Mehran University Research Journal of Engineering and Technology

https://doi.org/10.22581/muet1982.2998

2024, 43(2) 79-90

Modification of a convolutional neural network for the weave pattern classification

Noreen Akram [a], Rizwan Aslam Butt [a], Muhammad Amir [b, *]

[a] *Department of Telecommunications Engineering, NED University of Engineering and Technology, Karachi, Pakistan*

[b] *Department of Textile Engineering, NED University of Engineering and Technology, Karachi, Pakistan*

[*] Corresponding author: Muhammad Amir, Email: qureshi@neduet.edu.pk

Received: 25 August 2023, Accepted: 28 March 2024, Published: 01 April 2024

KEYWORDS

Pattern Recognition

Weave Fabric Type

VGG-16

CNN

Reinforcement Learning

ABSTRACT

The fabric quality in textile industry is characterized by the texture (weave pattern) as it plays a vital role for the production and design of best quality fabric. The earlier proposed automated weave identification methods based on image processing techniques are highly dependent on the lighting conditions. The machine learning methods have been reported to show better accuracy. However, they require very large training datasets, very high processing power and computation time. This study proposes improved accuracy with smaller dataset and reduced computation time by proposing a modification of VGG-16 model by adding two additional pooling layers. Using evaluation metrics of both models, the modified model results were analysed according to accuracy, balanced accuracy, and F1-score. On the basis of investigational outcomes, a comparison has been performed with earlier work. The results show that the proposed VGG-16 model is capable to achieve state-of-the-art accuracy and avoid unnecessary activation features by freezing the main convolutional base layers. Ultimately, as evidenced by the performance of the modified VGG-16 deep learning model, the proposed method demonstrated improved accuracy. The study results show that the proposed modified VGG-16 algorithm is able to recognize the features of provided database with 90% accuracy and F1-Score ranging from 0.8 to1.

1. Introduction

Fabric is a tremendous human invention that is an essential part of our daily life. This remarkable invention has progressed from hand-crafted textiles to modernized industrial machine-based textiles, holds great significance in the textile manufacturing industry [1]. There are various important aspects of woven fabrics production. Weave pattern plays a major role in designing/redesigning, its appearance analysis and structure of fabrics [2][3][4][5]. Presently, the identification of woven fabric patterns relies on manual procedures such as using human eyes behind a microscope of magnifying glass, generally, performed by an expert. Nevertheless, there are several disadvantages such as it is time-consuming, extensive labor is required and system is prone to human errors. Furthermore, manual inspection results may be impacted by the mental stress, dizziness, tiredness, physical stress in human beings leading to wrong recognition results. Therefore, in order to meet the production requirements, market challenges, customer demands, automated systems which can inspect/ recognize the different weave patterns which can classify fabric weave patterns have been proposed in the literature. In recent years, recognition of woven fabric patterns has gained popularity and achievements [6][7][8]. Simple image processing-based weave identification methods are also not efficient as they are very sensitive to varying lighting conditions [9].

The machine learning classification methods has been extensively reported to be very useful in automating the fabric identification and inspection processes. These classification methods are generally divided into two basic types. First method is dependent on the statics of textured fabric and the other one is dependent on the model or database method. According to Li et al.'s [10] technique, images are divided into smaller images using an adaptive mesh model to acquire gray-scale features after photometric data processing using an adaptive wiener filter and histogram equalization to obtain data from various directions. The study was limited to examining solely basic woven fabrics, but texture-based statistical approach was proven to be reliable because it took the weaving structure and extracted the interlacing points. The database/model-based approach, in contrast, matches and identifies fabric weave patterns using a classification or identification algorithm. In order to classify previously recognized weave patterns that were recorded in a database, Kuo et al. [11] and Pan et al. [12][13][14] used back-propagation systems. Kuo and Kao [9] used the different models which could extract data of fabric in terms of matrix generation. The size of the used database, however, had an impact on the efficacy and accuracy of these methods. In contrast, Fan et al. [15] created a system for recognizing textures, segmenting yarn, and applying Gabor filters. gradient accumulation of grey levels, and the K-means clustering algorithm. Because the method did not consider the relevance of lighting effect when taking pictures, it proved to be unsuitable for images with indistinct texture. Schröder et al. [16] developed a novel method for predicting weave patterns, widths, yarn pathways, and their changes. However, they still used a manual selection of model parameters in their approach, Later, an approach put proposed by Trunz et al. [17] relied on recognising and focusing stitch types on knitted fabric by determining the basis of grid-like structure. They focused mainly on knit and purl stitch patterns in their experiments. These approaches, based on single pictures for fabric prototyping, are based on the ideas of reverse engineering. Following this, Yildiz introduced a technique for reducing dimensionality based on principal component analysis for the feature extraction and categorization of flawed fleece fabric. He came to the conclusion that the K-nearest neighbour classifier was more accurate than Naive Bayes. This method emphasized on only on binary pattern instead of complex texture patterns [18]. For the purpose of extracting fabric features, Li et al. [19] used the support vector machine (SVM) to categorize diverse fabric textures. In comparison to employing each feature extraction approach alone, they improved recognition performance by combining GLCM and LBP. However, the application of this method to nonwoven and knitted materials was not explored and it exclusively relied on manual feature engineering. A technique known as local feature similarity (LFS) was later developed by Guo et al. [20]. The detection of woven fabric pattern was then accomplished by Xiao et al. [21] in which warp and weft cross points were located using fuzzy c-means clustering (FCM). However, the experiments were constrained by the inability to correctly identify double-yarn weave patterns and the substantial rotational changes. Khan [22] then presented a model that was able to identify the fabric's weave and colour. After altering the yarn fabric's appearance and colour, it was sturdy as well. Later, Liu et al. [23] presented an optimized convolutional neural network (CNN) method for fabric defect recognition that incorporated a visualization methodology for complicated textures. Only a small number of photos were selected in a very restrictive environment for experiments. Ouyang et al. [24] introduced a pioneering approach to address the detection of fabric blemishes by incorporating a paired-potential activation layer in CNN. This method effectively improved the accuracy of identifying fabric flaws, even when dealing with imbalanced datasets and intricate characteristics, by leveraging statistical blemish data. Expanding on this work, AN et al. Recently, convolutional neural networks (CNNs) have played astonishing performance in various domains, including object recognition [25][26], tracking [27]. CNNs excel at recognizing patterns in images by automatically learning high-level descriptive features, eliminating the need for manual feature engineering employed in old-fashioned mechanisms. This model is composed of several key components, including convolutional, pooling, and fully connected layers as shown in Fig.1.

