electronics

MDPI

Article

Graph- and Machine-Learning-Based Texture Classification

Musrrat Ali [1,*], Sanoj Kumar [2,*], Rahul Pal [3], Manoj K. Singh [4] and Deepika Saini [5]

[1] Department of Basic Sciences, PYD, King Faisal University, Al Ahsa 31982, Saudi Arabia
[2] School of Computer Science, UPES, Dehradun 248007, Uttarakhand, India
[3] Department of Mathematics, UPES, Dehradun 248007, Uttarakhand, India
[4] School of Computer Science Engineering & Technology, Bennett University, Greater Noida 201310, Uttar Pradesh, India
[5] Department of Mathematics, Graphic Era (Deemed to be) University, Dehradun 248002, Uttarakhand, India
* Correspondence: mkasim@kfu.edu.sa (M.A.); sanoj.kumar@ddn.upes.ac.in (S.K.)

Abstract: The analysis of textures is an important task in image processing and computer vision because it provides significant data for image retrieval, synthesis, segmentation, and classification. Automatic texture recognition is difficult, however, and necessitates advanced computational techniques due to the complexity and diversity of natural textures. This paper presents a method for classifying textures using graphs; specifically, natural and horizontal visibility graphs. The related image natural visibility graph (INVG) and image horizontal visibility graph (IHVG) are used to obtain features for classifying textures. These features are the clustering coefficient and the degree distribution. The suggested outcomes show that the aforementioned technique outperforms traditional ones and even comes close to matching the performance of convolutional neural networks (CNNs). Classifiers such as the support vector machine (SVM), K-nearest neighbor (KNN), decision tree (DT), and random forest (RF) are utilized for the categorization. The suggested method is tested on well-known image datasets like the Brodatz texture and the Salzburg texture image (STex) datasets. The results are positive, showing the potential of graph methods for texture classification.

Keywords: texture classification; horizontal visibility graph; natural visibility graph; feature extraction; image natural visibility graph; classifiers; machine learning

Citation: Ali, M.; Kumar, S.; Pal, R.; Singh, M.K.; Saini, D. Graph- and Machine-Learning-Based Texture Classification. *Electronics* **2023**, *12*, 4626. https://doi.org/10.3390/electronics12224626

Academic Editor: Hyunjin Park

Received: 11 October 2023
Revised: 8 November 2023
Accepted: 9 November 2023
Published: 12 November 2023

1. Introduction

Texture has useful features that can be used to research various image processing and machine vision tasks, including image retrieval, classification, segmentation, and synthesis. Texture analysis is still a difficult subject because real textures are not always easy to understand. In computer vision, "texture" can mean a lot of different things. One idea says that texture comprises changes in visual sharpness that make patterns that appear over and over again [1]. The surface's physical characteristics might be to blame for these patterns, which reflect different wavelengths of light. Smooth surfaces project light at a predetermined angle, while uneven surfaces reflect light everywhere. Humans can recognize texture, but automatic systems sometimes need specialized computational approaches. The way pixel brightness and intensity shift across a picture defines texture in image analysis [2]. The term "texture" is commonly employed as the principal adjective to characterize the basic elements of an image. Texture analysis plays a crucial role in various computer vision applications, including object recognition, surface defect detection, pattern recognition, and medical image analysis. Texture analysis is a fundamental component of computer vision, wherein the initial step involves extracting pertinent data from an image in order to characterize its texture. Various methods and strategies are employed in this operation [3]. Any number that describes the texture of an image should not be affected by changes in scale, rotation, or any other smooth change in pixel strength. Because changes in rotation, translation, or scale do not affect how people see the texture, this is the case.

Obtaining the features and choosing the classifier are the two most important parts of texture classification.

Several methods for extracting features have been created in the last few decades, and many attributes have been studied to better describe the texture information in an image [4]. A number of old and new techniques have been used to do this, including the gray-level co-occurrence matrix (GLCM) [5], Haralik descriptors [6], wavelet transform [7], Markov random fields [8], Gabor texture discriminator [9], local binary patterns (LBP) [10], and bioinspired texture descriptor (BiT) [11]. In computer vision, natural language processing, pattern identification, speech recognition, picture recognition, self-driving cars, and recommendation systems, deep learning has a wide range of uses. One deep learning model in particular, the convolutional neural network (CNN) model, has excelled at pattern recognition and image classification. It can learn attributes from datasets without human interaction [12]. An alternative approach combines a CNN with handmade features from standard methods [13,14]. Thus, experts have started studying compact and reliable texture descriptors.

A modified visibility graph (VG) was used to classify a texture dataset in this work. The degree distribution was obtained using the image natural visibility graph (INVG) and image horizontal visibility graph (IHVG) [15] and fed into a classifier for classification. This study processed images using the Lacasa et al. [16] approach for the image visibility graph (IVG). The degree distribution and clustering coefficients using the IHVG and INVG were found to be essential for capturing minor graph substructures [16]. The primary contributions of this study are delineated as follows:

- This study employed a modified visibility graph to classify a texture dataset using a graph-based classification method.
- The degree distribution information extracted from the IHVG and INVG was used as the input for a specialized classifier.
- This analysis primarily focused on capturing minor substructures by examining the clustering coefficient with the degree distribution for the IHVG and INVG.

In Section 2, a quick and concise analysis of texture feature extraction using the conventional approach and convolution neural networks is presented. In Section 3, several strategies for representing time series as graphs are discussed, and the concept of complex networks is introduced briefly. Section 3.3 elaborates on the NVG and HVG in images, also known as the image natural and the image horizontal visibility graph, and on the procedure for generating graphs from images. Section 3.4, which also includes methods for feature extraction, briefly discusses the degree distribution and clustering coefficient of a graph. Section 3.5 discusses the standard classifiers that were used in this investigation, such as SVM, DT, KNN, and RF. The results, suggestions, and plans for future work are explained in Section 4. There is a thorough flowchart of the algorithm and an analysis of the standard texture dataset that was used in the study.

2. Related Work

This section summarizes the background of texture classification techniques, including both classical approaches and deep learning applications.

2.1. Texture Classification Based on Traditional Methods

Several feature extraction approaches have been discovered and applied to texture classification challenges. The GLCM introduced by Haralicket et al. [6] is an old texture categorization feature extraction approach. The texture information generated by GLCM is dependent on grayscale, kernel size, and direction. Haralick developed fourteen texture elements that offer spatial context information, such as contrast, homogeneity, dissimilarity, angular second moment, energy, and correlation [5,6,17–19]. Following GLCM, the most commonly used LBP is used to extract textural features [20]. Ojala and Pietikainen presented the original LBP in 1999 [21] based on the statistical distribution of the uniform local binary pattern rotation and histogram equalization invariants [22]. The MRF statistical feature

extraction approach is employed for pixel or other spatial correlation properties [8]. Wavelet-transform [23] divides pictures into four sub-bands. These four sub-bands indicate the finest scale wavelet coefficient, which we divide into four parts again to create the coarse level coefficient, and so on, until we reach the ultimate scale [7]. In Table 1, we summarize the accuracy of classical approaches applied to various texture datasets.

Table 1. Summary of texture classification based on traditional methods.

References	Purpose	Features	Model	Dataset	Accuracy (%)
[5,24]	Image texture classification	Statistical features, correlation	GLCM	Brodatz texture images	99.043
[7]	Texture classification	Wavelet statistical features	Wavelet transform	VisTex image dataset	Mean-97.80
[10]	Texture classification	SNELBP features	SNELBP	KTH-TIPS	95.97
[1]	Texture characterization	GLCM Haralick	CatBoost classifier	Outex	99.30 99.84%
[1]	Texture characterization	GLCM Haralick	CatBoost classifier LDA classifier	KTH-TIPS	92.01 98.35
[8]	Texture classification	Order statistics, histogram of configuration	Markov random field	Brodatz texture database	87
[9]	Color texture classification	Local class	GaRCIA	VisTex	Train-98.4 Test-89.2
[11]	Texture descriptor quantify	BiT descriptor	SVM classifier	Salzburg Outex KTH-TIPS	92.33 99.88 97.87

2.2. Texture Classification Based on CNN

CNNs are extensively employed in the domain of image classification. To separate the texture image, its ability to capture high-level features is crucial [25]. CNNs collect m on basic visual properties in the first convolution layer, then use this information to construct more complex features in the deepest layer [12]. The effectiveness of CNN algorithms is summarized in Table 2 when used on various texture datasets.

