings of this study can be concluded as follows:

(1) For the sake of analyzing existing design examples and supporting design, this study has first analyzed the industrial practice and the state of the art methods for computer-aided digital pattern generation in CHAPTER 1. Next in CHAPTER 2, the related topics have been reviewed, including computer-aided digital pattern design in fashion industry, image processing and understanding techniques, an overview of convolutional neural network, as well as design

generation methods/systems. The findings from these review have addressed the defined research objectives (i) and (ii) accordingly.

(2) R-system leverages CNN activations and repeated patterns' autocorrelation properties for efficient repeated pattern detection fulfilling objectives (iii) and (iv) is proposed in CHAPTER 4. Repeated pattern detection is a classic computer vision task that supports many downstream applications. Based on CNN activations reflecting the corresponding location information of repeated patterns within an unknown image, coarse repeated pattern size is predicted. Particularly, to process the image efficiently, the R-system leverages the boundary detection result to assist CNN activations selection. Next, the coarse size is optimized by a method leveraging template matching with a dynamic threshold. As a result, an image with a repeated pattern grid and a patch of the repeated pattern are obtained. To evaluate the proposed method, a dataset (RPD) has been built in this study with 841 design images, covering regular repeated pattern images and near-regular repeated pattern images. The proposed R-system has been evaluated and compared to other state-of-the-art systems on this dataset and other public dataset. The experimental results have demonstrated that our work achieves a good balance between accuracy and efficiency on both regular and near-regular images. Specifically, the accuracy

of our method is 0.673, which is 20% higher than the baseline method and the time cost is only 1/9 of the baseline method. Our work also gets promising results on manufactured environmental images. Detail ablation study and comparative experiments have been conducted to demonstrate the effectiveness of each submodule in the proposed R-system. It has shown that the R-system is effective for image understanding and it simplifies subsequent tasks of image understanding, namely design element extraction.

(3) E-system has been proposed in CHAPTER 5 to automatically extract and vectorize design elements from an unknown design image, fulfilling the defined research objectives (v) and (vi). In particular, an unsupervised segmentation method has been developed to extract design elements from digital pattern images. Unsupervised segmentation separates the foreground design elements and the image background by iteratively optimizing the parameters of the network and segmentation results in comparison to that from Graph-based Segmentation (GS). Moreover, a novel design element deconstruction method based on color quantization has been proposed for design element vectorization. Since vector-based design elements contain geometric and color parameters, by doing design element deconstruction based on color quantization, both the geometric and color information can be obtained simultaneously. After that,

meaningless vector paths will be removed. As a result, the vector-based core design elements in compact structures from an unknown image will be obtained. Comprehensive experiments were conducted to evaluate the proposed E-system. In terms of extraction of core design elements from design images, an experiment on a dataset of 114 digital design images, the average processing time cost is about 10 seconds, and its efficiency even outperforms humans. The success detection rate is 96%, which is higher than other state of the art systems. In terms of experiments for design element vectorization, the proposed method is proven efficient (with a processing time of 3 seconds on average), and the outputs are compact. The proposed system has outstanding application prospects. Furthermore, the proposed E-system has been applied to develop a design element dataset with 684 vector-based design elements.

(4) G-system has been developed in CHAPTER 6 for generation of stripe, check, and motif vector-based digital patterns, fulfilling the defined research objectives (vii) and (viii). The proposed G-system takes advantage of the learned design information from the R-system and E-system, especially the vector-based design elements, color combinations, and layouts, and builds corresponding parametric models for the generation of the three classes of digital pattern, namely stripe, check, and motif patterns. An innovative coloring method has

been developed that representative colors are analyzed and extracted to create a 'color palette' from a reference image and colors are reapply in various combinations to create different colorways/schemes of the new designs. Pantone standard color swatches are widely applied in the fashion industry for color communication. Instead of manually matching Pantone codes, the G-system has designed an algorithm to automatically recommend best matched Pantone codes to support and ease the design process. To do so, a total of 1275 Pantone colors with their RGB values are collected and recorded. By calculating the Euclidean distance between standard Pantone color and the color extracted from a reference images, three Pantone codes that are most similar to the extracted colors, namely the Pantones with the highest Cosine Similarity, are selected and recommended. The proposed G-system can automatically and efficiently generate vector-based stripe, check and motif patterns, as well as provide Pantone color information. Furthermore, the generated patterns have been applied on some 3D fashion products for demonstrative purpose. It shows that the proposed G-system can be applied in diverse applications, and is able to support both professional designers and general users without professional design training for create aesthetic pleasing digital patterns. A survey study has been conducted among a sample of 30 fashion experts to evaluate the aesthetic level of the generated results and the effectiveness of the design support system

(G-system), and the survey received show very positive results and feedback towards the proposed G-system. The G-system promotes the use of artificial intelligence for design generation that meets human aesthetics and eases the fashion product development.

In summary, the three proposed intelligent systems formulate a new REG framework, which is effective and useful to understanding and mining useful knowledge from existing digital pattern images and reuses such information for automatic new digital pattern creation. The generated digital patterns are aesthetic pleasing, following the design rules adopted in the fashion industry, and ready for real production with diverse applications. This new framework makes contributions to both the academia and fashion industry in digitization and design support.

7.2 Recommendations for Future Work

Despite the achievements of this thesis, there are still certain limitations, and a number of areas for improvement are identified as follows.

