Proceeding Paper

Attention-Guided Deep Learning Texture Feature for Object Recognition Applications †

Sachinkumar Veerashetty

Department of Computer Science and Engineering, Sharnbasva University, Kalaburagi 585103, India; sveerashetty@gmail.com

† Presented at the International Conference on Recent Advances on Science and Engineering, Dubai, United Arab Emirates, 4–5 October 2023.

Abstract: Image processing-based pattern recognition applications often use texture features to identify structural characteristics. Existing algorithms, including statistical, structural, model-based, and transform-based, lack expertise for specialized features extracted around potentially defective regions. This paper proposes an attention-guided deep-learning texture feature extraction algorithm that can learn features at various regions with varying complexities, addressing the lack of expertise in existing techniques. This approach can be used for applications such as minor fabric defects and hairline faults in PCB manufacturing.

Keywords: texture features; deep learning; object recognition; PCB manufacturing; attention-guided deep learning models

1. Introduction

In computer vision and object recognition applications, the extraction of texture features plays a significant role [1]. The machine learning algorithm is trained to recognize objects using texture features that are extracted from the image. The analysis of satellite or aerial imagery, facial recognition, biometric object recognition, texture enhancement and robot vision for unmanned aerial vehicles, texture synthesis for computer graphics, and image compression are just a few examples of the numerous applications where texture analysis is important [2]. Numerous texture extraction techniques have been proposed since 1960 [3]. These methods convert an image’s texture into a feature vector that describes its characteristics. This feature vector can be applied to later tasks like classifying textures. Analyzing the pixel neighborhood is necessary to capture the spatial context because an image’s texture contains spatial context information. The majority of methods turn an image into a collection of small-scale features before representing it globally. The local texture features are aggregated to produce a global representation using operations like sum, max, min, etc.

The existing texture feature extraction algorithms can be classified into two categories: traditional and learning-based [4]. Traditional feature extraction algorithms extract various statistical-, structural-, spectral- and model-based features from image. But these features are not adaptive to fine-level fabric textural differences and lack specialization to certain regions desired by applications. Learning-based approaches have recently been proposed for the extraction of texture features. The learning-based approaches can be categorized as vocabulary learning strategies, extreme learning approaches, and deep learning approaches. The learning-based method has higher capabilities compared to traditional feature extraction algorithms. Among the learning-based approaches, deep learning-based approaches have recently gained importance due to their ability to learn intricate features with much handcrafting, as needed in traditional feature extraction algorithms. Though the deep learning approaches avoid handcrafting, they still have problems related to generalized learning and lack selective intricate learning based on specialization regions desired by
applications. This work proposes a solution to this problem using attention-based deep learning feature extraction. The proposed solution identifies the specialization regions in the image through frequency domain analysis and an LBP-based convolutional kernel is designed to extract more intricate features at salient regions compared to other regions. Following are the novel contributions of the work.

(i) A novel specialization region selection algorithm based on frequency domain analysis using Quaternion wavelet transform;

(ii) A novel LBP convolutional kernel to extract more intricate features at the specialization regions.

The discriminative ability of features for object recognition applications improves with specialization and more intricate features at the specialization region.

This paper is organized as follows: Section 2 details the proposed attention-based deep-learning texture feature extraction technique. Section 3 provides the results of the discriminative ability of the proposed deep learning feature for different applications. Section 4 presents the conclusion and scope for further research.

2. Materials and Methods

Andrearczyk et al. [5] introduced a deep learning-based technique for texture feature extraction, replacing traditional filters. They modified CNN architecture to reduce shape emphasis, but this led to higher-dimensional features and lacked compactness. They also used a modified AlexNet model for image texture extraction [6].


Kociołek et al. [12] used CNN for texture directionality detection, but found it lacked granularity and compaction support. Sabine et al. [13] explored deep CNN models, but not specific regions. Zhang et al. [14] combined convolutional and encoding layers, experimenting with various encoders, but did not focus on extracting features from specific regions. Sabino et al. [15] proposed a multilayer network for texture feature extraction, but faced computational issues. Barburiceanu et al. [16] extracted textures from deep learning models, but lacked region specification. Anwer et al. [17] combined LBP with a deep learning model to create TEX-Nets, but the fusion did not consider specific image regions.

Jia et al. [18] used a two-stage recurrent neural network to extract shape and texture features, while Kasthuri et al. [19] combined deep learning with Gabor filters for face recognition. Simon et al. [20] combined deep architecture features with luminance information. Bello et al. [21] utilized CNN to extract color texture features, revealing superior discriminative ability compared to hand-crafted descriptors.

