Symmetry-Based Classification of Regular Textures based on filters bank and Random Forest

AOULALAY Ayoub[1], Abderrahim El MHOUTI[1], and Mohammed MASSAR[3]

[1] ISISA, FS, Abdelmalek Essaadi University, Tetouan, Morocco
[3]Analyse Non Linéaire Appliquée

E-mail address: aoulalayayoub@gmail.com, aelmhouti@uae.ac.ma, , massarmed@hotmail.com.

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Texture analysis, a vital component of computer vision and image processing, plays a pivotal role in fields such as decoration and art. This study focuses on the classification of regular textures into 17 distinct wallpaper patterns based on their symmetry operations. Utilizing computer vision techniques and a filter bank approach, we compared three methods: Gabor filter bank, CNN-trained filters, and ImageNet pretrained filters, in conjunction with a random forest model. The results revealed that ImageNet pretrained filters performed exceptionally well, achieving 87% accuracy in the 'wallpaper17' dataset and 81% in the 'wallpaper04' dataset.

Keywords: texture classification; wallpaper geometric groupes; filter bank; deep convolutional neural network

1. INTRODUCTION

Texture recognition is an important topic in the field of computer vision. Various schools of thought have attempted to define texture, and numerous definitions have been proposed. One of these definitions posits that texture can be defined as an arrangement of basic patterns according to a rule of repetition [3]. According to this definition, texture can be categorized into three major categories: regular texture, non-regular texture, and near-regular texture. Regular texture often includes human-made textures, such as art and decorative patterns, although it is rarely found in nature. Regular texture is characterized by the repetition of a basic motif to cover a plane. Regular texture falls into 17 classes based on the rules governing pattern repetition [2]. These classes are referred to as 'wallpaper patterns.

A wallpaper pattern is a two-dimensional design that repeats itself. It is created by replicating a fundamental motif using basic geometric transformations. These symmetry transformations include translation, rotation, mirror reflection, and glide reflection. It is believed that any wallpaper pattern in Euclidean space can be categorized into one of the 17 crystallographic classes based on its symmetries [1]. These classes were initially introduced by the mathematician Evgraf Fedorov in the late 19th century. Each wallpaper group is defined by its complex symmetry operations, and they are crucial for comprehending a wide range of man-made patterns, including art and decorative designs [9][10].

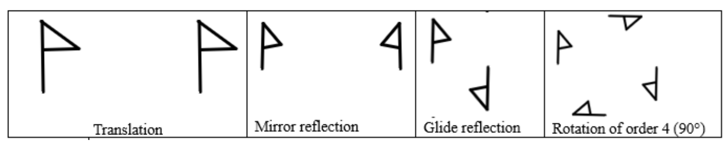

Figure 1: different type of symmetry operations.

The 17 wallpaper classes are commonly represented by a notation introduced by the French crystallographer Charles-Victor Mauguin. This notation employs four letters: 'p,' 'c,' 'g,' and 'm,' which describe the symmetries within the group. 'P' signifies a primitive pattern, which has the simplest lattice with lattice points only at the corners. 'C' indicates that the lattice has additional points at its center. 'M' denotes the presence of mirror reflections within a pattern. Lastly, 'G' signifies the existence of glide reflections. So These wallpaper groups are denoted as p1, p2 , p3, p3m1,p31m, p4, p4m , p4g , pm , pg, pmg , pgg , p6, p6m , cm , cmm and pmm. Where the number that is presented in some patterns indicates the highest order of rotational symmetry that is 1-fold, 2-fold, 3-fold, 4-fold or 6-fold[1]. The simplest pattern, denoted as P1, is generated using only two translations to cover the plane. P2 consists of a pattern generated using one translation and four rotations of order two. PM is generated using translations and mirror reflections with parallel axes. PG is generated using glide-reflections with parallel axes. Cm includes mirror reflections and glide-reflections where the axes of the mirror reflections are parallel. PMM, on the other hand, is generated with two reflections with perpendicular axes and a two-fold rotation. PMG combines a two-fold rotation, glide reflection with perpendicular axis, and a mirror reflection. PGG combines two rotations of order two and glide reflections with perpendicular axes, without mirror reflections. CMM combines a mirror reflection in two perpendicular axes and a two-fold rotation. P4 is generated with a rotation of order four. P4M has two four-fold rotations and mirror reflections in four axes. P4G includes rotations of order four, mirror reflections, and glide reflections with axes parallel to the mirror reflection axes. P3 contains rotations of order three (120°). P3M1 is a group that combines a three-fold rotation and mirror reflection. P31M is a group generated by a three-fold rotation, mirror reflection, and glide reflections, with the axes of the glide reflections midway between parallel mirror reflection axes. P6 contains a six-fold rotational axis. P6M has six-fold rotations, as well as rotations of order two and three. It also includes mirror reflections and glide reflections in six directions, with the axes of the glide reflections midway between parallel mirror reflection axes.

