National Academy for Computer Training and Research (NACTAR)

Research Report

On

Image Processing and Artificial Intelligence (AI) for Fabric Defect Detection in the Textile Industry of Bangladesh

Reference No: 57.21.0000.008.38.002.22.5

Submitted to

The Director
National Academy for Computer Training and Research (NACTAR), Bogura.

Submitted by

CAPITA FINTECH

Powered by BengalAI

Executive Summary

The textile industry in Bangladesh, a significant sector of the country's economy, faces challenges in quality control and inspection, particularly in fabric defect detection. Traditional manual inspections are time-consuming, labor-intensive, and prone to errors, leading to significant losses for manufacturers. To address this, Capita Fintech has developed a fabric defect detection methodology utilizing cutting-edge technologies such as deep learning and transfer learning models.

The methodology begins with data collection and preparation, where a substantial number of fabric images, including both defective and normal samples, were gathered from our industry partner's facility. Despite some challenges, the Capita Fintech team (hereafter, referred as team or the team) successfully obtained a diverse dataset by adjusting camera settings and enhancing image quality through processing techniques.

Model development involved investigating two types of deep learning models: pure convolutional neural network (CNN) based models and transfer learning models based on pre-trained architectures. After extensive experimentation, the MobileNet model emerged as the most effective choice, achieving exceptional performance metrics such as high Area Under Curve (AUC) AUC scores (0.9886 for training and 0.9981 for validation) and accuracy values (0.9518 for training and 0.9680 for validation). This model demonstrates excellent precision (0.9502 for training and 0.9837 for validation) and recall (0.9575 for training and 0.9528 for validation), indicating its ability to minimize false positives and detect true positives accurately.

Evaluation of the models on video frames highlighted challenges due to lower resolution, compression, and temporal complexities. It emphasized the need for training the model on a more diverse dataset comprising various video sources to enhance performance. Capita Fintech acknowledges this and commits to securing additional resources to improve the model's performance on video frames, contributing to further advancements in fabric defect detection.

By implementing the fabric defect detection methodology, textile manufacturers in Bangladesh can significantly improve product quality, reduce inspection time, enhance productivity, and minimize defects. This, in turn, positively impacts their reputation, profitability, and competitiveness in the global market. Furthermore, the methodology aligns with the Smart Bangladesh Mission, which aims to leverage technology for economic growth and digital transformation.

In conclusion, Capita Fintech's fabric defect detection methodology offers a powerful solution to the challenges faced by the textile industry in Bangladesh. The utilization of deep learning and transfer learning models, specifically the MobileNet based transfer learning model, showcases impressive performance in accurately identifying fabric faults. This methodology provides valuable insights for future research and establishes a solid foundation for leveraging technology to drive improvements in fabric quality control and inspection processes.

Table of Contents

Table of Figures

Table of Tables

Background:

The Fourth Industrial Revolution, also known as 4IR, is currently underway and bringing about significant changes to the global landscape. One of the most noticeable impacts of this revolution is the transformation of businesses and industries, with the textile industry being no exception [1]. The textile industry is one of the most important sectors of the economy in Bangladesh. It is the largest manufacturing sector and accounts for more than 80% of the country's total exports [2]. However, the industry faces several challenges, including quality control and inspection. One of the major challenges in the textile industry is fabric defect detection, which is mostly accomplished by visual inspections in Bangladesh. This process is time-consuming, labor-intensive, and prone to errors. Manual defect detection is less effective than that of automatic detection as approximately 70% of defects can be identified with the most skilled inspection worker [3]. Fabric defects can be caused by numerous factors such as weaving or knitting defects, dyeing defects, printing defects, or finishing defects [4]. These defects can result in significant losses for textile manufacturers as they reduce the quality of the fabric and can lead to rejected products, which in turn can affect the manufacturer's reputation and profitability. In the literature, it is reported that fabric defects can cause reduced fabric prices which can cause up to 45%-65% losses to the producer [5].

By harnessing the power of cutting-edge technologies like Artificial Intelligence (AI) and other digital tools, textile manufacturers can be more efficient and produce less faulty products which can make significant improvements in both quality and quantity in their production processes, leading to better products and services for their customers.

With the advancements in computer vision and machine learning algorithms, it is now possible to develop automated systems that can accurately detect fabric defects in real-time, reducing the need for manual inspection and improving the overall quality of the fabrics produced. By implementing such systems, textile manufacturers can reduce inspection time, improve productivity, and minimize defects, resulting in cost savings and increased customer satisfaction [1], [3], [6].

In Bangladesh, the government has launched the Smart Bangladesh Mission, which aims to transform the country into a digital economy by leveraging the power of technology. The mission aims to improve the quality of life for citizens and accelerate economic growth by focusing on key areas such as digital infrastructure, e-governance, and digital skills development [7].

The Smart Bangladesh Mission is intricately linked to the 4IR as it seeks to leverage the power of digital technologies to drive growth and development in various industries, including the textile industry [8]. By adopting technologies such as AI and image processing for fabric defect detection, textile manufacturers in Bangladesh can not only improve the quality of their products but also enhance their competitiveness in the global market. This, in turn, can contribute to the overall economic growth of the country.

There are various techniques that have been reported in the literature for defect detection using various machine learning models [9]. Classical Machine Learning Algorithms like Dictionary Learning Based (DLB) Algorithm can be used to identify the fabric defects and effectiveness of this method which has been validated by Kang and Zhang [14]. This method learns from training or test image and then from fabric image without defect. The presented method by Li et al [15] uses eigenvalue decomposition instead of singular value decomposition (SVD) for dimensionality reduction, making it easy to implement and effective with adequate image contrast.

Due to recent developments in processing and GPU based computation power, neural network-based methods, especially deep learning-based methods demonstrated immense potential for model fault detection. One of the most used deep learning techniques for fabric defect detection is Convolutional Neural Networks (CNNs). CNNs can learn hierarchical representations of image features and can be used for both classification and segmentation tasks [10]. Zhao et al [11] used a CNN-based approach which was able to detect defects with more than 92 % accuracy.

The multibox single-shot detector (SSD), a one-stage detector based on a convolutional neural network (CNN), has shown good detection performance in object detection. Liu et al. [16] improved the fabric defect detection algorithm and obtained rational and efficient results. In addition, Ouyang et al. [17] presented a CNN-based framework fabric defect verification algorithm that uses a dynamic activation layer and a torque potential function to solve the classification of unbalanced data and achieves good results.

Deep learning techniques have been utilized by researchers to improve textile product quality and production efficiency through fabric defect detection with satisfying results by Wu et al [18]. However, there are still challenges in practical application, such as the demand for high real-time performance and the difficulty in obtaining sufficient defective image data for training [19]. There are two types of algorithms: one-stage and two stage detectors suggested by Wu et al. [18]. One-

Stage detectors offer high line detection speed but often have lower accuracy. Two-stage algorithms, on the other hand, have higher accuracy but slower detection speed, which may not meet real-time production needs. This also applies to material defect detection, where the choice between one-stage and two-stage algorithms depends on balancing speed and accuracy to meet specific industry needs. Therefore, the choice of algorithm should be based on actual scenarios and application requirements.

Algorithms based Deep Convolutional Neural Networks (DCNN) have achieved satisfactory results in visual tasks and are widely used in industrial scenarios. Liu et al. [20] used DCNN to detect defects in fabrics with complex textures, specifically designed for real-world textile production environments with limited resources. Zhu et al. [21] proposed an efficient DCNN architecture called Efficient Defect Detectors (EDD) that adjusts the input resolution, depth, and width with a scaling strategy to extract more low-level fabric defect detection functionality.

In recent years, Generative Adverse Networks (GANs) have gained popularity [22] in computer vision and graphics applications. They have been used to detect defects in fabrics, where they can adapt to different textures by learning from existing patterns. Liu et al.[23] used a multilevel GAN model to synthesize defect samples, and Le et al.[24] used the Wasserstein GAN in combination with transfer learning and multi-model ensemble framework. Zhao et al.[25] proposed a CNN model based on visual long-term visual memory (VLTM), while Wang et al.[26] activated the CNN self-awareness mechanism to detect fabric defects. Wang et al.[27] also combined traditional and deep learning methods for error detection.

