Fault Detection in Fabrics(Woven & Non - Woven) using Convolutional Neural Networks

Submitted in partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Information Technology Discipline

Submitted To

SVKM's NMIMS,
Mukesh Patel School of Technology Management & Engineering,
Shirpur Campus (M.H.) (12pt)

Submitted By :(12)

Yash Uday Randive Enrollment No – 70011116037
Ajinkya Bhamre Enrollment No – 70011116005

Under The Supervision Of:

Mr. Prashant Udawant
Information Technology Department, Assistant Professor

DEPARTMENT OF INFORMATION TECHNOLOGY
Mukesh Patel School of Technology Management & Engineering
SESSION: 2019-20

CERTIFICATE

This is to certify that the work embodies in this Project entitled **"Fault Detection in Fabrics(Woven & Non-Woven) using Convolutional Neural Networks "** being submitted by

"Yash Uay Randive" (Roll No.: 70011116037)

"Ajinkya Bhamre" (Roll No.: 70011116005)

----

for partial fulfillment of the requirement for the award of "**Bachelor of Technology** in **Information Technology"** discipline to "SVKM'[s] NMIMS, Mumbai (M.H.)" during the academic year 2019-20 is a record of bonafide piece of work, carried out by him under my supervision and guidance in the **"Department of Information Technology", MPSTME, Shirpur (M.H.).** [14pt]

APPROVED & SUPERVISED BY:(14pt)

Mr. Prashant Udawant
(Information Technology Department, Assistant Professor)

FORWARDED BY(14pt)

(Dr. N.S. Choubey)
H.O.D., IT
MPSTME, Shirpur Campus

(Dr. Nikhlesh K. Sharma)
Director (Engineering Programs)
Shirpur Campus

DEPARTMENT OF INFORMATION TECHNOLOGY
Mukesh Patel School of Technology Management & Engineering

CERTIFICATE OF APPROVAL

The Project entitled **"Fault Detection in Fabrics(Woven & Non-Woven) using Convolutional Neural Networks"** being submitted by

"Yash Uday Randive" (Roll No.:70011116037)"

"Ajinkya Bhamre" (Roll No.:70011116005)"

has been examined by us and is hereby approved for the award of degree **"Bachelor of Technology in Information Technology Discipline"**, for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted. [14]

(Internal Examiner) **(External Examiner)**

Date: **Date:**

DEPARTMENT OF INFORMATION TECHNOLOGY
Mukesh Patel School of Technology Management & Engineering

DECLARATION

We,

Yash Uday Randive

Ajinkya Bhamre

The students of **Bachelor of Technology** in **Information Technology discipline**, S**ession: 2019-20, MPSTME, Shirpur Campus,** hereby declare that the work presented in this Project entitled **"Fault Detection in Fabrics(Woven & Non - Woven) using Convolutional Neural Networks"** is the outcome of our work, is bonafide and correct to the best of our knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any other university or anywhere else for the award of any degree or any professional diploma.

(Yash Uday Randive)
SAP ID.: 70011116037

(Ajinkya Bhamre)
SAP ID.: 70011116005

Date: April 07, 2020

DEPARTMENT OF INFORMATION TECHNOLOGY
Mukesh Patel School of Technology Management & Engineering

ACKNOWLEDGEMENT

After the completion of this Major Project work, words are not enough to express my feelings about all those who helped me to reach my goal; feeling above this is my indebtedness to The Almighty for providing me this moment in life.

It's a great pleasure and moment of immense satisfaction for me to express my profound gratitude to Prof. **Prashant Udawant**, **Assistant Professor**, Information Technology Department, MPSTME, Shirpur, whose constant encouragement enabled me to work enthusiastically. Their perpetual motivation, patience and excellent expertise in discussion during progress of the project work have benefited me to an extent, which is beyond expression. Their depth and breadth of knowledge of Information Technology field made me realize that theoretical knowledge always helps to develop efficient operational software, which is a blend of all core subjects of the field. I am highly indebted to them for their invaluable guidance and ever-ready support in the successful completion of this project in time. Working under their guidance has been a fruitful and unforgettable experience.

We express my sincere thanks and gratitude to Dr. N.S. Choubey, Head of Department, Information Technology Department, MPSTME, Shirpur, for providing necessary infrastructure and help to complete the project work successfully.

We also extend my deepest gratitude to Dr. R.S. Gaud, Director, Shirpur Campus and Dr. Nikhlesh Kumar Sharma, Director (Engineering Programs), Shirpur Campus for providing all the necessary facilities and true encouraging environment to bring out the best of my endeavors.

We sincerely wish to express my grateful thanks to all members of the staff of Information Technology department and all those who have embedded me with technical knowledge of Information Technology during various stages of B.Tech. Information Technology.

We would like to acknowledge all my friends, who have contributed directly or indirectly in this Major Project work.

The successful completion of a Major Project is generally not an individual effort. It is an outcome of the cumulative effort of a number of persons, each having their own importance to the objective. This section is a vote of thanks and gratitude towards all those persons who have directly or indirectly contributed in their own special way towards the completion of this project.

Yash Uday Randive
Ajinkya Bhamre

ABSTRACT

This paper is based on the implementation work done by the authors and the results of fabric fault detection. Automatic fabric inspection is important to maintain the fabric quality. For a long time the fabric defects inspection process is still carried out with human visual inspection, and thus, insufficient and costly. Hence the automatic fabric defect inspection system is required to reduce the cost and wastage of time caused by defects. The development of a fully automated web inspection system requires robust and efficient fabric defect detection algorithms. In this project we have proposed an effective method for detecting faults in fabrics. Here we are using Convolutional Neural Networks which would eventually result in better accuracy of fabric fault detection. Reports generated would eventually reduce the time for human inspection of overall daily work. To maintain accuracy we have created a dataset with equal amounts of defected and non-defective fabric images, furthermore the dataset includes both woven and non-woven fabrics. Our project classifies fabrics into defective/faulty and non-defective/not-faulty fabric images. Consequently, our model has achieved an accuracy of ≈81% (80.85%).

