



# Texture classification for visual data using transfer learning

Vinat Goyal<sup>1</sup> · Sanjeev Sharma<sup>1</sup>

Received: 10 March 2022 / Revised: 1 June 2022 / Accepted: 19 November 2022 /

Published online: 10 December 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

The texture is the most fundamental aspect of a picture that contributes to its recognition. Computer vision challenges such as picture identification and segmentation are built on the foundation of texture analysis. Various images of satellite, forestry, medical etc. have been identifiable because of textures. This work aims to offer texture classification models that will outperform previously presented methods. In this work, transfer learning was applied to attain this goal. MobileNetV3 and InceptionV3 are the two pre-trained models employed. Brodatz, Kylberg, and Outex texture datasets were used to evaluate the models. The models achieved excellent results and achieved the objective in most cases. Classification accuracy obtained for the Kylberg dataset were 100% and 99.89%. For the Brodatz dataset, the classification accuracy obtained was 99.83% and 99.94%. For the Outex datasets, the classification accuracy obtained was 99.48% and 99.48%. The model outputs the corresponding label of the texture of the image.

**Keywords** Texture classification · Computer vision · Transfer learning · MobileNetV3 · InceptionV3 · Deep learning

## 1 Introduction

The texture is the fundamental quantity of an image that aids in its identification. Texture analysis forms the foundation for computer vision problems like image recognition, image retrieval [37] and segmentation. Various images of satellite [30], forestry [27], medical [10], etc have been identifiable because of textures in them. The texture of an object provides important insights into the properties and behaviour of these objects. These insights later help in the computer vision tasks related to such objects when their shape doesn't help. Texture today is one of the key components in the analysis of images. This makes the task of

---

✉ Vinat Goyal  
vinatgoyal19@cse.iiitp.ac.in

Sanjeev Sharma  
sanjeevsharma@iiitp.ac.in

<sup>1</sup> Indian Institute of Information Technology, Pune, India

texture classification important. For the past years, there has been a lot of effort to develop models that can identify and classify these textures efficiently.

Classic machine learning approaches used for this task include using hand-engined features to extract information and using a statistical algorithm like SVM in the final layer for classification [43]. These approaches were previously preferred, but in recent times these approaches have been outperformed by deep learning methods, particularly convolutional neural networks. After the win of AlexNet [18] in the 2012 ImageNet large-scale visual recognition challenge, there has been an exponential growth in the usage of convolutional neural networks for image classification tasks. Today, significant models in computer vision for tasks like image classification, segmentation, recognition, etc., use convolutional neural networks.

CNN's learn feature vectors with weight sharing and local connectivity, which detects patterns at all locations in the image. Initial layers of a CNN learn simple features like the edges, and the deeper ones learn more complex features. CNN's can learn texture patterns of various complexity and scales. Novel convolution neural network models have better performance than the classic machine learning algorithms. This paper aims to propose models that would perform better than the previously proposed models and improvise the texture classification approach.

This paper proposes a transfer learning approach for the texture classification problem. Transfer learning is an approach wherein the intuition uses the knowledge gained while learning to classify classes of one dataset to a different data set of related problems. Transfer learning aims to focus on leveraging labelled data from one feature space to enhance the classification of other entirely different learning spaces. This approach works well when the source dataset (on which the model is trained) and the target dataset (the one in the study) are of a similar domain, making their feature spaces similar. In transfer learning, the top layer of the pre-trained model is replaced by a new layer with the number of neurons equal to the number of classes of the target dataset.

There are two types of transfer learning approaches. The first is feature extraction, wherein only the top layer is trained on the target dataset, freezing the rest of the dataset. The frozen layers are used as feature extractors on the target dataset, training only the top layer. The idea is that a feature vector trained on one kind of data set can extract valuable features on another data set. The second type is the fine-tuning of the model wherein only a few or none of the layers are frozen, and the rest of the layers along with the top layer are trained on the target dataset.

Transfer learning helps leverage the knowledge learnt by a model on one data set to extract information on another data set. Transfer learning also reduces the time of learning all the weights of the convolution layer. Using knowledge of a pre-trained model might also help in complete learning of the problem task compared to building a model from scratch. The pre-trained models used in this paper are MobileNetV3 and InceptionV3. The presented work focusses on:

- Study about transfer learning on texture datasets.
- Achieving better results on the provided benchmark datasets than previous work on the same datasets.

The rest of the paper is organised as follows. Section 2 discuss the literature survey of the related work. Section 4 cover the study of material and methods. Section 4 presents the experiments and results. At last, we are concluding work in Section 5.

## 2 Literature review

There has been a lot of research dedicated to texture analysis owing to the importance it holds in the field of computer vision. In 1993, [29] used two powerful algorithms, Principal Component Analysis and Multiscale Autoregressive models, on the Brodatz dataset. The variety of homogenous and non-homogenous images studied in this paper was more significant than those in the previous work. This approach got better results than the models proposed before it. In 1994 an energy-based approach was proposed in [38]. This model got an accuracy of over 90 for the classification of images.

Statistical methods are considered one of the earliest methods for texture analysis of the image, which have given good results on standard texture datasets. Ramola et al. [31] discusses the different statistical approaches like grey level concurrence matrix (GLCM), Local binary pattern(LBP), auto-correction function(ACF) and histogram pattern. Their research and discussion concluded that GCLM is the best approach for texture analysis. The major drawback of the GLCM model is the high matrix dimensionality and high correlation between harlick features. Feng et al. [9] and [5] have also implemented such statistical models on standard data sets and got good results.

Xu et al. [42] proposed a novel robust texture descriptor on variance in rotation, scale and illumination, which combines the dominant orientation analysis and multifractal analysis based on the Gabor filter. This approach was then implemented on the Brodatz and Outex datasets.

Sana and Islam [32] proposed power-law transform (PLT) to extract new spectral texture features. This technique outperformed the widely used Gabor features. As seen, machine learning approaches have had excellent results on standard datasets for texture analysis. However, these algorithms require handmade features for feature extraction. Also, such models cannot be used for feature extraction of images of another dataset, as seen in deep learning architectures with the help of transfer learning.

Zheng et al. [44] proposed an eight feature learning model alongside a deep learning perceptron based architecture. This paper showed the deep learning model's advantage over the other model. In recent years, convolutional neural networks have surpassed the standard artificial neural networks in the field of computer vision. CNNs have also revolutionised other fields like natural language processing, image and video recognition, information retrieval, grayscale colourisation, and multi-dimensional data processing and have surpassed many machine learning algorithms. Y LeCun proposed CNNs, Boser [21] (1989), three decades ago but did not get popular then because of lack of data and computational power. Today there is abundant data available, the computational power of computers has drastically increased, and there has been a lot of development in developing better optimisation algorithms. Algorithms like stochastic gradient descent with momentum (SGDM) and RMSprop have emerged as the favourites for optimisation. All these factors have contributed to the success of CNNs today[23]. Many CNN architectures such as AlexNet [17], VGG [36], ResNet [11], MobileNet [33], etc have emerged and are being used widely.

Simon and Vijayasundaram [35] Proposed a standard convolution neural network for the task of classification of images of flower and KTH data sets. This paper achieved excellent results as compared to its predecessors. A modified version of CNN is proposed in [2] called T-CNN, which is built on the intuition that the overall shape information extracted by the fully connected layers of a classic CNN is of minor importance in texture analysis. Therefore, an energy measure from the last convolution layer is pooled, connected to a

fully connected layer. This idea was inspired by the classic neural networks and filter bank approach. Jain et al. [13] proposed an Optimal Probability-Based Deep Neural Network (OP-DNN) for multi-type skin disease prediction and achieved an accuracy of 95%.

Dixit et al. [6] proposes another approach to classification where whale optimisation algorithm (WOA) is used along with the CNN. Results of this model on the Kylberg, Brodatz and Outex datasets are compared to the results obtained by other models on the same data set. This model gained excellent results and beat other models in comparison. Another such work was [14] where the authors used a new optimisation module Knowledge-Based-Search (KBS), along with Moth–Flame Optimization (MFO). Their work performed well in a dynamic environment.

As discussed, Deep Neural Networks require a large amount of data. When trained on a small dataset, their generalisation performance is limited. Liu et al. [22] proposes the use of relative position network (RPN) and relative mapping network (RMN) for skin lesion image classification with a small dataset. They were able to achieve an accuracy of 85%. Deep learning architectures have the advantage that one model trained on a vast dataset can extract features of images from another dataset. This approach is called transfer learning.

