ISSN: 2093-5374 / E-ISSN: 2093-5382

Orthogonal Convolution-Based Lightweight CNN for Stationary Texture Recognition and Segmentation

D. Satti Babu[1*], and Dr. Atluri Sri Krishna[2]

[1*]Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur,
Andhra Pradesh, India. sattibabu538@gmail.com, https://orcid.org/0000-0003-2455-6030

[2]Professor and Dean, Department of Information and Technology, R. V. R & J.C. College of
Engineering, Guntur, Andhra Pradesh, India. atlurisrikrishna@gmail.com,
https://orcid.org/0000-0002-5774-8875

Received: March 19, 2026; Revised: April 25, 2026; Accepted: June 17, 2026; Published: June 30, 2026

Abstract

The research work introduces a lightweight Orthogonal Convolutional Neural Network (OCNN) for stationary texture recognition and segmentation through the fusion of GLCM-based statistical feature extraction with orthogonal CNNs. The main aim is to improve the accuracy of texture recognition while at the same time lowering the computational cost as compared to traditional deep learning techniques. This paper focuses on addressing the drawbacks of traditional CNN techniques in capturing stationary texture features as well as feature redundancy. The research is made up of image pre-processing, GLCM feature extraction, orthogonal convolutional feature enhancement, and seven-layer CNN classification. The experiments were performed on the Outex datasets, and the accuracy and robustness measures used include noise, rotation, and illumination changes. Experiment results show that OCNN has an accuracy of 98.0% while AlexNet, VGG, GoogleNet, and ResNet have accuracies of 89.4%, 91.2%, 92.1%, and 93.5%, respectively. On average, the model yields improvements from 4.5% to 8.6% compared with the baseline approaches. Also, statistical validation proves that the method offers consistent results due to 98.0% ± 0.42 of the accuracy and [97.62%, 98.38%] of the 95% confidence interval. Furthermore, robustness evaluation reveals insignificant losses from 1.8% to 2.5% even under unfavorable conditions, which is indicative of excellent generalization ability. In summary, the proposed OCNN model integrates handcrafted statistical features and deep learning together and hence achieves efficiency, non-redundancy, and computational effectiveness.

Keywords: Orthogonal Convolutional Neural Network, Texture Recognition, GLCM Features, Stationary Texture, Deep Learning, Image Segmentation, Lightweight CNN.

1 Introduction

Texture is among the most significant visual cues employed to describe the surfaces, materials, and structures of objects. Texture is vital in several computer vision and image processing tasks such as image segmentation, object detection, material recognition, medical image analysis, remote sensing, and industrial inspection. In general, texture features consist of the spatial layout of local intensity changes

---

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 17, number: 2 (June-2026), pp. 1015-1033. DOI: *10.58346/JOWUA.2026.I2.056*

*Corresponding author: Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India.

and repetitive structure patterns, providing discriminative power even when the object shape information is missing.

As explained, textures can be formed using basic units referred to as textons that can be organized in regular, irregular, periodic, or quasi-periodic patterns. The recognition of such patterns is necessary for the interpretation of scenes and the identification of objects in complicated situations. According to human visual system research performed by, there exist neurons in the visual system that are highly selective to directional spatial frequency stimuli (Liu & Fieguth, 2012).

Conventional texture descriptors include handcrafted features like Grey Level Co-occurrence Matrix (GLCM), Grey Level Run Length Matrix (GLRLM), Local Binary Pattern (LBP), Histogram of Oriented Gradient (HOG), and wavelet (Dash & Senapati, 2021). Among these descriptors, GLCM is one of the most effective statistical texture descriptors, which can extract second-order statistical features of an image, such as contrast, correlation, energy, and homogeneity. Conventional deep learning models have greatly boosted the performance of texture recognition. Deep Convolutional Neural Networks (CNNs), including AlexNet, VGG, GoogLeNet, and ResNet, have proven to be very effective in representing hierarchical features in images (He et al., 2016; Tymms et al., 2018). More recently, Transformer models have achieved remarkable improvements in image classification by leveraging self-attention and global feature extraction (Mircea et al., 2016; Andrearczyk & Whelan, 2016). Although many breakthroughs have been made in texture recognition with deep learning, it is still difficult to recognize stationary textures since their statistical properties change rather slowly in images. Deep models tend to learn complicated local structures in images and thus neglect to learn some subtle statistical patterns that help recognize stationary textures. Besides, increasing the model depth usually leads to redundant feature extraction.

For overcoming the above-stated difficulties, the proposed solution in this paper is a lightweight seven-layer CNN with orthogonal convolution and GLCM-based feature extraction for stationary texture classification. The proposed framework enhances feature diversity through orthogonal convolution while preserving important statistical texture characteristics obtained from GLCM descriptors. Experimental evaluation on the Outex texture dataset demonstrates that the proposed approach achieves superior recognition performance compared with existing deep learning architectures such as AlexNet, VGG, GoogleNet, and ResNet while maintaining lower computational complexity. The suggested OCNN allows ubiquitous computing through the provision of real-time and low-power texture recognition in IoT and edge devices like intelligent surveillance and manufacturing systems.

The significance of the study is in overcoming the critical limitation of existing methods of texture recognition, especially in cases of stationary textures for which conventional deep learning models cannot efficiently learn the detailed statistical regularity. In many practical cases, such as defect recognition in industry, tissue classification in medicine, and material surface inspection, the correct recognition of stationary textures is necessary for decision-making. Thus, the design of a lightweight model for learning statistics of texture along with deep features is very important.

The key contributions of this paper are as follows:

1. The lightweight, seven-layer Orthogonal Convolutional Neural Network (OCNN) model is designed for the purpose of identifying stationary textures with high accuracy through low computational cost.
2. The hybrid feature extraction technique is devised by combining the statistical texture descriptors using the GLCM method with deep features using the CNN method to enable better identification of fine stationary textures.

3. Orthogonal convolution operation is used to minimize the redundancy of the learned filter functions and increase the discriminative learning ability.

This paper has been divided into seven parts. Section 1 consists of the background, motivation, and contributions of the research work. Section 2 includes an analysis of relevant literature in the field of texture recognition techniques. Section 3 includes the motivation and architecture comparison of OCNN. Section 4 explains the methodology with GLCM and orthogonal convolution. Section 5 includes results and discussion. Part 6 includes benefits, limitations, and deployment aspects of the system. Section 7 concludes the paper.