The studies on deep learning methods for extracting texture features found that none are region-specific. However, for applications like fabric defect detection and manufacturing hairline defects, it’s crucial to extract features uniformly across the entire image, focusing on areas where defects are more likely to occur. This specialized feature extraction can significantly improve the accuracy of defect classification.

The proposed solution uses frequency domain analysis to generate salient regions from an image, an unsupervised clustering algorithm, a modified CNN with an LBP convolutional kernel, and texture feature extraction as shown in Figure 1.
Figure 1. Architecture of the proposed solution.

2.1. Proposal Generation

HIS transform is applied to the input RGB image before performing FDA. HSI transform on RGB color image is performed as follows:

\[ H = \begin{cases} \theta, & B \leq G \\ 360 - \theta, & B > G \end{cases} \]  

(1)

\[ S = 1 - \frac{3}{(R + G + B)} \min(R, G, B) \]  

(2)

\[ I = \frac{1}{3}(R + G + B) \]  

(3)

where \( \theta \) is given as

\[ \theta = \arccos \left\{ \left( \frac{1}{2} \right) \left[ (R - G) + (R - B) \right] / ((R - G)^2 + (R - B)(G - B))^{\frac{1}{2}} \right\} \]  

(4)

where \( R, G, B \) are the pixel values for red, green and blue component of the image.

The HSI image can be represented in quaternion form as

\[ f(n, m) = H(n, m)\mu_1 + S(n, m)\mu_2 + I(n, m)\mu_3 \]  

(5)

where \( (n, m) \) is the location of the pixel. \( H(n, m), S(n, m)I(n, m) \) are the Hue, Saturation, and Intensity of pixel at \( (n, m) \). The value of \( \mu \) is selected as per the condition.

The quaternion Fourier transform is performed for pixel at \( (u, v) \) as

\[ F(u, v) = F_1(u, v) + F_2(u, v)\mu_2 \]  

(6)
The Gaussian Quaternion high-pass filter in two dimensions is used to smooth images by removing high frequency components in specific ranges. A Gaussian Quaternion high-pass filter function is defined as

$$H_q(u, v) = 1 - e^{\frac{D^2(u, v)}{2\sigma^2}}$$

where \(\sigma\) is a measure of Gaussian spread and \(D(u, v)\) is the distance of point \((u, v)\) to the origin of the frequency rectangle \((M/2, N/2)\). It is calculated as

$$D(u, v) = \left[ \left( u - \frac{M}{2} \right)^2 + \left( v - \frac{N}{2} \right)^2 \right]^{1/2}$$

The Gaussian high-pass filter offers superior smoothing compared to ideal filters, and the resulting coefficients are then processed using the inverse quaternion transform. The inverse transform is given as

$$f_i(n, m) = \frac{1}{\sqrt{MN}} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} e^{-2\pi i \frac{(mu + nv)}{MN}} f_i(u, v)$$

The inverse transform image is evaluated for salient regions based on color contrast, with regions with sharp color contracts being marked as proposal regions. The probability of a salient region to a proposal region is found as

$$P_R = \frac{1}{WC}$$

where

$$WC = \sum_{j=1}^{n} W(i, j) \cdot ||mc_i - mc_j||$$

The weight \(W\) is calculated based on the area of super pixel segment and spatial similarity as

$$W(i, j) = |SP_j| Sim_d(i, j)$$

When \(P_R\) is greater than the threshold, then the salient region is selected as the proposal region.

2.2. Attention Matrix Generation

The attention matrix is used for feature learning with different intensities across regions. A dataset of images is collected, with each image split into grids of size \(n \times n\). A binary matrix representing each grid is created, with initial values of 0 if the grid falls in a proposal region. K mean clustering is performed on the binary matrix, and the attention matrix is generated from the binary matrices of the higher density cluster.

2.3. Modified CNN with LBP Kernel

This study proposes a modified CNN with a convolutional kernel between Gaussian and LBP, convolving pixel regions and applying a new kernel matrix for learning intricate features using LBP. For an image region marked by an attention matrix, multi-scale representation is formed, applying a Laplacian of Gaussian (LoG) operator. It is calculated as

$$\text{LoG}(x, y) = -\frac{1}{\Pi \sigma^2} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) e^{\frac{x^2 + y^2}{2\sigma^2}}$$

\(\text{LoG}\) is usually 0 for a uniform image and it changes to positive or negative depending on the darkness level of the image.
In each scale, $LBP$ is calculated. The $LBP$ of each scale is logically $\text{AND}$ with the region to generate the feature map as

$$C(q) = \sum_{m=1}^{M} \text{AND}(LBP(\text{scale}(q,m)), \text{region}(q)). \quad (15)$$

The effect of $\text{LoG}$ is approximated using a discrete convolutional kernel, as shown in Figure 2a. The 2-D $\text{LoG}$ for different values of $\sigma$ is shown in Figure 2b.

$$\begin{bmatrix}
0 & 0 & 1 & 0 & 0 \\
0 & 1 & 2 & 1 & 0 \\
1 & 2 & -16 & 2 & 1 \\
0 & 1 & 2 & 1 & 0 \\
0 & 0 & 1 & 0 & 0
\end{bmatrix}$$

(a)  
(b)

Figure 2. (a) Kernel Function. (b) LoG Function.