Shallow CNN architectures like AlexNet and VGG-16 [28] are constructed by stacking multiple blocks together. On the other hand, deeper CNN architectures, such as ResNet, exhibit greater complexity due to their utilization of intricate interconnectedness among layers. The following Fig. 2 shows the feature extraction and classification using the CNN Network. The study in [29] introduced an enhanced technique known as faster-CNN. The authors proposed a modified form conventional VGG-16 with a deep residual network for feature extraction, resulting in an increased feature pyramid module and a higher number of anchor frames. This modification further enhanced the precision of fault identification. Moreover, to address the challenges arising from the uneven distribution of defect samples and the limited variation in available color cloth defect samples. Another method proposed by K. Zhang et al. [24] introduced accurate identification of defect categories and their locations in color fabric image datasets, resulting in improved defect detection. In [29] and VGG-16, a prominent CNN architecture is AlexNet, which was created by Alex Krizhevsky in association with Ilya Sutskever and Geoffrey Hinton. Conversely, deeper convolutional neural network (CNN) architectures, such as ResNet, employ intricate interconnections among layers. For deep model utilization, a pre-trained CNN model based on ResNet-50, which is residual network architecture, has been employed. Recently, Reinforcement Learning (RL) [30] has also gained popularity as a classification technique. This approach tackles the computational challenges associated with utilizing large images and makes the classification process faster. To address this issue, they proposed an RL agent that dynamically determines the resolution of each image provided to the detector. Throughout the training process, the agent receives double rewards as incentives. When an image is largely made up of small items, it chooses images with higher resolution for the fine-level detector, and when an image is primarily made up of huge objects, it chooses images with lower resolution for the coarse-level detector.

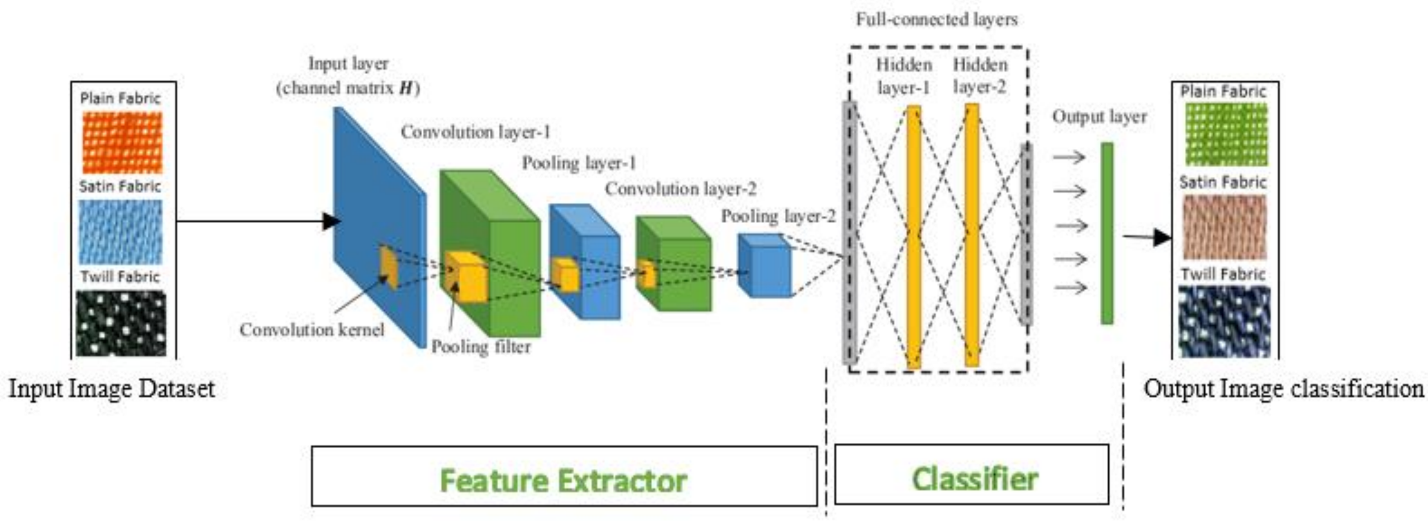


Fig. 1. CNN Architecture

The Fabric Dataset plays a significant role in image processing classification and recognition of the weave. From the extensive literature review it is evident that there is a limited fabric database availability. Moreover, the standard VGG-16 process is computationally extensive and not optimized for the fabric image classification Therefore, to address these shortcomings, the proposed work contributed to the creation of fabric samples and as well improving the VGG-16 process. To improve the accuracy of the VGG-16, the study proposes using additional layers while freezing the existing layers for improving the accuracy with smaller dataset with reduced computation time. Three additional layers were added in VGG-16; Batch normalization layer, fully connected layer and Dropout layer. The Dropout layer was added to avoid overfitting. The fully connected layer was added to capture the high content features of the fabric images and the normalization layer was added to adjust and scale the features of dataset by performing mean and convergence. The proposed research work will help in improving the automatic weave identification and inspection process in the textile industry [31]. This will also reduce the human errors such elusion impact by eliminating the human involvement in this process. The rest of the manuscript is organized as follows. The section 2 presents the methodology of this research work. The section 3 explains the model evaluation. Whereas the section 4 presents the results with a detailed discussion.

2. Methodology

2.1 Image acquisition System

For the proposed research work, Digital Microscope is used which is zoomed of up to 1600× and an 8-megapixel camera can perform image acquisition of woven fabric texture images. In addition, with a white and illuminated background, a test bed was prepared with a transparent acrylic sheet in order to allow the camera to capture images considering brightness of light as shown Fig. 2.

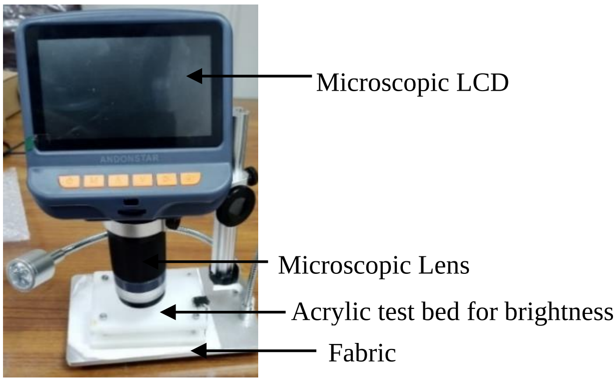
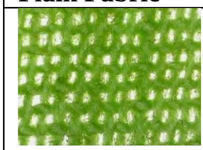
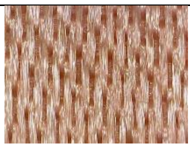
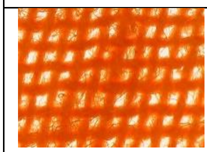
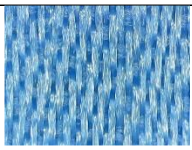
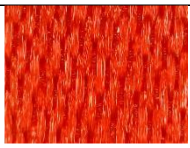
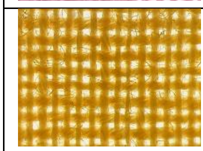
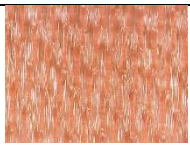


Fig. 2. Woven Image Acquisition System

Total 600 images for three different types of fabrics have been considered. (200 images for Twill, 200 images for Satin and 200 images for Plain fabric). Out of these 600 images, 10 percent images are kept as test images while 90 percent of the remaining images are kept as train images i.e. 160 x 3=480 images are used are trained images and 120 images out of 600 are used as test images. The image resolution for the research work is 640 x 480. Table 1 shows few sample images of three different categories (plain, satin, and twill weave fabrics).

Table 1

Selected samples of woven fabric images from the dataset.

Plain Fabric	Satin Fabric	Twill Fabric
		
		
		
		

2.2 Dataset Preparation

After getting the raw images, different techniques of data augmentation have been applied. The data augmentation performs various manipulations on whole dataset, for instance, flipping, scaling, lighting, and skewing. This resulted in the expansion of dataset. Moreover, deep learning models perform well on big datasets. With the help of augmentation, the model trains more effectively as it in order to increases the size of dataset. Furthermore, it reduces over fitting issue and increases generalization capability with expanding dataset (which is a major problem), it is kind of regularization which is performed on overall dataset without any changes.