A more optimized version of our model would be one which classifies the type of faults in a fabric image as well as maps the location of the fault in the image thereby generating a report of the same. This functionality is expected to be achieved in the future.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 PURPOSE

Fault Detection is one of the final and most important steps involved in the fabric manufacturing process. The manual version of this process that is practiced in most manufacturing units has shown about 60% accuracy thereby making way for 40% inaccuracies in fault detection. Inaccuracies in fault detection in fabrics incurs enormous financial losses to fabric manufacturers due to rejection of the produced fabric by the customer because of the faults that were undetected during the manufacturing process. This report defines a basis to our research in the domain of **Fabric Fault Detection** that has led to the development and completion of our project. This project report consists of an accumulation of techniques(in the Literature Review) used to automate the fault detection process in fabrics using machine learning preceded by our methodology of solving the aforementioned problem by using **Convolutional Neural Networks** in order to automate the fabric fault detection process through images of the fabric whilst improving the accuracy of the same**.**

1.2 SCOPE

Future scope of this project lies in optimizing the algorithm further with a larger fabric dataset in order to increase its fault detection accuracy as well as integrate it with a **web-based portal that will be connected to a CMOS (Complementary Metal-Oxide Semiconductor) camera** which will continuously send pictures of the fabric roll to our web-portal that will **classify the fabric images as defective/not-defective** along with the position of the defect and its exact classification (line defect, hole defect, texture defect, oil defect, discoloration defect, etc.). The complete proposed service provided by our project in the future **can be monetized by charging fabric manufacturers a fee based on per fault detected by the system.**

1.3 OVERVIEW

This project report consists of a Literature Survey of 10 closely coherent papers out of the 40 research papers reviewed by the project team. Consequently, it consists of the methodology adopted by the project team to accomplish the proposed Fabric Fault Detection algorithm which involves the use of Convolutional Neural Networks to detect and classify faulty fabric images. Furthermore, information about the relevant activation, optimization and loss functions is also stated along with the description architecture of the CNN (Convolutional Neural Network) that has been built for the aforementioned purpose. In addition to this, snapshots of the algorithm’s code have been displayed under all relevant sections.

CHAPTER 2

LITERATURE SURVEY

[1] Adaboost learning for fabric defect detection based on HOG and SVM

In this paper,

1. Firstly the image is split into the image blocks with size 32 x 32.
2. The block-based features are encoded with histograms of oriented gradients (HOG) and get a 144-dimensions vector.
3. The Adaboost algorithm is performed to automatically select 32 features from the 144-dimensions which reduces the computation complexity. adaboost is performed to automatically select a small set of discriminative HOG (histogram of oriented gradients) features.
4. In the end, (SVM) Support Vector Machine is used to classify the fabric defects. Experimental results show us the efficiency of the algorithm.

[2]Fabric defect detecting and rank scoring based on fisher's criterion discrimination

In this paper, the essential surface of the fabric picture is broke down with the autocorrelation capacity and it is separated dependent on fisher's measure.

1. Before all else, nearby imperfections are broke down by quality assessment.
2. The size of the fabric surface units is gained by figuring the autocorrelation work in weft and wrap bearing.
3. Then the measures of dissected surface units are taken as a model to fragment the fabric picture.
4. Basic units are sectioned during filtering. The fisher standard is utilized to delegate units to each class simultaneously.
5. In the end, Fabric is being estimated by the relationship of picture pixel and size of the picture. At that point, they are positioned scale by America four-point framework.

[3]Fabric defects segmentation approach based on texture primitive

In this paper, two types of defects of fabric texture are classified, which are intensity defects and geometric defects. An intensity defect can cause a noticeable change in the gray-level intensity values, such as oil stain, hole, etc. A geometrical defect is a distortion, which does not significantly change the gray level histogram at that point but rather changes the existing spatial correlation between the pixels, such as, slack end, etc. Some defects have both characteristics, such as knees, temple marks, etc.

1. The auto-correlation curve presents a better periodicity and texture primitive by a vertical and horizontal periodicity.
2. The texture primitive template is calculated after getting the size of the texture primitive.
3. The original image is subdivided into blocks, and then the gray mean of every corresponding pixel in each block is calculated according to the formula.
4. The automatic inspection fabric defects is gradually to attenuate background, which is the normal texture regions, and the defected region is accentuated at the same time so that the defects are detected at last. After that last stage is constructing the mean value image and automatic threshold segmentation using Ostu's approach.

[5]Fabric Defect Detection Based on Biological Vision Modeling

In this paper, imperfection detection depends on natural vision demonstrating and low-position portrayal with a recreation of organic vision, which can follow notable articles.

It is motivated by various leveled data handling of the organic visual framework, a fabric deformity system by demonstrating the natural vision. A descriptor dependent on coding retinal ganglion cells are utilized to portray a wide range of fabric surfaces and the low-position portrayal is embraced by model visual saliency. The word reference learning mix in the low-position model is to denoise the saliency map and a laplacian regularization term is coordinated into the low-position portrayal model to develop holes between abandoned area and foundation. To improve productivity, LADMAP rather than ADMM is embraced for the built model. The quality and amount of test results proposed a technique which is progressively successful, strong and versatile.

[6] Auto encoder-Based Fabric Defect Detection with Cross-Patch Similarity

In this paper, to maintain the texture area in the reconstructed defective patch, a novel auto encoder with cross-patch similarity is proposed for fabric defect detection. To make use of the repetitive texture patterns in the test image, the similar non-defective patches for each candidate defective patch are found, and their corresponding latent variables are weighted combined to be a reference for modifying the original latent variable of the candidate patch. By iterative update process, the output for each defective patch decoded by the modified latent variable will be non-defective, thus the defect area in the patch can be effectively detected from the reconstruction residual.

7]Fabric Defect Detection Based on Pattern Template Correction

In this paper, novel format dependent on the amendment of imperfection detection techniques is utilized on the pictures which contain occasional structure. At the outset, fabric pictures are divided into cross-sections. In light of misalignment, impacts are being decreased. Grids that are without deformity are picked as the normal layout for uniform reference. With the arrangement between deformity free and flawed examples abandons are recognized. The characterization includes an improved E-V strategy with layout based rectification and unified preparation. The subsequent stage, which is a flawed shape laying out furnishes pixel-level outcomes with the assistance of limit division.