These 17 patterns can be further categorized into four major groups based on their fundamental properties. The first class is the 'Regular' class, which contains only one type of symmetry operation. This group includes P1, P3, and P6 classes. The second group is 'Semi-Regular,' which contains patterns generated with multiple types of symmetry operations in their unit cells. This group includes P2, P4, P3M1, and P31M. The third group is called the 'Dihedral Group,' which includes patterns that have only mirror reflection and translation operations in their unit cells. This group includes PMM, PMG, PGG, and CMM. The fourth group is the 'Miscellaneous Group' and contains the remaining wallpaper patterns: P4G, P4M, P6M, P6G, P3, and P2MG

In this work, the objective is to automatically recognize and classify wallpaper patterns based on their respective 17 groups and the four major categories to which they belong. While this is an important topic, particularly in the field of decoration, there have been few studies conducted in this area. To achieve this goal, supervised machine learning and deep learning techniques were employed. A dataset consisting of 170 images of wallpaper patterns was created. From this data, we created two datasets based on the challenge at hand. In the first dataset, the data is divided into 17 classes representing the wallpaper's symmetric groups. The other dataset has four classes representing the major categories into which wallpaper classes fall.

The recognition of wallpaper symmetry classes and regular textures can be approached from various angles. Some methods involve initially extracting the primitive pattern and subsequently identifying repetitions [1][6]. Another approach to addressing this challenge involves analyzing the overall appearance of the texture [7]. The 17 wallpaper classes represent textures that are distinctive and exhibit varying degrees of dissimilarity in appearance. Some classes may exhibit similarities to others, while some display significant differences. Effectively capturing these differences will be crucial for distinguishing between these classes. Motivated by this, we conduct the classification of wallpaper patterns based on their overall textural appearance.

It has been widely acknowledged that filter banks are highly suitable for addressing many challenges in texture recognition [4][5]. In this approach, a set of filters is generated, and these filters are then applied to the input image through convolution. The responses obtained from this operation are utilized as features for machine learning and deep learning models. Filters designed for texture analysis can capture valuable information, including details about the directionality and frequency of repetitive patterns, while disregarding irrelevant block information. This makes filter banks a valuable tool for texture description

In this study, with the aim of determining the class of wallpaper patterns, we propose the use of a filter bank approach for describing regular textures, the features extracted from the filter bank will feed a random forest classifier [28]. To describe texture, we undertake and compare three strategies. The first strategy involves employing a handcrafted Gabor filter bank [8], which is widely used in the literature. Gabor filter banks can capture both the orientation and frequency of a texture, rendering them highly effective tools for distinguishing among the 17 wallpaper patterns. this approach proved to be very efficient achieving a very good accuracy.

The second strategy involves using filters trained from the constructed dataset. This is done by training a shallow CNN on top of the dataset and then using the convolutional layers as feature descriptors. This approach achieves good performance; however, the design of the CNN should differ from one task to another, making it non-generalizable.

To overcome this problem, another CNN-based approach was conducted. Instead of using filters pretrained on our dataset, ImageNet pretrained filters are used. it is believed that the middle layers of a CNN pretrained on ImageNet are capable of capturing rich information about texture, as texture is a mid-level abstract feature. This approach proved to be efficient on both datasets, achieving very good accuracy.

The level of challenge presented by both datasets varies, and it has been observed that the performance of classifying wallpaper patterns into 17 symmetry groups is better than classifying them into the four major categories. The best accuracy was achieved by the ImageNet pretrained filter bank approach on both datasets, with an 88% accuracy in classifying the wallpaper into 17 classes and an 81% accuracy in classifying them into the four major groups.

In this study, the aim is to advance the recognition and classification of wallpaper patterns based on their symmetry and textural properties. By exploring various approaches, including filter banks, we seek to contribute to the field of texture recognition and provide valuable insights for applications in decoration, design, and beyond.

The remaining sections of this paper are organized as follows: the 2nd section investigates the related works covering some of the recent studies relating to regular textures classification using computer vision. The 3rd section looks at the material and methodology design. This section states the context of the recognition and classification of wallpaper patterns based on their respective 17 groups and the four major categories to which they belong. The 4th section discusses and includes an analysis of the expected results. The last section concludes the study and outlines the future works.

2. RELATED WORKS

Texture classification is a robust field with numerous works dedicated to material classification [20], texture description [19], and medical image analysis based on texture [11], among others. Regular texture recognition is a highly significant topic, and it has been thoroughly investigated by numerous researchers. Regular texture can be defined as an arrangement of repetitive basic motifs following strict rules. It has often been studied in the analysis of decorative images [17] and the examination of fabric defects [16].

The process of analyzing regular textures often involves the extraction of the basic pattern, followed by the definition of the repetition rule. The work by [12] utilizes forward differences in the superposition of distance matching functions to determine the size of basic motifs in fabric textures. This method was employed in the process of detecting fabric defects. DMF (Distance Matching Functions) was also used in [13], in conjunction with Haar wavelets for periodicity detection. In [14], the analysis relies on the peaks of the autocorrelation function to determine regular texture primitives and extract periodicity. The peaks of the autocorrelation function are also employed, in combination with genetic algorithms, in the research conducted in [15] to extract basic repetitive motifs in Islamic geometric patterns. Gabor filters are employed in [18] to detect fabric defects in regular textures belonging to wallpaper symmetric groups. The output of Gabor wavelets is divided into blocks, which are used for the automated detection of defects. The approaches undertaken in these studies all analyze textures from a local-to-global perspective. In these approaches, the initial step involves locating the primitive pattern and subsequently analyzing its distribution to determine the repetition rule. Another approach to tackle this challenge is to analyze the overall appearance of the texture. This is driven by the observation that each class of regular texture within the wallpaper group possesses a unique appearance, which can be comprehensively described using texture descriptors.