During the research work, various augmentation techniques were performed on images, for instance, vertical and horizontal flips, shifting, zooming, brightness manipulation shearing and rotation. During image acquisition, variations occurred so therefore it is necessary to rotate images. The woven fabric interweaving pattern was identified through zooming. The data augmentation workflow has shown in Fig. 3.

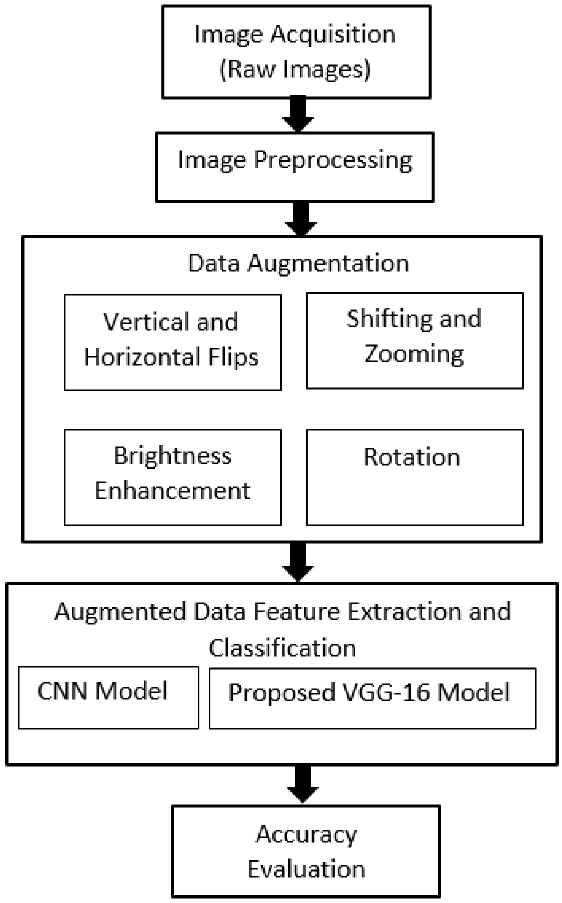


Fig. 3. Work Flowchart for Data Augmentation

3. Model Evaluation

This research work focus on reinforcement deep learning models for fabric image classification and identification which includes VGG-16 and CNN. The database images are tested with both algorithms (i.e. the existing CNN algorithm and modified VGG-16 network) and analysed their results in terms of accuracy.

3.1 CNN Model Analysis

Total 600 images for three different types of fabrics have been considered. (200 images for Twill, 200 images for Satin and 200 images for Plain fabric). Out of these 600 images, 10 percent images are kept as test images while 90 percent of the remaining images are kept as train images i.e. 160 x 3=480 images are used are trained images and 120 images out of 600 are used as test images. The methodology for the training and testing process for CNN network is given as in Fig. 4.

For CNN Models, following steps have been taken for the model development test and training.

Firstly, three categories are formed for the classification of three types of fabric as Twill, Satin and Plain. Afterwards, categories for two datasets are declared as Test and Train. There are two basic types of statistical methods i.e. the descriptive statistics and inferential statistics. When data has to be summarized using indexes such as mean, median, standard deviation, the descriptive statistics is used, while inferential statistics is preferred to draw statistical conclusions [32]. In the next step, mean of all images are calculated to get the average height and width of all images. Once the average is obtained, GPU of the device has been activated.

In order to verify the image lies in its own category, random images are picked for verification. This is shown in Fig. 5. In the next step, the CNN model summary is given with the Layer Type, Output Shape, and total number of trainable and non-trainable parameters. This model summary is shown in Fig. 6.

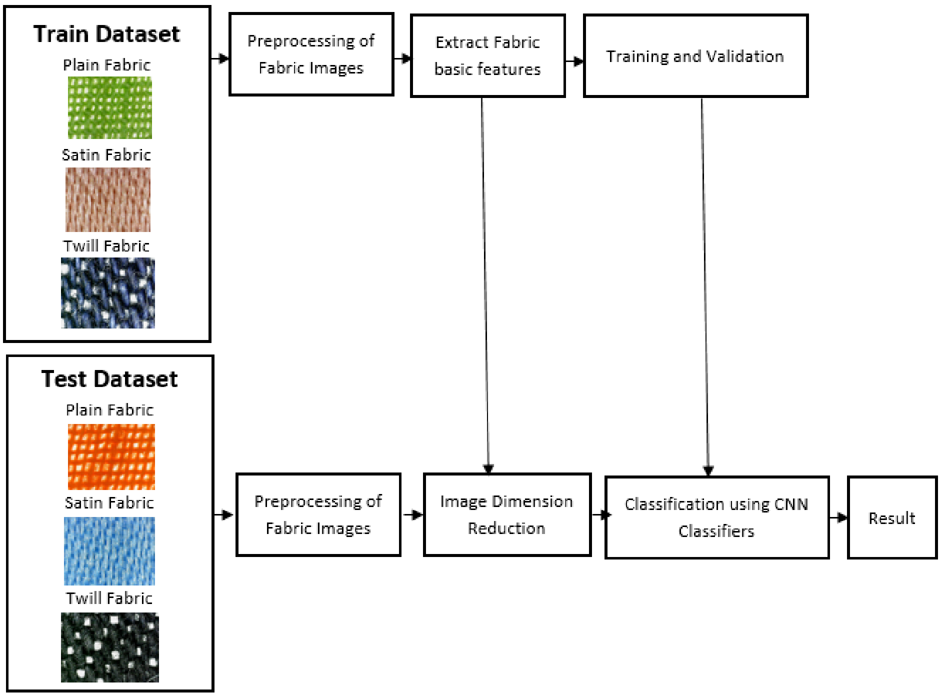


Fig. 4. The Methodology for The Training And Testing Process For CNN Network

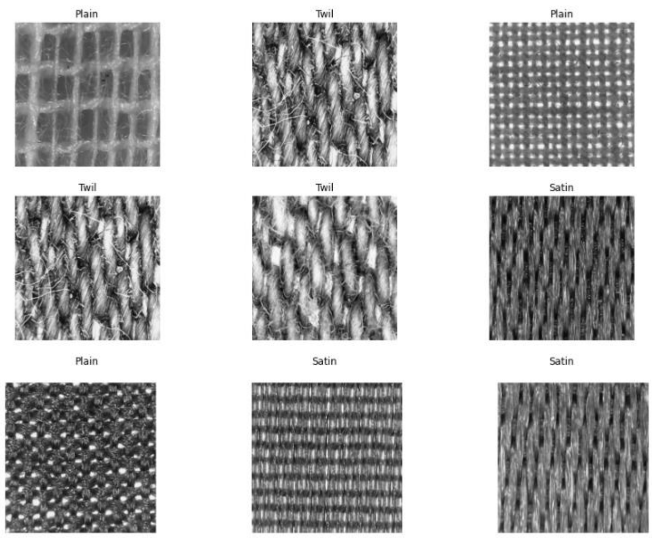


Fig. 5. Verification Of Images Belongs in Their Own Category

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 220, 220, 32)	832
max_pooling2d_3 (MaxPooling 2D)	(None, 218, 218, 32)	0
dropout_3 (Dropout)	(None, 218, 218, 32)	0
conv2d_4 (Conv2D)	(None, 218, 218, 64)	18496
max_pooling2d_4 (MaxPooling 2D)	(None, 218, 218, 64)	0
dropout_4 (Dropout)	(None, 218, 218, 64)	0
conv2d_5 (Conv2D)	(None, 218, 218, 128)	73856
max_pooling2d_5 (MaxPooling 2D)	(None, 72, 72, 128)	0
dropout_5 (Dropout)	(None, 72, 72, 128)	0
flatten_1 (Flatten)	(None, 663552)	0
dense_1 (Dense)	(None, 3)	1990659