[8]Fabric Fault Detection Using Digital Image Processing

In this paper, in the fabric are recognized by the Automatic Fabric fault examination technique. Because of this strategy, at the hour of assembling itself we get top-notch fabric, it suggests the rapid creation. At the hour of assembling fabric in material, the faults present on fabric are recognized by MATLAB programming utilizing some Image

Processing systems. Picture Processing is extremely useful in light of the fact that all the systems applied to the faulty picture is valuable to get fault free picture. Since quality assessment is a significant part of mechanical assembling. A fabric fault can happen because of machine faults, gap, shading dying, yarn issues, scratch, poor completing, soil spots, unnecessary extending, breakpoint. In picture handling, right off the bat, a picture is gained of a particular piece of fabric utilizing a camera. The procured picture is then proposed to the MATLAB programming in which the picture is put away for additional preparation. The preparing activities are performed on the picture, for example, dark transformation, histogram, histogram adjustment, paired change, include extraction-edge and limit highlight, and so on.

[9]Automated Fabric Defect Detection

In this paper, absconds are distinguished utilizing Image handling and neural systems. The outcomes showed that the utilization of light shafts dependent on the shade of fabric material is more compelling than the white light bar. A preprocessing step is required to address the non-uniform brightening and picture de-noising. It evacuates lopsided light of the picture brought about by sensor defaults (vignetting), non-uniform brightening of the scene, or direction of the surface. Morphology is characterized right now a lot of picture handling tasks that procedure pictures dependent on shapes. Expansion and disintegration are essential morphological activities. Enlargement adds pixels to the limits of articles in a picture while disintegration evacuates pixels on object limits. After the utilization of morphological changes, the size of the picture was changed over into grayscale to decrease the clamor. Thresholding is utilized to make a parallel picture from a grayscale picture. The camera catches the pictures of moving fabric with a goal of 544x548. the edge grabber is accustomed to synchronizing camera catch recurrence with the speed of the creation line so as to catch the whole fabric. Convolutional Neural Network (CNN) API is utilized for arranging abandons.

[10]Implementation of Fabric Fault Detection System Using Image Processing

In this paper, the obtaining of picture division is performed and isolates the advanced pictures into various portions. It is utilized to find articles and limits in the picture. A district-based division procedure is utilized right now. After the picture is isolated in portions, the estimation of pixels of both of the standard sections is put away in the preparation dataset. At runtime when the picture is caught it is isolated in the sections. Isolated portions are then contrasted and the coordinating section in the preparation dataset. Fragments vary in esteem and are considered as faulty portions. Shading vision is isolated utilizing RGB shading space. RGB shading space depicts hues as far as the pixel's an incentive as far as Red, Green and Blue shading. RGB values are then changed over into HSV values. Which depicts pixel shading as far as tone, immersion, and worth. The HSV esteems are then utilized in the code to decide the highlights of every pixel and afterward put away in support. A transport line with an infrared sensor is utilized to recognize whether there is the material present on a transport line. In the event that present, at that point the camera takes video feed. Web camera snatches casings of fabric present on the transport line. Pixel estimations of the edge are sent to a PC through JMyronlibrary. It gets 10 edges for every second. By this strategy, fabric faults are identified with a precision of 96.15%.

CHAPTER 3

PROBLEM DEFINITION & PROPOSED SOLUTION

3.1 PROBLEM STATEMENT

Existing manual methods of fault detection in fabrics require active manpower and are extremely inaccurate(accuracy ≈ 60%) which account for 40% inaccuracies in the process of fault detection in fabrics that incurs high financial losses to fabric manufacturers due to the production of a low quality/defective product. Furthermore, the cost of maintaining human resource at a factory specifically for the task of fault detection costs fabric manufacturers money as well as space in their manufacturing units.

The problem statement of this project revolves around how to provide an automated solution to the process of fault detection in fabrics in order to increase accuracy of the process thereby decreasing the requirement of human resource and the cost of the same.

3.2 PROPOSED SOLUTION

The proposed solution to the aforementioned problem statement pertaining the Fault Detection in Fabrics is as follows,

1. **Using fabric images to detect faults through an algorithm programmatically**
2. **Building a dataset of faulty and non-faulty fabric images to train, validate and test a neural-network based model**
3. **Building a Convolutional Neural Network model to classify images as defective/not-defective and providing an output in the form of '0' for not-defective fabric and '1' for defective fabric**
4. **Integrating the aforementioned model with a web-based application to be paired with CMOS Cameras that click pictures of the fabric roll and piloting the system in fabric manufacturing units as a proof-of-concept in the future.**

CHAPTER 4

DESIGN

4.1 ARCHITECTURAL DIAGRAM

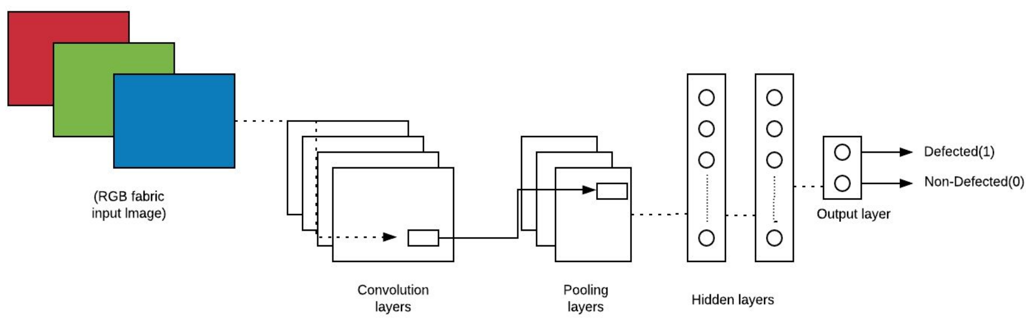

Fig. 4.1.a Architectural Diagram of the CNN

1. **RGB Fabric Input Image :** An RGB (128 x 128 x 1) fabric image will be fed to the model as input
2. **Convolution Layers :** These layers extract features from the fabric image at a pixel level
3. **Pooling Layers :** These layers calculate the maximum value for each patch of the feature map.
4. **Hidden Layers :** Layers of the CNN contributing to the Prediction as well as backpropagation
5. **Output Layer :** The Output Layer has two nodes for the only two probable outputs i.e. ‘0’ and ‘1’. 0 => Non-Defective and 1 => Defective