Handcrafted texture classification uses feature extraction methods like GLCM, MRF, LBP, wavelet transform, and BiT for classifiers like SVM or KNN. The basic method extracts and categorizes features with a few lines of code and is simple to use. Traditional methods cannot be scaled to large datasets because they look at every image separately and pull out traits from each one, which takes time and resources. CNNs learn properties from images directly and outperform older methods on complex datasets. However, CNN models demand large datasets and are hard to identify with categorization features. The visibility graph is generated using time series or image data, and then features are selected based on difficulties utilizing graph properties. Graphs have global and local features. Lacasa et al. [16,26] suggest that visibility graphs can help with image processing and classification because they show small parts of a graph's structure, while global features show how the graph is put together as a whole.

Table 2. Summary of texture classification based on CNNs.

References	Purpose	Features	Model	Dataset	Accuracy (%)
[27]	Land use classification from satellite images	Image texture features	Depth feature extraction using customized CNN	PaviaU dataset, Salinas dataset, Indian Pines	CNN ELEM 90
[28]	Texture classification	Feature optimization	CNN optimized through WOA	Kylberg Brodatz OutexTC00012	99.71 97.43 97.70

Table 2. *Cont.*

References	Purpose	Features	Model	Dataset	Accuracy (%)
[29]	Texture classification	Fusion of AlexNet and VGG	AlexNet layer VGG net layer	Brodatz KTH-TIPS CUReT	98.76 100 99.76
[13]	Texture classification	CNN features and Gabor features	CaffeNet	Cifar10 dataset	79.16
[30]	Texture classification	CNN features	New CNN proposed	Brodatz texture database	Error rate (mean) 17.2
[31]	Classification of different ship types	Multiscale rotation invariance CNN features	New CNN developed based on CaffeNet	BCCT200- RESIZE data	98.33
[32]	Classify benign and malignant masses	Deep texture features	SVM classifier and ELM	Breast CAD of 400 cases	80.6 to 91
[12]	Texture classification	CNN features	Pre-trained CNN model	KTH-TIPS CURET-Gray	ResNet 98.75, 97.22 DenseNet 99.35, 98.06

2.3. Limitations of Traditional and CNN-Based Texture Classification

Texture classification using standard methods has a number of drawbacks, such as the fact that traditional methods rely on kernel size, direction, and threshold values, all of which might have an effect on the performance of the method. In conventional approaches, determining which parameters are ideal can be a difficult and time-consuming process. Traditional approaches concentrate on low-level elements; they may have difficulty capturing the intricate and high-level details that are present in textures. In order to achieve a high level of accuracy with CNN-based texture classification, a large dataset is required for training. It is essential that the training data be of high quality. Data that are noisy or have been incorrectly labeled can have a detrimental effect on the performance of CNN-based texture classification.

To extract important regions, edges, and textures from images for segmentation, classification, and object recognition [33], INVG and IHVG algorithms are effective and simple. Laifan et al. [15] used degree distribution measures to sort textures into groups on a well-known dataset. The degree distribution feature checks the strength of edges to find important edges and features in an image for classifying it. However, the degree distribution feature only reflects image edge connectivity, not spatial relationships between regions. Small differences in pixel values that affect degree distribution may also impair model accuracy. The clustering coefficient feature shows an image's local structure and can find pixel clusters or related regions. It finds texture-matched spots in images. To grasp an image's structure, we need global and local relationships. In this work, we classified textures using degree distribution, the clustering coefficient, and a combination of both. Putting together degree distribution and clustering coefficient data to make a feature vector that shows the overall and detailed structure of a picture might help with accurately classifying textures. Together, these two traits often show less noise than either one by itself. This is because they show both how an image is put together overall and how pixels are connected locally. In this study, information was obtained from both merged and separate INVG and IHVG graphs. When you combine INVG and IHVG graphs, you obtain a clearer image structure description, which makes texture recognition more accurate.

3. Methodology

3.1. Complex Network

A graph that contains non-trivial topology properties of the real system is referred to as a complex network. It is difficult to predict the collective behavior of complex systems from their individual components, but the complex network (graph) of a system makes us capable of understanding them. Complex networks are used to represent many different kinds of complicated systems because of their properties, which include the prediction of time series and the classification of images. Because they have such a wide range of

applications, time series and image analysis are both extremely significant in a variety of fields. Complex networks are a useful tool for understanding the characteristics of both time series and images. This is because the visibility graph technique establishes strong links between complex networks, time series, and images. There are different types of visibility graphs, like horizontal visibility graphs, natural visibility graphs, multiplex visibility graphs, image horizontal visibility graphs, and image natural visibility graphs, that can be used to map time series and images into complex networks [34].

3.2. Visibility Graph (VG)

In a visibility graph, every entity is denoted as a node, and the existence of a connection between any two nodes indicates that the entity can be seen by the other nodes within the graphical representation. In a two-dimensional space, objects are frequently represented as line segments. There is a direct line of sight between the endpoints of two segments connected by an edge. Applications for visibility graphs include things like route planning, robot navigation, and computer graphics. Several algorithms, including proximity networks, visibility graphs, and transition networks, have been devised to create complicated networks out of time series [35]. The visibility graph method is a useful tool for going from time series and image data to a graphical representation. As mentioned by Lacasa et al. [36], the visibility graph is a simple way to add time series to the network. First, we will discuss two types of visibility graphs that can be made from a single-variable time series and one way that they could be expanded to images.

3.2.1. Natural Visibility Graph

The NVG is a type of visibility graph that is constructed using time series data. It is a method of describing a time series underlying the relationship architecture. Each time series data point is represented as a node in an NVG, and an edge between two nodes indicates that the related data points are mutually observable. In particular, two data points are deemed mutually observable if no other data point lies on the straight line segment that connects them. An NVG is built by computing the pairwise visibility between all pairs of data points in the time series data. Many intriguing aspects and uses of NVGs have been discovered, including the characterization of chaotic systems, the discovery of patterns and trends in time series data, and the detection of abnormalities or outliers. They have also been utilized in the development of time series analysis machine learning techniques, such as those based on graph neural networks. Figure 1 depicts an illustration utilising the same time series data as the natural visibility graph.

For the time series data $s = f(r)$, let $S = s_1, s_2, ..., s_N$ be an ordered series of N real-valued data. A undirected VG is a type of undirected graph that comprises a set of N nodes, with each node $a \in [1, N]$ assigned a label based on the chronological sequence of its corresponding datum, s_a. Consequently, the variable s_1 is assigned to the node with index $a = 1$, s_2 is assigned to the node with index $b = 2$, and so forth. Subsequently, a pair of nodes denoted as a and b (with the assumption that $a < b$ without loss of generality) are deemed to be interconnected by an undirected link if it is feasible to create a straight line that connects s_a and s_b without crossing any intervening datum s_c in Equation (1), where $a < b < c$. The following convexity visibility criteria must be satisfied for the nodes a and b to be connected.

$$s_c < s_a + \frac{c-a}{b-a}(s_b - x_a), \qquad \forall (a < c < b). \tag{1}$$

The undirected and unweighted visibility graph can be represented by the adjacency matrix (A_{ab}) in Equation (2).

$$A_{ab}^{(VG)} = A_{ba}^{(VG)} = \prod_{c=a+1}^{b-1} f\left(s_b + (s_a - s_b)\frac{r_b - r_c}{r_b - r_a} - s_c\right) \tag{2}$$

where f is a Heaviside function.