Transfer learning is another popular approach for defect detection with minimal number of image data. In transfer learning, a set of trained features from a pre-trained CNN is the starting point for training a new model. This approach can help to reduce the amount of training data needed and improve the performance of the model. In a study by Dafu (2019) [12], transfer learning was used to detect fabric defects in real-time with an accuracy of 95%. In addition to CNN-based methods, deep reinforcement learning (DRL) has also been explored for fabric defect detection. DRL involves training an agent to interact with an environment to maximize a reward signal. Rasheed et al [13] a comprehensive review of various Computer Vision Techniques for Fabric Defect Detection.

In this project, we have developed a fabric fault detection methodology to automatically detect fabric fault using data collected from local textile factory in Bangladesh.

Scope and Deliverables:

The main objective of this project is to develop a fabric fault detection model using image processing and artificial intelligence techniques. The project explored various deep learning-based methods such as Convolutional Neural Networks (CNNs) and transfer learning using industry-standard pre-trained models. Additionally, the project also explored the use of deep reinforcement learning models for fabric defect detection.

The scope of this project includes the following tasks:

- Collecting a diverse set of fabric images with several types of defects to be used for training and testing the model.
- Preprocessing the collected images to enhance the quality of the images and extract relevant features for the training process.
- Developing a deep learning-based model, such as CNN, for fabric defect detection.
- Exploring transfer learning method using industry-standard pre-trained models, such as ResNet and Inception, to improve the accuracy and efficiency of the model.
- Evaluating the performance of the developed models based on various performance metrics such as accuracy, precision, and recall.

The deliverables of this project will include the following:

- The developed code for the fabric defect detection model along with the necessary documentation.
- A diverse dataset of fabric images with different types of defects, which can be used for further research in this area.
- A seminar to present the developed model, its performance, and its potential applications to stakeholders, including textile manufacturers, industry experts, and academic researchers.
- A journal-quality report in English and Bengali, describing the project's methodology, results, and conclusions. This article will be submitted to reputable international and national journals for publication.

Project Inception Activities

The Capita Fintech team made considerable efforts for seamless progress of the project towards achieving the project's objectives. At the beginning of the project, the team visited NACTAR headquarters in Bogura and industry partner’s facility in Narshingdi, performed intensive literature review and broken down all the activities to accomplish the best outcome of the project. The team reestablished a relationship with the industry partner and were able to collect a adequate number of defective fabric images and non-defective images from partner's facility in Narshingdi. The team has started working on the background study, literature review, and prepared the inception report and performing image labeling, augmenting, and reviewing various deep learning-based algorithms.

NACTAR Site Visit

During the first week of work, the team visited NACTAR and met with the Honorable Director-NACTAR to discuss the terms of reference (TOR) in detail. The meeting covered the project's scope, objectives, timelines, and expected outcomes. The team received valuable feedback and suggestions from the Director, which helped refine the project's TOR.

Fig 1: NACTAR and Capita Fintech Team in Bogura

Industry Partner’s Site Visit:

In the second week, we established our relationship with our industry partner to get approval to capture high-quality images of fabric defects using our resources. As part of our efforts to collect diverse and comprehensive image data for the fabric fault detection model, our team conducted a site visit to our industry partner's facility in Narshingdi.

During the visit, we were able to observe their production line and manual defect detection techniques and gain valuable insights into the textile manufacturing process. These images has been used to train and validate our computer vision models for detecting fabric faults. The site visit was a crucial step in our data collection process and allowed us to develop a deeper understanding of the needs and challenges faced by our industry partner.

Fig 2: A view of our industry partner’s facility

The team also took several additional photos during this visit, including images of the textile manufacturing processes, the fabric samples, and the equipment used for testing and inspection.

Past Works Review

We have identified relevant research articles, conference proceedings, and books that have been published in the area of fabric fault detection using deep learning techniques. We will then critically analyze and summarize the literature to identify the strengths and limitations of the existing techniques. This literature review as well as our professional experience in image processing and computer vision-based experience served as a foundation for our study and helped us to develop our solution for this project. In addition, we also reviewed our other related projects where we have used various image processing and deep learning-based techniques.

.

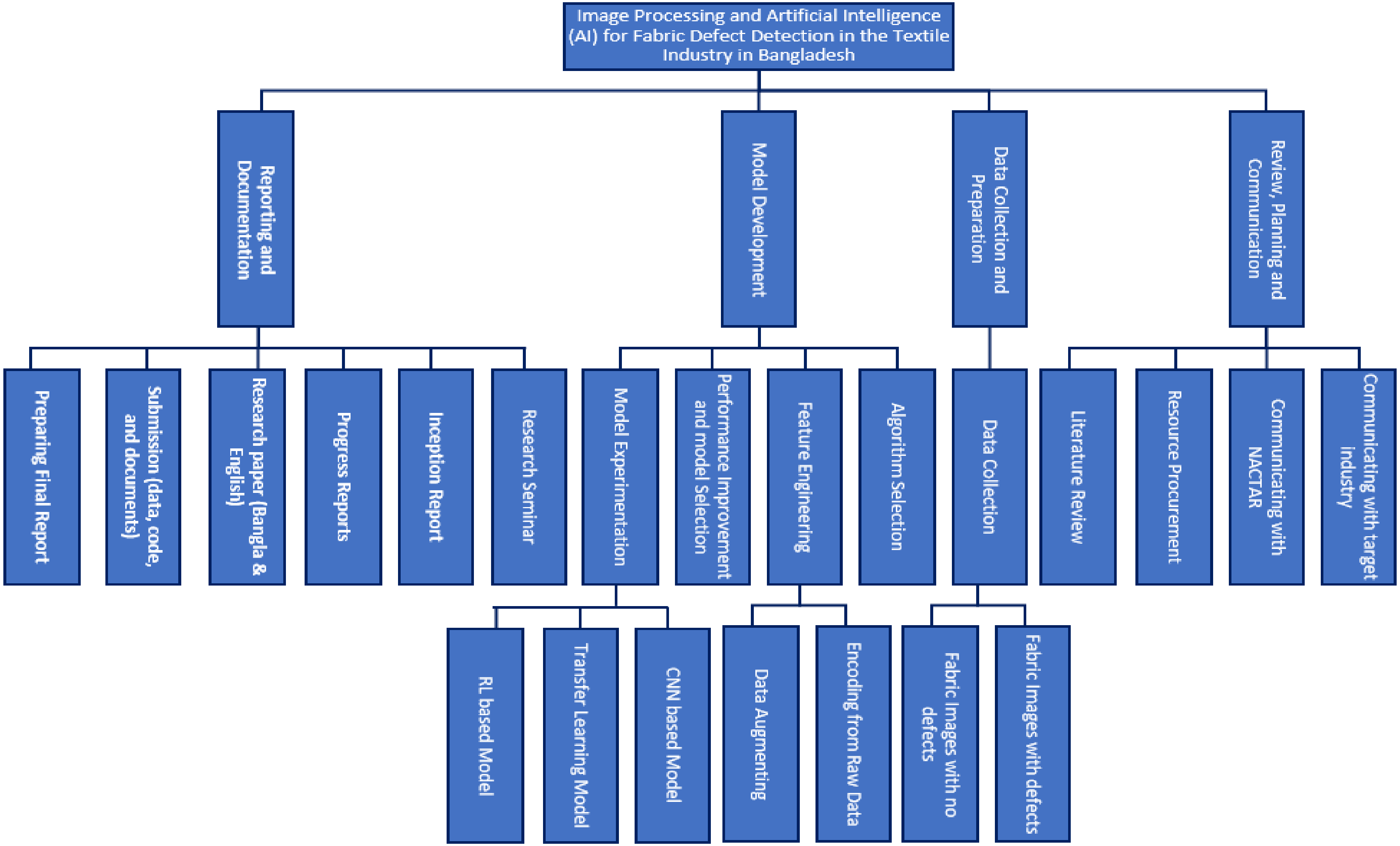

Figure 1: Work breakdown structure for accomplishing the proposed research objectives under the proposed framework

Project Work Breakdown

In order to achieve the research objectives using the proposed research methodology, we have broken down various activities into measurable and feasible components. The work breakdown structure is illustrated in Figure 2.

Methodology

The following are key components for our fabric defect detection methodology:

Data Collection and Preparation

Following the site visit to our industry partner's facility in Narshingdi, Capita Fintech team initiated data collection from the textile mills of our industry partner. With the support of our industry partner, team were able to collect an adequate number of defective fabric images and normal images. The data collection process was carefully planned to ensure that we capture a wide range of fabric defects and normal fabric samples. The collected images have been used to develop and train our computer vision models to detect and classify fabric faults accurately. Figure 2 shows Sample images collected from our industry partner’s facility.

