Journal of Soft Computing Paradigm (ISSN: 2582-2640)
www.irojournals.com/jscp/

Fabric Fault and Extra Thread Detection using Convolutional Neural Network

Sowmiya A[1], Karunamoorthy B[2]

[1]Post Graduate Student, Department of Electrical and Electronics Engineering, Kumaraguru College of Technology, Coimbatore, India

[2]Associate Professor, Department of Electrical and Electronics Engineering, Kumaraguru College of Technology, Coimbatore, India

Email: [1]sowmiya.21mes@kct.ac.in, [2]karunamoorthy.b.eee@kct.ac.in

Abstract

A planar substance made of textile fibers is called fabric. The main reason why defective fabrics are produced is loom malfunctions. A specialized computer vision system called a fabric inspection system is used to find fabric flaws in order to ensure product quality. In this paper we classify the defect by using Convolutional Neural Network. Utilizing a special type of class-based ensemble convolutional neural network architecture, the defect recognition system is built. The experiment is carried out using several textile fiber kinds. There is four layers in CNN to classify the defect that is Convolution, ReLU, Pooling, Fully Connected layer. A number of well-known CNN architectures, such as Inception, ResNet, VGG, MobileNet, DenseNet, and Xception to classify the defect are tested in this study. Finally, The study demonstrates the result by classification and proves how accurately the defects are identified.

Keywords: Convolution, ReLU, Pooling, Inception Method, MobileNet.

1. Introduction

The usage of Artificial Intelligence (AI) software is a common way for producing art nowadays, and neural style transfer is essential for producing work with various aesthetics. In games as well as in films and pictures, neural style transfer is applied. Convolution neural networks are essential for both image classification and image creation because the machine

DOI: https://doi.org/10.36548/jscp.2023.2.005
Received: 01.05.2023, received in revised form: 01.06.2023, accepted: 11.06.2023, published: 21.06.2023

learns by taking the information from the images that are given into them. Three Images are used in the neural style transfer.

a) **Enter an Image:** The content image that is sent to the model

b) **Image of Style:** The preferred style over the supplied image

c) **Generate a Stylized Image:** This image serves as the final output image when the content and style have been integrated.

The ability to identify faults, excess thread, and classify defects during production is crucial for the fabric business. Traditionally, human-oriented fault identification has been a labor-intensive, time-consuming process. In this method, operator distraction, eye strain, and distraction can all result in significant losses in fabric production. This makes image processing and automatic fabric control methods based on artificial intelligence inevitable. The most crucial aspects of each layer of the image are captured by CNN, which is crucial in the style transfer process. The process begins with a blank image that is randomly populated with pixels and updated every epoch to serve as the result image. In each epoch, the similarity of this output image to the input image and the style image is evaluated. The loss is used to determine how comparable things are. The difference between the input and output images' content and style is referred to as content loss and style loss, respectively. A subclass of neural networks called convolutional neural networks (CNNs) have showed promise in tasks like classifying and identifying images. CNN has demonstrated success in identifying Fabric flaw problems, animals, face masks, objects, and track signs in addition to powering the vision in robots and self-driving automobiles.

A. Classification of Images

Under supervised learning, image classification entails placing an image in a category to which it belongs. A single tag is assigned to the entire image during image categorization.

B. Object Identification

Because photos contain multiple objects, the next step is to figure out where each object is located. All items in that image are detected as a consequence of object localization, and a

rectangle box is painted around them to highlight them. The bounding box is the rectangular rectangle that surrounds the items.

C. Detection of Objects

By detecting all the items in the image, drawing a rectangle box around it, and assigning a tag to each object based on the class to which it belongs, the object detection procedure combines image classification and object localization duties.

D. Segmentation

Segmentation is the process of splitting an image into smaller segments, each of which is a group of pixels represented by masks. The two different segmentation approaches utilized in the computer vision area are listed below.

a. **Semantic Segmentation:** The semantic segmentation task entails assigning a tag to each pixel of the image.

b. **Instance Segmentation:** Despite belonging to the same class, instance segmentation recognizes the objects in the image as distinct instances. The Study has investigated various transfer learning techniques for assigning a single label to the complete image.

