# Real-Time Fabric Defect Detection using Deep Learning


Author

ZAVEEN WAQAR

Regn # 00000206728

Supervisor

DR HASAN SAJID

DEPARTMENT OF ROBOTICS AND ARTIFICIAL INTELLIGENCE

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD


MARCH, 2021

# Real-Time Fabric Defect Detection using Deep Learning

Author

ZAVEEN WAQAR

Regn # 00000206728

A thesis submitted in partial fulfillment of the requirements for the degree of

MS Robotics and Intelligent Machines Engineering

Thesis Supervisor:

Dr. Hasan Sajid

Thesis Supervisor's

Signature:__________________________________________

DEPARTMENT OF ROBOTICS AND ARTIFICIAL INTELLIGENCE

SCHOOL OF MECHANICAL & MANUFACTURING ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY,

ISLAMABAD

JUNE, 2021

**FORM TH-4**

# National University of Sciences and Technology

**MASTER THESIS WORK**

We hereby recommend that the dissertation prepared under our supervision by: **Mr. Zaveen Waqar Regn#00000206728** Titled: **"Real-Time Fabric Defect Detection using Deep Learning"** be accepted in partial fulfillment of the requirements for the award of **MS Robotics & Intelligent Machine Engineering** degree. **(Grade_____)**

**Examination Committee Members**

1. Name: Dr. Yasar Ayaz  Signature:________________

2. Name: Dr. Jawad  Signature:________________

Supervisor's name: Dr. Hasan Sajid  Signature:________________

Date:____________________

____________________
Head of Department

____________________
Date

**COUNTERSIGNED**

Date:___________

______________
Principal

# Thesis Acceptance Certificate

It is certified that the final copy of MS Thesis written by *Zaveen Waqar* (Registration No. 00000206728), of Department of Robotics and Intelligent Machine Engineering (SMME) has been vetted by undersigned, found complete in all respects as per NUST statutes / regulations, is free from plagiarism, errors and mistakes and is accepted as a partial fulfilment for award of MS Degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in this dissertation.

Signature: ___________________
Name of Supervisor: Dr. Hasan Sajid
Date: ___________________

Signature (HOD): ___________________
Date: ___________________

Signature (Principal): ___________________
Date: ___________________

# Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

ZAVEEN WAQAR

Regn # 00000206728

Dr. Hasan Sajid

(Supervisor)

# Declaration

I certify that this research work titled “*Real-Time Fabric Defect Detection using Deep Learning”* is my own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources it has been properly acknowledged / referred.

ZAVEEN WAQAR

Reg# 00000206728

# Copyright Statement

**Acknowledgements**

*Dedicated to my exceptional wife and parents whose tremendous support and cooperation led me to this wonderful accomplishment.*

# Abstract

Defect detection in fabric production lines is a crucial and indispensable task to maintain quality in the textile industry. The current manual annotation scheme causes the fabric industry considerable losses. In order to address this issue a real-time detection system is proposed in this thesis. The system is based on a SOTA deep-learning detection algorithm which is optimized to achieve real-time performance. The detection architecture is trained on a self-gathered SOTA dataset from an industrial environment. Deploying the trained model on an actual real-time operating compactor resulted in 89% accuracy when evaluated.

**Key Words:** *Detection, Single-Shot, Faster-RCNN, Fabric Defects*

# Table of Contents

**FORM TH-4**.....iii
**Thesis Acceptance Certificate**.....ii
**Plagiarism Certificate (Turnitin Report)**.....iii
**Declaration**.....iv
**Copyright Statement**.....v
**Acknowledgements**.....vi
**Abstract**.....viii
**Table of Contents**.....ix
**List of Figures**.....xi
**List of Tables**.....xii
**List of Abbreviations**.....xiii
**CHAPTER 1 INTRODUCTION**.....1
1.1 Background.....1
1.2 Problem Statement.....1
1.3 Aims and Objectives.....4
1.4 Research Methodology.....4
1.5 Summary.....4
**CHAPTER 2 LITERATURE REVIEW**.....6
2.1 Non-Learning Based Techniques.....6
2.2 Deep-Learning Based Approaches.....6
2.3 Conclusion.....7
**CHAPTER 3 DATASETS**.....9
3.1 Existing Datasets.....9
3.2 Proposed Datasets.....10
3.2.1 Data Collection.....10
3.2.2 Data Annotation.....15
3.2.3 Dataset Details.....15
3.2.4 Comparison With Other Datasets.....16
**CHAPTER 4 PROPOSED ALGORITHM**.....17
4.1 Image Pre-Processing.....17
4.2 Feature Extraction.....18
4.2.1 Activation maps.....18
4.3 Detection Algorithm.....20
**CHAPTER 5 ALGORITHM PERFORMANCE EVALUATION USING BENCHMARK PROBLEMS**......23
5.1 Benchmark Problems.....23
5.2 Qualitative Performance.....23
5.3 Quantitative Performance.....25
5.4 Computational Performance.....29

**CHAPTER 6 RESULTS AND DISCUSSIONS........................................................................................30**
**CHAPTER 7 CONCLUSION AND FUTURE RECOMMENDATIONS..........................................................31**
7.1 Conclusion.....................................................................................................................................31
7.2 Future Recommendations.............................................................................................................31
**REFERENCES..............................................................................................................................................33**

# List of Figures

Figure 1.1: Industrial fabric compactor....2
Figure 1.2: Camouflage textured cloth....3
Figure 3.1: Camera mount on the textile compactor....8
Figure 3.2: G-streamer pipeline....9
Figure 3.3: Annotation button on the compactor....11
Figure 3.4: Nvidia Jetson TX2 setup....12
Figure 3.5(a): LED light....13
Figure 3.5(b): Diffused warm light....13
Figure 4.1: Workflow of proposed system....15
Figure 4.2: VGG-16....17
Figure 4.3: ResNET-50....17
Figure 4.4: Inception-V2....18
Figure 4.5: Basic architecture of faster RCNN....20
Figure 4.6: Basic architecture of SSD....21
Figure 5.1: Object detection evaluation metrics….22
Figure 5.2(a): Defect detection examples....25
Figure 5.2(b): Defect detection examples....26
Figure 5.2(c): Defect detection examples....27
Figure 5.2(d): Defect detection examples....28

# List of Tables

Table 3.1: Technical specifications of Logitech C920 1080p web camera............................................................11
Table 3.2: Specifications of Nvidia Jetson TX2 board............................................................................................12
Table 3.3: Specifications of dataset........................................................................................................................15
Table 3.4 : Training dataset specifications..............................................................................................................16
Table 3.5: Comparison of datasets..........................................................................................................................16
Table 4.1: Number of trainable parameters against different CNN's......................................................................20
Table 5.1: Values of confusion matric for the proposed algorithm for fabric defect detection of SSD................23
Table 5.2: Values of evaluation metrics of SSD......................................................................................................23
Table 5.3: Values of confusion matric for the proposed algorithm for fabric defect detection of Faster-RCNN. 24
Table 5.4: Values of evaluation metrics of Faster-RCNN-----------------------------------------------------------------24
Table 5.3: Results on unotimized algorithm............................................................................................................29
Table 5.4: Results on optimized algorithm..............................................................................................................29

# List of Abbreviations

| Abbreviation | Description |
|---|---|
| **AF** | Auto-correlation Function |
| **AR** | Auto-Regressive |
| **CM** | Occurrence Matrices |
| **CNN** | Convolution Neural Networks |
| **CPU** | Central Processing Unit |
| **FT** | Fourier Transform |
| **FPS** | Frame Per Second |
| **GB** | Giga Bytes |
| **GPU** | Graphics Processing Unit |
| **LED** | Light Emitting Diode |
| **MRF** | Markov Random Field |
| **NN** | Neural Networks |
| **RCNN** | Region based Convolution Neural Networks |
| **RGB** | Red-Green-Blue |
| **SSD** | Single Shot Multi-box Detector |
| **TILDA** | Textile Texture Database |
| **USB** | Universal Serial Bus |
| **VGG** | Visual Geometry Group |

# CHAPTER 1
# INTRODUCTION

## 1.1 Background

Production of fabric is a vital part of the textile industry where textile manufacturing is done via long industrial lines of machines with a compacter at the end. The cost and turn-around time are the key elements that define the quality of the product. Wasted material and the time invested in human actuation to identify and handle it are what directly affect the overall quality outcome. With vast amounts of fabric creation, errors on the surface are bound to occur. Some reasons for defects in fabrics include machine defects or spoils, faulty yarns and extreme stretching. An estimate of 70 different kinds of fabric defects are noted some examples being rust stains, holes, stitching errors and needle breaking [1]. A study pointed out that the defects in fabrics cause the industries losses between 45% to 65% and that faulty fabrics influence the sale of the product and results in loss of revenue for textile industries [2], therefore making it a vital issue to be dealt with. Former methods to combat the problem included manual fabric detection using human vision which eventually proved to be less efficient due to carelessness, lack of ability to notice small defects and optical illusions [3]. Consequently, automated methods using image and video processing techniques were required to reduce human labor, costs and errors and to improve efficiency and accuracy in fabric defect detection [4,5].