(1) In the R-system reported in CHAPTER 4, the principle of leveraging pre-trained CNN models to extract multi-level features and CNN activations of images to predict the repeated pattern sizes are explained. It is found that the amount of

selected CNN activations positively relates to the computation cost and the detection accuracy. In order to efficiently detect the repeated patterns from unknown images, the R-system only selects a few representative CNN activations based on the assumptions of input image characteristics (e.g. near regular repeat, no rotation or perspective effects). For other types of images, it is necessary to propose another voting algorithm to consider more CNN activations and improve the robustness and accuracy of repeated pattern detection. Another area that is worthy to further research is investigating the difference in feature extraction ability among different CNNs. With a better CNN activation voting method, more CNN filters could be extracted from deeper CNNs, it is attractive research on how to better balance detection accuracy and time consumption.

(2) The proposed design element extraction method in the E-system leverages an unsupervised segmentation method and several traditional computer vision techniques, which are still inadequate in treating complex patterns (see examples in Figure 5-14). Further improvement should be made by adding knowledge-based constraints in the CNN model to handle more complex images. In addition, Potrace (Selinger, 2003) has been used as our vectorization approach. Although Potrace is efficient, the level of detail should be further

improved. It is worth investigating new vectorization methods in the future, for example, vectorization by directly processing the segmentation masks.

(3) In terms of the G-system, different mathematic or parametric models have been developed for different types of digital patterns. Particularly, the layout of design elements can be described mathematically and used to generate various digital patterns quickly. However, G-system is still limited in variations of the generated patterns. For example, only one design element was considered for motif pattern generation in the current system, and their repeat structures are not rich enough to cover the actual design requirements. Therefore, the system will be further researched to cover the generation of motif patterns with multiple design elements and/or more complex layout arrangements, such as polygon repeat structure. Moreover, a user-friendly interface will be designed to allow general users without professional design training to take part in the design process.

APPENDIX 1: Questionnaire Survey on Vector-based Digital Pattern Generation System (G-system)

Confidentiality Statement

Participation in this survey is voluntary. All data collected through this survey is kept strictly confidential. Respondent identities will not be revealed in any report or presentation of survey results.

Section A: System evaluation

Please rate your agreement to following statements in 5-point scale.

(5 means = 'strongly agree' whereas 1 means = 'strongly disagree').

1. I assessed the digital designs of *stripe* patterns in general with aesthetic values

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

2. The digital design of *strip* patterns can meet the general requirements for production.

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

3. I am pleased/satisfied with the digital designs of *stripe* patterns, which are comparable to the designs created by professional designers, in terms of novelty of output designs and compliance of industrial regulations on design creation.

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

4. I assessed the digital designs of *check* patterns in general with aesthetic values

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

5. The digital design of *check* patterns can meet the general requirements for production.

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

6. I am pleased/satisfied with the digital designs of *check* patterns, which are comparable to the designs created by professional designers, in terms of novelty of output designs and compliance of industrial regulations on design creation.

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

7. I assessed the digital designs of *motif* patterns in general with aesthetic values

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

8. The digital design of *motif* patterns can meet the general requirements for production.

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

9. I am pleased/satisfied with the digital designs of *motif* patterns, which are comparable to the designs created by professional designers, in terms of novelty of output designs and compliance of industrial regulations on design creation.

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

10. The system is efficient and convenient to use.

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

11. The system is useful to support digital pattern design.

1 *Strongly disagree* 2 *Disagree* 3 *Neutral* 4 *Agree* 5 *Strongly agree*

12. What are your suggestion(s) to improve the system?

Section B: Participants Information

13. Please select what best describes your experience in the fashion industry.

A. I am a college student of fashion and related program

B. I have less than 3 years of experience in fashion and related industry

C. I have more than 3 years of experience in fashion and related industry

14. What gender do you identify as?

A. Male

B. Female

C. Prefer not to say

15. What is your age?

A. 20 - 25 years old

B. 25 - 30 years old

C. 35 - 40 years old

D. 40+

E. Prefer not to say

16. Where is your company located?

A. North America/Central America

B. South America

C. Europe

D. Africa

E. Asia

F. Australia

G. Caribbean Islands

H. Pacific Islands

I. Other: ______

J. Prefer not to say

17. What are the product categories of your company? (select all applicable answers)

A. Accessories

B. Active

C. Colour

D. Denim

E. Footwear

F. Intimates

G. Jewellery

H. Kidswear

I. Knitwear

J. Materials & Textiles

K. Menswear

L. Prints & Graphics

M. Swimwear

N. Womenswear and Youth

O. Other: ______

P. Prefer not to say

REFERENCES

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., & Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(11), 2274-2282.

Adobe. (2022). *Adobe illustrator 2022: Image trace*. http://www.adobe.com/

Ahmad, J., Muhammad, K., & Baik, S. W. (2018). Medical image retrieval with compact binary codes generated in frequency domain using highly reactive convolutional features. *Journal of medical systems*, *42*(2), 1-19.

Akata, Z., Perronnin, F., Harchaoui, Z., & Schmid, C. (2013). Good practice in large-scale learning for image classification. *IEEE transactions on pattern analysis machine intelligence*, *36*(3), 507-520.

Al-Saffar, A. A. M., Tao, H., & Talab, M. A. (2017). Review of deep convolution neural network in image classification. 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET),

Albert, F., Gomis, J. M., Valor, M., & Valiente, J. M. (2004). Methodology for graphic redesign applied to textile and tile pattern design. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems,

Aldrich, W. (2013). *Fabrics and pattern cutting*. John Wiley & Sons.

Ambrose, G., Harris, P., & Ball, N. (2019). *The fundamentals of graphic design*. Bloomsbury Publishing.

Appalaraju, S., & Chaoji, V. (2017). Image similarity using deep CNN and curriculum learning. *arXiv preprint arXiv:1709.08761*.

ApparelMagic. (2022). *Apparel Software for Total Control*. Retrieved 2022, August 30 from https://apparelmagic.com/

Avramović, D., Vladić, G., Kašiković, N., & Ivan, P. (2013). Aplicability of golden ratio rule in modern product design. *Journal of Graphic Engineering and Design*, *4*(1), 29-35.

