An Advanced Machine Vision Technique for Quality Control of Fabric in the Textile Industry

by
Rifat Arman Chowdhury
21201640
Souharda Bhattacharjee
24341238
Sajib Sarker
22101767
Asim Ajwad Gani
24241128
Pulak Deb Roy
23241078

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2025

©2025 Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing a degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Rifat Arman Chowdhury
21201640

Souharda Bhattacharjee
24341238

Sajib Sarker
22101767

Asim Ajwad Gani
24241128

Pulak Deb Roy
23241078

Approval

The thesis titled **"An Advanced Machine Vision Technique for Quality Control of Fabric in the Textile Industry"** submitted by

1. Rifat Arman Chowdhury - 21201640
2. Souharda Bhattacharjee - 24341238
3. Sajib Sarker - 22101767
4. Asim Ajwad Gani - 24241128
5. Pulak Deb Roy - 23241078

Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of **B.Sc. in Computer Science** on June 20, 2025.

Examining Committee:

Supervisor: (Member)

Dr. Md. Ashraful Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator: (Member)

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department: (Chair)

Dr. Sadia Hamid Kazi
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

The textile industry in Bangladesh, one of the leading sectors of the country's economy, requires flawless product quality to ensure their worldwide reputation. However, one of the major setbacks they face is the detection of defects in fabric. Traditionally, quality control has relied on manual inspection, which is time-consuming, inefficient, and prone to human error; which incurs significant financial losses to the industry. Fabric defects introduced during manufacturing or processing make visual inspection necessary; however, the repetitive and monotonous nature of this task makes manual detection unreliable. This paper proposes a system to inspect faults on fabric in the production line of textile industries using machine vision techniques to ensure greater precision. We analyzed established deep learning based object detection models such as YOLOv8n, Single Shot MultiBox Detector (SSD) with VGG16 backbone, and Faster R-CNN with ResNet-50, and further developed a novel two-stage pipeline architecture. In our hybrid approach, YOLOv8l is employed for initial defect detection, followed by EfficientNet-B0 for classification. A custom dataset, containing a total of 9,705 images across four defect classes, was developed using pictures taken from textile factory environments to simulate real industrial settings. We also designed a prototype system for industrial automation to integrate our model into practical workflows. Fabric defect detection, automated using machine vision techniques is a growing research focus in the area, offering efficient, fast, and scalable solutions to quality control. While previous methods have been widely used, they often suffer from limitations in adaptability, accuracy, and real-world deployment. Our proposed model not only addresses these drawbacks, but also enables reliable inspection of fabric faults in industrial production. It is especially suitable for developing an economical and customizable system for fault detection. The pipeline achieved a precision of 0.842, recall of 0.819, and F1-score of 0.830—demonstrating both accuracy and industrial applicability.

Keywords:

Fabric Defect Detection; Deep learning; Computer-Vision; Textile Industry; Machine Vision; Dataset

Table of Contents

Chapter 1

Introduction

1.1 Introduction

The textile industry faces many production issues such as manufacturing equipment and yarn failures resulting in defected fabrics. Fabrics with far too many defects are disregarded which imposes economic losses for the factories.These defects reduce the overall quality of the fabric and can result in major financial losses for businesses. Manually inspecting fabric is expensive and not always effective. Even experienced workers can only detect about 60-70% defects[26]. On the other hand, automated fabric defect detection offers a better solution, lowering costs and achieving a much higher accuracy rate(approximately 90%)[27]. Bangladesh's economy comprised mainly of its textile industry ever since the 1960's. Today Bangladesh is one of the largest exporters of garments worldwide. The industry still plays a major role in Bangladesh's economy by providing more income to the ever growing population of the country. In order for this sector to remain as influential as it is, it needs to be unwavering. Despite this, the production rate is declining recently and one of the main issues behind that is quality inspection is still being carried out manually. Defect detection can be done in two different approaches, one is Process Inspection and the other is Product Inspection. Inspecting the weaving procedure in real-time is basically process inspection but often unused due to the nature of its complexity. In contrast, product inspection is widely implemented as it focuses on inspecting the manufactured fabrics[10]. Our paper focuses on product inspection by analyzing how object detection models can improve defect detection. In the textile industry, machine vision based models become an important field of study for detecting defects by extracting information from the fabric images. Various research provides an in depth understanding of machine vision techniques. Gray-scale mean values of rows and columns of an image for defect detection were analyzed by Thomas and Cattoen but it remains sensitive to the illumination changes. Another researcher named Ye, used a different approach which was efficient with translations and rotations despite performing poorly with complex textures. He used histogram driven fuzzy inference models[9]. Despite these methods showcasing strength in detecting defective from the images, they arent any good at classifying the defects. Despite other high-frequency analysis techniques such as wavelet transforms, Gabor filters, CLAHE-based detection, and Fourier transforms performs better, the computational costs are higher[15][21]. Machine vision is growing at rapid rates showcasing promising results for the future of quality control. Thus it showcases a promising future in the automation of detecting defects, a necessity as manual inspection has an accuracy of only 60%[9].

1.2 Motivation

The RMG sector, being one of the top industries of Bangladesh, leaves little room for error in quality control in order to maintain its worldwide reputation. However, it faces an imminent danger of downfall if product quality is not kept up to the mark. To control the quality, fabric defect detection is the key. Traditional methods used to this day are mostly manual, which leaves it at risk of being prone to human errors. In some parts, automated techniques are implemented, but they still face limitations due to various factors such as lighting, computation inefficiencies, etc., and it has been observed that there are still a plethora of fields for improvement. This research aims to maximize the output while minimizing the failures of detection. Automated machine vision techniques offer real-time operations that provide precise defect detection at a scalable level. The industry requires a stronger approach to achieve better efficiency and adaptability combined with increased robustness. A new comprehensive defect detection system will develop through combining advanced image preprocessing with computer vision and deep learning techniques to achieve improved accuracy, speed, and scalability of defect detection. The ultimate goal of this research is to revolutionize the quality control aspect of the RMG sector, ensuring maximum defect detection and elimination of potential financial losses,and boosting the economy of the country.

1.3 Research Problem

To prepare for the automation of fabric quality control in the textile industry, a whole variety of challenges need to be overcome. Currently, different factories have different setups for quality control of fabric, they go through 5 steps of inspection and almost all of them are different from each other. To ensure our system works in all types of industrial settings, the CNN model needs to be trained with data befitting most industrial settings. Sensitive data such as images vary in different light conditions, and since we are relying on them to inspect defects, the data should be acquired under similar lighting settings. Even shadows and wrinkles might be misinterpreted as defects, which makes securing the perfect data troublesome. Also while collecting data, proper balance of each class of data needs to be ensured to guarantee our model works best. Now this can be a challenge of its own as defects occurring on fabrics are random and each defect has a varying rate which causes an imbalance of data. There are different types of fabrics each very different from one another with varying textures, patterns, and prints. The model needs to work on all types of fabrics. For data preprocessing, techniques like CLAHE require intricate tuning as over-enhancement may produce artifacts and under-enhancement may fail to make defects more prominent. Different image processing techniques cater to different types of defects. For example, using thresholding on images with holes properly segments it but when applied to an image containing oil stains, it completely misses it rather, makes the image more noisy. Applying computer vision techniques also requires tuning of its parameters. After models are trained, applying them in real time is a challenge as latency might hamper the processing of data. In real-time applications, hardware limitations are a big issue when dealing with high-resolution data and computing complex algorithms. Both hardware and software need to be in perfect sync. All of these factors contribute to the fact that a working system might not actually work in real time. Also while collecting data, proper balance of each class of data needs to be ensured to guarantee our model works best. Now this can be a challenge of its own as defects occurring on fabrics are random and with each defect having a varying rate which causes an imbalance of data.

1.4 Research objective

The urge for an automated defect detection system is growing as manual inspection constraint is unavoidable. Only 60% of defects can be detected in manual inspection when the fabric is rolling at a speed of 30cm per second. This research intends to develop an automated system which can identify certainly 90% of the defects in a minimum of 2 square mm defect size, where the fabric is 2 meters broad and moves at a speed of 1 meter per second [9]. Data needs to be collected in factory environments to ensure the system functions in real-life situations ensuring a balanced class of dataset is obtained. The system needs to adapt to different types of fabrics too offering a general solution. Data preprocessing techniques, such as CLAHE and thresholding, will be explored and optimized for different defect types to enhance defect visibility. Different CNN models will be trained with the dataset, selecting the best performing model based on the evaluation metrics. Furthermore, the research will focus on optimizing the selected model for real-time deployment by minimizing latency and addressing hardware constraints, ensuring hassle-free integration of both software and hardware components. Ultimately, the objective is to design an efficient and scalable automated quality control system that meets the diverse requirements of the textile industry. If the implementation becomes successful, it can transform Bangladesh's textile industry by assessing production quality and reducing financial losses. Regardless of whether multiple automated models have been implemented, a more effective implementation is yet to be done.

Chapter 2

Literature Review

2.1 Background:

Machine vision technology is gaining popularity for being used in manufacturing industries in order to improve quality control and efficiency. Significant advancement has been required in this field due to the extremely high demand and the need to minimize defects. The machine vision market was valued at USD 9.3 billion in 2020 and is expected to grow to USD 16.5 billion by 2025 [18]. This expansion reflects the increasing reliance on automation in production processes. Automated inspection systems equipped with machine vision have demonstrated improvements in industrial productivity. Studies have shown that not only does automation enhance productivity by 30%, it also cuts down labor costs.[19].These systems support real-time data collection and analysis as well, contributing to a 5-6% increase in productivity [20].

Machine vision involves the use of cameras and specialized lighting systems to capture detailed images of manufacturing components. These images are processed using digitization, thresholding, segmentation, and edge detection to extract relevant information. The processed data helps in identifying product defects, ensuring that each item meets quality standards. If any defect is detected, the system can take necessary action, such as removing faulty products from the production line. Additionally, real-time feedback mechanisms allow for continuous monitoring and adjustment, ensuring that quality and efficiency are maintained throughout the manufacturing process. By integrating machine vision with automated inspection, industries can achieve greater consistency in production and reduce the margin of error in quality control [13][14].

2.2 Related Work:

Several studies on the application of machine vision for defect detection, process optimisation, and quality control in industrial settings are included in the related works. The findings illustrate the efficacy of machine vision systems and their incorporation with artificial intelligence for immediate decision-making and automation. Kumar(2004)[2] presented a variety of approaches along with their respective advantages and disadvantages in a reviewed journal. He stated that Gray-level thresholding serves as a direct method for achieving high contrast in fabric. This method finds flaws by looking for big changes in the colour of the pixels. High contrast defects lead to sudden variations in pixel intensity, producing peaks or troughs in the signal. Variations can be identified by establishing a threshold. This method enables rapid and effective inspection due to its minimal computational requirements and straightforward nature. Norton-Wayne et al. (1992)[3] eliminated random noise and focused on the clustered triggers while identifying defects travelling at speeds of one metre per second. The primary concern lies with the fabric during gray-level thresholding, as it is effective solely for high contrast materials that produce an unreliable signal. The modification proposed by Norton-Wayne[4] addressed the issue of false alarms. The random noise caused this issue, leading to the clustered triggers being addressed only after thresholding. Along with a fast adaptable thresholding limit process, they found low contrast flaws in galvanised metal strips. However, it was not effective in identifying defects in low contrast fabrics.

In another paper, Larabi, S., and Robertson, N. M.,(2020)[28] discussed the Contour detection through edge-based methods that identify image intensity and texture irregularities ties which allows mathematical models to extract boundary information using algorithms. Existing detectors such as Sobel, Prewitt, and Canny detectors function as traditional edge detection methods, which measure gradients for pixel intensity changes but remain prone to noise interference. The energy-minimizing curves of snakes and Gradient Vector Flow (GVF) snakes automatically move toward detecting object boundaries. The process of grouping and optimizing edges relies on principles of proximit,y together with continuity and closure, to create unified contour structures. The computational expenses of these methods grow higher when dealing with large edge gaps. The edge detection system HED uses deep learning algorithms to generate edge probability map,s while needing extensive training resources and data. Edge-based methods struggle to address noise sensitivity and contour fragmentation while retaining performance alongside computational complexity and domain dependency, which makes them suitable only for simple applications until they integrate with region or learning-based approaches. The mentioned techniques support applications including medical imaging. object detection, and autonomous vehicles, because they require precise contour detection.

Almedia, Moutinho and Matos-Carvalho(2021)[1] presented a new custom CNN approach for defect detection on fabric. They investigated more than 50 types of fabric defects. Meanwhile, the accuracy jumped from 75% to 95% when FN reduction is used across the CNN. The authors' proposed system elevated the accuracy level and reduced the completion time. Before using the custom CNN architecture along with FN reduction, they acquired images through CMOS (Complementary Metal-Oxide-Semiconductor) sen sor and to improve the defective area, they used histogram equilibrium in pre-processing. The process is stopped whenever a defect was found and notified the operator is notified. In the study, the new method has been tested on four fabric defect datasets, such as Fabric-Net-Dataset (FD-NT), TILDA (FR-TL), MVTec Anomaly Detection Dataset (FD-MV), and Fabric Stain Dataset (FD-ST). The accuracy has been demonstrated as compared to four available fabric detection algorithms and four various deep models of learning. As an approximate mean, the provided method carried out better than all eight approaches in every dataset. The CNN model still does not perform the absolute best on every dataset (like the FD-MV), as the new version performs very poorly under some conditions. However, the new CNN model outperforms the old model the most given that it was tested on a wider range of defect types. It is also simple in its architecture, and hence faster than deep models such as MobileNetV2. Some other methods such as Local Binary Patterns (LBP) can be equivalent with it in the efficiency of some cases but fail to generalize in other data sets. This model has a major characteristic of working with little training data, which makes it an effective algorithm in fabric defect detection and hence competing not only with other automatic means, but also with the human eye in most cases.

In their article, Gopika G Nair and Vishal Trivedi(2024)[37], gave a thorough summary of how Artificial Intelligence (AI) and Machine Learning (ML) could be tried in the quality control unit of the textile and garment industry. In the described paper, the review-based approach was used to examine the use of different AI and ML algorithms, Support Vector Machines (SVM), Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Genetic Algorithms (GA) in quality control in the textile and garment industry. They apply these techniques in defect detection, classification and prediction in yarns, fabrics, and garments. The findings discussed in the studies reviewed show a highly accurate increase in inspection accuracy with certain models showing up to 96.5% accuracy on fabric defect detection and overall across applications, an accuracy of 91.5% on average. The said findings affirm that AI and ML have a future in boosting efficiency, lowering costs, and enabling automated and quality manufacturing.

Chakraborty, Moore, and Parrillo-Chapmans(2021)[38] studied the problem of anomaly detection in printed textiles, specifically color dots and print misalignments, based on a Convolutional Neural Network (CNN). A dataset of flawed fabric images was made specifically, and CNN models were trained on different hyperparameters with the goal of a better accuracy. The researchers benchmarked their CNN model to other deep learning models, which are VGG, DenseNet, Inception, and Xception, and they found that VGG models are the most effective in detecting the defects. The findings show the possibility of automating deep learning and improving the uniformity of quality monitoring of textiles, and therefore, less market dependency on manual inspection.

The authors presented an enhanced version of YOLOv4 in their paper, " A Fabric Defect Detection Method Based on Deep Learning," to detect fabric defects more precisely[20]. Their aim was to make defects more clear, so they applied CLAHE (Contrast Limited Adaptive Histogram Equalization), which refers to the technique that helps to increase the quality of the image. They also used the K-means clustering to optimise anchor boxes, and this enables the model to make more accurate predictions of the irregular and small defects whose example is holes and stains. This problem of the original YOLOv4 was that it employed MAX-Pooling in the Spatial Pyramid Pooling (SPP) layer, which may end up losing valuable details. To address this problem, they instead replaced it with Soft-Pooling which is more intelligent and retains more information of the features by giving the pixel proportional weights rather than simply taking the greatest value. Key Improvements that were seen:

- Soft-Pooling preserves the smaller details, increasing the performance of the SPP layer.
- The model catches the defects more easily because CLAHE renders them sharper.
- K-means clustering optimizes the anchor box to have a closer localization of defects.

Integrated with these upgrades, this technique is more accurate and precise in detecting the defects in fabrics as compared to standard YOLOv4.

In the paper Shahrabadi et al.[22] put traditional machine learning methods head-to-head with modern deep learning models for spotting fabric defects. The previous methods, such as SVM and ANN depend a lot on manual feature extraction tricks, Gray-Level Co-Occurrence Matrix (GLCM) and Discrete Wavelet Transform (DWT) of extracting textures and edges of fabric images. On the other hand, studies based on deep learning (DL), particularly Convolutional Neural Networks (CNNs), shows that AlexNet and VGG16 have demonstrated to be considerably more efficient in detecting complex defects on fabrics. CNNs, unlike the typical models, independently learn useful features of raw image data, thus eliminating feature extraction by hand. AlexNet had an accuracy of 98.2%, and VGG16 had 98.1% tremendously overthrowing traditional machine learning algorithms. The paper highlights the advantages deep learning has to offer in term of defect detection like great accuracy, scalability and ability to detect and classify numerous defect types, such as stains and yarn inconsistencies. Classical analysis of traditional ML models (such as SVM and ANN based on GLCM and DWT) and the modern DL models (including CNN, AlexNet, and VGG16) has tried to assess the pros and cons of each approach. Applying DL models to detect defects on a fabric gives a significant advantage in terms of performance and flexibility in comparison with traditional techniques.

H. Li et al. (2024)[48] improved the accuracy of detecting leather defects by introducing a new method Multi-Layer Residual Convolutional Attention (MLRCA). They also demonstrated that semantic feature representation in the backbone network can be improved using MLRCA,while the new architecture called ML-FPN is the result of its integration into the feature pyramid network. Thus improving multi-scale feature fusion. They also used a Side-Aware Boundary Localization (SABL) detection head to more precisely localize and differentiate similar defect types. Their proposed model achieves strong performance with AP = 83.4, AP50 = 89.7, AP75 = 85.6 and strong results across different object sizes (APS = 71.3, APM = 89.9, APL = 88.9) with their custom experimental leather dataset. Thus we can see from the results that the model is competent enough to detect the smallest of defects.

In the journal [21], they focused on syncing the Internet of Things (IoT), cloud-based infrastructure and deep learning methods to create a framework that will be able to detect the defects in the manufacturing plant in real time. The system will implement IoT-connected cameras placed between the production line to obtain an image of the product where the image is processed in real time through edge computing. This architecture greatly reduces latency and implements immediate defect recognition offering a vital feature to support high-speed industrial operation. They also applied Convolutional Neural Networks (CNNs) in order to distinguish defective and non-defective photos. Among the reviewed models, EfficientNetB0 model performed better than others such as VGG16 and Inception v3, not only in accuracy with but also being far more efficient. It achieved a detection rate of 96.88% with low computational cost which qualifies it to be used in real time applications. The study also incorporates a component of regression models which would help in improving the quality of manufacturing. One is the K-Nearest Neighbors (K-NN) and Decision Trees, which are used to predict the ideal parameters to minimize the defect rates. The R 2 score of the Decision Tree model was very high, at 0.99, a score that showed that the model will be powerful in predicting process optimization. Such a smart machine vision system is an excellent demonstration of how the implementation of machine learning, IoT, and edge computing may vastly improve the precision and speed of the defect idetection process. Therefore, it positively adds to the overall development of the Industry 4.0 manufacturing technologies. IoT-implemented vision scene to capture in-real-time data. Image classification networks CNNs(EfficientNetB0, VGG16, Inception v3) Process prediction models(KNN and decision trees) of the simplification of the manufacturing process The methods complement each other to constitute a fully-fledged method of detecting flaw in real-time.detection and optimization of process.

- Vision system with IoT with real time process
- Image Classification using convolutional neural networks such as VGG16, EfficientNetB0 and Inception v3
- Regression Models for optimisation such as KNN's and Decision Trees

The combination of these methods provides an overall system of real-time defect detection and model tuning. The CNN models are tried out and the best one was (EfficientNetB0), however, the other components (IoT, CNN, and process prediction models). are part of the total system and not adopted separately or compared on their own.

Norton Wayne et al. made the binary and frequency models in 1992 [9] to establish a defect detection system with an aim of 95% precision emphasizing on defect size as low as 2 square millimeters. This was aimed based on the assumption that a credible inspection mechanism would increase the quality of the products and minimize wastage. The researchers improved a manual fabric inspection of tubular fabric into a machine by incorporating 2048-element line scan camera that captures photos of fabric motion at a resolution of 0.5 mm in 1-meters width . The illumination, which is a 35 kHz fluorescent lamp which was to have the lowest flicker and the most luminous, was created. The binary model makes the data simpler by transforming gray images into binary form with pixels being in either of the two classes, an abnormality or not according to a pre-set threshold. The binary filtering methods are employed to examine adjacent pixels that make up the trigger and gives a signal that will decrease the noise and picks up the detection accuracy. Such approach can help to identify lower-contrast defects through the establishment of thresholds nearer to signal mean so that accuracy in detection increases. The frequency model examines the frequency part of the image information by techniques such as Fourier transformation, which makes it possible to detect the patterns and anomalies which are not identified easily in the spatial domain. This enables it to be good in detecting periodic defects or changes in the texture of the fabric. Nevertheless, the model is complex and requires a huge processing power and complicated algorithms, so it is hard to analyze in real time. Additionally, the frequency-based techniques are prone to noise and may end up with false positives and missed likelihoods, particularly where the symptoms of defects are covered with noise. In addition, the calculation intensity poses a problem of real-time processing, which prevents fast decision-making in the high-speed manufacturing industry.

The study [39], Subjected to continuous updates and improvements along with periodic updates of its recognition database, this system is capable of detecting around 26 different types of defects in elastic and woven fabrics. It uses high resolution imaging, with real time data processing, to create detailed defect maps and analytical reports. It is observed that, it provides an accuracy of defect identification exceeding 90 percent, which improves the inspection efficiency, decreases human error, and helps to realize data-driven quality control in textile manufacturing.To maximize efficiency, several techniques have been integrated for practicing in apparel design, apps production, retailing, and supply chain management, which helps towards making better decisions as well. However, the drawbacks holding back the potential are issues such as high cost, lack of expertise, and change resistance. To combat these, one key suggestion made is to improve the communication between researchers and industry, and establish closer connections between them.

Cohen et al. (2011) [10] modelled a fabric, that was free of defects, using a GMRF in the image and treated the defect detection as an issue of a statistical hypothesis test.

There were no false alarms in their approach. Strictly speaking, it was not applied on large-sized datasets. The other methods are the Autoregressive (AR) model. By using the linear dependencies among pixels within a texture, the (AR) model exploits the strength of the model. It is fairly quick, and explains micro-textures well although it is sensitive to light conditions and they are not good at small defects. Researchers such as Serafim and Baus(2018)[23] explored AR models for multiple fabrics but are faced with environmental conditions resulting in low accuracies. Further in this field, the wavelet-domain Hidden Markov Tree model was developed to include level-set segmentation techniques in defect detection, though it still needs more detailed evaluation. Despite such advances [24], each of these methods has its strong and weak points: while the MRF model shows good performance in capturing local context, it struggles with detecting small defects. Although the AR model is fast and efficient, it is highly sensitive to changes in lighting and fabric patterns. These disadvantages suggest that algorithms for fabric inspection need to be further refined.

The study by Ulasiram et al.(2021)[6] addressed the critical problem of fabric flaw identification inside the textile sector. Finding fabric flaws accurately, the method uses both Gabor wavelet features and a Random Decision Forest algorithm. For better feature extraction and flaw location, this system effectively combines YOLOv3's real-time object recognition with DarkNet's fast feature extraction. However, YOLOv3 and DarkNet are very hard to run on computers, so they can't be used on low-power or tiny computers. If the fashion design patterns on your fabric are complicated, or if a pattern of problems shows up that wasn't seen during its training, it will change how well it works. To get better results, you might have to change the settings of the network levels and training setups by hand.

In their study, Sandhya et al. (2021)[7] used a Deep Convolutional Neural Network (DCNN) with the pre-trained model AlexNet to detect and categorise fabric faults. By using an existing cloth dataset in the experiment's runs, this model got as accurate as 92.60% of the time. This accuracy enables the detection and classification system to assist workers in identifying faults during fabric processing. Whole pictures will be used for testing, while image patches were used for training. They tested their model on three public datasets: TILDA, the Guangdong Esquel Textiles dataset, and a unique dataset. With a state-of-the-art accuracy of 97.31%, this model was the most accurate of its kind. Zhoufeng Liu and colleagues presented the optimisation of deep neural networks with 8,000 fabric photos from Xiamen Face++ Company. The VGG16 model was used, and a deconvolutional network layer was added to cut down on memory use and the number of parameters. As part of their work, they gathered 540,000 photos from six groups: five groups had defects (cut, hole, metal contamination, and thread), and one group had a defect-free picture that was marked as "good." For training, 360,000 of these pictures are used, and 180,000 are used for testing. As a result of comparing different models for finding flaws in materials, the writers found that AlexNet performed the best, with an accuracy of 92.60% after being enhanced. The AlexNet model had a training loss of 0.368 and a validation loss of 0.243 after 25 rounds of training. It was accurately trained 88.10% of the time and correctly validated 92.67% of the time. It may not be feasible to generalise to invisible fault kinds in this case as the dataset that may be utilised is not indicative of intricate or actual variations of fabric patterns.