The modified CNN uses an attention matrix-based convolution to extract texture features from an input image, resulting in 1024 dimension texture features. The architecture includes default convolution in unmarked regions and convolution using the calculated value as shown in Figure 3. The convolution flow discussed so far is summarized in Figure 4.

Figure 3. Modified CNN.
3. Results

The attention-based deep learning texture feature and SVM classifier were tested on two datasets: the texture database Out-ex_TC_00013 [22] and the plant village dataset. The texture database offers 68 RGB images in 68 categories, while the plant village dataset contains numerous plant species in both healthy and diseased states. The proposed solution was compared to CNN, transfer learning, a deep convolutional neural network, and Deep Lumina. Texture features were classified using SVM classifier, and performance was measured in terms of accuracy, precision, recall, and the F1-score [23,24]. Results are presented in Tables 1 and 2.

Table 1. Results for texture database.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Proposed</th>
<th>[17]</th>
<th>[11]</th>
<th>[21]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>93.94</td>
<td>89.76</td>
<td>88.5</td>
<td>85.06</td>
</tr>
<tr>
<td>Precision</td>
<td>94.76</td>
<td>90.82</td>
<td>89.25</td>
<td>85.12</td>
</tr>
<tr>
<td>Recall</td>
<td>93.94</td>
<td>89.76</td>
<td>88.5</td>
<td>84.06</td>
</tr>
<tr>
<td>F1-score</td>
<td>94.35</td>
<td>90.29</td>
<td>88.87</td>
<td>84.58</td>
</tr>
</tbody>
</table>

Table 2. The results for plant village dataset.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Proposed</th>
<th>[17]</th>
<th>[11]</th>
<th>[21]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>97.41</td>
<td>92.63</td>
<td>91.76</td>
<td>90.54</td>
</tr>
<tr>
<td>Precision</td>
<td>96.42</td>
<td>93.41</td>
<td>92.34</td>
<td>92.34</td>
</tr>
<tr>
<td>Recall</td>
<td>96.43</td>
<td>92.11</td>
<td>90.21</td>
<td>89.98</td>
</tr>
<tr>
<td>F1-score</td>
<td>96.42</td>
<td>92.89</td>
<td>91.12</td>
<td>90.14</td>
</tr>
</tbody>
</table>

The proposed solution has an average accuracy of 97.41%, which is at least 4.78% higher than previous works. As shown in Table 3, the proposed solution consistently outperformed existing works for all nine plant cases.
Table 3. Results for categories in plant village dataset.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Apple</td>
<td>99.74</td>
<td>93.13</td>
<td>93.46</td>
<td>90.54</td>
<td>99.82</td>
<td>93.13</td>
<td>91.26</td>
<td>92.84</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Cherry</td>
<td>100</td>
<td>94.13</td>
<td>93.16</td>
<td>91.34</td>
<td>99.43</td>
<td>93.13</td>
<td>90.16</td>
<td>90.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Corn</td>
<td>97.45</td>
<td>91.43</td>
<td>93.16</td>
<td>92.34</td>
<td>98.41</td>
<td>92.23</td>
<td>90.16</td>
<td>90.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Grape</td>
<td>99.53</td>
<td>92.43</td>
<td>92.26</td>
<td>91.34</td>
<td>97.99</td>
<td>93.23</td>
<td>91.36</td>
<td>91.16</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Peach</td>
<td>98.1</td>
<td>92.13</td>
<td>92.26</td>
<td>92.34</td>
<td>94.15</td>
<td>93.12</td>
<td>90.26</td>
<td>90.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The proposed features’ performance was compared with other deep learning models like Alexnet, VGG16, and Resnet for the Texture dataset, as shown in Table 4.