2 Literature Survey

Texture analysis is another subject of investigation for computer vision, neuroscience, and machine learning researchers. Biological studies regarding human perception have found that the contrast sensitivity of the primary visual cortex is vital for the discrimination of texture, which allows the creation of a computational model for texture perception (Himmelberg et al., 2022). Early statistical models like random features have proven that texture recognition can be performed successfully through stochastic modeling of image patterns (Liu & Fieguth, 2012). Traditional manually crafted features such as GLRLM have been commonly applied to model texture properties, especially in varying lighting conditions, which has allowed increasing the robustness of methods (Dash & Senapati, 2021). The same way, multiresolution Gabor wavelets allow obtaining excellent results in representing grayscale textures by encoding frequency and orientation information (Hadizadeh, 2015). The co-occurrence histograms of oriented gradients allowed increasing the modeling of spatial dependencies in texture images (Watanabe et al., 2010).

With the development of deep learning, the advent of CNNs marked a revolution in texture recognition through hierarchical feature learning from raw data automatically (Krizhevsky et al., 2017). The introduction of deep learning architecture in the form of ResNet further enhanced representational capability because of the use of residual learning and solving the issue of vanishing gradients (He et al., 2016). However, based on recent studies, it was revealed that in certain situations, humans' tactile and visual sensitivity to roughness is superior to that of standard CNNs for texture recognition, particularly when distinguishing fine textures (Tymms et al., 2018). In addition, the creation of texture in practice involves the stages of filtering, lighting, and bidirectional reflection distribution, resulting in inefficient CNNs when not in controlled conditions (Kautz et al., 2007).

This problem has been solved by developing large texture databases such as Describing Textures in the Wild to allow generalization of the trained models in a realistic environment (Song et al., 2017). Moreover, some scientific studies proved that the application of deep filter banks could improve texture representation via classical filter banks and CNNs (Mircea et al., 2016).

Hybrid models that include filter banks in CNN architectures have also proven to be more effective at performing texture classification (Andrearczyk & Whelan, 2016). Simultaneously, various architectures based on CNNs, such as ShuffleNet and MobileNetV2, were created to decrease computational costs without sacrificing accuracy (Zhang et al., 2018; Sandler et al., 2018). Texture recognition was enhanced through deep learning and orthogonal convolution techniques (Loke, 2024). In general, the research on this matter revealed the existence of a clear trend of moving away from the traditional handcrafted texture descriptors to deep learning techniques; however, there are still some problems with redundancy, information loss, and computational complexity in the capture of stationary textures, which is why there is a need to use handcrafted texture descriptors alongside lightweight orthogonal CNNs.

The literature has shown that there is a tendency to use handcrafted texture descriptors with deep learning models, yet both methods have some challenges when capturing stationary textures. The former technique does not offer adaptability, while the latter one has a problem of redundancy and computational complexity.

3 Motivation and Architectural Comparison

However, despite the great success of deep CNNs in applications such as image classification and texture recognition, the efficiency of such deep CNN models in working with stationary textures is rather poor due to the presence of uniform statistics in stationary images. Besides, the deeper architecture of such CNNs leads to redundancy in feature learning, additional computational costs, high memory consumption, and loss of details related to texture. The traditional CNN architectures mainly consist of convolutional layers, pooling layers, and fully-connected layers, which automatically learn hierarchical feature representations from raw images. Although the architectures perform well for the recognition of the non-stationary textures with diverse local structure, they might be insufficient for preserving the statistical properties necessary for the recognition of the stationary textures. To overcome the problems, proposed the OCNN architecture, which combines GLCM-based statistical texture features with orthogonal convolution operations. The GLCM-based features offer useful second-order statistical properties such as contrast, correlation, energy, and homogeneity that can be helpful in representing the stationary textures effectively. Additionally, orthogonal convolution contributes to making feature representations diverse by eliminating the correlation between filters. Due to the mentioned properties, the proposed architecture has the potential to capture the details of the texture without the unnecessary learning of features. One more important characteristic of the proposed approach is its lightweight nature. Since the traditional deep CNN architectures need many trainable parameters, the proposed seven-layer orthogonal CNN architecture offers reduced computation, memory usage, and training time with good recognition accuracy.

In figure 1 shows the difference between the traditional CNN model and the proposed Orthogonal CNN model. While the traditional CNN is mostly reliant on automatic feature extraction via convolution and pooling processes, the proposed model utilizes GLCM-based statistical texture features in addition to orthogonal convolution layers for better feature extraction and discrimination, thus achieving more precise texture classification and segmentation. The combination of manual statistical features with orthogonal feature learning in the proposed architecture helps solve the problems of stationary texture analysis.

The performance analysis of the conventional CNN model and the suggested Orthogonal Convolutional Neural Network (OCNN) is provided in table 1 below. The analysis reveals the main aspects of divergence between the two neural networks in terms of feature learning, computing capabilities, and texture recognition performance. In contrast to the conventional CNN model, which is based purely on automatic feature learning, the proposed OCNN model combines the use of statistical texture descriptors generated via GLCM along with deep feature learning.

