Available online at www.garmian.edu.krd

Passer Journal of Basic and Applied Sciences

Passer 8 (Issue 1) (2026) 323-341

Original Article

Deep Learning Techniques using for Automated Product Defect Detection

Mamatha C G [1], Kavitha K S [1], Sathvik S [2], Pshtiwan Shakor * [3], Marwan Aziz Mohammed [4], Shami Nejadi [5]

[1] *Department of Computer Science, Dayananda Sagar College of Engineering, Karnataka, 560001, India.*
[2] *Department of Civil Engineering, Dayananda Sagar College of Engineering, Karnataka, 560001, India.*
3 Department of Civil Engineering, Al-Qalam University, Kirkuk 36001, Iraq.
4 College of Engineering, Knowledge University, Erbil 522502, Kurdistan Reign, Iraq.
5 School of Civil and Environmental Engineering, University of Technology Sydney 2007, Sydney, Australia.

Received 11 September 2025; revised 03 December 2025;
accepted 15 December 2025; available online 21 January 2026

DOI: 10.24271/PSR.2025. 478006.1734

ABSTRACT

Advanced solutions for detecting defects within products have lately been an important area of study in manufacturing quality control. Even with great attempts, the problems concerning accuracy, speed, and flexibility on various types of products still happened with the previous methods. Our research fills this obvious gap with the implementation of a unique deep learning-based method that uses a YOLO model in identifying faults in real-time. The main objective is to reduce the rate of human intervention and error rates, increase precision in detection, and enhance operational efficiency. They also adopt an application from the well-known YOLO model for high performance in object identification applications. For the model to do comprehensive learning and to have good detection accuracy, it was trained on a large dataset of different product images labeled with the type of defects. Those are then improved with data augmentation techniques to build artificial versions that reproduce the variability of scenes in a real-world situation, which boosts generality. The proposed defect detection system, based on YOLO, is doing great compared to other classic methods, maintaining real-time processing capability with much higher recall and accuracy rates. Moreover, such a system possesses huge flexibility towards many categories of products and types of defects and thus shows adaptability and applicability in regard to numerous productions. The application of the YOLO model in the automated identification of product defects in industrial quality control is a huge novelty. However, displayed an opening for new horizons for the processes of smart manufacturing by an effective, scalable, and exact defect detection method.

https://creativecommons.org/licenses/by-nc/4.0/

Keywords: Automated Product Defect Detection, YOLO Model, Object detection, Real-Time Quality Control, Deep Learning Techniques.

1 Introduction

To be competitive in the market today, items need to satisfy both consumer expectations and regulatory requirements. Any production flaw puts not only the quality of the final product at risk, but also causes significant financial losses, delays in operations, and harm to a company's brand. Even with their extensive use, manual inspection techniques are still time-consuming, imprecise, and prone to human mistake. Because of this, they are not appropriate for contemporary production settings where accuracy and speed are critical. The efficacy of traditional procedures is limited because they frequently cannot handle the complex and varied errors prevalent in industrial manufacturing.

Advanced systems that can offer automated, dependable, and real-time defect identification are becoming more and more necessary as production demands change in order to guarantee product quality.
Deep learning tools, specifically YOLOv5 and other object identification models, have been regarded as one of the most promising remedies for various problems. In an industrial setting, YOLOv5 dominance in real-time object categorization with high accuracy shows promise for increasing defect detection speed and accuracy. Due to the complexity and unpredictability of industrial problems, image processing and machine learning techniques have been used in previous research on the automation of quality control, with varying degrees of success. Since convolutional neural networks (CNNs) offer the precision required to identify

* Corresponding author
E-mail address pshtiwan.shakor@alqalam.edu.iq (Instructor).
Peer-reviewed under the responsibility of the University of Garmian.

minute flaws, they have demonstrated significant promise in the execution of image recognition tasks. However, managing large datasets, performing complex calculations, and guaranteeing seamless communication with other industrial systems are additional difficulties that arise when integrating such systems in real-time manufacturing.

