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

Dr. Md. Ashraful Alam
Department of Computer Science and Engineering
BRAC University, Dhaka, Bangladesh
ashraful.alam@bracu.ac.bd

Souharda Bhattacharjee
Department of Computer Science and Engineering
BRAC University, Dhaka, Bangladesh
souharda.bhattacharjee@g.bracu.ac.bd

Asim Ajwad Gani
Department of Computer Science and Engineering
BRAC University, Dhaka, Bangladesh
asim.jawad.ghani@g.bracu.ac.bd

Rifat Arman Chowdhury
Department of Computer Science and Engineering
BRAC University, Dhaka, Bangladesh
rifat.arman.chowdhury@g.bracu.ac.bd

Sajib Sarker
Department of Computer Science and Engineering
BRAC University, Dhaka, Bangladesh
sajib.sarker@g.bracu.ac.bd

Pulak Deb Roy
Department of Computer Science and Engineering
BRAC University, Dhaka, Bangladesh
pulak.deb.roy@g.bracu.ac.bd

***Abstract*—The textile industry, which utilizes both natural and synthetic materials, often encounters defects during production, leading to substantial financial losses. Traditionally, defect detection has relied on manual inspection, which is time-consuming, inefficient, and prone to human error. This paper provides an overview as to how to overcome this automatically, using Machine Vision techniques to be more precise. We analysed existing object detection models such as Yolov8n, Single Shot MultiBox Detector (SSD) with VGG16 backbone, Faster R-CNN with ResNet-50 and a custom model. We also developed our own hybrid approach which is a two stage pipeline with stage 1 detecting the defect using yolov8l and the next stage classifying the defect using EfficinetNet-B0. Our pipeline obtained a result of precision 84.2, recall 81.9, f1 83.0 We introduced a custom dataset containing 9075 images containing 4 classes of fabric defects collected directly from textile factories to mimic industrial environments where the quality inspection takes place. Furthermore, we designed a prototype system for industrial automation integrating our model. Our aim is not only to build a capable object detection model with good accuracy and lower computational costs but also to show how this can be implemented in real life through our prototype system design.**

***Index Terms*—Fabric Defect Detection; Deep learning; Computer-Vision; Financial Losses; Machine Vision**

I. Introduction

A. Introduction

In the textile industry, issues like machine breakdowns and yarn breakage can easily lead to fabric defects. These defects lower 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% of defects [26]. On the other hand, automated fabric defect detection offers a better solution by lowering costs and achieving a much higher accuracy rate (approximately 90%) [27].

This research aims to emphasize product inspection, exploring how automatic detection methods can improve defect detection. Machine-vision-based fabric defect detection has become a significant research area in the textile industry, involving the extraction of defect-related characteristics from textile images. Various studies provide a detailed review of machine vision methods for fabric defect detection. As machine vision gains popularity, it offers a potential solution for quality control. The traditional reliance on human inspection, where inspectors miss nearly 40% of defects [9], highlights the need for automation in defect detection.

B. Motivation

The RMG sector is one of Bangladesh's top industry which requires no room for error. Thus quality control is a vital step in production which is not upto mark due to manual implementations which are error prone. Existing automated techniques are not upto mark and requires improvements. Thus this paper aims to improve the automation of quality assurance in the textile industry using machine vision empowering the countries RMG sector.

C. Research Problem

To prepare for the automation of fabric quality control, a whole variety of challenges need to be overcome. Different factories have different setups. To ensure our system works in all types of industries, the CNN model needs to be trained in proper way. Lighting condition should be same for every images as it can create differences. Even shadows and wrinkles can be misinterpreted as defects. 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. For data preprocessing, techniques like CLAHE require intricate turning as over enhancement may produce artifacts. Different image processing techniques cater to different types of defects. 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.

D. Research objective

The limitations of manual inspection indicate the need for an automated system capable of accurately identifying fabric defects. A human inspector typically detects only about 60% of defects and can inspect fabrics moving at a speed of 30 cm per second. In contrast, the goal of this research is to develop an automated system capable of detecting at least 90% of significant defects, with a lower computational cost [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. Different existing and hybrid CNN models will be trained with the dataset, selecting the best performing model based on the evaluation metrics and. 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.