Filter bank is widely used technique in texture recognition and segmentation. This approach has many advantages over other approaches as it can captures various information about the content of the image. It was also shown that filter bank can capture orderless features that represents most the texture of an image. the author in [25] showed that CNN is like bank of filters with increasing complexity going deeper with the network. they exploit this propriety by developing a CNN that focuses more on texture information. Their idea was that overall shape information extracted by the fully connected layers of a classic CNN is of minor importance in texture analysis. The complexity of the features trained by CNNs increases with the depth of the network. Therefore, the last convolution layer extracts complex features which respond to objects such as a nose, a face or a human body. The fully connected layers use the response to these features to obtain information about the overall shape of the image and calculate a probability distribution over the different classes in the last fully connected layer. This design is suitable for exploring the arrangement of less complex features from the previous layers and their sparse spatial response for an object recognition scheme. In the work [29], Gabor filters are employed to automatically detect urban and tree features in aerial and LIDAR images. To achieve this, a set of Gabor filters is created by adjusting two parameters of the Gabor function: the standard deviation and the orientation. A thresholding operation is applied to the Gabor filter responses to obtain segmented images that separate trees and urban features from the rest of the images.

The study presented in [30] has demonstrated the effectiveness of deep convolutional layers in Convolutional Neural Networks (CNN) as robust texture descriptors when used as a filter bank. These deep layers are combined with traditional texture encoders such as the bag of visual words and Fisher vectors, yielding promising results. In [31], the authors utilized a Gabor filter bank for texture segmentation. To enable highly efficient feature extraction, they maximized the frequency variance in the Gabor function, capturing rich and distinctive features.

The primary focus of the work [32] was regular textures. To begin, they created a new dataset specifically for regular textures. Following this, they tested several algorithms for texture classification. These algorithms ranged from fine-tuning pretrained models to utilizing Fisher vectors encoders on top of the convolutional layer filters. These approaches achieved excellent performance in the classification of regular textures.

In this work, we conduct a three-strategy process for regular texture recognition. We were inspired by these studies to

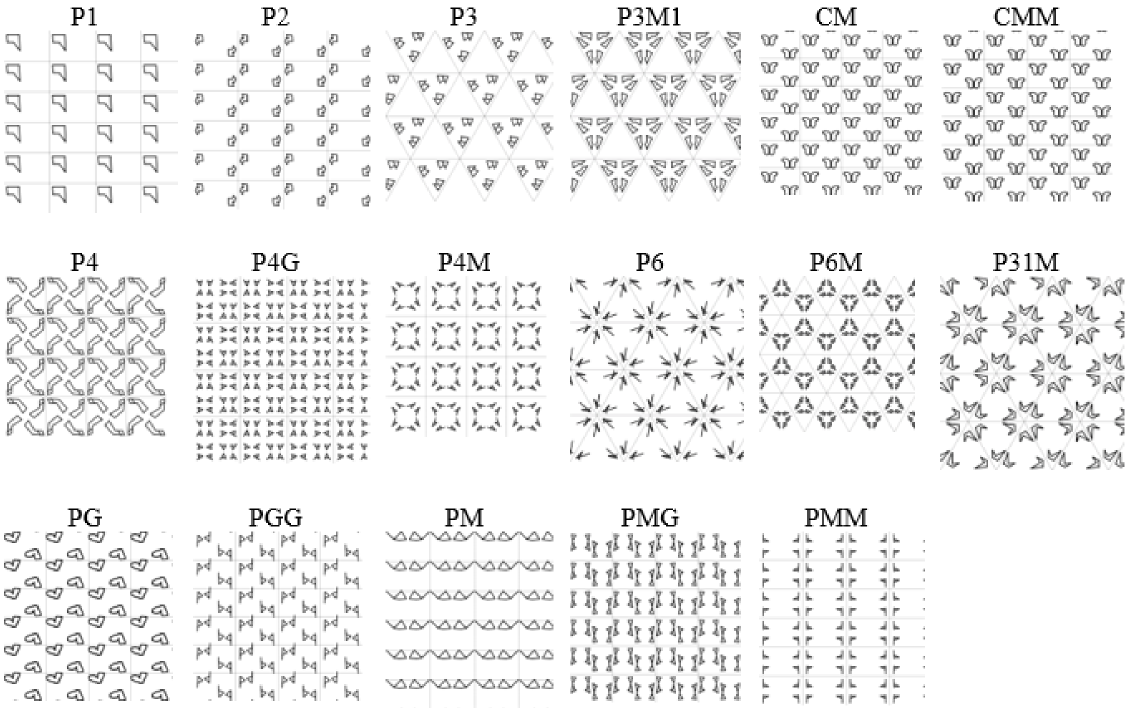

Figure 2: example of patterns belonging to 17 wallpaper classes

use filter bank approach as we think it the best suitable to tackle texture classification challenges. we used Gabor filter bank, CNN filters, and ImageNet filters to address the challenge of the classification of regular texture based on their symmetry, into corresponding wallpaper geometric groups

3. MATERIAL AND METHODS

A. data

To classify wallpaper geometric patterns, a dataset comprising seventeen (17) distinct wallpaper classes, each of which contains 100 images is utilized. These images were generated using the website [21], ensuring high quality with no noise or lighting discrepancies. such high-quality numerical images are opted to solely focus on the task of classifying 2D geometric patterns into their respective 17 wallpaper classes, without distractions from other challenges such as pattern localization, noise, scale, and viewpoint variations.