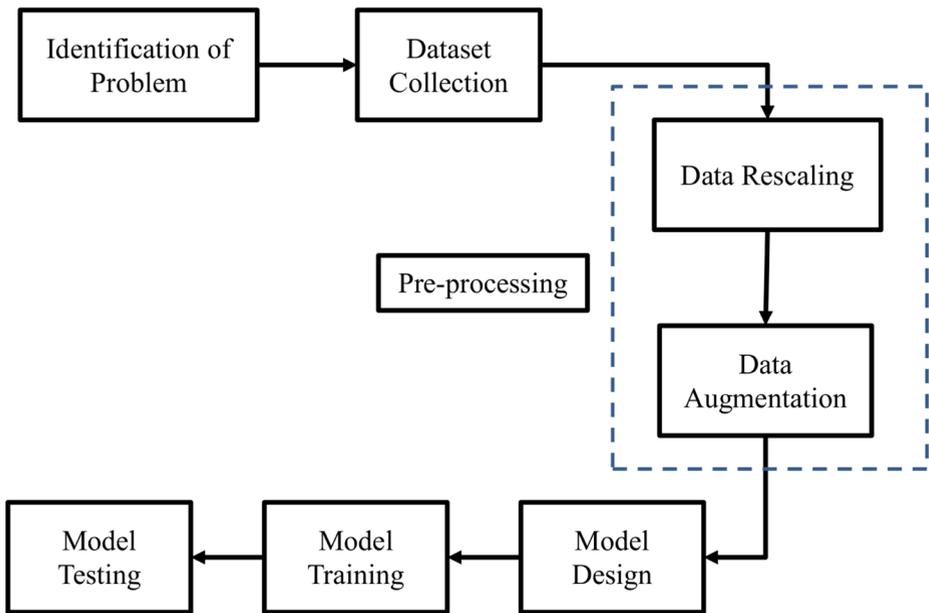
Kazi and Panda [16] uses the transfer learning technique to determine three different types of fruits and their relative freshness and got great results. Kundo et al. [19] proposes a bagging ensemble of three transfer learning models, InceptionV3, ResNet34 and DenseNet201, that outperformed the state of the art methods by 1.56%. Nadeem et al. [26] uses transfer learning for Pakistani traffic-sign recognition. They use a model trained on the German traffic-sign recognition, and with additional pre-processing and regularisation, they achieved competitive results on a small available dataset. This paper uses the approach of transfer learning. Transfer learning has also been widely employed in the medical domain. Arora et al. [3] used a transfer learning-based approach for detecting COVID-19 ailment in lung CT scan. They achieved a precision of 100% using the MobileNet architecture on the SARS-COV-2 CT-Scan dataset.

In recent years transformer-based architectures have revolutionised every domain of deep learning. A transformer-based architecture was originally proposed in [41] where authors proposed an attention mechanism based architecture, dispensing with recurrence and convolutions entirely. Their model was experimented on two translation tasks and outperformed the other models in terms of results and training time. Dosovitskiy et al. [7] proposed Vision Transformers (ViT) inspired from the transformer architectures for Natural Language Processing (NLP) tasks. Their study showed that ViT outperformed the conventional convolutional networks in terms of results and training time on standard datasets like the ImageNet.

The following sections of this paper discuss the materials and methods used and the experiments and results obtained. The last section summarises the paper and talks about the future scope.

### 3 Materials and methods

Figure 1 depicts the flowchart followed. The first step was to find the problem statement. The following step was to collect the related dataset to the problem statement. After the data was collected, it was preprocessed to make it of desirable format and size. The pre-processing stage also included data augmentation, which was done to avoid over-fitting the model. After pre-processing, models were designed for the problem statement, then tested on the pre-processed



**Fig. 1** Flow graph

dataset. Transfer learning models are used to classify the different datasets collected. We use the MobileNetV3 and the InceptionV3 models for the classification task.

### 3.1 Dataset

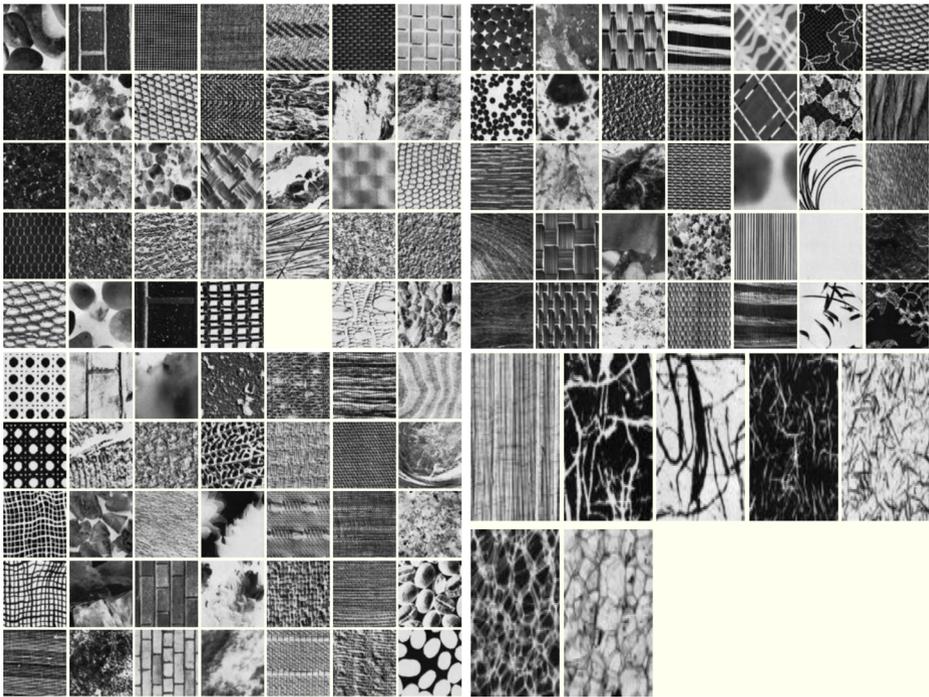
We have used three standard benchmark datasets of the texture classification problem. These are the Brodatz dataset, Kylberg dataset and the Outex dataset. Below is the summary of these datasets.

#### 3.1.1 Brodatz dataset

Brodatz dataset [4] is a very popular dataset for texture classification problems. The dataset has been referred from the University of Southern California[]. The original dataset did not contain the rotated images. In this paper, we have proposed these rotations using 40 different rotation angles on these images. This dataset has 112 classes. The samples of this dataset are displayed in Fig. 2. The summary of this dataset is given in Table 1.

#### 3.1.2 Kylberg dataset

The Kylberg dataset is another widely used dataset for texture classification problems. This dataset has 2 versions (1) with rotation patches and (2) without rotation patches [20]. We have used v1.0, which is the version without rotation patches. The classes of this dataset are blanket1, blanket2, canvas1, ceiling1, ceiling2, cushion1, floor1, floor2, grass1, lentils1, linseds1, oatmeal1, pearlsugar1, rice1, rice2, rug1, sand1, scarf1, scarf2, screen1, seat1, seat2, sesameseeds1, stone1, stone2, stone3, stoneslab1 and wall1. The samples of this dataset are displayed in Fig. 3. The summary of this dataset is given in Table 2.



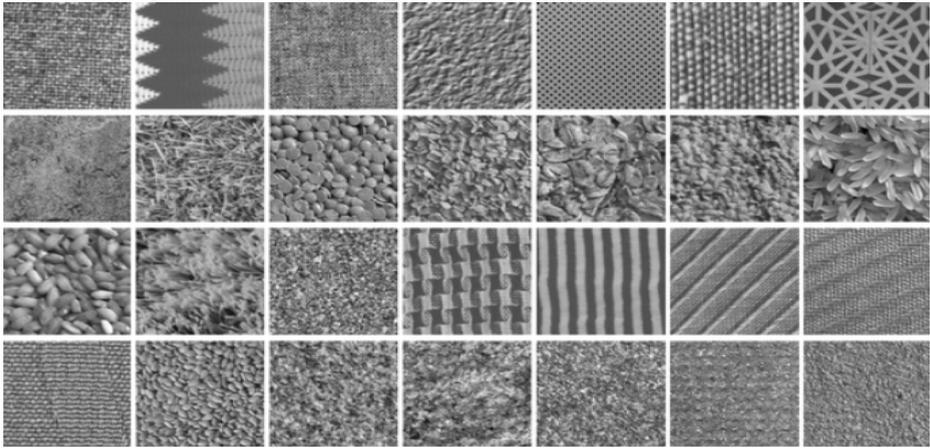
**Fig. 2** Samples of the Brodatz dataset

### 3.1.3 Outex dataset

The Outex [28] database has a lot of datasets. We are using the Outex\_TC.00012 dataset of this database. We have referred to this dataset from the University of OULU. The classes of this dataset are canvas001, canvas002, canvas003, canvas005, canvas006, canvas009, canvas011, canvas021, canvas022, canvas023, canvas025, canvas026, canvas031, canvas032, canvas033, canvas035, canvas038, canvas039, tile005, tile006, carpet002, carpet004, carpet005 and carpet009. The samples of this dataset are displayed in Fig. 4. The summary of this dataset is given in Table 3.