## 1.2 Problem Statement

The setup includes a fabric machine that has a horizontal actuation system on which the cloth moves. The inspection process has to be carried out while the fabric is in motion. The width of the belt is about 3 meters where-as the speed of the belt varies from 20 to 200 meters per minute.

Figure 1.1: Industrial fabric compactor.

The primary problem to address is the detection of all the defects in the fabric irrespective of their size and clarity. The current method that involves the use of manual labor results in an increase in faulty fabric due to high human error rate. To counter this a learning base approach is suggested which has its own issues to address.

The dilemma surrounding the matter of real-time fabric defect detection firstly involves the gathering of training data for our learning-based algorithm. The reason for manual collection is due to the fact that there is a scarcity of public textile datasets that are designed to counter this issue. The ones that exist, lack in quality as well as quantity. The procurement of a quality is a difficult task due to hard-work involved in the deployment of active sensory equipment in an industrial environment. The data gathered has to be cleared of false positives added due to the amount of people that happen to move in-front of the sensory equipment. The extraction of individual frames from the video data and the annotation of the data is also a tedious process.

Apart from the data, the areas to address also include the imbalance of the color shifts in fabric, which means that not all the colors in the fabric are of the same amount in the dataset. Another issue is the heat accumulation that can cause harm to the embedded machinery deployed.

Figure 1.2: Camouflage textured cloth.

## 1.3 Aims and Objectives

The aims and objectives can be summarized in the following points.

1. Detecting defects of fabric on the industrial lines moving at an average speed of 28 meters per minute.
2. Differentiating of features from occlusions such as a hand of the workers.
3. Differentiating texture from defect and making sure the defect is detected rather than a texture or pattern in the fabric.
4. The gathering of data for the learning-based algorithm.
5. Ensuring an accuracy greater than 85% to pass human accuracy of 79% as noted in [6].

## 1.4 Research Methodology

The research conducted can be classified as an applied research where the aim was to develop an industrial product. The mentioned approach was tried and tested in a real time industrial environment.

It was an exploratory study where a few SOTA algorithms were deployed and their performance was tested. The research can also be classified as deductive because a combination of theories was tested compare and validate. The research involves both primary data that was manually gathered as well as secondary data that the multiple theories were tested on. Both qualitative as well as quantitative results were drawn from the analysis.

## 1.5 Summary

A competent learning-based detection algorithm is suggested. In this thesis, ResNet-50, Inception-v2 and VGG-16 are compared for feature extractors where-as SSD[44] and Faster-RCNN[43] are compared as detection algorithms and later results are highlighted in the form of images and graphical representations and that is followed by the last section in which we conclude our work.

# CHAPTER 2
# LITERATURE REVIEW

## 2.1 Non-Learning Based Techniques

Over the years many automated approaches have been used to detect fabric defects. The approach based on spatial dissemination of gray values [11], namely statistical approach, used different representations such as auto-correlation function (AF) [12], occurrence matrices (CM) [13] and fractal dimension [14]. The methods involving the use of Fourier transform (FT) [15], wavelet transform (WT) [16] and Gabor transform [17] come under spectral approach taken for image detection. The model-based approach concentrated on the stochastic modelling in image processing which lead to three classes: covariance, 1D and 2D. Autoregressive (AR) [18] focused on 1D class while Markov random field (MRF) [19] was built for 2D. Another approach for image detection was structural approach which considered the texture as a composition of texture primitives [20]. Although recent rapid developments in computer vision technology have brought new methods for fabric defect detection, their limitations still hinder the effectiveness and efficiency of defect detection for the following reasons [21]:

- ❖ The current techniques typically incur high computational cost and result in false detection, which is impractical for real-time inspection.
- ❖ The empirical parameters cannot handle different types of fabrics well.
- ❖ Current methods are sensitive to defect size and shape, leading to a high level of false positives.

## 2.2 Deep Learning Based Approaches

Learning approach consisting of neural network (NN) models employ organization principles such as learning and generalization [22] making them ideal to combat shortcomings faced through other approaches as they can: learn complex non-linear input-output relations, work effectively because of different training methods, have suitable real-time performance that suits industrial application and can be utilized in weaving and knitting machines to expose and analyze errors. Convolution neural networks (CNN) are a special class of neural networks that are efficient in image classification and allocation [23,24]. Moreover, recently, CNN's have also been proven to demonstrate image semantic segmentation [25]. As data sets were mostly image based and exclusive of video streaming data type, previous learning approaches utilized to address the problem of fabric defect detection only made use of CNN's ability to classify images as either defected or not, thus, making these approaches Non-Real-time. Furthermore, if textured fabric as shown in figure 2 was presented, most non-deep learning methodologies failed to accurately differentiate between texture and defects. Another issue that surrounds the fabric defect detection domain is that of small defects in the fabric. This could easily be solved using a large input size image so that the receptive field of CNN observing the defect at the output feature map can retain the defect information in order to be detected. Taking this approach greatly increases our inference time resulting in low frame-rates and in turn causing non-real-time performance.

Young-Joo Han [26] proposed an approach of utilizing stacked convolutional denoising auto-encoders, which was trained on a self-made dataset. Convolutional Auto-Encoders mentioned in [27]. This study employed the use of un-supervised learning, to counter the need for the excess data to train supervised models. The approach of detecting defected regions using the output of the autoencoders has the leverage over learning-based approaches due to the non-requirement of labeled data. Their proposed system included two steps, the first stop was the generation of defects using gaussian noise and confirmation of defects by industry experts. The second step was the training of autoencoders to generate the output mask of the defected region. This approach suffered from a lower detection rate compared to a learning method as well as the self-generated artificial data that would eventually cause a bias between different types of defects and it would not be able to function with textured fabrics.

## 2.3 Conclusion

From all the methods mentioned above, it can be concluded that none can be applied simply in a real time environment. Non-deep learning techniques are not able to generalize the problem set and are far to in-complex to address the problem. Most of the deep learning-based solutions suggested suffer in their high computational complexity and operational time, but the accuracy of deep learning-based techniques is unmatched, which is why the suggested technique proposed is also deep learning based.

# CHAPTER 3

# Datasets

## 3.1 Existing Datasets

There are a few fabric datasets which are not primarily for defect detection but can be morphed to meet our requirements. They have been a part of research by a number of teams. There is a database available on Berkeley Computer Vision Group website. Some famous texture datasets are Brodatz [28] and VisTex [29]. Some datasets like CURet [30] and KTH-TIPS [31] have been used as well but they are not very well distinguished. University of Hong Kong presented a dataset of fabric containing patters which had 106 samples, of which 50 are free of defect and 56 are not [32]. A study showed 25 textile images using the same dataset [33], while another study produced 30 defect free samples and the same number of defected samples [34], thus the number of images is variating in different works.

There is no denying that analysis of textile texture is an important part in the research of textile, but these works have not been considered here as they do not focus entirely on textiles and fabrics. Moreover, due to these methods being old they lack images of defected samples in a lot of cases.

There is a vast availability of textile datasets, some of them are discussed ahead. A well-known textile dataset is Textile Texture Database (TILDA) [35], established by Technical University of Hamburg in 1995. Its formation took place within the framework of the working group Texture Analysis of the DFG`s (Deutsche Forschungsgemeinschaft) major research program "Automatic Visual Inspection of Technical Objects". On the whole it contains eight representative textile types and further there exist eight different classes for various kinds of textile and for every class there are 50 TIF images with 768 x 512 pixels. Moreover, 8 bit gray-level images are obtained by relocation and rotation of the textile samples. The whole database includes 3200 TIF pictures having a total size of 1.2 Giga bytes, but it is a private data set and researchers have to pay to use it. PARVIS [37], another private dataset exists which has 2 types of textile kinds with 1117 elements.

## 3.2 Proposed Dataset

### 3.2.1 Data Collection

Figure 3.1: Camera mount on the textile compactor.

As can be seen in figure 3.1, two Logitech C920 cameras were placed perpendicular to the fabric at a distance which was adequate for capturing the full width of the cloth in the adjacent frames. The decision for choosing Logitech C920 1080p web cameras, with the specifications shown in table 3.1, was its high-quality frames and a cheaper price point. The cameras, as can be seen in the above figure, were easily mounted on the light sources that illuminated the fabric. The camera operates at the native USB device class which made it easy to extract the raw video streams.

| **TECHNICAL SPECIFICATIONS** | |
|---|---|
| Max Resolution: | 1080 p/30 fps - 720p/ 30 fps |
| Focus type: | Autofocus |
| Lens type: | Glass |
| Built-in mic: | Stereo |
| Diagonal field of view (dFoV): | 78° |

Table 3.1: Technical specifications of Logitech C920 1080p web camera.