From the same pool of collected images, two datasets were created. The first one encompasses the aforementioned seventeen wallpaper classes, while the second comprises the four major categories within these classes (as previously explained). figure 2 represent samples from each class.

To comprehend the distribution of classes in the two datasets, t-distributed stochastic neighbor embedding (t-SNE) is employed.

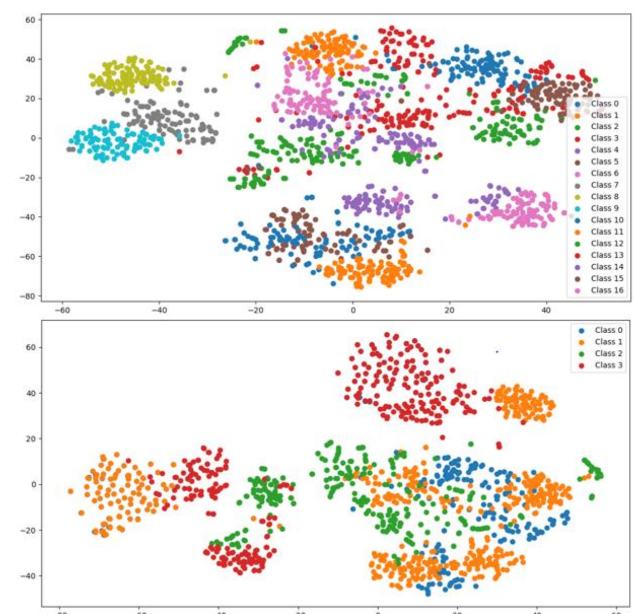

Figure 3: T-SNE visualization of the datasets the upper one represents the WAMMPAPER17, the bottom is for WALPAPER04

T-SNE [33] is a dimensionality reduction machine learning algorithm often utilized to visualize high-dimensional data in a lower dimension. The advantage T-SNE holds over other dimensionality reduction techniques is its focus on preserving the relationships between data points. Similar data in a higher-dimensional space is represented by closely positioned points in the lower-dimensional space. The distribution of our two datasets using T-SNE is illustrated in figure 3. Observing the figure, it's evident that for the WALLPAPER04 dataset, the classes overlap, and some classes lack density in a particular region, indicating a lack of inter-class similarity in the initial data. The same trend is observed in the WALLPAPER17 dataset, where the majority of classes overlap. All these findings imply that classifying these two datasets is challenging, requiring the utilization of perfect features to separate the classes.

B. Filters bank

A bank of filters is a transform-based approach widely utilized in texture classification literature. This method involves creating a set of various filters, each designed to extract specific information from the image. Each filter is convolved with the image, resulting in a response that contains information about the image's content, such as edges, corners, and spatial frequencies. The bank of filters extracts a multitude of details from the image and combines these responses to describe the image comprehensively. In the field of texture recognition, the bank of filters is one of the most popular descriptors.

In this work, our objective is to classify wallpaper group patterns using a filter bank. To achieve this, we employ and compare three different approaches: Gabor filter bank, CNN filters and CNN's pre-trained filters on ImageNet. We will delve into the details of each of these methods in the following sections.

Gabor filters

The Gabor filter, named after Nobel laureate Dennis Gabor, stands as one of the prominent methods for texture analysis. The fundamental concept behind this approach involves computing the local feature transform. This method comprises creating a bank of filters generated by the Gabor function. The Gabor function itself consists of two essential components: a sinusoidal function modulated by a Gaussian window [8]. The sinusoidal part determines the orientation, while the Gaussian component assigns the weight to the filter. The popularity of the 2-D Gabor filter has grown because it mimics the visual system of mammals in analyzing and extracting complex patterns. It responds to patterns at specific frequencies and orientations.

In texture feature extraction, the Gabor filter isolates texture based on frequency and orientation [22], utilizing various parameters as indicated in the following equation

$$G(x, y, \sigma, \theta, \lambda, \gamma, \varphi) = \exp\left[-\frac{x'^2 + y'^2\,\gamma^2}{2\sigma^2}\right] . \exp\left[i\left(2\pi\frac{x'}{\lambda} + \varphi\right)\right] \quad (1)$$

Where $x' = x\cos\theta + y\sin\theta$ and $y' = x\sin\theta + y\cos\theta$

σ describe the standard deviation of the Gaussian function and it controls the width of the Gaussian window, γ is the aspect ratio, θ controls the direction of the filter. λ is the wavelength of the sinusoidal factor and φ represents the phase offset.

$$O = G(x, y) * I(i, j) \quad (2)$$

An input image is convolved with a Gabor filter, resulting in a response that captures texture with a specific frequency and orientation while blocking out other textures. To account for various scenarios, a bank of filters is constructed by altering the parameters of the Gabor function G(x, y, σ, θ, λ, γ, φ). Different combinations of these parameters can characterize distinct textures within an image.