**Table 1** Summary of the Brodatz dataset

Features	Value
Number of Classes	112
Number of samples/ class	40
Total number of samples	4480
Texture patch size	640*640pixels
Format of image	8 bit grey scale PNG
Total size of dataset	1.02 GB



**Fig. 3** Samples of the Kylberg dataset

### 3.2 Data preprocessing and splitting

Data preprocessing is one of the most critical steps. This step makes the raw data compatible with the deep learning model. Images in the Outex and Brodatz datasets are in GIF format and converted to compatible models.

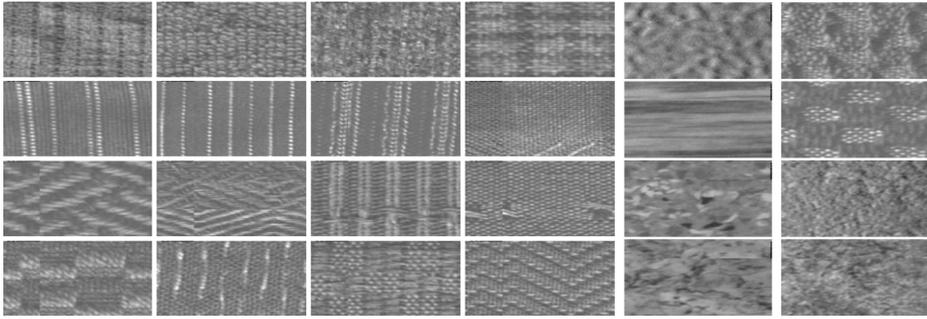
After the data is converted to a compatible format, the images are then resized to a size of  $224 \times 224 \times 3$ , making it compatible with the pre-trained model. After preprocessing, the data is split. The Kylberg, Outex\_TC\_00012 and the brodatz dataset are split in a ratio of 80:20 into training and testing data.

### 3.3 Data augmentation

As discussed earlier, data augmentation refers to the act of creating more data out of the already existing data. The intuition is that the image of the surface of texture rotated by an angle or flipped along an axis remains the image of that surface. Since there is only 1 image available for each class in the Brodatz dataset, more images are produced by rotating the original images by different angles. Data augmentation is also done for all the data of all three datasets. Figure 5 shows a sample of data received after subjecting the data of the Kylberg dataset to data re-scaling and data augmentation. Figure 6 shows a sample of

**Table 2** Summary of the Kylberg dataset

Features	Value
Number of Classes	28
Number of samples/ class	160
Total number of samples	4480
Texture patch size	$576 \times 576$
Format of image	8 bit grey scale PNG
Total size of dataset	1.76 GB



**Fig. 4** Samples of the Outex dataset

data received after subjecting the data of the Brodatz dataset to data re-scaling and data augmentation. Figure 7 shows a sample of data received after subjecting the data of the Outex dataset to data re-scaling and data augmentation.

### 3.4 Proposed model

This paper uses the methodology of using pre-trained models called transfer learning. Intuition uses the knowledge gained by a model on one problem to solve another similar problem. This methodology reduces the time spent on training a model from scratch. Also, using a pre-trained model might be able to learn the problem entirely compared to a model trained from scratch. This paper uses the MobileNetV3 and InceptionV3 models. For each approach, the last dense layer (classification layer) of the pre-trained model is replaced with a softmax layer suitable for classifying the texture of classes of that dataset. In this work, the following transfer learning techniques were implemented (Fig. 8):

- Feature extraction: Here, we froze all the model layers and trained only the added dense layer. Here the pre-trained model is only used as a feature extractor for the classifier.
- Full fine-tuning: Here, the whole pre-trained model was fine-tuned using the data in use.

#### 3.4.1 Transfer learning

It becomes difficult to collect enough data to build a model from scratch in many world applications. In such scenarios, the idea of transfer learning comes in. As discussed earlier, transfer learning is an approach wherein a model trained on a vast data set is used to solve

**Table 3** Summary of the Outex dataset

Features	Value
Number of Classes	24
Number of samples/ class	40
Total number of samples	960
Texture patch size	128*128
Format of image	8 bit grey scale GIF
Total size of dataset	0.16 GB

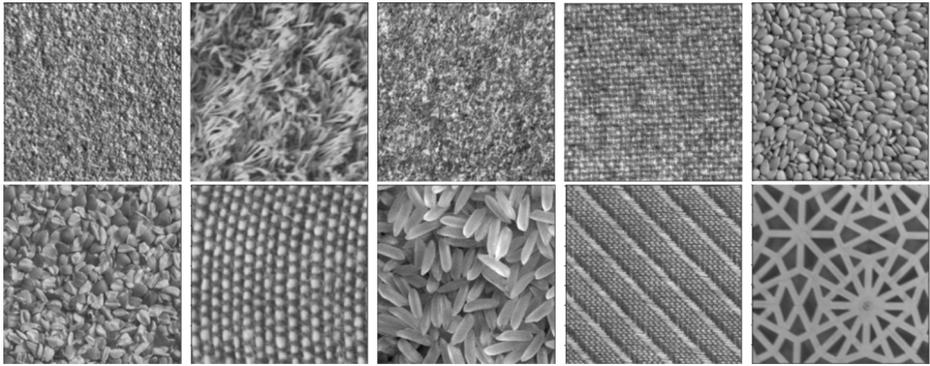


Fig. 5 Sample images of Kylberg data set after pre-processing and data augmentation

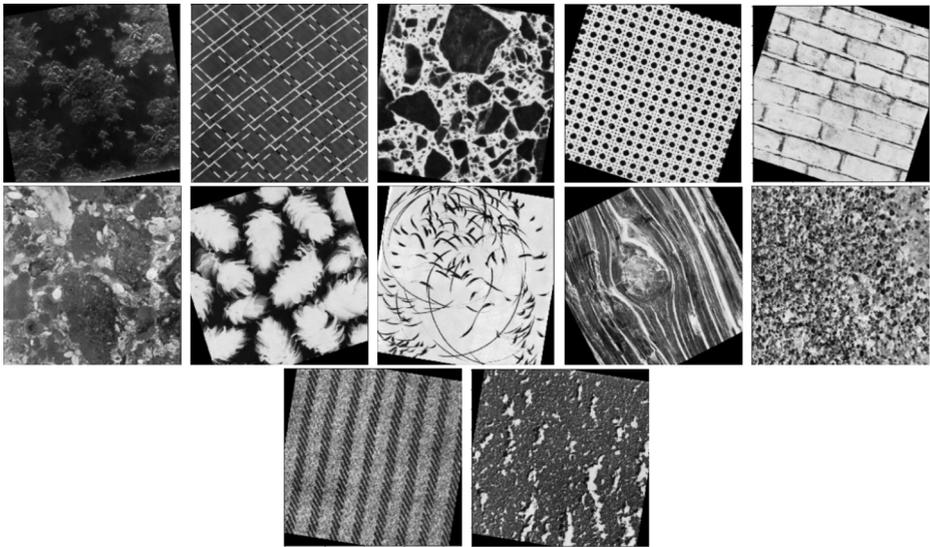


Fig. 6 Sample images of Brodatz data set after pre-processing and data augmentation