This computational board used for this operation was the Nvidia Jetson TX2 board with the specifications shown in table 3.2, which was more than capable of handling two separate streams of 1080p video at 30fps. The board was chosen specifically for its capability of managing to capture the concurrent camera streams, running them through a g-streamer pipeline for compression and running a hand detection algorithm on them. The setup can be seen in figure 3.4. The g-streamer pipeline was as such that we took the raw stream from the camera and encoded it to h264 compression and saved it to an mp4 sink. The pipeline can be seen below:

```
v412src num-buffers=50 ! queue ! x264enc !
mp4mux ! filesink location=defect_video.mp4
```

Figure 3.2: G-streamer pipeline.

| TECHNICAL SPECIFICATIONS | |
|---|---|
| GPU: | 256-core NVIDIA Pascal™ GPU architecture with 256 NVIDIA CUDA cores |
| CPU: | Dual-Core NVIDIA Denver 2 64-Bit CPU<br>Quad-Core ARM® Cortex®-A57 MP-Core |
| Memory: | 8GB 128-bit LPDDR4 Memory<br>1866 MHz - 59.7 GB/s |
| Storage: | 32GB eMMC 5.1+500GB Samsung SSD |
| Power: | 7.5W / 15W |

Table 3.2: Specifications of Nvidia Jetson TX2 board.

Initially a button was integrated in the compactor as can be seen in figure 3.3 so that the manual inspector can press the button and record a 10 second clip in which defects would've occurred in the fabric. The reason for this was to minimize the time required that would be wasted in the future searching for the defected frames as well as the minimization of the space complexity involving the need for a lot of storage of the video data no matter which compression method we use.


Figure 3.3: Annotation button on the compactor.

Later the button was removed due to the tech not pressing the button that often and less production of data, instead, a basic hand detection algorithm [38] was employed which would trigger a 10 second video recording as soon as a hand was detected in the video which streamlined the inflow of quality data at a faster pace. This was placed in between the g-streamer pipeline and would only send 10 seconds of frames to the video sink if a hand was detected.

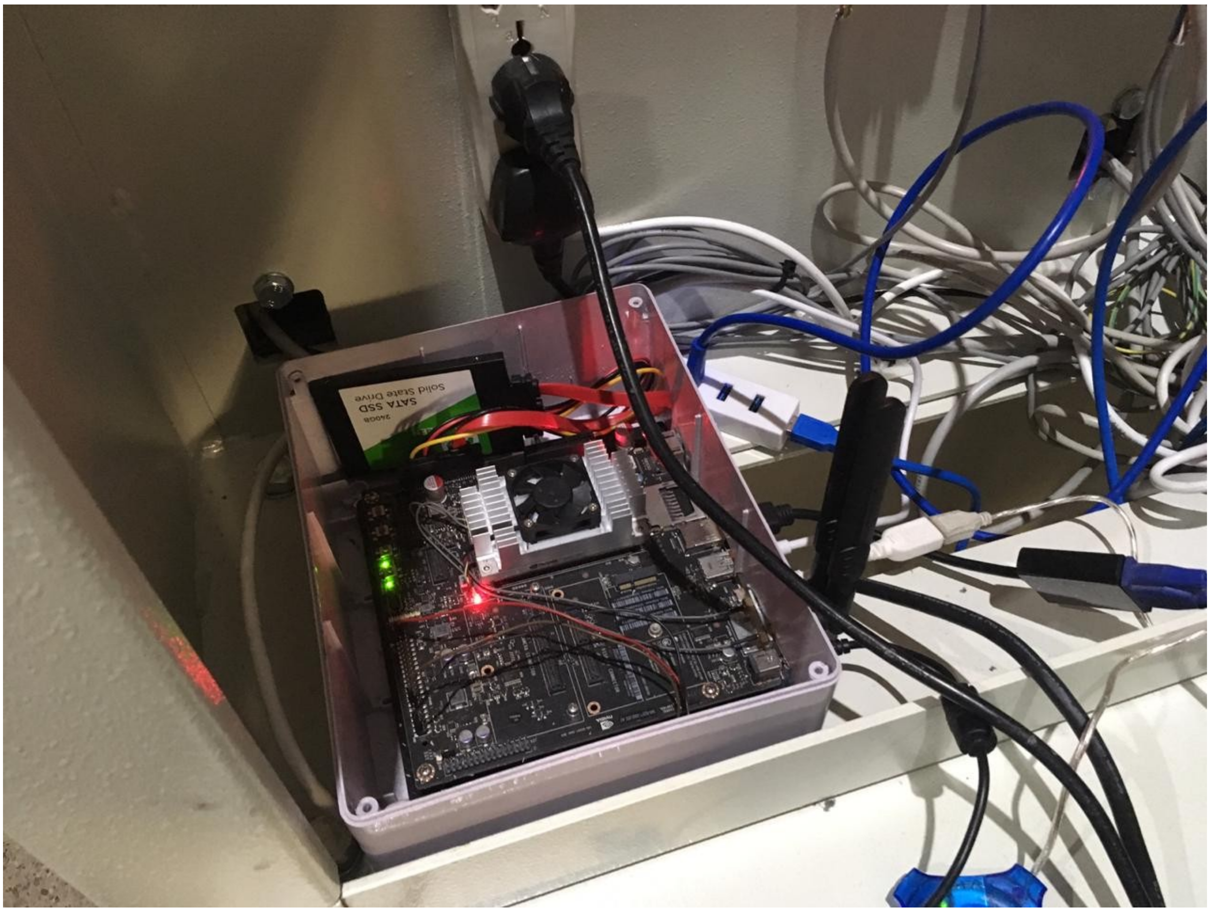


Figure 3.4: Nvidia Jetson TX2 setup.

Light was also a key area due to the factory being a closed environment with no natural lighting. We initially tried an LED light as can be seen in figure 3.5(a). This caused a tile pattern on the resulting image. We moved to diffused tube lights, figure 3.5(b), with a warm color which helped create a very clean dataset.

F

igur  e 

3.5(a): LED light. Figure 3.5(b): Diffused warm light.

### 3.2.2 Data Annotation

As the videos were recorded at 30fps per video clip, we extracted 300 frames per video clip. For the annotation of the individual frames, we used a tool called label-Img which is a well-known annotation tool used for detection algorithms that produces annotations in .XML format. A box was drawn on all the individual defects that were in a frame with a strict standard and attention towards accurate annotation as the better the annotations the better the learning algorithm is able to generalize the defects.

### 3.2.3 Dataset Details

| **DATASET SPECIFICATIONS** | |
|---|---|
| Resolution: | 1920 x 1080 |
| Color: | RGB |
| Bit Depth: | 24 |

Table 3.3: Specifications of dataset.

| TRAINING DATASET SPECIFICATIONS | | |
|---|---|---|
| Sr. | Data-Type | Data-Volume |
| 1. | Training | 6,000 |
| 2. | Validation | 1,000 |
| 3. | Test | 1,000 |

Table 3.4: Training dataset specifications.

### 3.2.4 Comparison with Other Datasets

| **Data Characteristic** | **Our Dataset** | **TILDA** | **PARVIS** | **AFID** | **[16]** | **KTH-TIPS** | **VisTex** |
|---|---|---|---|---|---|---|---|
| **Resolution** | 1920x1080 | 768 x 512 | - | 4096x256 | - | 200x200 | 256x256-512x512 |
| **Number of Images** | 33430 | 3200 | 1117 | 247 | 106 | 243 | 20 |
| **Defected Images** | 23568 | 7 | 11 | 106 | 50 | - | - |
| **Color Space** | RGB | Gray-scale | Gray-scale | Gray-scale | - | RGB | RGB |

Table 3.5: Comparison of datasets.

As can be concluded from the data presented above in tables 3.3, 3.4 and 3.5, in consideration of both a qualitative as well as quantitative analysis, our dataset is superior to other datasets available. As learning approaches have a high data requirement, it helps solve that issue. The distribution of the number of defects is a bit uneven but that can be addressed on the algorithm end.

# CHAPTER 4
# Proposed Algorithm

## 4.1 Image Pre-Processing

The need for pre-processing is essential in the input pipe-lining of any supervised algorithm. For this particular case, where the issue mainly resides with either texture or the size of the object, the goal is to make the defect apparent so that it is not missed by the detection algorithm. Increasing the contrast of the input image is a way of highlighting the defects in the fabric as those causes the defected pixels to have a greater difference in the RGB color space than the non-defected ones. An argument can be made that it also highlights the texture which causes the texture to be detected as a defect but in our study increasing the contrast by an amount contributes in better detection performance. The original image also has to be resized which is an imperative step due to the issue with the reduction of training speed as well as inference speed. For the Faster-RCNN detector we used 1000x800 where-as for SSD we used 500x500.