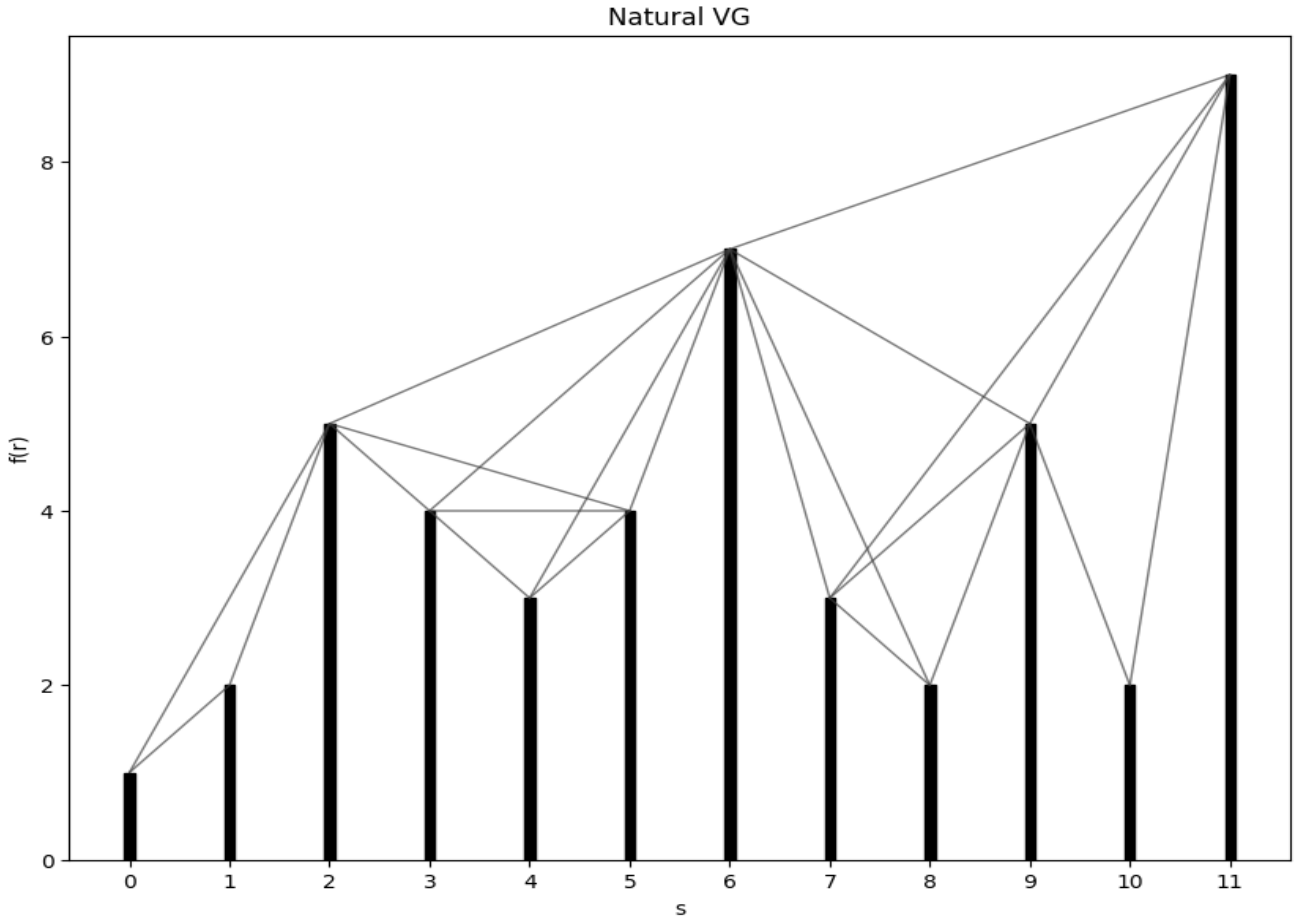


Figure 1. Natural visibility graph from time series data.

3.2.2. Horizontal Visibility Graph

An HVG is a type of visibility graph that is created with time series data [37]. Each time series data point is represented as a node in a horizontal visibility graph, and an edge is drawn between two nodes if the line connecting them is strictly horizontal and does not intersect any other data points in the time series data. This means that if two data points on the horizontal axis have a clear line of sight, an edge in the HVG connects them. Figure 2 depicts an illustration utilising the same time series data as the horizontal visibility graph. The horizontal visibility graph is a sub-graph of the natural visibility graph. The construction of a horizontal visibility graph involves the application of a restricted visibility criterion and the imposition of horizontal visibility. The connection between nodes a and b is established if and only if it is possible to draw a horizontal line that links s_a and s_b, without intersecting any intermediate datum s_c, where $a < c < b$. Alternatively, it can be observed that the connection between a and b in the HVG is dependent upon satisfying the following ordering criterion in Equation (3):

$$s_c < \inf(s_a, s_b), \quad \forall (a < c < b). \tag{3}$$

There is an edge (a, b) if $s_c < \inf(s_a, s_b)$ for all c with $a < c < b$, so the adjacency matrix (A_{ab}^{HVG}) for horizontal visibility graph is mentioned in Equation (4),

$$A_{ab}^{(HVG)} = A_{ba}^{(HVG)} = \prod_{c=a+1}^{b-1} f(s_a - s_c) f(s_b - s_c). \tag{4}$$

where f is a Heaviside function.

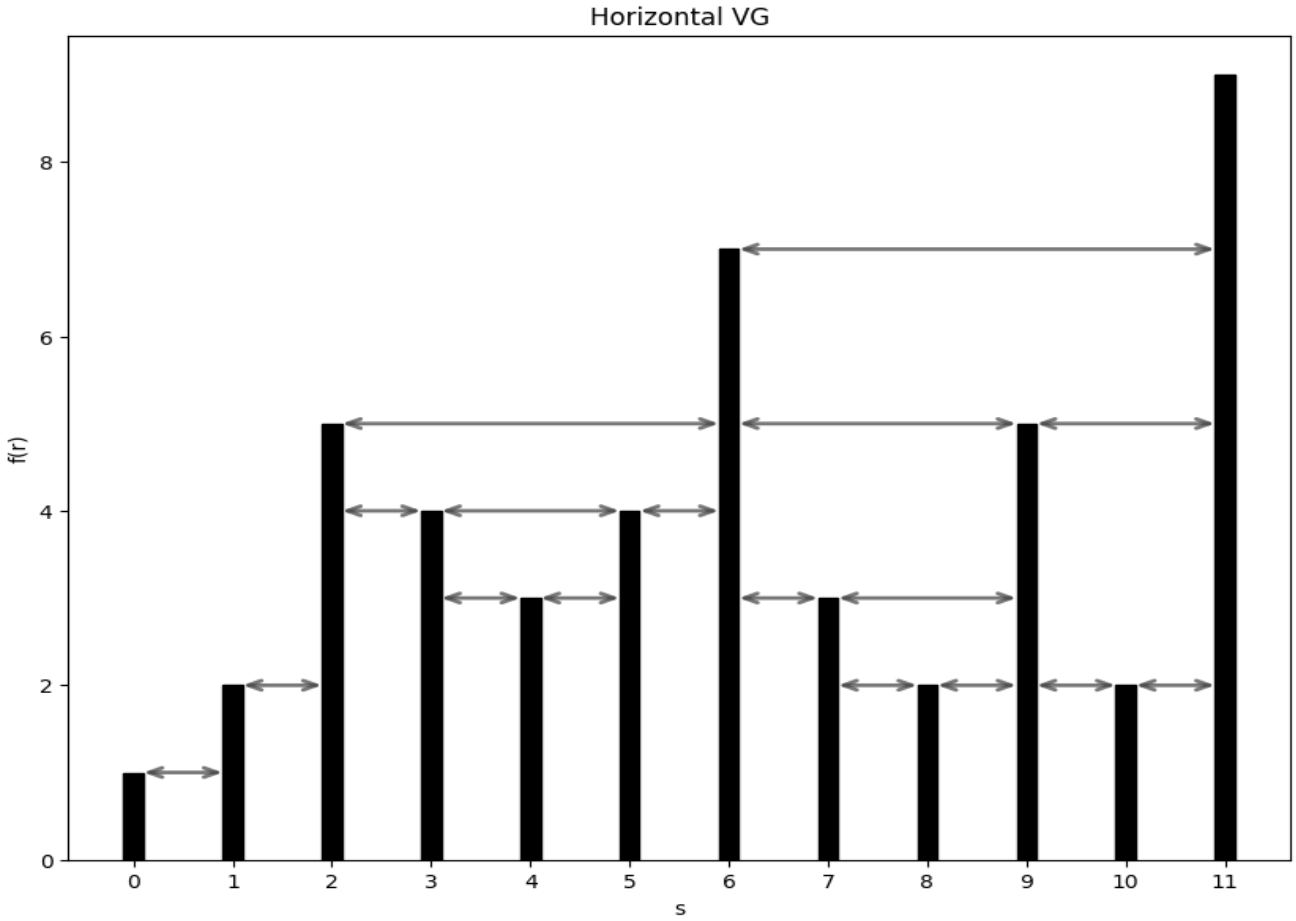


Figure 2. Horizontal visibility graph from time series data.

Various graph-theoretical techniques can be employed to examine the resulting HVG and extract information regarding the fundamental time series data. HVGs have been employed in various disciplines such as neuroscience, meteorology, economics, and music analysis.