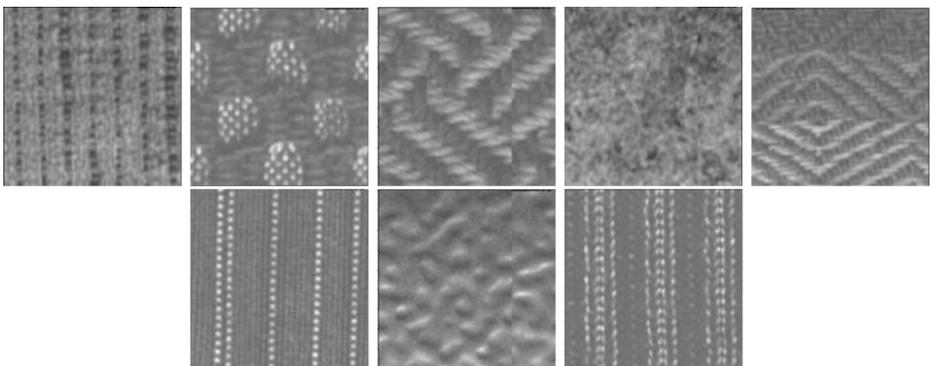
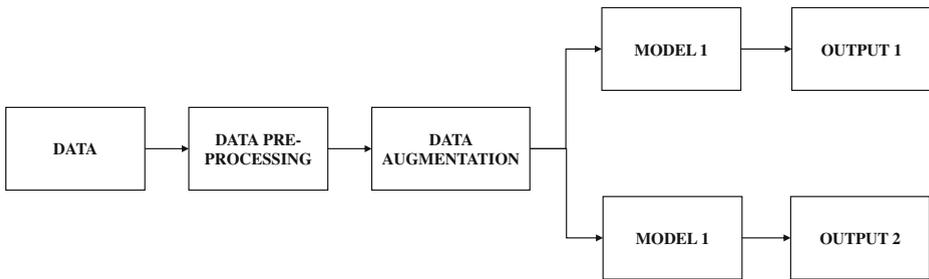


Fig. 7 Sample images of Outex\_TC\_00012 data set after pre-processing and data augmentation



**Fig. 8** Proposed model

a related problem. In the medical domain, the number of samples is limited because the procedure of collecting the data is both expensive and complicated. In such situations, using a pre-trained model is more effective than training a model from scratch. One such example is breast cancer classification [34] where the goal is to classify whether a cancer is malignant or benign. The paper compared the results from a pre-trained model and a model trained from scratch. Results obtained by transfer learning surpassed those obtained by a model trained from scratch. In this paper, we have used TensorFlow Hub to import such pre-trained models without their top layers. A softmax layer is then added to these layers. For the Kylberg dataset, only the last layer is trained. For the Outex and brodatz datasets, the models were fine-tuned.

**MobileNetV3** MobileNet was proposed by Sandler, Howard [33]. This model has achieved a great balance between performance and computation cost. MobileNet offers an extremely efficient network architecture that can easily match the requirements for mobile and embedded applications. This paper makes use of the MobileNetV3 small model, which was proposed in [12]. TensorFlow Hub is used to use the MobileNetV3 model, which has been trained on ImageNet (ILSVRC-2012-CLS) data. The model is used as feature extraction for the Kylberg dataset without tuning. The model is fully fine-tuned for Outex and the brodatz datasets. Figure 9 summarises the MobileNetV3 architecture.

**InceptionV3** InceptionV3 [40] is the third edition of Google’s Inception Convolutional Neural Network. The Inception modules are well-designed convolution modules that can generate discriminatory features and reduce the number of parameters. The InceptionV1 model was introduced at the 2014 ILSVRC classification challenge, where VGGNet [36] was also presented for the first time. Both gained similar results. However, Inception architecture had the advantage of performing well even under strict constraints on memory and computational budget.

The InceptionV1 [39] model overcame the problem of variation of information by having different sizes of filters and a wider network. It is 22 layers deep (27, including the pooling layers). It uses global average pooling at the end of the last inception module. It is a deep network and is subject to the vanishing gradient problem. To prevent the middle part of the network from “dying out,” it uses two auxiliary classifiers.

Neural networks perform better when convolutions don’t alter the dimensions of the input drastically. Reducing the dimensions too much may cause loss of information, known as a

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 224, 224, 3)]	0	[]
conv2d_94 (Conv2D)	(None, 149, 149, 32)	864	['input_2[0][0]']
batch_normalization_94 (BatchNormalization)	(None, 149, 149, 32)	96	['conv2d_94[0][0]']
activation_94 (Activation)	(None, 149, 149, 32)	0	['batch_normalization_94[0][0]']
conv2d_95 (Conv2D)	(None, 147, 147, 32)	9216	['activation_94[0][0]']
...	...	...	...
...	...	...	...
concatenate_3 (Concatenate)	(None, 8, 8, 768)	0	['activation_185[0][0]', 'activation_186[0][0]']
activation_187 (Activation)	(None, 8, 8, 192)	0	['batch_normalization_187[0][0]']
mixed10 (Concatenate)	(None, 8, 8, 2048)	0	['activation_179[0][0]', 'mixed9_1[0][0]', 'concatenate_3[0][0]', 'activation_187[0][0]']
avg_pool (GlobalAveragePooling2D)	(None, 2048)	0	['mixed10[0][0]']
predictions (Dense)	(None, 1000)	2049000	['avg_pool[0][0]']

Total params: 23,877,881

Trainable params: 24,048

Non-trainable params: 23,853,833

**Fig. 9** Summary of the MobileNetV3 model

“representational bottleneck.” InceptionV2 [40] model overcame this problem by expanding the filterbanks. InceptionV2 also used clever factorization methods to make the convolution more efficient in terms of computation complexity.

The InceptionV3 had all the upgrades that InceptionV2 had. In addition, it used RMSProp Optimizer, BatchNorm in the Auxillary Classifiers, and Label Smoothing to prevent overfitting. Figure 10 summarises the MobileNetV3 architecture.

## 4 Experiments and results

### 4.1 Hardware and software setup

Tesla K80 GPU and 13 GB RAM used for training along with TensorFlow, Keras, and Scikit-learn libraries in Google Colab, coded in Python 3.7.10.

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	[(None, 224, 224, 3)]	0	[]
conv2d_94 (Conv2D)	(None, 149, 149, 32)	864	['input_2[0][0]']
batch_normalization_94 (BatchNormalization)	(None, 149, 149, 32)	96	['conv2d_94[0][0]']
activation_94 (Activation)	(None, 149, 149, 32)	0	['batch_normalization_94[0][0]']
conv2d_95 (Conv2D)	(None, 147, 147, 32)	9216	['activation_94[0][0]']
...	...	...	...
...	...	...	...
concatenate_3 (Concatenate)	(None, 8, 8, 768)	0	['activation_185[0][0]', 'activation_186[0][0]']
activation_187 (Activation)	(None, 8, 8, 192)	0	['batch_normalization_187[0][0]']
mixed10 (Concatenate)	(None, 8, 8, 2048)	0	['activation_179[0][0]', 'mixed9_1[0][0]', 'concatenate_3[0][0]', 'activation_187[0][0]']
avg_pool (GlobalAveragePooling2D)	(None, 2048)	0	['mixed10[0][0]']
predictions (Dense)	(None, 1000)	2049000	['avg_pool[0][0]']

Total params: 23,877,881

Trainable params: 24,048

Non-trainable params: 23,853,833

**Fig. 10** Summary of the InceptionV3 model

## 4.2 Training and testing data

The Kylberg, Brodatz and the Outex datasets are split into training data (80%) and testing data (20%). Adam optimisation and categorical cross-entropy loss functions are used in all cases. A learning rate of 0.01 has been used. The batch size for the training was set to 32. The proposed model 1 for the Kylberg dataset is only fully trained on the training data. Rest in all other cases, the pre-trained model is used as a feature vector, and only the top added layer is trained on the training data.

## 4.3 Evaluation criteria

In the prediction phase, seven quantitative performance measures were computed to assess the reliability of trained models using the validation data, including precision, recall, f1-score, accuracy, macro-avg, weighted-avg and Cohen kappa score. These metrics are computed based on True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN).

$$\text{Precision} = \frac{TP}{TP + FP} \quad (1)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

$$\text{F1Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + TN + FP} \quad (4)$$

$$\text{Weightedavg} = F1_{class1} * W1 + F1_{class2} * W2 + F1_{class3} * W3 + \dots + F1_{classn} * Wn \quad (5)$$

F1classm : F1 score of class m

$$\text{Macroavg} = F1_{class1} + F1_{class2} + F1_{class3} + \dots + F1_{classn} \quad (6)$$

F1classm : F1 score of class m Cohen kappa score:

$$K = \frac{p0 - pe}{1 - pe} \quad (7)$$

p0 = relative observed agreement among raters, pe = the hypothetical probability of chance agreement.

## 4.4 Training single convolution mode

All the images in the .gif or the .ras format were converted to a compatible format. After that, All the images of the three datasets in the study were rescaled to a size of 224\*224. The images were then normalised to make the values of their pixels range from 0-1. The Kylberg and Brodatz datasets were then subjected to data augmentation before passing them to the proposed model.