4.2 DATA FLOW DIAGRAM

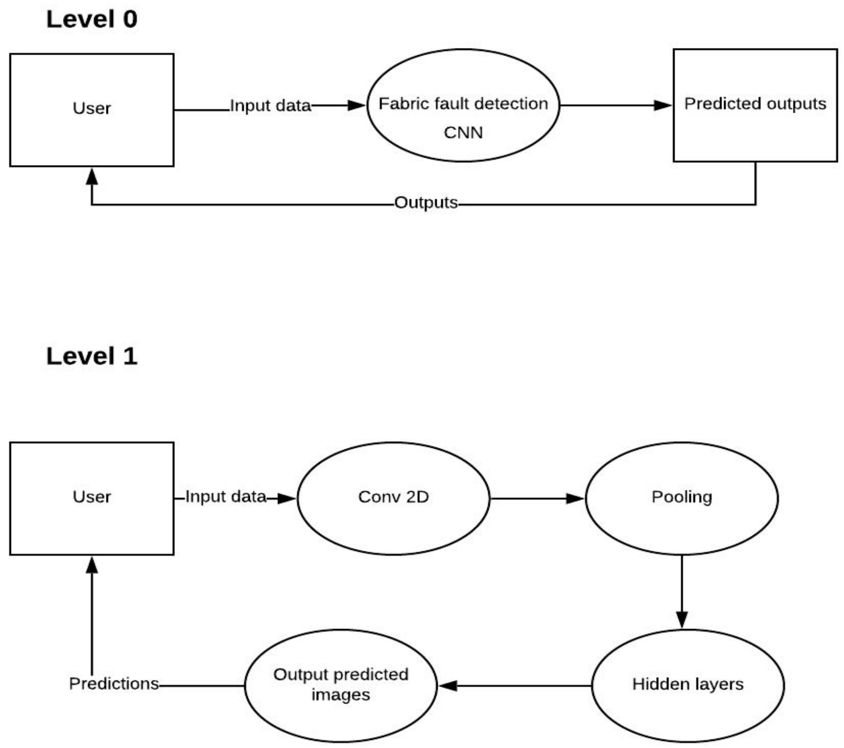

Fig. 4.2.a DFD Level 0 (top) ; DFD Level 1(bottom)

From the DFDs shown in **Fig. 4.2.a** It is clear that the image data(inputs) generally **propagates forward** in the algorithm, although it is important to note that **backpropagation** may take place within the **Hidden Layers** of the CNN.

4.3 USE CASE DIAGRAM

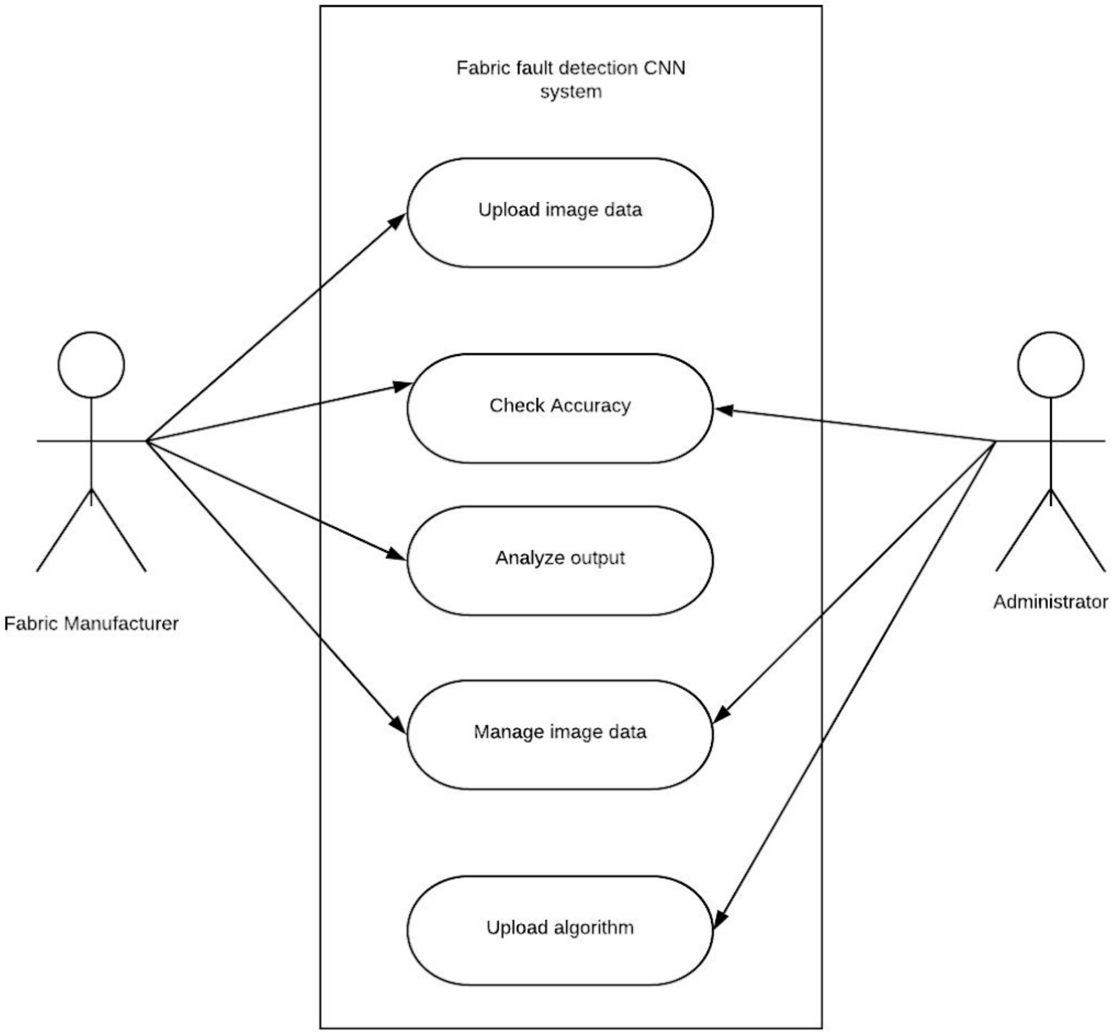

Fig. 4.3.a Use Case Diagram of the Proposed System

According to the Use Case Diagram shown in **Fig. 4.3.a** the roles of a Fabric Manufacturer and Administrator are evident. Notably, the Administrator should be uploading the updated algorithm into the system from time to time to increase the accuracy of the outputs of the system. Furthermore, the Fabric Manufacturer who is the end user of the system should upload Images of a minimum size 64x64 pixels to obtain optimum results.