3.3. Image Visibility Graph

In an IVG, every pixel is called a node, and lines connect two nodes if the line joining them does not collide with any other pixels. An IVG edge connects pixels with a clear line of sight. For the purpose of assessing image features such as pattern recognition, texture classification, and filters, a direct approach that is used to transform scalar data into graphs is the family of image visibility graphs. This method is used to turn scalar data into graphs. The idea of the visibility graph is developed even further when viewed in conjunction with its corresponding visual representation [16]. The generated IVG can then be analyzed to obtain information about the underlying image using various graph-theoretical metrics. Image analysis, texture identification, and pattern recognition are just a few of the disciplines for which IVGs have been used.

3.3.1. Image Natural Visibility Graph

An INVG is a form of visibility graph that is built using an image's attributes. Each pixel in the image is represented as a node in an INVG, and an edge is drawn between two nodes if the line joining them does not cross with any other pixels in the image and their grayscale values meet certain conditions. A pixel's grayscale value must be either a local maximum or a local minimum in the row or column to which it belongs, and the

grayscale values of two pixels joined by an edge must be either both local maxima or both local minima.

An INVG is a graph with N^2 nodes, each identified by its datum indices, denoted as I_{ab}. The nodes ab and $a'b'$ are connected if

- $a = a'$ OR $b = b'$ OR $[a = b \pm t$ AND $b = b \pm t]$ for some integer t and
- the NVG definition algorithm establishes a connection between I_{ab} and $I_{a'b'}$. This algorithm is executed on an ordered sequence that comprises I_{ab} and $I_{a'b'}$.

3.3.2. Image Horizontal Visibility Graph

IHVGs are produced using image attributes. An edge is drawn between two nodes in an IHVG if the line connecting them is strictly horizontal, does not collide with any other pixels in the image, and the grayscale values of the two pixels accomplish specific criteria (the image must be grayscale, and each row must contain the local maxima and minima).

An IHVG is a graph consisting of N^2 nodes, where each node in the graph is identified by the indices of its corresponding datum, denoted as I_{ab}. The nodes ab and $a'b'$ are connected if

- $a = a'$ OR $b = b'$ OR $[a = b \pm t$ AND $b = b \pm t]$ for some integer t and
- the HVG definition algorithm establishes a connection between I_{ab} and $I_{a'b'}$, both of which are included in an ordered sequence.

Graph-theoretical metrics can be used to analyze INVGs and IHVGs to learn about the image. Both are used for texture recognition, image segmentation, image retrieval, etc.

3.4. Feature Extraction

A graph is generated from the INVG and IHVG, which include rooted data about that image. The INVG and IHVG can extract graph attributes such as topological plots, global and local features, and multiplex features [16]. Global features offer topological information about the entire graph, while local attributes consider small substructures [34]. The INVG and IHVG both calculate each pixel's degree and clustering coefficient. In order to calculate the degree and the clustering coefficient for each pixel, a graph employing the INVG and IHVG is generated.

3.4.1. Degree Distribution

The probability distribution of the degree throughout the entire network is known as the degree distribution. According to the [38] theorem, the HVG of N vertices is in bijection with their degree distribution, indicating that degree distribution is important for understanding global properties. Additionally, paper [26] mentions that degree distribution is a good feature by which to study different spatiotemporal dynamical systems.

3.4.2. Clustering Coefficient

The graph's structure can be seen in the vertex's immediate surroundings. While performing classification tasks, the local clustering coefficient measures clique proximity. The clustering coefficient [27] indicates the degree of network node grouping. X_i is the vertex clustering coefficient. The local clustering coefficient is defined in Equation (5),

$$X_i = \frac{2(\text{Number of connected triangles, including node } i)}{x_i(x_i - 1)} \tag{5}$$

where x_i denotes the number of the vertex's neighbors.

This study presents the computation of a degree distribution and clustering coefficient, which are used as input vectors for the classifier. Furthermore, the process of generating a novel feature vector involves the combination of both degree distribution and clustering coefficients for the purpose of classification. The combination of both features is generally more robust to noise in the image than either feature alone, as the two features provide more information about both the overall structure of the images and

the local connectivity of pixels. Also, the combination of an INVG and IHVG captures different aspects of the image structure and allows for a more complete and enhanced representation of the image. Therefore, features are extracted from the combination of these graphs, which can improve the accuracy of image texture classification tasks.

Accuracy is one way to measure how well a classification model works. In the context of binary classification, it is also possible to evaluate accuracy based on the positives and negatives as shown in Equation (6).

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Positive} + \text{False Negative} + \text{True Negative}} \quad (6)$$

3.5. Classifiers for Texture Classification

A specific type of machine learning system used to categorize incoming data is called a classifier. After extracting features using an INVG and IHVG, associating them with a class label enables rapid classification of texture images based on their class titles. Regression and classification are carried out using a decision tree (DT), a supervised machine learning method. The objective is to construct a model that categories the label class value through acquiring simple decision rules from the feature. The SVM is a prevalent and extensively employed supervised machine learning method that is applied for both classification and regression tasks. Establishing a decision boundary that can classify n-dimensional space is the goal of the SVM algorithm. It is a better classifier than others for identifying and categorizing different textures [32]. RF is the most used technique for categorization issues. Random forest builds a decision tree from the different samples, using the average for regression and the majority vote for classification. Other machine learning classifiers include KNN, LDA, and others. In this study, the SVM, RF, DT, and KNN classifiers were employed for texture classification.

4. Experimental Results and Discussion

The performance of the suggested approach was evaluated by conducting an analysis on grayscale images from two distinct datasets; namely, the Brodatz texture image and Salzburg texture image (STex) datasets. The Brodatz texture database contains 24 different textures, with the first 12 being natural textures and the last 12 being artificial textures. More details about the datasets can be found in [39]. Some examples of these texture images are shown in Figure 3. The STex dataset [40] has 476 color texture images. However, STex has many classes of texture images but only two classes, i.e., miscellaneous (misc) and wood, are taken into account for texture image classification. It contains 85 different texture images and the examples of these texture images are shown in Figure 4. The cv2 Python package was used to convert texture images to 64×64 pixels so that they could work with different dataset sizes and computing needs. Using the NaturalVG and HorizontalVG methods from the ts2vg Python library, texture images were converted to INVG and IHVG graphs [41]. For each image, pixel degrees were extracted and a degree distribution feature was constructed. The source code for extracting the features can be found on github (https://github.com/rahulpaljrf/graph_features/tree/main, accessed on 1 October 2023). For image texture classification, the degree distribution feature collects edge strengths and can identify relevant edges and features. Furthermore, the clustering coefficient for every pixel in each image is calculated for both of the graphs. The clustering coefficient feature captures the local structure of an image, which can be used to identify regions that are highly connected, and it is used for identifying regions in an image that belong to the same texture. To make our model more robust, the degree distribution and the clustering coefficient features from both the graphs were combined into a single feature vector; the resulting feature vector captures both the global and local structure of an image, which helps to increase the accuracy of texture classification tasks. Once more, the degree distribution and clustering coefficient are extracted using a combination of INVG and IHVG graphs in order to increase the feature vector. A more thorough and improved depiction of the picture structure is possible

because of the combination of the INVG and IHVG graphs, which capture many facets of the image structure. These feature vectors are created, and then they are fed into different classifiers including SVM, DT, KNN, and RF.

Based on an IVG algorithm, a methodology was created for classifying texture images. Using this standardized method, classification tasks were carried out on databases of grayscale images. The process is explained in detail in Figure 5. The results of the categorization experiments are presented in Tables 3 and 4 for the Brodatz and Salzburg texture picture datasets, respectively. This investigation allowed us to assess the IVG algorithm's performance in precisely identifying textures based on the underlying graphs.