### 4.4.1 Kylberg dataset

The first dataset to be studied was the Kylberg dataset. The first model is developed using the MobileNetV3 small model, trained on the ImageNet dataset. The top layer of the pre-trained model is removed and replaced by a softmax layer with 28 classes. The model was

**Table 4** Classification report for model 1 Kylberg dataset

	precision	recall	f1-score	support
Accuracy	–	–	1.00	896
Macro Avg	1.00	1.00	1.00	896
Weighted Avg	1.00	1.00	1.00	896

fully fine-tuned, i.e. all the model layers were trained on the training dataset. The proposed model was trained for 10 epochs on the training dataset. The model achieved an accuracy of 100% on the testing dataset. The classification report and confusion matrix of model 1 on testing it on testing data are shown in Table 4 and Fig. 11 respectively. The accuracy vs epochs graph and the loss vs epochs graph of model1 for the Kylberg dataset while training is shown in Fig. 12.

The second model is developed using the InceptionV3 model trained on the ImageNet dataset. The top layer of the pre-trained model is removed and replaced by a softmax layer with 28 classes. The pre-trained model was used as a feature extractor, i.e. all the layers of the pre-trained model were frozen, and only the top layer was trained on the training dataset. The proposed model was trained for 10 epochs on the training dataset. The model achieved an accuracy of 99.8883% on the testing dataset. The classification report and confusion matrix of model 2 on testing it on testing data is shown in Table 5 and Fig. 13 respectively. The accuracy vs epochs graph and the loss vs epochs graph of model1 for the Kylberg dataset while training is shown in Fig. 14.

#### 4.4.2 Brodatz dataset

The second dataset to be studied was the Brodatz dataset. The first model is developed using the MobileNetV3 small model, trained on the ImageNet dataset. The top layer of the pre-trained model is removed and replaced by a softmax layer with 112 classes. The pre-trained

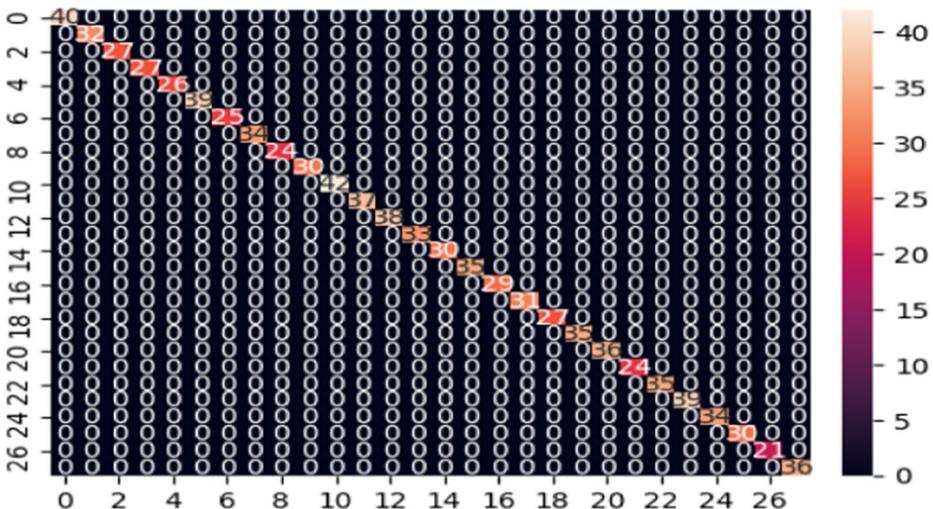


Fig. 11 Model 1 confusion matrix for the Kylberg dataset

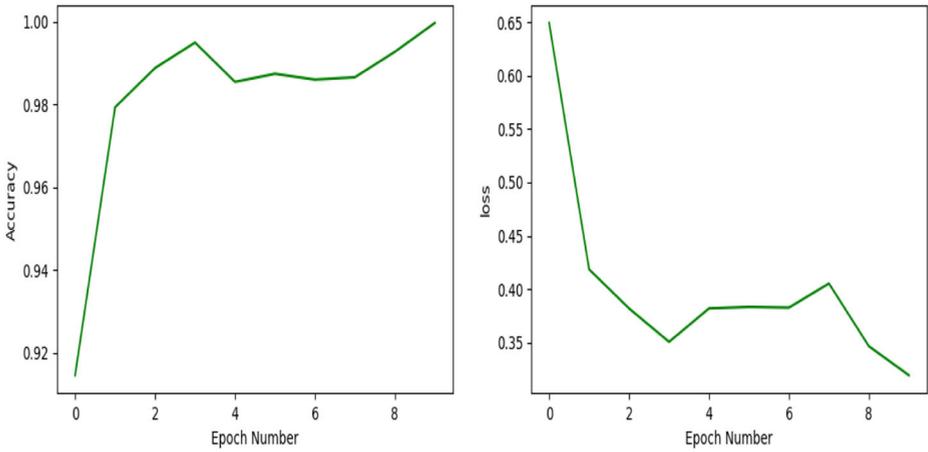


Fig. 12 Model 1 accuracy and losses graph for the Kylberg dataset

Table 5 Classification report for model 2 Kylberg dataset

	precision	recall	f1-score	support
Accuracy	—	—	1.00	896
Macro Avg	1.00	1.00	1.00	896
Weighted Avg	1.00	1.00	1.00	896

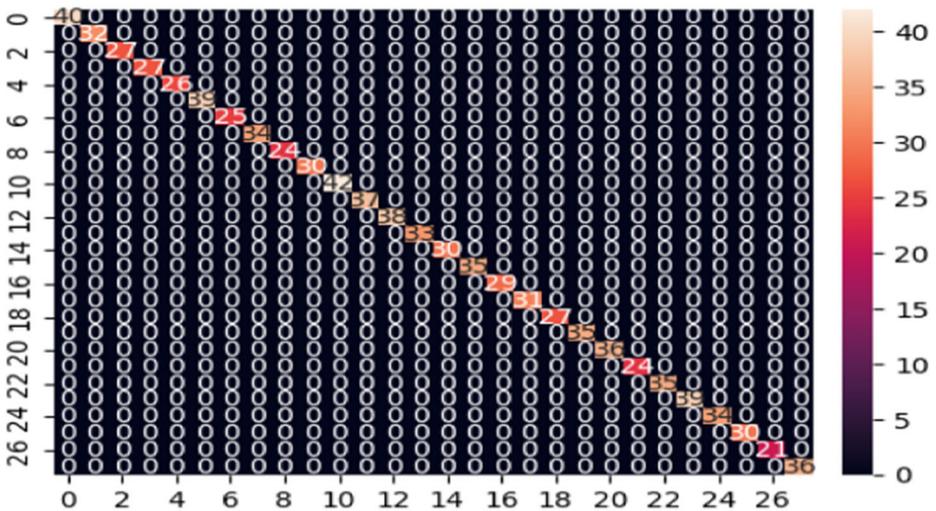
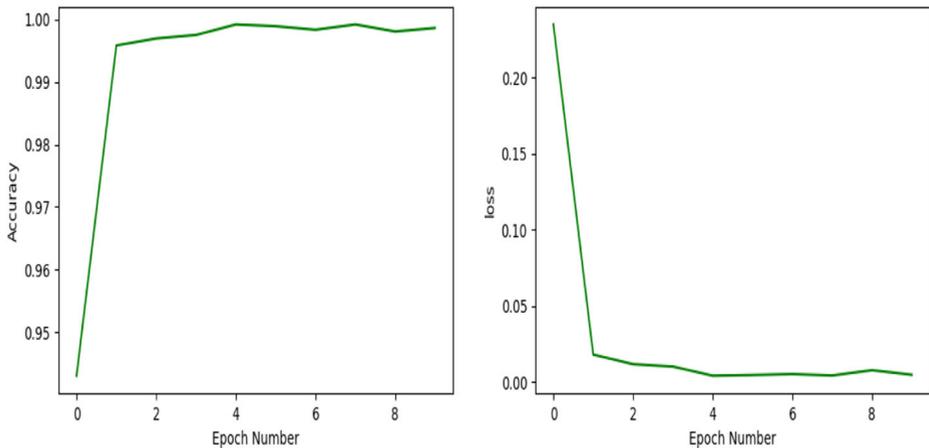


Fig. 13 Model 2 confusion matrix for the Kylberg dataset



**Fig. 14** Model 2 accuracy and losses graph for the Kylberg dataset

model was used as a feature extractor, i.e. all the layers of the pre-trained model were frozen, and only the top layer was trained on the training dataset. The proposed model was trained for 7 epochs on the training dataset. The model achieved an accuracy of 99.6651% on the testing dataset. The classification report of model 1 on testing it on the testing data is shown in Table 6. The accuracy vs epochs graph and the loss vs epochs graph of model1 for the Brodtz dataset while training is shown in Fig. 15.