II. LITERATURE REVIEW

A. Background:

Machine vision is gaining popularity in manufacturing industries for improving quality control and efficiency. Significant advancement is requiring as the demand is extremely high. The machine vision market was valued at USD 9.3 billion in 2020 and expecting to grow USD 16.5 billion in 2025 [18]. Automated inspection systems with machine vision have demonstrated improvements in productivity by 30% and also reducing labor costs [20]. Machine vision involve cameras, specialized lighting systems to capture detailed images of the component. Then its processed using digitization, thresholding, segmentation and edge detection to extract information which helps in identifying product defects, ensuring each items meet quality standard. If any defect is detected, the system take necessary actions. Furthermore, real-time feedback mechanisms allow continuous monitoring, adjustment and ensuring quality and efficiency [13] [14].

B. Related Work:

The research article written by Arshad and Shahzad (2024) [15] demonstrates the integration of deep learning models such as ResNet and VGG-16 to detect fabric defects. While earlier methods such as Gabor filters and gray level co-occurrence matrix(GLCM) might have laid the groundwork, they are computationally heavy, resulting in feature redundancy and unable to handle intricate patterns. Between three categories of defects, horizontal defects, vertical defects, and holes, each had a balanced dataset of 3630 images with 80% for training and 20% for testing. While VGG-16 took longer to train, it achieved an accuracy of 73.91% compared to ResNets' 67.59%. Despite being fast, ResNet struggled with smaller and more subtle defects. There are limitations to this approach. To start with, since both models are computationally heavy, it is more challenging to use them for larger datasets and can pose a problem when scaling up for industrial applications. The current system can only detect one defect per image making it impractical as fabrics might contain multiple defects. The models rely on static images which voids any hope of real-time detection.

Frouke Hermens(2024) [43] explored the effectiveness of YOLOv8, a state-of-the-art object detection model for automating video annotation in behavioral research context such as tracking surgical tools or daily use objects. The results show that YOLOv8 performs with a high accuracy rate even when trained on relatively small datasets, particularly in controlled lab environments with consistent backgrounds. However the model struggles to generalize when the same object appears in different backgrounds, which can be mitigated by training on a more diverse dataset. experiments comparing YOLOv8 with earlier versions (YOLOv3 and YOLOv5) and different model sizes(nano,small,medium) reveal that YOLOv8 offers superior performance with efficient training and inference. Overall this model proves to be a highly accessible and reliable tool for behavioral researchers needing automated object detection, provided that training conditions are well-matched to the models environment.

Zhang et al.(2025) [**?**] introduced a new idea of detecting the defects in the fabric using an improved lightweight convolution neural network (CNN) model. The proposed approach by the authors incorporates a lightweight version of the CNN, which will improve the identification of the defects on fabrics, in the real-time application scenario. This model is able to integrate the feature extraction and defect classification in a way that uses very minimal available computation power which makes it run effectively on lower end devices. 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 .

III. Dataset

Our system will enhance the training stage of our fabric defect detection model using a novel dataset, specifically curated straight from the RMG factories located in Bangladesh. We want to use our system to bring some significant improvements in the process of discovering fabric defects where we will be using a new dataset created to address the practical needs of RMG factories in Bangladesh. Compared to the large collection of publicly 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.

A. Data Collection

Our research utilizes a custom dataset 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 1 The data collection process employed in this manner ensures that the dataset 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 I.

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 I
Factory list with corresponding visit dates.

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 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.