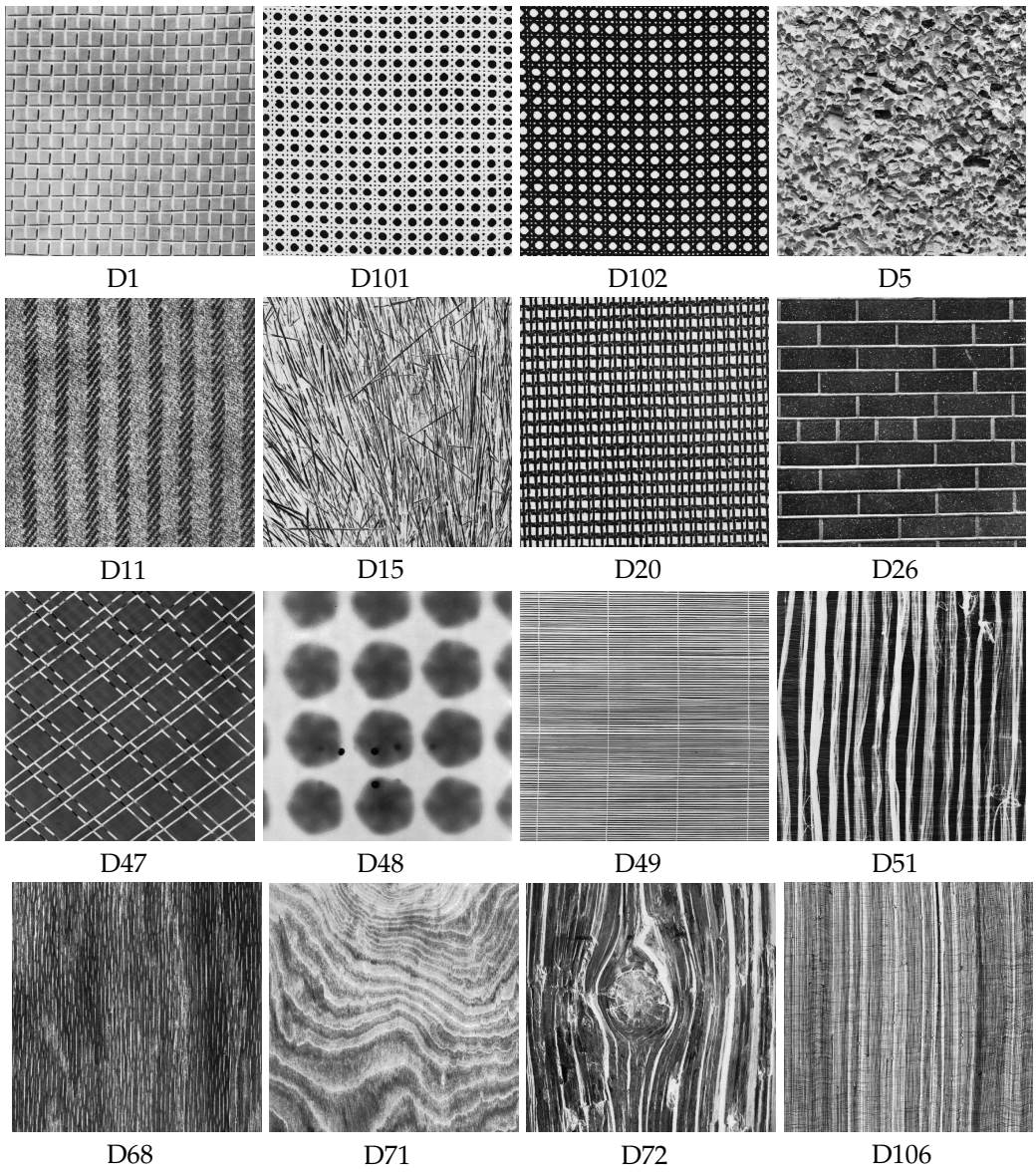


Figure 3. Sample texture images from the Brodatz texture image dataset.

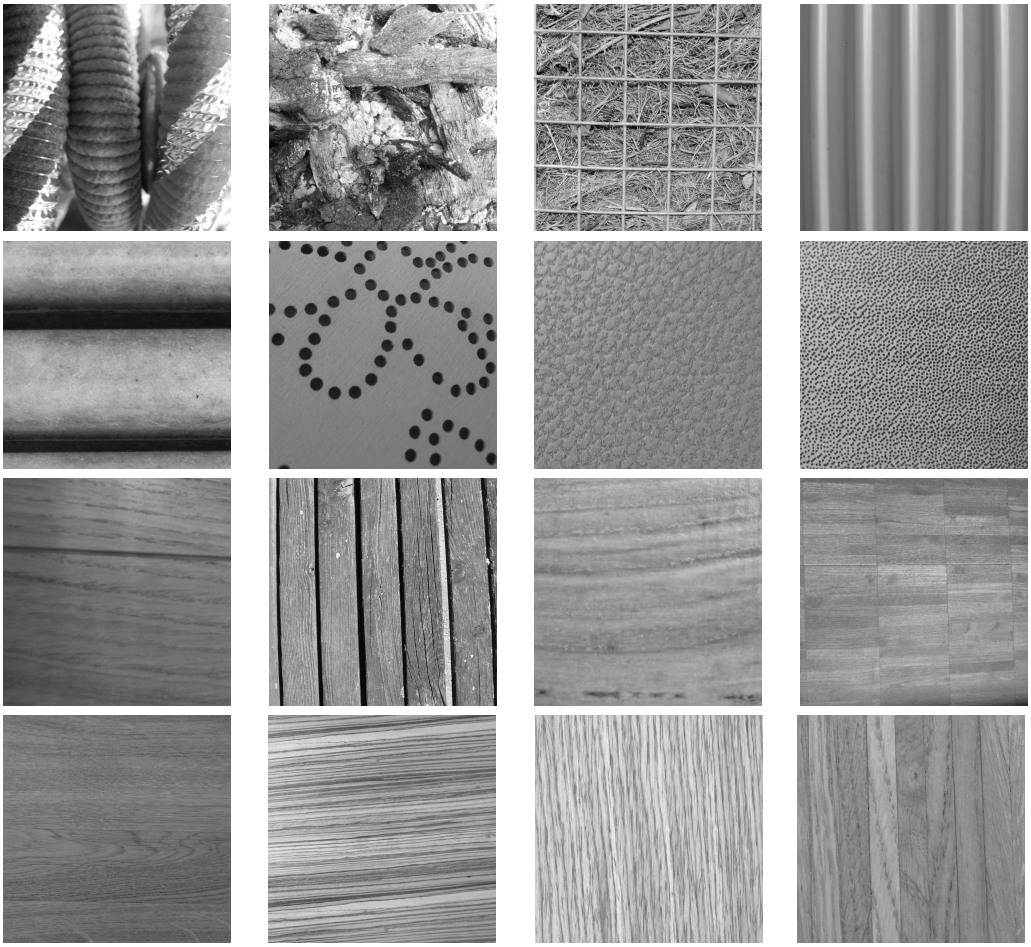


Figure 4. Sample texture images from the Salzburg texture image dataset.

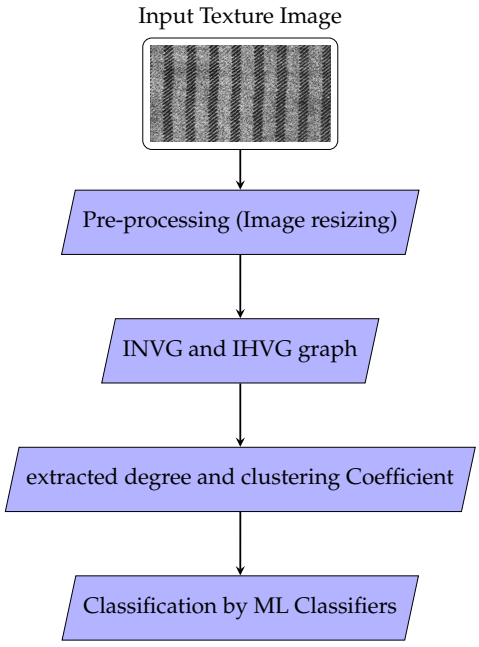


Figure 5. Flowchart of the proposed method.

Table 3. The classification outcomes for the Brodatz texture image dataset.

Dataset	Graph	Feature	Machine Learning Classifier			
			RF (%)	DT (%)	KNN (%)	SVM (%)
Brodatz	INVG	Degree distribution	80	80	60	60
		Clustering coefficient	80	60	60	60
		Combination of both	80	80	60	60
	IHVG	Degree distribution	100	80	100	80
		Clustering coefficient	80	80	60	60
		Combination of both	100	80	100	80
	INVG+IHVG	Degree distribution	80	80	80	80
		Clustering coefficient	80	80	60	60
		Combination of both	80	80	80	80

Table 4. The classification outcomes for the Salzburg texture image dataset.