The second model is developed using the InceptionV3 model trained on the ImageNet dataset. The top layer of the pre-trained model is removed and replaced by a softmax layer with 112 classes. The pre-trained model was used as a feature extractor, i.e. all the layers of the pre-trained model were frozen, and only the top layer was trained on the training dataset. The proposed model was trained for 7 epochs on the training dataset. The model achieved an accuracy of 99.8884% on the testing dataset. The classification report of model 2 on testing it on the testing data is shown in Table 7. The accuracy vs epochs graph and the loss vs epochs graph of model1 for the Brodatz dataset while training is shown in Fig. 16.

#### 4.4.3 Outex dataset

The third dataset to be studied was the Outex dataset. The first model is developed using the MobileNetV3 small model, trained on the ImageNet dataset. The top layer of the pre-trained model is removed and replaced by a softmax layer with 112 classes. The pre-trained model was used as a feature extractor, i.e. all the layers of the pre-trained model were frozen, and only the top layer was trained on the training dataset. The proposed model was trained for 5 epochs on the training dataset. The model achieved an accuracy of 99.479% on the testing

**Table 6** Classification report for model 1 Brodatz dataset

	precision	recall	f1-score	support
Accuracy	—	—	0.9967	896
Macro Avg	1.00	1.00	1.00	896
Weighted Avg	1.00	1.00	1.00	896

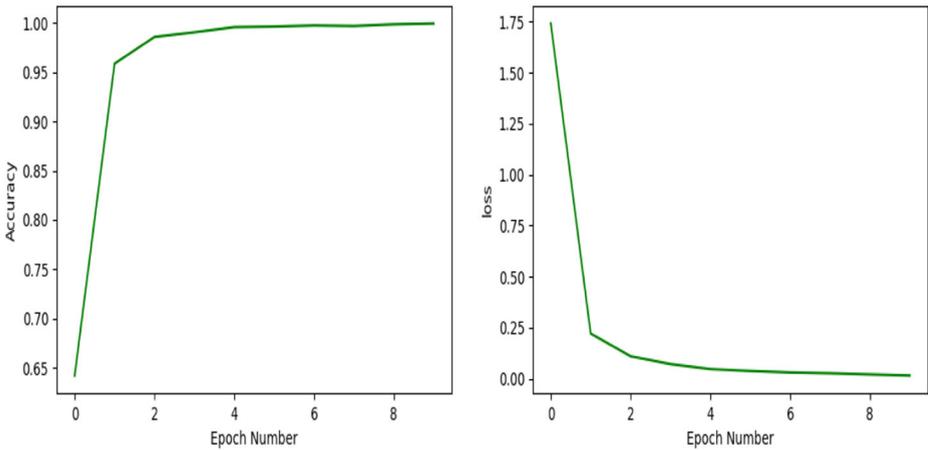


Fig. 15 Model 1 accuracy and losses graph for the Brodatz dataset

Table 7 Classification report for model 2 Brodatz dataset

	precision	recall	f1-score	support
Accuracy	—	—	0.9933	896
Macro Avg	0.99	0.99	0.99	896
Weighted Avg	0.99	0.99	0.99	896

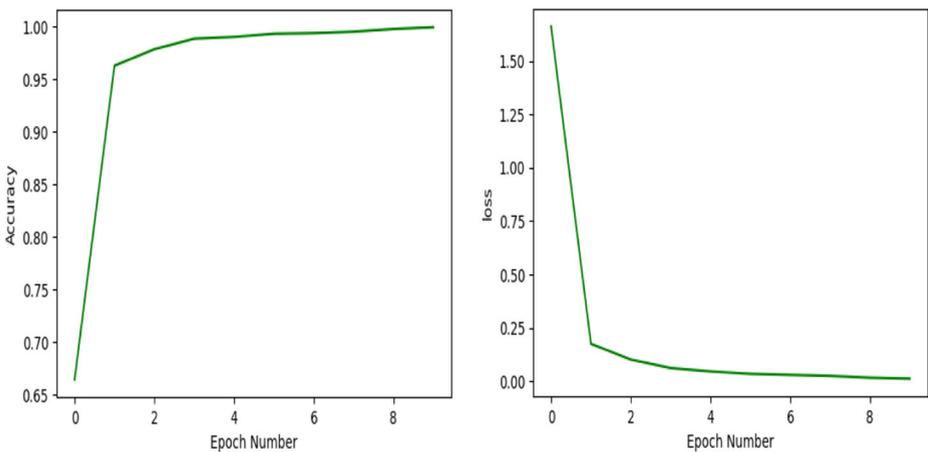


Fig. 16 Model 2 accuracy and losses graph for the Brodatz dataset

**Table 8** Classification report for model 1 Outex dataset

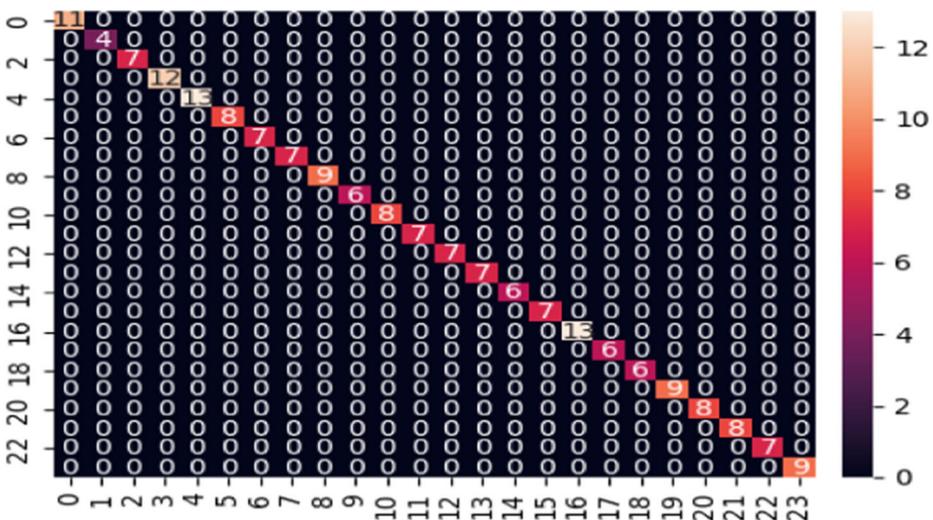
	precision	recall	f1-score	support
Accuracy	–	–	1.00	192
Macro Avg	1.00	1.00	1.00	192
Weighted Avg	1.00	1.00	1.00	192

dataset. The classification report and confusion matrix of model 1 on testing it on testing data is shown in Table 8 and Fig. 17 respectively. The accuracy vs epochs graph and the loss vs epochs graph of model1 for the Outex dataset while training is shown in Fig. 18.

The second model is developed using the InceptionV3 model trained on the ImageNet dataset. The top layer of the pre-trained model is removed and replaced by a softmax layer with 112 classes. The pre-trained model was used as a feature extractor, i.e. all the layers of the pre-trained model were frozen, and only the top layer was trained on the training dataset. The proposed model was trained for 5 epochs on the training dataset. The model achieved an accuracy of 99.479% on the testing dataset. The classification report and confusion matrix of model 2 on testing it on testing data are shown in Table 9 and Fig. 19 respectively. The accuracy vs epochs graph and the loss vs epochs graph of model1 for the Outex dataset while training is shown in Fig. 20.