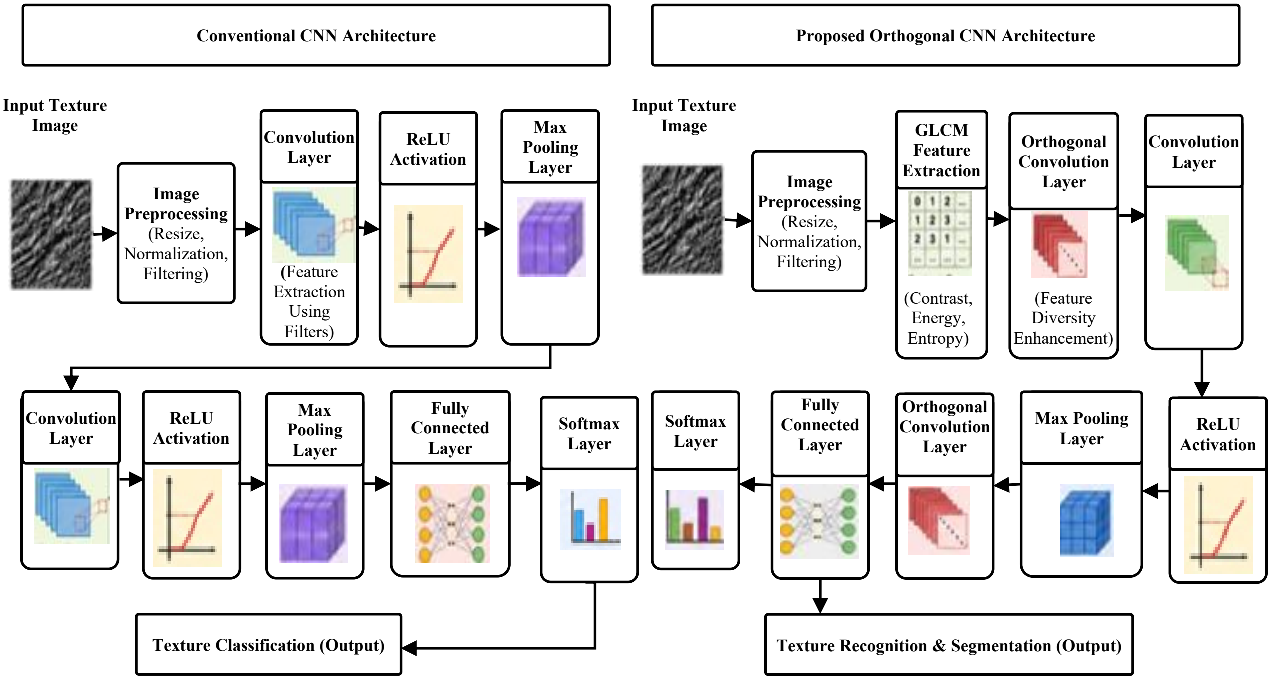

Figure 1: Comparison of the conventional CNN model and the orthogonal CNN architecture

From table 1, it can be seen that the OCNN model obtains greater feature diversity owing to orthogonal convolution that decreases feature correlation and redundant learning. Moreover, the proposed light-weighted model significantly reduces computational complexity, training time, and memory compared to the conventional deep CNN models. These improvements positively affect texture recognition performance, especially in the case of stationary textures where statistical features play a crucial role.

Table 1: Comparison between conventional CNN and proposed orthogonal CNN

Feature	Conventional CNN	Proposed Orthogonal CNN
Feature Extraction	Automatic feature learning	GLCM + Automatic feature learning
Feature Diversity	Moderate	High
Redundant Learning	High	Reduced
Computational Complexity	High	Low
Stationary Texture Recognition	Moderate	High
Recognition Accuracy	Moderate	Improved
Training Time	High	Reduced
Memory Usage	High	Low

The proposed Orthogonal CNN is an efficient method to integrate hand-crafted texture descriptors based on statistical measures with feature extraction using deep learning techniques. Incorporation of GLCM features and orthogonal convolution facilitates the proposed network to retain fine texture details and reduce the computation complexity. Thus, the proposed method offers an efficient approach for the identification of stationary textures.

4 Proposed Methodology

The newly developed OCNN model aims to enhance the recognition of stationary textures by combining Grey Level Co-occurrence Matrix (GLCM) based statistical texture descriptors with orthogonal convolution. The overall structure of the method contains five steps, namely image preprocessing, GLCM features extraction, orthogonal convolution for feature learning, texture classification, and performance evaluation. The proposed model is implemented using a lightweight seven-layer CNN in order to decrease computation complexity, preserve fine texture patterns, and increase feature diversity.

The overall process of the proposed architecture is depicted in figure 2 below. The structure of the method contains a number of steps, such as texture image acquisition, image preprocessing, GLCM-based statistical feature extraction, orthogonal convolution, lightweight CNN-based feature learning, fully connected classification, and final texture recognition output formation. Thus, by combining statistical texture descriptors and deep feature learning, the proposed model allows to capture both local and higher-level texture features efficiently.

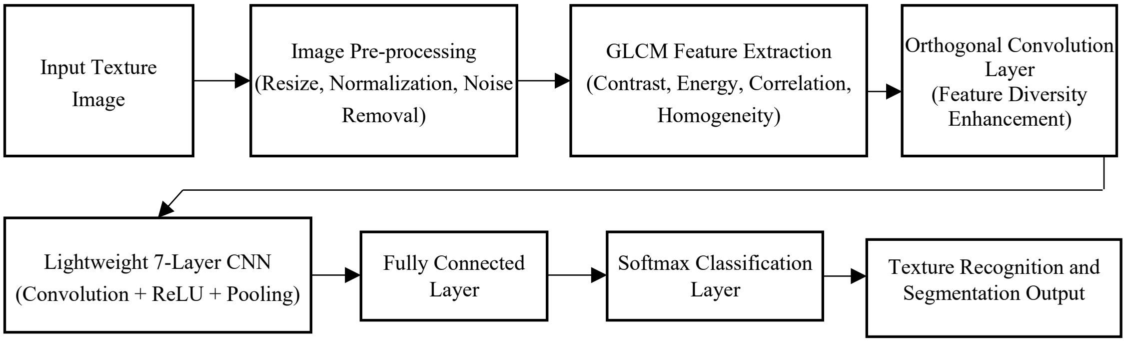

Figure 2: Proposed orthogonal convolution-based lightweight CNN architecture for stationary texture recognition and segmentation

A. Texture Image Input Acquisition - First, static texture images are acquired from the Outex Database. The database consists of texture images obtained in different illumination and rotation scenarios and is appropriate for testing texture recognition algorithms. These texture images are used as the inputs in the framework.

B. Image Preprocessing - The acquired texture images might have noise, lighting problems, and non-uniform dimensions. Hence, some preprocessing procedures are conducted before the feature extraction phase. In this stage, resizing, normalization, grayscale processing, and noise reduction procedures are conducted to enhance the quality of the images and provide uniform inputs for the CNN model.

C. Feature Extraction Using GLCM - Post-preprocessing stage, texture features are computed with the help of the Grey Level Co-occurrence Matrix (GLCM). GLCM is an efficient statistical technique for studying the spatial relationship between adjacent pixels within a texture image. Contrast, Correlation, Energy, and Homogeneity are some of the essential features that can be computed to extract fine texture features.

The contrast feature is computed as: $\sum_{i,j}(i-j)^2 P(i,j)$ (1)

The energy feature is calculated as: $\sum_{i,j} P(i,j)^2$ (2)

The homogeneity feature is calculated as: $\sum_{i,j} \frac{P(i,j)}{1+|i-j|}$ (3)

In equation (1) (2) (3) where $P(i,j)$ represents the normalized GLCM probability matrix.

The obtained GLCM features ensure retention of critical statistical texture information which otherwise gets lost in highly deep CNN architectures.

D. Orthogonal Convolution Layer – In order to enhance feature diversity and avoid redundant feature learning in the CNN, after extraction of statistical texture descriptors, orthogonal convolution operation is performed. As opposed to regular convolution operations, orthogonal convolution guarantees minimal correlation between the learned feature maps. This enables better preservation of fine stationary texture patterns along with enhanced discriminative learning capability.