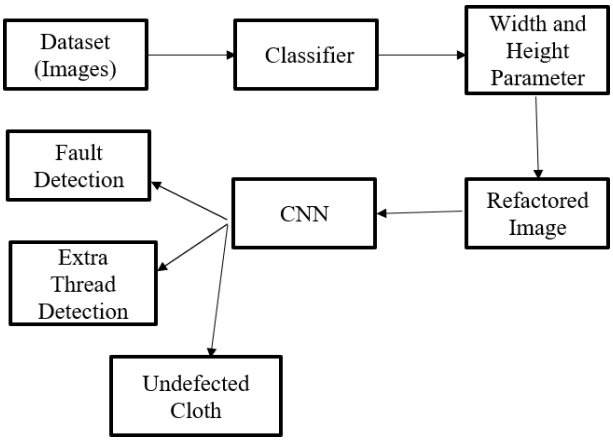

Figure 1. Proposed Block Diagram

2. Related Work

Neural style transfer incorporates the style with the input image to produce the output image. Neural style transfer techniques are used in gaming settings, photographs, and videos to improve their efficacy. These days, this method is applied to a variety of platforms, such as the Prisma app, which transforms photos into beautiful paintings. Users can upload specific content and style photographs to websites like deepart.io to make their own artwork.

According to recent study on neural style transfer, a style may be carried over from one picture to the entirety of a video. The neural style transfer techniques can also be used on applications that display the face with and without makeup.

3. Convolutional Neural Network

Convolutional layers are used to reduce the number of parameters in the image and speed up the training of the model.

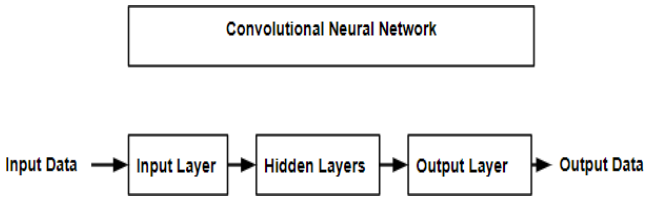

Figure 2. Basic Block of CNN

A. Input Layer

The CNN's input layer must contain image data. The representation of image data is done using three-dimensionalmatrices. Into a single column it must be changed. Before being fed into the input, a picture with a dimension of 30 30,or 900, must be transformed to 900 x 1. The dimension of theinput during training with "m" samples will be (900, m).

B. Convolutional Layer

This layer is also referred to as a feature extractor layer because it is where the properties of image are extracted.The Convo layer, which performs the convolution operation and figures out the dot product between the filter and the input Image's receptive

field, is connected to a section of the image (a local area with the same size as the filter). A single integer that reflects the output volume is the operation's output. Then, using a Stride, we repeat the process by advancing the filter over the following receptive region of the same input picture. Repeat the same steps over and over until the entire image has been processed. The input for the following layer will be the output.

C. ReLU Layer

All of the negative values from the filtered images are eliminated and replaced with zeros in this ReLU layer.

D. Pooling Layer

A pooling layer is used to decrease the spatial volume of the input image after convolution. It is used in the space between two convolution layers. Using FC after Convo layer without also applying pooling or max pooling will be computationally expensive. As a result, using maximum pooling is the only technique to reduce the spatial volume of the input image. The pooling layer doesn't have any parameter however it does contain two hyperparameters: Filter(F) and Stride (S).

a. Pick a window size.

b. Choose a stride.

c. Pass the window

d. From each window take the maximum value.

E. Fully Connected Layer

Neurons, weights, and biases are all present in fully linked layers. Neurons in one layer are linked to neurons in another layer in this way. It is employed to educate individuals how to classify the images into several groups.

F. Softmax Layer

The top layer is CNN's SoftMax or Logistic layer. It's located at the bottom of the FC layer. SoftMax is used for multi-classification, while logistic is used for binary classification.