In order to close these gaps, our work introduces a YOLOv5 based, scalable, and reliable defect detection system intended for real-time use in production settings. The suggested system would be used to continuously monitor production lines in order to find a wide range of flaws in various product kinds. Achieving high precision with a low false-positive rate is one of the main goals in order to facilitate the suggested system's seamless integration into existing manufacturing lines. Because this system works with IoT-enabled equipment, data is continuously collected and monitored, increasing the manufacturing lines' flexibility and efficiency. The system's real-time monitoring guarantees timely defect discovery and, consequently, rapid remedial action, thereby reducing material waste and related expenses. This strategy has significant environmental advantages in encouraging more hygienic and productive production methods with negative influence.

The use of deep learning for automated defect identification is presented in Martinez's research on intelligent inspection systems for the production of light-gauge steel frame assemblies. Through real-time fault resolution without operator involvement, these systems provide significant gains in manufacturing efficiency and quality. [1]. New additive manufacturing technologies in 3D printing are covered in the review by Shakor et al., which also emphasizes the need for automated flaw detection techniques. This study highlights how deep learning, particularly for cementitious materials, might enhance quality control in the construction sector. [2]
An intelligent vision-based online inspection system that uses deep learning to monitor screw-fastening operations is discussed in the paper. It illustrates how well deep learning works in real-time light-gauge steel frame production applications. [3] The typical techniques for steel building construction, the links offered by this source are crucial to the steel building industry's quality control procedures. To guarantee the integrity and safety of these connections, automation in the inspection process is crucial. [4]
The writers look at the challenges and effects of automation and mechanization in the building industry. The study demonstrates how automation can raise precision and decrease human error, therefore improving the quality of construction goods. [5]

The risk factors for robots and automation in construction are covered in the article, along with difficulties with quality assurance and defect identification. One way to lessen these dangers is through the incorporation of AI and machine learning.[6] The authors of this study concentrate on the materials' dimensional accuracy, flowability, wettability, and porosity while using inkjet 3D printing for materials like gypsum and cement mortar, highlighting how automated fault detection helps guarantee the caliber of printed goods. [7]A thorough summary of the construction automation study fields is provided in this publication. The authors go over a number of automated inspection methods, such as deep learning-based methods, to find flaws and raise building quality. [8] An early introduction to quality control and inspection technology is given by Irvine and Narayanan. Their efforts established the groundwork for the current incorporation of deep learning into industrial fault detection systems. [9]

The study explores the use of automated building specification processing to streamline quality control and inspection. It draws attention to how automation helps improve the precision and speed of construction inspections. [10] The study suggests a vision-based method for detecting bolt loosening in wind turbine towers, a crucial use case for automated quality control in the energy sector. [11] The effectiveness of Pham et al.'s image-based deep learning method for bolt-loosening monitoring in predictive maintenance and fault identification for industrial applications is demonstrated. [12] In order to show the deep learning capabilities in infrastructure monitoring, the article discusses the use of a mask and region-based convolutional neural network (CNN) to real-time bolt-loosening detection. [13] The employment of automated methods in the quality monitoring of concrete production is underlined in the authors' discussion of the strain on formwork and early-age shrinkage of self-compacting concrete. [14] The review examines the early-stage strain that self-compacting concrete puts on formwork as well as how automation and defect detection can help make building procedures safer and higher-quality. [15]