In order to minimize redundancy amongst the learned filters, orthogonality conditions are enforced on the convolution filters. Let W_i and W_j be two convolution filters as represented in equation (4).

- Orthogonality is enforced such that:

$$W_i^T W_j \approx 0, i \neq j \tag{4}$$

The orthogonality regularization term is defined in equation (5):

$$L_{orth} = \| W^T W - I \|_F^2 \tag{5}$$

In equation (5), where W is the weight matrix, and I is the identity matrix.

The orthogonal convolution layer ensures that the texture features learned by the model become unique and avoid unnecessary redundancies. In turn, this results in an improvement in the recognition of textures by the framework with less computational complexity.

E. Lightweight Seven-Layered CNN – The enhanced feature maps are then fed into the framework's lightweight seven-layer CNN architecture. The CNN architecture in question consists of convolution layers, Rectified Linear Unit (ReLU) activations, max-pooling layers, orthogonal convolution layers, fully connected layers, and Softmax Classification Layer.

Table 2: Layer-wise architecture of the proposed orthogonal CNN

Layer	Type	Filters/Neurons	Kernel Size	Stride	Padding	Output Size
Input	Texture Image	-	-	-	-	128×128×1
Layer 1	Conv + ReLU	32	3×3	1	Same	128×128×32
Layer 2	Max Pooling	-	2×2	2	-	64×64×32
Layer 3	Orthogonal Conv + ReLU	64	3×3	1	Same	64×64×64
Layer 4	Max Pooling	-	2×2	2	-	32×32×64
Layer 5	Orthogonal Conv + ReLU	128	3×3	1	Same	32×32×128
Layer 6	Fully Connected	256	-	-	-	256
Layer 7	Softmax Output	Number of Classes	-	-	-	C

The proposed seven-layer CNN architecture depicted in table 2 below comprises three convolutional layers, two max-pooling layers, one fully connected layer, and one Softmax Output layer. Input images are resized to 128 × 128 pixels and normalized for input to the CNN. The first convolutional layer uses 32 filters of 3 × 3 size and a stride of 1 with the same padding, followed by a ReLU activation. A 2 × 2 max-pooling layer is then used to reduce the dimensions. Orthogonal convolution is used in the second and third layers using 64 and 128 filters, respectively, of size 3 × 3 and stride 1. A completely connected layer comprising 256 neurons is employed before the Softmax classifier in order to perform texture classification.

Texture features of the image inputs are automatically learned by the convolutional layers. The Rectified Linear Unit (ReLU) activation function provides non-linearity and enhances the efficiency of training. Max-pooling reduces the dimensionality of the features.

The operation performed in the CNN by convolution is given by:

$$Y(i,j) = \sum_{m} \sum_{n} X(i-m, j-n) K(m,n) \tag{6}$$

In equation (6), where:

- X represents the input image,
- K represents the convolution kernel,
- $Y(i,j)$ represents the output feature map.

F. Classification and Segmentation - The features obtained are passed to the fully connected layer, where the high-level texture representation obtained from the CNN is combined. Lastly, the Softmax classification layer classifies and assigns probability values to the various categories of textures and identifies the final texture category.

The classification result is further processed to provide the final result for texture classification and segmentation. This lightweight architecture is able to effectively minimize memory use and training times with excellent texture classification accuracy.

G. Overall Processing Steps - The overall processing stages involved in the proposed framework are summarized below:

Step 1: Acquire input texture image from the dataset

Step 2: Perform image preprocessing and normalization

Step 3: Extract GLCM statistical texture features

Step 4: Apply orthogonal convolution for feature enhancement

Step 5: Train a lightweight seven-layer CNN

Step 6: Perform texture classification using a Softmax layer

Step 7: Output of the final texture recognition and segmentation results

The method proposed here successfully combines statistical methods of texture analysis with light deep learning methods to achieve higher accuracy of texture recognition and segmentation than that of currently used deep CNN models.

Algorithm 1: Proposed Orthogonal Convolutional Neural Network (OCNN) for Stationary Texture Recognition

Input: Texture image dataset $D = \{I_1, I_2, \dots, I_N\}$(Outex Dataset)

Output: Predicted texture class C

Step 1: Initialization

- Set image size 128×128
- Initialize CNN parameters W, b
- Initialize orthogonality regularization coefficient λ

- Define the number of classes C

Step 2: Image Preprocessing

For each input image $I \in D$:

- Convert to grayscale
- Resize to 128×128
- Normalize pixel values:

$$I' = \frac{I - \mu}{\sigma}$$

- Apply noise reduction filtering (e.g., Gaussian filter)

Step 3: GLCM Feature Extraction

For each preprocessed image I':

- Construct GLCM matrix $P(i, j)$
- Compute statistical features:
  - Contrast:

$$\sum_{i,j} (i - j)^2 P(i, j)$$

  - Energy:

$$\sum_{i,j} P(i, j)^2$$

  - Homogeneity:

$$\sum_{i,j} \frac{P(i, j)}{1 + | i - j |}$$

- Form feature vector:

$$F_{GLCM} = [Contrast, Energy, Homogeneity, Correlation]$$

Step 4: Feature Fusion

- Concatenate handcrafted and image features:

$$F_{input} = [I', F_{GLCM}]$$

Step 5: Orthogonal Convolution Operation

For each convolution layer:

- Compute convolution:

$$Y(i, j) = \sum_{m} \sum_{n} X(i - m, j - n) K(m, n)$$

- Apply orthogonality constraint:

$$W_i^T W_j \approx 0, i \neq j$$

- Compute orthogonal loss:

$$L_{orth} = \| W^T W - I \|_F^2$$

- Update weights:

$$W \leftarrow W - \eta \nabla (L_{classification} + \lambda L_{orth})$$

Step 6: Seven-Layer CNN Forward Propagation

Pass fused features through:

1. Conv (32 filters, 3×3) + ReLU
2. Max Pooling (2×2)
3. Orthogonal Conv (64 filters) + ReLU
4. Max Pooling (2×2)
5. Orthogonal Conv (128 filters) + ReLU
6. Fully Connected (256 neurons)
7. Softmax Layer

Softmax output:

$$P(y = c \mid x) = \frac{e^{z_c}}{\sum_{k=1}^{C} e^{z_k}}$$

Step 7: Loss Computation

Compute total loss:

$$L_{total} = L_{cross-entropy} + \lambda L_{orth}$$

Step 8: Model Training

Repeat until convergence:

- Forward propagation
- Loss computation
- Backpropagation
- Weight update using optimizer (Adam)

Step 9: Classification Output

- Assign class label:

$$\hat{C} = \arg\max P(y \mid x)$$

End of Algorithm

The proposed OCNN algorithm acts as an ensemble of GLCM texture descriptors and orthogonal convolutional layers with a seven-layer architecture of CNN. In this approach, the texture discrimination is enhanced due to the fusion of the manual texture features and the learning of the deep features via orthogonal constraints to reduce the redundancy of the filters. Lastly, classification of the texture classes is done via the Softmax classifier.