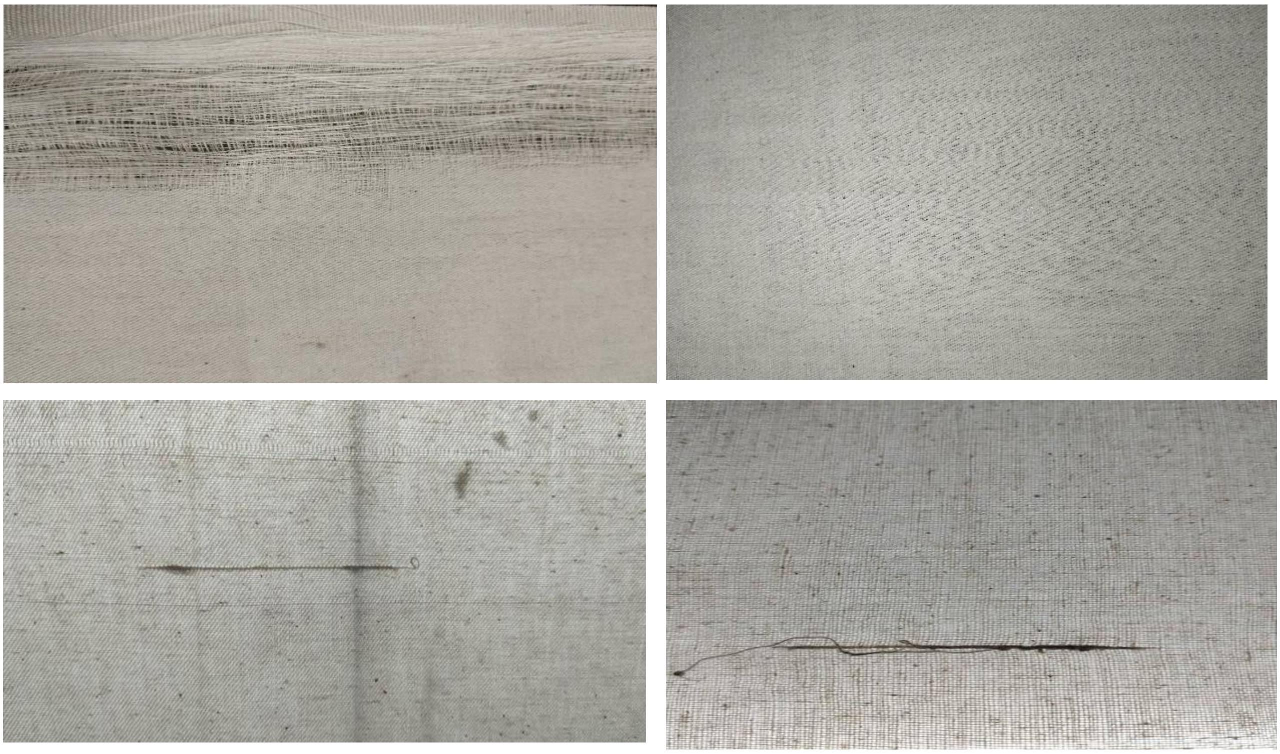

Figure 2: Sample images collected from our industry partner’s facility.

The team encountered some challenges during data collection, including issues with lighting and image quality. However, the team was able to overcome these challenges some extents by adjusting the camera settings and using image processing techniques to enhance the quality of the collected images.

Model Development:

The Capita Fintech team has investigated two types of deep learning models for fault detection- pure CNN-based models and Transfer learning models based on various pre-trained models. Details of those models used have been described below:

Pure CNN-based Models

Various pure CNN-based model architectures have been investigated to train the models using good quality data and defective quality data. This architecture was derived after experimenting with various combinations of convolutional, pooling, and dense layers with different activation functions, optimizers, and learning rates. The architecture was chosen based on its performance on the training and validation datasets, and it was observed to have a good trade-off between performance and computational complexity.

The final CNN based model architecture consists of a series of convolutional layers with increasing numbers of filters (32, 64, 128, 256, and 512), which are followed by max pooling layers that reduce the spatial dimensions of the output. The convolutional layers use a 3x3 filter and the Rectified Linear Unit (ReLU) activation function, which is commonly used in deep learning to introduce non-linearity into the model.

The model also includes two fully connected (dense) layers with 1024 and 512 units, respectively, both using the ReLU activation function and a dropout layer with 0.5 probability to prevent overfitting. The output layer has a single unit and uses the sigmoid activation function to produce a binary classification output (good or defective).

Finally, the model is compiled with the binary cross-entropy loss function and the RMSprop optimizer with a learning rate of $1e^{-4}$. This optimizer is commonly used for deep learning tasks and is well-suited for problems like this project. The best performing pure CNN-based model is saved in the specified file path for further evaluation with test dataset. Figure 3 shows the developed model architecture for the CNN based model.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_45 (Conv2D)          (None, 148, 148, 32)      896

 conv2d_46 (Conv2D)          (None, 146, 146, 32)      9248

 max_pooling2d_25 (MaxPoolin  (None, 73, 73, 32)       0
 g2D)

 conv2d_47 (Conv2D)          (None, 71, 71, 64)        18496

 conv2d_48 (Conv2D)          (None, 69, 69, 64)        36928

 max_pooling2d_26 (MaxPoolin  (None, 34, 34, 64)       0
 g2D)

 conv2d_49 (Conv2D)          (None, 32, 32, 128)       73856

 conv2d_50 (Conv2D)          (None, 30, 30, 128)       147584

 max_pooling2d_27 (MaxPoolin  (None, 15, 15, 128)      0
 g2D)

 conv2d_51 (Conv2D)          (None, 13, 13, 256)       295168

 conv2d_52 (Conv2D)          (None, 11, 11, 256)       590080

 max_pooling2d_28 (MaxPoolin  (None, 5, 5, 256)        0
 g2D)

 conv2d_53 (Conv2D)          (None, 3, 3, 512)         1180160

 max_pooling2d_29 (MaxPoolin  (None, 1, 1, 512)        0
 g2D)