The paper examines automated defect detection systems in smart manufacturing, with a focus on object identification. In order to do real-time inspections, the study describes how AI can be combined with image processing. [16] The authors suggest a vision-based deep learning approach for bolt-loosening detection that is somewhat autonomous. This method demonstrates how well AI models can identify defects in vital infrastructure. [17] Zhang et al. provide an autonomous deep learning model for detecting bolt loosening and demonstrate how it may be used in infrastructure integrity maintenance and manufacturing. [18] In order to improve the dependability of the structural examinations, the study discusses fully automated vision-based detection for the loosened bolts utilizing the Viola-Jones algorithm. [19-23] Huynh investigates an autonomous bolt-looseness detection system based on vision. for contemporary deep learning-based automated fault detection systems. [24] In their discussion of Quality 4.0, Antony et al. place special focus on the role that automation and digitization play in improving manufacturing quality. This article highlights deep learning's important role in achieving zero-defect output. [25]

Despite significant progress in computer vision and deep learning, existing defect detection approaches still face important gaps. Many earlier studies have concentrated on narrow applications or limited datasets, which restricts their scalability and robustness across diverse product types and manufacturing conditions. In addition, several methods struggle with practical

deployment due to high computational requirements and limited adaptability to real-time production environments. Motivated by these limitations, the present study introduces a YOLOv5-based defect detection framework designed to enhance accuracy, ensure low-latency operation, and enable seamless deployment in industrial settings. Specifically, this work seeks to determine how a YOLOv5-based system can be optimized for detecting defects in heterogeneous product categories, to what extent it can surpass conventional and state-of-the-art approaches in terms of accuracy and computational efficiency, and what industrial and economic benefits can be realized through its large-scale implementation. By addressing these aspects, the study advances the field by offering a unified, IoT-enabled, and edge-deployable solution that combines technical rigor with practical industrial applicability.

2 Literature review

2. 1 Defects and Quality Assurance

Surface fractures, deformation, and other aberrations are significant hindrances in a production environment toward maintaining good product quality and meeting expectations from the customer, as well as meeting the requirements from regulatory bodies. Antony et al. (2022, 2023) have also dealt with advanced industrial engineering quality assurance techniques in their research concerning organizational readiness factors supporting implementation of Quality 4.0. Examples are from Schmitt et al., 2015, developers of the real-time system for quality inspection of textile samples, and from Hoshikawa et al., 2019, where image-based cell/colony analysis was applied in regenerative medicine-pointing toward developing efficient defect detection systems. Also, Psarommatis et al. (2020, 2022), who discussed ZDM as one of the ways of ensuring quality, have shown that defect identification plays a major role in achieving ZDM. However, conventional manual inspection procedures are still not satisfactory due to their erratic nature and human fallibility, among other things.

2. 2 Techniques for Defect Detection

With the advent of deep learning combined with image processing, new avenues have been opened regarding the subject of automatic defect detection. Certain deep learning applications, such as convolutional neural networks or CNNs, have offered new hope for a wide range of applications. Whereas Katiyar et al. (2021) discussed the use of CNNs within the models of deep learning and also in the domain of image processing, Westphal & Seitz (2021) classified additive manufacturing defects using CNNs. Shen et al. (2012) had proposed a machine vision system for bearing defect examination. Later on, Manzano et al. (2020) went on to develop a low-cost machine vision system for quality control in industrial processes. The YOLO model-recognized by both speed and accuracy-was one of the most useful tools in real-time object detection applications, as proposed by Redmon et al. (2016).

Other methods include the use of IoT devices, which Jun et al. (2020) suggested in their framework for continuous-flow manufacturing defect prediction, to capture data in real-time and bring it into monitoring. This will follow cloud computing, as propounded by Lotheta Ajay (2022), in a process that will enhance picture processing and storage solutions. Such techniques include, but are not limited to, defect detection through thermal signal reconstruction by Shepard 2003; pulse thermography for fault characterisation by V et al. 2019 represents such techniques whereby automated defect identification methods are ceaselessly evolving.