Dataset	Graph	Feature	Machine Learning Classifier			
			RF (%)	DT (%)	KNN (%)	SVM (%)
STex	INVG	Degree distribution	81.81	68.18	63.63	72.72
		Clustering coefficient	82.35	70.58	70.58	88.23
		Combination of both	76.47	76.47	70.58	70.58
	IHVG	Degree distribution	63.63	63.64	68.18	54.54
		Clustering coefficient	76.58	76.47	68.88	58.88
		Combination of both	82.35	77.77	76.47	68.18
	INVG+IHVG	Degree distribution	81.81	77.27	68.81	72.72
		Clustering coefficient	82.35	76.47	68.18	72.27
		Combination of both	82.35	76.47	70.47	74.71

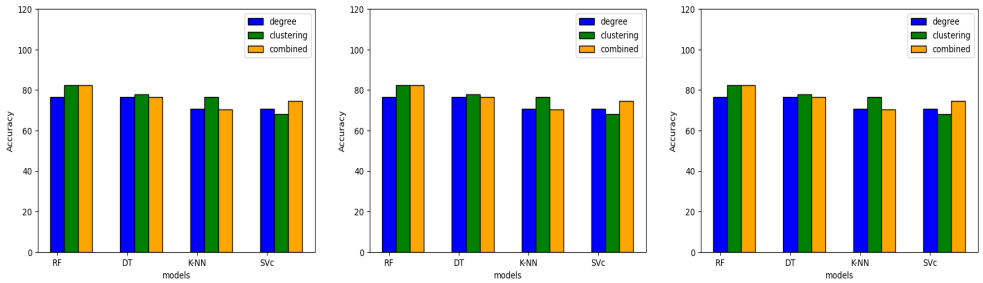
4.1. Classification Results on Brodatz Texture Image Dataset

Table 3 illustrates the classification accuracy of four distinct machine learning classifiers, i.e., RF, DT, KNN, and SVM, on the Brodatz texture image dataset. These classifiers were applied to three separate graph features; primarily, INVG, IHVG, and a feature combining INVG and IHVG. The degree distribution and clustering coefficient characteristics of the INVG graph reached an accuracy of 80% for RF and 60% for DT, KNN, and SVM. The accuracy of RF and DT increased to 80% when the two features were

integrated, whereas KNN and SVM continued to be accurate just 60% of the time. This lends credence to the idea that the degree distribution feature is a more informative one for this dataset. The degree distribution feature alone is able to reach an accuracy of 100% for the RF and KNN graphs, but the DT and SVM graphs are only able to achieve an accuracy of 80%. The accuracy of all classifiers can be increased to 80% by utilizing the clustering coefficient feature. The accuracy of RF and KNN once again improves to 100% when both characteristics are merged; however, DT and SVM maintain their previous accuracy level of 80%. It can be deduced that the degree distribution feature is quite useful for this dataset and that combining both features can improve the performance of the model. When the INVG and IHVG are combined, all classifiers reach an accuracy of 80%, which is consistent regardless of the types of features they employ. This leads one to believe that the features contained within this dataset are well-balanced and offer the classifiers comparable quantities of information.

Overall, these results emphasize the significance of selecting informative graph features for machine learning classification tasks on texture image datasets, as well as the benefits of combining various graph features to improve a model's performance.

Figure 6a shows the accuracy of the different machine learning classifiers on three features extracted from the INVG graph. It shows that clustering coefficients and combined features are very useful for the INVG graph for analyzing grayscale image attributes.



(**a**) (**b**) (**c**)

Figure 6. Accuracy of (**a**) INVG, (**b**) IHVG, and (**c**) combined graphs on Brodatz dataset.

Figure 6b shows the accuracy of the different machine learning classifiers on three features extracted from the IHVG graph. It shows that degree distribution and combined features are useful for the IHVG graph for analyzing grayscale image properties.

Figure 6c shows the accuracy of different machine learning classifiers on three features extracted from the combined graph. It also suggests that the degree distribution and combined features are more useful in analyzing grayscale images for the combined graph.

Figure 7a shows the accuracy of the degree distribution feature on different machine learning classifiers applied on the INVG, IHVG, and a combination of both graphs. RF and DT classifiers achieved 80% accuracy for the INVG graph, while KNN and SVM classifiers achieved 60%. The KNN classifier achieved the highest accuracy of 100% for the IHVG graph, while DT, RF, and SVM classifiers achieved 80%. For the combined graph all classifiers achieved the same accuracy of 80%. These results suggest that degree distribution helps train machine learning classifiers to analyze graph features from diverse graphs. The performance of the classifiers changed depending on the graphs.

Figure 7b shows the accuracy of the clustering coefficient feature on different machine learning classifiers applied on the INVG, IHVG, and a combination of both graphs. The RF classifier achieved the highest accuracy of 80% for the INVG graph, while DT, KNN, and SVM classifiers achieved 60%. For the IHVG and combined graph, RF and DT classifiers both had the highest accuracy of 80%, while KNN and SVM classifiers had 60% accuracy. These results suggest that the clustering coefficient is a moderately useful feature for analyzing graph features.

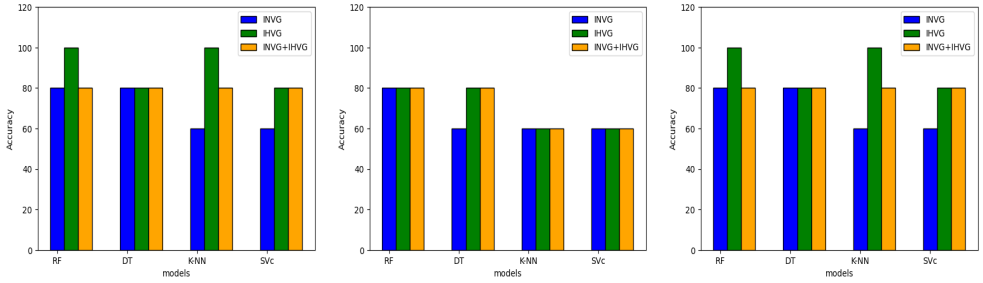


Figure 7. Accuracy of (**a**) degree distribution, (**b**) clustering coefficient, and (**c**) combined features on Brodatz dataset.

Figure 7c shows the accuracy of the combination of both degree and clustering coefficient features on different machine learning classifiers applied on the INVG, IHVG, and a combination of both graphs. RF and DT classifiers achieved 80% accuracy for the INVG graph, while KNN and SVM classifiers achieved 60%. RF and KNN classifiers achieved an accuracy of 100% for the IHVG graph, while DT and SVM classifiers had 80% accuracy. For the combined graph, all classifiers achieved the same accuracy of 80%. These results suggest that combined features are useful for analyzing graph features.

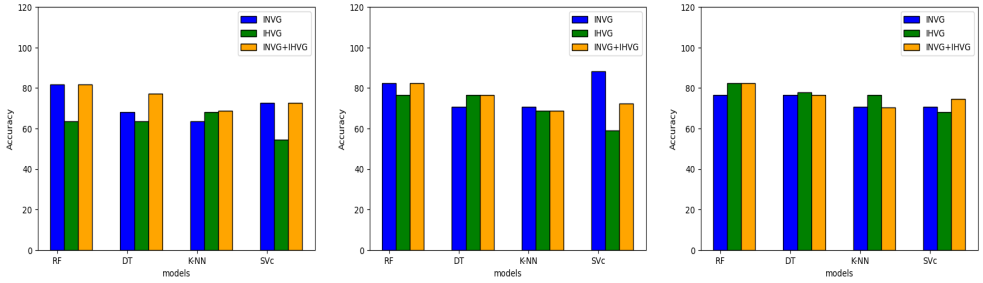
4.2. Classification Results on Salzburg Texture Image Dataset

In the same way, Table 4 shows the classification accuracy of four different machine learning classifiers (RF, DT, KNN, and SVM) on three different graph features (mainly INVG, IHVG, and features from the combined INVG and IHVG graph) from the STex dataset.

RF, SVM, and KNN classifiers all performed better on the degree distribution feature of the INVG graph, whereas the DT classifier performed the best on the clustering coefficient feature, with an accuracy of 70.58%. The accuracy of DT and SVM classifiers increased when both features were used together, but that of the RF and KNN classifiers dropped significantly. The DT classifier, when trained on the combined features, had the greatest accuracy of 76.47%. The degree distribution feature was less accurate than the clustering coefficient feature for the IHVG graph for all classifiers except KNN. All classifiers became more accurate when the two features were merged, with the RF classifier having the best accuracy (82.35%). The degree distribution feature achieved greater accuracy on the RF, DT, and SVM classifiers when INVG and IHVG graphs were combined, while the clustering coefficient feature achieved greater accuracy on KNN classifiers. The accuracy of all classifiers increased when the two features were merged, with the RF classifier achieving the best accuracy of 82.35%.