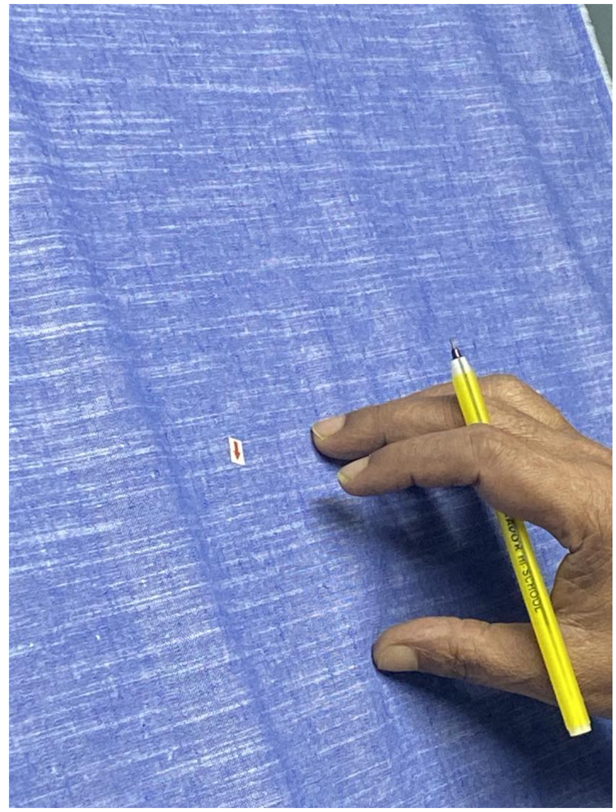
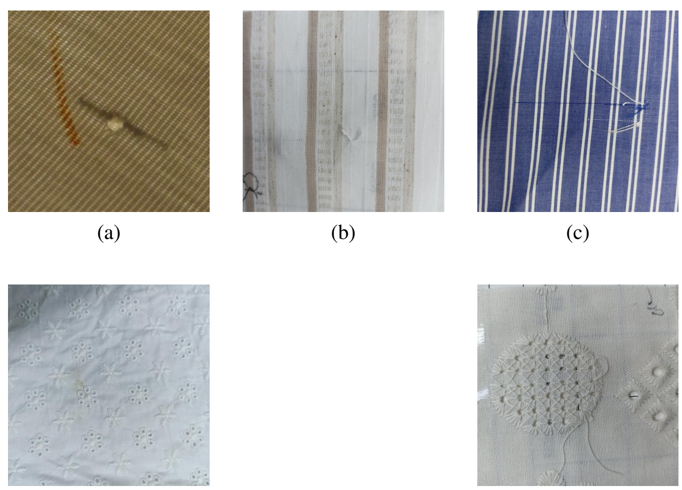


Fig. 1. QC operator marking a defect during inspection.



(a) (b) (c)

(d) (e)

Fig. 2. Examples of different fabrics under consideration.

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.

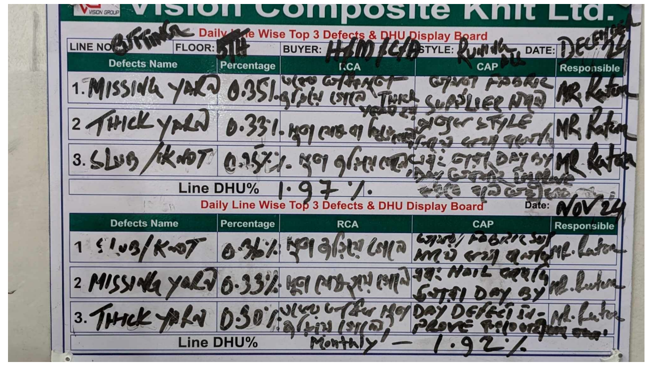


Fig. 3. Daily line-wise top 3 defects and DHU display board.