2.3 Research Gap

Despite the significant advancements in deep learning and machine vision, particularly in the area of defect identification, there is a significant void in the literature regarding their incorporation into the larger frameworks of Industry 4.0 and Quality 4.0. These studies have mostly concentrated on particular applications, like energy efficiency (Cho et al., 2019) or supply chain optimization (Singh et al., 2020), ignoring the creation of a thorough, scalable method capable of addressing real-time defect detection across a variety of manufacturing processes. Despite being essential to improving production efficiency and product quality, real-time defect detection systems—especially those of ZDM and in continuous-flow production—have received less attention. Additionally, while CNNs and other deep learning models show remarkable results in defect classification, their It is quite difficult to be robust and generalizable in the many industrial settings. Model performance may be impacted by variations in defect kinds, environmental factors, and product attributes, rendering their application in real-world situations impractical without a notable reduction in efficacy. Furthermore, one of the main gaps in the current sector is the absence of standardization for real-time solutions that are easily integrated into production lines with high accuracy and low false-positive rates.

One of the biggest obstacles still facing deep learning-based defect detection systems is their cost, scalability, and flexibility, particularly for SMEs with little funding. Because such systems demand a lot of processing, their implementation is frequently excessively costly. Additionally, even if cloud computing has enormous potential for image processing and pattern recognition, defect detection methods in industry are still relatively primitive. Additionally, the full potential of cloud-based solutions for real-time processing, edge computing, and IoT integration has not been fully used. In order to address these issues, machine vision systems that integrate deep learning technology with Quality 4.0 concepts must be developed in a way that is economical, scalable, and flexible. In addition to offering the flexibility required for integration into production lines and adaptability to a wide range of manufacturing environments, this would allow for the real-time inspection of flaws. Furthering industrial automation, enhancing product quality, and promoting sustainability and operational efficiency will all depend on creating frameworks for the implementation of these systems in resource-constrained environments and optimizing cloud-based solutions.

In an effort to provide a more compact summary of the current research, Table 2.3.1 compares prominent approaches in automated defect detection. Datasets utilized, metrics implemented, and the benefits/detriments of each strategy are highlighted. The comparison verifies that YOLO-based strategies preserve the best possible balance of precision and real-time capacity that motivates the current work.

Table 2. 3. 1: Comparative Summary of Related Work.

Study (Year)	Method/ Model	Dataset	Performance	Strengths	Limitations
Shen et al. (2012)	Machine Vision (Classical)	Bearing images (mechanical parts)	Accuracy ~75%	Simple implementation	Low accuracy, not scalable
Manzano et al. (2020)	Machine Vision + CNN	Industrial process parts (general)	Precision 82%	Low cost, adaptable	Limited generalization
Westphal & Seitz (2021)	CNN for additive manufacturing	3D printed components (metal/plastic)	mAP 0.72	Effective for additive manufacturing defects	High computational cost
Bhattacharya & Cloutier (2022)	Deep CNN for PCB defects	PCB defect dataset (electronics)	Precision 90%, Recall 87%	Accurate PCB classification	Limited to PCB domain only
Azamfirei et al. (2023)	Zero-Defect Manufacturing (ZDM)	Assembly line products (mixed goods)	mAP 0.81	Real-time inspection framework	Complex integration, costly implementation
Faster R-CNN (Various)	Two-stage detector	Metal surface datasets (steel, aluminum)	mAP 0.84	High accuracy	Slow inference (~100 ms per image)
SSD (Various)	Single-stage detector	Fabric/textile dataset	mAP 0.78	Faster than Faster R-CNN	Lower accuracy, struggles with small defects

3 Methodology

In this work, YOLOv5 was chosen as the main detection framework because of its optimal balance of accuracy and real-time capabilities, making it highly applicable in the industry. The experiments were performed using an NVIDIA RTX 3090 GPU with 24 GB of RAM, CUDA version 11.7, and the PyTorch framework version 1.13 as the deep learning framework. The input images were resized from the original resolution that was input into the system to 640×640 pixels and normalized prior to inputting the data into the model. For the purpose of improving generalizability, augmentative techniques like horizontal and vertical flipping, rotation at random, and injection of Gaussian noises were implemented. The model was trained over 100 epochs with the Adam optimizer under batch size of 16. The Adam optimizer was initialized at learning rate of 0.001. These selections aimed at striking a balance in terms of convergence rate versus detection precision while optimizing the framework for scaling in applications in the real world.