Bandinelli, R., Rinaldi, R., Rossi, M., & Terzi, S. (2013). New product development in the fashion industry: an empirical investigation of Italian firms. *International Journal of Engineering Business Management*, *5*(Godište 2013), 5-31.

Bang, S., Korosteleva, M., & Lee, S. H. (2021). Estimating Garment Patterns from Static Scan Data. Computer Graphics Forum,

Bay, H., Tuytelaars, T., & Gool, L. V. (2006). Surf: Speeded up robust features. European conference on computer vision,

Bergen, S., & Ross, B. J. (2012). Automatic and interactive evolution of vector graphics images with genetic algorithms. *The Visual Computer*, *28*(1), 35-45.

Bertolotti, F., Macrì, D. M., & Tagliaventi, M. R. (2004). Social and organisational implications of CAD usage: a grounded theory in a fashion company. *New Technology, Work and Employment*, *19*(2), 110-127.

Bhargavi, K., & Jyothi, S. (2014). A survey on threshold based segmentation technique in image processing. *International Journal of Innovative Research and Development*, *3*(12), 234-239.

Blázquez, M. (2014). Fashion shopping in multichannel retail: The role of technology in enhancing the customer experience. *International Journal of Electronic Commerce*, *18*(4), 97-116.

Bleau, A., & Leon, L. J. (2000). Watershed-based segmentation and region merging. *Computer Vision and Image Understanding*, *77*(3), 317-370.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Bolon-Canedo, V., & Remeseiro, B. (2020). Feature selection in image analysis: a survey. *Artificial Intelligence Review*, *53*(4), 2905-2931.

Bouton, G. D. (2017). *CorelDRAW X8: The Official Guide*. McGraw-Hill.

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Briggs-Goode, A., & Townsend, K. (2011). Textile design: principles, advances and applications.

Brunelli, R. (2009). *Template matching techniques in computer vision: theory and practice*. John Wiley & Sons.

Bylinskii, Z., Kim, N. W., O'Donovan, P., Alsheikh, S., Madan, S., Pfister, H., Durand, F., Russell, B., & Hertzmann, A. (2017). Learning visual importance for graphic designs and data visualizations. Proceedings of the 30th Annual ACM symposium on user interface software and technology,

Cai, Y., & Baciu, G. (2011). Detection of repetitive patterns in near regular texture images. 2011 IEEE 10th IVMSP Workshop: Perception and Visual Signal Analysis,

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. European conference on computer vision,

Camargo, L. R., Pereira, S. C. F., & Scarpin, M. R. S. (2020). Fast and ultra-fast fashion supply chain management: an exploratory research. *International Journal of Retail & Distribution Management*, *48*(6), 537-553.

Canny, J. (1986). A computational approach to edge detection. *IEEE transactions on pattern analysis machine intelligence*(6), 679-698.

Cao, Y., Chan, A. B., & Lau, R. W. (2017). Mining probabilistic color palettes for summarizing color use in artwork collections. SIGGRAPH Asia 2017 Symposium on Visualization,

Celebi, M. E., & Aslandogan, Y. A. (2005). A comparative study of three moment-based shape descriptors. International Conference on Information Technology: Coding and Computing (ITCC'05)-Volume II,

Cetinic, E., & She, J. (2022). Understanding and creating art with AI: Review and outlook. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *18*(2), 1-22.

Chai, D., & Bouzerdoum, A. (2000). A Bayesian approach to skin color classification in YCbCr color space. 2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium (Cat. No. 00CH37119),

Chen, T.-W., Chen, Y.-L., & Chien, S.-Y. (2008). Fast image segmentation based on K-Means clustering with histograms in HSV color space. 2008 IEEE 10th Workshop on multimedia signal processing,

Chen, W.-d. D., Wei. (2008). An improved median-cut algorithm of color image quantization. 2008 International Conference on Computer Science and Software Engineering,

Cheng, H.-D., Jiang, X. H., Sun, Y., & Wang, J. J. P. r. (2001). Color image segmentation: advances and prospects. *34*(12), 2259-2281.

Cheng, M.-M., Zhang, F.-L., Mitra, N. J., Huang, X., & Hu, S.-M. (2010). Repfinder: finding approximately repeated scene elements for image editing. *ACM Transactions on Graphics (TOG)*, *29*(4), 1-8.

CLO. (2022). *CHANGING THE WORLD WITH VIRTUAL GARMENTS*. Retrieved 2022, August 30 from https://www.clo3d.com/zh/

Cohen-Or, D., Sorkine, O., Gal, R., Leyvand, T., & Xu, Y.-Q. (2006). Color harmonization. In *ACM SIGGRAPH 2006 Papers* (pp. 624-630).

Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 603-619.

Conners, R. W., & Harlow, C. A. (1980). Toward a structural textural analyzer based on statistical methods. *Computer Graphics and Image Processing*, *12*(3), 224-256.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05),

Delon, J., Desolneux, A., Lisani, J. L., & Petro, A. B. (2005). Automatic color palette. IEEE international conference on image processing 2005,

Demir, G., Çekmiş, A., Yeşilkaynak, V. B., & Unal, G. (2021). Detecting visual design principles in art and architecture through deep convolutional neural networks. *Automation in Construction*, *130*, 103826.

Dewan, J. H., & Thepade, S. D. (2020). Image retrieval using low level and local features contents: a comprehensive review. *Applied Computational Intelligence and Soft Computing*, *2020*.

Dhakar, L. (2020). *COLOR THIEF*. Retrieved 2022, August 30 from https://lokeshdhakar.com/projects/color-thief/#getting-started

Ding, Y. (2021). *Deep learning-based fashion advising* the Hong Kong Polytechnic University].