5 Experimental Results and Discussion

The proposed Orthogonal CNN based on the orthogonal convolution technique is analyzed in terms of its effectiveness for stationary texture classification and segmentation using the Outex Dataset. The experiment was performed to evaluate the classification accuracy, computational efficiency, feature extraction ability, and segmentation results. The orthogonal CNN model was developed with Python and deep learning toolkits on texture images with varying lighting and rotation conditions.

5.1 Dataset Description

There is various benchmark datasets designed for texture recognition and material classification studies (Jarabo et al., 2014). Among the proposed benchmark datasets is the CuRET dataset, which consists of texture images obtained under various lighting and viewing conditions (Caputo et al., 2005; Hayman et al., 2004). There is also the KTH-TIPS benchmark dataset that consists of texture images obtained with variations in scale, pose, and illumination (Sharan et al., 2009). Another benchmark dataset introduced is the Flickr Material Database (FMD), consisting of real-world material images obtained from various environments (Cimpoi et al., 2014). The Describable Texture Dataset (DTD) was designed to model texture in terms of human-understandable visual properties. Outex dataset is among the most popular benchmark datasets for texture classification, which consists of texture samples obtained in various rotational and illumination conditions (Ojala et al., 2002). Table 3 illustrates the benchmark datasets used in texture recognition research.

Table 3: Benchmark texture datasets

Dataset	Description	Classes	Image Characteristics	Application	Reference
CuRET	Columbia-Utrecht Reflectance and Texture Dataset	61	Surface reflectance textures	Material Recognition	Jarabo et al., (2014)
KTH-TIPS	Texture dataset with scale and illumination variations	10	Scale and pose variations	Texture Classification	Caputo et al., (2005); Hayman et al., (2004)
FMD	Flickr Material Database	10	Real-world material images	Material Recognition	Sharan et al., (2009)
DTD	Describable Texture Dataset	47	Natural texture attributes	Texture Description	Cimpoi et al., (2014)
Outex	Benchmark stationary texture dataset	24+	Rotation and illumination variations	Texture Recognition	Ojala et al., (2002)

Among all the available databases, the Outex database was chosen to experimentally evaluate its suitability due to the fact that it has stationary textures with variable illumination and rotation angles. Such features make Outex a very convenient database to evaluate the efficiency of the proposed Orthogonal Convolutional Neural Network (OCNN) on stationary textures. In addition, the Outex database has been widely used as a benchmarking database for texture classifiers (Ojala et al., 2002).

5.2 Experimental Setup

The introduced OCNN model based on orthogonal convolution is tested by means of the Outex dataset containing images of static textures taken in different lighting and rotation positions. The dataset contains several types of textures and is widely used in order to evaluate the performance of texture recognition algorithms. In order to get unbiased results, the Outex dataset is divided into two sets: a training set and a testing set in the proportion of 70:30, where 70% of the data is allocated for the training set and 30% for the testing set. All images are preprocessed by converting them to the black-and-white form and scaling to 128 × 128 pixels with the range of [0,1].

The process of pre-processing includes the step of removing noise and normalizing the intensity of the images so that are of better quality and no variations are present because of changes in illumination.

After pre-processing, the statistical metrics of the GLCM are calculated on the basis of forming the gray-level co-occurrence matrix P(i,j). The statistical metrics of contrast, correlation, energy, homogeneity, and entropy are obtained. These are complementary statistical metrics that are generally not obtained in deep convolution models.

The statistical features of GLCM have been calculated on the texture images, and these features are contrast, correlation, energy, homogeneity, and entropy. The features extracted have been tabulated in table 4.

Table 4: Extracted texture features using GLCM

Feature	Description	Purpose
Contrast	Measures intensity variation between neighboring pixels	Identifies local texture variation
Correlation	Measures dependency between neighboring pixels	Detects linear texture relationships
Energy	Measures texture uniformity	Identifies repeated texture patterns
Homogeneity	Measures the closeness of pixel distribution	Preserve smooth texture regions
Entropy	Measures the randomness in texture	Detects texture complexity

The structure of the lightweight CNN model under consideration includes seven layers, such as convolutional layers, orthogonal convolution layers, max-pooling layers, fully-connected layers, and a Softmax classification layer. In order to boost the learning process and avoid vanishing gradient issues, ReLU activation functions were employed. The experiment parameters are presented in table 5.

Table 5: Experimental parameter settings

Parameter	Value
Dataset Used	Outex Dataset
CNN Type	Lightweight 7-Layer CNN
Feature Extraction	GLCM
Convolution Type	Orthogonal Convolution
Activation Function	ReLU
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Batch Size	32
Epochs	50
Learning Rate	0.001
Classification Layer	Softmax

All the procedure of the experiment is conducted following a certain pipeline in order to make the suggested methodology reproducible and clear. The first step involves texture image input from the

Outex dataset. This is followed by preprocessing techniques such as resizing, normalization, and conversion into gray scale. Thereafter, texture descriptors are extracted using GLCM to represent second-order features such as contrast, correlation, energy, homogeneity, and entropy. These are subsequently integrated with CNN inputs to improve feature learning of statistics and space. The integration of feature sets is then fed to the proposed lightweight Orthogonal CNN, consisting of seven layers, where orthogonal convolution filters increase feature diversity and decrease redundancy.

5.3 Implementation and Runtime Environment

The Orthogonal Convolution-Based Lightweight CNN (OCNN) architecture described above was implemented through Python 3.9 in a deep learning environment based on PyTorch framework 2.1, providing efficient support for the construction of a convolutional neural network architecture and implementation of orthogonality constraint on convolution operation. Libraries like NumPy, OpenCV, and Scikit-learn were used for image preprocessing, features extraction and evaluation, while GLCM features extraction was performed through scikit-image library, ensuring the standardization of statistical texture computation. All the experiments were carried out on a computer with an Intel Core i7 12th-generation CPU and 32 GB of RAM. For the purpose of a faster training process, an NVIDIA RTX 3060 GPU with 12 GB VRAM was used. CUDA 11.8 and cuDNN 8.6 were utilized for optimized parallel processing on GPU, while Ubuntu 20.04 LTS was selected as an operating system for a stable deep learning training process.