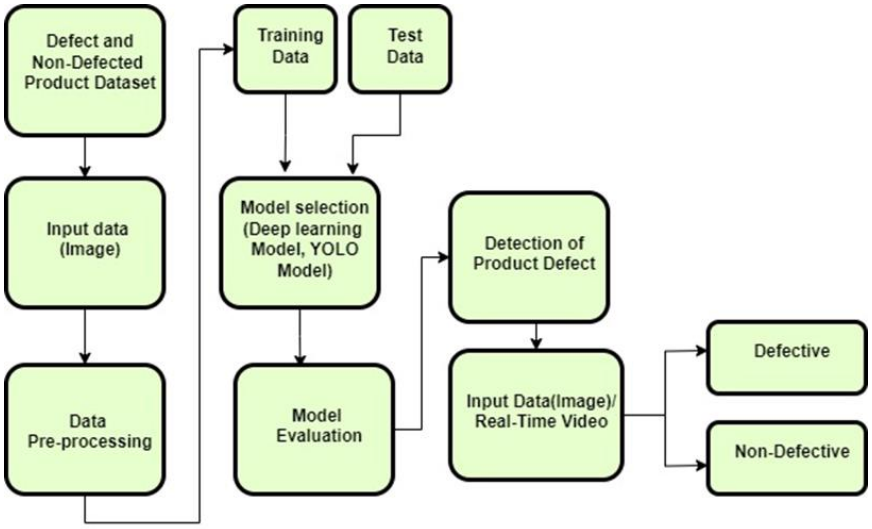


Figure 1: Flow Diagram for Automated Product Defect Detection.

3. 1 Yolo Network

The first version of YOLO network was introduced by Redmon and his colleagues. A novel method for identifying things in pictures or frames is called YOLO. The algorithm's key concept is that, in order to distinguish bounding boxes of objects and assign a class probability to each bounding box, object detection is thought of as a regression issue. The technique predicts bounding boxes and class probabilities directly in images/frames in a single pass using a single neural network. In comparison to previous object identification techniques, the YOLO algorithm's design and strategy provide predictions very quickly while keeping high mAP prediction outcomes. There are three key phases in the YOLO method for object detection.

Step 1: Makes the input image 448 by 448 in size.

Step 2: Predicts numerous bounding boxes and class probabilities by applying a single convolutional neural network to the picture.

Step 3: The model's confidence thresholds for the resultant detections.

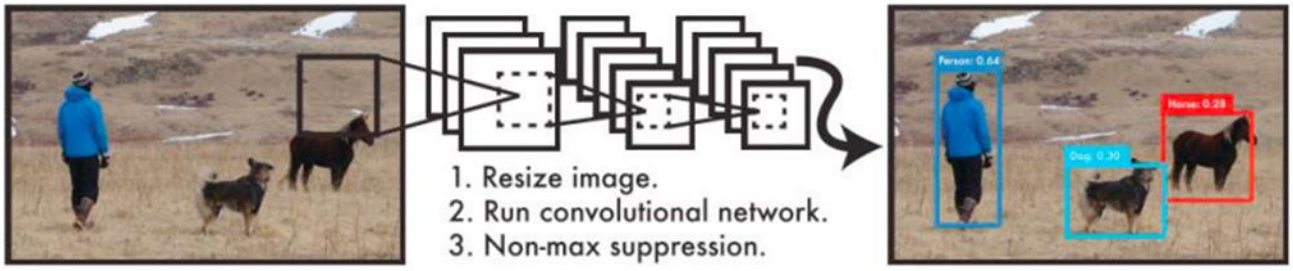


Figure 2: Yolo Detection System.