Dominici, E. A., Schertler, N., Griffin, J., Hoshyari, S., Sigal, L., & Sheffer, A. (2020). Polyfit: Perception-aligned vectorization of raster clip-art via intermediate polygonal fitting. *ACM Transactions on Graphics (TOG)*, *39*(4), 77: 71-77: 16.

Dong, G., & Xie, M. (2005). Color clustering and learning for image segmentation based on neural networks. *IEEE transactions on neural networks*, *16*(4), 925-936.

Doubek, P., Matas, J., Perdoch, M., & Chum, O. (2010). Image matching and retrieval by repetitive patterns. 2010 20th International Conference on Pattern Recognition,

du Buf, J. H., Kardan, M., & Spann, M. (1990). Texture feature performance for image segmentation. *Pattern recognition*, *23*(3-4), 291-309.

Dutta, S., & Chaudhuri, B. B. (2009). A color edge detection algorithm in RGB color space. 2009 International conference on advances in recent technologies in communication and computing,

Egmont-Petersen, M., de Ridder, D., & Handels, H. (2002). Image processing with neural networks—a review. *Pattern recognition*, *35*(10), 2279-2301.

El Amin, A. M., Liu, Q., & Wang, Y. (2016). Convolutional neural network features based change detection in satellite images. First International Workshop on Pattern Recognition,

FashionUnited. (2022). *Global Fashion Industry Statistics*. Retrieved 2022, August 30 from https://fashionunited.com/global-fashion-industry-statistics/

Favreau, J.-D., Lafarge, F., & Bousseau, A. (2017). Photo2clipart: Image abstraction and vectorization using layered linear gradients. *ACM Transactions on Graphics (TOG)*, *36*(6), 1-11.

Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International journal of computer vision*, *59*(2), 167-181.

Field, M., & Golubitsky, M. (2009). *Symmetry in chaos: a search for pattern in mathematics, art, and nature*. SIAM.

Fogg, M. (2006). *Print in fashion: design and development in fashion textiles*. Batsford.

Gijsenij, A., Vazirian, M., Spiers, P., Westland, S., & Koeckhoven, P. (2022). Determining key colors from a design perspective using dE-means color clustering. *Color Research & Application*.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics,

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139-144.

Guerrero, P., Bernstein, G., Li, W., & Mitra, N. (2016). PATEX: exploring pattern variations. *ACM Transactions on Graphics*, *35*(4), 1-13.

Guo, M., Huang, D., & Xie, X. (2021). The Layout Generation Algorithm of Graphic Design Based on Transformer-CVAE. 2021 International Conference on Signal Processing and Machine Learning (CONF-SPML),

Guo, Y., Ge, X., Yu, M., Yan, G., & Liu, Y. (2019). Automatic recognition method for the repeat size of a weave pattern on a woven fabric image. *Textile Research*

Journal, *89*(14), 2754-2775.

Hafner, J., Sawhney, H. S., Equitz, W., Flickner, M., & Niblack, W. (1995). Efficient color histogram indexing for quadratic form distance functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(7), 729-736.

Hampshire, M., & Stephenson, K. (2006). *Communicating with pattern : stripes*. RotoVision.

Hann, M. A. (2019). *The grammar of pattern* (First edition. ed.). CRC Press, Taylor & Francis Group.

He, J., Zhang, S., Yang, M., Shan, Y., & Huang, T. (2019). Bi-directional cascade network for perceptual edge detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,

He, J., Zhang, S., Yang, M., Shan, Y., & Huang, T. (2020). BDCN: Bi-directional cascade network for perceptual edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *44*(1), 100-113.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. Proceedings of the IEEE international conference on computer vision,

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition,

Hembree, R. (2006). *The complete graphic designer: a guide to understanding graphics and visual communication*. Rockport publishers.

Hiremath, P., & Pujari, J. (2007). Content based image retrieval using color, texture and shape features. 15th International conference on advanced computing and communications (ADCOM 2007),

Hoshyari, S., Dominici, E. A., Sheffer, A., Carr, N., Wang, Z., Ceylan, D., & Shen, I.-C. (2018). Perception-driven semi-structured boundary vectorization. *ACM Transactions on Graphics (TOG)*, *37*(4), 1-14.

Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE transactions on information theory*, *8*(2), 179-187.

Hua, W., Hou, M., Yang, S., & Dong, Y. (2019). Discrimination of cultural relics similarity based on phash algorithm and sift operator. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *42*, 571-575.

Huang, H., Mok, P. Y., Kwok, Y., & Au, J. (2012). Block pattern generation: From parameterizing human bodies to fit feature-aligned and flattenable 3D garments. *Computers in Industry*, *63*(7), 680-691.

Humeau-Heurtier, A. (2019). Texture feature extraction methods: A survey. *IEEE Access*, *7*, 8975-9000.

Hurley, R. A., Randall, R., O'Hara, L., Tonkin, C., & Rice, J. C. (2017). Color harmonies in packaging. *Color Research & Application*, *42*(1), 50-59.

Inglis, T., & Kaplan, C. S. (2012). Pixelating vector line art. SIGGRAPH Posters, Inkscape. (2020). *Inkscape Project*. https://inkscape.org

Inoue, N., Furuta, R., Yamasaki, T., & Aizawa, K. (2018). Cross-domain weakly-supervised object detection through progressive domain adaptation. Proceedings of the IEEE conference on computer vision and pattern recognition,

Itten, J., & Veres, P. (1961). The Art of Color the Subjective Experience and Objective Rationale of Color.