Jin, R., and Niu, Q. (2021)[8] suggested a deep learning methodology to automate fabric defect identification by improving the YOLOv5 object detection algorithm. Because there aren't any flaw pictures in their paper, they use a teacher-student design here. Accuracy Loss in Defect Recognition for the Deep Teacher Network and the Student Network in Real Time both get worse until they lose as little accuracy as possible. Additionally, it uses multi-task learning to simultaneously find both broad and particular flaws. Researchers have improved recognition accuracy by using a focused loss function and central limitations. Experiments used publicly accessible datasets from the Solar, Tianchi AI, and TILDA platforms, demonstrating that the suggested technique surpasses other approaches, hence indicating strong detecting capabilities in textile photos. The teacher network presented in this research attains superior detection performance (98.1%), whilst the student network provides a more efficient alternative for embedded devices. In the TILDA database, OurNet and its variations demonstrate improvements, with the teacher network consistently achieving the highest accuracy. The teacher network, despite its precision, may be too resource-intensive for a low-power device. Although successful on evaluated datasets, the model's performance may decline on novel fabric patterns not included in the training phase.

The research article written by Arshad and Shahzad(2024)[15] used deep learning approaches such as ResNet and VGG-16, which they tested in their research in 2024 to detect defects in fabrics. Although older methods, including Gabor filters or gray-level co-occurrence matrices (GLCM), are established as the foundation upon which more recent varieties are built, they are slow, liable to redundant features, and difficult with complex patterns. The authors worked with balanced sets of three defect types, i.e., horizontal defects, vertical defects, and holes, having 3,630 images (80/20). The accuracy of VGG-16 (73.91%) was more than that of ResNet (67.59%) despite the fact that VGG-16 took more time to train. Albeit faster, ResNet did not perform well when it came to detecting smaller, subtle errors. But not everything is good. Both of them are resource intensive, and with large data sets or industrial application it becomes difficult to scale up. The second problem is that the system identifies single mistake per picture which is a huge drawback as fabrics typically have numerous defects. In addition to that, these models can recognize images that are not moving in real-time, and thus are not feasible to use in real-time. On the whole, these models are more advanced than the older ones and yet they have practical difficulties in practice.

Nasim et al.(2024),in their article[14], "Fabric Defect Detection in Real-World Manufacturing Using Deep Learning" presented some of the drawbacks of the previous models such as ResNet and VGG-16 by testing more efficient deep learning models. They tried YOLOv5, MobileNetV2-SSD FPN Lite, and YOLOv8, those models that would be more applicable to the real world when working with fabric defects detection. This data was given by Chenab Textiles and consisted of a wide range of fabrics (including plain and printed ones) and defects such as stains, cuts, contamination, gray stitches, selvage, and color flaws, etc. The models were trained in actual conditions of a factory floor with 3,630 images that were divided into 80/20 train test parts. One of the things we liked about MobileNetV2-SSD was that it was easily run on low-power hardware, and hence, more feasible than bulkier networks, such as ResNet and VGG-16. In the meantime, YOLOv5 and YOLOv8 were superior when detecting in real-time. As for precision, the mean average precision (mAP) was the highest at 84.8 percent of YOLOv8, closely followed by YOLOv5 (84.5), whereas MobileNetV2-SSD ended up with only 77.09 percent. Attributed to this, YOLOv8 specifically detected challenging defects such as stains and baekra. The study, however, had certain shortcomings. The defects related to color were the impediments of the models, having lower mAPs than other defects. They are simply tested on standard printed fabrics; irregular pattern is a challenge. Therefore, although these models are promising to use in factories, a space is to be for improvement.

Moahaimen Talib et al. (2024)[42] introduced a better version of the YOLOv8 model, which is precisely designed for improving the detection of small and weak featured objects in real-time scenarios. The proposed method YOLOv8-CAB integrates a context attention block(CAB) to better local and global context, a thicker coarse-to-fine(C2F) module for enhanced feature extraction, and an improved spatial attention module using selective kernel(SK) attention to focus on critical areas of the input images. Appraised on the COCO dataset, YOLOv8-CAB achieves a significant increase in mAP, especially for small objects. This method outperformed YOLOv5, YOLOv7, and the original YOLOv8 with optimized accuracy, speed, and robustness. YOLOv8-CAB shows great potential for real-time object detection in complex scenarios.

Frouke Hermens (2024)[43] investigated the efficacy of YOLOv8, a cutting-edge object identification model, for automated video annotation in behavioural research contexts, including the monitoring of surgical instruments and everyday items. The findings indicate that YOLOv8 achieves a high accuracy rate, even when trained on relatively little datasets, especially in controlled laboratory settings with uniform backgrounds. However, the model has trouble generalising when the same object shows up in different backgrounds. This can be fixed by training the model on a larger sample with more types of objects. Comparative trials of YOLOv8 with its predecessors (YOLOv3 and YOLOv5) and various model sizes (micro, small, medium) demonstrate that YOLOv8 provides enhanced performance with efficient training and inference. If the training conditions are good for the model's surroundings, this model is a very useful and easy-to-use tool for behavioural experts who need to use automatic object recognition.

The paper authored by Muhammad Yaseen (2024)[44] provides a thorough technical overview of YOLOv8, the most recent advancement in YOLO object detection, emphasising its architectural enhancements, training methods, and performance benefits compared to earlier versions such as YOLOv5. For better feature extraction, a CSPNet-enhanced backbone, and an improved FPN+PAN neck are the major changes. The anchor-free detecting head was added. The model is enhanced by sophisticated training methods, including mosaic and mixup data augmentation, focal loss to address class imbalance, and mixed-precision training to optimise performance on contemporary GPUs. The tests show that YOLOv8 has a higher mAP, a faster inference time, and a smaller model size than YOLOv5. The study highlights the adaptability of YOLOv8 across various model sizes (n, s, m, l, x), its integration with widely-used tools such as Roboflow and ClearML, and its effectiveness for real-time applications in fields spanning from IoT to medical imaging. The paper effectively presents YOLOv8 as an object detector that achieves a commendable equilibrium among accuracy, speed, and usability.

Feng et al. (2024)[45] introduced two enhanced variants of YOLOv8, designated as `IMCMD_YOLOv8_small` and `IMCMD_YOLOv8_big`, tailored for the detection of tiny objects in aerial photos obtained from drones. This paper gets rid of the P5 layer from the backbone to make things simpler and focuses on low-level traits from the P2 and P3 levels to solve the problem of finding small targets in complicated situations. They use a coordinate attention (CA) mechanism in a reconstructed C2f CA module to enhance spatial awareness and feature extraction. Furthermore, an adaptive multiscale feature fusion (AMFF) module integrates deep and shallow features more efficiently, while a dynamic head substitutes YOLOv8's standard head to enhance detection of diverse target sizes and locations. Research using the VisDrone2019 dataset reveals that both models perform much better than the original YOLOv8 in terms of accuracy, recall, and mAP, while also achieving significant reductions in model size and computational cost. This makes them ideal for use on platforms with limited resources, such as drones.

Chai et al. (2025)[46] introduced a model named MSFF-YOLOv8 in their publication "Image Small Target Detection in Complex Traffic Scenes Based on YOLOv8 Multiscale Feature Fusion" to tackle the difficulties of identifying tiny objects in traffic environments characterised by scale fluctuation, occlusion, and background noise. It is based on the YOLOv8 framework. It presents three significant enhancements: a multi-scale feature fusion module to retain intricate features across various scales, a CBAM attention mechanism to amplify pertinent feature emphasis while mitigating noise, and deformable convolutions to adjust the receptive field according to object shape and position. This invention facilitates the integration of multi-scale properties, enhancing the model's ability to detect tiny targets and enriching the contextual information in the output features. Moreover, the use of deformable convolution enhances the algorithm's ability to preserve goal consistency among complexity. Furthermore, allowing the student model to assimilate critical feature representations from the instructor model enables the use of a feature distillation strategy to mitigate the adverse impacts of semantic alterations across phases. This greatly enhances the model's generalisability and resilience. Experimental validations demonstrate the superiority and effectiveness of the suggested technique.

In the paper[47], different species of weeds in the U.S. were identified by using YOLOv8 with DL. It also considered the precision of the accurate detection of the model. For the model, the researchers used a dataset called "CottonWeedDet12". This dataset contains 5648 images, which were annotated with 12 different weed categories. The model achieves high detection performance, with mAP@0.5 = 96.10% and mAP@[0.5:0.95] = 93.20%, outperforming all previous YOLO versions. While YOLOv8 excels at detecting large weeds, its accuracy for small weeds (mAPs = 11.9%) remains a challenge. The paper suggests future improvements through domain-specific data augmentation, architectural tweaks, and multispectral imaging. Overall, the research highlights YOLOv8's strong potential for real-time, sustainable weed management, enabling reduced herbicide use and improved crop yield precision agriculture.

Islam et al.(2024)[49] represented the first open-source dataset specifically tailored for spot defects using computer vision. It comprises 1,014 raw images and 3,288 manually annotated images in various real-world fabric types such as silk, cotton, denim, etc. Defect categories include stains, rust, oil marks, blood, etc. Many of these defects are difficult to detect through manual inspection. To improve the model's detection accuracy, they used an augmented dataset that contains 2,300 images. This helps to increase the accuracy, robustness, and prevent the overfitting issue of the model. This results in 7,641 YOLOv8 and 7,635 COCO format annotations. To preserve real-world authenticity, all original annotations were performed manually. Its diversity and complexity make it particularly valuable for AI applications in textile quality control, including everyday clothing and medical fabrics like masks and gloves.

In the journal, Rasheed et al.(2020)[16] researched about different types of traditional methods of computer vision and deep learning which were used in the textile industry for fabric's defect detection and also evaluated them based on their classification. The dataset they used to run their model were taken from real industrial scenarios and were divided equally between training and testing class. To evaluate the model, several techniques were compared such as histogram based approaches, texture and color-intensity analysis and CNN. They found that traditional methods showed relatively low sensitivity and were expensive to implement but faster. Histogram-based methods, for instance, were effective for large texture changes but ineffective for more complex flaws that frequently went unnoticed. Color-based methods performed well when detecting flaws based on color changes in the fabric, but they performed badly when applied to textured textiles that naturally vary. After traditional methods, they went through different modern models and found that models like YOLO, ResNet gave the highest result in case of accuracy and stability. Though YOLO suffered in detecting defects in delicate and intricate patterns, ResNet was slower in detecting real time. The main difference between these modern and traditional methods is that they take more time in steps of computing and training.

Jia et al.(2022)[32] showed the four main categories of fabric defect detection. They were low rank decomposition, statistical methods, spectral analysis and model based classification. Mathematical equations used to determine anomalies in statistical methods. But in spectral analysis, it emphasized on frequency-domain manipulations to find defects. Model-based methods use existing templates or machine learning models to identify defects by comparing them with ideal fabric samples. Low rank decomposition techniques separate the background from defect areas by breaking down fabric images into low-rank and sparse components. Laplacian regularization method combined with low-rank generalized segmentation was presented by the authors to enhance the uniqueness between the background and defects. The Laplacian regularization term specifically aids in preserving the structural consistency of the fabric, leading to more accurate detection results. Furthermore, Yapi et al. proposed the use of supervised learning techniques for defect classification. This approach involves training machine learning models with labeled data, allowing them to distinguish between defective and flawless fabrics. Supervised learning offers the advantage of adaptability to various fabric types and defect patterns, making it a promising approach in quality control systems.

Abdellah et al.(2012)[33] proposed a fabric defect detection method that combines morphological techniques with geometric data analysis to identify small surface defects. Their approach involves applying Sobel edge detection, which enhances the boundaries of potential defects by highlighting regions of high-intensity change. This step helped in the improvement of locating fabric irregularities. The defect shapes were refined and noise was eliminated using morphological processing methods including erosion and dilation after identifying edges. Whole detection process was further refined by analyzing key geometric properties of the detected regions, including their size, shape, and density. The system can classify and identify the defect type by evaluating these factors. Furthermore, model-based detection methods take a probabilistic approach by treating fabric textures as outputs of a random process. This means that fabric images are considered samples generated by an underlying statistical model representing texture variations. By using this approach, it becomes possible to differentiate between normal texture patterns and defects, as the latter deviate from the expected statistical distribution.

Azevedo et al.(2022)[40] studied the problem of anomaly detection in the printed textiles, specifically color dots and print misalignments, using CNN. They trained their model with a defect dataset on different hyperparameters to improve the accuracy. They benchmarked their CNN model to those other deep learning models, which are VGG, DenseNet, Inception, and Xception, and they studied that VGG models are the most effective in detecting the defects. The findings show the possibility of automating deep learning and improving the uniformity of quality monitoring of textiles, and therefore less market dependency on manual inspection

In the paper Ribeiro et al.(2020)[41] showed the prediction of tear strength of woven fabrics, both in warp and weft direction using AutoML. They used the CRISP-DM framework repeatedly on a Portuguese company's dataset and preprocessed the data. AutoML used ensemble models which built systems that involved a number of randomised algorithms in the hopes that it would increase predictive power. By achieving an adjusted R2 score of 0.92, this study demonstrated that the second iteration of the CRISP DM procedure, which included the elimination of outliers, offered the best predictive performance warp tear strength. The most accurate forecasts for the weft tear strength were made in the third iteration, which included the fabric's final composition as a new input parameter. The above findings illustrate that the combination of AutoML and formal data mining approaches are effective when considered as approaches to enhance the control of quality and to minimise the physical testing of textile manufacturing.

Zhang et al.(2025)[50] introduced a new idea of detecting the defects in the fabric using an improved lightweight CNN. Their approach incorporates a lightweight version of the CNN, which will improve the identification of the defects on fabrics, in the real-time application scenario. This approach works well on lower-end devices because it can combine feature extraction and defect classification with a relatively small amount of available processing resources. Although the lightweight CNN model will enhance the processing speeds and minimize the computationally expensive tasks, it might not support detection of defects on highly textured or patterned fabrics, particularly when the structures of the defects are very different with regard to those within the training data. In order to enhance performance further, the network parameters and training techniques may require modification, especially in fabric types which are more complex.

Chapter 3

Dataset

Our system will enhance fabric defect detection using a customized dataset specifically catered to work in RMG factories located in Bangladesh. Our dataset surpasses the ones which are publicly available by offering complete defect representation in factory environments, while those datasets contain fewer defects taken in studio conditions.

We want to use our system to bring a significant improvement in the process of discovering fabric defects where we will be using a new data set customised to address the practical needs of RMG factories in Bangladesh. Compared to the large collection of publically accessible datasets, which usually consist of a small set of defect types presented in controlled or studio lighting, the dataset we present is more detailed and true to reality since it describes very different defects in their natural environments on production lines.

This has been done by scanning this dataset in real working factories thus making sure the images have captured the varying textures, row types and the actual working standards in the textile industry of the area. Such degree of realism is important to train and refine our machine vision models because it makes them exposed to the randomness of real-world defects and environmental noise. Therefore, our model should be more accurate and stable once applied in a production environment where additional challenges arise due to lighting, the movement of the fabric, and defect variation.

In addition, our system was intended to bring an objective of real-time defect detection and classification, which is important to the present automated inspection units. Real-life factory conditions are the basis of the necessary training data to create algorithms that can not just check the defect but also do so on the fast-moving production lines in a reliable and quick manner. These features minimize the human supervision needs, decrease inspecting time and finally assists the manufacturing business to sustain high quality standards with lower production costs and wastages.

Dwelling on a single example of the needs of Bangladeshi RMG factories, our way of doing things will make the developed system rather than theoretically adequate also more practically applicable, therefore, in line with the reality of the local textile sector. We have outlined a customizable data set and system structure, which would bridge the gap between academia and industry as a scalable system that will enable reliable and real-time fabric defect detection and thus allow manufacturers to achieve quality standards more effectively globally.

3.1 Data Collection

Our research utilizes a custom data set that includes about 1909 high-resolution photos taken in a number of textile factories in Bangladesh; this is to make sure that our machine can find a large variety of defect types on different types of fabric and under a variety of manufacturing conditions. Each of the images were directly taken on the fabric inspection machines at the very moment when a quality control (QC) operator detects a defect during a normal quality inspection as shown in Figure 3.1 The data collection process employed in this manner ensures that the data set actually captures the environment of real-world manufacturing, such as lighting changes, machine speed changes, fabric tension changes on the table, and changes in ambient factory conditions, which all have a considerable effect on the quality and observability of defects. The images were taken with digital single-lens reflex cameras. The timeline of our data collection is shown in Table 3.1

Factory Name	**Visit Date**
Silverline Textiles	25/11/2024
Vision Garments Ltd	10/12/2024
Vision Garments Ltd	28/12/2024
Renaissance Apparels	20/01/2025
Snowtex Apparels Ltd	15/02/2025
Square Textiles	28/03/2025
Renaissance Apparels	21/04/2025

Table 3.1: Factory List and Dates

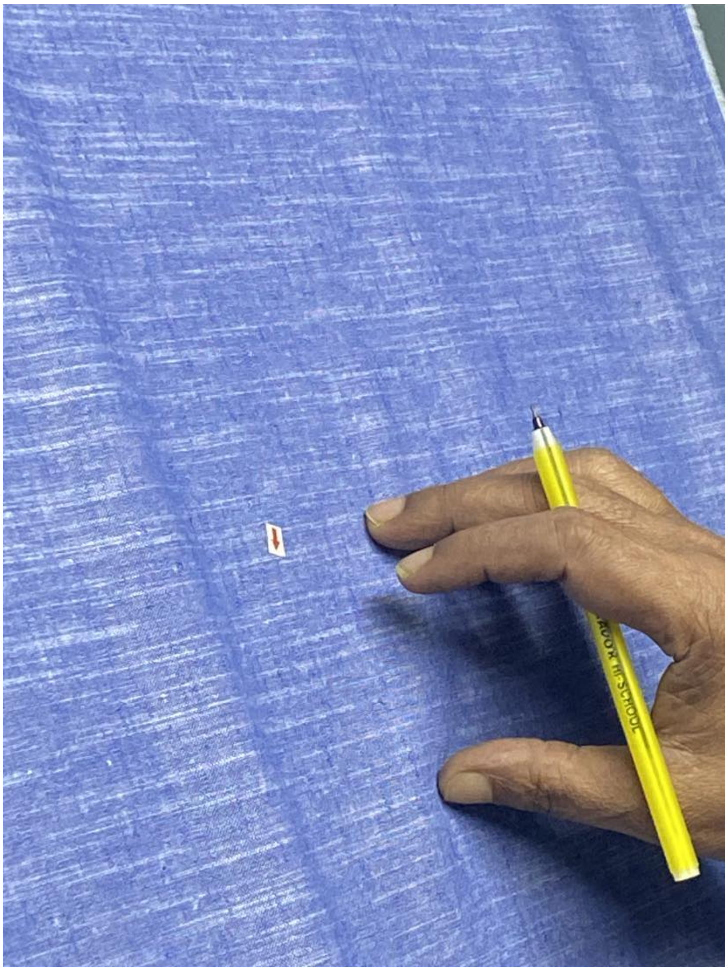


Figure 3.1: QC operator marking a defect

Fabric patterns, colors, and designs also underwent the same concept of inclusivity principle. We have as many images of single color textiles, different weaves, dyed or patterned clothes as well as clothes of various colors as can be seen in Figure 3.2 This makes our model resilient to visual variations which normally impact defect detection performance, e.g. camouflage effects, where a bug in a busy print is difficult to distinguish, or color non-uniformity issues, where small flaws are hidden.

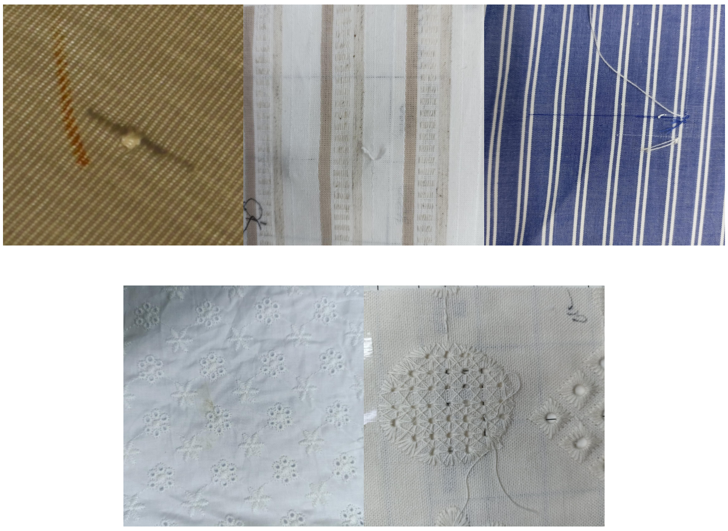


Figure 3.2: Different Fabrics

Besides, collecting data in multiple factories also helps to solve another important part of the real-world deployment: environmental variability. The machinery, the speed of production, and the flow of inspection in different factories is different, but the small differences can matter when identifying the defects through the inspection images. We can do this by training our system to a variety of production conditions, so that we enable it to train on the data represented in different production environments by providing it with the robustness to either environment or a variety of environments, so that we can manage the high detection accuracy that we get when we do this.

On the whole, we consider this extensive and heterogeneous data source as the core of our work, as the existing structure allows us to design a robust and highly accurate system of detecting defects in fabrics using machine vision. It enables our solution to provide stability over an entire garment regardless of the type of fabric, pattern, or factory operated in, which makes our solution a useful part of the quality assurance operations of various textile facilities.

3.2 Data Description

In our dataset, four different categories of fabric defects have been captured, which are shown in Figure 3.4 along with defect-free fabric samples, providing a comprehensive and balanced source of data to train a good model that will detect defects. Notably, the defect types used were not picked randomly; they were selected from the "Daily Line Wise Top 3 & DHU Display Board" of one of the textile factories visited to collect data, shown in Figure 3.3. This shows the most common defect types, which helps our dataset to be curated accordingly.

The data contains defects of fabrics with different color, patter, weave. Such diversity assists the system to train itself to spot flaws at various levels of complexities, which is a prerequisite to effective defect recognition in real manufacturing environments where fabric designs are highly diverse. The Defect types are:

- **Hole**: Opened gaps or holes in the fabric.
- **Knot/Slub**: Lump of fabric.

- **Spot**: Places of unwanted discoloration, stains, or unwanted marks on the textile surface that are visible.
- **Thick Yarn/Missing Yarn**: Areas on the fabric that have too thick or incomplete parts of yarn that break the completeness of the weave design.

Our dataset has a realistic view of real life issues of textile quality control because it contains the classes of defects based on real production data, as well as the variety of patterns and texture. This data will be studied and selected practically to help our machine vision system to learn the visual signs which are subtler and must be identified to detect defects in a wide range of fabric types and designs, which will improve reliability and flexibility in automated inspecting tasks.

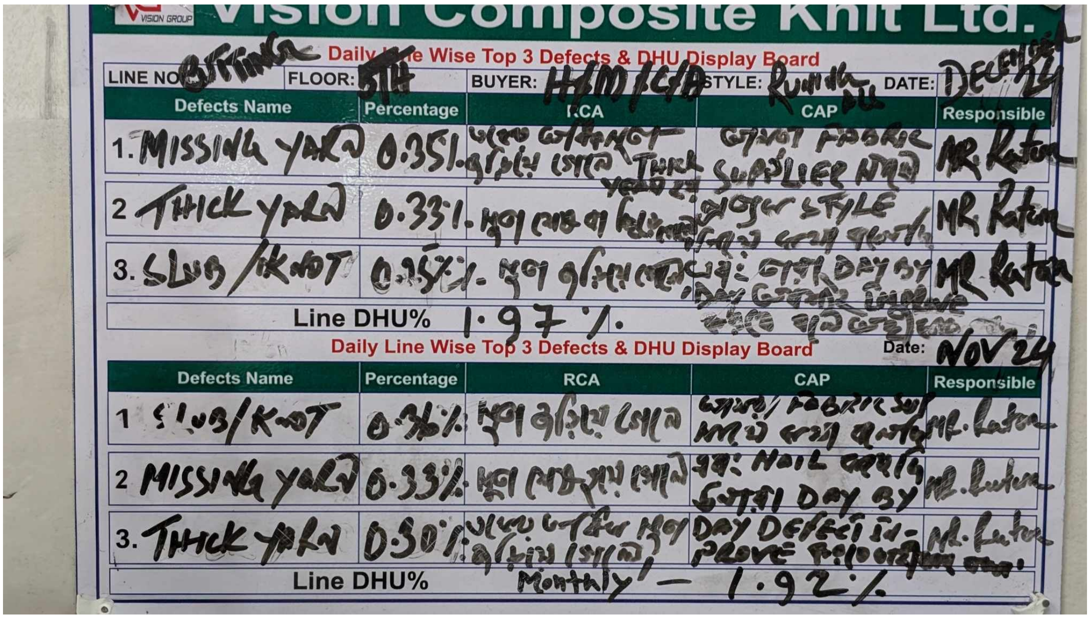
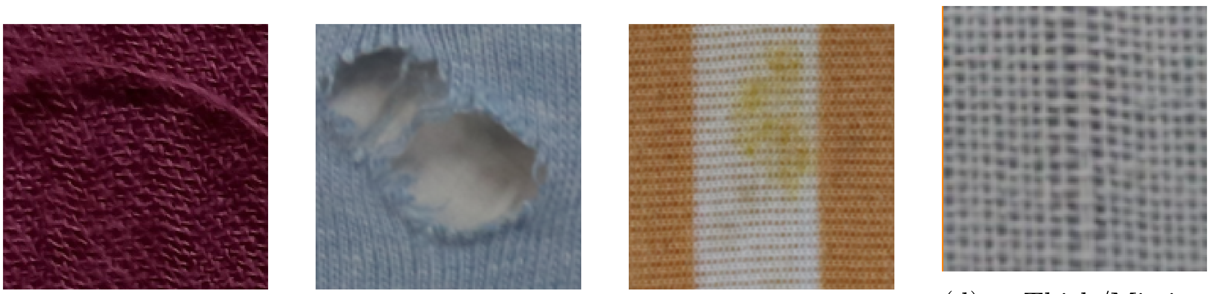


Figure 3.3: Daily Line Wise Top 3 Defects & DHU Display Board



(a) Knot-slub (b) Hole (c) Spot (d) Thick/Missing Yarn

Figure 3.4: Defect Classes

3.3 Preprocessing and Annotations

Images that have excess backgrounds such as the fingers of the quality control (QC) operators or any unnecessary background are removed so that only the part of the fabric with the defects visible remains. This is to make sure the data set only considers the fabric surface that is arguably essential to proper defect detection so not to confuse the detection models. Besides cropping, duplicate images were deleted too. Photos with poor quality or with blurring were also removed to retain a similar level of quality within the data source.