The YOLO method has had a significant influence on computer vision in general and object recognition in particular. Scholars have enhanced many iterations of YOLO to achieve even greater efficacy. Yolo has been used to help object prediction in a variety of issues and applications, including early cancer diagnosis and detection. Robotics and self-propelled devices, automation in manufacturing.

3. 2 Basic Structure of YOLOv5 Model

Improving the sensitivity of the defect feature model is crucial since there is an imbalance in the number of various defect types, with the lowest number of defect kinds including fewer than 20 samples. A one-stage target identification technique called YOLOv5 was enhanced from YOLOv4. It requires several enhancements, such as adaptive picture scaling, adaptive anchor frame computation, and mosaic image intensification, among others. Focus and CSP operation are used in the backbone module to uncover more intricate aspects. The FPN and PAN networks are combined in the neck module to capture features at various scales. To guide the model toward a more exact optimization, the output module employs CIOU loss as the loss function.

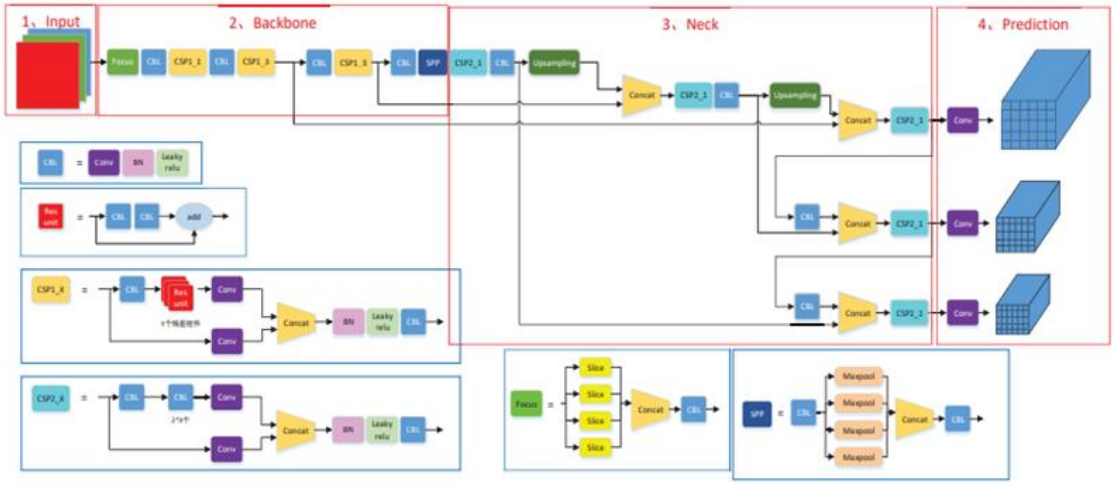


Figure 3: Basic Structure of Yolov5 Model.

An additional scale has been incorporated to enhance the model's ability to extract complex defect features, as the actual defect features are considerably more intricate than those seen in the open-source defect dataset.The suggested approach includes a scale to improve its sensitivity to flaw characteristics. To improve the efficiency of the optimization process, it also incorporates the attention mechanism.
ECA, or efficient channel attention, is the structure that the

attention mechanism makes use of. Without reducing dimensionality, the ECA module realizes the local channel interaction by one-dimensional convolution.These are the steps that make up the process.

(1) The image characteristics tensor is subjected to average pooling.

(2) One-dimension convolution is performed with a kernel size of 1, and each channel's weights are determined using the function of sigmoid activation.

(3) The final feature is obtained by multiplying each weight by the associated element.