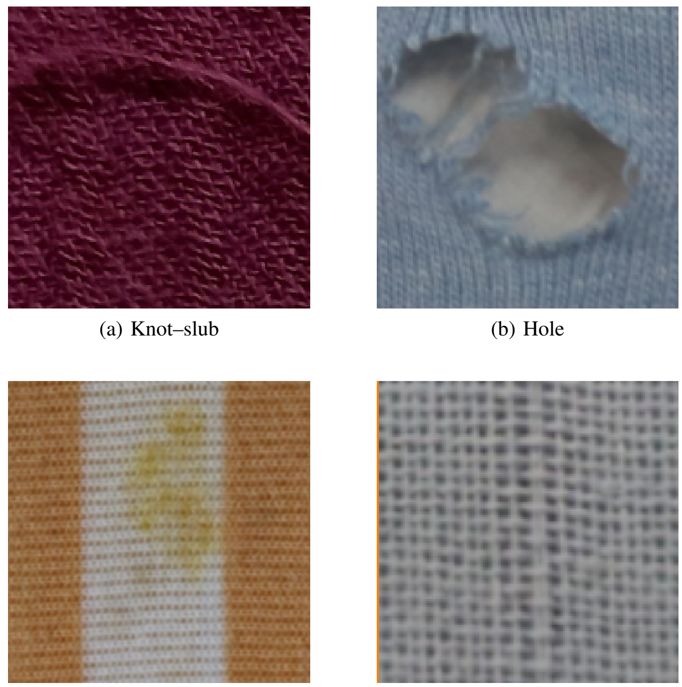
B. Data Description

In our dataset, four different categories of fabric defects have been captured, which are shown in Figure 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. 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**: Open gaps or punctures in the fabric structure.
- **Knot/Slub**: Lumps or irregularities in the yarn that appear as thick spots on the fabric.
- **Spot**: Unwanted discolorations, stains, or visible marks on the textile surface.
- **Thick Yarn/Missing Yarn**: Sections of the fabric with overly thick yarn or absent yarn, disrupting the weave pattern.

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.



(a) Knot–slub (b) Hole

(c) Spot (d) Thick/Missing Yarn

Fig. 4. Representative defect classes in fabric inspection.

C. Class Representation and Data Imbalance

Defects in fabrics are random and have 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. The dataset consists of 138 holes, 245 knot-slub, 302 Thick-Missing yarn and 191 spots as shown in figure 6. We can clearly see from here that classes such as holes appear less than thick-missing yarn. This class difference is small but still adressable. Thus to mitigate this, we opted towards oversampling the hole class. Having a balanced class count helps the model to avoid overfitting. Figure 5 shows the class percentages before and after it was balanced.

D. Preprocessing and Annotations

Images that have excess backgrounds such as the fingers of the quality control (QC) operators or any unnecessary

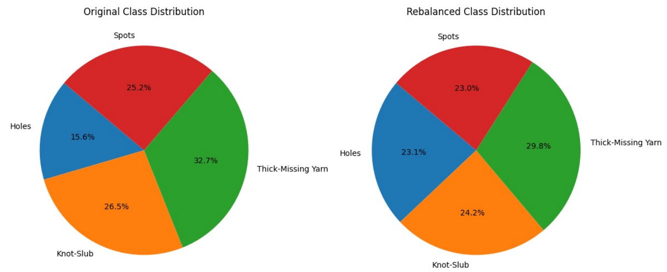


Fig. 5. Class distribution and their corresponding weights.

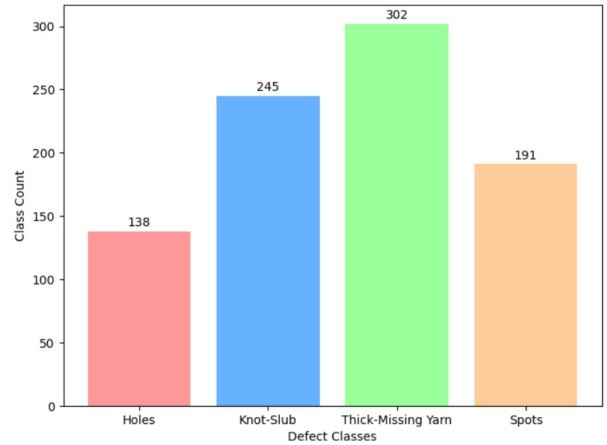


Fig. 6. Original class count.