Table 4. Comparison with other deep learning models for texture dataset.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Proposed</th>
<th>Alexnet</th>
<th>VGG16</th>
<th>Resnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>93.94</td>
<td>84.16</td>
<td>85.15</td>
<td>85.06</td>
</tr>
<tr>
<td>Precision</td>
<td>94.76</td>
<td>85.23</td>
<td>86.21</td>
<td>85.12</td>
</tr>
<tr>
<td>Recall</td>
<td>93.94</td>
<td>84.61</td>
<td>85.65</td>
<td>84.06</td>
</tr>
<tr>
<td>F1-score</td>
<td>94.35</td>
<td>85.19</td>
<td>86.13</td>
<td>84.58</td>
</tr>
</tbody>
</table>

The proposed attention-based deep feature extraction model outperforms other deep learning models by at least 9% for texture datasets, and its performance was compared to Alexnet, VGG16, and Resnet for the plant village dataset; the result is given in Table 5.

Table 5. Comparison with other deep learning models for plant village dataset.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Proposed</th>
<th>[17]</th>
<th>[11]</th>
<th>[21]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>97.41</td>
<td>90.13</td>
<td>90.76</td>
<td>90.72</td>
</tr>
<tr>
<td>Precision</td>
<td>96.42</td>
<td>91.11</td>
<td>91.34</td>
<td>91.84</td>
</tr>
<tr>
<td>Recall</td>
<td>96.43</td>
<td>90.21</td>
<td>90.31</td>
<td>89.81</td>
</tr>
<tr>
<td>F1-score</td>
<td>96.42</td>
<td>91.69</td>
<td>91.32</td>
<td>90.24</td>
</tr>
</tbody>
</table>

From Figure 5, compared with other deep learning models, the proposed attention-based deep feature extraction had at least 7% higher accuracy for texture datasets.
Figure 5. Comparison of accuracy with other deep learning models for texture database.

The proposed solution significantly reduces feature extraction time by 48%, primarily due to fewer CNN layers, with the modified CNN functionality contributing the most to this reduction.

The time taken for the feature in the proposed solution in compared against other deep learning feature extraction methods and result is shown. Compared to other deep learning models, the proposed solution took less time due to use a lesser number of layers.

The accuracy of the SVM classifier with proposed feature extraction is tested for different SVM kernels and the result is given in Table 6.

Table 6. Accuracy of SVM Classifier with proposed feature extraction with different SVM Kernels.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Radial Basis Kernel</th>
<th>Linear Kernel</th>
<th>Polynomial Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>97.41</td>
<td>94.13</td>
<td>94.76</td>
</tr>
<tr>
<td>Precision</td>
<td>96.42</td>
<td>95.11</td>
<td>95.34</td>
</tr>
<tr>
<td>Recall</td>
<td>96.43</td>
<td>93.21</td>
<td>93.31</td>
</tr>
<tr>
<td>F1-score</td>
<td>96.42</td>
<td>94.69</td>
<td>94.32</td>
</tr>
</tbody>
</table>

The proposed texture features performed better with a radial-basis kernel of SVM compared to Linear and Polynomial kernels.

The performance of the proposed features along with the radial-basis kernel was measured for various values of $C$, and the result is given in Figure 6.

Figure 6. Time taken for each stage in proposed solution.

The proposed solution achieved an accuracy of 0.1 at $C = 0.1$, with no significant increase in accuracy after increasing $C$, as shown in Figure 7. The ROC curve plot in Figure 8 reveals a highest ROC area of 90.7%, indicating higher sensitivity. The modified CNN’s peak accuracy is achieved at epoch 10, indicating its superior sensitivity.
8 reveals a highest ROC area of 90.7%, indicating higher sensitivity. The modified CNN's peak accuracy is achieved at epoch 10, indicating its superior sensitivity.

Figure 7. Accuracy vs. C Value.

Figure 8. ROC Plot.

4. Conclusions

In this work, a deep learning algorithm for extracting texture features was proposed. The proposed method for extracting texture features can learn complex features with varying intensities across an image region. To accomplish this, proposal regions based on frequency domain analysis were selected, and an attention matrix was created from the proposal regions. The proposed features were able to provide at least 4% more accuracy when compared to existing works when the discriminative ability of proposed feature extraction was tested against various cases. In this study, attention vector generation was accomplished through unsupervised clustering. Future work will include improving it with application-defined cues through active learning.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: The author would like to thank the reviewers for their constructive comments, which improved the overall quality of this paper.

Conflicts of Interest: The author declares no conflict of interest.
References


12. Kociolek, M.; Koźlowski, M.; Cardone, A. A Convolutional Neural Networks-Based Approach for Texture Directionality Detection. Sensors 2022, 22, 562. [CrossRef]


16. Barburiceanu, S.; Meza, S.; Orza, B.; Malutan, R.; Terebes, R. Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture. IEEE Access 2021, 9, 160085–160103. [CrossRef]


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.