 flatten_5 (Flatten)         (None, 512)               0

 dense_15 (Dense)            (None, 1024)              525312

 dropout_10 (Dropout)        (None, 1024)              0

 dense_16 (Dense)            (None, 512)               524800

 dropout_11 (Dropout)        (None, 512)               0

 dense_17 (Dense)            (None, 1)                 513

=================================================================
Total params: 3,403,041
Trainable params: 3,403,041
Non-trainable params: 0
_________________________________________________________________
```

Figure 3: CNN based model architecture for Fabric Fault Detection

Transfer Learning

As stated earlier, transfer learning with a pre-trained model involves using a model that has already been trained on a large dataset, and adapting it for a new, related task. This approach is especially useful when the new task has limited data available for training like this project, as the pre-trained model can leverage the knowledge it has already acquired to improve performance on the new task. As this project has a very limited budget, time and scope, transfer learning in the best option to train model for fabric fault detection. In this project, various pre-train models have been investigated to identify best model for Fabric Fault Detection. Figure 4 shows general research methodology along with the conceptual framework of Transfer Learning based CNN model.

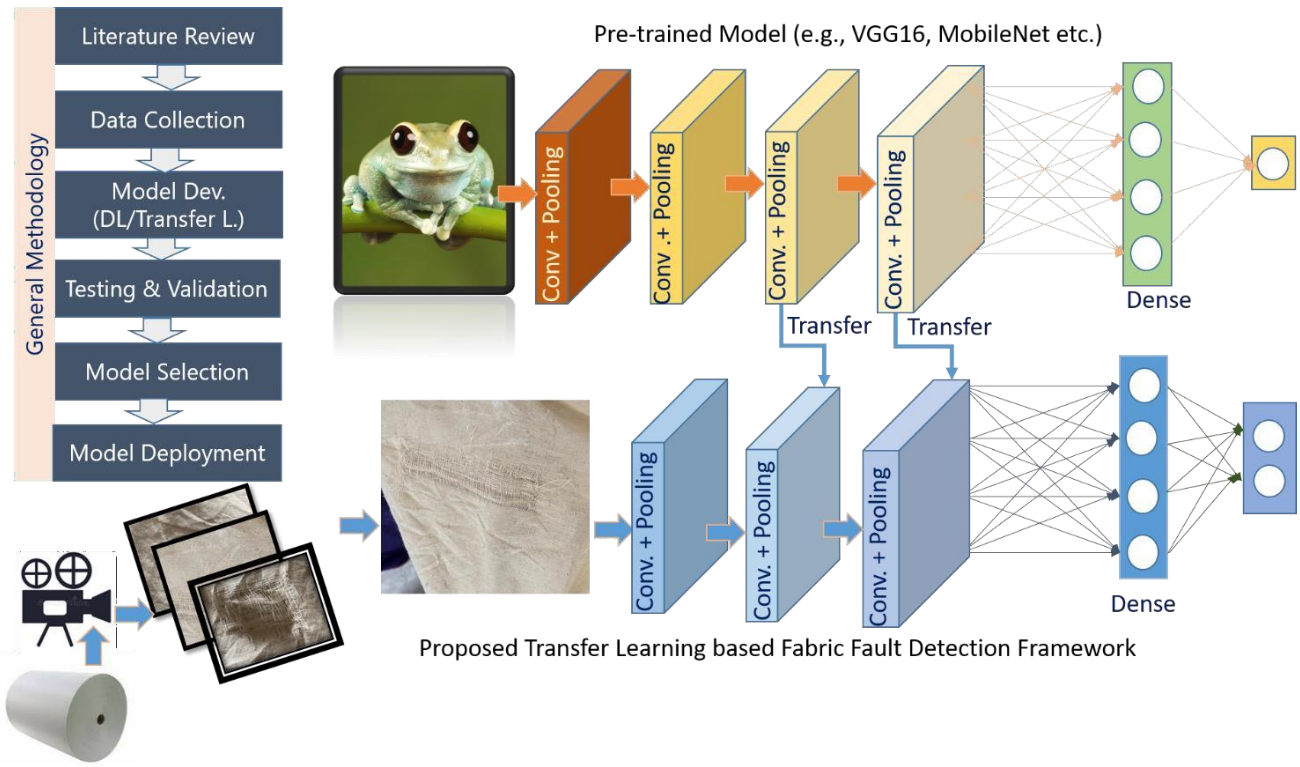

Figure 4: General research methodology and conceptual framework of Transfer Learning

Following pure CNN based model architecture, team investigated various CNN based models with extracted features with numerous pre-trained models, such as ResNet50 or VGG16 to speed up training and improve performance. The Team has investigated five pre-train models, namely, VGG16, ResNet50, Xception, DenseNet121, and MobileNet pre-trained models. Like pure CNN based method, team has performed various experiments by selecting feature from various layers of pre-training models. In all cases, team performed experiments by changing CNN architecture and various hyper parameters. Details of various pre-trained based models have been discussed in the following section.

Transfer Learning Model using VGG16:

VGG16 is a pre-trained CNN model that was developed by the Visual Geometry Group (VGG) at the University of Oxford. It is a deep learning model that has 16 layers, including 13 convolutional layers and 3 fully connected layers. The model was trained on the ImageNet dataset, which is a large collection of labeled images used for training computer vision models.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 vgg16 (Functional)          (None, 7, 7, 512)         14714688

 conv2d_13 (Conv2D)          (None, 7, 7, 512)         2359808

 conv2d_14 (Conv2D)          (None, 7, 7, 512)         2359808

 max_pooling2d_6 (MaxPooling  (None, 3, 3, 512)        0
 2D)

 conv2d_15 (Conv2D)          (None, 3, 3, 512)         2359808

 max_pooling2d_7 (MaxPooling  (None, 1, 1, 512)        0
 2D)

 flatten_4 (Flatten)         (None, 512)               0

 dense_23 (Dense)            (None, 4096)              2101248

 dropout_7 (Dropout)         (None, 4096)              0

 dense_24 (Dense)            (None, 4096)              16781312

 dropout_8 (Dropout)         (None, 4096)              0

 dense_25 (Dense)            (None, 1)                 4097

=================================================================
Total params: 40,680,769
Trainable params: 25,966,081
Non-trainable params: 14,714,688
_________________________________________________________________
```

Figure 5: Model Architecture for TVGG16 Transfer Learning Model.

For this project, a CNN model has been developed that uses a pre-trained VGG16 model as the base architecture. The VGG16 model is loaded with pre-trained weights on the ImageNet dataset and has its layers frozen to avoid changing its weights during training. The CNN then adds additional convolutional and pooling layers, followed by dense layers for classification. The convolutional layers use a rectified linear unit (ReLU) activation function and are followed by max pooling layers to downsample the feature maps. The dense layers have ReLU activation functions and are followed by dropout layers to prevent overfitting. The output layer uses a sigmoid activation function to produce a binary classification result. The model is compiled using binary crossentropy loss and the RMSprop optimizer with a low learning rate. The best performing model based on validation accuracy is saved using the ModelCheckpoint callback. Finally, the model is trained using fitting with the training and validation data generators, with callbacks set up for early stopping and checkpointing. Figure 5 shows proposed architecture of transfer learning model based on VGG16 pre-trained model.

Transfer Learning Model based on ResNet50:

This CNN model is based on the ResNet50 architecture. ResNet50 is a convolutional neural network architecture that was introduced by Microsoft researchers in 2015. ResNet50 was pre-trained on the ImageNet dataset, which consists of millions of images and thousands of object categories. This pre-training allows the network to learn features that are useful for a wide range of computer vision tasks, even if the target task has a different distribution of images and objects. The ResNet50 model is frozen, and new convolutional and pooling layers are added to the model. Then, dense layers are added, and the final output layer is a single neuron with a sigmoid activation function. The model is compiled with binary cross-entropy loss, the RMSprop optimizer with a learning rate of 2e-5, and accuracy as the evaluation metric. Figure 6 shows the full architecture of the ResNet50 based CNN model.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50 (Functional)       (None, 7, 7, 2048)        23587712

 conv2d_16 (Conv2D)          (None, 7, 7, 512)         9437696

 conv2d_17 (Conv2D)          (None, 7, 7, 512)         2359808

 max_pooling2d_8 (MaxPooling  (None, 3, 3, 512)        0
 2D)

 conv2d_18 (Conv2D)          (None, 3, 3, 512)         2359808

 max_pooling2d_9 (MaxPooling  (None, 1, 1, 512)        0
 2D)