Total params: 2,083,843
Trainable params: 2,083,843
Non-trainable params: 0

Fig. 6. CNN Model Summary Describing the Layer Type And Total Number Of Parameters

3.1.1 Max-pooling dropout during training time

In a traditional convolutional network, all the layers including fully connected layers, dropout layers and pooling layers, the dropout Eq. [33][32] is characterized as follows:

$$a_j^{(l+1)} = Pool\left(a_1^{(l)}, \dots, a_i^{(l)}, \dots, a_n^{(l)}\right), i \in R_j^{(l)} \quad (1)$$

Here $a_1^{(l)}, \dots, a_i^{(l)}, \dots, a_n^{(l)}$ = activations

In the above Eq. (1) at layer 1, $R_j^{(l)}$ is pooling region in which j and l indicates the activity of each neuron in it. The pooling function in this Eq. excludes the unnecessary features which results in reduction of computational cost and complications of the upper layers. The pooling function is mainly characterized into two major categories. One is average-pooling, and the other one is max-pooling. These are the most commonly used methods. The average-pooling accepts all the similar activities whereas the max-pooling only considers the strongest activations. Another thing in max-pooling is that the unnecessary units are excluded. This results in forward propagation Eq. (2).

$$\hat{a}^{(l)} \sim m^{(l)} * a^{(l)} \quad (2)$$

$$a_j^{(l+1)} = Pool\left(\hat{a}_1^{(l)}, \dots, \hat{a}_i^{(l)}, \dots, \hat{a}_n^{(l)}\right), i \in R_j^{(l)} \quad (3)$$

Here $*$ denotes element wise product and $m^{(1)}$ is a binary mask with each element $m_i^{(l)}$drawn independently from a Bernoulli distribution.

Here $\hat{a}_n^{(l)} = \text{droupout activations}$, which means that there is the presence of useful information of features. These features are further forwarded to the pooling layers.

By incorporating dropout into the max-pooling layers, the selection of the strongest activation as the output is no longer guaranteed. This introduces a stochastic (random) aspect to the max-pooling process during training.

$$\Pr\left(a_j^{(l+1)} = \acute{a}_i^{(l)}\right) = p_i = pq^{n-i}, (i = 1,2,, \dots . n) \quad (4)$$

The output of the pooling region shows a multinomial distribution in Eq. (4), in which i is the chosen index, representing the pooled activation. This Eq. gives the probability of the activated units (i.e. $a_1^{(l)}, \dots, a_i^{(l)}, \dots, a_n^{(l)}$)though pooling.

$$a_j^{(l+1)} = \acute{a}_i^{(l)}, where\ i \sim Multinomial\ (p_0, p_1, p_2, \dots ., p_n) \quad (5)$$

This Eq. (5) represents that multinomial method is similar to sampling on the basis of selection of index i.

For finding the size of a feature map at layer l, the number of possibly trained models C is:

$$C = (1+t)^{rs/t} = (\sqrt[t]{1+t})^{rs} \quad (6)$$

The max-pooling dropouts for the convolutional layers have been trained exponentially. This is done with the help of Eq. (6).

Here s = size of feature map at layer l , t = size of pooling regions, r = feature maps

Therefore, rs/t represents the number of pooling regions.

The variables used in Eq. 1, 2,3,4,5 and 6 shows the feature training phase. The average-pooling accepts all the similar activities whereas the max-pooling only considers the strongest activations. Another thing in max-pooling is that the unnecessary units are excluded. There is a total of 2,083,843 parameters which are trained, and it can be seen from the following figure. (This is shown in Fig. 6.)

3.1.2 Probabilistic weighted pooling during testing

When dropout is applied during training in fully connected layers, it is essential to utilize the complete network with all its hidden units. However, to mitigate the impact of having twice as many active units [1], or having their activations reduced by half during training due to dropout, the outgoing weights of the units must be halved.

$$a_j^{(l+1)} = p \times \max\left(a_1^{(l)}, \dots ., a_i^{(l)}, \dots, a_n^{(l)}\right) \quad (7)$$

The term "scaled max-pooling" is used to describe the process of reducing the intensity of the strongest activation within a pooling region based on the retaining probability.

While scaled max-pooling is frequently efficient in practice, it might not always be the best strategy during the testing phase.

$$a_j^{(l+1)} = \sum_{i=0}^{n} p_i\, \acute{a}_i^{(l)} = \sum_{i=1}^{n} p_i\, \acute{a}_i^{(l)} \quad (8)$$

Eq. (4)'s definition of the probability pi gives a precise representation of the mentioned probability [31]. Each index selection i corresponds to a separate model, making this probabilistic weighted summing an effective way to average various models. The claim that probabilistic weighted pooling, as opposed to scaled max-pooling, provides a more accurate approximation of averaging all potential dropout models is supported by empirical evidence.

Max-pooling dropout during training can be understood as a multinomial distribution-based random selection of activations from a pooling zone. With more input units in the pooling layers, there are exponentially more trained networks that could be created using max-pooling dropout. However, a brand-new pooling technique known as probabilistic weighted pooling is implemented during the testing phase. This pooling technique functions as a model averaging technique. With the above-mentioned layers, there are a total of 2, 083,843 numbers of parameters have been trained. In this step, the accuracy of the model is determined by considering 50 EPOCHS. After running the code, each epoch cycle gives information about the validation loss and validation accuracy of the training and testing phase. This information can be seen in the following Fig. 7. in the first 11 epochs.

```
Epoch 1/50
4/4 [==============================] - 5s 1s/step - loss: 65.5315 - accuracy: 0.3248 - val_loss: 1.0948 - val_accuracy: 0.4286
Epoch 2/50
4/4 [==============================] - 1s 305ms/step - loss: 2.4597 - accuracy: 0.5299 - val_loss: 1.0954 - val_accuracy: 0.5000
Epoch 3/50
4/4 [==============================] - 1s 306ms/step - loss: 1.2177 - accuracy: 0.3761 - val_loss: 1.0994 - val_accuracy: 0.2857
Epoch 4/50
4/4 [==============================] - 1s 314ms/step - loss: 1.0804 - accuracy: 0.3932 - val_loss: 1.0987 - val_accuracy: 0.2857
Epoch 5/50
4/4 [==============================] - 1s 308ms/step - loss: 1.0614 - accuracy: 0.4359 - val_loss: 1.0945 - val_accuracy: 0.2857
Epoch 6/50
4/4 [==============================] - 1s 309ms/step - loss: 1.0678 - accuracy: 0.5128 - val_loss: 1.0888 - val_accuracy: 0.2857
Epoch 7/50
4/4 [==============================] - 1s 303ms/step - loss: 1.0108 - accuracy: 0.4530 - val_loss: 0.9462 - val_accuracy: 0.5000
Epoch 8/50
4/4 [==============================] - 1s 301ms/step - loss: 0.9088 - accuracy: 0.6239 - val_loss: 0.9190 - val_accuracy: 0.6429
Epoch 9/50
4/4 [==============================] - 1s 309ms/step - loss: 1.1246 - accuracy: 0.4615 - val_loss: 1.0613 - val_accuracy: 0.6429
Epoch 10/50
4/4 [==============================] - 1s 302ms/step - loss: 0.9947 - accuracy: 0.4872 - val_loss: 0.8719 - val_accuracy: 0.5714
Epoch 11/50
4/4 [==============================] - 1s 302ms/step - loss: 0.9587 - accuracy: 0.5897 - val_loss: 1.0068 - val_accuracy: 0.6429
```