Overall, these results show that the choice of feature depends on the graphs and that different features have different levels of importance for different classifiers. The combination of features can enhance the accuracy of classification, but the benefit is dependent on the graphs and classifier used.

Figure 8a shows the accuracy of the degree distribution feature on the different machine learning classifiers applied on the INVG, IHVG, and a combination of both graphs. For the INVG graph, RF and SVM classifiers were more accurate than DT and KNN classifiers. The accuracy of all classifiers slightly decreased for the IHVG graph. On a combined graph, RF and DT classifiers were more accurate than KNN and SVM. These results suggest that degree distribution is a moderate feature for training machine learning classifiers to analyze graph features made from different graphs on the STex dataset.



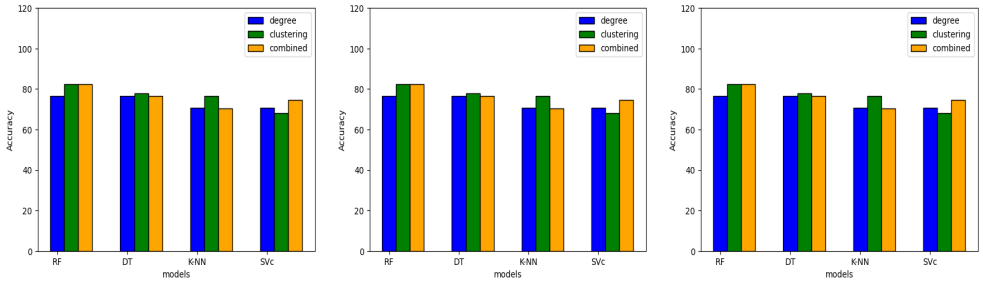
(**a**) (**b**) (**c**)

Figure 8. Accuracy of (**a**) degree distribution, (**b**) clustering coefficient, and (**c**) combined features on STex dataset.

Figure 8b shows the accuracy of the clustering coefficient feature on the different machine learning classifiers applied on the INVG, IHVG, and a combination of both graphs. SVM and RF classifiers performed better compared to KNN and DT classifiers for the INVG graph. For the IHVG and combined graph, RF and DT classifiers performed better than KNN and SVM.

Figure 8c shows the accuracy of the combination of both degree and clustering coefficient features on different machine learning classifiers applied on INVG, IHVG, and a combination of both graphs. All classifiers achieved similar accuracy for all graphs, which means they suggest that the combined features are also very useful for analyzing the graph.

Figure 9a compares machine learning classifier accuracy on three features extracted from the INVG graph. It also shows that the clustering coefficients and combined features are very useful for the INVG graph for analyzing grayscale image structures.



(**a**) (**b**) (**c**)

Figure 9. Accuracy of (**a**) INVG, (**b**) IHVG, and (**c**) combined graphs on STex dataset.

Figure 9b compares machine learning classifier accuracy on three features extracted from the IHVG graph. It also suggests that the clustering coefficient and combined features are very useful for the IHVG graph for analyzing grayscale images.

Figure 9c compares machine learning classifier accuracy on three features extracted from the combined graph. It also suggests that combined features enhance the accuracy and are useful in analyzing grayscale image properties.

4.3. Comparison with Some Existing Methods

This section compares the classification results obtained using various existing methods (Table 5). These methods encompass Gabor, GLCM, hybrid features [42], and CNN features such as MobileNetV3, InceptionV3, VGG-M-FC, and VGG-VD-16-FC. The findings of the investigation indicate that IHVG graphs, particularly those that include degree features and a combination of degree and clustering coefficient features, provide superior performance compared to traditional and CNN features when applied to the Brodatz texture image dataset. Moreover, the INVG graph, when using clustering coefficient fea-

tures, demonstrates superior accuracy compared to CNN features when applied to the Salzburg texture image dataset.

Table 5. Comparison results with some existing features.

Dataset	Feature	Accuracy (%)
Brodatz	Gabor [24]	43.429
	Gabor and GLCM [24]	48.995
	Hybrid feature [42]	89.28
	MobileNetV3 [43]	99.67
	InceptionV3 [43]	99.33
	IHVG (degree feature)	100
	IHVG (combination of degree and clustering)	100
Salzburg	VGG-M-FC [44]	82.5
	IHVG (combination of degree and clustering)	82.35
	VGG-VD-16-FC [44]	83.3
	INVG (clustering feature)	88.23

5. Conclusions and Future Work

In the process of texture classification, the NVG and HVG feature extraction methods are utilized. They have the capability of capturing crucial information as well as complicated patterns in texture data, which may assist in obtaining additional discriminative features. The efficiency of this method will vary depending on the comprehensiveness and type of texture collection. Additionally, the construction of these graphs is computationally intensive, which is especially problematic for large datasets. VG and its derivatives are often used to turn temporal data into complex networks. This technology has been successfully applied to image processing. This paper has presented a novel methodology for classifying textures in the Brodatz and Salzburg texture image databases. The suggested strategy involves the development of image visibility graphs, with the utilization of degree distribution and clustering coefficients as key aspects. This method can be extended to the color image algorithm, which may aid intelligent medical image assessment. This will enable microstructure discovery in complex 3D scenes. Images can be analyzed using NVGs and HVGs for spatial relationships, contours, and object recognition. The method can extract features for machine learning tasks like pattern identification and anomaly detection beyond texture classification. Graph neural networks' visible graph approach will also be improved. Thus, a robust color image method could aid image classification and processing. NVGs and HVGs are useful in many domains that study patterns, visibility, and time series relationships due to their adaptability and versatility. The possible applications of these methods are continually expanding as researchers delve into their applicability across many scientific and technical domains.

Author Contributions: Conceptualization, R.P., S.K. and M.K.S.; methodology, R.P., S.K. and M.K.S.; software, M.A., R.P. and M.K.S.; validation, M.A., S.K., R.P., M.K.S. and D.S.; formal analysis, M.A., S.K. and R.P.; investigation, M.A., R.P., M.K.S. and S.K.; resources, M.A., S.K. and D.S.; data curation, M.A., S.K. and R.P.; writing—original draft preparation, R.P., S.K. and M.K.S.; writing—review and editing, M.A. and D.S.; visualization, M.A., R.P., S.K., M.K.S. and D.S.; supervision, R.P., S.K. and M.K.S.; project administration, M.A.; funding acquisition, M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Grant No. 4783).

Data Availability Statement: The data presented in this study are openly available in Salzburg Texture Image Database (STex) at https://wavelab.at/sources/STex [40] and in The USC-SIPI Image Database at https://sipi.usc.edu/database/database.php [39].

Conflicts of Interest: The authors assert that they have no conflicts of interest.