Jalal, G., Maudet, N., & Mackay, W. E. (2015). Color portraits: From color picking to interacting with color. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems,

Jing, J., Pinghua, X., Xiaowan, S., Jingwen, C., & Ruibing, L. (2022). Automatic coloration of pattern based on color parsing of Sung porcelain. *Textile Research Journal*, 00405175221113088.

Kabbai, L., Abdellaoui, M., & Douik, A. (2019). Image classification by combining local and global features. *The Visual Computer*, *35*(5), 679-693.

Kaledo. (n.d.). *Kaledo, Lectra's design software to create products, storyboards, technical sketches*. Retrieved 2022, August 30 from https://www.lectra.com/en/products/kaledo

Kanezaki, A. (2018). Unsupervised image segmentation by backpropagation. 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP),

Karami, E., Prasad, S., & Shehata, M. (2017). Image matching using SIFT, SURF, BRIEF and ORB: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*.

Kaur, D., & Kaur, Y. (2014). Various image segmentation techniques: a review. *International Journal of Computer Science and Mobile Computing*, *3*(5), 809-814.

Ke, Y., & Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.,

Kerautret, B., & Lachaud, J.-O. (2019). Geometric Total Variation for Image Vectorization, Zooming and Pixel Art Depixelizing. Asian Conference on Pattern Recognition,

Khajeh, M., Payvandy, P., & Derakhshan, S. J. (2016). Fashion set design with an emphasis on fabric composition using the interactive genetic algorithm. *Fashion Textiles*, *3*(1), 8.

Kim, W., Kanezaki, A., & Tanaka, M. (2020). Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE transactions on image processing*, *29*, 8055-8068.

Ko, E., Kim, E., Taylor, C. R., Kim, K. H., & Kang, I. J. (2007). Cross-national market segmentation in the fashion industry: A study of European, Korean, and US consumers. *International marketing review*.

Kopf, J., & Lischinski, D. (2011). Depixelizing pixel art. In *ACM SIGGRAPH 2011 papers* (pp. 1-8).

Koppermann. (n.d.). *TEX-DESIGN™ - Your creative software in 2 minutes*. Retrieved 2022, August 30 from https://www.koppermann.com/tex-design-your-creative-software-in-2-minutes/?lang=en

Kovacs, B., O'Donovan, P., Bala, K., & Hertzmann, A. (2018). Context-aware asset search for graphic design. *IEEE Transactions on Visualization and Computer Graphics*, *25*(7), 2419-2429.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems,

Kumar, R., Tripathi, R., Marchang, N., Srivastava, G., Gadekallu, T. R., & Xiong, N. N. (2021). A secured distributed detection system based on IPFS and blockchain for industrial image and video data security. *Journal of Parallel and Distributed Computing*, *152*, 128-143.

Kuo, C., Huang, Y., Su, T., & Shih, C. (2008). Computerized Color Distinguishing System for Color Printed Fabric by Using the Approach of Probabilistic Neural Network. *Polymer-plastics technology and engineering*, *47*(3), 264-272. https://doi.org/10.1080/03602550701866808

Kuo, C., Lee, C., & Shih, C. (2017). Image database of printed fabric with repeating dot patterns part (I)–image archiving. *Textile Research Journal*, *87*(17), 2089-2105.

Kuo, C., Shih, C., & Lee, J. (2008). Separating Color and Identifying Repeat Pattern Through the Automatic Computerized Analysis System for Printed Fabrics. *Journal of Information Science and Engineering*, *24*(2).

Lai, P., & Westland, S. (2020). Machine learning for colour palette extraction from fashion runway images. *International Journal of Fashion Design, Technology and Education*, *13*(3), 334-340.

Lectra. (n.d.). *What is Gerber AccuMark?* Retrieved 2022, August 30 from https://www.lectra.com/en/products/gerber-accumark-accunest-fashion

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Lertrusdachakul, T., Ruxpaitoon, K., & Thiptarajan, K. (2019). Color Palette Extraction by Using Modified K-means Clustering. 2019 7th International Electrical Engineering Congress (iEECON),

Lettry, L., Perdoch, M., Vanhoey, K., & Van Gool, L. (2017). Repeated pattern detection using CNN activations. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV),

Li, J., Yang, J., Hertzmann, A., Zhang, J., & Xu, T. (2019). Layoutgan: Generating graphic layouts with wireframe discriminators. *arXiv preprint arXiv:1901.06767*.

Li, R., Zhou, Y., Zhu, S., & Mok, P. (2017). Intelligent clothing size and fit recommendations based on human model customisation technology. 25th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG 2017,

Li, T., Wu, B., Yang, Y., Fan, Y., Zhang, Y., & Liu, W. (2019). Compressing convolutional neural networks via factorized convolutional filters. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Li, Y., Mao, J., Zhang, X., Freeman, W. T., Tenenbaum, J. B., & Wu, J. (2020). Perspective plane program induction from a single image. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Likas, A., Vlassis, N., & Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition*, *36*(2), 451-461.

Lin, H.-C., Wang, L.-L., & Yang, S.-N. (1997). Extracting periodicity of a regular texture based on autocorrelation functions. *Pattern recognition letters*, *18*(5), 433-443.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. European conference on computer vision,

Lin, W.-C., Hays, J., Wu, C., Liu, Y., & Kwatra, V. (2006). Quantitative evaluation of near regular texture synthesis algorithms. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06),

Lindeberg, T. (2012). Scale invariant feature transform.

Liu, J., & Liu, Y. (2013). *Grasp recurring patterns from a single view* Proceedings of the IEEE conference on computer vision and pattern recognition,

Liu, J., Psarakis, E. Z., Feng, Y., & Stamos, I. (2018). A kronecker product model for repeated pattern detection on 2d urban images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(9), 2266-2272.