Fig. 7. CNN Model Accuracy

3.2 The VGG-16 Network

VGG-16 models demonstrate strong capabilities in nonlinear fitting, robustness, and generalization. Based on experimental results, the modified VGG approach proposed in this study proves effective in accurately identifying and classifying different types of fabric images. The VGG-16 architecture consists of 144 million parameters, employing a combination of small-sized convolutional kernels [30]. It features 16 convolutional layers with small-sized kernels (3 x 3), three fully connected layers, five max-pooling layers, and an output classifier layer utilizing the Softmax nonlinear activation function. In comparison to AlexNet, the VGG-16 architecture possesses a larger number of parameters, resulting in increased computational requirements and memory usage. Therefore, this is an expensive model. The architecture of VGG-16 has shown in Fig. 8.

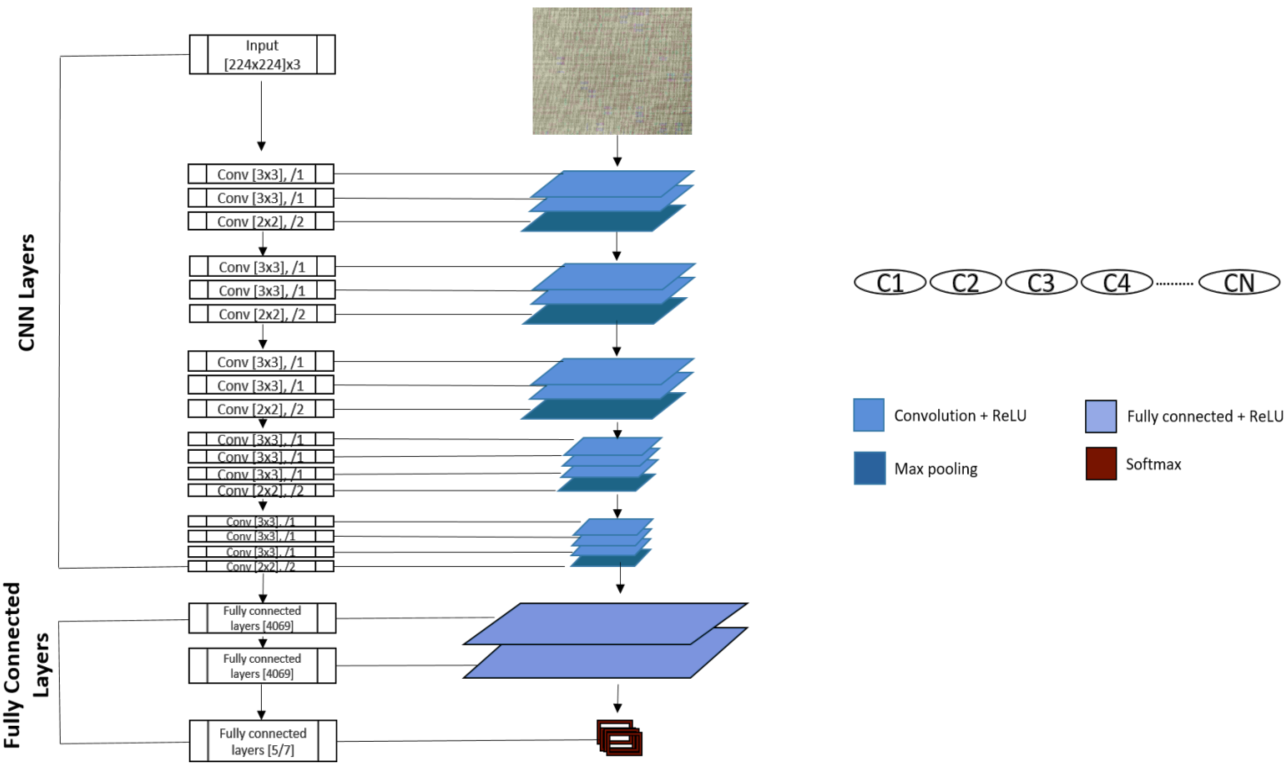


Fig. 8. VGG-16 Model

3.3 The Proposed Modified VGG-16 Model

Initially, experiments were conducted using a traditional convolutional neural network approach. However, this approach had certain limitations that negatively affected its accuracy: The process of establishing the dataset was time-consuming. The method struggled to handle fabrics with large curved or overlapped yarns effectively.

In the proposed modified VGG network, the training procedure focused on keeping the initial convolutional layers unchanged while solely training the additional custom layer connected to the base network. The primary objective of freezing these layers was to enhance the rate of convergence and avoid any issues with gradient explosion during the training process. Once the texture features were successfully extracted, the classification was carried out by comparing the predicted class with the actual class. This approach resulted in a reduction in both the computational cost of the network and the total number of trainable parameters within the modified VGG-16 model as this modified model is eliminated the unnecessary activations (i.e. features). This is achieved by freezing the convolutional layers. The research methodology for the proposed modified VGG-16 model is comprised of the following steps:

Step one: The fabric images are obtained and augmented using digital microscope to form fabric image database. Then the acquired images are converted to grey scale. Next, Wiener filter is used to remove the noises.

Step two: Noise-free images are then used to train modified VGG network. The network classifies identifies the different types of fabric for the images provided in the dataset.

Step three: The performance of modified network is checked, and different evaluation parameters are evaluated (i.e. accuracy, precision, F-1 score)

Finally, noise-free image dataset is split into 90% and 10% for Training and Validation respectively. All these mentioned steps are shown in Fig. 9 below.

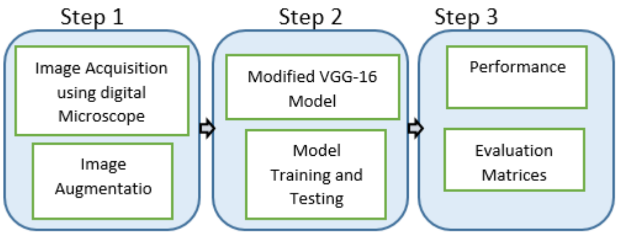


Fig. 9. The Modified VGG-16 Model Approach

Total 600 images for three different types of fabrics have been considered. (200 images each for Twill, Satin and Plain fabric). Out of these 600 images, 10 percent images are kept as test images while 90 percent of the remaining images are kept as train images i.e. 160 x 3=480 images are used are trained images and 120 images out of 600 are used as test images.