An example illustrating the creation of different filters by manipulating the parameters of the Gabor function is presented in figure 4. As depicted, changes in σ, θ, λ, and γ yield different filters. This set of filters serves as the descriptor used for extracting texture features.

CNN filters

one approach to extracting features from texture images is through the use of CNN descriptors, which involve utilizing the convolutional layers of Convolutional Neural Networks (CNNs). CNNs have gained popularity in recent years for various computer vision tasks, including texture recognition. One key advantage of CNNs over traditional methods is their ability to learn descriptors.

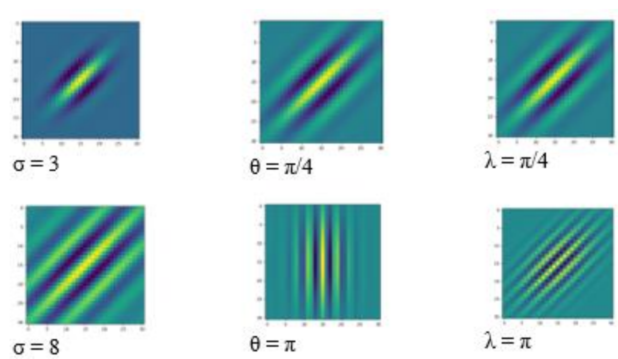

Figure 4: illustration of the impact of changing different parameters in the gabor function.

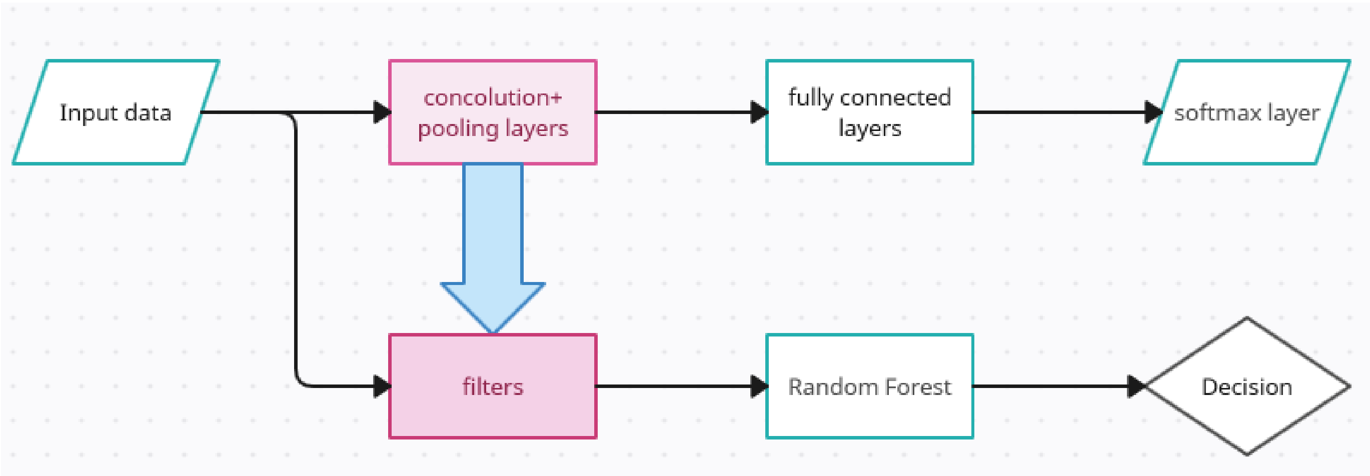

Figure 5: CNN filter bank approach, the weights of the filters are obtained from training a CNN in the top of the data.

This means that CNNs do not require meticulous design of image descriptors; instead, they autonomously construct their descriptors based on the available data. These descriptors are learned from the data through backpropagation and gradient descent [23], making them well-suited for the problem at hand.

Another advantage of CNNs over traditional methods is their hierarchical nature of feature extraction.

A CNN can generally be divided into two parts: the feature extraction part, which learns suitable descriptors for the given problem, and the classification part, which utilizes the extracted features to make decisions. The feature extraction part consists of a series of layers, each containing a set of filters. Each layer extracts features and constructs a feature map, which is then passed to the next layer of filters for further feature extraction. This process continues hierarchically until the final layer is reached. This hierarchical approach means that the deeper a layer is in this process, the more abstract the features become [24]. For example, the first layer extracts low-level features such as edges and color blobs, while intermediate layers extract more complex features like textures, and the last layer extracts features that are specific to the problem, becoming increasingly abstract.

However, despite the power of CNNs, they have some significant drawbacks that need to be addressed and rectified. One such drawback is their data dependence. As mentioned earlier, CNNs learn descriptors directly from the data, and constructing a robust CNN often necessitates a substantial number of trainable parameters. This, in turn, requires access to a significant volume of training samples. In many instances, including our own, only a limited amount of data is available.

The convolutional layers of CNNs are believed to extract orderless features that are well-suited for texture recognition challenges [25]. These convolutional layers function as filter banks, and research has indicated that the initial layers act as Gabor-like filter banks [29]. In this study, the aim is to delve deeper into this assumption within the context of the demanding task of classifying wallpaper group patterns. this challenge is approached from a texture perspective, where we consider the global appearance and differences among wallpaper group patterns. Consequently, two CNN approaches were employed to tackle this problem, considering the constraints posed by limited data.