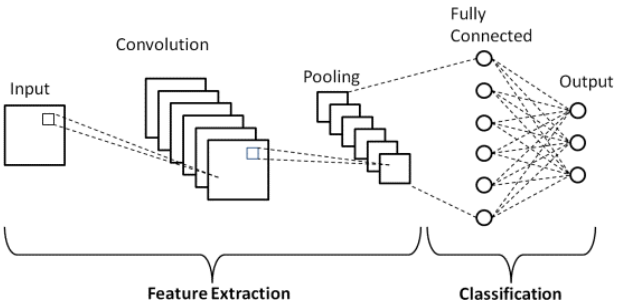

Figure 3. Layers of CNN

Convolutional Layer:

$$W_{out} = \frac{W - F + 2P}{S} + 1 \quad (1)$$

Padding Layer:

$$W_{out} = \frac{W - F}{S} + 1 \quad (2)$$

Flatten the image:

$$W * H * Pixel \quad (3)$$

Where,

W is the actual image size H is height of the image F is the filter size

P is the padding

S is the stride

4. Architecture

The InceptionV3 model is a 42-layer, pre-trained image classification model. Inceptionv3 delivers the most accurate classifications of the many Inception model variations. It has numerous layers that perform operations like as convolutions, average pooling, and maximum pooling. On the ImageNet dataset, the Inception model is assessed to be 78.1 percent accurate.

A. Inception Model – Operation

a) Convolution: The procedure of convolution is used to blend information. To change an image the convolution technique includes applying kernels to each pixel.

b) Pooling: The dimensions of the feature maps can be lowered by pooling.

B. Inception Model Architecture

a) Factorized Convolutions: The network should ultimately have fewer parameters but be computationally efficient. An inception network is designed to reduce the number of parameters while maintaining efficiency.

b) Smaller Convolutions: Other pre-trained models' convolutions are typically more complicated, resulting in longer training times. The inception model, on the other hand, replaces the bigger convolution with a smaller convolution, resulting in a reduction in complexity.

c) Asymmetric Convolutions: The Inception network employs 3*1 followed by 1*3 convolutions instead of 3x3.convolutions. The network's main goal is to use the two filters mentioned above to reduce the number of parameters.

d) Auxiliary Classifier: Auxiliary classifiers are employed in Inceptionv3 to increase the depth of the network. In the network, the classifiers act as regularizes.

e) Grid Size Reductions: This method is typically used during the pooling stage. This is done to limit the amount of feature maps that the max pooling process produces.

C. VGG-19 Pretrained Model:

It has correctly classified 1000 classes in the ImageNet Database is a pre-trained model produced. The ImageNet database has a total of 1000 picture classes, which are organized into three folders: a training set with 1.2 crore photos, a validation folder with 50,000 fruit images, and a test folder with 150,000 fruit images. VGG_19 is a sort of convolutional neural network that uses the ImageNet database to train and gives excellent results. Visual Geometry Group is abbreviated as VGG19.

The layers of the VGG19 model are depicted in the image

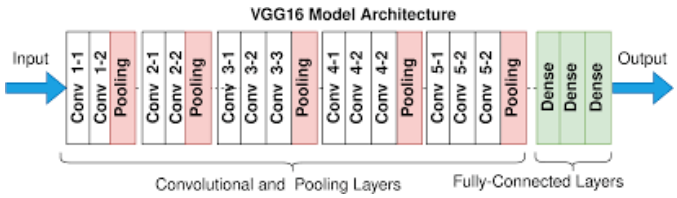

Figure 4. VGG16 Model Architecture

VGG-19 has approximately 19 layers, and images must transit through all 19 layers of CNN before being classified. VGG-19 has several 3x3 filters in each layer that can help extract useful information from a picture, such as edges and lines. The VGG-19 consists of 16 convolution layers, one SoftMax layer, three completely connected layers, five pooling levels, and five pooling layers. The categorized image is obtained from the SoftMax layer, which is the last layer. All of the layers of VGG-19 can be divided into three categories:

a) Convolutional Layer: Small 3x3 filters are applied to the input image in this layer, determining essential properties such as edge, line, and so on. This layer's filters recognize characteristics based on intensity changes. As a result, feature maps are produced by this layer.

b) Pooling Layer: Pooling is a crucial step in lowering the size of convolutional layer feature maps. Overfitting is mitigated by the pooling layer. Maximum pooling is used in VGG-19 to perform a unique type of pooling. With the most relevant data, max pooling results in feature maps.

c) Fully Connected Layer: While the previous two layers provide crucial information, this layer is critical in classifying the image into the appropriate class.

d) Purpose of VGG-16 Model: It is a CNN that has deep layers and work on pretrained version of the network and trained on millions of images from the ImageNet database.