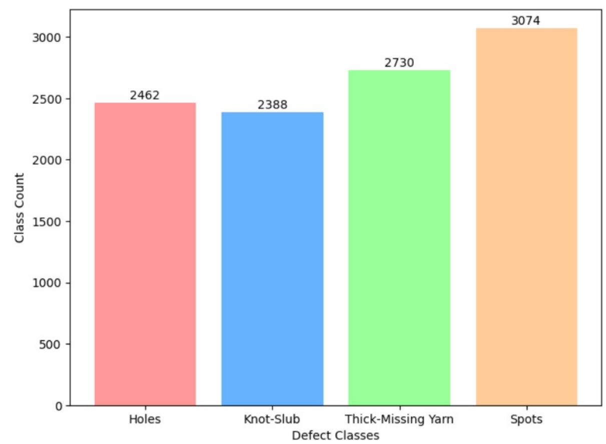


Fig. 7. Final class count.

background are removed so that only the part of the fabric with the defects visible remains. 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. All the defects were carefully marked through the usage of bounding box annotations.. 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. 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. 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. 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.

E. 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 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

IV. Research Methodology

A. Work Flow

The development of our defect detection system followed a structured methodology involving model experimentation, comparative evaluation, and final integration. After data collection and preprocessing, multiple object detection models were fine-tuned and benchmarked. Their performance was measured using standard evaluation metrics. A two-stage hybrid model—YOLOv8l for detection and EfficientNet-B0 for classification—was ultimately selected to be proposed based on accuracy and computational efficiency.

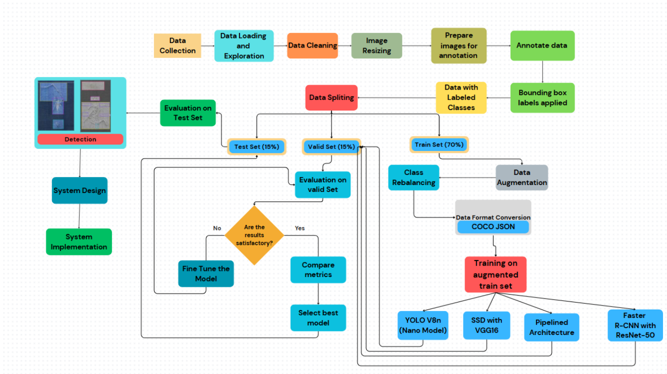


Fig. 8. System workflow for model training and evaluation.

B. Model Selection

YOLOv8n (Nano): YOLOv8n, the lightweight variant of the YOLOv8 family, was initially tested for baseline performance. Its anchor-free design and real-time inference capabilities allowed rapid experimentation, but had limited detection accuracy on small and complex defect regions.

SSD with VGG16 Backbone: SSD applies a single-shot detection strategy using multiple default anchor boxes. Combined with the VGG16 backbone, it achieved decent accuracy and real-time speed. However, its performance on multi-scale defect localization and rare classes was suboptimal.

Faster R-CNN with ResNet50: Faster R-CNN, integrated with a ResNet50 backbone, offered high detection accuracy and effective proposal refinement. Despite strong performance, its relatively high inference time made it less suitable for real-time or edge scenarios.

C. Pipelined Architecture (YOLOv8l + EfficientNet-B0)

The final proposed system utilizes a two-stage pipeline combining YOLOv8l for binary defect detection and EfficientNet-B0 for multi-class defect classification. This design ensures efficient computation by applying classification only to regions marked as defective.

Stage 1 – YOLOv8l for Binary Detection: YOLOv8l processes the input image to detect potential defect locations. It employs a three-part architecture:

- *Backbone:* Uses C2f-enhanced modules for feature extraction and residual connections to preserve spatial details.
- *Neck:* Fuses multi-scale features via upsampling and concatenation to improve detection across varied defect sizes.
- *Head:* Outputs bounding boxes and a binary class label (defect/no defect) using anchor-free regression and classification branches.

Cropped Regions of Interest (ROIs) from positive detections are then passed to the second stage. The YOLOv8l architecture is shown in Fig. 9.