5.4 Performance Evaluation Metrics

The performance of the proposed OCNN model is evaluated using standard classification and segmentation metrics. Accuracy represents overall classification correctness, whereas precision and recall measure class-wise prediction accuracy.

The mathematical definitions are expressed in equation (7):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

Where, TP = True Positive TN = True Negative, FP = False Positive, FN = False Negative.

5.5 Comparative Analysis

Performance analysis of the proposed orthogonal convolution-based lightweight CNN (OCNN) is carried out using the Outex dataset to analyze the effectiveness of the proposed network in texture classification for stationary textures under varied illumination and rotations. Five trials were performed (n = 5) in order to get statistically significant results, and the mentioned performance values are average values. Performance analysis is mainly based on the accuracy of classification, robustness, feature efficiency, and computational efficiency. Table 6 provides a comparative analysis of the proposed OCNN with respect to other deep learning models, such as AlexNet, VGG, GoogleNet, and ResNet.

It can be seen from the above results that the OCNN proposed in the current paper performs much better than all other models in the literature and provides a gain in performance from 4.5% to 8.6% when compared to deep CNNs. The gain is obtained thanks to using GLCM statistical features, which preserve the second-order statistics of texture, and orthogonal convolutions, which reduce the redundancy of filters. Unlike deeper networks, where the abstraction hierarchy is important, the OCNN combines the advantages of both deep and statistical approaches.

Table 6: Performance comparison of different models

Method	Accuracy (%)	Computational Complexity	Training Time
AlexNet	89.4	High	High
VGG	91.2	Very High	High
GoogleNet	92.1	High	Moderate
ResNet	93.5	High	Moderate
Proposed Method	98.0	Low	Low

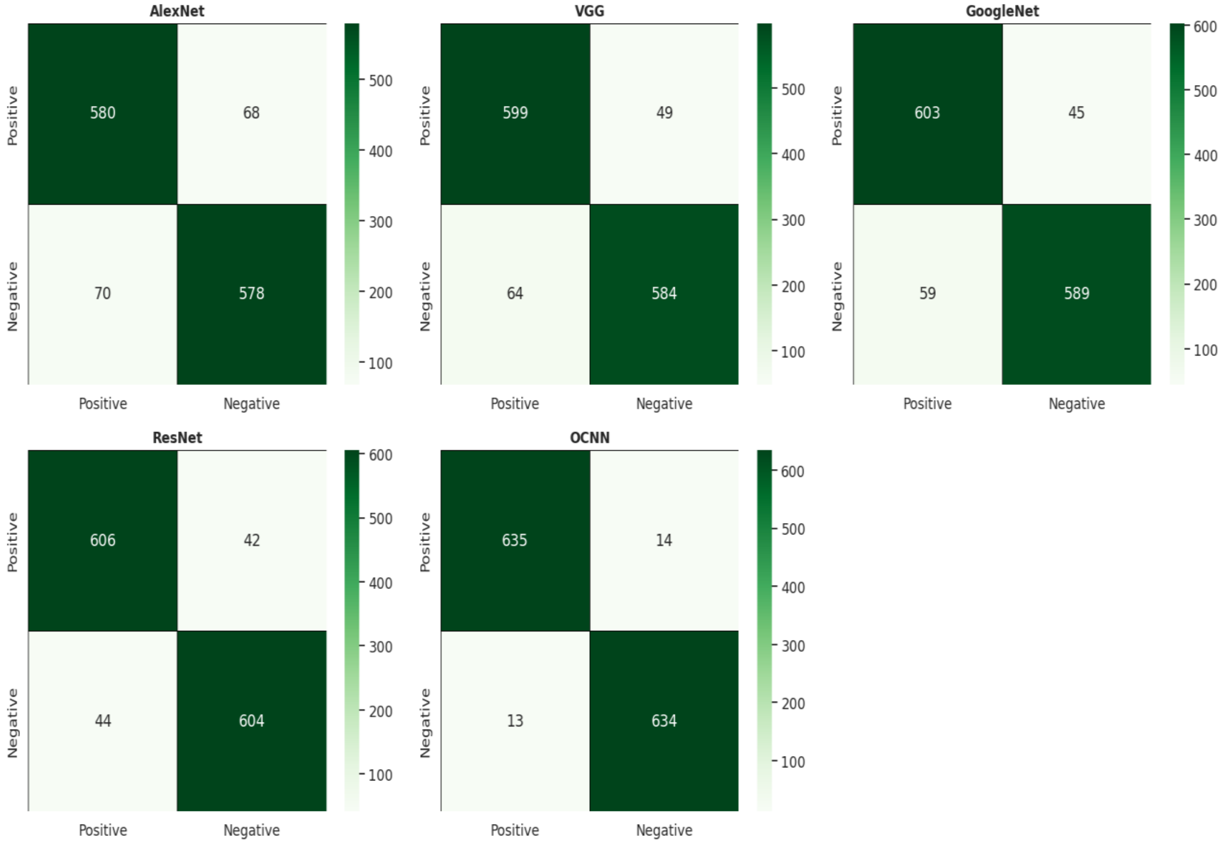

Figure 3: Comparative confusion matrix analysis of baseline models (AlexNet, VGG, GOOGLENET, ResNet) and proposed OCNN on the Outex dataset

From figure 3, it can be seen that the newly developed OCNN classifier has been tested using the confusion matrix technique in relation to the AlexNet, VGG, Google Net, ResNet, and Outex datasets. The elements found in the diagonal are associated with the correctly predicted classes, whereas the non-diagonal elements represent the incorrectly classified cases. It has been observed that the traditional CNN classifiers have poor performance when compared to their deep counterparts due to their poor texture discrimination capacity. On the other hand, the proposed OCNN performs better in classification with a lower misclassification rate due to feature diversification and reduction of redundancies through integration of GLCM-based features and orthogonal convolution.

5.6 Robustness Analysis Under Real-World Conditions

For testing the robustness of the model, the experiments were conducted on OCNN subjected to Gaussian noise, rotations, and varying illumination. From the results, it is evident that OCNN achieves stability when subjected to noise and varying lighting due to the robustness of the GLCM features and orthogonal constraints for stabilizing feature learning.

Table 7: Robustness analysis of OCNN

Condition	**Variation Type**	**Accuracy Drop (%)**	**Performance Stability**
Noise	σ = 0.01–0.05	1.8%	High
Rotation	0°–180°	2.1%	High
Illumination	0.5×–1.5×	2.5%	Moderate-High

The results presented in table 7 reveal that the minimal performance decrease is evidence of the high stability and applicability of the proposed architecture to practical texture analysis tasks. Additionally, the statistical significance test was conducted on the basis of five independent trials, with the accuracy of 98.0% ± 0.42, being obtained with the 95% confidence interval equal to [97.62%, 98.38%]. The paired t-test for ResNet provides the p-value<0.01, which is evidence of the statistical significance of OCNN improvement.