Liu, J., Psarakis, E. Z., Feng, Y., & Stamos, I. (2019, Sep). *A Kronecker Product Model for Repeated Pattern Detection on 2D Urban Images* IEEE Trans Pattern Anal Mach Intell, https://www.ncbi.nlm.nih.gov/pubmed/30040628

Liu, K., Zeng, X., Bruniaux, P., Tao, X., Yao, X., Li, V., & Wang, J. (2018). 3D interactive garment pattern-making technology. *Computer-Aided Design*, *104*, 113-124.

Liu, K., Zeng, X., Tao, X., & Bruniaux, P. (2019). Associate Design of Fashion Sketch and Pattern. *IEEE Access*, *7*, 48830-48837.

Liu, K., Zhu, C., Tao, X., Bruniaux, P., & Zeng, X. (2019). Parametric design of garment pattern based on body dimensions. *International Journal of Industrial Ergonomics*, *72*, 212-221.

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),

Liu, S., & Zhang, L. (2008). Textile pattern generation technique based on quasi-regular pattern theory and their transform. 2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application,

Liu, Y., Agarwala, A., Lu, J., & Rusinkiewicz, S. (2016). Data-driven iconification. Expressive,

Liu, Y., Collins, R. T., & Tsin, Y. (2004). A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE transactions on pattern analysis machine intelligence*, *26*(3), 354-371.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. Proceedings of the IEEE conference on computer vision and pattern recognition,

Long, J. L., Zhang, N., & Darrell, T. (2014). Do convnets learn correspondence? *Advances in neural information processing systems*, *27*.

Lopes, R. G., Ha, D., Eck, D., & Shlens, J. (2019). A learned representation for scalable vector graphics. Proceedings of the IEEE/CVF International Conference on Computer Vision,

Lu, S., Mok, P., & Jin, X. (2014). From design methodology to evolutionary design: An interactive creation of marble-like textile patterns. *Engineering Applications of Artificial Intelligence*, *32*, 124-135.

Lu, S., Mok, P. Y., & Jin, X. (2017). A new design concept: 3D to 2D textile pattern design for garments. *Computer-Aided Design*, *89*, 35-49.

Malik, K., & Robertson, C. (2021). Landscape Similarity Analysis Using Texture Encoded Deep-Learning Features on Unclassified Remote Sensing Imagery. *Remote Sensing*, *13*(3), 492.

Matsuda, Y. (1995). Color design. *Asakura Shoten*, *2*(4), 10.

Mechelli, A., & Vieira, S. (2019). *Machine Learning: Methods and Applications to Brain Disorders*. Academic Press.

Meng, Y., Mok, P. Y., & Jin, X. (2012). Computer aided clothing pattern design with 3D editing and pattern alteration. *Computer-Aided Design*, *44*(8), 721-734.

Meskaldji, K., Boucherkha, S., & Chikhi, S. (2009). Color quantization and its impact on color histogram based image retrieval accuracy. 2009 First International Conference on Networked Digital Technologies,

Mohamad, F. S., Manaf, A. A., & Chuprat, S. (2010). Histogram matching for color detection: A preliminary study. 2010 International Symposium on Information Technology,

Mok, P. Y., Xu, J., Wang, X., Fan, J., Kwok, Y., & Xin, J. H. (2013). An IGA-based design support system for realistic and practical fashion designs. *Computer-Aided Design*, *45*(11), 1442-1458.

Monga, V., & Evans, B. L. (2006). Perceptual image hashing via feature points:

performance evaluation and tradeoffs. *IEEE transactions on image processing*, *15*(11), 3452-3465.

Mosleh, S., Abtew, M. A., Bruniaux, P., Tartare, G., Loghin, E.-C., & Dulgheriu, I. (2021). Modeling and simulation of human body heat transfer system based on air space values in 3d clothing model. *Materials*, *14*(21), 6675.

Naik, D., & Shah, P. (2014). A review on image segmentation clustering algorithms. *Int J Comput Sci Inform Technol*, *5*(3), 3289-3293.

Narkhede, H. (2013). Review of image segmentation techniques. *International Journal of Science and Modern Engineering*, *1*(8), 54-61.

Naz, S., Majeed, H., & Irshad, H. (2010). Image segmentation using fuzzy clustering: A survey. 2010 6th international conference on emerging technologies (ICET),

Nazar, M., Khan, R. Q., Perveen, M., & Khan, W. Q. (2017). Web branding harmonizer: Need of color harmonies and its solution in website development. 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions)(ICTUS),

Neupane, P., Tuladhar, A., Sharma, S., & Tamang, R. (2019). Extracting Unknown Repeated Pattern in Tiled Images. International Conference on Hybrid Intelligent Systems,

Ng, P. C., & Henikoff, S. (2003). SIFT: Predicting amino acid changes that affect protein function. *Nucleic acids research*, *31*(13), 3812-3814.

Nikolaou, N., & Papamarkos, N. (2009). Color reduction for complex document images. *International Journal of Imaging Systems Technology*, *19*(1), 14-26.

Nixon, M., & Aguado, A. (2019). *Feature extraction and image processing for computer vision*. Academic press.

O'Donovan, P., Agarwala, A., & Hertzmann, A. (2015). Designscape: Design with interactive layout suggestions. Proceedings of the 33rd annual ACM conference on human factors in computing systems,

Ogata, Y., & Onisawa, T. (2007). Interactive clothes design support system. International conference on neural information processing,

Optitex. (n.d.). *2D & 3D INTEGRATED PATTERN DESIGN SOFTWARE*. Retrieved 2022, August 30 from https://optitex.com/products/2d-and-3d-cad-software/

Oracle. (2022). *What is SCM (Supply Chain Management)?* Retrieved 2022, August 30 from https://www.oracle.com/hk/scm/what-is-supply-chain-management/

Pak, M., & Kim, S. (2017). A review of deep learning in image recognition. 2017 4th international conference on computer applications and information processing technology (CAIPT),

Palmer, S. E., & Schloss, K. B. (2010). An ecological valence theory of human color preference. *Proceedings of the National Academy of Sciences*, *107*(19), 8877-8882.