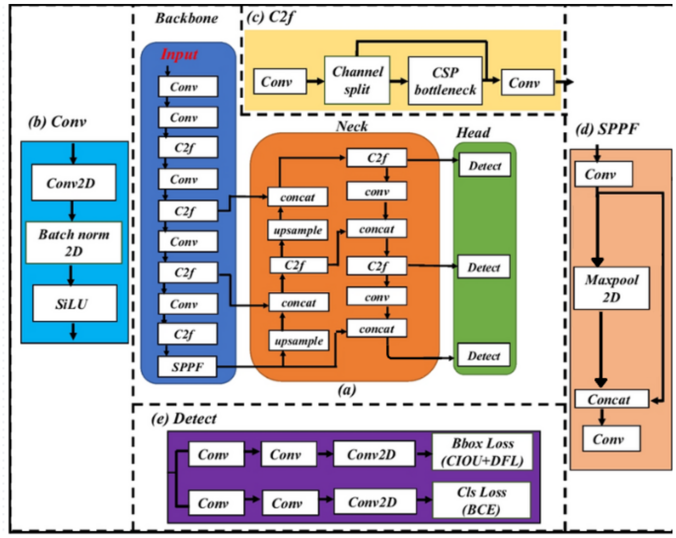


Fig. 9. YOLOv8l architecture used for binary defect detection.

Stage 2 – EfficientNet-B0 for Multi-Class Classification: Each ROI is resized to 224×224 and classified using EfficientNet-B0. The model utilizes MBConv blocks with Squeeze-and-Excitation modules, achieving high accuracy with low parameter count. It outputs softmax probabilities across predefined defect classes.

Integration and Control Flow: A control module manages the flow between detection and classification:

- Detect defects using YOLOv8l.
- If detected, crop each bounding box into an ROI.
- Preprocess and send ROIs to EfficientNet-B0.
- Combine detection and classification results for final output.

This modular pipeline enables scalable and interpretable defect detection with deployment flexibility across resource-constrained and industrial environments. The complete workflow is summarized in Fig. 10.

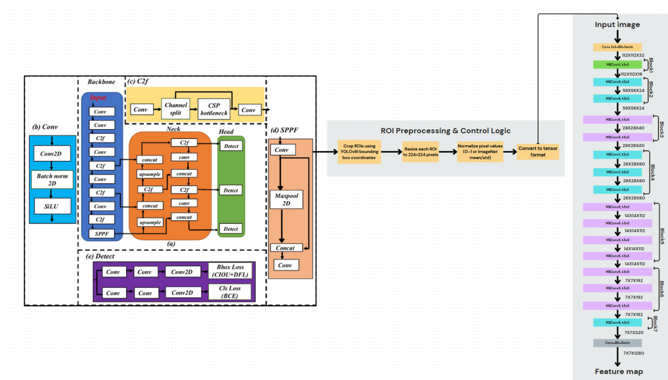


Fig. 10. Proposed pipeline workflow integrating YOLOv8l and EfficientNet-B0.

D. 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:
$$\text{Precision} = \frac{TP}{TP + FP}$$
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:
$$\text{Recall} = \frac{TP}{TP + FN}$$
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:
$$\text{F1 Score} = 2\frac{Precision Recall}{Precision + Recall}$$

V. Implementation and Result Analysis

A. 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.

Learning Rates

- YOLOv8n: 0.001
- SSD: 0.001
- Faster R-CNN: 0.0001
- Pipelined Architecture: 0.001

Batch Sizes

- YOLOv8n: 16
- SSD: 16
- Faster R-CNN: 4
- Pipelined Architecture: 8

Optimizers

- YOLOv8n: AdamW
- SSD: Adam
- Faster R-CNN: Adam
- Pipelined Architecture: Adam

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, and F1-score. 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.

B. Performance Evaluation

For each model, we used the same evaluation metrics—precision, recall, and F1 score—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:

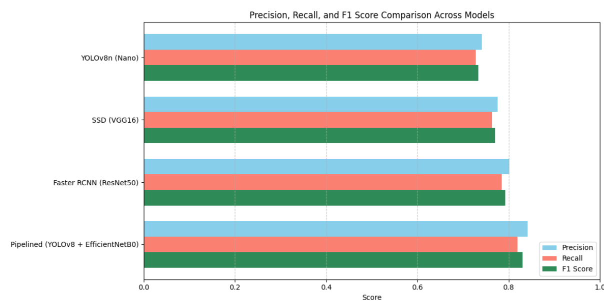


Fig. 11. Precision, Recall, and F1 Score Comparison Across Models

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.