 flatten_5 (Flatten)         (None, 512)               0

 dense_26 (Dense)            (None, 4096)              2101248

 dropout_9 (Dropout)         (None, 4096)              0

 dense_27 (Dense)            (None, 4096)              16781312

 dropout_10 (Dropout)        (None, 4096)              0

 dense_28 (Dense)            (None, 1)                 4097

=================================================================
Total params: 56,631,681
Trainable params: 33,043,969
Non-trainable params: 23,587,712
_________________________________________________________________
```

Figure 6: Model Architecture for ResNet50 Transfer Learning M

Transfer Learning Model based on MobileNet:

MobileNet is a family of neural network architectures that are designed to be computationally efficient and run on mobile and embedded devices. It achieves this by using depth-wise separable convolutions and other optimizations to reduce the number of parameters and operations required. The architecture was specifically designed for mobile and embedded vision applications. It has a small memory footprint and can run on devices with limited computational resources. The model was pre-trained on the large-scale ImageNet dataset, which contains millions of images from a thousand different classes.

In this project, a MobileNet based CNN model has been proposed. The first step in this model development process is to load the pre-trained MobileNet model and freeze its layers. This means that the weights of the pre-trained layers are not updated during training, allowing us to keep the learned features from the ImageNet dataset. We then add a custom head to the model, consisting of a global average pooling layer, followed by a fully connected layer with 1024 units and ReLU activation, and a final output layer with a sigmoid activation function. The global average pooling layer reduces the spatial dimensions of the feature maps to a single value per channel, which helps to reduce the number of parameters in the model and prevent overfitting.

We then compiled the model using the Adam optimizer with a learning rate of 0.0001 and binary cross-entropy loss. The model is then trained on the dataset using the fit method, with a batch size of 5 and 50 epochs. The ModelCheckpoint callback method has been used to save the best-performing model based on validation accuracy. The length of this architecture is too big fit this report. Therefore, it has not been added here. However, anyone can generate this architecture just using model.summay() using the code come along with is report.

Transfer Learning Model based on DenseNet:

DenseNet is a neural network architecture in which each layer is connected to every other layer in a feedforward fashion. This dense connectivity pattern allows for the propagation of gradients through the network more easily, which can lead to faster training and better performance.

The architecture proposed here is based on the DenseNet121 model, which is a CNN model that has shown strong performance in image classification tasks. The model is pretrained on the ImageNet dataset, which contains millions of labeled images, and the weights are loaded into the model using the weights='imagenet' parameter.

In this implementation, the top layers of the DenseNet121 model are removed by setting include_top=False, and the remaining layers are frozen to prevent them from being updated during training. A global spatial average pooling layer is added to reduce the dimensionality of the output from the convolutional layers. Three fully connected layers are added to the model, with 1024, 1024, and 512 nodes respectively, and an activation function of ReLU. These layers serve as the feature extraction and abstraction layers of the model, and are used to learn higher level features from the lower level features extracted by the convolutional layers.

Finally, a binary classification layer with a sigmoid activation function is added to the model to output a probability of the input image belonging to the positive class (in this case, the presence of a defect). The model is compiled using the RMSprop optimizer with a learning rate of 2e-5 and a binary cross-entropy loss function. Like MobileNet model, the length of this architecture is too big to fit in this report. Therefore, it has not been added here. However, anyone can generate this architecture just using model.summay() using the code come along with is report.

Transfer Learning Model based on Xception:

For this model, Xception model has been used as pretrain model. Xception is a deep convolutional neural network architecture that was proposed by François Chollet in 2016. It was introduced as an extension of the Inception model architecture, with the aim of improving the efficiency and performance of deep neural networks. The name "Xception" is derived from "Extreme Inception", which refers to the use of depth-wise separable convolutions to replace the standard convolutions used in the Inception model.

After loading the model, the layers are frozen, and additional layers are added, including convolutional, pooling, and dense layers. The model is compiled with binary cross-entropy loss and RMSprop optimizer with a learning rate of 2e-5. The best model based on validation accuracy is saved using the ModelCheckpoint callback, and training can be stopped early if validation accuracy stops improving using the EarlyStopping callback. The method returns a history object containing training and validation accuracy and loss values for each epoch if training is performed. Figure 7 shows the architecture of Xception based transfer learning model.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 xception (Functional)       (None, 7, 7, 2048)        20861480

 conv2d_23 (Conv2D)          (None, 7, 7, 512)         9437696

 conv2d_24 (Conv2D)          (None, 7, 7, 512)         2359808

 max_pooling2d_10 (MaxPoolin  (None, 3, 3, 512)        0
 g2D)

 conv2d_25 (Conv2D)          (None, 3, 3, 512)         2359808

 max_pooling2d_11 (MaxPoolin  (None, 1, 1, 512)        0
 g2D)

 flatten_6 (Flatten)         (None, 512)               0

 dense_29 (Dense)            (None, 4096)              2101248

 dropout_11 (Dropout)        (None, 4096)              0

 dense_30 (Dense)            (None, 4096)              16781312

 dropout_12 (Dropout)        (None, 4096)              0

 dense_31 (Dense)            (None, 1)                 4097

=================================================================
Total params: 53,905,449
Trainable params: 33,043,969
Non-trainable params: 20,861,480
_________________________________________________________________
```

Figure 7: Model Architecture for Xception Transfer Learning Model

Model Summary:

Five pre-trained CNN models, in addition to core CNN models, were examined and utilized in the creation of a fabric fault detection model. Table 1 presents parameters summary of various model developed and investigated.

Table 1: Summary of various model parameters

SL	Model	Total Parameters	Trainable Parameters	Non-Trainable Parameters
1	Pure CNN Model	11,267,361	11,267,361	0
2	Transfer Learning Model based on VGG16	40,680,769	25,966,081	14,714,688
3	Transfer Learning Model based on ResNet50	56,631,681	33,043,969	23,587,712
4	Transfer Learning Model based on MobileNet	4,279,489	1,050,625	3,228,864
5	Transfer Learning Model based on DenseNet	9,662,017	2,624,513	7,037,504
6	Transfer Learning Model based on Xception	53,905,449	33,043,969	20,861,480

Data Preparation and Model Training:

Data preparation is one of the most critical steps in machine learning, as the performance of the model directly depends on the quality of the input data. In this study, the data preparation process involves renaming files and selecting training, validation, and test datasets randomly. Renaming files with a specific suffix (i.e., 'good' or 'defect') was done to organize files with the same label, making it easier to read and work with them. Selection of training, validation, and test datasets were done such way that it ensures that the model is trained on a diverse range of samples and is capable of generalizing well to unseen data. Considering computing resources, we have used 1200 images for model training, 500 images for model validation and 250 images for model training. Finally, the data was kept in the appropriate directories to streamline the model training process, allowing for easy access to the data during training and testing.

After preparing data, six models described earlier were trained using various parameters in order to obtain the best performing model. The training process involved the use of a small batch size of 5 and 50 epochs. This was done to ensure that the models were not overwhelmed with large amounts of data and that they were trained efficiently. Throughout the training process, the models were evaluated using a validation set, and only the best-performing model was saved for future use.

In this case, the six models were trained using different combinations of parameters to determine which ones would perform the best.

Model Results

The evaluation of the six deep transfer learning-based models (Pure CNN Model and 5 pre-trained model using pre-trained models) involved the analysis of several metrics, including training and validation accuracy, area under the curve (AUC), precision, and recall. Additionally, several performance plots, such as the training and validation UC and accuracy plots, were used to provide a visual representation of the models' performance.

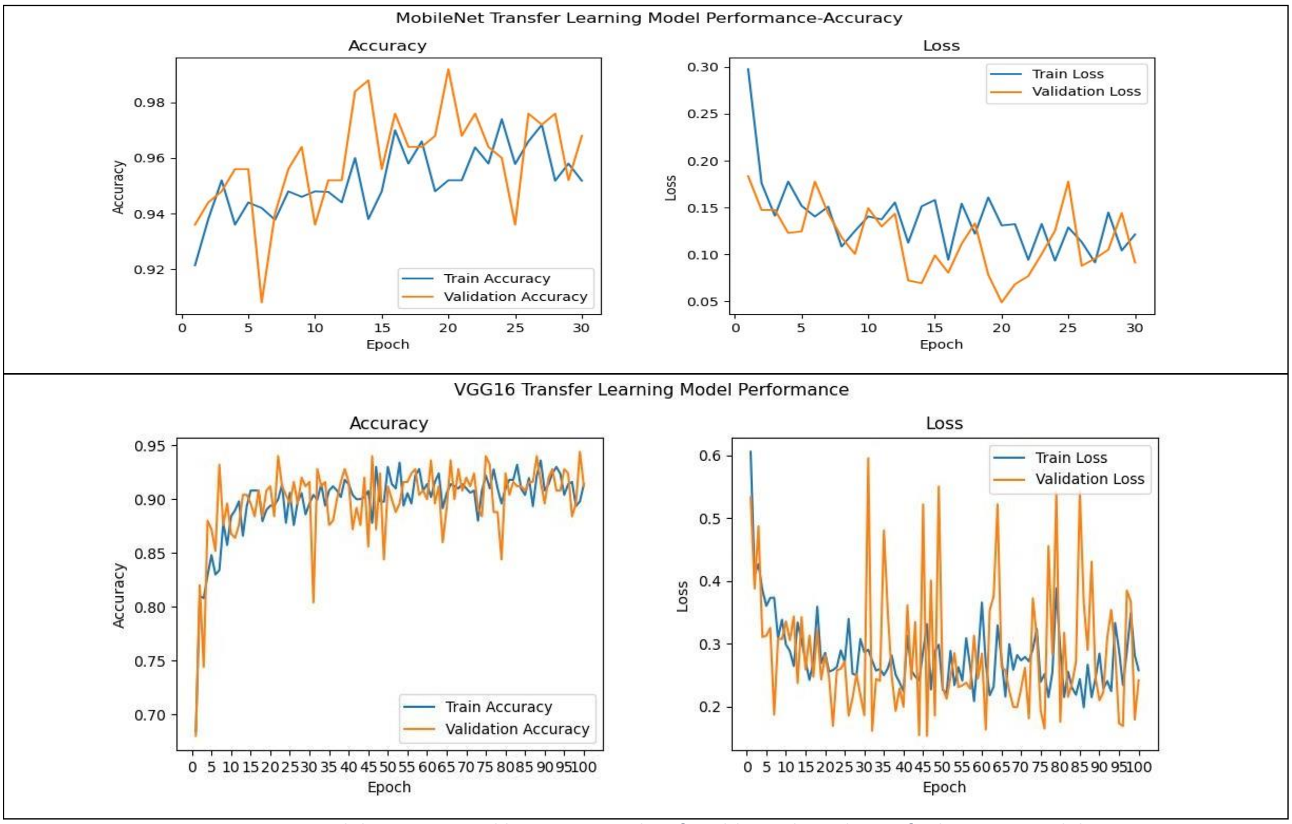

Figure 8: Model accuracy and loss over epochs of MobleNet based transfer learning model