Papamarkos, N., Atsalakis, A. E., & Strouthopoulos, C. P. (2002). Adaptive color reduction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*

(Cybernetics), *32*(1), 44-56.

Park, M., Brocklehurst, K., Collins, R. T., & Liu, Y. (2009). Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(10), 1804-1816.

Phillips, P., & Bunce, G. (1993). *Repeat patterns: A manual for designers, artists, and architects*. Thames and Hudson.

Pinho, A. J., & Ferreira, P. J. (2011). Finding unknown repeated patterns in images. 2011 19th European Signal Processing Conference,

Pritts, J., Chum, O., & Matas, J. (2014). Detection, rectification and segmentation of coplanar repeated patterns. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,

Qayum, M. A., & Naseer, M. (2017). A fast approach for finding design repeat in textile rotary printing for fault detection. *The Journal of The Textile Institute*, *108*(1), 62-65.

Quinn, L., Hines, T., & Bennison, D. (2007). Making sense of market segmentation: a fashion retailing case. *European Journal of Marketing*.

Reddy, P., Gharbi, M., Lukac, M., & Mitra, N. J. (2021). Im2vec: Synthesizing vector graphics without vector supervision. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Rodriguez-Pardo, C., Suja, S., Pascual, D., Lopez-Moreno, J., & Garces, E. (2019). Automatic extraction and synthesis of regular repeatable patterns. *Computers Graphics*, *83*, 33-41.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention,

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. European conference on computer vision,

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. 2011 International conference on computer vision,

Sartori, A., Culibrk, D., Yan, Y., & Sebe, N. (2015). Who's afraid of itten: Using the art theory of color combination to analyze emotions in abstract paintings. Proceedings of the 23rd ACM international conference on Multimedia,

Sayem, A. S. M., Kennon, R., & Clarke, N. (2010). 3D CAD systems for the clothing industry. *International Journal of Fashion Design, Technology and Education*, *3*(2), 45-53.

Schindler, G., Krishnamurthy, P., Lublinerman, R., Liu, Y., & Dellaert, F. (2008). Detecting and matching repeated patterns for automatic geo-tagging in urban environments. 2008 IEEE Conference on Computer Vision and Pattern Recognition,

Seivewright, S. (2012). *Basics fashion design 01: Research and design* (Vol. 1). A&C Black.

Selinger, P. (2003). Potrace: a polygon-based tracing algorithm. *Potrace (online), http://potrace. sourceforge. net/potrace. pdf (2009-07-01)*, *2*.

Seo, Y.-J., & Ha, J.-S. (2007). Analysis of graphic design trend in fashion design. *Journal of the Korean Society of Costume*, *57*(10), 156-168.

Shan, Y. (2018). Application of Cognitive Neuroscience in Color Composition in Graphic Design. *NeuroQuantology*, *16*(5).

Shen, I.-C., & Chen, B.-Y. (2021). Clipgen: A deep generative model for clipart vectorization and synthesis. *IEEE Transactions on Visualization and Computer Graphics*.

Shimoda, W., & Yanai, K. (2017). Learning food image similarity for food image retrieval. 2017 IEEE Third International Conference on Multimedia Big Data (BigMM),

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:.06857*.

Spahiu, T., Shehi, E., & Piperi, E. (2014). Advanced CAD/CAM systems for garment design and simulation. 6th International conference of textile,

Srivastava, D., Wadhvani, R., & Gyanchandani, M. (2015). A review: color feature extraction methods for content based image retrieval. *International Journal of Computational Engineering and Management*, *18*(3), 9-13.

Stricker, M. A., & Orengo, M. (1995). Similarity of color images. Storage and retrieval for image and video databases III,

Studd, R. (2002). The textile design process. *The Design Journal*, *5*(1), 35-49.

Swain, M., & Ballard, D. (1991). Color indexing. *International journal of computer vision*, *7*(1), 11-32.

Swain, M. J., & Ballard, D. H. (1991). Color indexing. *International journal of computer vision*, *7*(1), 11-32.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition,

Tabraz, M. (2017). Importance Of Fashion Cad (Computer-Aided Design) Study For Garment Industry In Bangladesh. *Internatıonal Journal Of Scientific& Technology Research*, *6*.

Tan, X., & Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE transactions on image processing*, *19*(6), 1635-1650.

Tao, C., Zhou, J., & Yin, M. (2017). Automatic identification of textile pattern consecutiveness based on similarity space. *Textile Research Journal*, *87*(2), 224-231.

Teunissen, J., Dervote, H. V., & Brand, J. (2015). *FASHION, GRAPHIC DESIGN & THE BODY*. Terra.

Torii, A., Sivic, J., Pajdla, T., & Okutomi, M. (2013). Visual place recognition with repetitive structures. Proceedings of the IEEE conference on computer vision and pattern recognition,

Tsetimtheo. (2020). *PATTERN AND MOTIF*. http://www.tansicollege.edu.ng/content/week-78-pattern-and-motif

Tuytelaars, T., Turina, A., & Van Gool, L. (2003). Noncombinatorial detection of regular repetitions under perspective skew. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(4), 418-432.