5. Images Used in Implementation

In the dataset folder it contains defected and un-defected images and in the data preprocessing part it converts all the images into arrays. Then import directory and categories and create two list one is data and another one is labels then I append all the image into that list.

Dataset that contains holes, lines, stains are stored in defected image folder and one with no defects is stored in i un-defected image folder to classify the images using CNN.

A. Process for Image Classification

a. Use custom datasets.

b. Import all necessary modules, such as Pandas and NumPy.

c. Create a dataset directory (os.listdir).

d. Organize all of the photographs under one list with the appropriate label name.

e. Resize the image to a scale of (60*60*1) for the image classification procedure.

f. All the resized images are added to a list that is created with two list images and labels.

g. using np.array to turn each image into an array.

h. Labels are strings, therefore use the label encoder method to transform them to integers.

i. Provide a test size and divide the data into train and test groups.

j. Provide a hyperparameter for the architecture, such as batch size, learning rate, or epochs.

k. Use CNN to crop the image and look for flaws (w*h*c).

l. Layer: ReLU, maxpooling/average pooling, fully Connected, convolution.

m. Construct convnet (network).

n. Define optimizer and loss (Adam).

o. Calculate the accuracy after training and testing the algorithm.

Figure 5 (a). Defected Image (hole)

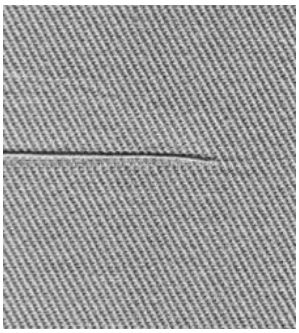

Figure 5 (b). Defected Image (line)

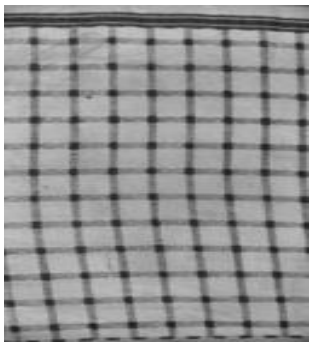

Figure 5 (c). Undefected Image

Table 1. Simulation Parameters

Parameter	Range
rotation_range	30
width_shift_range	0.2

height_shift_range	0.2
zoom_range	0.15
shear_range	0.15
Batch size	32
Horizontal flip	True
Fill mode	Nearest
Epochs	25
Color mode	Gray Scale

6. Simulation Results

The Proposed system produces an output for Image Classification it classifies whether it is lines, stains, holes or un-defected image then calculates the Loss, Accuracy, Validation Loss, Validation Accuracy of the dataset using the train and test algorithm.

Table 2. Simulation Results

Epochs	Loss	Accuracy	Validation Loss	Validation Accuracy
1	1.2293	0.4167	1.0517	0.5426
2	0.9601	0.6279	0.8654	0.6744
3	0.8129	0.6802	0.7488	0.7442
4	0.7128	0.7364	0.6668	0.7984
5	0.6369	0.7771	0.6165	0.8062

The output graph of validation accuracy shown in Fig. 6 (b)