The training and validation accuracy and loss plots provide insights into the models' training process and potential overfitting or underfitting. The plots for each model are shown in Figure 2, with the x-axis indicating the number of training epochs and the y-axis indicating the accuracy or loss value. In general, we observed that the training accuracy is higher than the validation accuracy for all models, which was an indication of potential overfitting. However, the extent of overfitting varies across models, with model ResNet50 showing the highest discrepancy between training and validation accuracy. The loss plots show a similar trend, with the training loss decreasing over time, while the validation loss reaches a minimum value and then starts to increase, indicating over fitting.

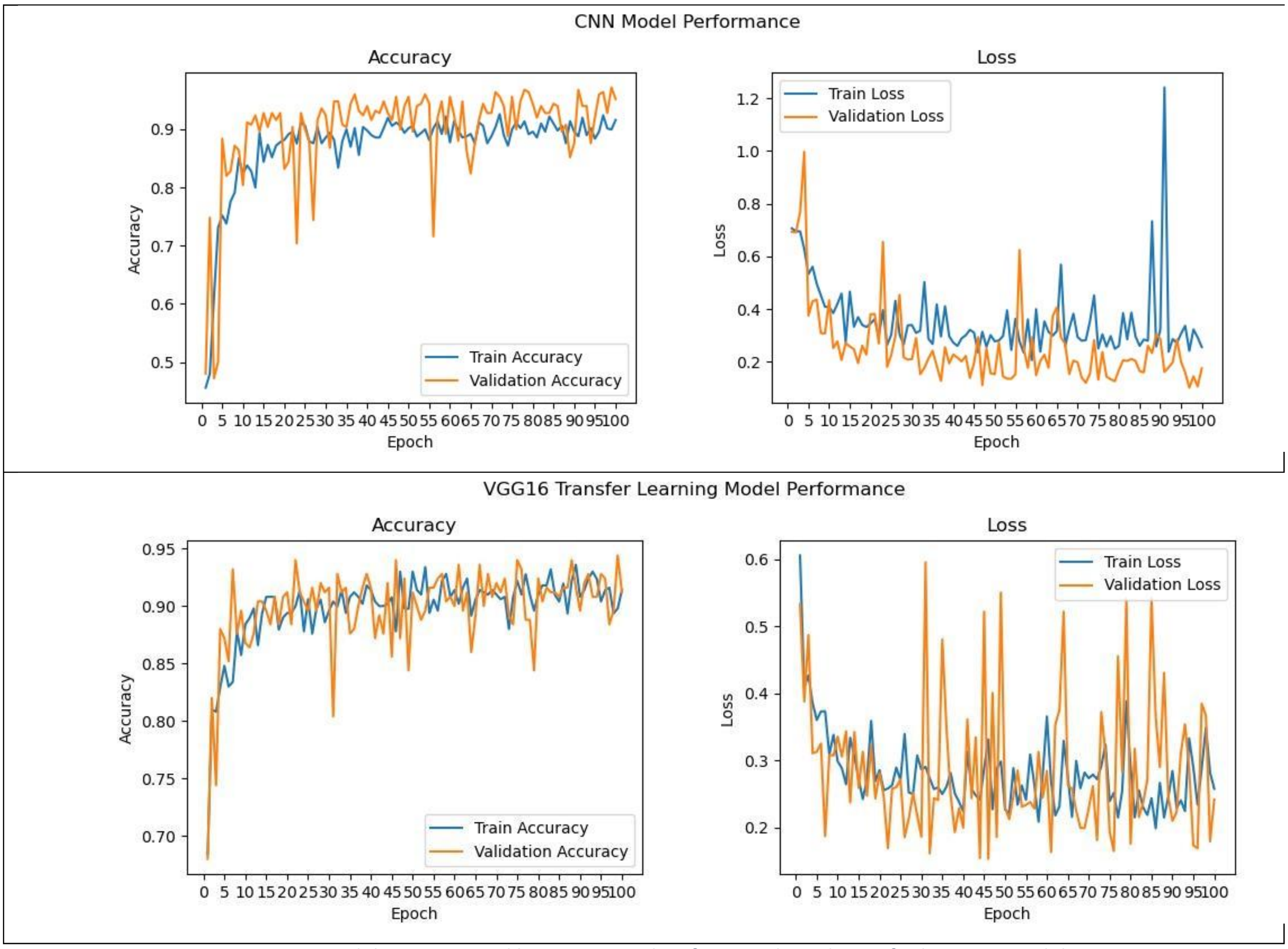

Figure 9: Model accuracy and loss over epochs of VGG16 based transfer learning model

From these plots, it is evident that the models consistently perform well in terms of training and validation accuracy and AUC. However, there were some noticeable spikes in AUC and accuracy, which may indicate that the models could benefit from additional data. Other factors, such as model complexity, optimization techniques, or even data quality, could also contribute to these fluctuations.

Overall, the evaluation of the six models suggests, it is found that the MobileNet based transfer learning model achieved the highest AUC scores among all the models, with values of 0.9886 for the training dataset and 0.9981 for the validation dataset. It also demonstrated excellent accuracy, precision, and recall values for both training and validation. With an accuracy of 0.9518 and 0.9680 on the training and validation datasets respectively, the MobileNet model shows remarkable capability in accurately identifying fabric faults. It also exhibits a high precision of 0.9502 and 0.9837 on the training and validation datasets respectively, indicating its ability to minimize false positives.

Additionally, the model achieved recall score of 0.9575 on the training dataset and 0.9528 on the validation dataset, indicating a high ability to detect true positives. Table 2 shows comparative results six models that have been trained and evaluated.

Table 2: comparative results six models

SL	Model based on	AUC		Accuracy		Precision		Recall	
		Train	Validation	Train	Validation	Train	Validation	Train	Validation
1	Pure CNN	0.9192	0.9892	0. 8695	0.9892	0.8397	0.9478	0.9053	0.9561
2	VGG16	0.9617	0.9711	0.9120	0.9200	0.8877	0.8931	0.9496	0.9512
3	ResNet50	0.4573	0.5000	0.4739	0.5280	0.4819	0.0	0.4743	0.0
4	MobileNet	0.9886	0.9981	0.9518	0.9680	0.9502	0.9837	0.9575	0.9528
5	DenseNet	9806	9851	9277	9480	9202	9231	9280	9756
6	Xception	0.9729	0.9964	0.9280	0.9720	0.9306	0.9837	0.9231	0.9603

The VGG16 based model performs well in terms of AUC, accuracy, precision, and recall. It achieved an AUC score of 0.9617 on the training dataset and 0.9711 on the validation dataset. The model's accuracy values of 0.9120 and 0.9200 on the training and validation datasets respectively indicate its capability to correctly classify fabric faults. With precision scores of 0.8877 on the training dataset and 0.8931 on the validation dataset, the VGG16 model shows a good ability to minimize false positives. The recall scores of 0.9496 and 0.9512 on the training and validation datasets respectively demonstrate its effectiveness in detecting true positives.

The ResNet50 model exhibits the lowest performance among the four models evaluated. It achieves significantly lower AUC scores of 0.4573 and 0.5000 on the training and validation datasets respectively. The model's accuracy scores of 0.4739 and 0.5280 on the training and validation datasets indicate that it struggles to accurately classify fabric faults. Moreover, the precision values for both training and validation are relatively low at 0.4819 and 0.0 respectively. The recall scores are also notably poor, with values of 0.4743 on the training dataset and 0.0 on the validation dataset.