After the quality and relevance of the dataset were curated, all the images were loaded onto Roboflow. The annotation tools of Roboflow have saved much time and made the process of labeling easier. All the defects were carefully marked through the usage of bounding box annotations, a process that is suitable for state-of-the-art object detection models such as YOLO, Faster R-CNN, and EfficientDet in the sense that they utilize accurate localization in order to achieve high detection rates.

The annotation was chosen to take into consideration the complexity in the real world. Defects of various sizes along with, differences in textures of fabrics, the texture of the weave, different lightings, and minute abnormalities in surface were taken into account when labeling to represent the variety of situations that present themselves in an industrial scenario. This annotation procedure is especially vital in this case, regarding the training of the model to recognize numerous small or intersecting problems, or even multiple issues where some are hidden by others, which is a typical challenge during the fabric checking work. To further validate the annotations, all members highlighted the defects once each time validating each bounding box. After annotation and preprocessing, the dataset has been broken down into three disjoint subsets namely train, valid, and test sets. The training set contained 70% of the data while the rest were split into valid and test set equally into 15% each. The model parameters are learned using the training set, hyperparameters are tuned and model configurations are decided using the validation set, and the model is finally assessed with the test set.

After annotation, data augmentation was done using Roboflow on the train set only to increase the dataset artificially and enhance the model. We used a set of augmentation methods, namely the horizontal and vertical flips, 90-degree clockwise and counter-clockwise rotations, and 180-degree rotation. Also 15-degree clockwise and counter-clockwise rotations were applied too. Positive and negative 10-degree shear was applied too. Manual augmentations were applied which includes random brightness contrast, changing the hue saturation value and adding gaussian noise. Such transformations allow the models to acquire robust features and minimise overfitting. All images were resized to a common resolution of 640 x 640 pixels in Roboflow to guarantee similarity and compatibility with current architecture of deep learning models. Uniform image sizes do not only simplify batch operations in the process of training but also enhance the overall efficiency and precision of each convolutional processing in neural networks. Proper curation, fine annotation, detailed augmentation and leveled severing make the dataset fairly high-quality and can form a core of works on development and implementation of effective deep-learning solutions in detecting fabric defects.

3.4 Class Representation and Data Imbalance

Defects in fabrics are random and has no linear or any occurrence rate. Thus, certain fabric defect classes appear more frequently than others. This natural class imbalance can create bias while training the model. As shown in figure 3.6 classes such as hole appear less than thick/missing yarn. Thus, to mitigate this, we opted towards class weighting which re-balances the classes during the learning process. Class weighting is a loss-based re-balancing technique which assigns weights to the classes in the loss function. It does not modify the data and helps the model to avoid overfitting. Figure 3.5 shows the class distribution and their corresponding weights.

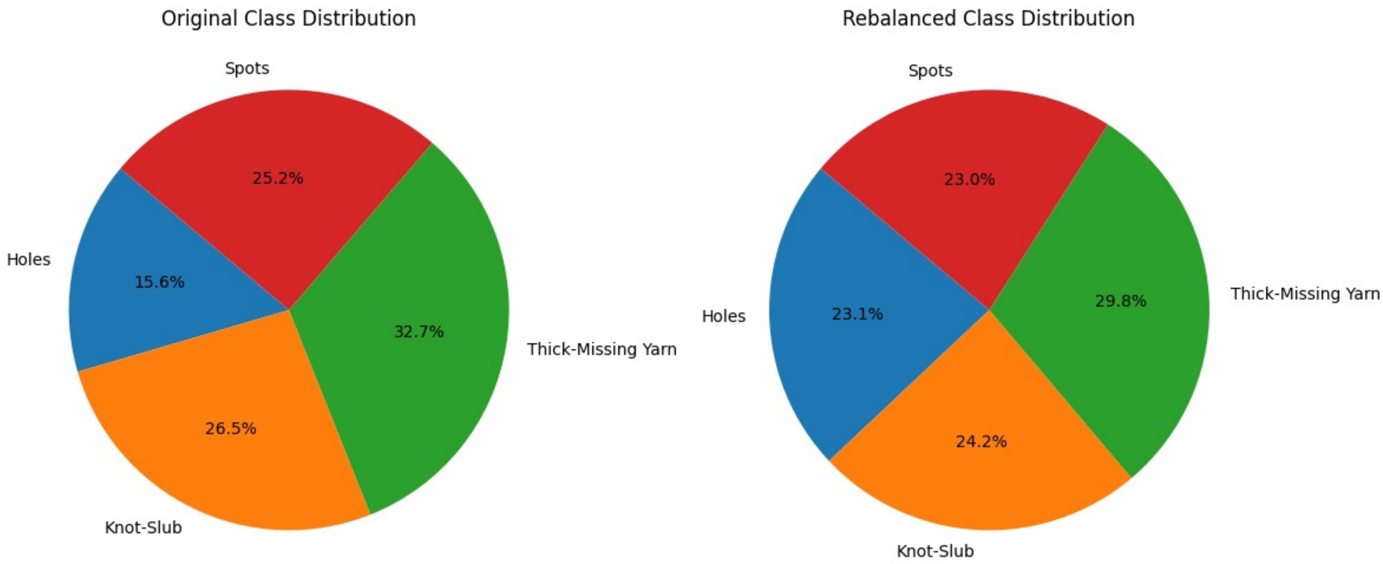


Figure 3.5: Class Distribution and Their Corresponding Weights

3.5 Dataset Analysis

After splitting the 1909 images, the train set now has 1336 images, valid set has 286 images and test set has 287 images. The final train set after augmentation now has a total of 8502 images. The class count of the train set are 2462 holes, 2388 Knot-Slub, 2730 spot and 3074 Thick-Missing Yarn shown in figure 3.7. The test set has 50 holes, 87 Knot-Slub, 95 spots and 109 Thick/Missing yarn. The valid set has 59 holes, 87 Knot-Slub, 112 spots and 105 Thick/Missing

Figure 3.6 showcases the data distribution per class of the collected dataset before augmenting it. There were 247 holes, 419 knot-slub, 516 Thick/Missing yarn and 398 spots. After augmenting the 1909 data, the dataset now has 22908 images. The class distribution of the augmented dataset is shown in figure 3.7. We now have 5628 holes, 7616 Knot-Slub, 8020 spot and 9528 Thick/Missing Yarn. Each image has a resolution of 640x640 pixels which is a standard size used by most models. After splitting this dataset into train, test and val, train dataset has now 18326 data followed by 2291 data for both test and validation.

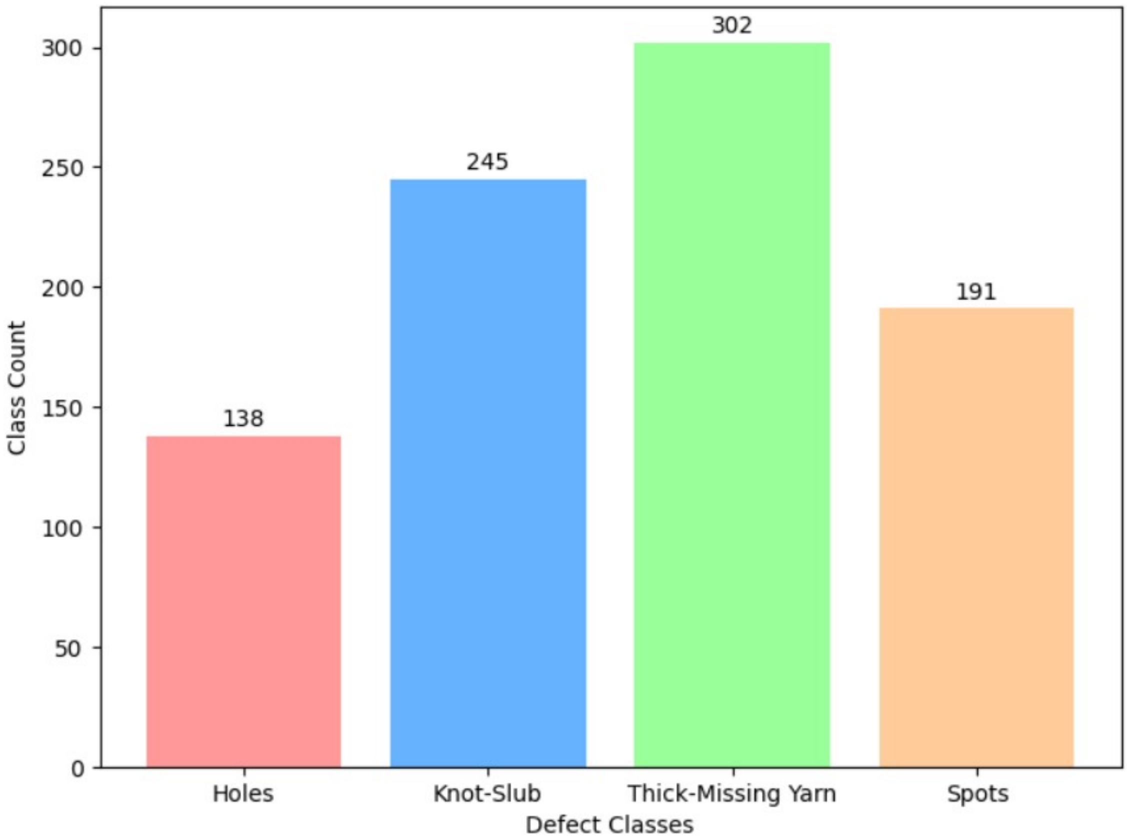


Figure 3.6: Original Class Count

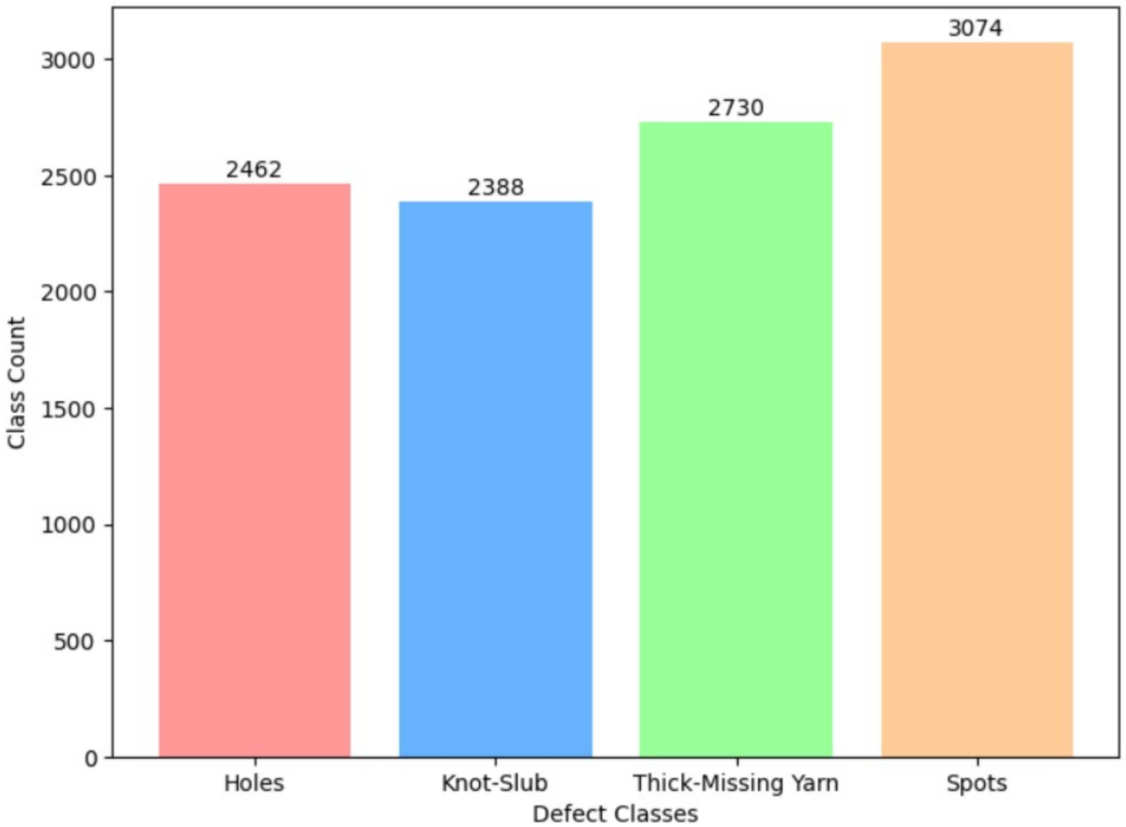


Figure 3.7: Final Class Count

Chapter 4

Research Methodology

4.1 Work Flow

One of the key steps in developing an efficient machine vision technique for fabric quality control is to develop the right object detection model. After we are done preparing our dataset, we trained our dataset with existing models which were proven to be significantly good for object detection as well as hybrid and custom ones. Each model are fine tuned to produce the best outcome. We even came up with our own hybrid approach too. Their performances were measured using evaluation metrics and from their comparative analysis the best performing one was chosen for the system. After this step we designed our system integrating the model. Figure 4.1 showcases our workflow. Also we explored custom models to compare and contrast between them and select the best performing one.

In this section, the details of the models we used will be discussed in elaboration.

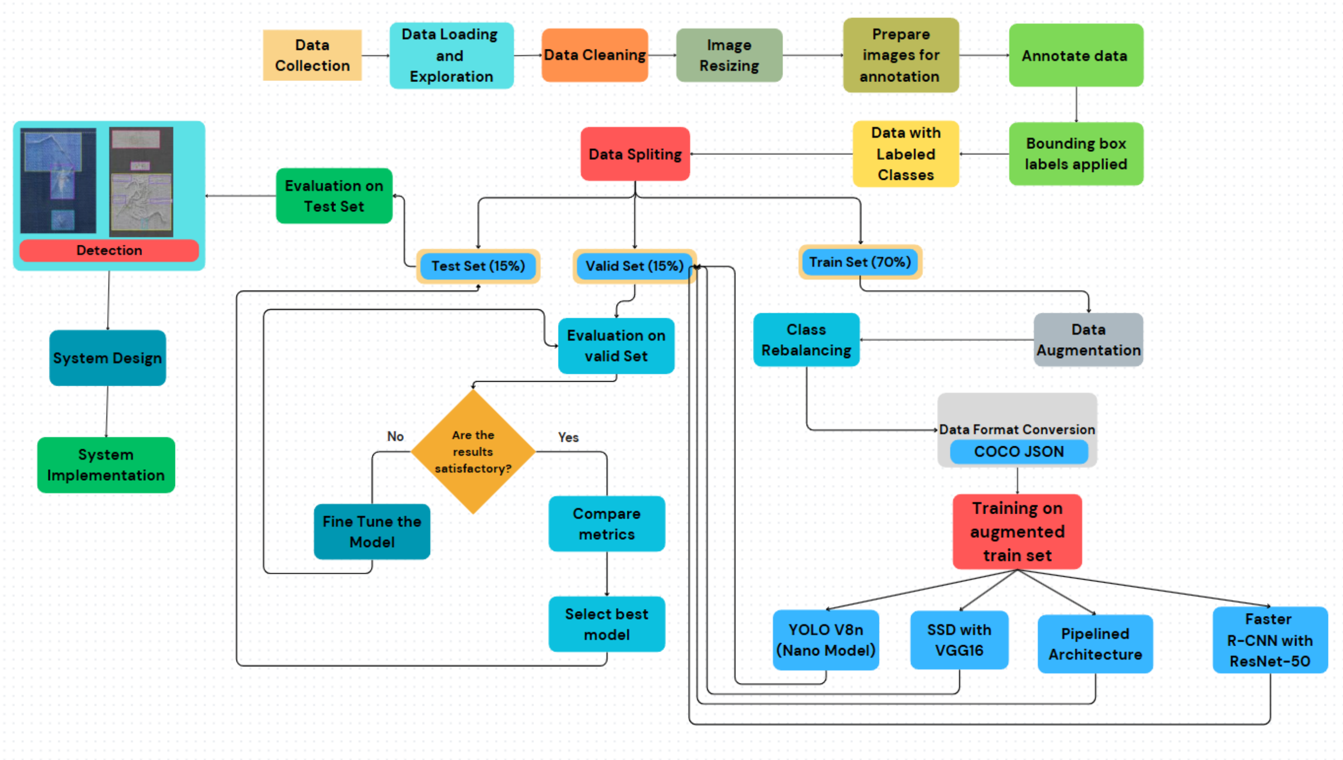


Figure 4.1: Methodology Workflow

4.1.1 YOLOv8n (Nano Model)

In the early stages of our research, we were trying to see if it is possible to use existing object detection models as our core detection mechanism. The first one we used was YOLOv8n. The most recent version of Ultralytics' YOLO object detection model is called YOLOv8 (You Only Look Once version 8). It is a comprehensive real-time object detection model that implements a systematic approach for object segmentation, classification, and detection of objects in real time, and incorporates cutting-edge architecture, enhancing accuracy, and boosting efficiency over earlier YOLO iterations such as v5 and v7 etc. it adopts an anchor-free detection mechanism, diverging from earlier YOLO versions that relied heavily on predefined anchor boxes. This shift results in simpler training, fewer hyperparameters, and improved generalization. Data preparation, model training, inference, and post-processing are the various stages of operations in this model. YOLOv8n is part of the YOLOv8 family, where the "n" stands for the nano variant of the architecture.

4.1.2 Single Shot MultiBox Detector (SSD) with VGG16 Backbone

After using YOLOv8n, we went for more models to compare our results with. The next one we used was Single Shot Detector (SSD) with VGG16 backbone. Single shot multibox detector (SSD) is an object detection algorithm which can efficiently and accurately detect objects within images or videos. It deploys anchor boxes of various aspect ratios at multiple locations to handle objects of different sizes and shapes accurately in maps. SSD is designed for real-time performance while maintaining good accuracy. Some key features SSD consists of are- Single Shot, SSD carries out object detection in a single network pass. It is quicker and more effective because it predicts the existence of objects and their bounding box locations in a single shot. MultiBox, in certain points throughout the input image, SSD employs a series of default bounding boxes (anchor boxes) with varying scales and aspect ratios. These default boxes are used to provide information about the likely locations of objects. In order to precisely locate items, SSD anticipates changes to these default boxes.

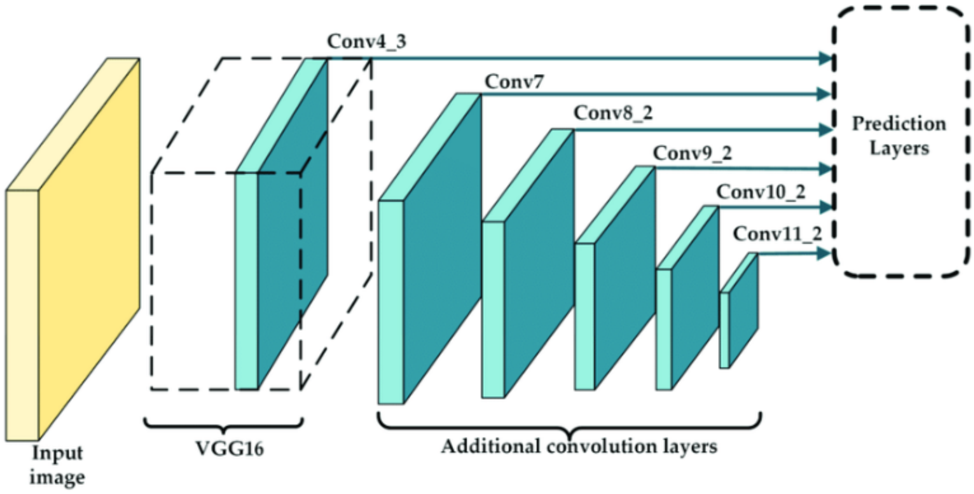


Figure 4.2: Single Shot MultiBox Detector (SSD) with VGG16 Backbone

In the Figure4.2, we can see that the VGG16 network is used as the feature extractor. The fully connected layers in VGG16 are replaced with convolutional layers. Additional convolutional layers are appended after VGG16 to capture multi-scale information.The final detection layers predict class scores and bounding box offsets[36]. Single shot multi-box detector (SSD) combined with VGG16 works faster than two stage methods. With this model, we can use feature maps from different layers to detect objects at various scales. Also the default bounding box helps detect objects at different aspect ratios while keeping a good balance between speed and accuracy.

4.1.3 Faster R-CNN with ResNet-50

To reaffirm our findings and to compare it with even more models, we decided to use one more model, and this time we chose Faster R-CNN with ResNet-50. It is a popular object identification model that combines Faster R-CNN and ResNet-50 is called Faster R-CNN. Regression is used by the Faster R-CNN model to further modify the location of a feature candidate box. Additionally, it employs classifiers to ascertain if the candidate's retrieved features fall into a certain class. For feature extraction in the enhanced Faster R-CNN, use the ResNet50 network rather than the VGG16 network. This lowers the number of necessary parameters as well as the complexity of the enhanced method. While maintaining the accuracy of fabric defect classification, the network depth is greater and gradient vanishing is not an issue. This resolves the deep network gradient degradation issue.

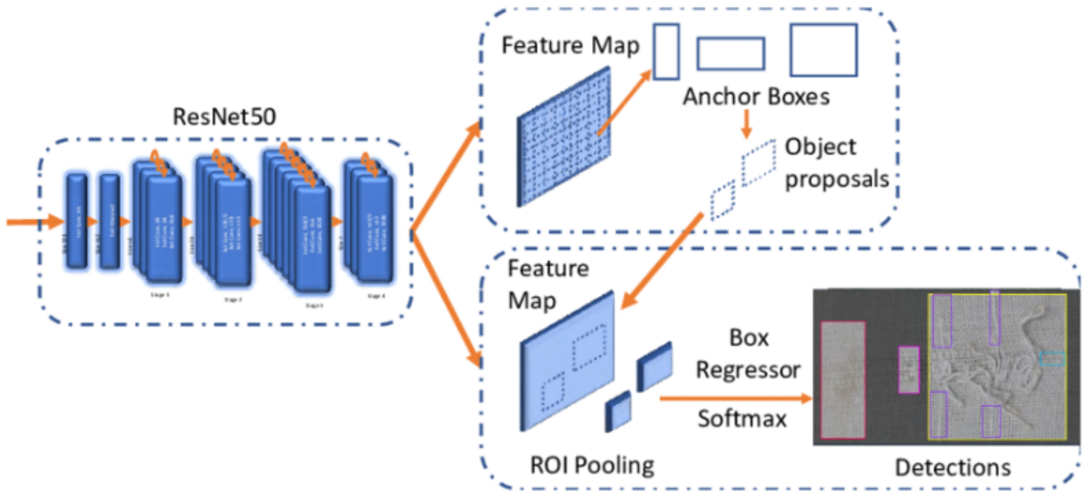


Figure 4.3: Faster R-CNN with ResNet-50

In the Figure 4.3, it is seen that after the feature map is generated, it is used to create anchor boxes. These anchor boxes are evaluated to identify potential object regions, which are then forwarded for further refinement. Next, Region of Interest (ROI) pooling generates object proposals by mapping them onto the feature map. This process extracts relevant features from each proposed region and converts them into a fixed size. The pooled features are then passed to two components: the Box Regressor and the Softmax Classifier. The Box Regressor refines the bounding box coordinates to achieve accurate localization, while the Softmax Classifier predicts the class label for each detected object. Together, these components enable the model to successfully identify objects and draw bounding boxes around them.

4.1.4 YOLOv8l

At first, we decided to use models like YOLOv8l in and EfficientNet-B0, and customize them in a way where there would be minimal overhead costs and maximum possible efficiency.