4.4 ACTIVITY DIAGRAM

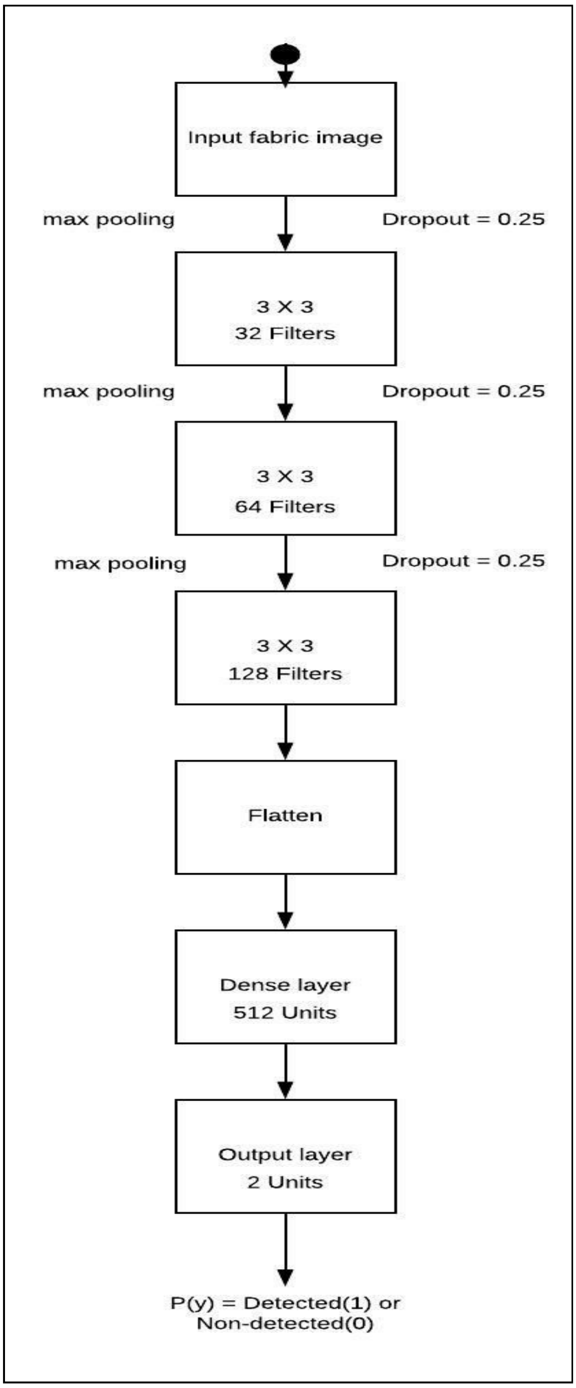

Fig. 4.4.a Activity Diagram of the CNN

The activity diagram shown in **Fig. 4.4.a** depicts the activities/functions in the various phases of the Convolutional Neural Network.

1. **Input Fabric Image :** The input is expected to be an RGB Fabric Image of a minimum size 64 x 64 pixels

2. **3 x 3 Kernels, 32/64/128 Filters :** These different convolution kernels each act as a different filter creating a channel/feature map representing something different. For example, kernels could be filtering top edges, bottom edges, diagonal lines and so on. In our CNN these kernels are filtering to fabric features such as wefts, warps and the inconsistencies within them.
3. **Flatten :** The flatten layer converts multi-dimensional image tensors to one single dimensional array that is processed by the model
4. **Dense Layer :** The dense layer contains the interconnected hidden layers of the CNN that contribute to the predictions in the output layer. Furthermore, the weights of these hidden layers constitute a great part of the deployed model.
5. **Output Layer :** This layer is the layer that displays the final predicted output. It consists of two nodes as the only two favourable outputs are '0' for a non-defected fabric image and '1' for a defected fabric image

CHAPTER 5

RESULT ANALYSIS/ IMPLEMENTATION

5.1 DATASET CREATION

The dataset for this project was created by both the team members Ajinkya Bamre and Yash Randive as no faulty fabric dataset is readily available on the internet. We collected individual images of non-defective as well as defective fabrics from google images and personal blog sites on blogspot.com. A total of 1000 images were collected and compiled into a training and testing dataset. The validation dataset was then programmatically created out of the train dataset in the CNN algorithm.

Each fabric image in the compiled dataset was renamed according to its nature i.e. defected0001, not-defected0001, etc.

5.2 DATA PREPROCESSING

In this step a **Pandas DataFrame** containing all the images was classified into two classes - **Defected and Not-Defected.** Since the classification was categorical we needed to map both the aforementioned classes to **'1'** and **'0'** respectively via the **DataFrame.replace({})** method.

In addition to this, the **ImageDataGenerator** class from the **keras.preprocessing.image** library was used to create multiple images out of one single image. The ImageDataGenerator class,

1. **Flips the image**
2. **Rotates the Image**
3. **Zooms the image**
4. **Shrinks the image**
5. **Crops the image**

Hence, the ImageDataGenerator creates **multiple images out of one image** that **adds up to the dataset.**

```
In [13]:
df["category"] = df["category"].replace({0: 'not-defected', 1: 'defected'})
```

```
In [14]:
train_df, validate_df = train_test_split(df, test_size=0.20, random_state=42)
train_df = train_df.reset_index(drop=True)
validate_df = validate_df.reset_index(drop=True)
```

```
In [15]:
train_df['category'].value_counts().plot.bar()
```

```
Out[15]:
<matplotlib.axes._subplots.AxesSubplot at 0x7f5fc4985710>
```

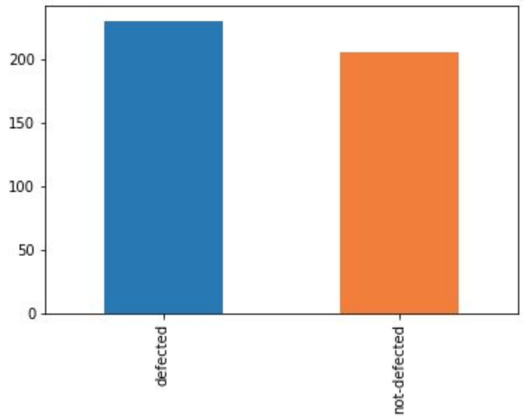

Fig. 5.2.a Using the DataFrame.replace({}) method to map defected and not-defect images in the dataset to numeric values 1 and 0 respectively