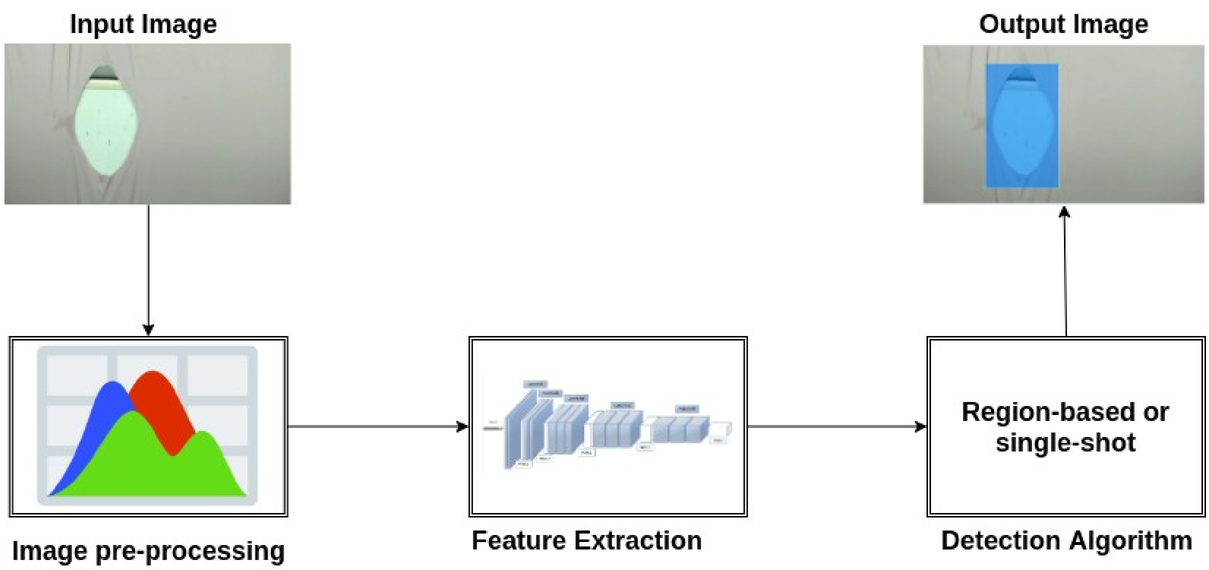



Figure: 4.1: Workflow of the proposed system.

## 4.2 Feature Extraction

Convolutional Neural Networks are the backbone of any detection algorithm as they provide the feature maps which the detection algorithm utilizes to regress to single or multiple bounding box or classes of the detected objects. There are many CNN architectures that have matured throughout the years. The trade-off component with respect to the CNN is also between time and accuracy. If the number of parameters is in abundance, the CNN will be able to generalize to a plethora of data, thus yielding high accuracy. On the down side, the increase in parameters results in the raise in time complexity. Multiple Convolution Neural Networks were considered and compared both having higher parameter count, such as VGG-16[39], as well as networks that had lesser parameters such as ResNET-50[40] and Inception-v2[41]. The goal was to choose a CNN which had the optimal receptive field in the output feature map. Moving towards our problem, fabric defects range from very large to very miniature sizes. The optimal CNN would be able to retain its receptive field till its last layers as well as have a lesser time complexity. The image chosen for comparing the different CNN architectures has a miniature defect to test the resulting receptive fields of the activation maps.

### 4.2.1 Activation Maps

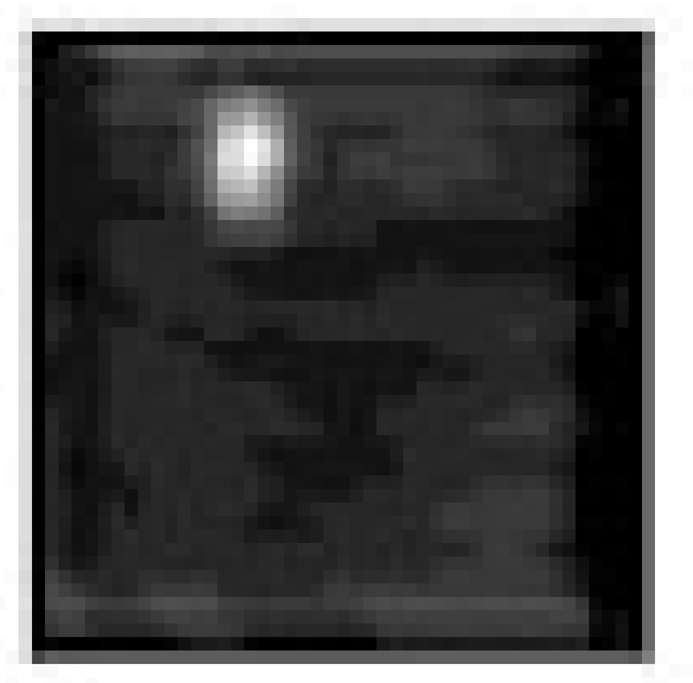


Figure 4.2: VGG-16.

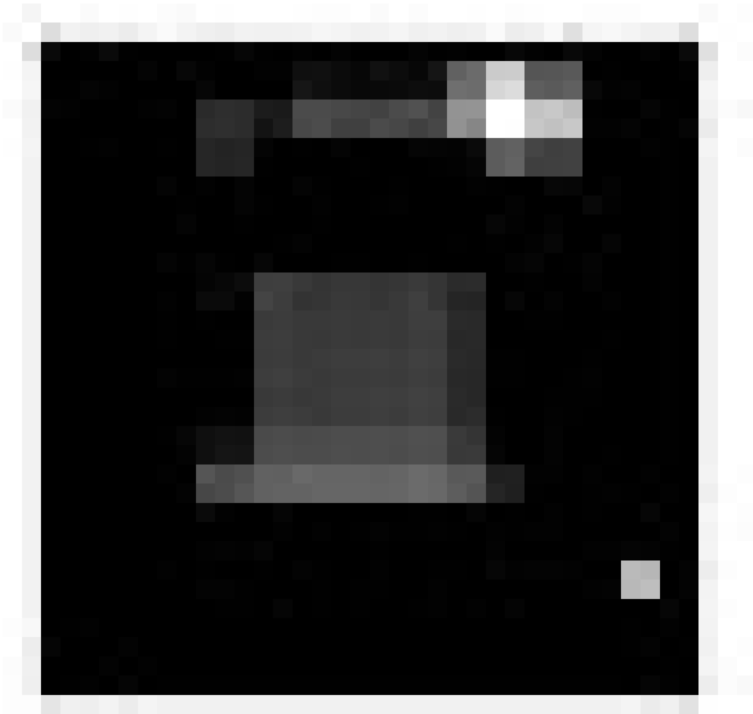


Figure 4.3: ResNET-50.

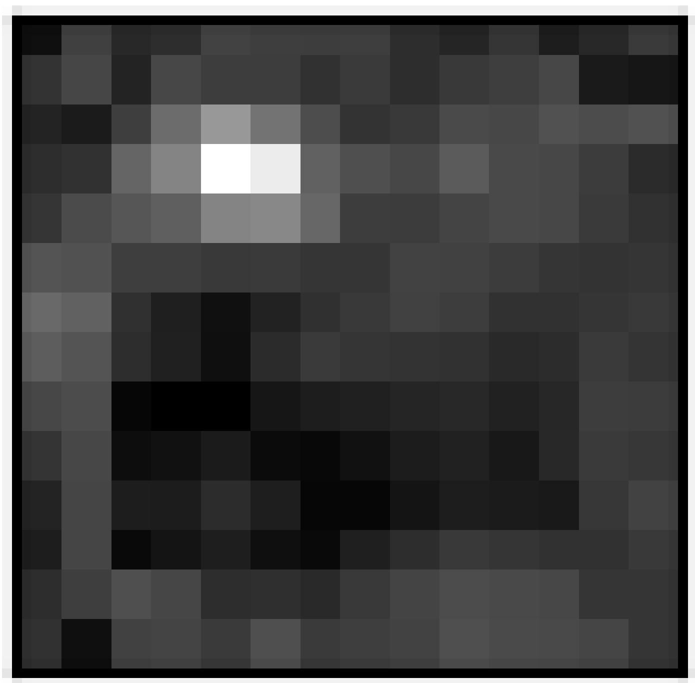


Figure 4.4: Inception-V2.

Comparing the activation maps of different CNN architectures, it can be clearly seen that VGG-16 and Inception-V2 retain the receptive field of the defect in the output feature maps, where as in the case of ResNET-50 it is difficult to identify. Also, the parameters of ResNET-50 are greater than Inception-V2, so we consider both VGG-16 and Inception-V2 as our options for experimentation.

| **CNN** | **Number of parameters** |
| --- | --- |
| VGG-16 | 138 Million |
| ResNET-50 | 26 Million |
| InceptionV2 | 23 Million |
| MobileNetV2 | 2.2 Million |

Table 4.1: Number of trainable parameters against different CNN's.

## 4.3 Detection Algorithm

The detection algorithm is another key module in our pipeline which is responsible for the actual detection of the defects. Recently, many detection algorithms have surfaced that can address the problem of detection veraciously. From all these available options two architectures were chosen for their all-rounded performance for metrics such as speed, accuracy and adaptability. The key areas to address in our problem while considering these architectures were the input size of the image as well as the need for high frames per second.