The first approach is illustrated in figure 5, it involves utilizing the CNN filters trained on the wallpaper dataset. Essentially, the CNN was trained through gradient descent and backpropagation using the wallpaper dataset, and then these trained filters were employed to extract features from the dataset, which are subsequently fed into a machine learning classifier. To adapt the CNN to the limited data available, we opt for a shallow CNN. It has been demonstrated that texture features are predominantly extracted in the initial layers [24]. After conducting a series of experiments, the best results were achieved using a CNN with few convolutional layers, surpassing even deeper architectures.

In the initial step, a CNN is trained on both datasets. After several experiments two CNN models were used depending on each performance on each dataset.

architecture with only a few parameters. It has been demonstrated that texture features are predominantly.

for ‘wallpaper04’ dataset The CNN comprises three convolutional layers stacked on top of each other. Max pooling is deferred until the last layer, following the approach outlined in [29], to maximize performance

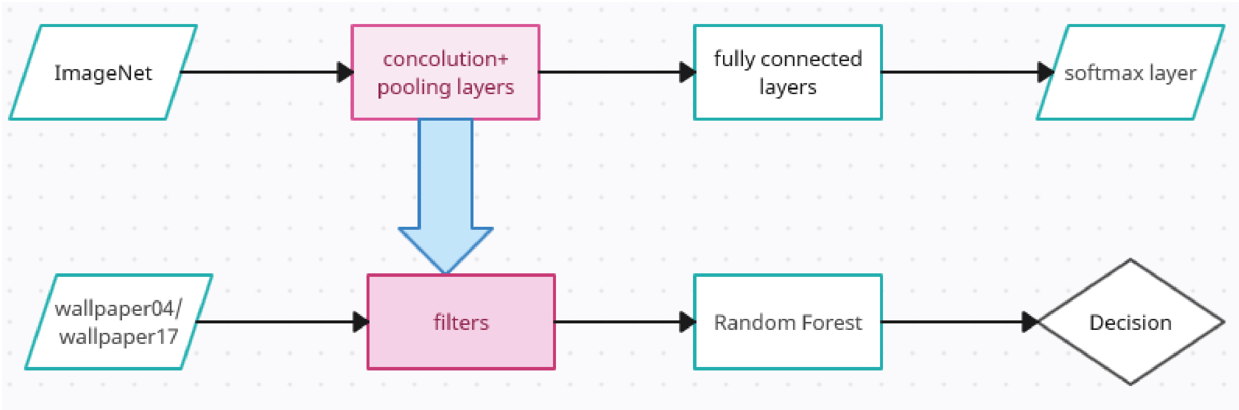

Figure 6: ImageNet pretrained filter bank approach, the weights of the filters are obtained from training a CNN in the top of the ImageNet dataset.

. The first layer employs filters with a receptive field of 5x5, which has been shown to be more effective than 3x3 or 7x7 filters. Meanwhile, the second- and third-layers use 3x3 filters to reduce the number of parameters. The extracted features are then passed to a softmax layer and trained for 30 epochs using the 'rmsprop' optimizer. Data augmentation techniques are applied to mitigate overfitting.
For 'wallapaper17' dataset. The CNN depicted in the figure employs a 4-layer architecture with 32 filters in the first two layers and 64 filters in the last two layers. Batch normalization [34] is applied after each convolutional layer, and max pooling follows every two layers. Sigmoid serves as the activation function for the network. The network undergoes training for a duration of 60 epochs.
After training, the features extracted by the convolutional layers are input into a random forest classifier for classification.

ImageNet pretrained filter bank.

The second strategy (presented in figure 6) involves using filters that have been trained on a large-scale dataset, a commonly referred-to approach in the literature as transfer learning. Transfer learning entails leveraging knowledge from a model trained on a dataset with abundant data to enhance a model's performance in a task where data is limited [26]. This approach is analogous to human experience, where, for example, an individual proficient in playing one musical instrument (such as the piano) can relatively quickly learn to play another musical instrument (like the violin) compared to someone with no prior musical expertise. In the context of image classification, models trained on ImageNet are frequently employed, given their numerous advantages.
ImageNet boasts an extensive array of classes encompassing various objects, ranging from fine-grained to coarse-grained categories. Furthermore, ImageNet comprises an extensive image collection, totaling 1.3 million images, which makes it highly suitable for training deep and wide CNN models. This abundance of data contributes to the transferability of knowledge gained from training these models on ImageNet, extending to diverse domains, including texture classification. In this study, VGG16 network [27] is chosen for its simplicity.
The rationale behind selecting this model lies in its simplicity, as the focus is primarily on the filter bank provided by this model. The VGG16 model, depicted in the figure, consists of 16 convolutional layers grouped into 5 blocks. We leverage features from the third middle layer, specifically the third block, as previous research has indicated that texture features reside at an intermediate level within the network. Our experiments also substantiate this hypothesis.

Random forest:

After the feature extraction, the classification stage is performed. In this work, we use a Random Forest classifier [28]. Random Forest, as its name indicates, is composed of a set of decision trees. Each tree is constructed from the data and is used to make decisions. The final decision is determined by averaging all the outputs or by a majority vote. Random Forest enhances the classification capability by randomly assigning input data to each tree.