In the proposed modified VGG-16 algorithm, the convolutional base layers have kept frozen so that the convergence rate can be improved. Another important factor is initiated to modify the network by adding new layers. More improved and enhanced features are extracted by training the three types of database images with the added new layers (i.e. batch normalization, fully connected and dropout layers). Another advantage of adding new layers is to increase the process time of the algorithm as the unnecessary and similar features are not trained by freezing the convolutional layers of CNN. Once the dataset is trained with the new added layers, it is being classified with the help of Softmax in order to distinguish between the three types of fabrics. The methodology for extracting the features using the modified VGG-16 network is shown in Fig. 10.

The five convolutional layers are frozen during training to ensure they remain unaffected by the training process. The model is trained on a dataset comprising 600 images, with 120 images reserved for testing and 480 images used for training.

4. Results and Discussion

The results of both the CNN and modified VGG-16 methods were evaluated in this section. The modified VGG model was trained and tested using a 6 GB graphical processing unit (GPU) with 16 GB of RAM. The implementation of the model was done using Python 3.6, with the Keras library utilized as the frontend and TensorFlow as the backend.

4.1 CNN Algorithm Results

In this step, accuracy and the loss graph of the CNN model can be obtained. The average accuracy of this model with our own created database images is 61% which is not good enough to classify three different types of images for Plain, Twill and Satin. This accuracy is obtained with the 50 cycles of epochs. This Loss and Accuracy graph is presented in Fig. 11.

In the next step, the model shows the images which are identified among the databases of Plain, Twill and Satin images in Fig. 12.

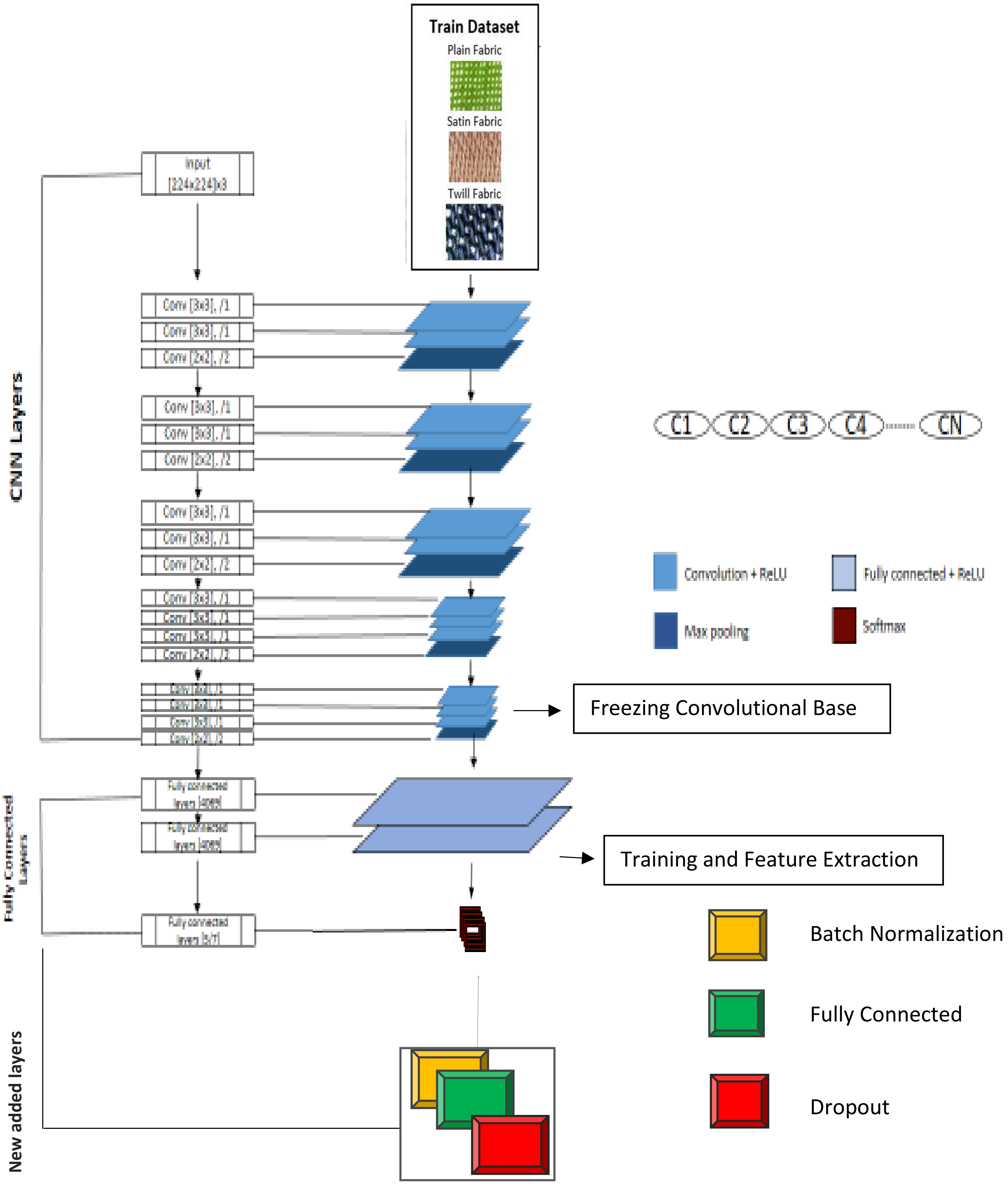


Fig. 10. Structure of Modified VGG-16 network.