The first algorithm is Faster-RCNN[43] which uses a region proposal network to propose viable regions with a higher chance of being inclusive of objects. This detection method has a great overhead on the cost due to the RPN and the addition of the Fast-RCNN architecture at the end of the network. None the less, Faster-RCNN makes up for its disadvantage in speed, with high accuracy. It also proved to be impervious to texture in the input images, which proves that it managed to learn the features associated with a defect in a higher dimensional space.

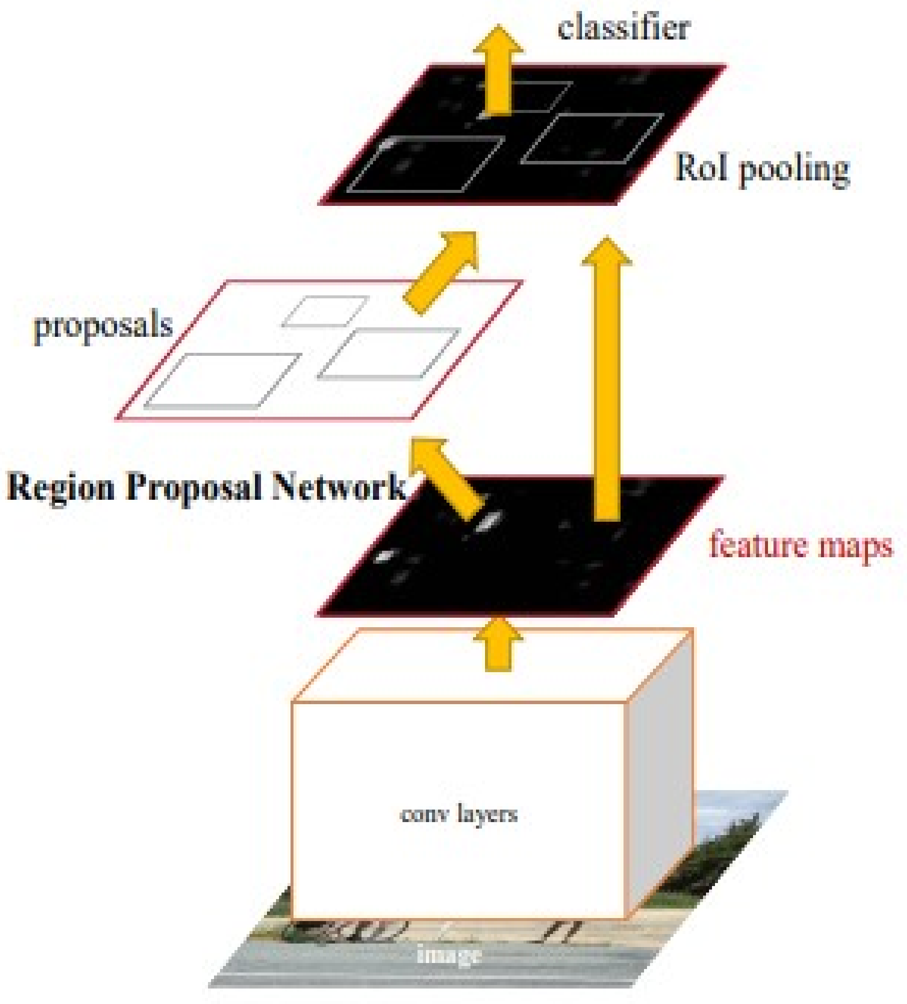
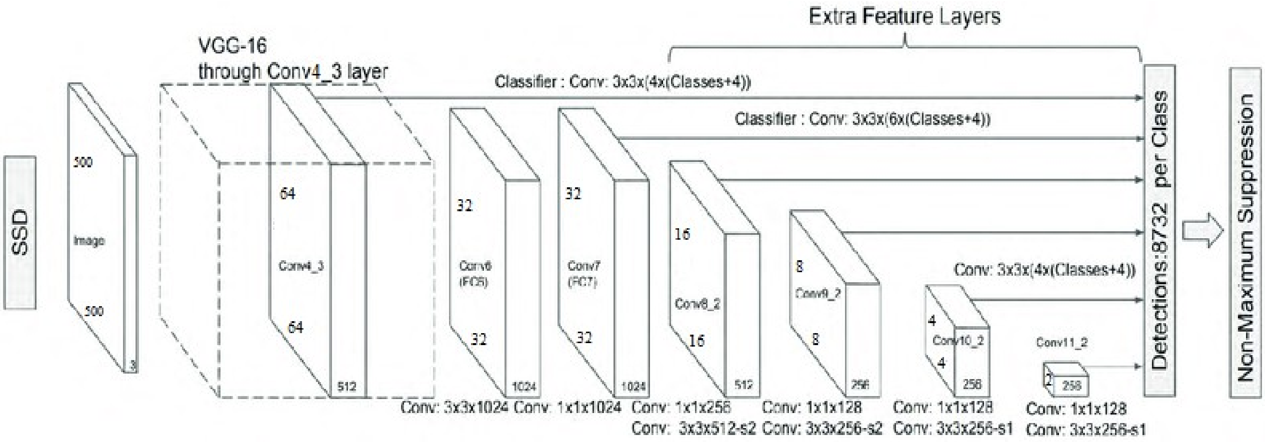



Figure 4.5: Basic architecture of Faster RCNN.

The second algorithm is SSD[44], this algorithm is currently ranked as one of the best detectors on standard data-sets such as COCO and PASCAL-VOC. By the use of multi-scale features and default boxes, SSD manages to reach the accuracy of Faster-RCNN without an RPN layer. Moreover, this greatly increases the frame-rate of SSD due to the less computational complexity of the architecture. As we used the vanilla implementation of the network it did not accept images greater than 500x500 which causes it to miss the smaller signals. The results on textured fabric still remain unaffected as the architecture is complex enough to learn the difference in features of defects compared to textures. If a simple vision algorithm had been concerned, texture would have been considered as a large defect, but SSD does possess the ability to classify texture as background in this case.

Figure 4.6: Basic architecture of SSD.




Comparing these two algorithms a clear trade-off between speed and accuracy emerges as can be seen in the upcoming section. Though, SSD does outperform Faster-RCNN in both speed and time at a lower scale, in our case, the input images were of 2MP resolution.

This results in almost 1/4th resizing of the input image to feed to the feature extraction network. This reduced size of the defects in turn makes the receptive field of the defects close to negligible in the output feature-maps, which culminates below average results for small defects. Whereas, with the 720p input image resolution of the Faster-RCNN shines in this regard and has no issue detecting smaller defects. A detailed training and pruning of the data-set could result in fruitful detections.

# CHAPTER 5

# Algorithm Performance Evaluation using Benchmark Problems

## 5.1 Benchmark Problems

A standard benchmark was considered including all the popular metrics associated with detection algorithms such as recall, precision, F-measure, accuracy etc. Though in comparison to previous detection studies, their metric of a successful detection was based on whether the frame under consideration had a frame or not, where as our algorithm localizes the defect with a degree of error as well. In this respect the accuracy metrics cannot directly be compared yet even though we localize the defect as well our resulting accuracies are still comparable as well as deployable.

$$
\begin{aligned}
precision &= \frac{TP}{TP + FP} \\
recall &= \frac{TP}{TP + FN} \\
F1 &= \frac{2 \times precision \times recall}{precision + recall} \\
accuracy &= \frac{TP + TN}{TP + FN + TN + FP} \\
specificity &= \frac{TN}{TN + FP}
\end{aligned}
$$

Figure 5.1: Object detection evaluation metrics.

## 5.2 Qualitative Performance

| | **DEFECTED IMAGE** | **DEFECT FREE IMAGE** |
|---|---|---|
| Detected as defected | TP 9672 | FP 3776 |
| z | FN 896 | TN 816 |

Table 5.1: Values of confusion matric for the proposed algorithm for fabric defect detection of SSD.

| **Sr.** | **MEASURE** | **VALUE** |
|---|---|---|
| 1 | Sensitivity/Recall | 0.915215746 |
| 2 | Precision | 0.717507418 |
| 3 | F measure | 0.804391217 |
| 4 | Specificity | 0.177700348 |
| 5 | Accuracy | **70.6900369%** |

Table 5.2: Values of evaluation metrics of SSD.

| | DEFECTED IMAGE | DEFECT FREE IMAGE |
|---|---|---|
| Detected as defected | TP<br>9876 | FP<br>2801 |
| z | FN<br>537 | TN<br>730 |

Table 5.3: Values of confusion matric for the proposed algorithm for fabric defect detection of Faster-RCNN.

| Sr. | MEASURE | VALUE |
|---|---|---|
| 1 | Sensitivity/Recall | 0.948429847 |
| 2 | Precision | 0.779048670 |
| 3 | F measure | 0.855435252 |
| 4 | Specificity | 0.206740300 |
| 5 | Accuracy | **76.0613884%** |

Table 5.4: Values of evaluation metrics of Faster-RCNN.