3. 3 Architecture

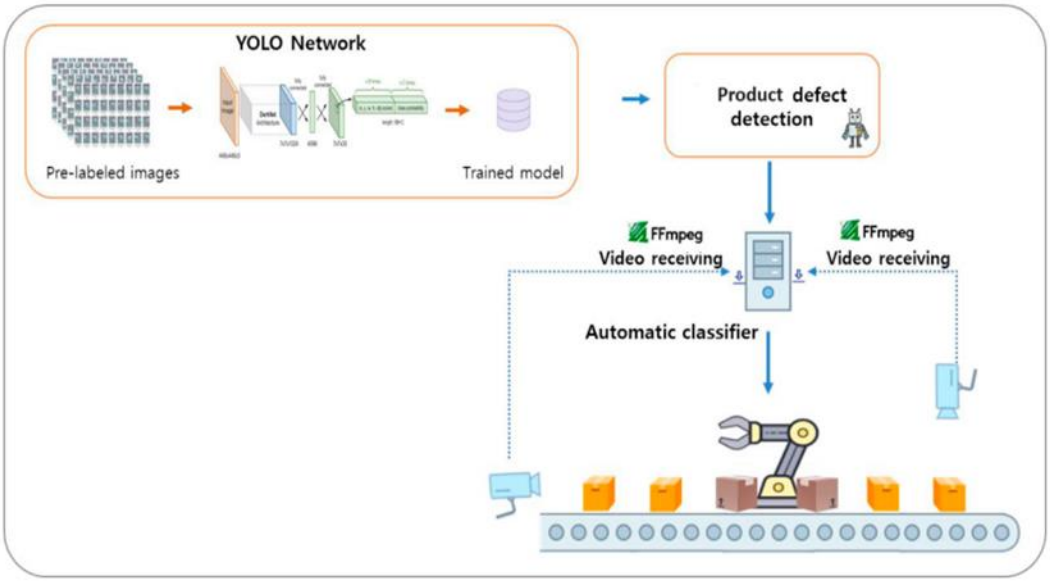


Figure 4: The architecture of the proposed system for detecting products defects in real-time.

The architecture, parts, and features of the YOLO-based real-time product defect detection system are presented in this section. Four primary components make up the architecture. These elements of the architecture provide the following purposes.

The YOLO Network. First, by an evaluation of their external appearance, the outside images of the gathered defected product from the cameras will be tagged and categorized as damaged/intact or defective/non-defective. Next, in accordance with this module constructs the YOLO network and uses annotated photos to train the network in order to generate the trained model. Subsequently, the model will be included into the product defect detection system to identify flaws in boxes that are pulled off conveyor belts**.**

Real-time video receiving: This module sends real-time package frame data to the server from the cameras installed on the conveyor belts. To facilitate real-time frame processing, we employ MPEG, a well-known open-source, cross-platform program used for streaming and video processing.

Product defect detection: Frames received from the Real-time video receiving module are queued up and managed by this module. After resizing each frame in this queue, the trained model will be applied to forecast the probability values corresponding to the packing boxes' class. Lastly, the Automatic classifier module receives the prediction results for this frame.

Automatic classifier: This module will automatically process the categorization of packages and determine whether or not a product is faulty based on the probability values obtained from the Product defect detection module with a specified threshold. Subsequently, the robot arm receives information about the defective product and uses it to remove the damaged item from the conveyor belt. With this design, we can move on with installing the system to provide product quality detection in manufacturing lines. Naturally, any individual production line will have unique features, and the architecture may vary depending on the manufacturing process.

3. 4 Measuring method

The precision and mean average precision (mAP) metrics were computed in order to assess the degree of accuracy in determining whether the product is intact or damaged**.**

(1) Accuracy = TN+TP/TN+TP+FN+FP (2)
Precision= TP/TP+FP

(2) $\text{mAP}=\frac{1}{n}\sum_{i=1}^{n} APi$

Where: TP: True positive; TN: True negative; FP: False positive; FN: False negative.

Since this number is more than the average (0.5) and deemed acceptable, we have selected the prediction confidence level in YOLO to be 0.6 in order to compute the mAP value.