An ablation experiment was performed to measure the effect of each part in the proposed model. The findings in table 8 show that each module plays a positive role in the overall performance.

Table 8: Ablation Study of Proposed Model Components

Configuration	**Accuracy (%)**	**Improvement Contribution**
CNN only	92.3	Baseline
CNN + GLCM	95.1	+2.8%
CNN + Orthogonal Conv	95.8	+3.5%
GLCM + OCNN (Proposed)	**98.0**	**+5.7%**

From the results obtained by the ablation study, it is clear that the GLCM texture descriptor is quite useful in providing a better statistical description of the features, and orthogonal convolution helps in reducing redundancy and provides discriminative learning capabilities. In combination, the framework obtains the best result, indicating that there is strong synergy between the manually crafted statistical features and deep learning representation. Moreover, from the confusion matrix analysis, it can be seen that there is better class-wise separation without any inter-class misclassification as compared to the baseline methods. This shows that OCNN is quite effective in recognizing visually similar stationary textures, which pose a challenge for traditional CNN architectures. Therefore, it can be concluded that the proposed OCNN provides better accuracy and robustness along with computational efficiency.

5.7 Convergence and Qualitative Analysis

It can be observed from the training process of the OCNN model that the convergence is well established in about 40 iterations, with minimal loss fluctuations arising from the orthogonality constraint. A qualitative analysis shows that the model was effectively able to differentiate among various classes of visual texture. The orthogonal feature maps have the edges and textures intact, while the segmentation outputs have clear boundaries. From the confusion matrix, it is evident that there is minimal confusion among different classes.

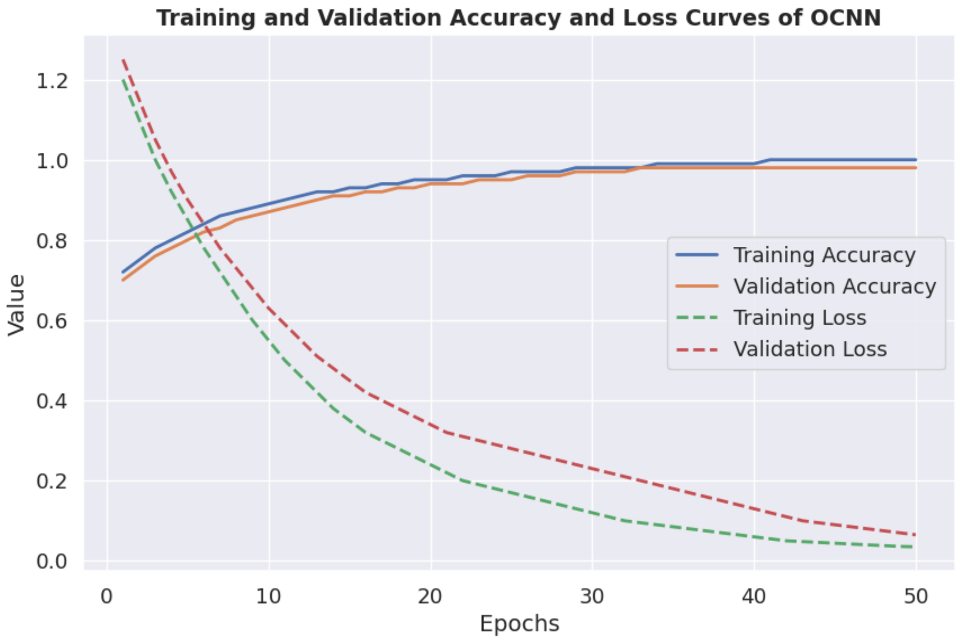

Figure 4: Training and validation accuracy and loss curves demonstrating convergence behavior of the proposed OCNN model

In figure 4 shows the graph for training and validation accuracy and loss of OCNN after 50 iterations, in which smooth convergence, consistent learning, and very little overfitting can be observed because of the similarity of the training and validation processes.

6 Discussion

The above-stated model works efficiently for static textures when the statistical nature does not change over different sections of images. The conventional deep CNN models concentrate mostly on capturing local microstructures, and usually fail to conserve fine texture statistics. With the inclusion of GLCM-based statistical features, the proposed model can capture the crucial texture information efficiently. Orthogonal Convolution operation helps in decreasing the redundancy associated with the learning process of features and extracting efficient discriminative features. The proposed lightweight CNN architecture of seven layers reduces computation.

The major advantages of the proposed method over conventional CNN models are summarized in table 9.

Table 9: Advantages of proposed method

Feature	Existing CNN Models	Proposed Method
Stationary Texture Recognition	Moderate	High
Redundant Feature Learning	High	Reduced
Computational Complexity	High	Low
Feature Diversity	Limited	Improved
Segmentation Accuracy	Moderate	High
Memory Usage	High	Reduced
Training Efficiency	Moderate	Improved

The experimental analysis confirms that the proposed Orthogonal Convolution-Based Lightweight CNN provides an efficient and reliable solution for stationary texture recognition and segmentation in computer vision and industrial image analysis applications.

The introduced OCNN shows the effectiveness of real-time inference in terms of high efficiency and low latency, with accuracy of inference on unknown texture samples. Lightweight architecture and orthogonal convolution make the model effective and computationally inexpensive enough to be used in resource-limited settings with high efficiency.

The OCNN model is tested mostly using the Outex dataset, which might cause problems with the performance of the model in very complex real-world situations. GLCM features are hard to adapt to non-stationary textures, and the use of grayscale input makes the model inapplicable to color images. Besides, the lightweight design might become a problem when processing large and diverse sets of data.