Training Image Generator

```
In [18]:
train_datagen = ImageDataGenerator(
    rotation_range=15,
    rescale=1./255,
    shear_range=0.1,
    zoom_range=0.2,
    horizontal_flip=True,
    width_shift_range=0.1,
    height_shift_range=0.1
)

train_generator = train_datagen.flow_from_dataframe(
    train_df,
    "../input/test-fabric/",
    x_col='filename',
    y_col='category',
    target_size=IMAGE_SIZE,
    class_mode='categorical',
    batch_size=batch_size
)
```

```
Found 435 images belonging to 2 classes.
```

Fig. 5.2.b Using the ImageDataGenerator class to generate multiple modified images out of one image

5.3 OUTLINING/BUILDING THE CONVOLUTIONAL NEURAL NETWORK MODEL

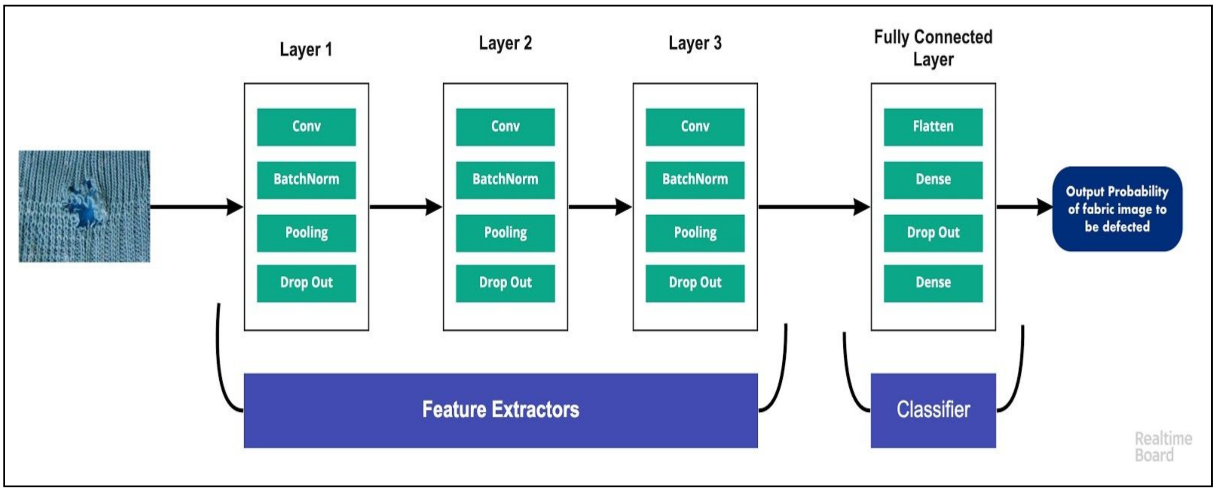

Fig. 5.3.a The CNN Model

1. **Input Layer:** It represents input image data. It will reshape the image into a single dimension array. Example the fabric image is 64x64 = 4096, it will convert to a (4096,1) array.
2. **Conv Layer:** This layer will extract features from image. This layer creates a convolution kernel that is wind with layers input which helps produce a tensor of outputs.
3. **Pooling Layer:** This layer will reduce the spatial volume of the input image after convolution. Pooling layers provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map.
    - Maximum Pooling(Max Pooling): Calculate the maximum value for each patch of the feature map.
4. **Fully Connected Layer:** It connect the network from a layer to another layer
5. **Output Layer:** It is the predicted values layer and will consist of 2 output nodes, one for each 0(not-defected) and 1(defected) image probability.

Batch Normalization : The layer will transform inputs so that they are standardized, meaning that they will have a mean of zero and a standard deviation of one. During training, the layer will keep track of statistics for each input variable and use them to standardize the data.

Dropout : Dropout is a technique used to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase.

Activation = 'relu' : The ReLu activation function is used in all the layers except the output layer as ther ReLu function returns the exact probability of either 0 or 1 that affects the weights for the next layer as well as affect the model while backpropagating.

Activation = 'softmax': Only the output layer uses the softmax activation function as it gives probability ranging between 0 and 1 as it's output. Hence, defected fabrics were mapped to 1 and not-defected fabrics were mapped to 0 for ease in understanding the prediction as well as feeding it to the model.

Flatten() : The flatten method is to convert a 3D Image (RGB Tensor) into a 1D Array which is easy for the model to understand and process.

Filters : We have mainly used 3 filtering layers in the CNN,

1. 32 filters , 3 x 3 kernel
2. 64 filters, 3 x 3 kernel
3. 128 filters, 3 x 3 kernel

Loss Function : 'categorical_crossentropy' as we have previously one-hot encoded the data.

Optimizer : 'rmsprop' - Root Mean Square Propagation : Here the weights function, **ΔWi(t)** is not a monotonously increasing function; where **'Wi'** is the weight index and **'t'** is the iteration at which the weight is being updated.

```
model.compile(loss='categorical_crossentropy', optimizer='rmsprop', metrics=['accu
racy'])
```

Fig. 5.3.b Selecting the Loss Function and the Optimizer

5.4 Fitting the Model

After creating the required amount of input, output and hidden layers of the model as well as defining the Loss Function and Optimizer to be used, the model needs to be trained on the training data, validated on the validation data and finally tested on the testing data.

```
In [162]:
epochs=3 if FAST_RUN else 15
history = model.fit_generator(
    train_generator,
    epochs=epochs,
    validation_data=validation_generator,
    validation_steps=total_validate//batch_size,
    steps_per_epoch=total_train//batch_size,
    callbacks=callbacks
)
```

Fig. 5.4.a Code to Fit the Training Data Into the CNN Model

The code snippet mentioned above infuses 15 units of data from the dataset into the algorithm in each epoch (iteration) to train the data and gives an output for each epoch as shown below :

```
Epoch 00014: ReduceLROnPlateau reducing learning rate to 6.25000029685907e-05.
Epoch 15/15
29/29 [==============================] - 4s 144ms/step - loss: 0.5342 - acc: 0.7629 - va
l_loss: 0.5041 - val_acc: 0.8085
```

Fig. 5.4.b Output after Fitting the Model

The output shown above shows the training loss to be 0.5342 while validation loss = 0.5041, furthermore the **Validation Accuracy = 80.85% ≈ 81%**. In addition to this it was observed that a learning rate of **6 .25000029685907e-05** was the optimal learning rate for the model.