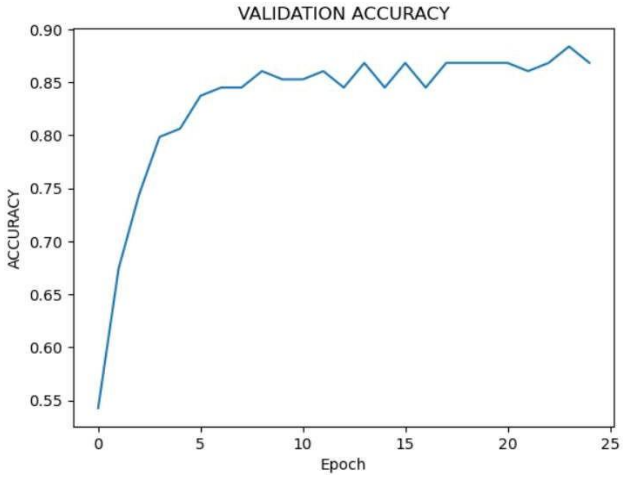

Figure 6 (b). Validation Accuracy

The output graph of model loss shown in Fig. 6 (c).

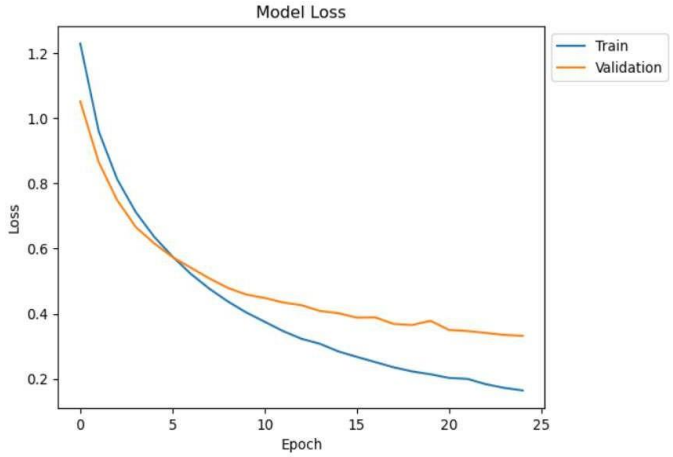

Figure 6 (c). Model Loss

The Classification of images results by entering the Datapath. When we enter the file path like

‘ . /dataset/holes/holes (12) .jpeg’

The image in the dataset folder displayed as an output.

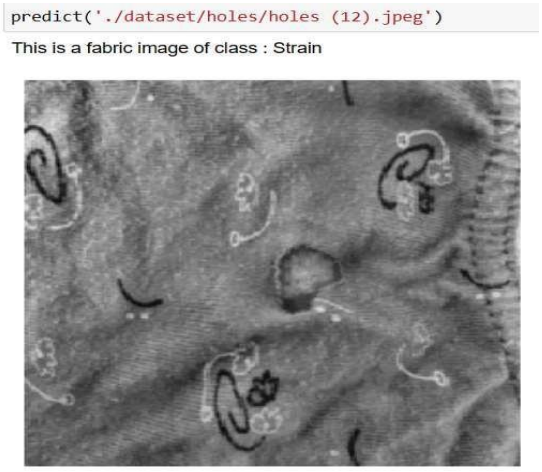

Figure 6 (d). Fabric Image of Class (strain)

‘ . /dataset/lines/lines (195) .jpeg’

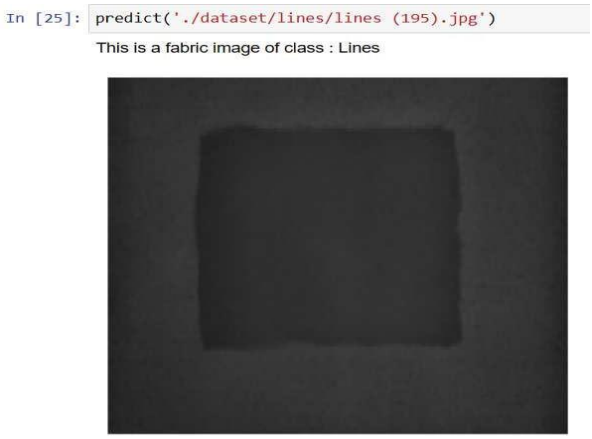

Figure 6 (e). Fabric Image (Lines)

‘ . /dataset/nodefect/nodefect (9) .jpeg’