## 5.3 Quantitative Performance

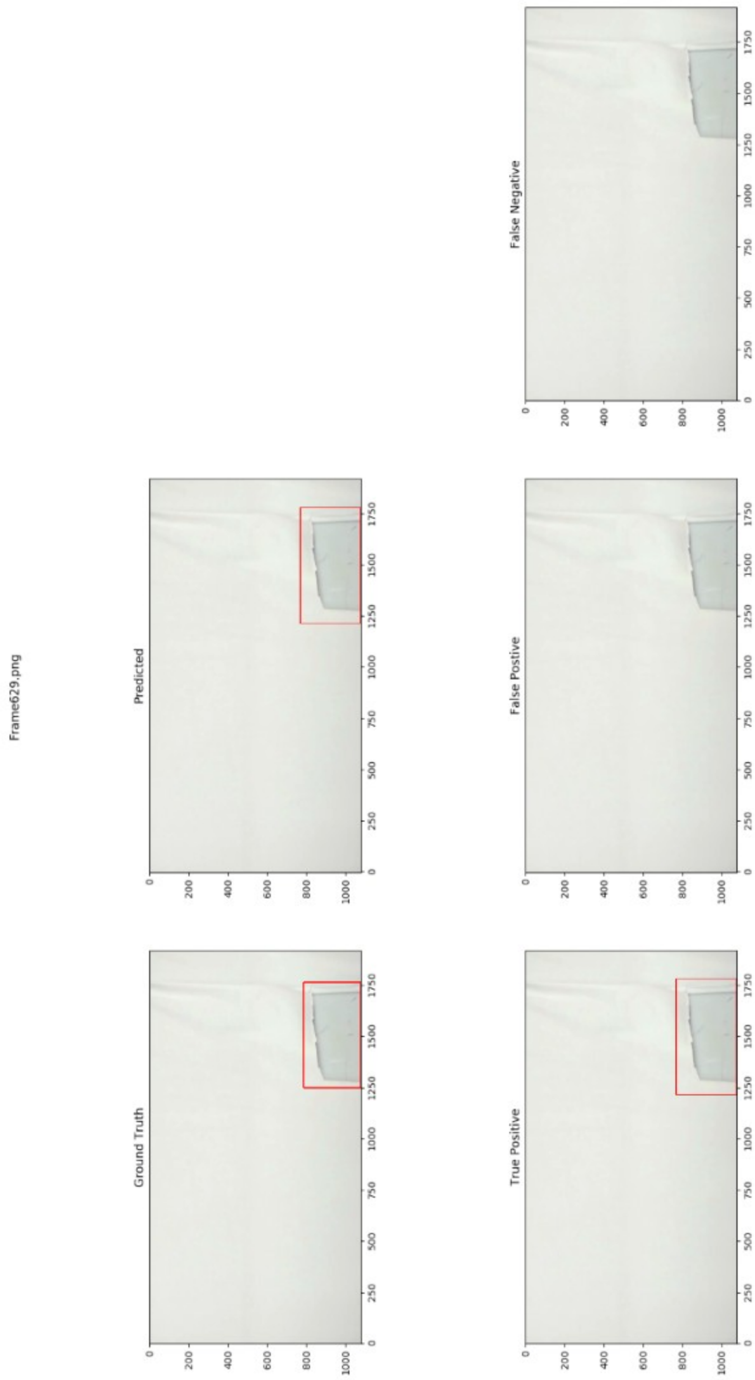
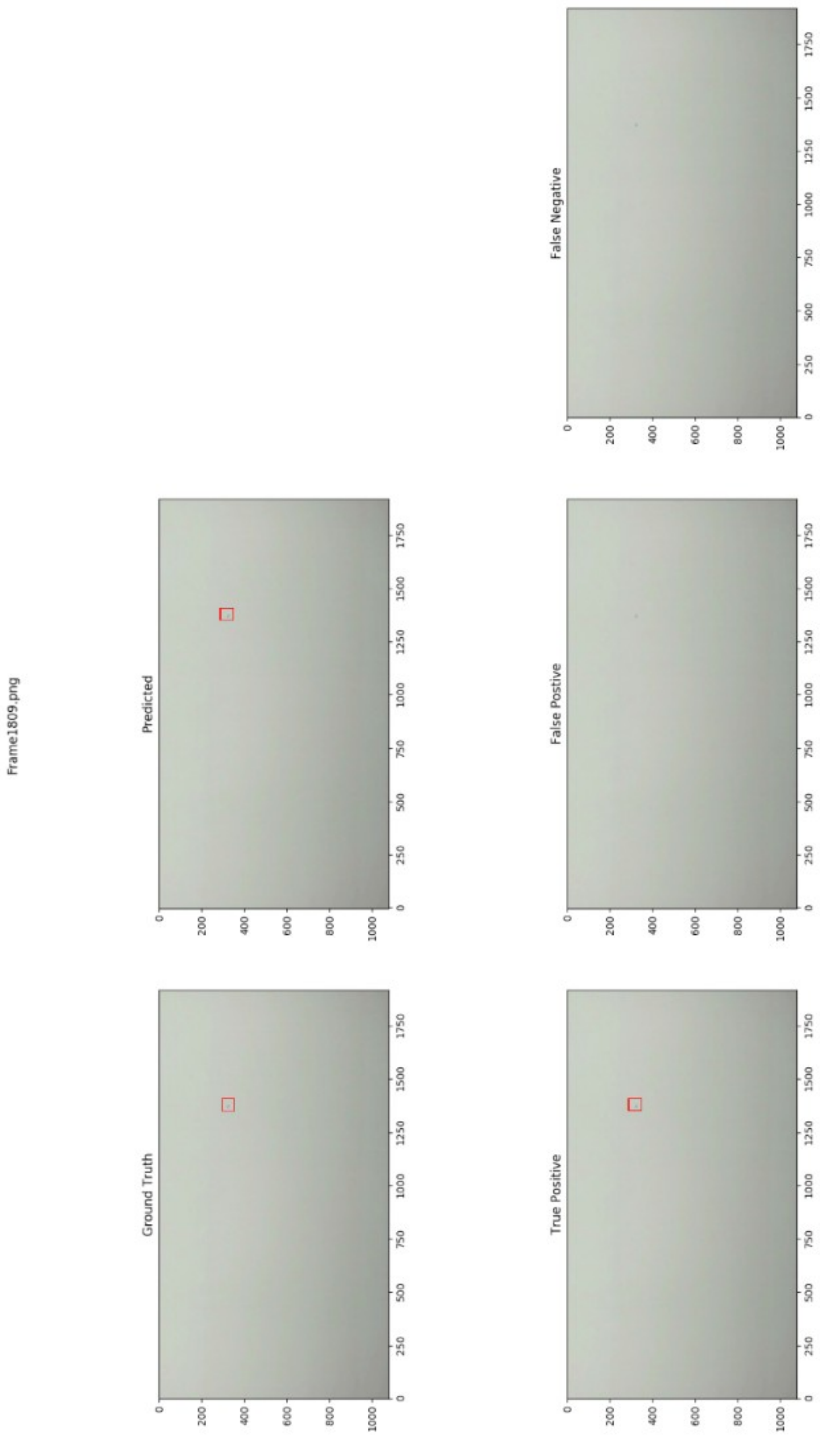



Figure 5.2(a): Defect detection examples.

Figure 5.2(b): Defect




detection examples.