Consequently, the **EarlyStopping** hyper parameter was set to a **patience = 10** in order to prevent the model from **overfitting**.

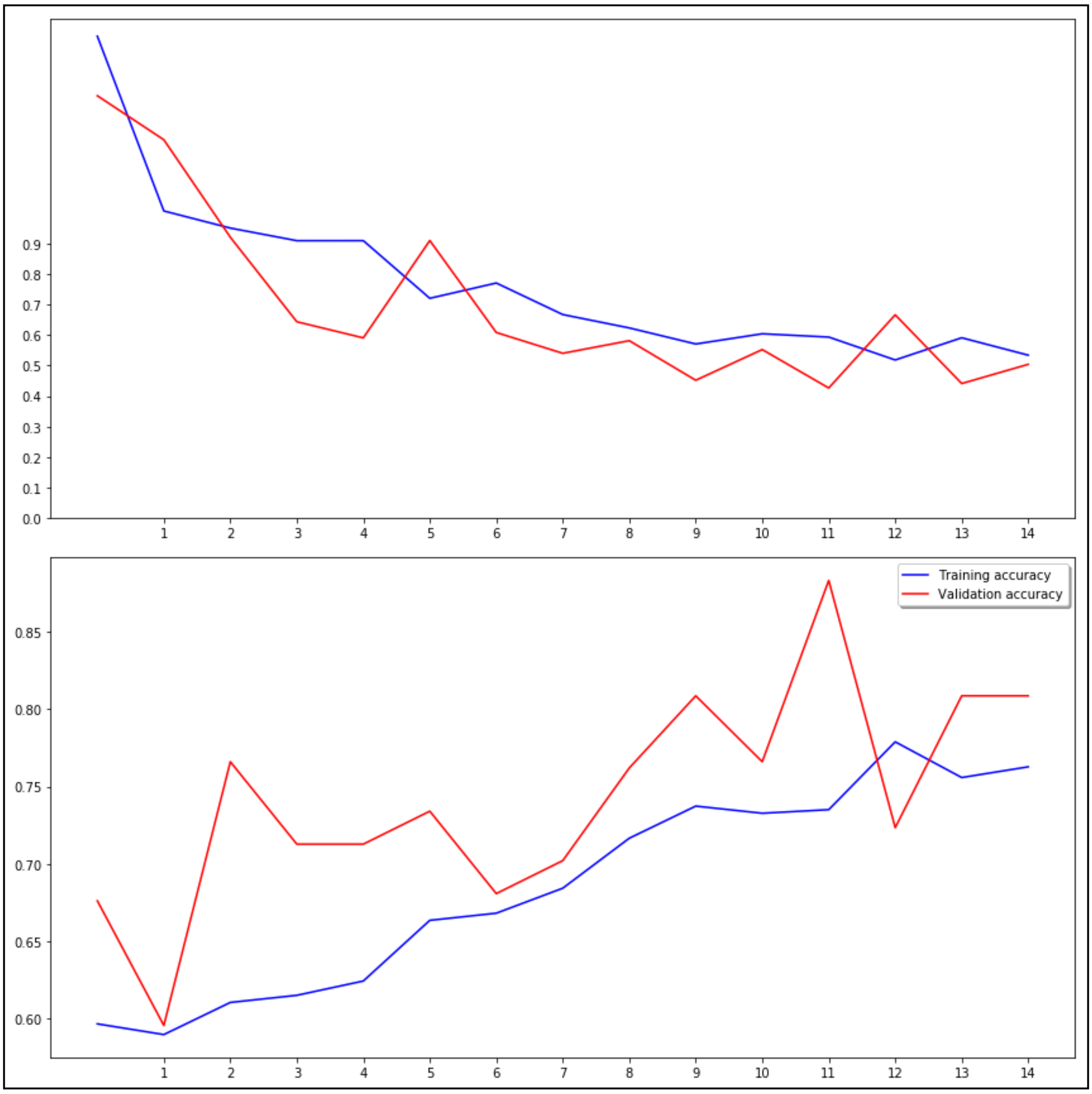

Fig. 5.4.c The figure on top depicts the Validation Loss(red) v/s Training Loss(blue), The figure on the bottom depicts the Validation Accuracy(red) v/s Training Accuracy(blue)

CHAPTER 6

TESTING

6.1 MODEL TESTING

After training the model on the train and validation dataset, the testing dataset was loaded into the model in a similar manner. Notably, the model had never 'seen' the test images. The test results obtained were as follows :