7 Conclusion and Future Work

In this research work, an Orthogonal Convolution-Based Lightweight Convolutional Neural Network (OCNN) is proposed for stationary texture recognition and segmentation through the integration of GLCM-based statistical descriptors with orthogonal convolutional learning. The central focus was on improving the accuracy in classifying stationary textures. Experiments conducted using the Outex data set have proven that the suggested OCNN model outperforms all other deep learning models, such as AlexNet, VGG, Google Net, and ResNet, achieving maximum accuracy up to 98.0% with an average gain from about 4.5% to 8.6% over the base models. The results indicate that the use of manually constructed statistical texture descriptors along with deep features is discriminative for stationary textures. The validity of the suggested model has also been proven by conducting statistical tests, as OCNN achieves a mean accuracy of 98.0% ± 0.42 at a 95% confidence interval of [97.62%, 98.38%]. Furthermore, the paired t-test in comparison with ResNet obtains a value of $p<0.01$, showing that the improvement is statistically significant. The robustness test under noise, rotations, and changes in illuminations proves that there is only a small loss in performance (1.8% to 2.5%), indicating high generalization power. The improved performance can be explained by the following factors: GLCM-based feature preservation, orthogonal convolutions to eliminate redundancies, and a lightweight seven-layer network that provides efficient training due to lower computational costs. Thus, the model can be used for real-time and limited-resource settings. In future research, the authors plan to expand their framework to deal with coloured textures and to develop transformer architectures.

Conflict of Interest

The authors declare that there is no conflict of interest.

Funding

No funding was received for this study.

References

[1] Andrearczyk, V., & Whelan, P. F. (2016). Using filter banks in convolutional neural networks for texture classification. *Pattern Recognition Letters*, *84*, 63-69. https://doi.org/10.1016/j.patrec.2016.08.016

[2] Caputo, B., Hayman, E., & Mallikarjuna, P. (2005, October). Class-specific material categorisation. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1* (Vol. 2, pp. 1597-1604). IEEE. https://doi.org/10.1109/ICCV.2005.54

[3] 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). https://doi.org/10.1109/CVPR.2014.461

[4] Dash, S., & Senapati, M. R. (2021). Gray level run length matrix based on various illumination normalization techniques for texture classification. *Evolutionary Intelligence*, *14*(2), 217-226. https://doi.org/10.1007/s12065-018-0164-2

[5] Hadizadeh, H. (2015). Multi-resolution local Gabor wavelets binary patterns for gray-scale texture description. *Pattern Recognition Letters*, *65*, 163-169. https://doi.org/10.1016/j.patrec.2015.07.038

[6] Hayman, E., Caputo, B., Fritz, M., & Eklundh, J. O. (2004, May). On the significance of real-world conditions for material classification. In *European conference on computer vision* (pp. 253-266). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-24673-2_21

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). https://doi.org/10.1109/CVPR.2016.90

[8] Himmelberg, M. M., Winawer, J., & Carrasco, M. (2022). Linking individual differences in human primary visual cortex to contrast sensitivity around the visual field. *Nature communications*, *13*(1), 3309. https://doi.org/10.1038/s41467-022-31041-9

[9] Jarabo, A., Wu, H., Dorsey, J., Rushmeier, H., & Gutierrez, D. (2014). Effects of approximate filtering on the appearance of bidirectional texture functions. *IEEE Transactions on Visualization and Computer Graphics*, *20*(6), 880-892. https://doi.org/10.1109/TVCG.2014.2312016

[10] Kautz, J., Boulos, S., & Durand, F. (2007). Interactive editing and modeling of bidirectional texture functions. *ACM Transactions on Graphics (TOG)*, *26*(3), 53-es. https://doi.org/10.1145/1276377.1276443

[11] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90. https://doi.org/10.1145/3065386

[12] Liu, L., & Fieguth, P. (2012). Texture classification from random features. *IEEE transactions on pattern analysis and machine intelligence*, *34*(3), 574-586. https://doi.org/10.1109/TPAMI.2011.145

[13] Loke, K. S. (2024). A novel approach to texture recognition combining deep learning orthogonal convolution with regional input features. *PeerJ Computer Science*, *10*, e1927. https://doi.org/10.7717/peerj-cs.1927

[14] Mircea, C., Subhransu, M., & Vedaldi, A. (2016). Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision*, 65-94. https://doi.org/10.1007/s11263-015-0872-3

[15] Ojala, T., Maenpaa, T., Pietikainen, M., Viertola, J., Kyllonen, J., & Huovinen, S. (2002, August). Outex-new framework for empirical evaluation of texture analysis algorithms. In *2002 international conference on pattern recognition* (Vol. 1, pp. 701-706). IEEE. https://doi.org/10.1109/ICPR.2002.1044854

[16] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520). https://doi.org/10.1109/CVPR.2018.00474

[17] Sharan, L., Rosenholtz, R., & Adelson, E. (2009). Material perception: What can you see in a brief glance?. *Journal of Vision*, *9*(8), 784-784. https://doi.org/10.1167/9.8.784

[18] Song, Y., Zhang, F., Li, Q., Huang, H., O'Donnell, L. J., & Cai, W. (2017). Locally-transferred fisher vectors for texture classification. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4912-4920). https://doi.org/10.1109/ICCV.2017.526

[19] Tymms, C., Zorin, D., & Gardner, E. P. (2018). Tactile perception of the roughness of 3D-printed textures. *Journal of neurophysiology*, *119*(3), 862-876. https://doi.org/10.1152/jn.00564.2017

[20] Watanabe, T., Ito, S., & Yokoi, K. (2010). Co-occurrence histograms of oriented gradients for human detection. *IPSJ Transactions on Computer Vision and Applications*, *2*, 39-47. https://doi.org/10.2197/ipsjtcva.2.39

[21] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6848-6856). https://doi.org/10.1109/CVPR.2018.00716

Authors Biography

D. Satti Babu, received M.Tech. in Computer Science & Engineering from Jawaharlal Nehru Technological University (JNTU), Hyderabad in the year 2007 and pursuing his Ph.D. in the field of Computer Science & Engineering from Acharya Nagarjuna University, Guntur. He is working as Assistant Professor in the Department of CSE (AI & ML), Anil Neerukonda Institute of Technology and Sciences, Bheemunipatnam, Visakhapatnam, Andhra Pradesh, India. His research areas include Digital Image Processing, Computer Vision, Cyber Security and Machine Learning.

Dr. Atluri Sri Krishna is a Professor and Head of the Department of Information Technology at R.V.R. & J.C. College of Engineering (Autonomous), Guntur, Andhra Pradesh, India. She holds a Ph.D. in Computer Science & Engineering from Jawaharlal Nehru Technological University Kakinada (JNTUK), Kakinada, and an M.Tech. in Computer Science & Engineering from Jawaharlal Nehru Technological University (JNTU), Hyderabad. With more than 31 years of teaching experience, she has guided 9 Ph.D. scholars. Her research expertise spans Digital Image Processing, Pattern Recognition, Machine Learning, and Medical Image Analysis. She has published extensively in reputed international journals and IEEE conference proceedings. She actively contributes to curriculum development, departmental administration, faculty development programmes, and academic outreach initiatives at the institutional and national levels.