VI. Hardware Implementation

This section describes the design and the hardware components used for the prototype of our system. 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 moving the fabric and a servo mechanism for marking defects. The system provides an efficient and modular solution for fabric quality control

A. System Overview

The system consists of several core components that work together to detect and mark fabric defects, as shown in 12 and detailed below:

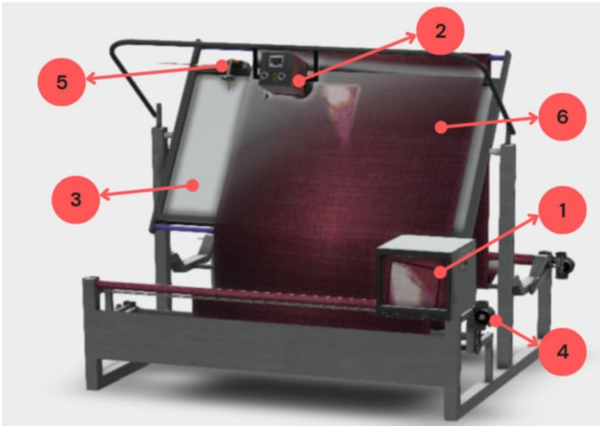


Fig. 12. System overview.

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 II
SYSTEM COMPONENT LABELS CORRESPONDING TO FIG. 12.

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 13 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.

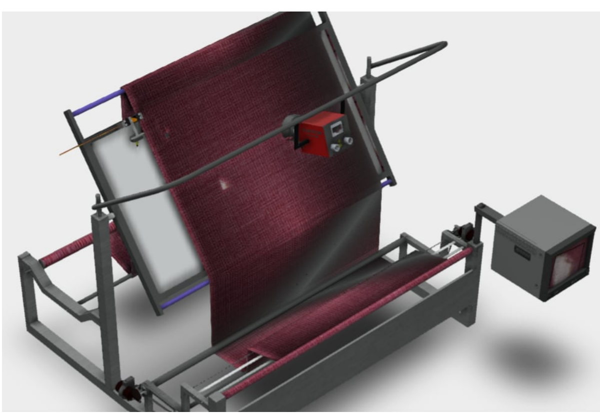


Fig. 13. Camera position above the fabric inspection setup.

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

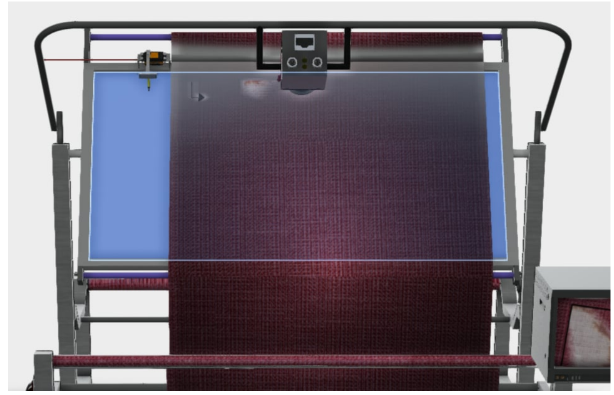


Fig. 14. Camera position with illuminated lightbox.

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 **??**

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.

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 **??** shows this.

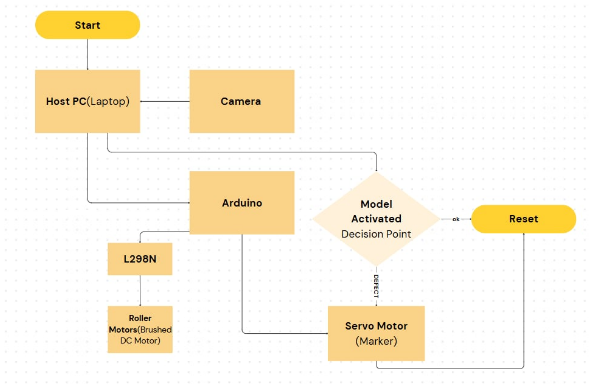


Fig. 15. System connection diagram.