### 4.5 Comparative study

The results of the 2 proposed models are compared with other recently proposed models. Table 10 shows the comparison between the two proposed models and other recently applied models on the Kylberg dataset. Table 11 shows the comparison between the two proposed



**Fig. 17** Model 1 confusion matrix for the Outex dataset

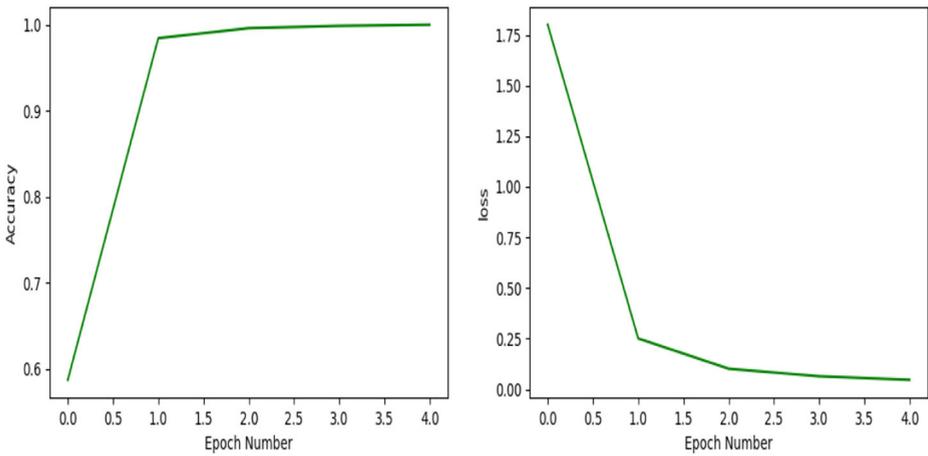


Fig. 18 Model 1 accuracy and losses graph for the Outex dataset

Table 9 Classification report for model 2 Outex dataset

	precision	recall	f1-score	support
Accuracy	—	—	1.00	192
Macro Avg	1.00	1.00	1.00	192
Weighted Avg	1.00	1.00	1.00	192

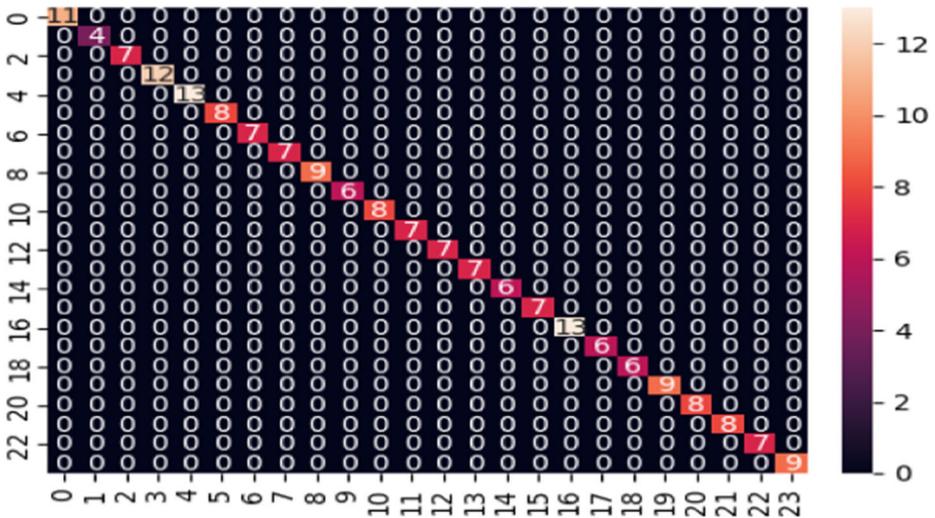
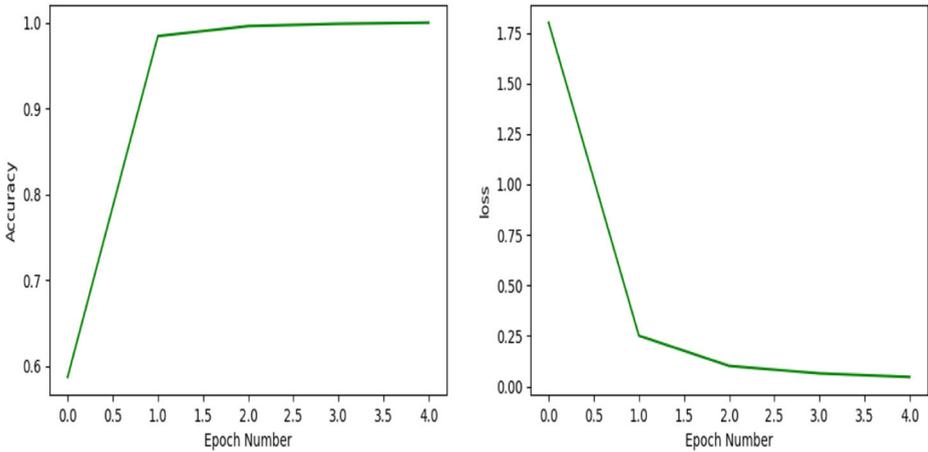


Fig. 19 Model 2 confusion matrix for the Outex dataset



**Fig. 20** Model 2 accuracy and losses graph for the Outex dataset

**Table 10** Performance comparison of our models with the existing techniques for the Kylberg dataset

Paper (reference)	Model/technique	Classification accuracy (%)
Andrearczyk and Whelan [2]	T-CNN-3	99.4±0.2
Kaya et al. [15]	KNN+nLBP(d=1)	99.64
El Khadiri et al. [8]	RALBGC	99.23
Kaya et al. [15]	LBP	97.97
Dixit et al. [6]	Modified CNN+WOA	99.71
Proposed model 1	MobileNetV3 (Fully fined tuned)	100
Proposed model 2	InceptionV3 (Feature extraction)	100

**Table 11** Performance comparison of our models with the existing techniques for the Brodatz dataset

Paper (reference)	Model/technique	Classification accuracy (%)
Kaya et al. [15]	LBP $\hat{2}$ and nLBP.d	99.26
El Khadiri et al. [8]	RALBGC, RLBGC	100
de Mesquita Sá Junior and Backes [25]	ELM based Signature ( $\Psi_{19,39}$ )	99.42
Ahmadvand and Daliri [1]	Hybrid feature vector	89.28
Dixit et al. [6]	Modified CNN+WOA	97.43
Proposed model 1	MobileNetV3 (Feature extraction)	99.67
Proposed model 2	InceptionV3 (Feature extraction)	99.33

**Table 12** Performance comparison of our models with the existing techniques for the Outex TC-00012 dataset

Paper (reference)	Model/technique	Classification accuracy (%)
Ahmadvand and Daliri [1]	Hybrid feature vector	90.78
Mehta and Egiazarian [24]	FbLBP	96.00
Dixit et al. [6]	Modified CNN+WOA	97.70
Proposed model 1	MobileNetV3 (Feature extraction)	100
Proposed model 2	InceptionV3 (Feature extraction)	100

models and other recently applied models on the Brodatz dataset. Table 12 shows the comparison between the two proposed models and other recently applied models on the Outex dataset.

#### 4.6 Discussion

In this experiment, the pre-trained models used are trained on the ImageNet dataset and openly available for use. The models were trained and tested using two cases. In the first case, the pre-trained model used a feature extractor, and only the last layer was trained on the dataset. The whole model was trained on the training dataset in the second case. The feature extraction case yielded better results and lesser training time in most cases. As mentioned in Section 3.2, all the images were rescaled to a size of  $224*224*3$  to make them compatible with the pre-trained models. The datasets were then split in a ratio of 80:20 for training and testing data. Tables 10, 11, and 12 in Section 4.5 showcase the comparison of the results of our method with the previously proposed methods. From the tables, it is evident that our methods have outperformed the previously proposed methods.

### 5 Conclusion and future Scope

Texture classification is an essential area of research that has attracted many researchers to propose different models. From the comparative study, it can be concluded that our models give better results than most of the existing models for the Kylberg and Outex datasets. Both models got a testing accuracy of 100 on the Kylberg and Outex datasets. Our models gave competitive results for the Brodatz dataset too. Despite using the models as only feature extractors (except for MobileNetV3 on the Kylberg dataset), the models have attained outstanding results. It means that the datasets in the study and the ImageNet dataset have very similar feature space. Hence, it can be concluded that transfer learning can be used to quickly solve tasks where the feature space of the target dataset is similar to the feature space of the dataset on which the pre-trained model is trained.

In future, we would like to test our models on more texture datasets and even use them for other domains like medical and aerial imagery. It is evident that the similarity of feature space of the source and target dataset has a massive impact on the model performance. This study used models which were trained on the ImageNet datasets. The authors also aim to extend this work to transformer based architectures. We would also like to expand the study by using the same model architectures trained on a different dataset. Using different source models for standard architectures and different target models can help understand transfer learning deeper.

**Funding** No funding was received to assist with the preparation of this manuscript.

**Data Availability** The Brodatz [4] and the Kylberg [20] datasets are publicly available and can be accessed using the link mentioned in the citation. The Outex [28] dataset is available on request from the authors of the cited paper.