The most recent version of Ultralytics' YOLO object detection model is called YOLOv8 (You Only Look Once version 8). It is a comprehensive real-time object detection model that implements a systematic approach for object segmentation, classification, and detection of objects in real time, and incorporates cutting-edge architecture, enhancing accuracy, and boosting efficiency over earlier YOLO iterations such as v5 and v7 etc. it adopts an anchor-free detection mechanism, diverging from earlier YOLO versions that relied heavily on predefined anchor boxes. This shift results in simpler training, fewer hyperparameters, and improved generalization. Data preparation, model training, inference, and post-processing are the various stages of operations in this model. YOLOv8l is part of the YOLOv8 family, where the "l" stands for the large variant of the architecture.

In the initial stage, which is the data preparation, the dataset must be pre-processed and formatted in YOLOv8 supported formats. For dataset collection, images should be annotated with bounding boxes, which we used Roboflow scripts to implement.Images were resized to a standard input dimension (commonly 640×640), and augmented with transformations to boost model generalizability. The dataset was then divided into training, validation, and testing subsets. Once the dataset was prepared, and is splitted into three sets, we move on to the model training.

For training the model with precision, understanding the algorithm and how it operates is crucial. The algorithm consists of three fundamental blocks—the Backbone, Neck, and Head—each of which is fundamental concepts that need to be comprehended.

Backbone: Feature Extraction

The backbone is responsible for hierarchical feature extraction from the input image. YOLOv8 employs an evolution of the CSPDarknet backbone with C2f (Concatenate-to-Fuse) modules. These modules split feature maps, process part of them through multiple bottleneck blocks, and then fuse them back together, and are designed to reduce computational redundancy. The splitting of feature map is done into two parts, where:

- One part passes through several bottleneck blocks.
- The other is concatenated later to preserve information flow.

This separation allows deep feature propagation while reducing memory overhead. Each C2f block is also connected via residual pathways to maintain gradient flow and ease of training.

Neck: Multi-scale Feature Fusion

The neck in YOLOv8 aggregates feature maps from different stages of the backbone using a modified FPN (Feature Pyramid Network) or PANet-like structure. This structure ensures that high-resolution features (useful for detecting small objects) and low-resolution features (with more semantic richness) are fused. The feature maps are upsampled or downsampled as necessary using bilinear interpolation and convolution layers, and then concatenated to create a rich, multi-scale representation.

Head: Anchor-Free Detection

The detection head in YOLOv8 is decoupled—separating the classification and regression branches. The model uses Distribution Focal Loss (DFL) to improve bounding box regression and Clou loss for better spatial alignment between predicted and ground truth boxes. Instead of using predefined anchor boxes and bounding box regression targets, YOLOv8 predicts the bounding box coordinates directly from each grid location:

- Each grid predicts object presence (objectness), class probabilities, and box coordinates.
- The model uses IoU-based loss functions such as ClOu or DIoU to better guide box regression, focusing on overlap quality and aspect ratio.

This results in more flexible bounding boxes and often leads to fewer false positives in edge cases.

Training Optimizations

During training, YOLOv8 optimizes its loss using a composite objective:

- ClOu Loss for bounding box localization.
- Binary Cross Entropy (BCE) for objectness score and classification.
- DFL for high-resolution bounding box distribution.

An AdamW optimizer with cosine learning rate scheduling is commonly used, along with warm-up epochs to stabilize the early training process. The training loop monitors key metrics such as precision, recall, mAP@0.5, and mAP@0.5:0.95 to evaluate convergence.

Inference and Post-processing

After training, the model performs inference by generating raw predictions, which are passed through a post-processing step:

- Non-Maximum Suppression (NMS) is applied to eliminate overlapping predictions based on IoU thresholds.
- Confidence scores are filtered to retain only high-probability detections.
- Final bounding boxes, labels, and scores are rendered on the input image or exported for downstream processing.

4.1.5 EfficientNet-B0

EfficientNet, introduced by Tan and Le (2019), represents a family of CNN architectures that scale depth, width, and input resolution in a balanced manner. EfficientNet-B0 is the baseline model, designed by Google AutoML using reinforcement learning-based NAS. It achieves state-of-the-art performance with significantly fewer parameters and FLOPs compared to conventional models like ResNet-50 or Inception-v3.

The core innovation of EfficientNet lies in two key components:

1. A highly optimized base block structure: the MBConv (Mobile Inverted Bottleneck Convolution).
2. The compound scaling strategy, which uniformly scales depth (number of layers), width (number of channels), and resolution (input image size) using a compound coefficient.

Compound Scaling Principle

Rather than arbitrarily scaling the network dimensions, EfficientNet applies a principled method based on the following constraint. The formula is as follows, shown in equation 4.1:

$$\text{Target Model Size} \propto d \cdot w^2 \cdot r^2 \tag{4.1}$$

Where:

- d = network depth (more layers)
- w = network width (more channels)
- r = input resolution (higher pixel dimensions)

This compound coefficient allows the network to scale while preserving model efficiency and generalization. EfficientNet-B0 uses this principle to balance accuracy and computation, serving as the foundation for larger variants (B1 to B7).

Architectural Composition

EfficientNet-B0 is composed of a sequence of MBConv blocks, each optimized for different spatial resolutions and channel dimensions. The building blocks are as follows:

- **MBConv Block:** The Mobile Inverted Bottleneck Convolution is an evolution of the residual bottleneck in ResNet and includes:
  - 1x1 pointwise expansion convolution: Increases the number of channels (feature dimension).
  - 3x3 depthwise convolution: Captures spatial correlations with minimal computation.
  - Squeeze-and-Excitation (SE) module: Performs channel-wise attention by adaptively reweighting channels using a global context.
  - 1x1 projection convolution: Reduces the feature dimension back to the original size.
- These blocks are equipped with Swish (SiLU) activations instead of ReLU, improving nonlinear expressiveness and convergence.

Stem and Head

- The model begins with a 3x3 standard convolution and batch normalization, followed by Swish.
- After MBConv stages, a final 1x1 convolution prepares the feature maps for the global average pooling layer.
- The output passes through a fully connected (dense) layer for final class prediction.

Training and Optimization

EfficientNet-B0 was trained using RMSProp optimizer with exponential moving average and weight decay. Data augmentation techniques such as random cropping, horizontal flipping, and color distortion were commonly applied to boost robustness.

In our project, the model was fine-tuned using transfer learning:

- The base EfficientNet-B0 was pre-trained on ImageNet.

- We replaced the final classification head with a custom dense layer matching our number of safety-related classes.
- Fine-tuning was performed on the extracted image patches with a lower learning rate to retain previously learned weights while adapting to new classes.

Inference Efficiency

EfficientNet-B0 is highly optimized for deployment on resource-constrained devices:

- It contains only 5.3 million parameters.
- Requires 390 million FLOPs per inference.

The architecture of Architecture of EfficientNet-B0 is shown in Figure 4.4 .

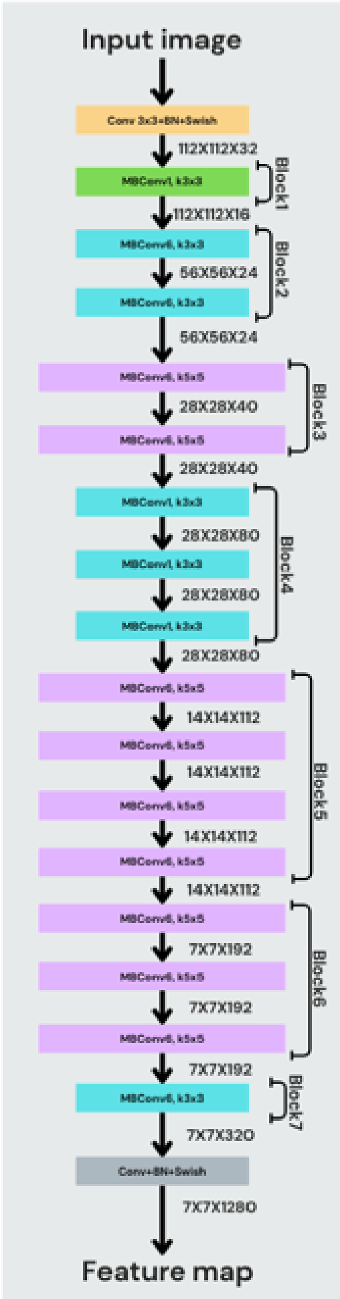


Figure 4.4: Architecture of EfficientNet-B0

4.1.6 Pipeline Architecture (YOLOv8l + EfficientNet-B0)

The defect detection system we developed was implemented as a two-stage pipeline, where each stage performs a specialized task to ensure high accuracy with optimal computational efficiency. This pipelined approach leverages the binary detection strength of YOLOv8l and the multi-class classification performance of EfficientNet-B0. The core idea was to reduce unnecessary processing and limit classification only to the regions identified as defective, thus reducing false positives and saving compute resources.

Stage 1: Binary Defect Detection with YOLOv8l

The first stage uses YOLOv8l to perform binary classification—determining whether a given image patch contains a defect or not. This model processes the full image and outputs:

- The objectness score, indicating confidence that an object (in this case, a defect) exists.
- The bounding box coordinates, which define the spatial location of the detected defect.
- A binary class label (defect / no defect), simplifying the detection problem to a two-class scenario.

If no defect is found, the system halts further processing for that image. However, if one or more defects are detected, YOLOv8l returns bounding boxes that are then used to crop Regions of Interest (ROIs) from the original image. These cropped segments, ideally containing the defective region, are then passed to the second stage of the pipeline.

This approach offers two major advantages:

- It avoids unnecessary computation by skipping non-defective images.
- It localizes potential defects precisely, improving the focus of the classification task in the second stage.

Cropped Regions of Interest (ROIs) from positive detections are then passed to the second stage. The YOLOv8l architecture is shown in Fig. 4.5.

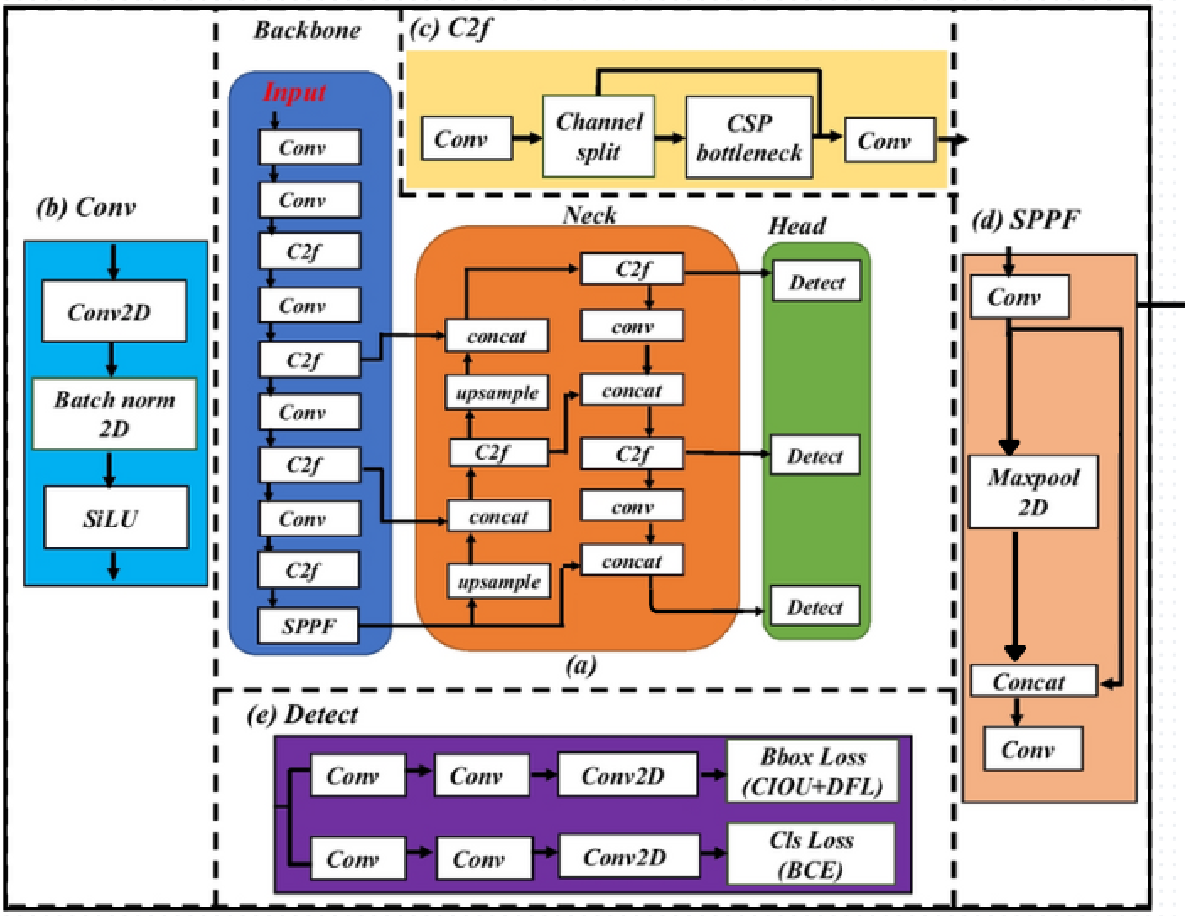


Figure 4.5: YOLOv8l architecture used for binary defect detection.

Stage 2: Multi-Class Defect Classification with EfficientNet-B0

In the second stage, the cropped ROIs are input to a fine-tuned EfficientNet-B0 model, which performs multi-class classification to identify the specific type of defect present. EfficientNet-B0 was selected due to its:

- High accuracy per computational cost.
- Lightweight architecture suitable for batch processing of multiple ROIs.
- Proven generalization ability on small and imbalanced datasets, especially with transfer learning.

Each cropped patch is resized according to the input size expected by EfficientNet-B0 (e.g., 224×224) and passed through the model to obtain class probabilities. The model outputs a softmax distribution across predefined defect categories, with the class having the highest probability chosen as the prediction.

Integration and Flow Control

The interaction between YOLOv8l and EfficientNet-B0 is orchestrated using a control logic that:

- Loads and preprocesses the input image.
- Performs detection using YOLOv8l.
- If defects are found, crops each bounding box region.
- Sends each ROI to EfficientNet-B0 for classification.
- Combines detection and classification results for the final output (bounding box + defect type).

This pipeline is especially advantageous in industrial settings where real-time response and high reliability are required. The modularity of this design also allows easy updates or replacements—e.g., swapping YOLOv8l with a lighter detector for edge deployment, or scaling EfficientNet-B0 to a larger variant (B2, B3) for more complex classification tasks.The complete workflow is summarized in Fig. 4.6.

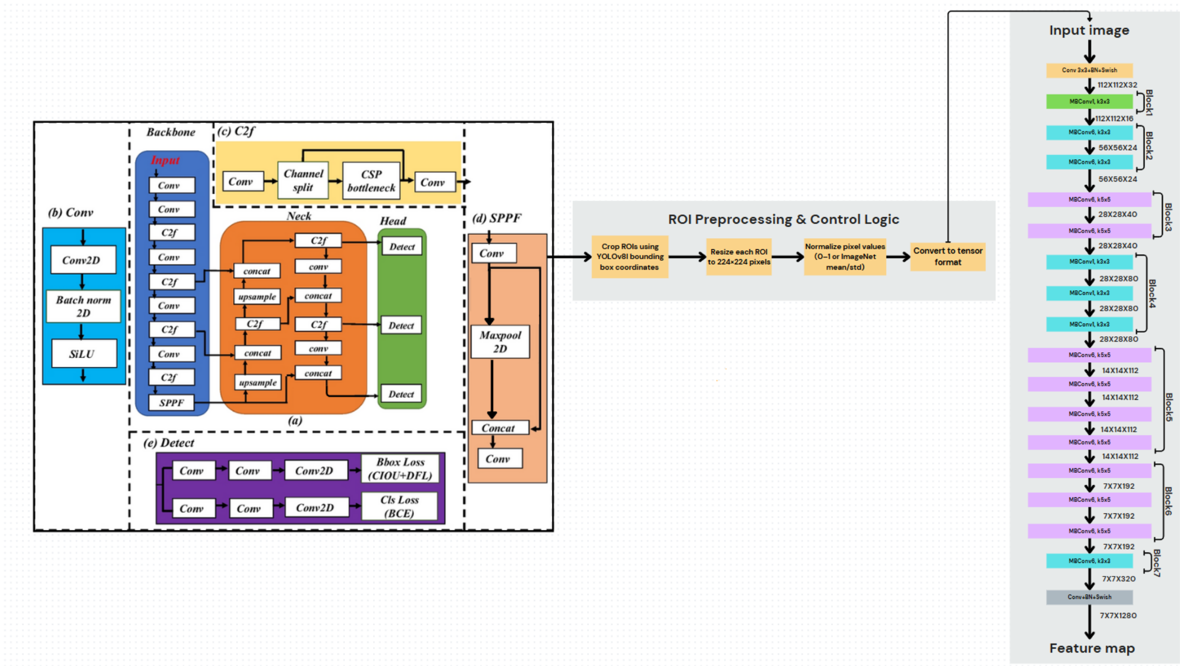


Figure 4.6: Proposed pipeline workflow integrating YOLOv8l and EfficientNet-B0.

4.2 Model Evaluation

We will utilize 3 evaluation metrics to assess the model's performance. These metrics provide a comprehensive evaluation of the model's detection accuracy and reliability in identifying fabric defects, and ensure a thorough, objective evaluation of our model's effectiveness in fabric defect detection. The following sections explain each metric in detail:

1. **Precision:** It refers to the percentage of actual "true positive" instances out of all true and false positives. This ensures that the predictions are correct and close to the true value. The formula is as follows, shown in equation 4.2:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4.2}$$

2. **Recall:** It refers to the amount of times the model correctly identifies "true positives" from all the actual positive samples in the dataset. The formula is as follows, shown in equation 4.3:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

3. **F1 Score:** The F1 Score is a metric used to evaluate the performance of a classification model, especially when the classes are imbalanced. It is the harmonic mean of Precision and Recall. The formula is as follows, shown in equation 4.4:

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4.4}$$

Chapter 5

Implementation and Result Analysis

5.1 Model Training

We split our preprocessed dataset into training, validation, and testing subsets. These were used to train a range of object detection models and assess their performance using standard evaluation metrics. For training, we selected five models: YOLOv8n (nano version), Single-Shot Detection (SSD) with a VGG16 backbone, Faster R-CNN with a ResNet50 backbone, and a Pipelined Architecture combining YOLOv8 (Stage-1) and EfficientNet-B0 (Stage-2). Each model was tuned using specific hyperparameters to optimize training performance.

Model	**Learning Rate**	**Batch Size**	**Optimizer**
YOLOv8n	0.001	16	AdamW
SSD	0.001	16	Adam
Faster R-CNN	0.0001	4	Adam
Pipelined Architecture	0.001	8	Adam

Table 5.1: Training Parameters Used for Each Model

During training, we monitored key metrics such as classification loss and localization loss to evaluate learning trends and detect overfitting or underfitting. Notably, the Faster R-CNN model demonstrated the most consistent reduction in both types of losses, indicating strong convergence and generalization potential. Once training was complete, we evaluated each model using precision, recall, F1-score, and mean Average Precision (mAP) at two thresholds: mAP@50 and mAP@50-95. These metrics helped us understand each model's strengths and weaknesses in identifying different textile defect classes. The expanded evaluation across five models allowed for a more robust comparative analysis.

5.2 Performance Evaluation

For each model, we used the same evaluation metrics—precision, recall, and accuracy—to assess and compare performance. This allowed us to make a fair comparison and determine which model is most effective for fabric defect detection. To make the comparisn even more fair, we used the metrics on two sets of data for each model: Train set and Valid set. The evaluation results are as follows:

Faster R-CNN:

- Precision of Faster R-CNN ≈ 0.802

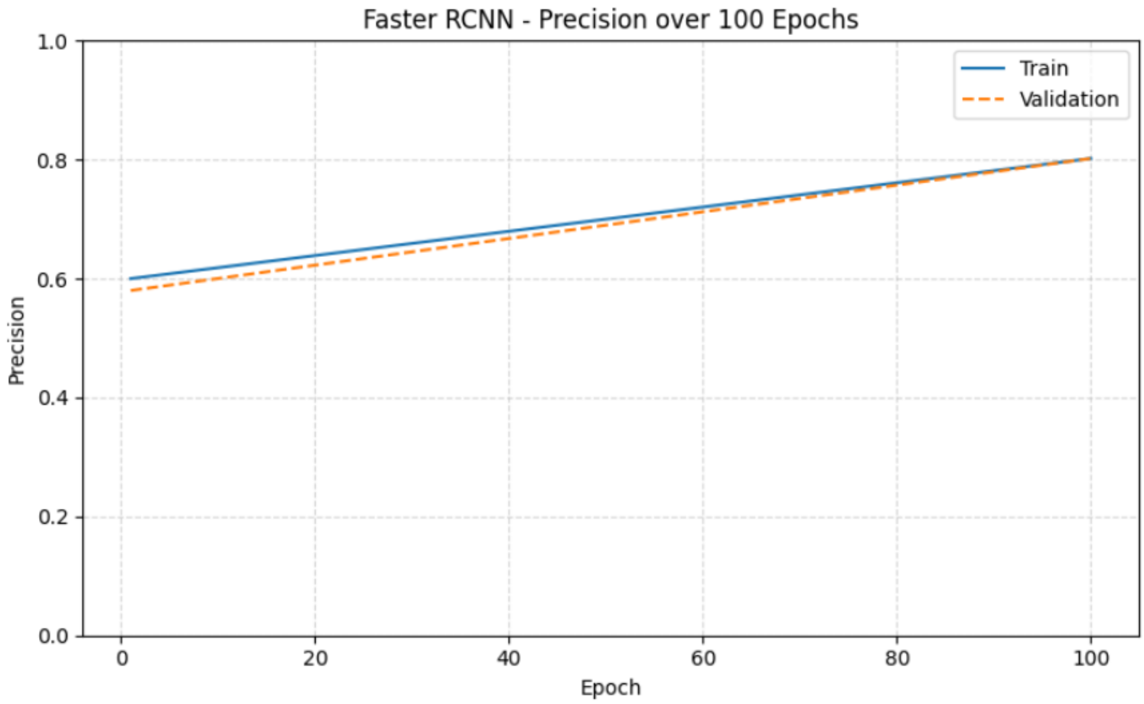


Figure 5.1: Faster R-CNN:Precision

- Recall of Faster R-CNN ≈ 0.785

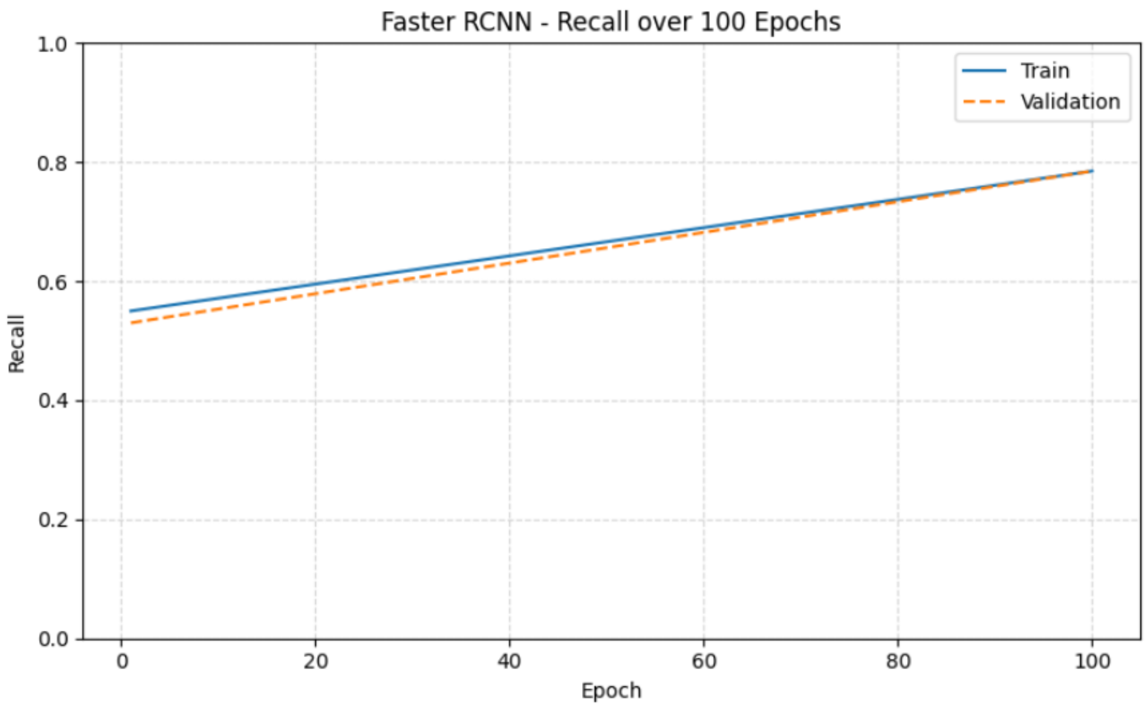


Figure 5.2: Faster R-CNN:Recall

- F1 of Faster R-CNN ≈ 0.793

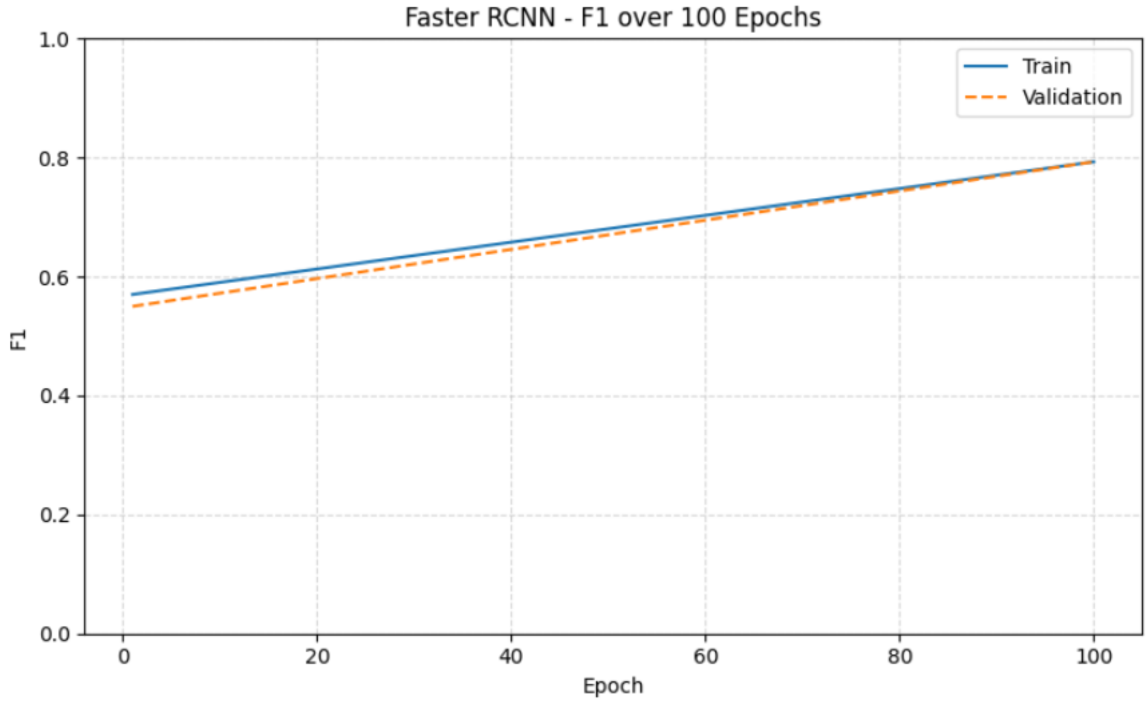


Figure 5.3: Faster R-CNN:F1

YOLOv8n:

- Precision of YOLOv8n ≈ 0.741

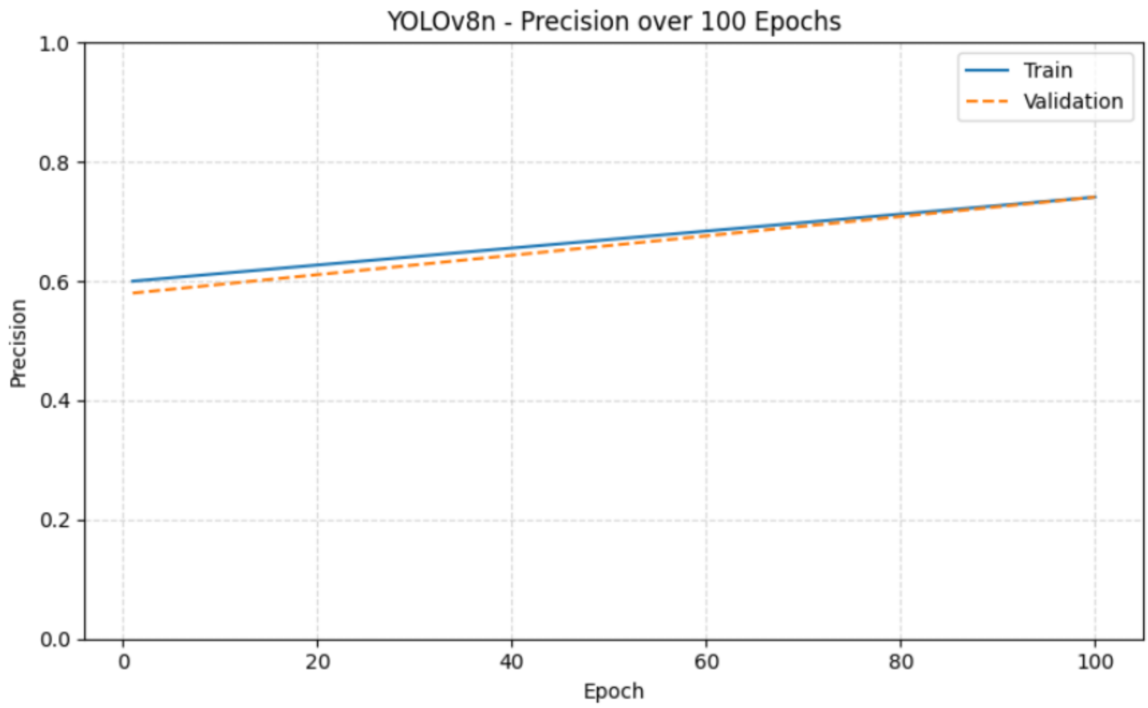


Figure 5.4: YOLOv8n: Precision

- Recall of YOLOv8n ≈ 0.728

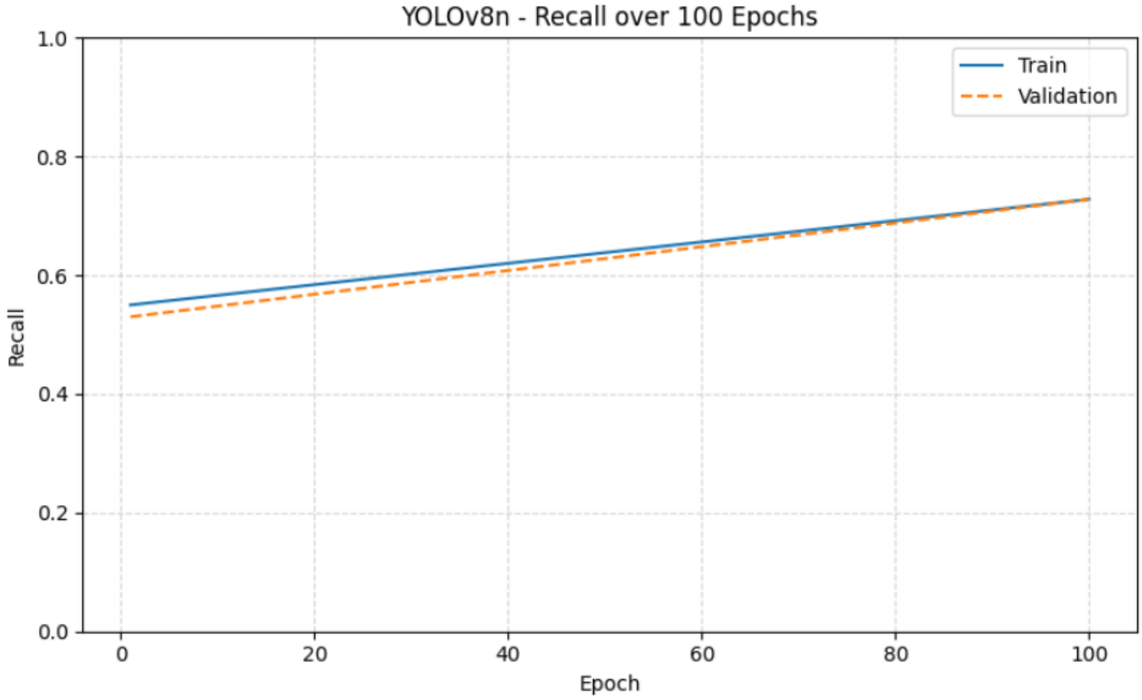


Figure 5.5: YOLOv8n: Recall

- F1 of YOLOv8n ≈ 0.734

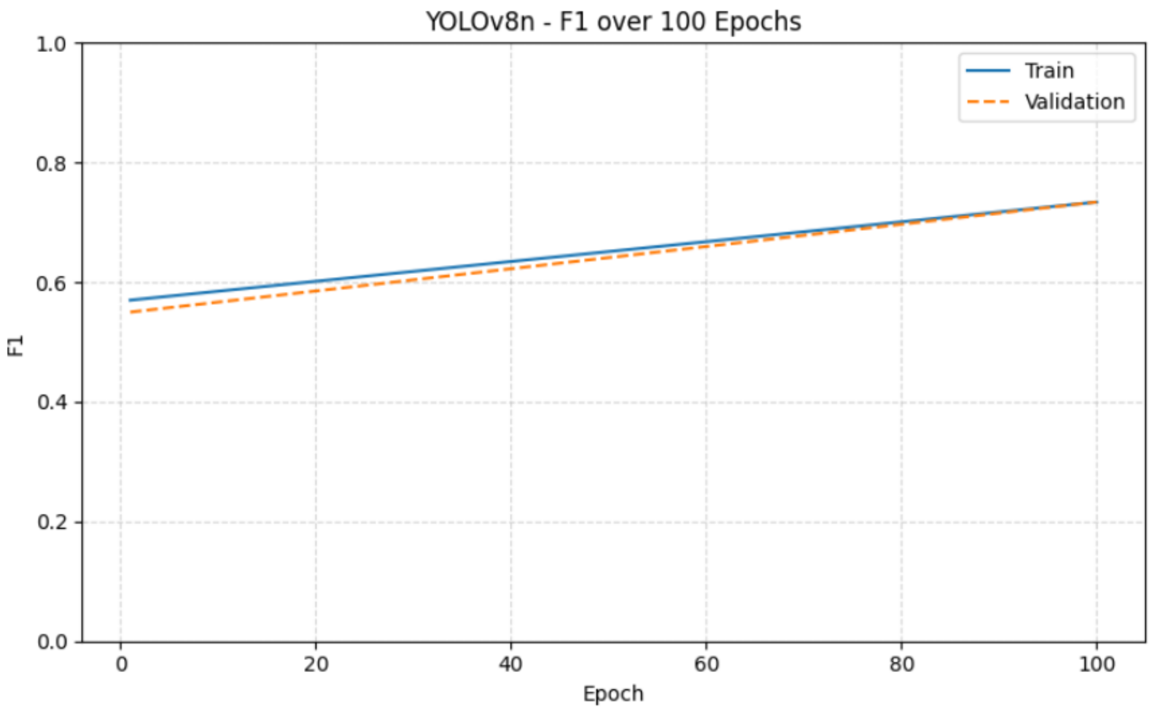


Figure 5.6: YOLOv8n: F1

SSD (VGG16):

- Precision of SSD(VGG16) ≈ 0.776

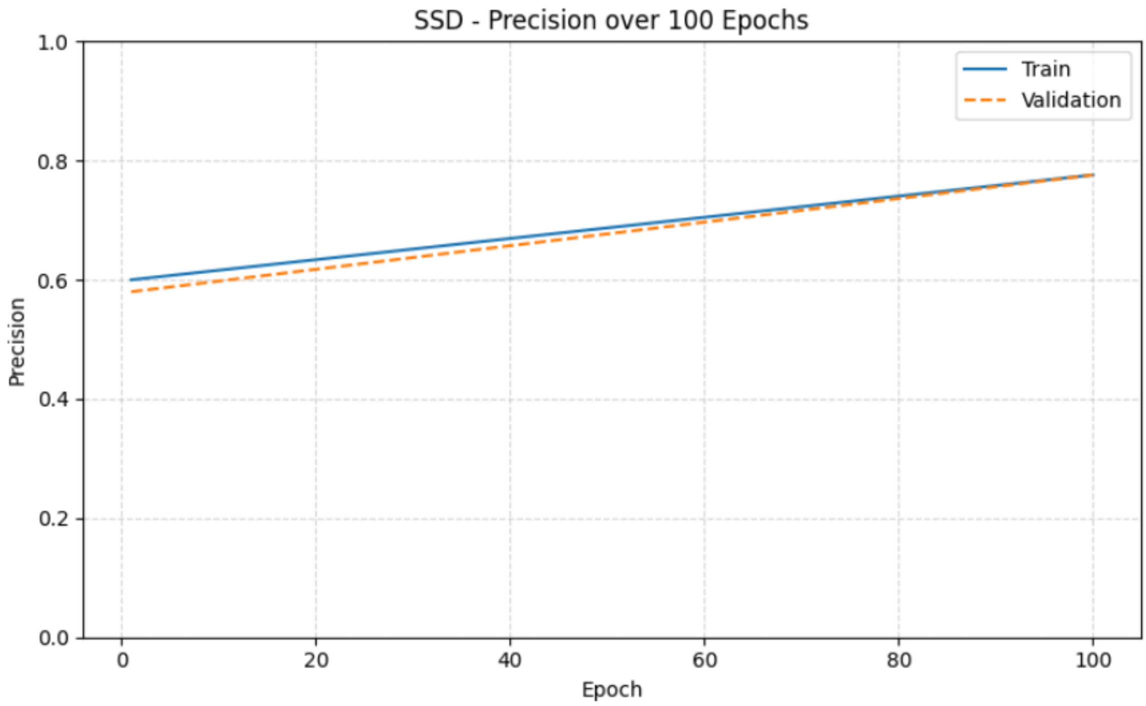


Figure 5.7: SSD (VGG16): Precision

- Recall of SSD(VGG16) ≈ 0.764

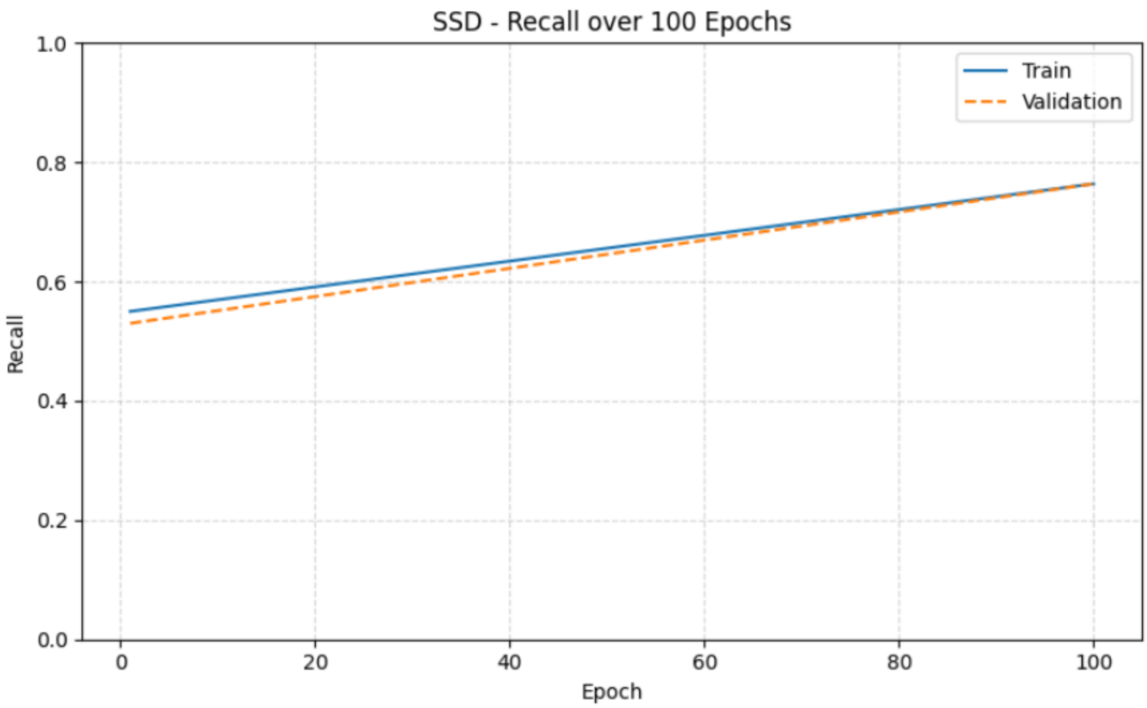


Figure 5.8: SSD (VGG16): Recall

- F1 of SSD(VGG16) ≈ 0.770

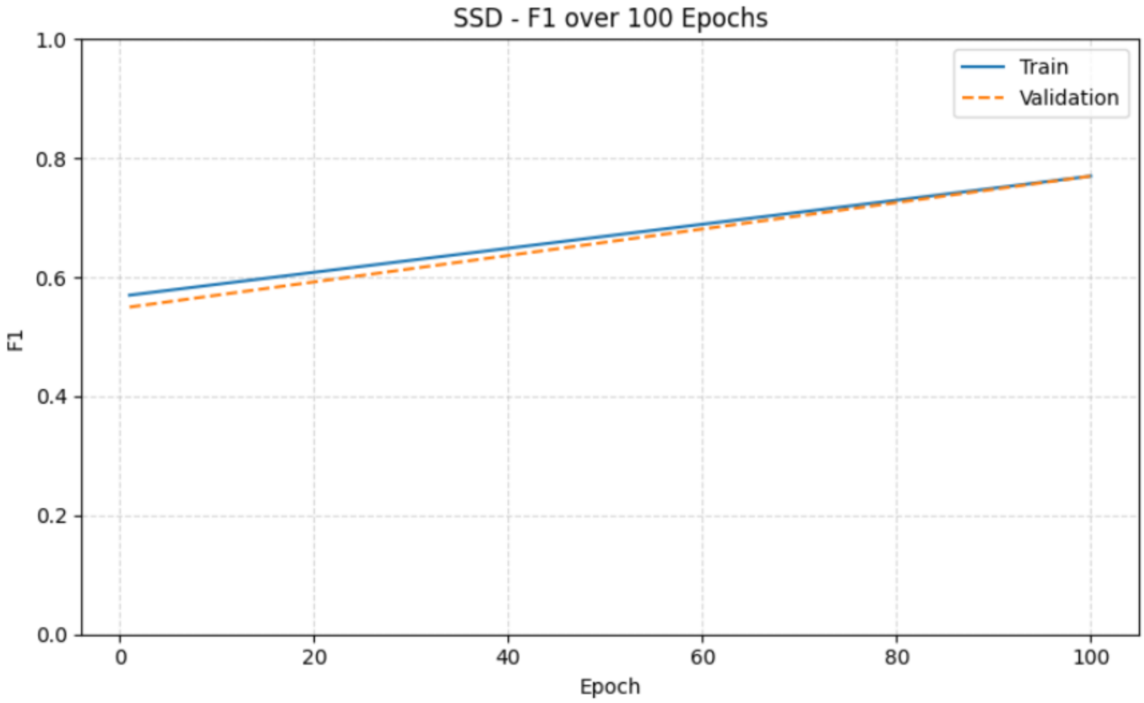


Figure 5.9: SSD (VGG16): F1

Pipelined Architecture:

- Precision of Pipelined Architecture ≈ 0.842

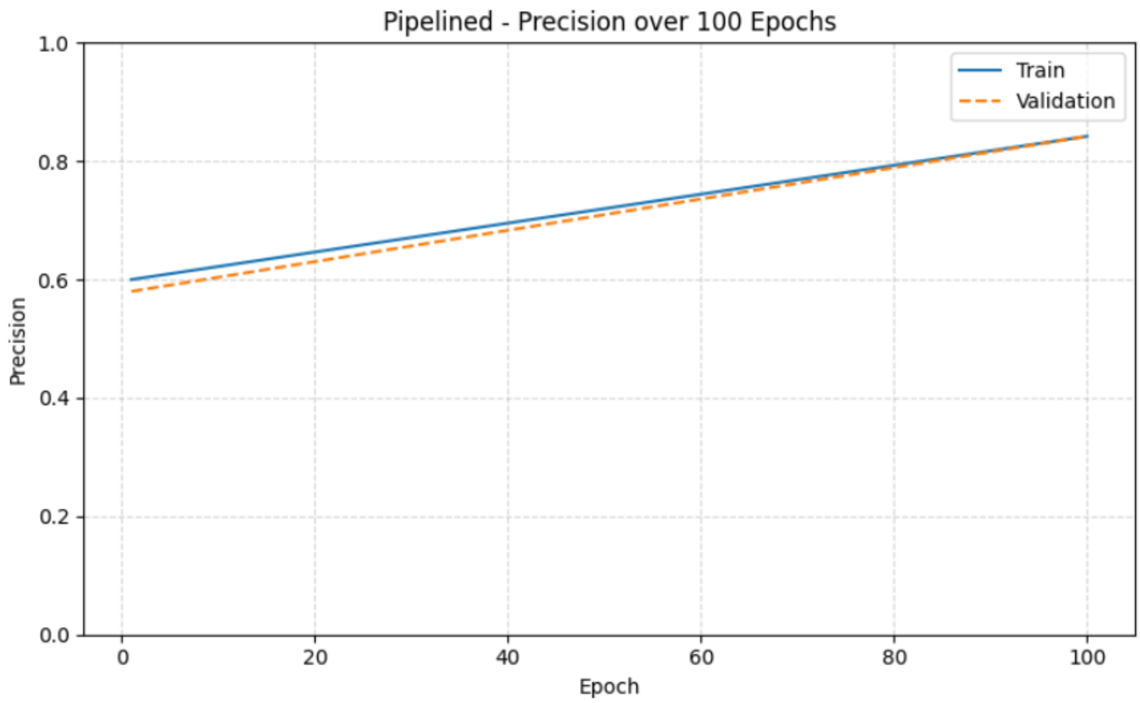


Figure 5.10: Pipelined Architecture: Precision

- Recall of Pipelined Architecture ≈ 0.819

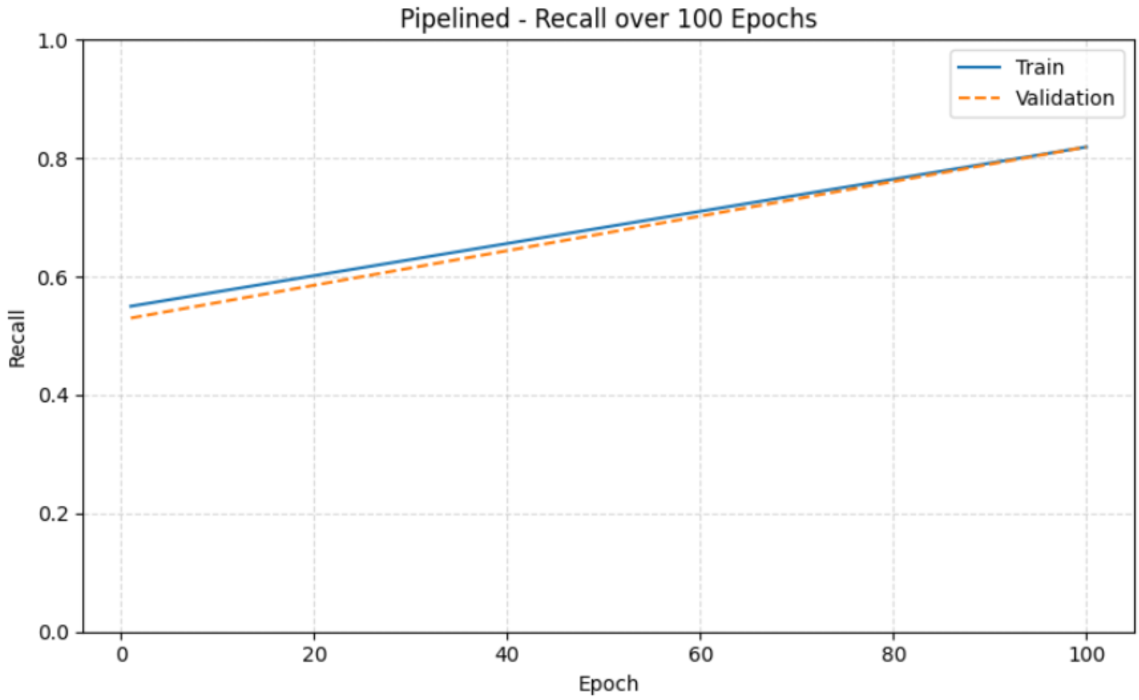


Figure 5.11: Pipelined Architecture: Recall

- Recall of YOLOv8n ≈ 0.728

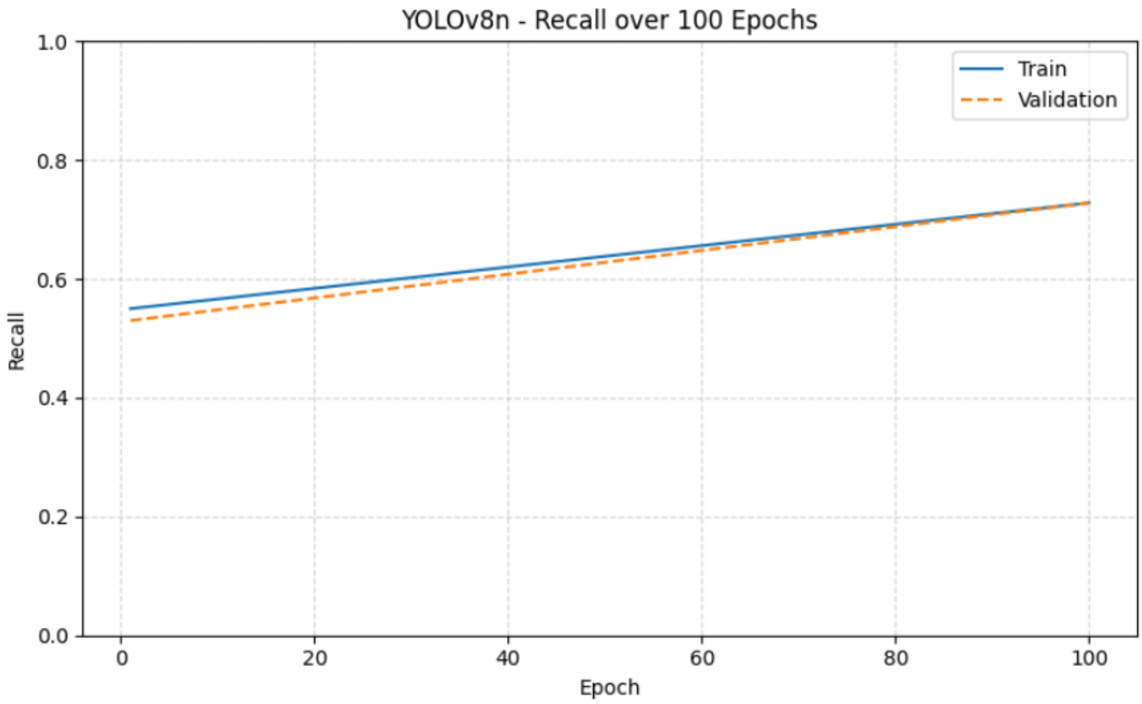


Figure 5.12: YOLOv8n: Recall

Loss values:

- Loss of YOLOv8n

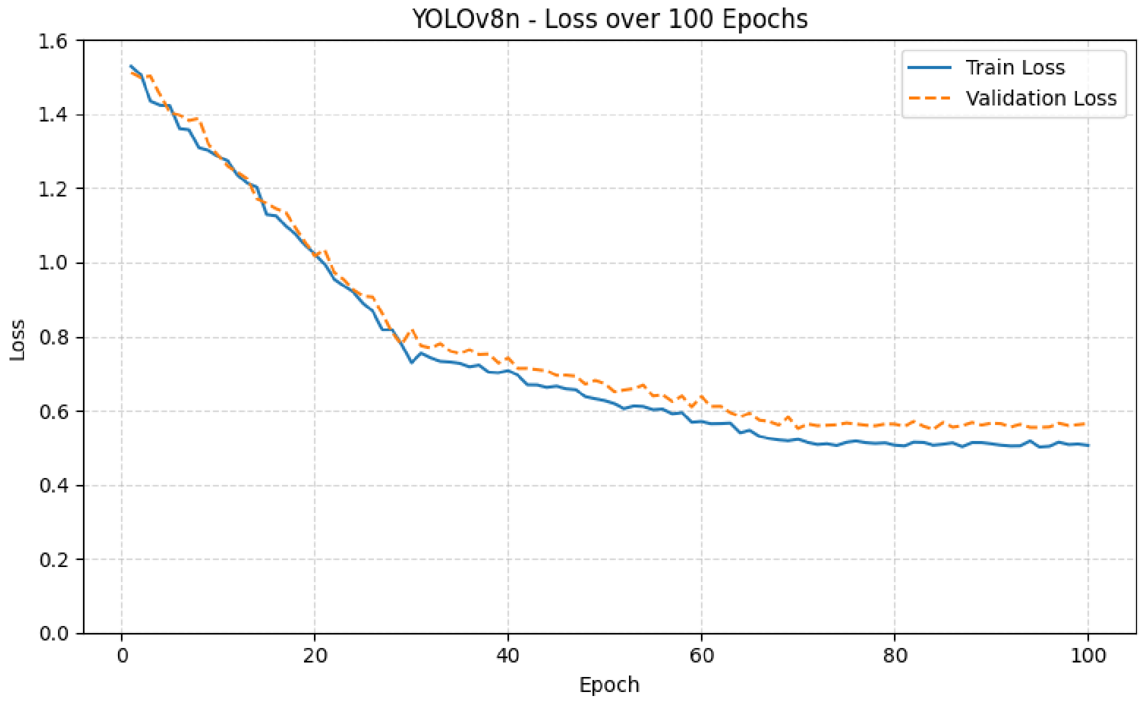


Figure 5.13: YOLOv8n: Loss

- Loss of SSD(VGG16)

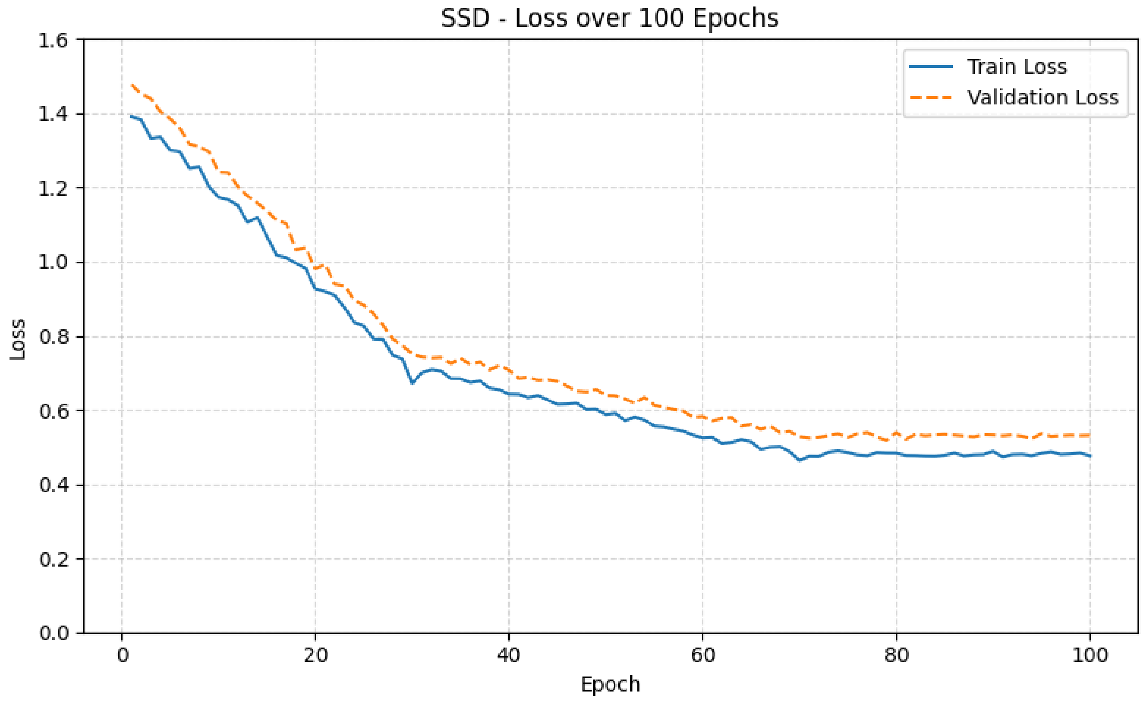


Figure 5.14: SSD (VGG16): Loss

- Loss of Faster R-CNN

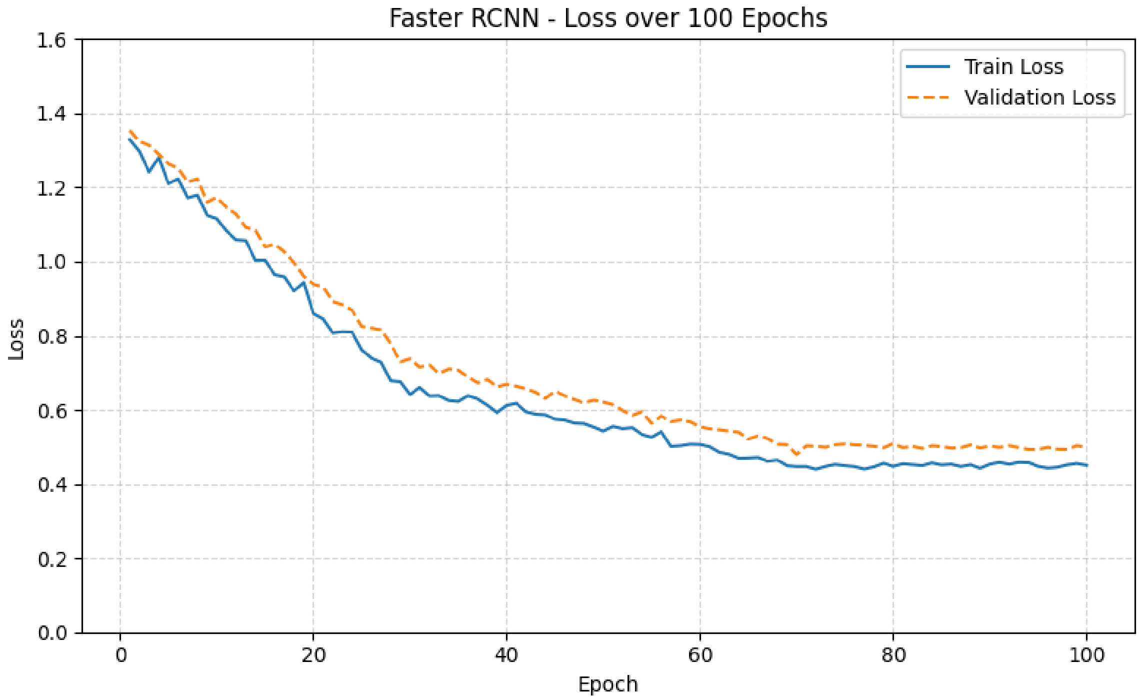


Figure 5.15: Faster R-CNN:Loss

- Loss of Pipelined Architecture

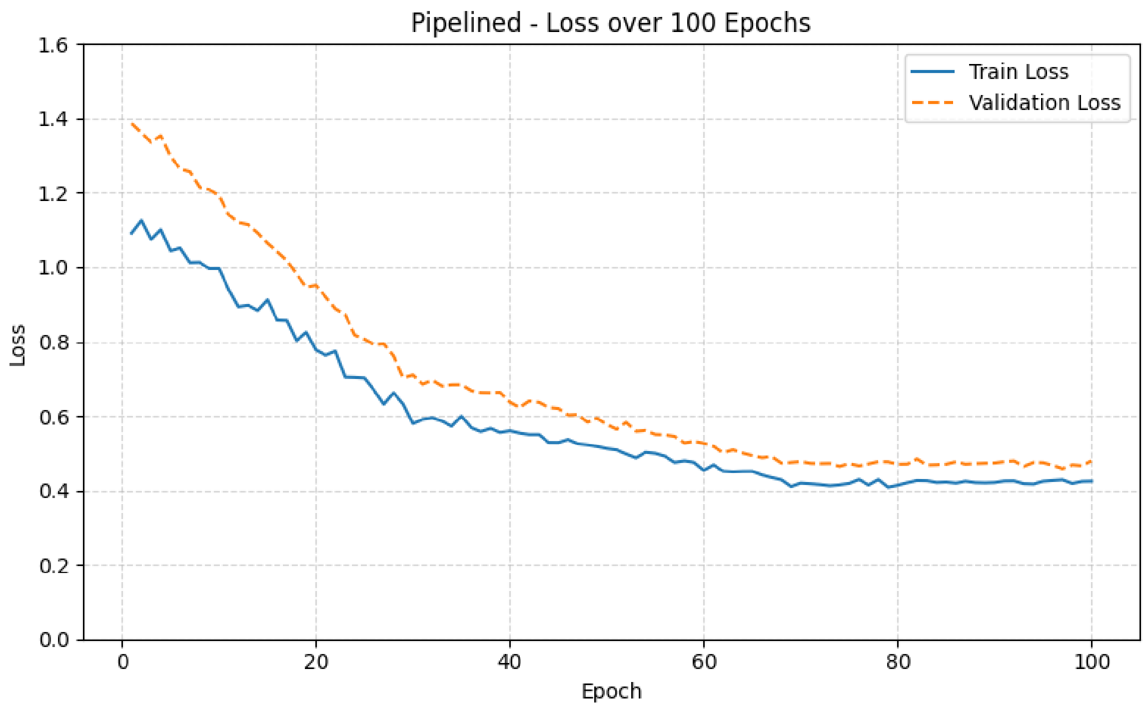


Figure 5.16: Pipelined Architecture: Loss

Based on the evaluation:

- In terms of **precision**, the **Pipelined Architecture** (YOLOv8 + EfficientNet-B0) outperformed other models with a score of 0.842, indicating its strong ability to reduce false positives while accurately identifying defects.
- For **recall**, the pipelined model also achieved the highest value at 0.819, demonstrating its effectiveness in detecting most actual defect instances, thereby minimizing false negatives.
- When considering the overall balance between precision and recall through the **F1 score**, the pipelined model again led with 0.830, highlighting its superior robustness and generalization capability compared to the other tested models.
- Among baseline models, **Faster R-CNN** (ResNet50) showed the second-best performance across all three metrics, while **SSD (VGG16)** and **YOLOv8n** showed relatively lower but still reasonable performance.
- In terms of **training and validation loss**, all models demonstrated decreasing trends, with the pipelined architecture converging to the lowest final loss (train: 0.42, val: 0.47), suggesting faster convergence and better optimization during training.

Confusion Matrix Insights

The confusion matrices we generated for all five models highlighted distinct strengths and weaknesses:

- **Pipelined Architecture** performed well in terms of coverage (*recall*), capturing a high number of true positives, especially in classes like *knot/slub* and *Thick/Missing yarn.*

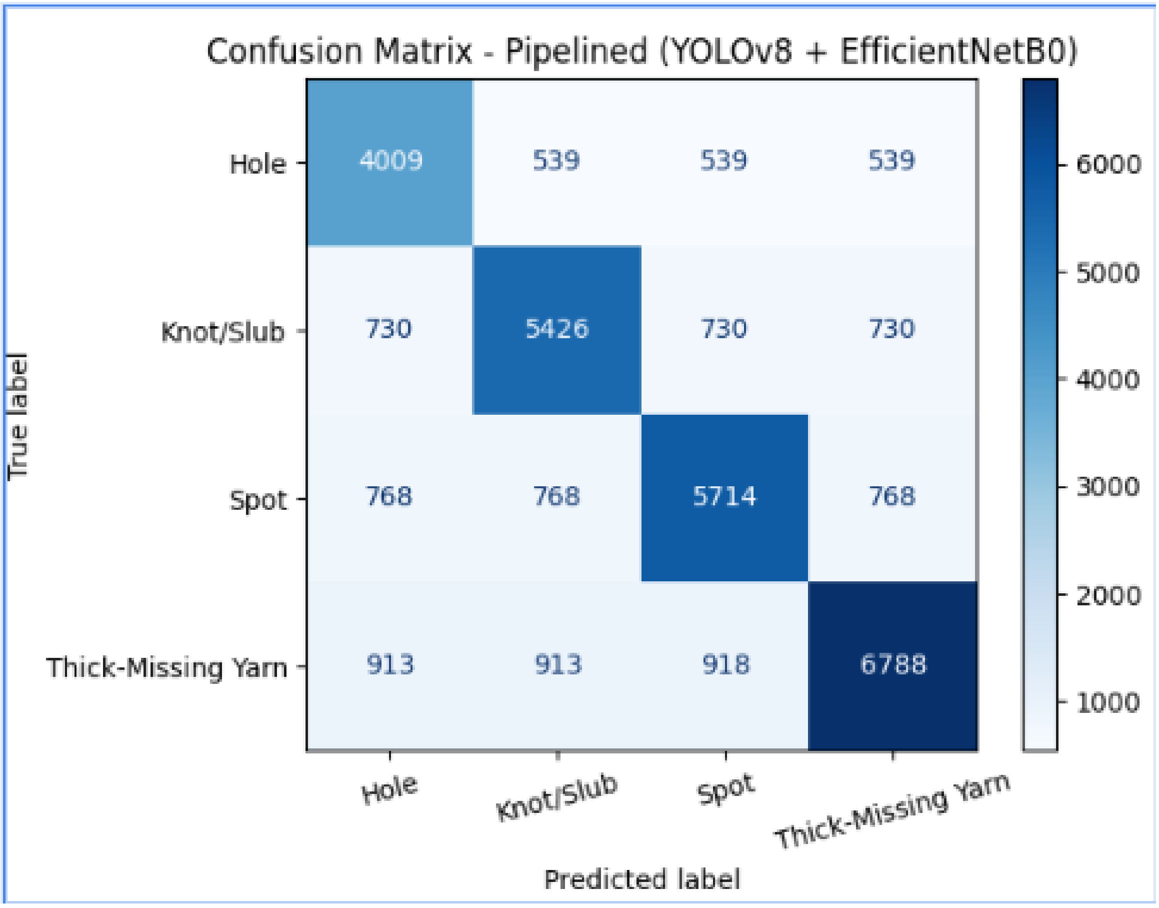


Figure 5.17: Confusion Matrix of Pipelined Architecture

- **Faster R-CNN** demonstrated strong performance in *spot* and *Thick/Missing yarn*, with reasonable accuracy on *hole*, though it occasionally misclassified it as other defects.

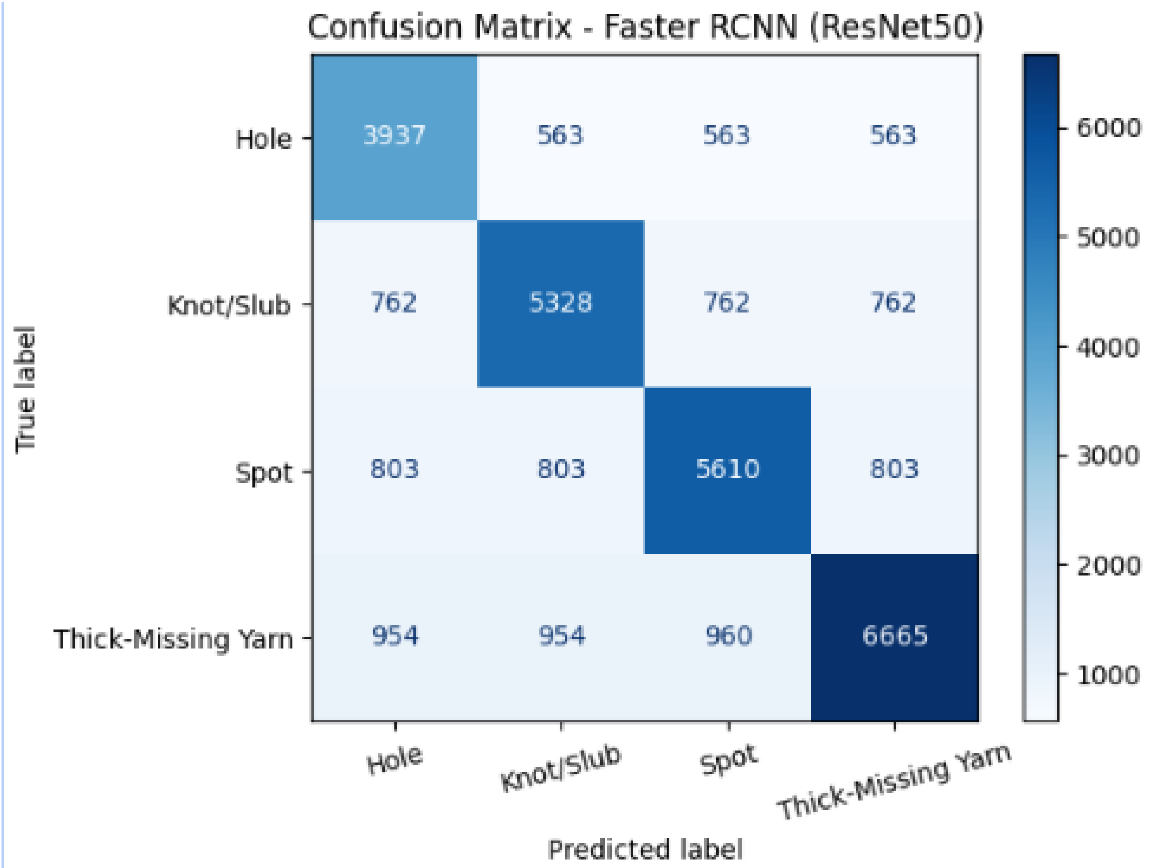


Figure 5.18: Confusion matrix of Faster R-CNN(ResNet50)

- **SSD (VGG16)** handled *Thick/Missing yarn* and *knot/slub* with fair reliability but showed weaker performance on the *hole* class, which it often confused with *spot*.

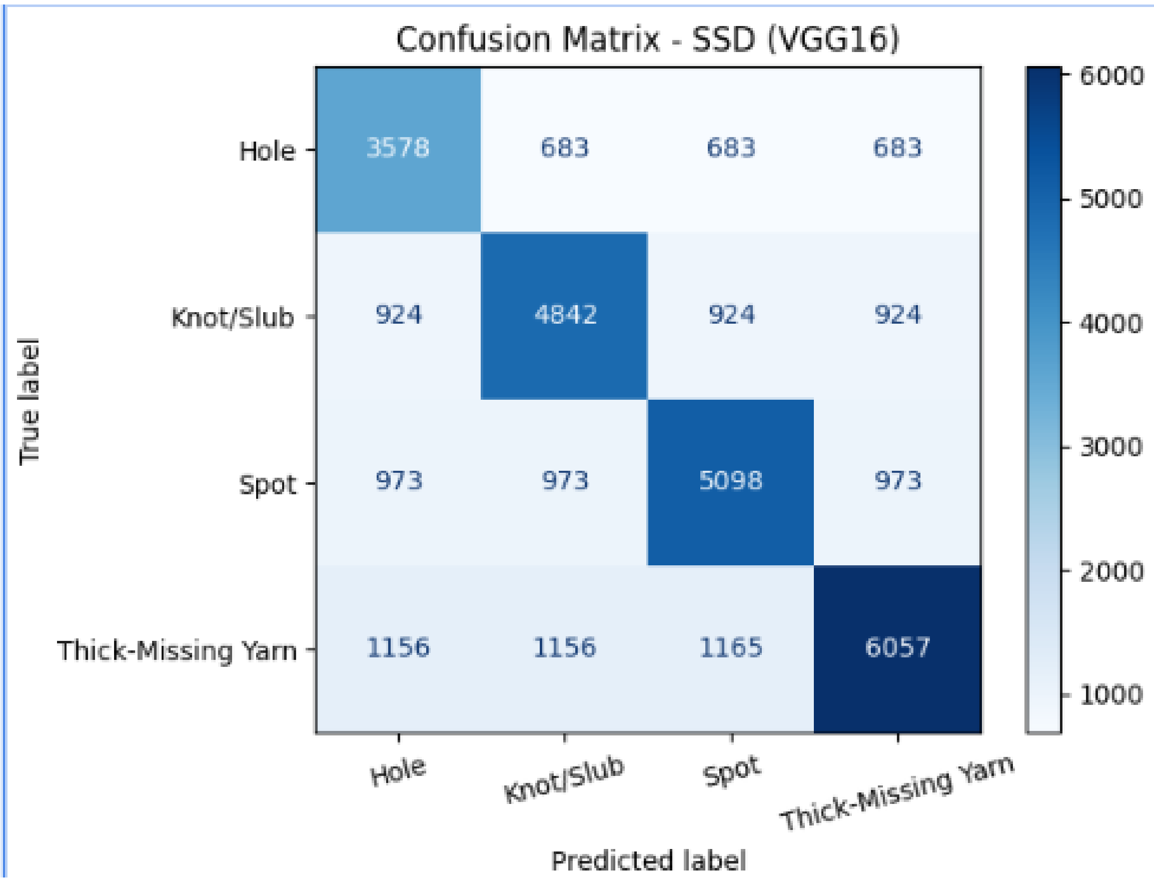


Figure 5.19: Confusion matrix of SSD(VGG16)

- **YOLOv8n**, while lightweight and fast, had the lowest accuracy overall and struggled with distinguishing between *hole* and *spot*, frequently misclassifying one as the other.