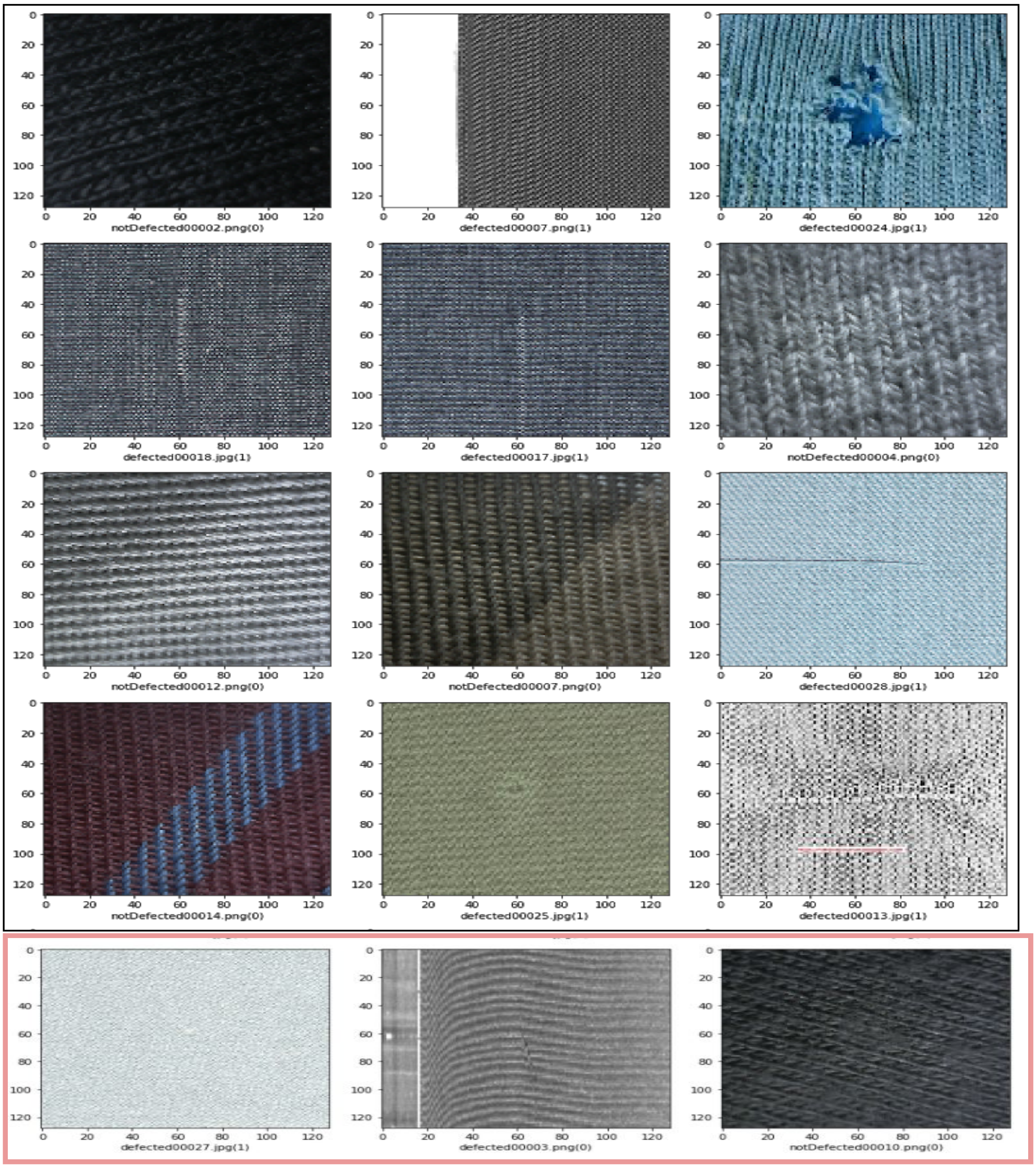

Fig. 6.1.a Test Outputs (defected00003.png was the image predicted incorrectly)

In the test outputs shown in **Fig. 6.1.a** a sample of **15 fabric images** was fed into the model out of which **14 times the model predicted correctly** while for one image it gave a wrong prediction.

6.2 PREPARING MODEL FOR DEPLOYMENT

After training, validation and testing and receiving ≈81% accurate predictions, the weights of the CNN were saved for deployment purposes. The code snippet for saving weights of the CNN is as shown below:

Save Model

In [163]:
```
model.save_weights("model.h5")
```

Fig.6.1.b Saving the weights of the model

Now, using these weights developers can easily integrate the model into their software applications.

CHAPTER 7

CONCLUSION & FUTURE WORK

7.1 CONCLUSION

We have successfully developed a Convolutional Neural Network for the purpose of identifying and classifying fabric images into Defective & Non- Defective images with an accuracy of **≈81%**. Furthermore, we have also compiled a dataset of over defective fabric images that can be expanded further to give way to more degrees of classification as well as increase the accuracy/optimize the model.

In addition to this, we have achieved an accuracy **≈21%** more than that of traditional, manual methods of fault detection in fabrics and are at a position to optimize the model further and integrate it with relevant hardware and software components in order to carry out our first proof-of-concept in the market.

7.2 FUTURE WORK

1. Dataset can be expanded in order to make way for more degrees of classification in fabric faults
2. The model can be integrated in a web-based portal connected to a CMOS camera at manufacturing units to detect faults in fabrics in real time
3. The model can be optimized to show the exact position of the faults in the fabric image and generate a relevant report about the faults
4. Once all the aforementioned changes are made, the model can be licensed to successful businesses in the textile industry.

REFERENCES

[1] Adaboost learning for fabric defect detection based on HOG and SVM- Ding Shumin, Liu Zhoufeng, Li Chunlei School of Electric & Information Engineering, Zhongyuan University of Technology Zhengzhou, China

[2] Fabric defect detecting and rank scoring based on fisher's criterion discrimination-SHENG-WANG LI, LI-WEI GUO, CHUN-HUA LI College of Information Science and Technology, Hebei University of Science and Technology, Shijiazhuang 050018, China

[3] Fabric defects segmentation approach based on texture primitive- SHUANG-WU ZHU , HONG-YANG HAO , PENG-YANG LI , MEI-HONG SHI , HUA QI. College of Electronic Information, Northwestern Polytechnical University, xi'an 710072,China . College of Textile and material, Xi'an Polytechnic University , xi'an 710048,China;

[4]Sparse Dictionary Reconstruction For Textile Defect Detection-Jian Zhou,Dimitri Semenovich,Arcot Sowmya and Jun Wang. College of Textiles, Donghua University, Shanghai, China

[5] Fabric Defect Detection Based on Biological Vision Modeling -CHUNLEI LI 1 , GUANGSHUAI GAO1,2, ZHOUFENG LIU1 , MIAO YU 1 , AND DI HUANG 2 1School of Electronic and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China 2School of Computer Science and Engineering, Beihang University, Beijing 100191, China

[6] Auto encoder-Based Fabric Defect Detection with Cross-Patch Similarity-Hu Tian, Fei Li Fujitsu Research & Development Center Co., Ltd., Beijing, China

[7]Fabric Defect Detection Based on Pattern Template Correction-Xingzhi Chang , Chengxi Gu , Jiuzhen Liang , and Xin Xu

[8]Fabric Fault Detection Using Digital Image Processing-Farida S.Nadaf, NayanaP.Kamble, RohiniB.Gadekar Under The Guidance of: P.S.Kadu Department of Electronic and telecommunication Fabtech Technical Campus, College of Engineering and Research,Sangola, India

[9]Automated Fabric Defect Detection-Prasanna Bandara, Thilan Bandara, Tharaka Ranatunga, Vibodha Vimarshana, Sulochana Sooriyaarachchi and Chathura De Silva Department of Computer Science and Engineering University of Moratuwa, Sri Lanka.

[10]Implementation of Fabric Fault Detection System Using Image Processing-Sunil L. Bangare, N. B. Dhawas, V. S. Taware, S. K. Dighe, P. S. Bagmare Assistant Professor, Department of Information Technology, Sinhgad Academy of Engineering, Pune & Research Scholar, Department of Computer Science & Engineering, K. L. University, Guntur (A.P.), India Department of Information Technology, Sinhgad Academy of Engineering, Pune, India,