Valor, M., Albert, F., Gomis, J. M., & Contero, M. (2003). Textile and tile pattern design automatic cataloguing using detection of the plane symmetry group. Proceedings Computer Graphics International 2003,

VectorMagic. (2021). *Vector Magic 2021: Easily Convert JPG, PNG, GIF Files to PDF, SVG, EPS Vectors*.

Vilumsone-Nemes, I., & Belakova, D. (2020). Reduction of material consumption for garments from checked fabrics. *Industria Textila*, *71*(3), 275-281.

Vitale, R., Prats-Montalbán, J. M., López-García, F., Blasco, J., & Ferrer, A. (2016). Segmentation techniques in image analysis: A comparative study. *Journal of Chemometrics*, *30*(12), 749-758.

Wang, W.-N., & Yu, Y.-L. (2005). Image emotional semantic query based on color semantic description. 2005 International Conference on Machine Learning and Cybernetics,

Wang, W., Yang, Y., Wang, X., Wang, W., & Li, J. (2019). Development of convolutional neural network and its application in image classification: a survey. *Optical Engineering*, *58*(4), 040901.

Wang, W., Zhang, G., Yang, L., & Wang, W. (2019). Research on garment pattern design based on fractal graphics. *Eurasip journal on image and video processing*, *2019*(1), 1-15. https://doi.org/10.1186/s13640-019-0431-x

Wang, W., Zhang, G., Yang, L., & Wang, W. (2019). Research on garment pattern design based on fractal graphics. *EURASIP Journal on Image Video Processing*, *2019*(1), 1-15.

Westrup, C., & Knight, F. (2000). Consultants and enterprise resource planning (ERP) systems. *ECIS 2000 Proceedings*, 178.

Wilson, J. (2001). *Handbook of textile design*. Elsevier.

Wirth, M. A. (2004). *Texture analysis*.

Xiang, J., Zhang, N., Pan, R., & Gao, W. (2019). Fabric image retrieval system using hierarchical search based on deep convolutional neural network. *IEEE Access*, *7*, 35405-35417.

Xie, Y., Xiao, J., Huang, K., Thiyagalingam, J., & Zhao, Y. (2018). Correlation filter selection for visual tracking using reinforcement learning. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(1), 192-204.

Xu, D., Shi, Y., Tsang, I. W., Ong, Y.-S., Gong, C., & Shen, X. (2019). Survey on multi-output learning. *IEEE transactions on neural networks and learning systems*, *31*(7), 2409-2429.

Xu, J. (2015). *Development of an integrated platform for online fashion sketch design* the Hong Kong Polytechnic University].

Xu, J., Mok, P., Yuen, C., & Yee, R. (2016). A web-based design support system for fashion technical sketches. *International Journal of Clothing Science Technology*.

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, *9*(4), 611-629.

Yang, B., Yan, J., Lei, Z., & Li, S. Z. (2015). Convolutional channel features. Proceedings of the IEEE international conference on computer vision,

Yang, M., Chao, H., Zhang, C., Guo, J., Yuan, L., & Sun, J. (2015). Effective clipart image vectorization through direct optimization of bezigons. *IEEE Transactions on Visualization and Computer Graphics*, *22*(2), 1063-1075.

Yang, M., Kpalma, K., & Ronsin, J. (2008). A survey of shape feature extraction techniques. *Pattern recognition*, *15*(7), 43-90.

Yin, Y., Chen, Z., Zhao, Y., Li, J., & Zhang, K. (2020). Automated Chinese Seal Carving Art Creation with AI Assistance. 2020 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR),

Yuheng, S., & Hao, Y. (2017). Image segmentation algorithms overview. *arXiv preprint arXiv:1707.02051*.

Zannettou, S., Caulfield, T., Blackburn, J., De Cristofaro, E., Sirivianos, M., Stringhini, G., & Suarez-Tangil, G. (2018). On the origins of memes by means of fringe web communities. Proceedings of the Internet Measurement Conference 2018,

Zauner, C. (2010). Implementation and benchmarking of perceptual image hash functions.

Zhang, D., & Lu, G. (2004). Review of shape representation and description techniques. *Pattern recognition*, *37*(1), 1-19.

Zhang, J., Pan, R., Gao, W., & Zhu, D. (2015). Automatic recognition of the color effect of yarn-dyed fabric by the smallest repeat unit recognition algorithm. *Textile Research Journal*, *85*(4), 432-446.

Zhang, K., Wang, W., Lv, Z., Fan, Y., & Song, Y. (2021). Computer vision detection of foreign objects in coal processing using attention CNN. *Engineering Applications of Artificial Intelligence*, *102*, 104242.

Zhang, S.-H., Chen, T., Zhang, Y.-F., Hu, S.-M., & Martin, R. R. (2009). Vectorizing cartoon animations. *IEEE Transactions on Visualization and Computer Graphics*, *15*(4), 618-629.

Zhang, X., Wang, J., Lu, G., & Zhang, X. (2020). Pattern understanding and synthesis based on layout tree descriptor. *The Visual Computer*, *36*(6), 1141-1155.

Zhao, N. (2020). *Computational Perception for Graphic Design and Elements Generation* City University of Hong Kong].

Zhao, N., Cao, Y., & Lau, R. W. (2018). What characterizes personalities of graphic designs? *ACM Transactions on Graphics (TOG)*, *37*(4), 1-15.

Zhao, Z.-Q., Zheng, P., Xu, S.-t., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, *30*(11), 3212-3232.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. Proceedings of the IEEE conference on computer vision and pattern recognition,

Zhou, W., Mok, P., Zhou, Y., Zhou, Y., Shen, J., Qu, Q., & Chau, K. (2019). Fashion recommendations through cross-media information retrieval. *Journal of Visual Communication and Image Representation*, *61*, 112-120.

Zou, Z., Shi, Z., Guo, Y., & Ye, J. (2019). Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*.