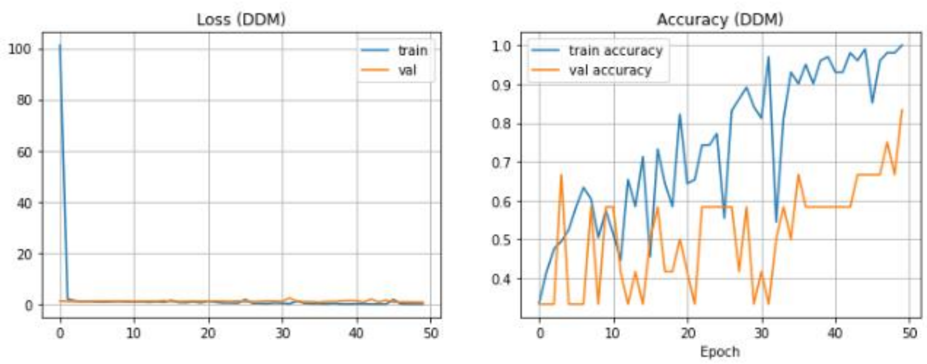


Fig. 11. Average Accuracy and Loss Graph of CNN Model

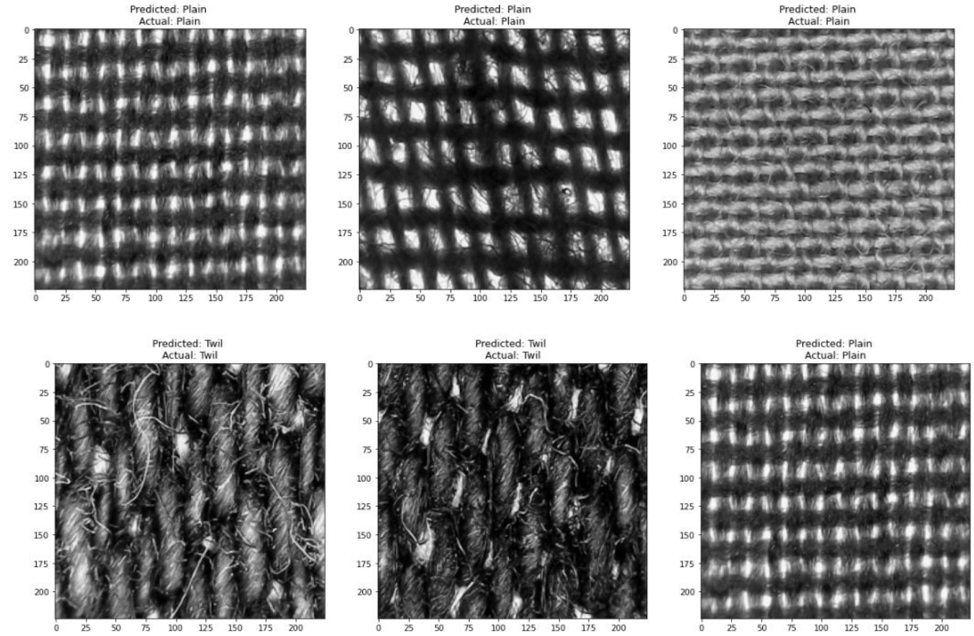


Fig. 12. Classification of Fabric types (Plain, Twill and Satin) using CNN Model

4.2 Modified VGG-16 Model Results

In the proposed model of VGG-16, the convolution base layers are frozen. The accuracy with 10 epochs of this proposed model is 90% which can be seen in Fig.13.

The following Fig. 14 shows the classification results for the modified model. The recognition and classification of three types of fabric is improved because of freezing the convolution layers and adding new layers. The similar units of weave pattern have eliminated.

```
Epoch 1/10
4/4 [==============================] - 19s 2s/step - loss: 16.1286 - accuracy: 0.2540 - val_loss: 16.6228 - val_accuracy: 0.5500
Epoch 2/10
4/4 [==============================] - 2s 638ms/step - loss: 5.4388 - accuracy: 0.7302 - val_loss: 8.4466 - val_accuracy: 0.8500
Epoch 3/10
4/4 [==============================] - 2s 636ms/step - loss: 0.5314 - accuracy: 0.9603 - val_loss: 5.9123 - val_accuracy: 0.9000
Epoch 4/10
4/4 [==============================] - 2s 652ms/step - loss: 0.1104 - accuracy: 0.9921 - val_loss: 4.4850 - val_accuracy: 0.9000
Epoch 5/10
4/4 [==============================] - 3s 551ms/step - loss: 0.0269 - accuracy: 0.9921 - val_loss: 4.0691 - val_accuracy: 0.9000
Epoch 6/10
4/4 [==============================] - 2s 636ms/step - loss: 6.2726e-07 - accuracy: 1.0000 - val_loss: 4.0760 - val_accuracy: 0.9000
Epoch 7/10
4/4 [==============================] - 2s 458ms/step - loss: 5.8940e-07 - accuracy: 1.0000 - val_loss: 4.1280 - val_accuracy: 0.9000
Epoch 8/10
4/4 [==============================] - 2s 367ms/step - loss: 3.5548e-06 - accuracy: 1.0000 - val_loss: 4.1591 - val_accuracy: 0.9000
Epoch 9/10
4/4 [==============================] - 3s 756ms/step - loss: 2.1406e-05 - accuracy: 1.0000 - val_loss: 4.1814 - val_accuracy: 0.9000
Epoch 10/10
4/4 [==============================] - 3s 924ms/step - loss: 4.4175e-05 - accuracy: 1.0000 - val_loss: 4.1930 - val_accuracy: 0.9000
Training Completed!
```

Fig. 13. Proposed VGG-16 Network Model Accuracy

4.3 Evaluation Metrics

The evaluation matrices for three basic types of fabric have been calculated by considering both algorithms i.e. proposed VGG-16 method and CNN algorithms. The accuracy of VGG-16 algorithm is found as 90% and all remaining parameters can be seen in Table 2.

Table 2

Fabric Classification and Accuracy Matrices of Proposed VGG-16

Fabric Type	Precision	Recall	F1-Score	Accuracy
Plain	0.71	1.00	0.83	0.90
Satin	1.00	0.78	0.88	0.90
Twill	1.00	1.00	1.00	0.90

While for the CNN Model, the maximum accuracy is found as 61%. This is due to the absence of additional layers of VGG-16 which extracted fabric features in good manner. The CNN performance indexes are shown in Table 3.

Table 3

The performance indexes for CNN model

Model	Precision	Recall	F1-Score	Accuracy
Plain	0.6	0.7	0.7	0.61
Satin	0.55	0.53	0.66	0.50
Twill	0.74	0.6	1.00	0.61

This result has shown that the processing time of the algorithm has been improved by freezing the convolutional layers. Furthermore, the accuracy of classifying the types of fabrics has been achieved till 90% by adding new layers. The dropout ratio for the newly added dropout layer was kept as 0.5.

5. Conclusion

Clearly, the modified VGG-16 gives better results than CNN (reinforcement learning method for classification and identification of basic types of fabrics, such as plain, satin and twill). According to the review of existing body of literature, many authors have played a part in the case study of deep learning in fabric classification. The main reason for supporting convolutional neural network (CNN) is due to poor inspection of fabric through conventional manual visual methods, moreover, its low on efficiency and not likely to be applied in industrial and textile industries in the long run. The modified VGG-16 algorithm does an amazing job in enabling this process in terms of decreasing the computational cost by extracting the necessary features in training process after adding the new layers. In addition, the experimental results show that the modified VGG-16 algorithm is better in terms of accuracy, speed, initial learning rate and dropout. In the near future, within a short time period, wireless network nodes can be added to create large set of database images which can be helpful in classifying and identifying the fabric types.

6. Acknowledgment

Authors acknowledged research grant from NED UET. Present thanks to Star Denim (Pvt) limited and Yunus Textile Mills Limited for Fabric samples support.

7. References

[1] P. F. Li, J. Wang, H. H. Zhang, and J. F. Jing, “Automatic woven fabric classification based on support vector machine,” IET Conf. Publ., vol. 2012, no. 598 CP, pp. 581–584, 2012, doi: 10.1049/cp.2012.1046.

[2] Y. Guo, X. Ge, M. Yu, G. Yan, and Y. Liu, “Automatic recognition method for the repeat size of a weave pattern on a woven fabric image,” Text. Res. J., vol. 89, no. 14, pp. 2754–2775, 2019, doi: 10.1177/0040517518801197.

[3] Z. Xiao et al., “Automatic recognition of woven fabric pattern based on TILT,” Math. Probl. Eng., vol. 2018, 2018, doi: 10.1155/2018/9707104.

[4] B. Khan, F. Han, Z. Wang, and R. J. Masood, “Bio-inspired approach to invariant recognition and classification of fabric weave patterns and yarn color,” Assem. Autom., vol. 36, no. 2, pp. 152–158, 2016, doi: 10.1108/AA-11-2015-100.

[5] Z. Liu, C. Zhang, C. Li, S. Ding, Y. Dong, and Y. Huang, “Fabric defect recognition using optimized neural networks,” J. Eng. Fiber. Fabr., vol. 14, no. 41, 2019, doi: 10.1177/1558925019897396.