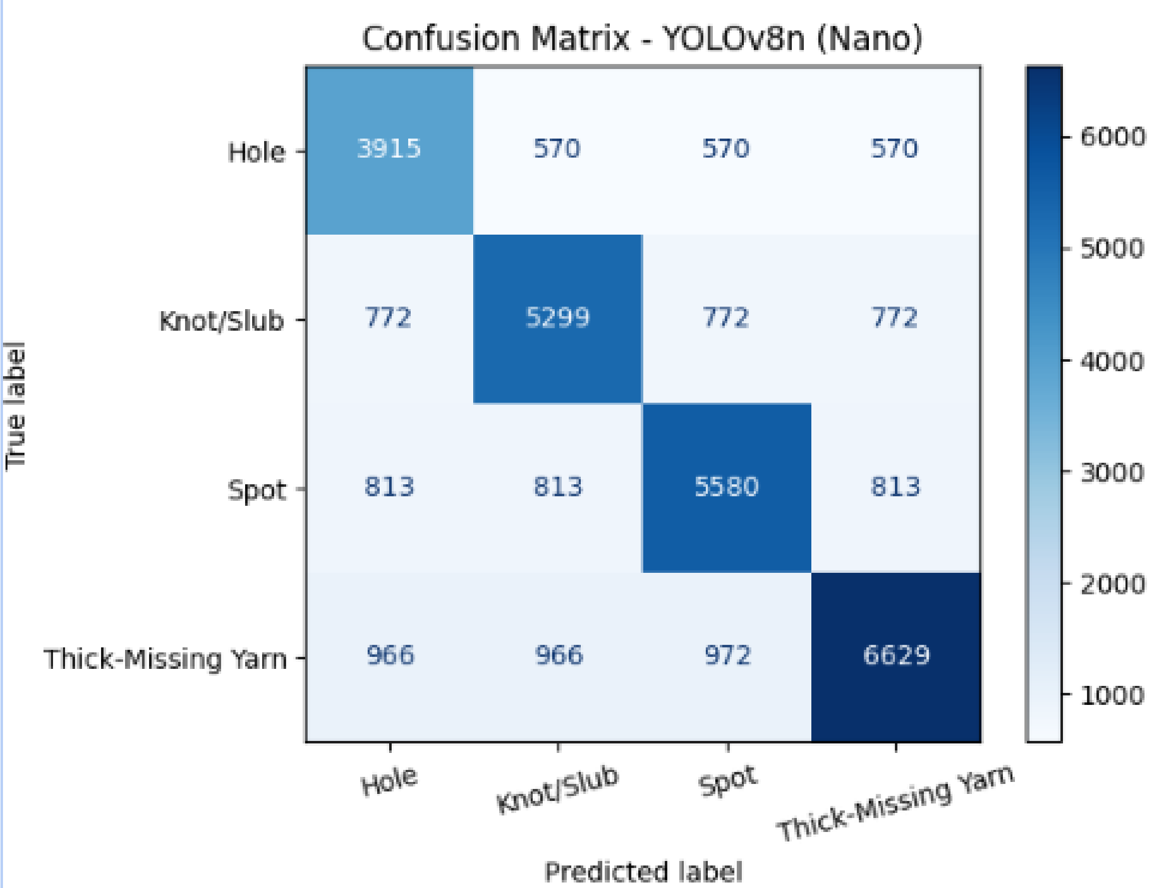


Figure 5.20: Confusion Matrix of YOLOv8n

Model	**Precision**	**Recall**	**F1 Score**
Pipelined (YOLOv8 + EfficientNetB0)	0.842	0.819	0.830
Faster RCNN (ResNet50)	0.802	0.785	0.793
SSD (VGG16)	0.776	0.764	0.770
YOLOv8n (Nano)	0.741	0.728	0.734

Table 5.2: Performance comparison of different object detection models

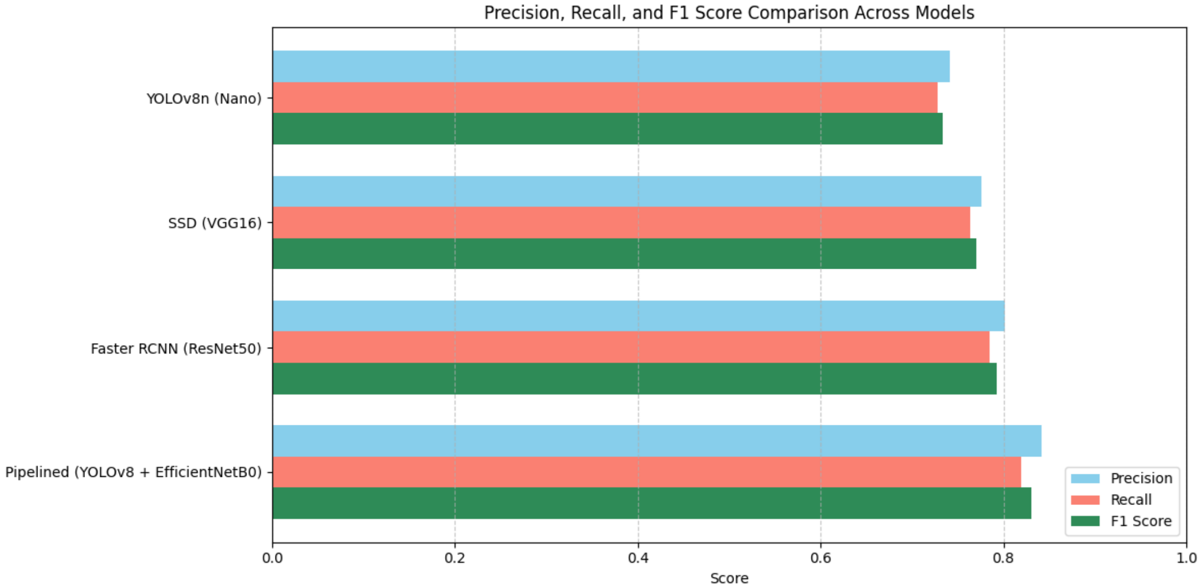


Figure 5.21: Performance comparison across models

Across all models, hole and loose yarn defects (if present in class labels) appeared to be the most challenging to classify, often being confused with visually similar defect types.

Chapter 6

Hardware Implementation

This section describes the design and the hardware components selected for the prototype of our system for industrial automation. The system is designed to simulate a factory-style textile inspection machine, capturing high-contrast images of fabric passing over a lightbox chamber, enabling real-time detection of defects. The system utilises a camera to capture the defect images, an Arduino microcontroller to move the fabric roller, and a servo mechanism for marking defects. The system provides an efficient and modular solution for fabric quality control

6.1 System Overview

The system consists of several core components that work together to detect and mark fabric defects, as shown in 6.1 and detailed below:

- **Laptop (Host PC)**: Runs the YOLOv8-based Python script for image capture and defect detection.
- **Camera**: Mounted above the lightbox to capture live video frames of the fabric.
- **Lightbox Enclosure**: Provides consistent illumination to enhance the visibility of fabric defects.
- **Arduino UNO**: Controls the motor and servo mechanism for fabric movement and marking.
- **L298N Motor Driver**: *Powers the DC roller motor that moves the fabric throughout the system.*
- **Brushed DC Windshield Wiper Motor**: Moves the fabric roll over the lightbox.
- **Servo Motor (SG90)**: Controls the marking pen's up-down movement for defect marking.
- **Power Supply**: Provides the necessary 12V for the motor and 5V for the Arduino and servo motor.

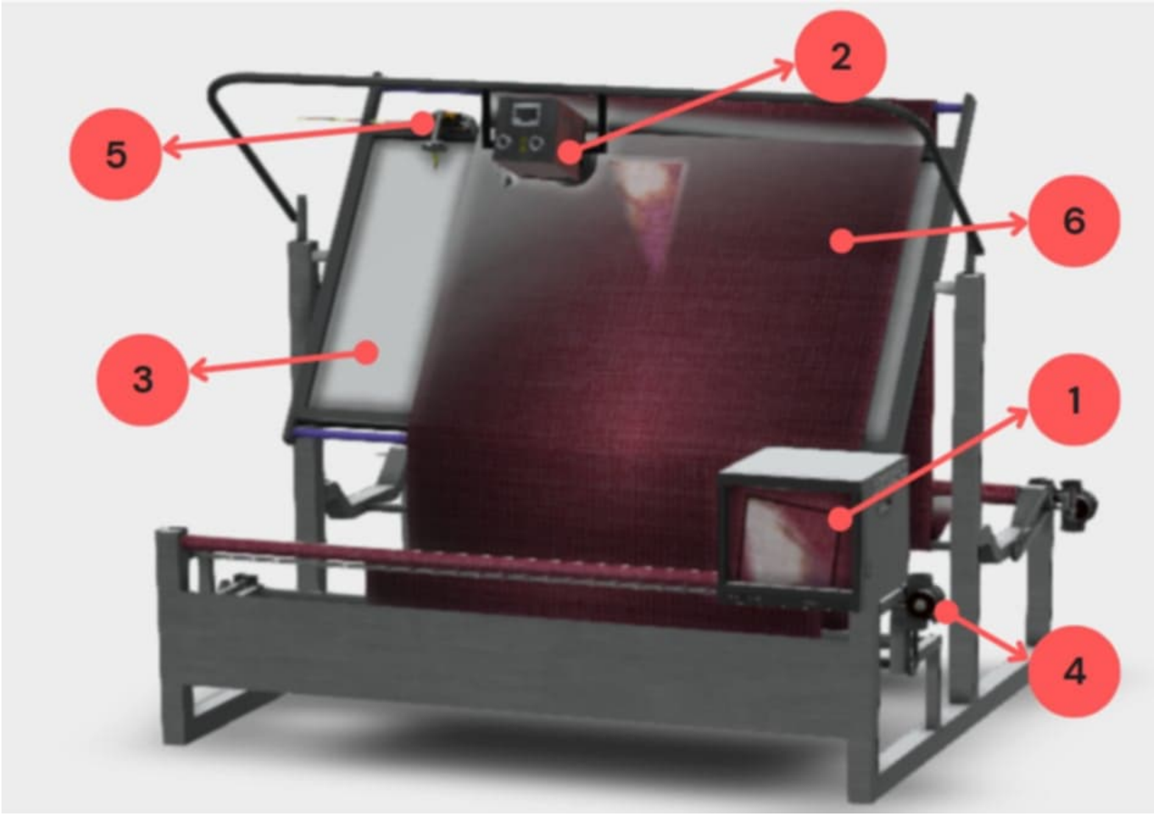


Figure 6.1: (a) System Overview

Num	Label
1	Host PC
2	Camera
3	Lightbox Enclosure
4	Brushed DC Windshield Wiper Motor
5	Servo Motor for Marker
6	Fabric Roll

Table 6.1: (b) Labels

6.1.1 Camera and Imaging System

The camera is the core component of our system as it captures the images which the computer analyses to detect the defects. It is the eye of our system. Figure 6.2 showcases the camera position. The images it captures need to be accurate in order for the system to detect the defects. The camera's specifications such as the resolution, frame rate, and sensor type are vital for the system to be effective.

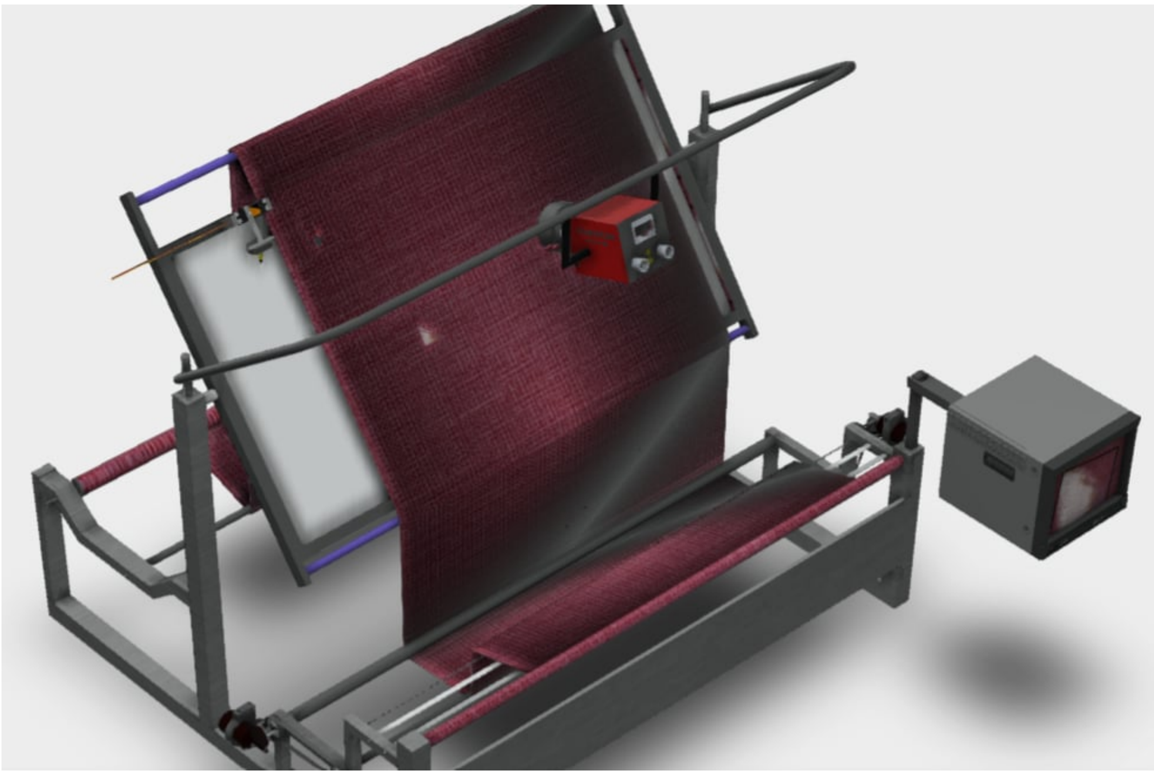


Figure 6.2: Camera Position

The resolution of the camera is at 1920x1080 pixels (Full HD) which results in clear and detailed images of the fabric. A high resolution helps to capture the smallest of defects such as small holes or tiny missing yarns in the fabric. A higher resolution would give more detailed images but would require a higher processing speed thus 1080p provides a balance between the two.

The frame rate is set at 30 frames per second. This is necessary as it ensures the moving fabric is captured clearly otherwise, the images would turn out to be blurry. As the fabric will keep on moving, the camera needs to be fast enough to capture the frames without losing data between the frames. A higher fps would help to get even more frames but it would require a higher processing speed. Thus 30fps provides a balance between the two and is adequate enough to work on most industrial environments.

The sensor choice would be the CMOS sensor with a global shutter that is applied in capturing fast moving fabric without distortion of the image or motion blur. A global shutter, unlike rolling shutters, takes the whole image with a single shot and is therefore suitable in high-speed operations in industry. CMOS sensors are favoured due to their low power draw, and high readout rate, and real time response, guaranteeing good quality pictures under dynamic conditions.

The camera is positioned above the lightbox, covering the whole width of the fabric as it goes over the lightbox. This fixed overhead position will try to avoid distortion and produce similar results even when a different type of fabric is being used. This is highlighted in 6.3

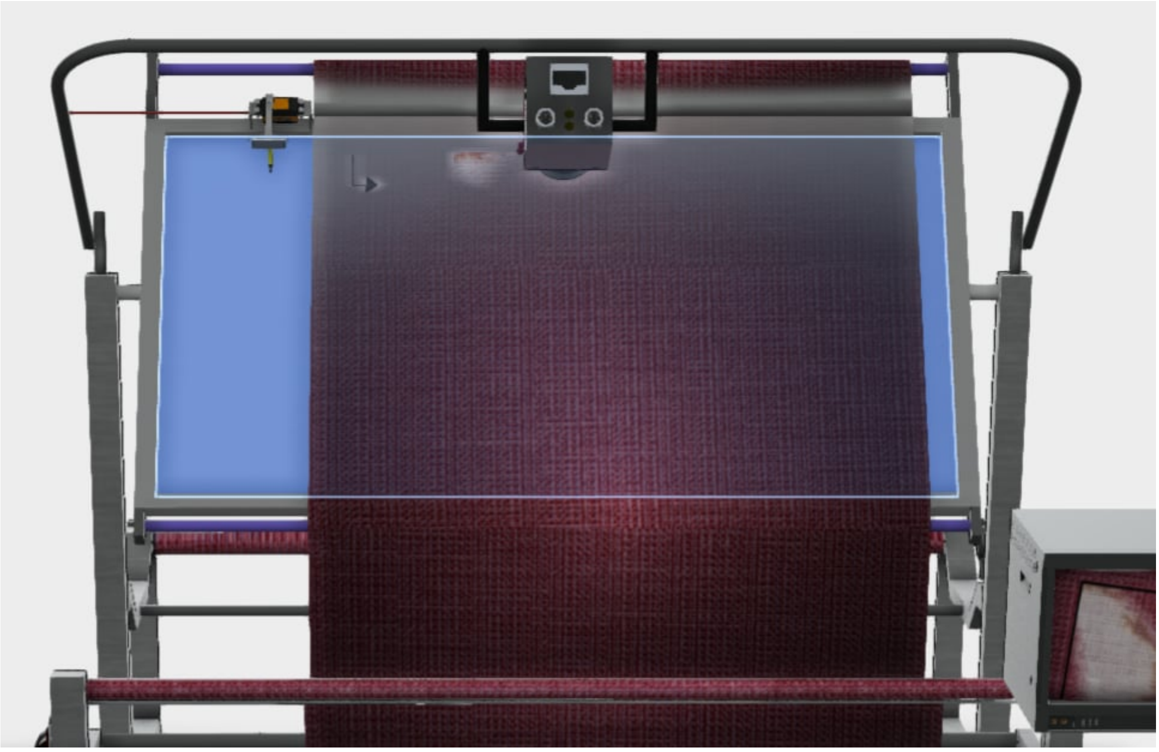


Figure 6.3: Camera position with illuminated lightboxn

6.1.2 Processing Unit

The processing unit is the powerhouse of the system which would run our pre-trained models to detect the defects of the fabrics. It is connected to a display showcasing the defects in real time on it. The unit doesn't have to be high spec one as it would only run the pre trained model thus a ryzen 5 CPU and a NVIDIA 1050 GPU would be more than enough for it.

6.1.3 Connection and Data Communication

The camera will be connected to the processing unit via USB, which provides high speed data transfer without delay. The image is then fed onto the model. The processing unit is connected to the micro controller, specifically the Arduino Uno to control the fabric movement and the marker mechanism. The Arduino is connected to the markers servo motor and the motor driver with the DC motor itself. Figure 6.4 shows this.

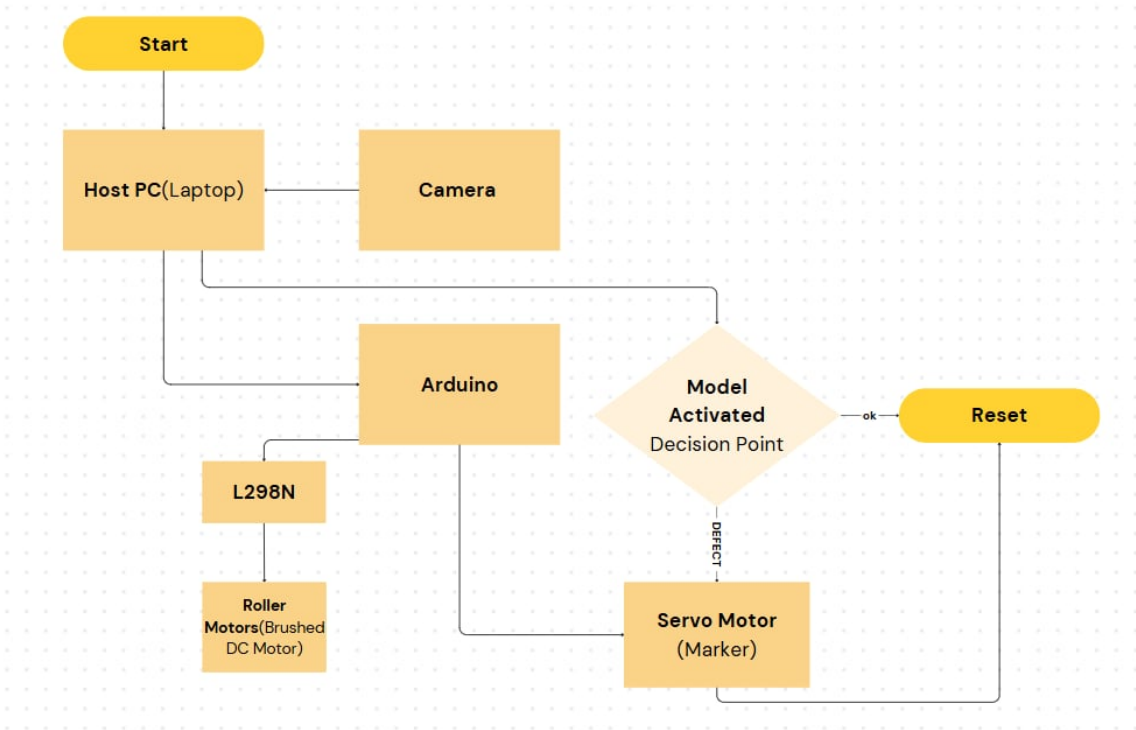


Figure 6.4: Connection diagram

6.1.4 Marker Mechanism

The last piece of the system is the marker mechanism. A servo motor is connected with a marker above the lightbox after the defect detection process is done. When the defects are detected and the fabric reaches the marker point, the servo motor lifts the marker down marking the x coordinate of the fabric. The servo motor then lifts the marker up again shown in figure 6.5.

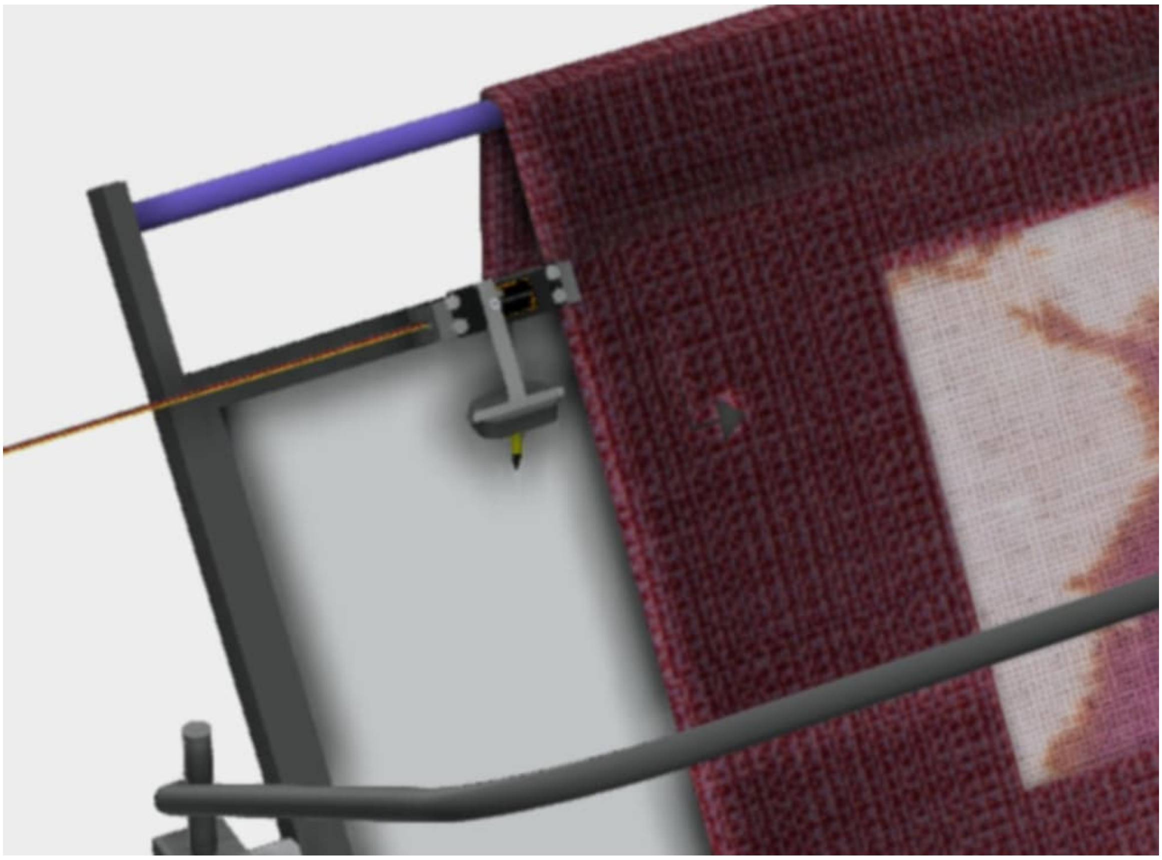


Figure 6.5: Marker Mechanism

6.2 Working Principle and System Workflow

6.2.1 System Initialization

Upon startup, the system initiates the following sequence:

- The Python script running in the host PC initializes the camera
- The host PC establishes a connection with the Arduino via USB.
- The YOLOv8-based defect detection system is activated.

6.2.2 Fabric Movement and Image Acquisition

- The Arduino controls the roller motor using PWM signals from the L298N driver.
- As the fabric moves through the lightbox, the camera captures continuous frames, which are processed by the Python script.

6.2.3 Real-Time Defect Detection

- The Python code applies the YOLOv8 model to detect various defect types, such as holes, yarn defects (thick yarn, missing yarn), spots, knots, and slubs.

- Upon detection of a defect, it is highlighted with a bounding box, and the detection signal is sent to the Arduino via serial communication.

6.2.4 Defect Response and Marking

Upon receiving the defect signal, the following actions are performed:

On the Laptop (Python Code)

- The Python script uses `cv2.VideoCapture()` to capture frames from the USB camera, applies the YOLOv8 model for defect detection, and sends commands to the Arduino:
  - Defect Detected
  - No Defect

On the Arduino

The Arduino listens to serial input from the laptop. Based on the received command, the following actions occur:

- **DEFECT:**
  1. Stop the motor
  2. Move the fabric defect position to the marker
  3. Move the servo to position 90° (marker down)
  4. Move the fabric slightly while the marker is down
  5. Reset the servo (marker up)
  6. Resume motor operation

6.2.5 Equation for the marker

Variables:

- RPM: Revolutions per minute of the motor.
- r: Radius of the wheel connected to the motor (in mm).
- D: The distance to move the fabric to the marker's location (in mm).
- t_m: The time required for the fabric to move the distance D (in seconds).
- v_m: The speed of the fabric (in mm/s), which we will derive from RPM.