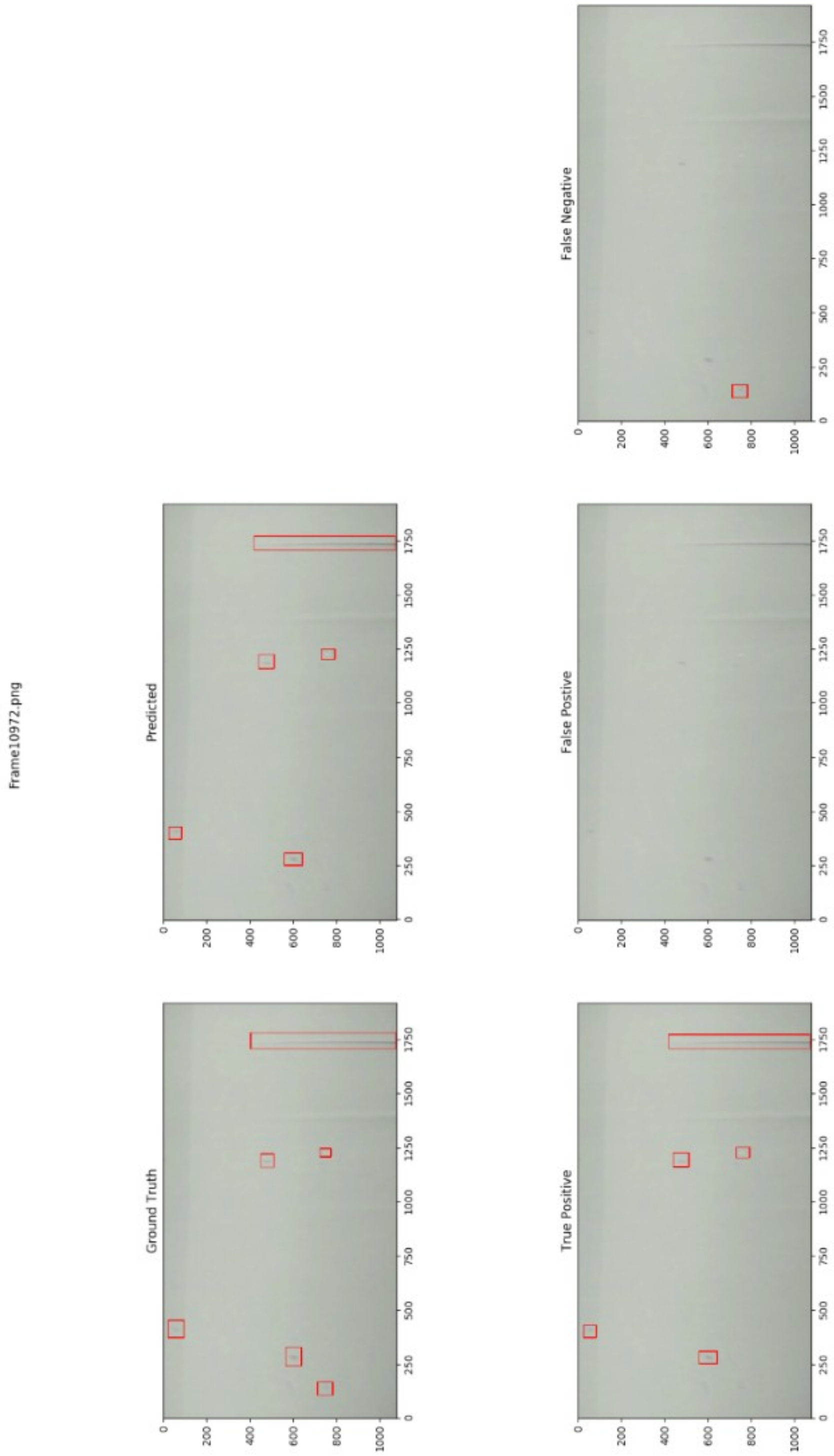



Figure 5.2(c): Defect detection examples.

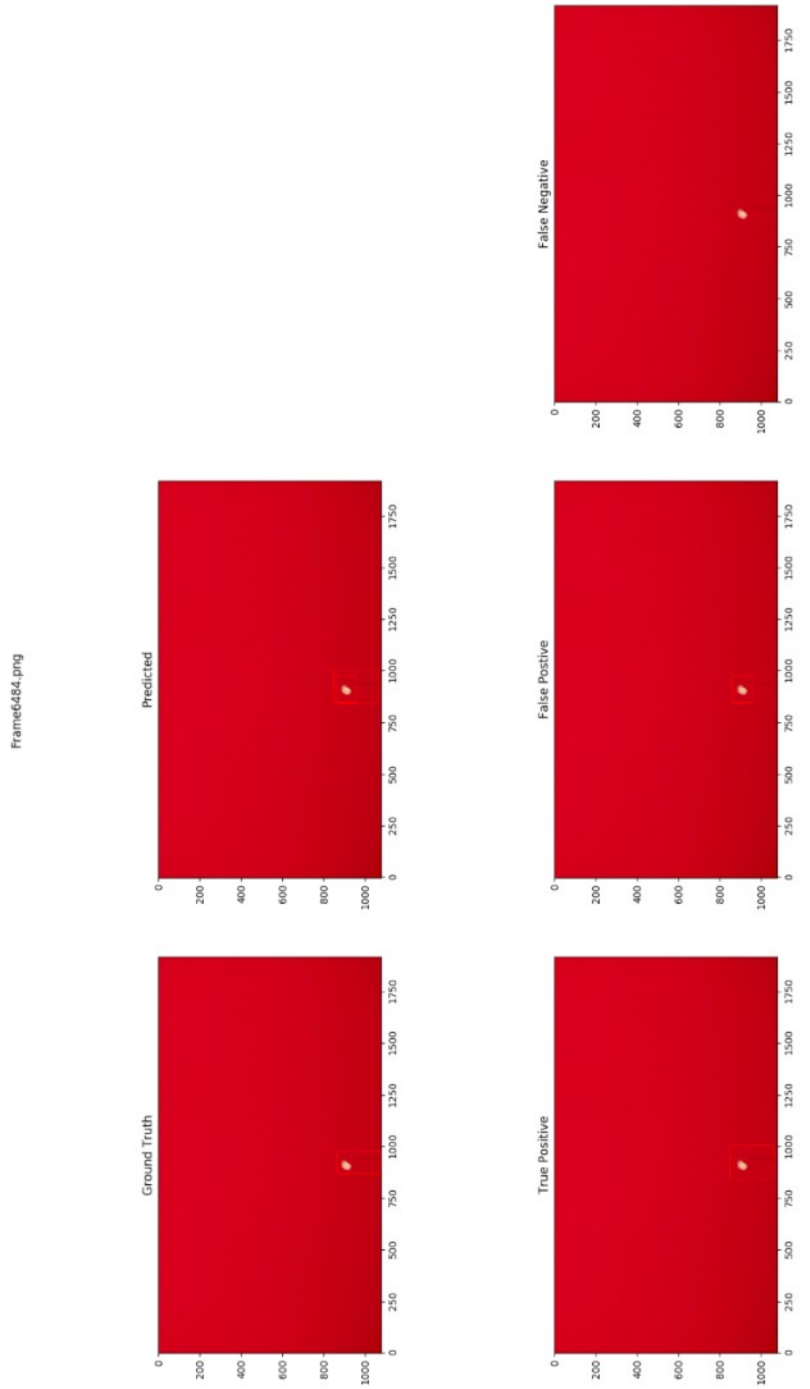



Figure 5.2(d): Defect detection examples.

## 5.4 Computational Performance

In order to increase the performance of these detection algorithms, different optimization techniques were applied to boost the frame rate. For SSD we optimized the model using Tensor-RT to float point 32 and increased the fps to 45 fps. For Faster-RCNN, Tensor-RT optimization was a bit complicated due to its complex network architecture. Alternatively, we used the graph-surgeon utility to port the Non-Maximum-Suppression (NMS) Layers in Faster-RCNN to function on the CPU instead of the GPU so that the computation could be divided and the performance could be increased. This resulted in Faster-RCNN managing to reach 25 fps which can be considered Real-Time.

| **DETECTION ALGORITHM** | **PROCESSOR** | **OS** | **GPU** | **FPS** |
|---|---|---|---|---|
| SSD | Intel Core i9 9900k | Ubuntu 16.04 | RTX-2080Ti | 22 |
| Faster-RCNN | Intel Core i9 9900k | Ubuntu 16.04 | RTX-2080Ti | 17 |

Table 5.5: Results on unoptimized algorithms.

| **DETECTION ALGORITHM** | **PROCESSOR** | **OS** | **GPU** | **FPS** |
|---|---|---|---|---|
| SSD | Intel Core i9 9900k | Ubuntu 16.04 | RTX-2080Ti | 45 |
| SSD | Nvidia Jetson TX-2 | Ubuntu 16.04 | Nvidia | 25 |
| Faster-RCNN | Intel Core i9 9900k | Ubuntu 16.04 | RTX-2080Ti | 27 |

Table 5.6: Results on optimized algorithm.

# CHAPTER 6
# Results and Discussion

## 6.1 Comparison of the compared Algorithms

Examining the results of the detection metrics of both of the proposed algorithms in tables [5.1] [5.2] [5.3] [5.4] it is clear that Faster-RCNN performs better in terms of theoretical performance. Its RPN layer enables it to detect even the smallest of defects as indicated the rate of missing a defect is low overall. Though SSD's results are satisfactory in comparison to Faster-RCNN, speed of operation is where it thrives. As can be seen in the tables [5.5] the raw performance of SSD due to its single shot network, outshines Faster-RCNN.

The reported compute device on which the algorithms were trained on are in no way suitable for the deployment of this algorithm due to its high cost as well as its high-power drain. The system size is also a key element, as there is often less space in industrial units to deploy an actual machine. For this we used a device capable of exceptional speed and power-efficiency. Also, it has to be compact enough to fit along-side the compactor electronics in order to avoid any logistic issues regarding the deployment. The Nvidia TX2 that we used for data collection is highly power efficient, extremely capable to run deep learning models and has an open-source board design to integrate with custom products. The results achieved on the TX2 board are mentioned in tables [5.4]. Faster RCNN was not optimized using Tensor-RT due which it could not be deployed on the TX2 board even though it could perform in real-time.

# CHAPTER 7
# Conclusion and Future Recommendations

## 7.1 Conclusion

In this research project, an innovative system is proposed to address the problem of Real-Time Fabric Defect Detection. The proposed algorithm is capable of detecting defects on all scales and occlusions. As it has been deployed in an industrial environment proves the system's ability to perform sturdily in any environment. Its high processing speed and high accuracy enable it to be deployed right away with very menial training required to adjust to any new type of defects that can surface in other industrial plants. Most of the datasets used in previous researches were of very less quantity over which any learning algorithm could easily generalize. Our dataset with its large quantity and high quality enables us to provide more accurate results.