References

1. Ataky, S.T.M.; Saqui, D.; de Matos, J.; de Souza Britto Junior, A.; Lameiras Koerich, A. Multiscale Analysis for Improving Texture Classification. *Appl. Sci.* **2023**, *13*, 1291. [CrossRef]
2. Backes, A.R.; Casanova, D.; Bruno, O.M. Texture analysis and classification: A complex network-based approach. *Inform. Sci.* **2013**, *219*, 168–180. [CrossRef]
3. Tuceryan, M.; Jain, A.K. Texture analysis. In *Handbook of Pattern Recognition and Computer Vision*; World Scientific Publishing Company: Singapore, 1993; pp. 235–276.
4. Liu, L.; Chen, J.; Fieguth, P.; Zhao, G.; Chellappa, R.; Pietikäinen, M. From BoW to CNN: Two decades of texture representation for texture classification. *Int. J. Comput. Vis.* **2019**, *127*, 74–109. [CrossRef]
5. Iqbal, N.; Mumtaz, R.; Shafi, U.; Zaidi, S.M.H. Gray level co-occurrence matrix (GLCM) texture based crop classification using low altitude remote sensing platforms. *PeerJ Comput. Sci.* **2021**, *7*, e536. [CrossRef] [PubMed]
6. Haralick, R.M.; Shanmugam, K.; Dinstein, I.H. Textural features for image classification. *IEEE Trans. Syst. Man Cybernet.* **1973**, *SMC-3*, 610–621. [CrossRef]
7. Arivazhagan, S.; Ganesan, L. Texture classification using wavelet transform. *Pattern Recognit. Lett.* **2003**, *24*, 1513–1521. [CrossRef]
8. Salinas, R.; Gomez, M. A new technique for texture classification using markov random fields. *Int. J. Comput. Commun. Control* **2006**, *1*, 41–51. [CrossRef]
9. Luimstra, G.; Bunte, K. Adaptive Gabor Filters for Interpretable Color Texture Classification. In Proceedings of the 30th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, 5–7 October 2022; pp. 61–66.
10. Luo, Q.; Su, J.; Yang, C.; Silven, O.; Liu, L. Scale-selective and noise-robust extended local binary pattern for texture classification. *Pattern Recognit.* **2022**, *132*, 108901. [CrossRef]
11. Ataky, S.T.M.; Koerich, A.L. A novel bio-inspired texture descriptor based on biodiversity and taxonomic measures. *Pattern Recognit.* **2022**, *123*, 108382. [CrossRef]
12. Simon, P.; Uma, V. Deep learning based feature extraction for texture classification. *Procedia Comput. Sci.* **2020**, *171*, 1680–1687. [CrossRef]
13. Tianyu, Z.; Zhenjiang, M.; Jianhu, Z. Combining cnn with hand-crafted features for image classification. In Proceedings of the 2018 14th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 12–16 August 2018; pp. 554–557.
14. Van Hoai, D.P.; Hoang, V.T. Feeding Convolutional Neural Network by hand-crafted features based on Enhanced Neighbor-Center Different Image for color texture classification. In Proceedings of the 2019 International Conference on Multimedia Analysis and Pattern Recognition (MAPR), Ho Chi Minh City, Vietnam, 9–10 May 2019; pp. 1–6.
15. Pei, L.; Li, Z.; Liu, J. Texture classification based on image (natural and horizontal) visibility graph constructing methods. *Chaos Interdisciplin. J. Nonlinear Sci.* **2021**, *31*, 013128. [CrossRef]
16. Iacovacci, J.; Lacasa, L. Visibility graphs for image processing. *IEEE Trans. Pattern Anal. Mach Intell.* **2019**, *42*, 974–987. [CrossRef] [PubMed]
17. Saini, D.; Kumar, S.; Singh, M.K.; Ali, M. Two view NURBS reconstruction based on GACO model. *Complex Intell. Syst.* **2021**, *7*, 2329–2346. [CrossRef]
18. Kumar, S.; Kumar, S.; Sukavanam, N.; Raman, B. Human visual system and segment-based disparity estimation. *AEU-Int. J. Electron. Commun.* **2013**, *67*, 372–381. [CrossRef]
19. Kumar, S.; Bhatnagar, G.; Raman, B. Security of stereo images during communication and transmission. *Adv. Sci. Lett.* **2012**, *6*, 173–179. [CrossRef]
20. Cavalin, P.; Oliveira, L.S. A review of texture classification methods and databases. In Proceedings of the 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), Niteroi, Brazil, 17–18 October 2017; pp. 1–8.
21. Ojala, T.; Pietikäinen, M. Unsupervised texture segmentation using feature distributions. *Pattern Recognit.* **1999**, *32*, 477–486. [CrossRef]
22. Liao, S.; Law, M.W.; Chung, A.C. Dominant local binary patterns for texture classification. *IEEE Trans. Image Process.* **2009**, *18*, 1107–1118. [CrossRef]
23. Kumar, S.; Kumar, S.; Raman, B. Image disparity estimation based on fractional dual-tree complex wavelet transform: A multi-scale approach. *Int. J. Wavelets Multiresolut. Inform. Process.* **2013**, *11*, 1350004. [CrossRef]
24. Suresh, A.; Shunmuganathan, K.L. Image texture classification using gray level co-occurrence matrix based statistical features. *Eur. J. Sci. Res.* **2012**, *75*, 591–597.
25. Hosny, K.M.; Magdy, T.; Lashin, N.A.; Apostolidis, K.; Papakostas, G.A. Refined Color Texture Classification Using CNN and Local Binary Pattern. *Math. Probl. Eng.* **2021**, *2021*, 5567489. [CrossRef]
26. Lacasa, L.; Iacovacci, J. Visibility graphs of random scalar fields and spatial data. *Phys. Rev. E* **2017**, *96*, 012318. [CrossRef] [PubMed]

27. Huang, F.; Lu, J.; Tao, J.; Li, L.; Tan, X.; Liu, P. Research on optimization methods of ELM classification algorithm for hyperspectral remote sensing images. *IEEE Access* **2019**, *7*, 108070–108089. [CrossRef]
28. Dixit, U.; Mishra, A.; Shukla, A.; Tiwari, R. Texture classification using convolutional neural network optimized with whale optimization algorithm. *SN Appl. Sci.* **2019**, *1*, 655. [CrossRef]
29. Roy, S.K.; Dubey, S.R.; Chanda, B.; Chaudhuri, B.B.; Ghosh, D.K. Texfusionnet: an ensemble of deep cnn feature for texture classification. In *Proceedings of the 3rd International Conference on Computer Vision and Image Processing: CVIP 2018*; Springer: Berlin/Heidelberg, Germany, 2020; Volume 2, pp. 271–283.
30. Tivive, F.H.C.; Bouzerdoum, A. Texture classification using convolutional neural networks. In Proceedings of the TENCON 2006—2006 IEEE Region 10 Conference, Hong Kong, China, 14–17 November 2006, pp. 1–4.
31. Shi, Q.; Li, W.; Zhang, F.; Hu, W.; Sun, X.; Gao, L. Deep CNN with multi-scale rotation invariance features for ship classification. *IEEE Access* **2018**, *6*, 38656–38668. [CrossRef]
32. Wang, Z.; Li, M.; Wang, H.; Jiang, H.; Yao, Y.; Zhang, H.; Xin, J. Breast cancer detection using extreme learning machine based on feature fusion with CNN deep features. *IEEE Access* **2019**, *7*, 105146–105158. [CrossRef]
33. Kumar, S.; Kumar, S.; Raman, B.; Sukavanam, N. Human action recognition in a wide and complex environment. In Proceedings of the Real-Time Image and Video Processing 2011, San Francisco, CA, USA, 24–25 January 2011; Volume 7871, pp. 176–187.
34. Wen, T.; Chen, H.; Cheong, K.H. Visibility graph for time series prediction and image classification: A review. *Nonlinear Dyn.* **2022**, *110*, 2979–2999. [CrossRef]
35. Zou, Y.; Donner, R.V.; Marwan, N.; Donges, J.F.; Kurths, J. Complex network approaches to nonlinear time series analysis. *Phys. Rep.* **2019**, *787*, 1–97. [CrossRef]
36. Lacasa, L.; Luque, B.; Ballesteros, F.; Luque, J.; Nuno, J.C. From time series to complex networks: The visibility graph. *Proc. Nat. Acad. Sci. USA* **2008**, *105*, 4972–4975. [CrossRef]
37. Luque, B.; Lacasa, L.; Ballesteros, F.; Luque, J. Horizontal visibility graphs: Exact results for random time series. *Phys. Rev. E* **2009**, *80*, 046103. [CrossRef]
38. Luque, B.; Lacasa, L. Canonical horizontal visibility graphs are uniquely determined by their degree sequence. *Eur. Phys. J. Spec. Top.* **2017**, *226*, 383–389. [CrossRef]
39. Abdelmounaime, S.; Dong-Chen, H. New Brodatz-based image databases for grayscale color and multiband texture analysis. *Int. Sch. Res. Not.* **2013**, *2013*, 876386. [CrossRef]
40. Hofbauer, H.; Huber, S. Salzburg Texture Image Database (STex). 2019. Available online: https://wavelab.at/sources/STex/ (accessed on 1 October 2023).
41. Bergillos, C. ts2vg 1.2.2. 2023. Available online: https://pypi.org/project/ts2vg/ (accessed on 1 October 2023).
42. Ahmadvand, A.; Daliri, M.R. Invariant texture classification using a spatial filter bank in multi-resolution analysis. *Image Vis. Comput.* **2016**, *45*, 1–10. [CrossRef]
43. Goyal, V.; Sharma, S. Texture classification for visual data using transfer learning. *Multimedia Tools Appl.* **2023**, *82*, 24841–24864. [CrossRef] [PubMed]
44. Bello-Cerezo, R.; Bianconi, F.; Di Maria, F.; Napoletano, P.; Smeraldi, F. Comparative evaluation of hand-crafted image descriptors vs. off-the-shelf CNN-based features for colour texture classification under ideal and realistic conditions. *Appl. Sci.* **2019**, *9*, 738. [CrossRef]