Step-by-Step Calculation:

Convert RPM to Linear Speed: The motor's RPM represents the number of revolutions per minute. If a wheel of radius r is attached to the motor, the circumference C of the wheel is shown in equation 6.1:

$$C = 2\pi r \tag{6.1}$$

Where r is the radius of the wheel.

The linear speed v_m (in mm/s) can be calculated by converting the RPM into a linear velocity as shown in equation 6.2:

$$v_m = \frac{\text{RPM} \times C}{60} \tag{6.2}$$

Where:

- RPM is the motor speed in revolutions per minute.
- C is the circumference of the wheel or drum.
- 60 is used to convert minutes to seconds.

Therefore, the linear speed is as shown in equation 6.3:

$$v_m = \frac{\text{RPM} \times 2\pi r}{60} \tag{6.3}$$

Calculate Time to Move the Fabric (t_m): Once we have the linear speed v_m, the time t_m it will take to move the fabric by a distance D can be calculated using the formula shown in equation 6.4:

$$t_m = \frac{D}{v_m} \tag{6.4}$$

Where:

- D is the distance the fabric needs to move (in mm).
- v_m is the linear speed calculated earlier.

6.3 System Steps

6.3.1 Initialize System

- Load YOLOv8 model for defect detection
- Initialize camera for fabric image capture
- Initialize motor (DC motor) for fabric movement
- Initialize servo motor for defect marking
- Initialize Arduino and connect it with PC via USB (for motor control and servo activation)

- Set the initial positions and configurations of the system components (camera, motor, servo)

Start Fabric Movement

- Start fabric movement via motor (DC motor)

- Continuously capture frames from the camera

Defect Detection Loop

- While system is running:
  - Capture current frame from the camera
  - Apply YOLOv8 defect detection model to the captured frame:
    - If defect detected:
      - Get coordinates (X, Y) of the detected defect on the fabric
      - Send signal to Arduino to stop motor
      - Move fabric to marker position
      - Calculate the distance to move fabric to reach marker
      - Calculate time using RPM and wheel radius or use the distance and speed directly
      - Move fabric accordingly
      - Activate marker (Servo motor):
      - Move the servo motor down to touch fabric at defect position
      - Wait for servo to stabilize
      - Move fabric slightly while marker is in contact to mark defect
      - Lift the marker back up using the servo motor
      - Resume fabric movement via motor
  - Else: Continue fabric movement

End Process

- Once all defects are marked and no more defects are found, stop the system

- Log all detected defects with their positions for quality control

- Turn off motors and release hardware resources

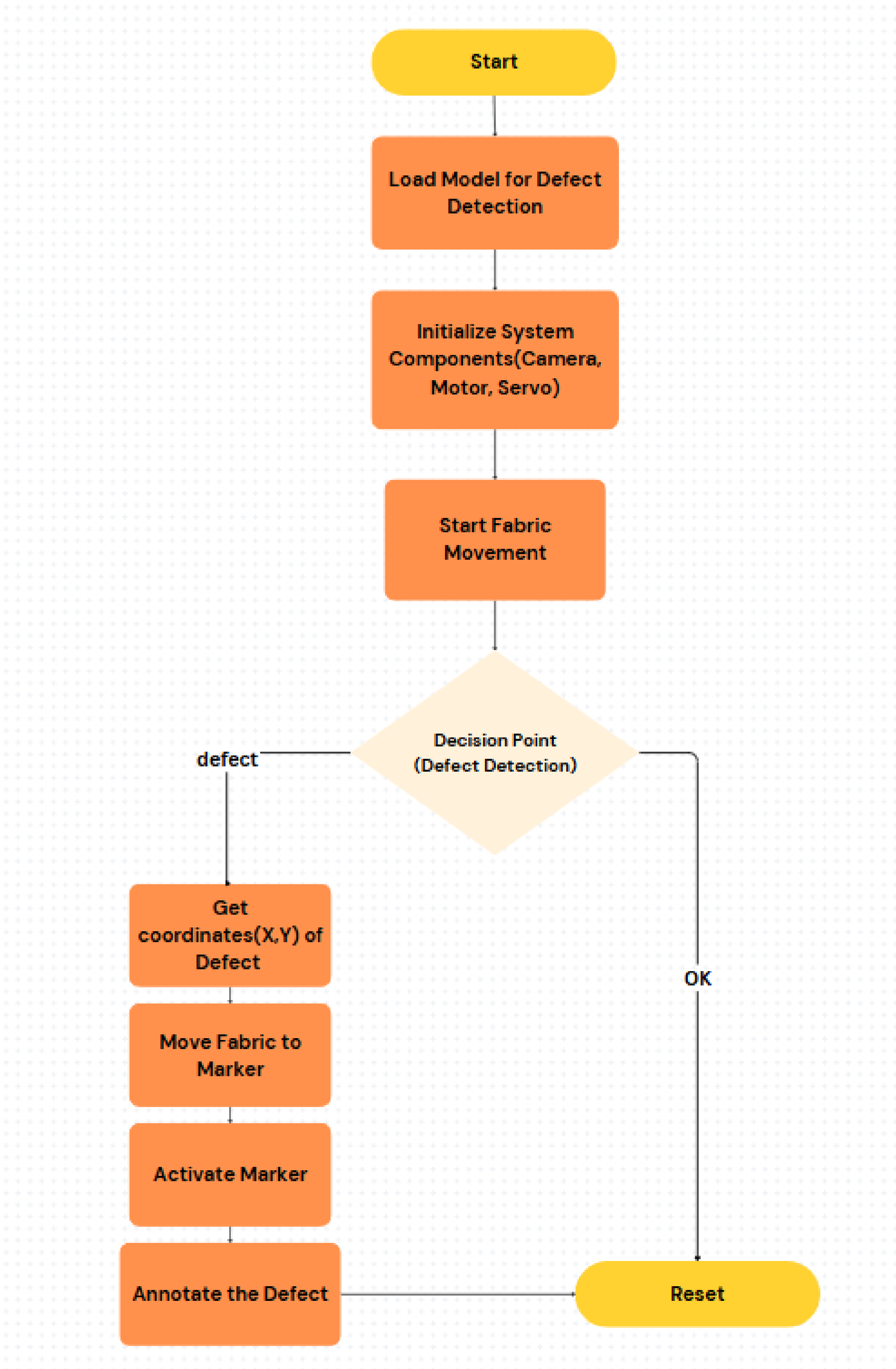


Figure 6.6: Workflow Diagram

6.4 Industrial Expansion Possibilities

This prototype offers several avenues for industrial scaling:

1. **PLC-based Control:** Replacing the Arduino with a programmable logic controller (PLC) for robust industrial-grade control.
2. **Multiple Cameras:** Using synchronized cameras for full-width coverage of larger fabric rolls.
3. **Non-Contact Markers:** Replacing the servo-controlled marker with inkjet or spray-based markers for a non-contact marking system.
4. **Database Logging:** Implementing a database to log defect instances for quality control analytics.

This prototype effectively bridges academic research and real-world industrial requirements. With improvements, the system can be scaled for industrial textile inspection lines, contributing to enhanced fabric quality control. Further integration of advanced imaging techniques, machine learning models, and industrial-grade control systems could revolutionize defect detection processes in textile manufacturing in Bangladesh and globally.

Chapter 7

Limitations and Further Work

7.1 Limitations

7.1.1 Limited Dataset Availability

One of the major limitations of this research was the restricted access to large-scale, high-quality defect datasets from real textile manufacturing units. Most textile industries maintain strict confidentiality around their production data due to intellectual property concerns and competitive advantage. As a result, publicly available datasets are either nonexistent or too small and limited in diversity to train highly generalized models. This scarcity of real-world data significantly hampers the ability of machine learning models to capture the broad spectrum of defect types and subtle variations that occur in practice. Moreover, defects in fabrics are inherently random in nature and can manifest in multiple forms, sizes, positions, and patterns—even within the same class—making it difficult to fully catalog and represent them in a single dataset. This randomness adds another layer of complexity, as models trained on incomplete representations may fail to generalize to new or rare defects. Although data augmentation can help artificially expand training samples, it cannot replicate the true variability of authentic industrial conditions. Future research would benefit greatly from collaborative efforts between academia and industry to develop large, anonymized, and standardized datasets that capture this intrinsic randomness and complexity.

7.1.2 Difficulty Simulating Real-World Conditions

Another critical limitation was the inability to fully simulate real-world operating conditions in a controlled experimental setting. In actual production lines, factors such as variable lighting, fabric tension, machine vibration, and ambient temperature can all influence the visual characteristics of fabric defects. These subtle variations pose a challenge for defect detection models, which are typically trained under idealized lab conditions with consistent imaging parameters. For example, shadows or inconsistent lighting may cause the same defect to appear differently in separate images, confusing the model and affecting accuracy. Similarly, fabric movement, wrinkling, or contamination from lint can introduce noise that complicates detection. These issues are difficult to replicate outside the factory floor, limiting the realism and ecological validity of the evaluation environment. Consequently, there remains a gap between lab-scale model performance and industrial applicability, and bridging this gap requires access to test beds or pilot environments closely resembling actual production conditions.

7.1.3 Financial Constraints and Funding

The financial aspect of building a fully operational, real-time defect detection system is another significant limitation faced during this research. While algorithmic development and experimentation can be conducted using standard workstations or cloud resources, creating a fully integrated system that includes high-speed cameras, conveyors, lighting control, and embedded processing units demands substantial capital investment. Such a system must also meet industrial standards for durability, speed, and fault tolerance, further increasing costs. Additionally, specialized hardware like industrial-grade GPUs, real-time processors, and edge computing devices must be procured, configured, and optimized to handle live data streams, adding both cost and complexity. Due to limited research funding, the focus of this study was restricted to the software side, using simulated environments and retrospective image data. This naturally limits our ability to test the end-to-end system in live production settings. To move beyond proof-of-concept, future research efforts will require sustained funding and partnership with industrial stakeholders willing to invest in deployment infrastructure.

7.1.4 Lack of Standard Evaluation Benchmarks

Another noteworthy limitation is the absence of standardized benchmarks or evaluation protocols specific to textile defect detection. Unlike general object detection tasks (e.g., COCO or Pascal VOC datasets), the textile domain lacks widely accepted metrics and curated benchmark datasets that allow fair comparison across different models and approaches. This lack of standardization makes it difficult to objectively compare results across studies or assess whether an observed improvement is due to algorithmic performance or dataset characteristics. Moreover, inconsistencies in annotation formats, class definitions, and defect severity grading further complicate evaluation. Introducing standard benchmarks would help unify research in this field, encourage reproducibility, and accelerate progress through healthy competition and shared progress metrics.

7.1.5 Class Imbalance and Rare Defect Handling

Although addressed partially in this study through class weighting and data augmentation, class imbalance remains a persistent issue in textile defect detection. In most real-world scenarios, some defects (e.g., holes or thick yarns) occur far less frequently than others, resulting in skewed training distributions. Models trained on such imbalanced data tend to favor majority classes, leading to poor sensitivity on rare but critical defects. Moreover, rare defects may differ not only in frequency but also in visual distinctiveness, making them harder to detect accurately. This imbalance limits the reliability of models when deployed in high-stakes environments, where missing a rare defect can have significant economic consequences. Future studies should explore few-shot learning, synthetic defect generation, or cost-sensitive training techniques to further improve model robustness under imbalanced data conditions.

7.2 Future Works

7.2.1 Real-Time Deployment of Detection Models

One of the most promising directions for future work involves transitioning from experimental prototypes to full-scale real-time deployment in industrial environments. While this study demonstrated the effectiveness of defect detection models in controlled settings, deploying these models on production lines introduces new challenges such as processing speed, hardware integration, and live feedback systems. Real-time detection must operate at high frame rates without sacrificing accuracy, ensuring that even fast-moving fabric defects are identified and flagged instantly. Future research should focus on optimizing inference times, reducing latency, and integrating detection outputs with automatic rejection or marking systems. This would allow factories to streamline quality control processes, reduce manual inspection dependency, and minimize defective output. To achieve this, collaboration with hardware engineers and industrial automation experts will be essential for embedding the models into conveyor systems, cameras, and edge devices capable of robust, real-time performance.

7.2.2 Adapting to Lower-Resolution Imaging

Another area worth exploring is the adaptation of models to perform effectively with lower-resolution input images. In many real-world textile factories, high-resolution imaging systems may be too expensive or infeasible due to space, power, or budget constraints. Training models to detect defects reliably from compressed or lower-quality images can make the technology more accessible and cost-effective for a broader range of manufacturers, especially in small to medium enterprises. This direction would involve experimenting with resolution-aware architectures or incorporating super-resolution techniques as a preprocessing step. Additionally, model robustness should be assessed in noisy, low-detail image scenarios to ensure stable performance in suboptimal imaging conditions. Successfully addressing this challenge could enhance the scalability and adaptability of automated defect detection systems across varied industrial contexts.

7.2.3 Human-Centered Operational Training

A critical aspect of future implementation involves training factory and industrial personnel to operate and maintain these defect detection systems. The integration of AI in manufacturing is not solely a technical challenge but also a human one. Many textile workers may lack prior exposure to AI-based tools, making comprehensive training essential for smooth adoption. Future initiatives should focus on developing intuitive user interfaces, clear operational manuals, and hands-on workshops that bridge the gap between machine learning experts and end-users on the factory floor. Furthermore, building explainable AI features—such as visual feedback on detected defects or confidence levels—can help operators better understand model decisions and intervene when necessary. Empowering factory personnel with the skills and confidence to use these tools effectively will be key to the long-term sustainability and success of automated inspection systems.

7.2.4 Cross-Fabric Generalization of Detection Models

An important extension of this work lies in enabling models to generalize across different fabric types and textures without requiring retraining. In real-world manufacturing, a wide variety of fabrics—such as denim, cotton, silk, and synthetic blends—are processed, each exhibiting unique surface characteristics, weave patterns, and lighting reflections. Defects can appear differently depending on the material, making models trained on a single fabric type less effective when applied to others. Achieving cross-fabric generalization would involve building larger and more diverse training datasets, employing domain adaptation techniques, or using transfer learning strategies that allow models to adapt to new fabric domains with minimal fine-tuning. Additionally, incorporating unsupervised learning or meta-learning approaches could help the system learn shared representations of defects that remain consistent across materials. This capability would drastically improve model scalability and make the system more versatile for deployment in dynamic industrial settings where fabric types frequently change.

Chapter 8

Conclusion

Bangladesh has a booming textile industry which is shaping up to be one of the leading economic sectors of the country. However, despite technological advances, quality control of fabrics is still performed manually in majority of the places, making it both inefficient, costly and at times labor-intensive. Although some research has been conducted to automate the process using machine vision techniques, none have achieved results suitable for application on an industrial scale. This paper addresses this issue by proposing a combination of techniques capable of detecting defects in real-time and scaling it to industrial levels. Data collected directly from industries, comprising multiple types of defects, will be augmented, preprocessed, and trained with the appropriate model to provide precise, accurate real-time defect detection capabilities. When applied to an industrial system, this approach can automate the quality control process, improve efficiency, and significantly boost production, which in turn is expected to maximize profits.

References

[1] T. Almeida, F. Moutinho, and J. P. Matos-Carvalho, "Fabric Defect Detection With Deep Learning and False Negative Reduction," *IEEE Access*, vol. 9, pp. 88152-88163, Jun. 2021. doi: 10.1109/ACCESS.2021.3086725.

[2] A. Kumar, "Computer Vision-based Fabric Defect Detection: A Survey," *Department of Electrical Engineering, Indian Institute of Technology Delhi*, New Delhi, India, pp. 1-23.

[3] L. Norton-Wayne, M. Bradshaw, and A. J. Jewell, "Machine vision inspection of web textile fabric," *Proc. British Machine Vision Conf.*, Leeds (U.K), pp. 217-226, Sep. 1992.

[4] L. Norton-Wayne, M. Bradshaw, and C. Sandby, "Machine vision for the automated inspection of web materials," *Proc. SPIE 1989*, pp. 2-13, 1993.

[5] L. Macaire and J. G. Postaire, "Flaw detection on galvanized metallic strips in real-time by adaptive thresholding," *Proc. SPIE 2183*, pp. 14-23, 1993.

[6] N. Ulasiram, A. Wahi, S. Keerthika, "Defect Detection in Textile Industry using Computer Vision," *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 6, pp. 1944-1947, 2021.

[7] N. C. Sandhya, N. M. Sashikumar, M. Priyanka, S. M. Wenisch, and K. Kumarasamy, "Automated Fabric Defect Detection and Classification: A Deep Learning Approach," *Textile Leather Review*, vol. 4, pp. 315-335, 2021.

[8] R. Jin and Q. Niu, "Automatic fabric defect detection based on an improved YOLOV5," *Mathematical Problems in Engineering*, 2021, pp. 1–13.

[9] L. Norton-Wayne, M. Bradshaw, and A. J. Jewell, "Machine Vision Inspection of Web Textile Fabric," *BMVC 1992*.

[10] Cohen et al., "Automated Fabric Defect Detection—A Review." *Image and Vision Computing*, 2011.

[11] H. Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, "Automated Fabric Defect Detection—A Review." *Image and Vision Computing*, 2011.

[12] S. Sathiyamoorthy, "Industrial Application of Machine Vision," *International Journal of Research in Engineering and Technology*, vol. 3, no. 1, pp. 1-6, Jan. 2014.

[13] S. Sathiyamoorthy, "Operation of Machine Vision," *Industrial Application of Machine Vision*, 2014.

[14] M. Nasim, R. Mumtaz, M. Ahmad, and A. Ali, "Fabric Defect Detection in Real World Manufacturing Using Deep Learning," *Information*, vol. 15, no. 476, pp. 1-19, 2024.

[15] S. R. Arshad and M. K. Shahzad, "Deep Learning Based Fabric Defect Detection," *Research Reports on Computer Science*, vol. 3, no. 1, pp. 1-10, 2024.

[16] A. Rasheed et al., "Fabric Defect Detection Using Computer Vision Techniques: A Comprehensive Review," *Mathematical Problems in Engineering*, 2020, pp. 1-24.

[17] MarketsandMarkets, "Machine Vision Market by Component, Application, and Region - Global Forecast to 2025," 2020.

[18] International Federation of Robotics (IFR), "World Robotics Report 2020," 2020.

[19] McKinsey and Company, "The State of AI in 2020," 2020.

[20] Q. Liu et al., "A Fabric Defect Detection Method Based on Deep Learning," *IEEE Access*, vol. 10, pp. 4284-4296, 2022.

[21] T. Benbarrad, M. Salhaoui, S. B. Kenitar, and M. Arioua, "Intelligent Machine Vision Model for Defective Product Inspection Based on Machine Learning," *Journal of Sensor and Actuator Networks*, vol. 10, no. 1, p. 7, 2021.

[22] S. Shahrabadi et al., "Defect detection in the textile industry using image-based machine learning methods: A brief review."

[23] P. Serafim and A. Basu, "Autoregressive models for fabric defect detection," *Journal of Textile Engineering*, vol. 45, no. 2, pp. 56-67, 2018.

[24] Y. Lin, "Wavelet-domain Hidden Markov Tree model for fabric defect detection," *Textile Research Journal*, vol. 89, no. 3, pp. 234-245, 2020.

[25] T.-M. Le, N.-P.-L. Le, N.-G.-N. Hua, N.-L. Bui, and V. Q. Dinh, "Insight Evaluation on Traditional and CNN Features," presented at the University of Information Technology, VNU-HCM, and Vietnamese-Germany University, Binh Duong, Vietnam.

[26] W. Wong and J. Jiang, "Computer vision techniques for detecting fabric defects," in *Applications of Computer Vision in Fashion and Textiles*, Elsevier, 2018, pp. 47–60.

[27] H. Xie, Y. Zhang, and W. Wu, "Fabric defect detection method combining image pyramid and direction template," *IEEE Access*, vol. 7, pp. 182320-182334, 2019.

[28] S. Larabi and N. M. Robertson, "An image analogies approach for multi-scale contour detection," *arXiv preprint arXiv:2007.11047*, Jul. 2020.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 779–788. `https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html`.

[30] R. Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE CVPR*, New Orleans, LA, 2014, pp. 580-587.

[31] R. Girshick, "Fast R-CNN," *arXiv e-prints*, 2015, 169: 1440–1448.

[32] Zhao Jia, Zhou Shi, Zheng Quan, Mei Shunqi, "Fabric defect detection based on transfer learning and improved Faster R-CNN," 2022.

[33] H. Abdellah et al., "Defects detection and extraction in textile imageries using mathematical morphology and geometrical features," *J Signal Process Theory Appl*, 2012; 1: 1–16.

[34] J. Pedro, "Detailed explanation of YOLOV8 architecture," *Medium*, `https://medium.com/@juanpedro.bc22/detailed-explanation-of-yolov8-architecture-part-1-6da9296b954e`.

[35] R. Girshick, "Fast R-CNN," *arXiv e-prints*, 2015, 169: 1440–1448.

[36] N. Vishwakarma, "Real-Time Object Detection with SSDs," *Analytics Vidhya*, `https://www.analyticsvidhya.com/blog/2023/11/real-time-object-detection-with-ssds-single-shot-multibox-detectors/`.

[37] G. G. Nair and V. Trivedi, "Application of AI and ML in Quality Control Department of Textile and Apparel Industry," Proceedings of the International Conference on Sustainable Design Practices, Apr. 2024.

[38] S. Chakraborty, M. Moore, and L. Parrillo-Chapman, "Automatic defect detection of print fabric using convolutional neural network," The Journal of The Textile Institute, vol. 112, no. 11, pp. 1943–1950, 2021, doi: 10.1080/00405000.2020.1867242.

[39] T. Oshima, "Investing in Fabric Inspection: AOI or AI?," Oshima Group, [Online]. Available: https://oshimagroup.com/f/investing-in-fabric-inspection-aoi-or-ai. [Accessed: 15-Jun-2025].

[40] J. Azevedo, R. Ribeiro, L. M. Matos, R. Sousa, J. P. Silva, A. Pilastri, and P. Cortez, "Predicting Yarn Breaks in Textile Fabrics: A Machine Learning Approach," Procedia Computer Science, vol. 207, pp. 2301–2310, 2022, doi: 10.1016/j.procs.2022.09.289.

[41] R. Ribeiro, A. Pilastri, C. Moura, F. Rodrigues, R. Rocha, and P. Cortez, "Predicting the Tear Strength of Woven Fabrics Via Automated Machine Learning: An Application of the CRISP-DM Methodology," in Proceedings of the 22nd International Conference on Enterprise Information Systems (ICEIS 2020), vol. 1, pp. 548–555, 2020, doi: 10.5220/0009411205480555.

[42] M. Talib, A. H. Y. Al-Noori, and J. Suad, "YOLOv8-CAB: Improved YOLOv8 for Real-time object detection," Karbala International Journal of Modern Science, vol. 10, no. 1, Jan. 2024, doi: 10.33640/2405-609x.3339.

[43] F. Hermens, "Automatic object detection for behavioural research using YOLOv8," Behavior Research Methods, vol. 56, no. 7, pp. 7307–7330, May 2024, doi: 10.3758/s13428-024-02420-5.

[44] M. Yaseen, "What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector," arXiv.org, Aug. 28, 2024. https://arxiv.org/abs/2408.15857.

[45] F. Feng, Y. Hu, W. Li, and F. Yang, “Improved YOLOv8 algorithms for small object detection in aerial imagery,” Journal of King Saud University - Computer and Information Sciences, vol. 36, no. 6, p. 102113, Jun. 2024, doi: 10.1016/j.jksuci.2024.102113.

[46] X. Chai, M. Zhao, J. Li, and J. Li, “Image small target detection in complex traffic scenes based on Yolov8 multiscale feature fusion,” Alexandria Engineering Journal, vol. 126, pp. 578–590, May 2025, doi: 10.1016/j.aej.2025.04.105.

[47] A. T. Khan, S. M. Jensen, and A. R. Khan, “Advancing precision agriculture: A comparative analysis of YOLOv8 for multi-class weed detection in cotton cultivation,” Artificial Intelligence in Agriculture, Feb. 2025, doi: 10.1016/j.aiia.2025.01.013.

[48] H. Li et al., “A multi-scale attention mechanism for detecting defects in leather fabrics,” Heliyon, vol. 10, no. 16, p. e35957, Aug. 2024, doi: 10.1016/j.heliyon.2024.e35957.

[49] F. Islam, N. Sumaya, M. F. Monir, and A. Islam, “FabricSpotDefect: an annotated dataset for identifying spot defects in different fabric types,” Data in Brief, vol. 57, p. 111165, Nov. 2024, doi: 10.1016/j.dib.2024.111165.

[50] X. Zhang, Y. Liu, and Z. Li, ”Fabric defect detection based on improved lightweight CNN model,” Fabric Defect Detection Based on Improved Lightweight CNN Model, 2025.