## Declarations

**Conflict of Interests** The authors declare that they have no conflict of interest.

## References

- Ahmadvand A, Daliri MR (2016) Invariant texture classification using a spatial filter bank in multi-resolution analysis. *Image Vis Comput* 45:1–10. <https://doi.org/10.1016/j.imavis.2015.10.002>
- Andrearczyk V, Whelan P (2016) Using filter banks in convolutional neural networks for texture classification. *Pattern Recogn Lett* 84:63–69. <https://doi.org/10.1016/j.patrec.2016.08.016>
- Arora V, Ng EYK, Leekha RS, Darshan M, Singh A (2021) Transfer learning-based approach for detecting covid-19 ailment in lung ct scan. *Comput Biol Med* 135:104575. <https://doi.org/10.1016/j.combiomed.2021.104575>
- Brodatz P (1966) Textures: A photographic album. Accessed June 2021. <http://sipi.usc.edu/database/database.php?volume=textures>
- Di Ruberto C (2017) Histogram of radon transform and texton matrix for texture analysis and classification. *IET Image Process* 11(9):760–766. <https://doi.org/10.1049/iet-ipr.2016.1077>
- Dixit U, Mishra A, Shukla A, Tiwari R (2019) Texture classification using convolutional neural network optimized with whale optimization algorithm. *SN Appl Sci* 1(6):655. <https://doi.org/10.1007/s42452-019-0678-y>
- Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al. (2020) An image is worth 16x16 words: transformers for image recognition at scale. *arXiv:2010.11929*
- El Khadiri I, Kas M, El Merabet Y, Ruichek Y, Touahni R (2018) Repulsive-and-attractive local binary gradient contours: new and efficient feature descriptors for texture classification. *Inf Sci* 467:634–653. <https://doi.org/10.1016/j.ins.2018.02.009>
- Feng J, Liu X, Dong Y, Liang L, Pu J (2017) Structural difference histogram representation for texture image classification. *IET Image Process* 11:118–125. <https://doi.org/10.1049/iet-ipr.2016.0495>
- Haralick RM, Shanmugam K, Dinstein I (1973) Textural features for image classification. *IEEE Transactions on Systems Man, and Cybernetics SMC-3*(6),610–621. <https://doi.org/10.1109/TSMC.1973.4309314>
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR), pp 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, Wang W, Zhu Y, Pang R, Vasudevan V et al (2019) Searching for mobilenetv3. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 1314–1324
- Jain A, Rao ACS, Jain PK, Abraham A (2022) Multi-type skin diseases classification using op-dnn based feature extraction approach. *Multimedia Tools and Applications* 81(5):6451–6476. <https://doi.org/10.1007/s11042-021-11823-x>
- Kalita DJ, Singh VP, Kumar V (2021) A dynamic framework for tuning svm hyper parameters based on moth-flame optimization and knowledge-based-search. *Expert Syst Appl* 168:114139. <https://doi.org/10.1016/j.eswa.2020.114139>
- Kaya Y, Ertuğrul OF, Tekin R (2015) Two novel local binary pattern descriptors for texture analysis. *Appl Soft Comput* 34(C):728–735. <https://doi.org/10.1016/j.asoc.2015.06.009>
- Kazi A, Panda SP (2022) Determining the freshness of fruits in the food industry by image classification using transfer learning. *Multimedia Tools and Applications* 81(6):7611–7624. <https://doi.org/10.1007/s11042-022-12150-5>
- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International conference on neural information processing systems-volume 1, NIPS'12. Curran Associates Inc., Red Hook, NY, USA, pp 1097–1105

18. Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. *Commun ACM* 60(6):84–90. <https://doi.org/10.1145/3065386>
19. Kundu R, Singh PK, Ferrara M, Ahmadian A, Sarkar R (2022) Et-net: an ensemble of transfer learning models for prediction of covid-19 infection through chest ct-scan images. *Multimedia Tools and Applications* 81(1):31–50. <https://doi.org/10.1007/s11042-021-11319-8>
20. Kylberg G (2011) The kylberg texture dataset v. 1.0. External report (Blue series) 35, Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden. Accessed June 2021. <http://www.cb.uu.se/gustaf/texture/>
21. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. *Neural Comput* 1(4):541–551. <https://doi.org/10.1162/neco.1989.1.4.541>
22. Liu XJ, Li KL, Luan HY, Wang WH, Chen ZY (2022) Few-shot learning for skin lesion image classification. *Multimedia Tools and Applications* 81(4):4979–4990. <https://doi.org/10.1007/s11042-021-11472-0>
23. Lu SY, Wang SH, Zhang YD (2020) A classification method for brain mri via mobilenet and feedforward network with random weights. *Pattern Recogn Lett* 140:252–260. <https://doi.org/10.1016/j.patrec.2020.10.017>
24. Mehta R, Egiazarian K (2016) Dominant rotated local binary patterns (drlbp) for texture classification. *Pattern Recogn Lett* 71(C):16–22. <https://doi.org/10.1016/j.patrec.2015.11.019>
25. de Mesquita Sá Junior JJ, Backes AR (2016) Elm based signature for texture classification. *Pattern Recogn* 51:395–401. <https://doi.org/10.1016/j.patcog.2015.09.014>
26. Nadeem Z, Khan Z, Mir U, Mir UI, Khan S, Nadeem H, Sultan J (2022) Pakistani traffic-sign recognition using transfer learning. *Multimed Tools Appl* 81(6):8429–8449. <https://doi.org/10.1007/s11042-022-12177-8>
27. Nasirzadeh M, Khazael AA, Khalid MB (2010) Woods recognition system based on local binary pattern. In: *Proceedings of the 2010 2nd international conference on computational intelligence, communication systems and networks, CICSYN '10*. IEEE Computer Society, USA, pp 308–313. <https://doi.org/10.1109/CICSyN.2010.27>
28. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 24(7):971–987. <https://doi.org/10.1109/TPAMI.2002.1017623>
29. Picard R, Kabir T, Liu F (1993) Real-time recognition with the entire brodatz texture database. In: *Proceedings of IEEE conference on computer vision and pattern recognition*, pp 638–639. <https://doi.org/10.1109/CVPR.1993.341050>
30. Pritt M, Chern G (2017) Satellite image classification with deep learning. In: *2017 IEEE applied imagery pattern recognition workshop (AIPR)*, pp 1–7. <https://doi.org/10.1109/AIPR.2017.8457969>
31. Ramola A, Shakya AK, Van Pham D (2020) Study of statistical methods for texture analysis and their modern evolutions. *Eng Reports* 2(4):e12149. <https://doi.org/10.1002/eng2.12149>
32. Sana JK, Islam MM (2018) Plt-based spectral features for texture image retrieval. *IET Image Process* 12(11):2065–2074. <https://doi.org/10.1049/iet-ipr.2018.5604>
33. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC (2018) Mobilenetv2: inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4510–4520
34. Shalla MR (2018) Breast cancer histology images classification: Training from scratch or transfer learning? *ICT Express* 4(4):247–254. <https://doi.org/10.1016/j.ict.2018.10.007>
35. Simon P, Vijayasundaram U (2020) Deep learning based feature extraction for texture classification. *Procedia Comput Sci* 171:1680–1687. <https://doi.org/10.1016/j.procs.2020.04.180>
36. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*
37. Singh VP, Srivastava R (2018) Improved image retrieval using fast colour-texture features with varying weighted similarity measure and random forests. *Multimedia Tools and Applications* 77(11):14435–14460. <https://doi.org/10.1007/s11042-017-5036-8>
38. Smith J, Chang SF (1994) Transform features for texture classification and discrimination in large image databases. In: *Proceedings of 1st international conference on image processing*, vol 3, pp 407–411. <https://doi.org/10.1109/ICIP.1994.413817>
39. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1–9
40. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2818–2826

41. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. *Advances in neural information processing systems* 30
42. Xu P, Yao H, Ji R, Sun X, Liu X (2011) A robust texture descriptor using multifractal analysis with gabor filter. In: *Proceedings of the second international conference on internet multimedia computing and service, ICIMCS '10*, pp 147–150. Association for Computing Machinery, New York, NY, USA. <https://doi.org/10.1145/1937728.1937763>
43. Yuan X, Yang Z, Zouridakis G, Mullani N (2006) Svm-based texture classification and application to early melanoma detection. In: *2006 international conference of the IEEE engineering in medicine and biology society*, pp. 4775–4778. <https://doi.org/10.1109/IEMBS.2006.260056>
44. Zheng Y, Zhong G, Liu J, Cai X, Dong J (2014) Visual texture perception with feature learning models and deep architectures. [https://doi.org/10.1007/978-3-662-45646-0\\_41](https://doi.org/10.1007/978-3-662-45646-0_41)

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.