After careful evaluation of the six models, the MobileNet model was selected as the most suitable choice for fabric fault detection. It consistently outperforms the other models in terms of AUC, accuracy, precision, and recall on both the training and validation datasets. With its high accuracy, precision, and recall scores, the MobileNet model demonstrates robust performance in accurately detecting fabric faults while minimizing false positives. It also achieves an exceptional AUC score, indicating its ability to effectively distinguish between fault and non-fault samples in the dataset. Therefore, the MobileNet model is recommended for fabric fault detection tasks.

After carefully selecting the model based on its training and validation scores, we proceeded to evaluate its performance using an independent testing dataset. The model exhibited excellent performance on the testing dataset as well, demonstrating its robustness and generalizability. Notably, the calculated AUC, Accuracy, Precision, and Recall scores were remarkable, measuring at 0.9611, 0.901, 0.901, and 0.898, respectively.

It is worth mentioning that while these scores suggest a slight indication of overfitting, it is important to consider the context in which it occurred. The testing dataset used for evaluation consisted of a relatively small sample of images. Thus, the observed overfitting is likely a result of the limited size of the test set rather than a true reflection of the model's performance.

Despite this observation, the overall performance of the model on the testing dataset remains highly promising and instills confidence in its ability to accurately classify and predict outcomes. The exceptional AUC, Accuracy, Precision, and Recall scores reinforce the model's efficacy and indicate its potential to be effectively deployed in real-world scenarios.

In addition to the evaluation of the six deep transfer learning-based models using the training, validation, and test datasets, the best model was also tested on different frames of collected video. The performance on the video frames was found to be less accurate compared to the performance on collected images. Several factors may have contributed to this lower performance on video frames, including the fact that video frames are often compressed and have lower resolution than collected images. This can result in loss of important visual information that is crucial for making accurate predictions. Additionally, the temporal aspect of video frames introduces more complexity than static images, as the model has to process a sequence of frames rather than just a single image. This can lead to challenges such as capturing the motion of objects in the video and handling occlusions and dynamic backgrounds.

Furthermore, the video frames were collected from a variety of sources, each with its own characteristics, which may have affected the model's ability to generalize to new video frames. These sources included different camera types and settings, varying lighting conditions, and different types of movements and activities captured in the videos.

These results highlight the need for training the model on a more diverse dataset with varying levels of quality and from a variety of sources to improve its performance on video frames. However, given the budget and time constraints, it may not be feasible to collect and label a large amount of additional data. Nevertheless, our company Capita Fintech is committed to improving this model by securing additional funding and resources to train it on more diverse and larger datasets, including high-quality video frames. The findings from this study provide valuable insights for future research in the domain of deep transfer learning-based models and their application in finance.

Our goal is to deploy this model in a real-world setting. Capita Fintech will conduct further research to deploy it in our industry partner’s facility in Narshingdi. However, that deployment will be out of scope of this project.

Resources and Budget

Capita Fintech leveraged a range of resources including in-house hardware and software, as well as open-source solutions and paid cloud services.

Hardware:

- High-performance computing system (CPU and GPU)
- Storage devices
- Cameras or sensors for data collection

Software:

- Deep learning frameworks (such as TensorFlow or PyTorch)
- Image processing libraries (such as OpenCV)
- Cloud computing platforms (such as AWS or Google Cloud)

Personnel:

- Data scientists and machine learning engineers for model development
- Domain experts in textile manufacturing for data collection and interpretation
- Project manager to oversee the project

Budget and Timeline:

The table below provides key milestones/ components, approximate time and approximate cost.

Table 3: Budget allocation of this project

Key Milestones/ Components	Duration	Cost (BDT)
Data collection	1 Week	10,000
Model development	3 Weeks	120,000
Business Validation	1 Week	50,000
Cloud Resources (e.g., AWS SageMaker or similar resources)	-	40,000
Publications and Reporting	2 Weeks	80,000
TOTAL	7 weeks	300,000

Stakeholders' Engagement

In this project, stakeholders played an integral role, and their engagement was critical to its success. Our stakeholders' engagement plan was well-designed, and it involved engaging with the primary stakeholder, an industry partner based in Narshingdi, Bangladesh. We collected data from them and sought their perspectives on the challenges and opportunities of implementing AI-based fabric defect detection systems. Their feedback was incorporated into the research findings, which were then shared with them for further suggestions and improvement.

The National Academy for Computer Training and Research (NACTAR) also played a crucial role in this research project. We engaged with them through work progress briefings and informed them any issues related to deadline.

We would like to express our sincere gratitude to the Honorable Director-NACTAR for his invaluable contributions to our research project. During our first week of work, we had the pleasure of meeting with the Director to discuss the project's terms of reference (TOR) in detail. His feedback and suggestions were instrumental in refining the project's scope, objectives, timelines, and expected outcomes.

We would like to extend our heartfelt thanks to the workers of our industry partner in Narshingdi, Bangladesh. Their cooperation and support in providing the data required for this research project have been invaluable. Without their participation, this project would not have been possible.

Limitations

As with any research project, there are limitations that we encountered during the course of our work. These limitations could have an impact on the generalizability and scalability of our research findings. Some of the potential limitations of our research are as follows:

Limited availability of high-quality image datasets: The availability of high-quality image datasets of fabric defects is essential to the success of our research project. However, due to our limited budget and limited access to our industry partner facility, the availability of such datasets is restricted.

Budget and Time: The approved budget and time are limited considering the potential scope of the project. Our team has managed to deliver the best possible outcome from this limited budget and time.

Future Work:

For future work, we recommend focusing on several areas to further improve the AI-based fabric defect detection system. Firstly, we suggest improving the variability of the images, including the quality of the images, the lighting conditions, and the camera angle, to make the system more robust and reliable. This can be achieved through the development of specialized imaging equipment and the implementation of controlled lighting conditions.

Secondly, image segmentation techniques can be explored to enhance the system's ability to identify and classify fabric defects accurately. This will require the acquisition of a larger dataset with more variation in fabric types, defects, and quality.

Thirdly, the mechanical equipment for fabric defect detection can be developed to enable automatic defect detection in real-time, thereby reducing the need for human intervention and increasing efficiency.

Additionally, we are interested in collaborating with NACTAR for future research projects that can explore the development of AI-based systems for applications in the textile industry or any other industry. This can include areas such as fabric quality control, production optimization, and supply chain management.

Our team has a keen interest in expanding our research efforts to AI-based technologies in municipal systems, particularly in the areas of clean technology. Our focus would be on reducing water loss and optimizing energy consumption in municipal systems such as pumping stations through the use of advanced AI algorithms and system engineering techniques. Additionally, we would like to explore the potential of incorporating green energy sources, such as solar and wind power, to improve the sustainability of these systems.

We believe that the integration of AI and clean technologies in municipal systems can have a significant positive impact on the environment, as well as on the efficiency and cost-effectiveness of these systems. We are eager to collaborate with institutions such as NACTAR and other industry partners to further our research in this field and contribute to the development of sustainable solutions for municipal systems in Bangladesh.

We are also interested in providing training on AI, Data Science, Machine Learning, and Large Language Models (e.g., ChatGPT), which are essential skills for the people of Bangladesh. We recognize the importance of staying up-to-date with the latest advancements in these technologies and are eager to share our knowledge and expertise with the team at NACTAR. Through hands-on training and workshops, we can equip NACTAR with the skills and tools necessary to stay at the forefront of AI and machine learning research and application. We believe that this will not only benefit the team at NACTAR but also contribute to the growth and development of the field as a whole in Bangladesh.

Conclusion:

The development of a fabric defect detection methodology using cutting-edge technologies like deep learning and transfer learning models presents a significant opportunity for the textile industry in Bangladesh. The manual inspection processes currently employed in the industry are inefficient, time-consuming, and prone to errors, leading to financial losses and a compromised reputation for manufacturers.

By leveraging the power of deep learning and transfer learning models, specifically the MobileNet model identified as the most effective choice, textile manufacturers can enhance their fabric defect detection capabilities. The methodology demonstrated remarkable performance metrics, including high AUC scores, accuracy, precision, and recall values on both the training and validation datasets. The MobileNet model proves to be a robust solution for accurately identifying fabric faults and minimizing false positives, thereby improving overall product quality.