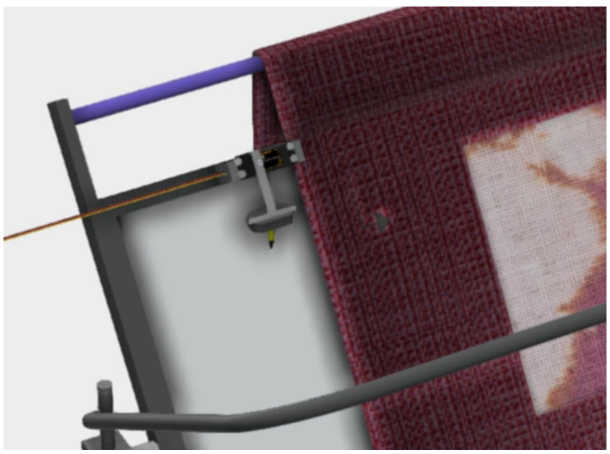


Fig. 16. Marker mechanism overview.

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 **??**.

B. Working Principle and System Workflow

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.

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.

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.

4) Defect Response and Marking: Upon receiving the defect signal, the following actions are performed:

5) 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) 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:

a) 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 given by:

$$C = 2\pi r$$

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:

$$v_m = \frac{RPM \times C}{60}$$

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:

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

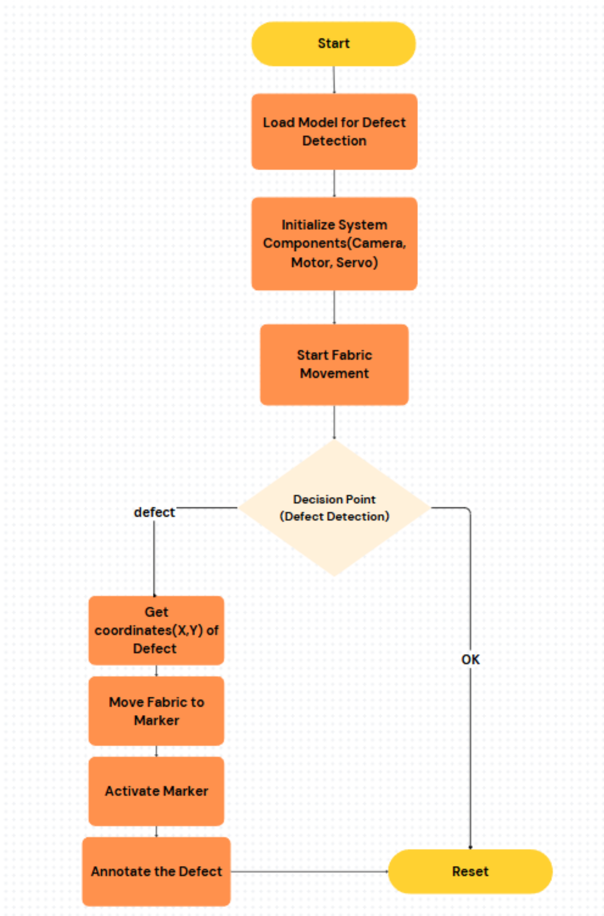


Fig. 17. Workflow diagram of the automated fabric inspection and marking system.

b) 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:

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

Where:

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

C. 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.

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.

VII. Limitations and Further Work

A. Limitations

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.

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. 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.

3) 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.

4) 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.

B. Future Works

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. 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.

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.

3) 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. 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.

VIII. 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, pre-processed, 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. Available: https://www.cv-foundation.org/openaccess/content$_{c}vpr_{2}016/html/Redmon_{Y}ou_{O}nly_{L}ook_{C}VPR_{2}$

[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] 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.