4. Results and Discussion

In this section, the key results of the experiment conducted to classify wallpaper geometric patterns into their respective wallpaper geometry groups are presented. The wallpaper patterns are classified into the 17 wallpaper groups and the four major groups, as explained previously. To achieve this, bank of filters approach is employed to extract texture from images. These features were then input into a random forest classifier for the final classification.

Three different banks of filters-based approaches were conducted in this experiment: the Gabor filter bank, CNN filters, and ImageNet pretrained filters. The dataset was divided into an 80% training set and a 20% testing set for evaluating classification accuracy.

- Gabor filter results.

As explained in the 'Materials and Methods' section, the Gabor filter is a highly effective tool for texture recognition. In this study, a filter bank is built by manipulating the parameters of the Gabor function. This set is created using various combinations of Gabor function parameters.

TABLE 1: the accuracy of Gabor filter bank approach in %

Dataset	Accuracy
Wallpaper17	**83**
Wallpaper04	78

While Gabor filters are a traditional method for texture recognition, they yield impressive results in classifying the challenging task of wallpaper pattern classification, achieving 83% accuracy in the wallpaper17 dataset and 78% accuracy in the wallpaper04 dataset. The variation in results reflects the complexity of the task, with the wallpaper04 classes exhibiting high intra-class variance. However, achieving a 78% accuracy rate is still remarkable. Gabor filters successfully capture the frequency, orientation, and repetition rules of repetitive patterns within the wallpaper group, which explains their strong performance in this task.

- CNN filter bank

For this approach , CNN filters trained on top of two datasets are utilize. After conducting numerous experiments, we have determined that two CNN models produce the best results, as explained in the 'Materials and Methods' section. The first model, CNN-17, designated for the wallpaper17 group, is a shallow CNN comprising four convolutional layers with 16, 16, 32, and 32 filters, respectively. This CNN undergoes training for 60 epochs. The second model consists of three convolutional layers with 16, 16, and 32 filters and is trained for 30 epochs. The reason for employing different CNN models for each dataset, despite having identical training samples, is because they address distinct problems with varying complexities, class inter-variance, and differing numbers of samples per class. The output from the final convolutional layer is used to extract features, which are subsequently input into a random forest model. The results are presented in the table.

TABLE 2: the accuracy of CNN filters approach in %

Dataset	CNN-17	CNN-04
Wallpaper17	**88**	23
Wallpaper04	20	**77**

From the table, it is evident that each CNN is specifically designed for its respective problem, and it performs poorly when applied to the other problem, despite some similarities between the two tasks and the fact that they share the same set of images, albeit divided into different classes. This presents a significant limitation, especially considering that CNN features are expected to be generalizable and transferable across various tasks. These results can be attributed to the limited size of the dataset and the inherent complexity of the problems at hand. Achieving better generalization would likely require a much larger training dataset.

- ImageNet pretrained filters

This method involves utilizing filters from a model that was trained on ImageNet. We employ the VGG16 model for its simplicity and the fact that it comprises five blocks of convolutional layers. Consequently, we can readily assess the influence of feature levels, ranging from low-level to high-level, as illustrated in the table.

TABLE 3: the accuracy of extraction features from different blocks of VGG16.

VGG block	Wallpaper17	Wallpaper04
Block2	86	76
BLOCK3	**88**	**81**
BLOCK4	84	73
BLOCK5	53	67

The best results were achieved by employing the filters from the first, second, and third blocks of the VGG16 network, which correspond to the features of the 8th layer. These features can be considered as being at a middle level of abstraction. This outcome can be explained by the observation that, in deeper networks, features become increasingly abstract. Early layers focus on extracting basic features like edges and corners, while later layers capture more task-specific features. The intermediate layers, in contrast, tend to extract middle-level features, such as textures.

The best results are achieved with 88% accuracy in the wallpaper17 classification and 81% in the wallpaper04 dataset, indicating the varying complexity of the problems. The pretrained filters approach outperforms the traditional Gabor filter approach by a slight margin, demonstrating that the traditional approach still has relevance in the field. Additionally, the pretrained filters outperform the CNN filters approach in the wallpaper04 dataset by 9%, while they perform at the same level in the wallpaper17 dataset. This suggests that pretrained filters are more suitable as the complexity of the problem increases, with a particular advantage in their ability to generalize. Pretrained filter approaches utilize the same filters for both datasets, unlike the CNN filter approach, where each problem requires a carefully designed CNN architecture, resulting in distinct solutions for each problem.

5. CONCLUSION

Regular texture recognition holds significant importance in the field of texture analysis, finding applications in various domains such as decoration and art. All regular textures can be categorized into 17 distinct wallpaper patterns based on their symmetry operations. In this work, computer vision approaches are employed to automatically classify these wallpaper patterns. This challenge is approached from two angles: first, by classifying patterns into each of the 17 wallpaper symmetric groups, and second, by grouping them into four major categories that share fundamental elements. For this purpose, two datasets comprising 170 wallpaper patterns are constructed. To facilitate classification, we adopted the filter bank approach, which is considered most suitable for this task. We compared three filter bank-based methods: Gabor filter bank, CNN-trained filters, and ImageNet pretrained filters. These methods were combined with a random forest model for classification. The best results were achieved using ImageNet pretrained filters, achieving 87% accuracy in the 'wallpaper17' dataset and 81% in the 'wallpaper04' dataset.