[6] W. Ouyang, B. Xu, J. Hou, and X. Yuan, “Fabric defect detection using activation layer embedded convolutional neural network,” IEEE Access, vol. 7, no. c, pp. 70130–70140, 2019, doi: 10.1109/ACCESS.2019.2913620.

[7] M. An, S. Wang, L. Zheng, and X. Liu, “Fabric defect detection using deep learning: An improved faster R-approach,” Proc. - 2020 Int. Conf. Comput. Vision, Image Deep Learn. CVIDL 2020, no. Cvidl, pp. 319–324, 2020, doi: 10.1109/CVIDL51233.2020.00-78.

[8] J. P. Laaban, “La nutrition parenterale chez les insuffisants respiratoires,” Med. Chir. Dig., vol. 17, no. 8, pp. 551–552, 1988.

[9] G. C. Guarnera, P. Hall, A. Chesnais, and M. Glencross, “Woven fabric model creation from a single image,” ACM Trans. Graph., vol. 36, no. 5, 2017, doi: 10.1145/3132187.

[10] K. Zhang, Y. Yan, P. Li, J. Jing, X. Liu, and Z. Wang, “Fabric defect detection using salience metric for color dissimilarity and positional aggregation,” IEEE Access, vol. 6, pp. 49170–49181, 2018, doi: 10.1109/ACCESS.2018.2868059.

[11] N. S. Beach, “(12) United States Patent,” vol. 1, no. 12, 1992.

[12] B. A. Krizhevsky, I. Sutskever, and G. E. Hinton, Cnn实际训练的, vol. 60, no. 6. 2012.

[13] S. S. Mukrimaa et al., No 主観的健康感を中心とした在宅高齢者における 健康関連指標に関する共分散構造分析Title, vol. 6, no. August. 2016.

[14] C. Wirth, R. Akrour, G. Neumann, and J. Fürnkranz, “A survey of preference-based reinforcement learning methods,” J. Mach. Learn. Res., vol. 18, pp. 1–46, 2017.

[15] C. Y. Shih, C. F. J. Kuo, and J. H. Cheng, “A study of automated color, shape and texture analysis of Tatami embroidery fabrics,” Text. Res. J., vol. 86, no. 17, pp. 1791–1802, 2016, doi: 10.1177/0040517515609253.

[16] J. Jing, M. Xu, P. Li, Q. Li, and S. Liu, “Automatic classification of woven fabric structure based on texture feature and PNN,” Fibers Polym., vol. 15, no. 5, pp. 1092–1098, 2014, doi: 10.1007/s12221-014-1092-0.

[17] H. Halepoto, T. Gong, and K. Kaleem, “Real-time quality assessment of neppy mélange yarn manufacturing using macropixel analysis,” in Tekstilec, vol. 62, no. 4, 2019, pp. 242–247. doi: 10.14502/Tekstilec2019.62.242-247.

[18] Z. Xiao, X. Nie, F. Zhang, L. Geng, J. Wu, and Y. Li, “Automatic recognition for striped woven fabric pattern,” J. Text. Inst., vol. 106, no. 4, pp. 409–416, 2015, doi: 10.1080/00405000.2014.924655.

[19] K. S. Kumar and M. R. Bai, “Deploying multi layer extraction and complex pattern in fabric pattern identification,” Multimed. Tools Appl., vol. 79, no. 15–16, pp. 10427–10443, 2020, doi: 10.1007/s11042-019-7421-y.

[20] Guo, Yingchun, Xin Ge, Ming Yu, Gang Yan, and Yi Liu. "Automatic recognition method for the repeat size of a weave pattern on a woven fabric image." Textile research journal vol. 89, issue no. 14, p. 2754-2775, 2019, doi: https://doi.org/10.1177/0040517518801.

[21] J. Li, W. Wang, N. Deng, and B. Xin, “A novel digital method for weave pattern recognition based on photometric differential analysis,” Meas. J. Int. Meas. Confed., vol. 152, p. 107336, 2020, doi: 10.1016/j.measurement.2019.107336.

[22] C. F. J. Kuo, C. Y. Shih, C. en ho, and K. C. Peng, “Application of computer vision in the automatic identification and classification of woven fabric weave patterns,” Text. Res. J., vol. 80, no. 20, pp. 2144–2157, 2010, doi: 10.1177/0040517510373630.

[23] R. Pan, W. Gao, J. Liu, H. Wang, and X. Zhang, “Automatic detection of structure parameters of yarn-dyed fabric,” Text. Res. J., vol. 80, no. 17, pp. 1819–1832, 2010, doi: 10.1177/0040517510369411.

[24] R. Pan, W. Gao, J. Liu, and H. Wang, “Automatic recognition of woven fabric patterns based on pattern database,” Fibers Polym., vol. 11, no. 2, pp. 303–308, 2010, doi: 10.1007/s12221-010-0303-6.

[25] R. Pan, W. Gao, J. Liu, and H. Wang, “Automatic recognition of woven fabric pattern based on image processing and BP neural network,” J. Text. Inst., vol. 102, no. 1, pp. 19–30, 2011, doi: 10.1080/00405000903430255.

[26] K. Schröder, A. Zinke, and R. Klein, “Image-based reverse engineering and visual prototyping of woven cloth,” IEEE Trans. Vis. Comput. Graph., vol. 21, no. 2, pp. 188–200, 2015, doi: 10.1109/TVCG.2014.2339831.

[27] Z. Fan, S. Zhang, J. Mei, and M. Liu, “Recognition of woven fabric based on Image processing and gabor filters,” 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems, CYBER 2017, Hawaii, USA: IEEE, 2017, pp. 996–1000. doi: 10.1109/CYBER.2017.8446321.

[28] H. Zhang, N. Meng, Y. Liu, and X. Zhang, “Performance evaluation for local anchor-based dual connectivity in 5G user-centric network,” IEEE Access, vol. 4, pp. 5721–5729, 2016, doi: 10.1109/ACCESS.2016.2606420.

[29] K. Yildiz, “Dimensionality reduction-based feature extraction and classification on fleece fabric images,” Signal, Image Video Process., vol. 11, no. 2, pp. 317–323, 2017, doi: 10.1007/s11760-016-0939-9.

[30] B. Uzkent, C. Yeh, and S. Ermon, “Efficient object detection in large images using deep reinforcement learning,” Proc. - 2020 IEEE Winter Conf. Appl. Comput. Vision, WACV 2020, pp. 1813–1822, 2020, doi: 10.1109/WACV45572.2020.9093447.

[31] M. A. I. Hussain, B. Khan, Z. Wang, and S. Ding, “Woven fabric pattern recognition and classification based on deep convolutional neural networks,” Electron., vol. 9, no. 6, pp. 1–12, 2020, doi: 10.3390/electronics9061048.

[32] P. Mishra, C. Pandey, U. Singh, A. Keshri, and M. Sabaretnam, “Selection of appropriate statistical methods for data analysis,” Ann. Card. Anaesth., vol. 22, no. 3, pp. 297–301, 2019, doi: 10.4103/aca.ACA_248_18.

[33] H. Wu and X. Gu, “Max-pooling dropout for regularization of convolutional neural networks,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9489, pp. 46–54, 2015, doi: 10.1007/978-3-319-26532-2_6.