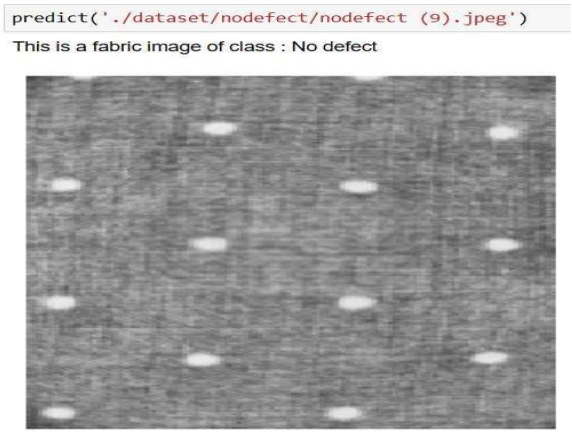

Figure 6 (f). Fabric Image (undefect)

7. Conclusion

The classification of defect like holes, strain, lines and un-defective cloth can be predicted by using Convolutional Neural Network. In the convolutional layer the study lines up the feature of the image. In the ReLU layer the negative values are removed and replaced with zeros. In the Pooling layer the image is shrunk into smaller size. It is Simulated by using Jupyter to classify the defect. Finally, the training and the testing of the datasets are carried out. The validation loss, Model loss and Accuracy observed for the proposed model shows that CNN based fault detection in fabric is efficient.

References

[1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, ‘‘Image quality assessment: From error visibility to structural similarity,’’ IEEE Trans. Image Process, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[2] M. Patil, S. Verma, and J. Wakode, ‘‘A review on fabric defect detection techniques,’’ Int. Res. J. Eng. Technol, vol. 4, no. 9, pp. 131–136, 2017.

[3] C.-H. Chan and G. K. H. Pang, ‘‘Fabric defect detection by Fourier analy- sis,’’ IEEE Trans. Ind. Appl, vol. 36, no. 5, pp. 1267–1276, Sep./Oct. 2000.

[4] Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, ‘‘Automated fabric defect detection—A review,’’ Image Vis. Compute, vol. 29, no. 7, pp. 442–458, 2011.

[5] Y. T. Ngan and G. K. H. Pang, ‘‘Regularity analysis for patterned texture inspection,’’ IEEE Trans. Autom. Sci. Eng, vol. 6, no. 1, pp. 131–144, Jan. 2009.

[6] Long, E. Shelhamer, and T. Darrell, ‘‘Fully convolutional networks for semantic segmentation,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit, Jun. 2015, pp. 3431–3440.

[7] L. Mak and P. Peng, ‘‘Detecting defects in textile fabrics with optimal Gabor filters,’’ Int. J. Comput. Sci, vol. 1, no. 4, pp. 274–282, 2006.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, ‘‘Dropout: A simple way to prevent neural networks from overfitting,’’ J. Mach. Learn. Res, vol. 15, no. 1, pp. 1929– 1958, 2014.

[9] F. Bianconi, L. Ceccarelli, A. Fernández, and S. A. Saetta, ‘‘A sequential machine vision procedure for assessing paper impurities,’’ Comput. Ind, vol. 65, no. 2, pp. 325–332, 2014.

[10] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, ‘‘Hypercolumns for object segmentation and fine-grained localization,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit, Jun. 2015, pp. 447–456.

[11] He, X. Zhang, S. Ren, and J. Sun, ‘‘Deep residual learning for image recognition,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit, Jun. 2016, pp. 770–778.

[12] S. Zheng et al., ‘‘Conditional random fields as recurrent neural networks,’’ in Proc. IEEE Int. Conf. Comput. Vis, Dec. 2015, pp. 1529–1537.

[13] O. Ronneberger, P. Fischer, and T. Brox, ‘‘U-net: Convolutional networks for biomedical image segmentation,’’ in Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent, Cham, Switzerland, 2015, pp. 234–241.

[14] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, ‘‘Automatic differentiation in PyTorch,’’ in Proc. 29th Annu. Conf. Neural Inf. Process. Syst. (NIPS), Long Beach, CA, USA, 2017, pp. 1–4.

[15] B. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, ‘‘Rethinking the inception architecture for computer vision,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.