Several important conclusions were drawn, such as the continued effectiveness of traditional approaches like Gabor filters, which consistently produce impressive results. Furthermore, filters from a CNN pretrained on ImageNet not only deliver high performance but also offer stability and generalization when compared to CNN filters trained on shallow datasets. However, it's essential to note that the results obtained in this work apply to images with perfect conditions (with no noise or illumination differences) and when the texture fills the entire image. This can be seen as a foundational step, addressing real-world scenarios where images are not ideal, and patterns are more complex.

In future work, we intend to enhance the dataset by increasing the number of samples and introducing greater challenges. This effort will help establish a baseline for a problem that is often overlooked.

References

1. Asha, V., Nagabhushan, P., & Bhajantri, N. U. (2012). Automatic extraction of texture-periodicity using superposition of distance matching functions and their forward differences. Pattern Recognition Letters, 33(5), 629-640.
2. Schattschneider, D. (1978). The plane symmetry groups: their recognition and notation. The American Mathematical Monthly, 85(6), 439-450.
3. Tamura, H., S. Mori, and Y. Yamawaki, “Textural Features Corresponding to Visual Perception,” IEEE Transactions on Systems, Man, and Cybernetics, SMC-8, pp. 460-473, 1978.
4. Randen, T. (1997). Filter and filter bank design for image texture recognition.
5. Cimpoi, M., Maji, S., & Vedaldi, A. (2015). Deep filter banks for texture recognition and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3828-3836).
6. 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.
7. Aoulalay, A., El Mhouti, A., & Massar, M. (2022, March). Classification of Islamic geometric patterns based on machine learning techniques. In 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET) (pp. 1-6). IEEE.
8. Turner, M. R. (1986). Texture discrimination by Gabor functions. Biological cybernetics, 55(2-3), 71-82.
9. Grünbaum, B., Grünbaum, Z., & Shephard, G. C. (1986). Symmetry in Moorish and other ornaments. In Symmetry (pp. 641-653). Pergamon.
10. Albert, F., Gomis, J. M., Blasco, J., Valiente, J. M., & Aleixos, N. (2015). A new method to analyse mosaics based on Symmetry Group theory applied to Islamic Geometric Patterns. Computer Vision and Image Understanding, 130, 54-70.
11. Collewet, G., Strzelecki, M., & Mariette, F. (2004). Influence of MRI acquisition protocols and image intensity normalization methods on texture classification. Magnetic resonance imaging, 22(1), 81-91.
12. Asha, V., Nagabhushan, P., & Bhajantri, N. U. (2012). Automatic extraction of texture-periodicity using superposition of distance matching functions and their forward differences. Pattern Recognition Letters, 33(5), 629-640.
13. Asha, V., Bhajantri, N. U., & Nagabhushan, P. (2013). Periodicity Extraction using Superposition of Distance Matching Function and One-dimensional Haar Wavelet Transform. arXiv preprint arXiv:1311.3808.
14. 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.
15. Nasri, A., Benslimane, R., & El Ouaazizi, A. (2014, November). A genetic based algorithm for automatic motif detection of periodic patterns. In 2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems (pp. 112-118). IEEE.
16. Hamdi, A. A., Sayed, M. S., Fouad, M. M., & Hadhoud, M. M. (2018, February). Unsupervised patterned fabric defect detection using texture filtering and K-means clustering. In 2018 international conference on innovative trends in computer engineering (ITCE) (pp. 130-144). IEEE.
17. Aoulalay, A., El Mhouti, A., & Massar, M. (2022, March). Classification of Islamic geometric patterns based on machine learning techniques. In 2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET) (pp. 1-6). IEEE.

18. NU, B. (2011). Automatic detection of texture defects using texture-periodicity and Gabor wavelets. In International conference on information processing (pp. 548-553). Springer, Berlin, Heidelberg.
19. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., & Vedaldi, A. (2014). Describing textures in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3606-3613).
20. Bell, S., Upchurch, P., Snavely, N., & Bala, K. (2015). Material recognition in the wild with the materials in context database. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3479-3487).
21. https://math.hws.edu/eck/js/symmetry/
22. Recio, J. A. R., Fernandez, L. A. R., & Fernández-Sarriá, A. (2005). Use of Gabor filters for texture classification of digital images. Física de la Tierra, 17, 47.
23. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
24. Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13 (pp. 818-833). Springer International Publishing.
25. Andrearczyk, V., & Whelan, P. F. (2016). Using filter banks in convolutional neural networks for texture classification. Pattern Recognition Letters, 84, 63-69.
26. Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.
27. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
28. Rigatti, S. J. (2017). Random forest. Journal of Insurance Medicine, 47(1), 31-39.
29. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:1602.07360. 30. CIMPOI, M., MAJI, S., & KOKKINOS, I. Deep filter banks for texture recognition, description, and segmentation. arXiv, 2015: 1507.02620 [2020-11-03].
31. Bresch, M. (2002). Optimizing filter banks for supervised texture recognition. Pattern recognition, 35(4), 783-790.
32. Liu, N., Rogers, M., Cui, H., Liu, W., Li, X., & Delmas, P. (2022). Deep convolutional neural networks for regular texture recognition. PeerJ Computer Science, 8, e869.
33. Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of machine learning research, 9(11).
34. Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning (pp. 448-456).