In terms of the algorithms compared it is clear that Faster-RCNN gets an edge over SSD due to its larger input size which could be directly contributing to its accuracy, but in turn it makes Faster-RCNN slow in comparison. Inception-v2 proved to be a well-rounded feature extractor providing the best trade-off between speed and accuracy.

## 7.2 Future Recommendations

Some approaches to further examine include:

Experimenting with the RPN layer of Faster-RCNN introducing attention networks as well as feature fusion to speed up the process of finding the object containing regions.

Incorporating different filters over the incoming camera stream physically or through software, to utilize the combination of different frames of the same defected image to increase the input feature space.

Increasing the exposure time of the camera module to capture more clear defects as blurred images can cause misclassifications, or procuring a higher FPS and a higher resolution camera and updating the logic board which is capable of handling the high bandwidth input stream from the updated camera.

The Identification of the detected defects along with their detection to help industrial operations to find the issue in their workflow and possibly find the cause of any highly occurring defects.

Tensor-RT optimization of the Faster-RCNN inception v2 network.

# REFERENCES


[1] Handbook T. Hong Kong Productivity Council. The Hong Kong Cotton Spinners Association. 2000.

[2] Kumar A. Computer-vision-based fabric defect detection: A survey. IEEE transactions on industrial electronics. 2008 Jan 4;55(1):348-63.

[3] Anagnostopoulos C, Vergados D, Kayafas E, Loumos V, Stassinopoulos G. A computer vision approach for textile quality control. The Journal of Visualization and Computer Animation. 2001 Feb;12(1):31-44.

[4] Chan CH, Pang GK. Fabric defect detection by Fourier analysis. IEEE transactions on Industry Applications. 2000 Sep;36(5):1267-76.

[5] Ngan HY, Pang GK, Yung NH. Automated fabric defect detection—a review. Image and vision computing. 2011 Jun 1;29(7):442-58.

[6] W. Wong and J. Jiang, “Computer vision techniques for detecting fabric defects,” in Applications of Computer Vision in Fashion and Textiles, Elsevier, Amsterdam, Netherlands, 2018.

[7] Girshick R, Donahue J, Darrell T, Malik J. Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence. 2015 May 25;38(1):142-58.

[8] Girshick R. Fast r-cnn. InProceedings of the IEEE international conference on computer vision 2015 (pp. 1440-1448).

[9] Liu Z, Liu X, Li C, Li B, Wang B. Fabric defect detection based on faster R-CNN. InNinth International Conference on Graphic and Image Processing (ICGIP 2017) 2018 Apr 10 (Vol. 10615, p. 106150A). International Society for Optics and Photonics.

[10] Liu Z, Liu S, Li C, Ding S, Dong Y. Fabric defects detection based on SSD. InProceedings of the 2nd International Conference on Graphics and Signal Processing 2018 Oct 6 (pp. 74-78).

[11] Jain AK, Duin RP, Mao J. Statistical pattern recognition: A review. IEEE Transactions on pattern analysis and machine intelligence. 2000 Jan;22(1):4-37.

[12] Chen CH, Stork D. Handbook of pattern recognition & computer vision. International Journal of Neural Systems. 1994;5(3):257..

[13] Latif-Amet A, Ertüzün A, Erçil A. An efficient method for texture defect detection: sub-band domain co-occurrence matrices. Image and Vision computing. 2000 May 1;18(6-7):543-53.

[14] Conci A, Proença CB. A fractal image analysis system for fabric inspection based on a box-counting method. Computer Networks and ISDN Systems. 1998 Nov 12;30(20-21):1887-95.

[15] Castellini C, Francini F, Longobardi G, Tiribilli B, Sansoni P. On-line textile quality control using optical Fourier transforms. Optics and lasers in engineering. 1996 Jan 1;24(1):19-32.

[16] Tsai DM, Chiang CH. Automatic band selection for wavelet reconstruction in the application of defect detection. Image and Vision Computing. 2003 May 1;21(5):413-31.

[17] Y. Zhang, Z. Lu, and J. Li, “Fabric defect detection and classificationusing gabor filters and gaussian mixture model,” inAsian conferenceon computer vision, pp. 635–644, Springer, 2009.

[18] Alata O, Ramananjarasoa C. Unsupervised textured image segmentation using 2-D quarter plane autoregressive model with four prediction supports. Pattern Recognition Letters. 2005 Jun 1;26(8):1069-81.
[19] Deng H, Clausi DA. Gaussian MRF rotation-invariant features for image classification. IEEE transactions on pattern analysis and machine intelligence. 2004 May 24;26(7):951-5.
[20] Chetverikov D. Structural defects: General approach and application to textile inspection. InProceedings 15th International Conference on Pattern Recognition. ICPR-2000 2000 Sep 3 (Vol. 1, pp. 521-524). IEEE.
[21] Hanbay K, Talu MF, Özgüven ÖF. Fabric defect detection systems and methods—A systematic literature review. Optik. 2016 Dec 1;127(24):11960-73.
[22] Kumar A. Neural network based detection of local textile defects. Pattern Recognition. 2003 Jul 1;36(7):1645-59.
[23] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014 Sep 4.
[24] LeCun Y, Bengio Y, Hinton G. Deep learning. nature (2015). May; 521 (7553): 436 10.1038/nature14539. 2015.
[25] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. InProceedings of the IEEE conference on computer vision and pattern recognition 2015 (pp. 3431-3440).
[26] Han YJ, Yu HJ. Fabric Defect Detection System Using Stacked Convolutional Denoising Auto-Encoders Trained with Synthetic Defect Data. Applied Sciences. 2020 Jan;10(7):2511.
[27] Zhang, Yifei. "A Better Autoencoder for Image: Convolutional Autoencoder." (2018).
[28] University of Southern California, "Volume 1: Textures", Signal and Image Processing Institute, Ming Hsieh Department of Electrical Engineering, <http://sipi.usc.edu/database/database.php?volume=textures>.
[29] Picard R, Graczyk C, Mann S, Wachman J, Picard L, Campbell L. Vision Texture Database, Media Laboratory, Massachusetts Institute of Technology (MIT) 2009, <http://vismod.media.mit.edu/pub/VisTex/VisTex.tar.gz>.
[30] Dana KJ, Van Ginneken B, Nayar SK, Koenderink JJ. Reflectance and texture of real-world surfaces. ACM Transactions On Graphics (TOG). 1999 Jan 1;18(1):1-34.
[31] Fritz, M., Hayman, E., Caputo, B., Eklundh, J.-O., Kth-tips database, 2006, <http://www.nada.kth.se/cvap/databases/kth-tips/>.
[32] Ngan, H. Y., Pang, G. (2009). Regularity analysis for patterned texture inspection. IEEE Transactions on Automation Science and Engineering, 6(1), 131-144.
[33] Yang, X. Z., Pang, G., Yung, N. (2002). Discriminative fabric defect detection using adaptive wavelets. Optical Engineering, 41(12), 3116-3126.
[34] Ngan, H. Y., Pang, G. K., Yung, S., Ng, M. K. (2005). Wavelet based methods on patterned fabric defect detection. Pattern Recognition, 38(4), 559-576.
[35] Workgroup on Texture Analysis of DFG, TILDA textile texture database, <http://lmb.informatik.uni-freiburg.de/research/dfgtexture/tilda>.
[37] PARVIS, "PARVIS," provided by PARVIS srl, <http://www.parvis.it/>.
[38] Sajid H, Sen-ching SC. VSig: Hand-gestured signature recognition and authentication with wearable camera. In2015 IEEE International Workshop on Information Forensics and Security (WIFS) 2015 Nov 16 (pp. 1-6). IEEE.
[39] Karen Simonyan, Andrew Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, Published as a conference paper at ICLR 2015, arXiv:1409.1556.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep Residual Learning for Image Recognition, arXiv:1512.03385v1 [cs.CV] 10 Dec 2015.

[42] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, Rethinking the Inception Architecture for Computer Vision, arXiv:1512.00567v3 [cs.CV] 11 Dec 2015.

[43] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, arXiv:1506.01497v3 [cs.CV] 6 Jan 2016.

[44] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, SSD: Single Shot MultiBox Detector, arXiv:1512.02325v5 [cs.CV] 29 Dec 2016.

Zaveen Thesis

ORIGINALITY REPORT

| 7% | 6% | 2% | 4% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Higher Education Commission Pakistan<br>Student Paper | 1% |
| 2 | Submitted to Bilkent University<br>Student Paper | 1% |
| 3 | Submitted to Intercollege<br>Student Paper | 1% |
| 4 | arxiv.org<br>Internet Source | <1% |
| 5 | Huosheng Xie, Yafeng Zhang, Zesen Wu. "Fabric Defect Detection Method Combing Image Pyramid and Direction Template", IEEE Access, 2019<br>Publication | <1% |
| 6 | citeseerx.ist.psu.edu<br>Internet Source | <1% |
| 7 | www.logitech.com<br>Internet Source | <1% |
| 8 | Submitted to Cambridge Education Group<br>Student Paper | <1% |