Moreover, the fabric defect detection methodology aligns with the objectives of the Smart Bangladesh Mission, which seeks to leverage technology to transform the country into a digital economy. By adopting advanced technologies and automated systems, textile manufacturers can enhance their competitiveness, increase productivity, reduce inspection time, and deliver higher-quality products to customers. This, in turn, contributes to the overall economic growth of Bangladesh.

While the evaluation of the methodology on video frames highlighted certain challenges, such as lower resolution and temporal complexities, it underscores the need for further research and investment in training the model on diverse and larger datasets. By addressing these challenges and refining the methodology, textile manufacturers can extend its benefits to video-based fabric defect detection, thus expanding its scope and impact.

We understand the significance of this project for our industry partner and NACTAR and ourselves as an AI company. The Capita Fintech team was fully dedicated to the success of this project. Our team recognized that this is our first project with NACTAR, and we were determined to make it a resounding success and nailed it as planned. We are poured our best efforts and allocating our top resources to ensure that this project meets and exceeds all expectations with very limited budget.

References

[1] LightCastle, "The 4th Industrial Revolution: How It Is Changing the Ready-Made Garments Sector of Bangladesh," 2023.

[2] "Bangladesh's textile industry hit hard by pandemic – DW – 02/12/2021." https://www.dw.com/en/covid-bangladeshs-textile-industry-hit-hard-by-pandemic/a-56552114 (accessed Mar. 20, 2023).

[3] Z. Jia, Z. Shi, Z. Quan, and M. Shunqi, "Fabric defect detection based on transfer learning and improved Faster R-CNN," *J Eng Fiber Fabr*, vol. 17, Jun. 2022, doi: 10.1177/15589250221086647/ASSET/IMAGES/LARGE/10.1177_15589250221086647-FIG2.JPEG.

[4] "23 Fabric Defects to Look Out for During Fabric Inspection." https://www.intouch-quality.com/blog/5-common-fabric-defects-prevent (accessed Mar. 20, 2023).

[5] K. Srinivasan, P. H. Dastoor, F. Radhakrishnaiah, and S. Jayaraman, "FDAS: A Knowledge-based Framework for Analysis of Defects in Woven Textile Structures," *http://dx.doi.org/10.1080/00405009208631217*, vol. 83, no. 3, pp. 431–448, 2008, doi: 10.1080/00405009208631217.

[6] M. Chen *et al.*, "Improved faster R-CNN for fabric defect detection based on Gabor filter with Genetic Algorithm optimization," *Comput Ind*, vol. 134, p. 103551, Jan. 2022, doi: 10.1016/J.COMPIND.2021.103551.

[7] "Smart Bangladesh Vision 2041 - a2i." https://a2i.gov.bd/a2i-missions/smart-bangladesh-vision-2041/ (accessed Mar. 20, 2023).

[8] "Smart Bangladesh-2041 and the 4IR." https://www.albd.org/articles/news/39704/Smart-Bangladesh-2041-and-the-4IR (accessed Mar. 21, 2023).

[9] "Optimal Defect Detection with Deep Learning | Features | Vision Spectra." https://www.photonics.com/Articles/Optimal_Defect_Detection_with_Deep_Learning/a66023 (accessed Mar. 21, 2023).

[10] M. Heidari *et al.*, "HiFormer: Hierarchical Multi-scale Representations Using Transformers for Medical Image Segmentation," Jul. 2022, doi: 10.1109/WACV56688.2023.00614.

[11] S. Zhao, L. Yin, J. Zhang, J. Wang, and R. Zhong, "Real-time fabric defect detection based on multi-scale convolutional neural network," *IET Collaborative Intelligent Manufacturing*, vol. 2, no. 4, pp. 189–196, Dec. 2020, doi: 10.1049/IET-CIM.2020.0062.

[12] Y. Dafu, "Classification of Fabric Defects Based on Deep Adaptive Transfer Learning," *Proceedings - 2019 Chinese Automation Congress, CAC 2019*, pp. 5730–5733, Nov. 2019, doi: 10.1109/CAC48633.2019.8996969.

[13] A. Rasheed *et al.*, "Fabric Defect Detection Using Computer Vision Techniques: A Comprehensive Review," *Math Probl Eng*, vol. 2020, 2020, doi: 10.1155/2020/8189403.

[14] Kang, X., & Zhang, E. (2020). A Universal and Adaptive Fabric Defect Detection Algorithm Based on Sparse Dictionary Learning. *IEEE Access*, *8*. https://doi.org/10.1109/ACCESS.2020.3041849

[15] Li, P., Liang, J., Shen, X., Zhao, M., & Sui, L. (2019). Textile fabric defect detection based on low-rank representation. *Multimedia Tools and Applications*, *78*(1). https://doi.org/10.1007/s11042-017-5263-z

[16] Z. Liu, S. Liu, and C. Li, "Fabric defects detection based on SSD," in *Proceedings of the 2nd International Conference on Graphics and Signal Processing: ICGSP'18*, et al., Ed., Sydney, Australia, October 2018.

[17] W. Ouyang, B. Xu, J. Hou, and X. Yuan, "Fabric defect detection using activation layer embedded convolutional neural network," *IEEE Access*, vol. 7, pp. 70130–70140, 2019.

[18] Wu, X., Sahoo, D., & Hoi, S. C. H. (2020). Recent advances in deep learning for object detection. *Neurocomputing*, *396*. https://doi.org/10.1016/j.neucom.2020.01.085

[19] Zhou, X., Li, Y., & Liang, W. (2021). CNN-RNN Based Intelligent Recommendation for Online Medical Pre-Diagnosis Support. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *18*(3). https://doi.org/10.1109/TCBB.2020.2994780

[20] Liu, Z., Zhang, C., Li, C., Ding, S., Dong, Y., & Huang, Y. (2019). Fabric defect recognition using optimized neural networks. *Journal of Engineered Fibers and Fabrics*, *14*. https://doi.org/10.1177/1558925019897396

[21] Zhou, T., Zhang, J., Su, H., Zou, W., & Zhang, B. (2021). EDDs: A series of Efficient Defect Detectors for fabric quality inspection. *Measurement: Journal of the International Measurement Confederation*, *172*. https://doi.org/10.1016/j.measurement.2020.108885

[22] Yan, K., Chong, A., & Mo, Y. (2020). Generative adversarial network for fault detection diagnosis of chillers. *Building and Environment*, *172*. https://doi.org/10.1016/j.buildenv.2020.106698

[23] Liu, J., Wang, C., Su, H., Du, B., & Tao, D. (2020). Multistage GAN for Fabric Defect Detection. *IEEE Transactions on Image Processing*, *29*. https://doi.org/10.1109/TIP.2019.2959741

[24] Le, X., Mei, J., Zhang, H., Zhou, B., & Xi, J. (2020). A learning-based approach for surface defect detection using small image datasets. *Neurocomputing*, *408*. https://doi.org/10.1016/j.neucom.2019.09.107

[25] Zhao, Y., Hao, K., He, H., Tang, X., & Wei, B. (2020). A visual long-short-term memory based integrated CNN model for fabric defect image classification. *Neurocomputing*, *380*. https://doi.org/10.1016/j.neucom.2019.10.067

[26] Wang, J., Liu, Z., Li, C., Yang, R., & Li, B. (2020). Self-attention Deep Saliency Network for Fabric Defect Detection. *Communications in Computer and Information Science*, *1160 CCIS*. https://doi.org/10.1007/978-981-15-3415-7_53

[27] Wang, J., Li, C., Liu, Z., Dong, Y., & Huang, Y. (2019). Combing deep and handcrafted features for NTV-NRPCA based fabric defect detection. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11859 LNCS*. https://doi.org/10.1007/978-3-030-31726-3_41

About Capita Fintech

Capita Fintech (Powered by Bengal AI) is a dynamic and innovative data-driven service and solution provider to various industries. We specialize in leveraging cutting-edge AI/ML techniques and innovative engineering to create solutions that are efficient, scalable, and cost-effective. We believe that our experience, expertise, and commitment to excellence make us the ideal partner for credit scoring model development project for financial institutions. The Capita Fintech is powered by Bengal AI.

Our team comprises highly skilled and experienced professionals who are committed to delivering high-quality solutions that meet our clients' specific needs irrespective of any industries. With a focus on innovation, efficiency, and cost-effectiveness, Capita Fintech is dedicated to helping our clients achieve their goals and stay ahead of the competition.

CAPITA FINTECH