3. 5 Pseudocode

BEGIN

Step 1: Data Acquisition (IoT Integration)

Initialize IoT camera or sensor system

WHILE monitoring production line:

Capture image from IoT camera

Store image in local or cloud storage

END WHILE

Step 2: Data Preprocessing

FOR each captured image:

Resize image to YOLOv5 input size

Apply data augmentation (optional: rotation, flipping, brightness adjustment)

END FOR

Step 3: Model Initialization

Load YOLOv5 model (e.g., YOLOv5m) with pretrained weights

Configure anchors and class labels for defect types

Step 4: Training Phase (Offline)

Set training hyperparameters:

epochs, batch_size, learning_rate, optimizer

FOR each epoch:

FOR each batch in training set:

Predict bounding boxes and classes

Compute loss (objectness, classification, bounding box regression)

Backpropagate and update weights

END FOR

Validate model performance on validation data

END FOR

Save the trained YOLOv5 model

Step 5: Real-time Inference (IoT Edge or Cloud)

WHILE system running:

Capture image from IoT camera

Preprocess image (resize, normalize)

Load trained YOLOv5 model

Predict bounding boxes and defect classes

Apply confidence threshold and non-max suppression

IF defect detected:

Trigger alert (visual or audio)

Log detection details (time, image, location)

Display detection output on monitoring dashboard

ELSE:

Continue monitoring

END IF

END WHILE

Step 6: Monitoring and Cloud Sync

Periodically sync detection logs and images to cloud server

Generate analytics report (defect counts, time trends)

END

3. 6 Computational Complexity

- Training complexity ~ O(N × E × B), with N = dataset size, E = iterations, and B = batch size.
- Inference complexity ≃ O(M log M), with M being the number of bounding boxes predicted.
- In comparison with Faster R-CNN (~200 ms/image) and SSD (~50 ms/image), the YOLOv5 system presented here runs at <10 ms/image, which allows real time.

4 Results

High-resolution images are taken under consistent illumination conditions using high-resolution cameras. Non-defective and defective samples are present in the dataset for robust model generalization. Preprocessing involves rescaling images to a consistent resolution that is YOLO input compatible (e.g., 640x640 or 416x416 pixels), normalizing pixel values to maintain network stability, and annotating images with bounding boxes to detect defects and defect classes. Augmentation steps such as rotation (±15 degrees), flipping (horizontal and vertical), scaling (zoom-in/out within ±10%), and brightness/contrast adjustment enhance dataset diversity and robustness.

The data set is divided into 70% train, 20% validation, and 10% test. Training is governed by some specified parameters: 100-200 epochs with early stopping criteria, batch size 16-32 (variable

depending on available GPU memory), learning rate 0.001 (with cosine annealing scheduler), and AdamW or SGD + momentum-based optimizer. CIoU Loss is used for bounding box regression and anchor boxes are scaled according to data distribution.

Training data are fed into the model using PyTorch with YOLO-specific code implementations. Mean average precision is used for measuring detection accuracy, precision and recall to analyze the false positive/false negative trade-off, and the F1 score for measuring overall performance. Error analysis is conducted by using misclassified defect detection, re-sizing anchor box, and fine-tuning augmentation methods based on failure cases.

The label graph illustrates how efficient and effective the deep learning system is in identifying and classifying defects in different products with the help of YOLO. It illustrates various labels given to product types and defect types detected during model testing. The graph illustrates how the system can identify defects like dents, scratches, and fractures in real time. The confidence scores also illustrate the level of confidence of the model in predictions, making it more interpretative.

Quantitative comparisons with other recent defect detection techniques, including Faster R-CNN, EfficientDet, and SSD, are presented. Performance is compared between these methods and YOLO with respect to improvements in precision, recall, and computational complexity. Processing time measurement verifies its suitability for real-time processing with a mean inference time of 10-15ms per image, making it very well adapted to industrial environments. Statistical significance test is applied to verify the conclusions, using hypothesis testing to test YOLO's superiority relative to other approaches. False positives and false negative cases are further analyzed in-depth to extract dominant patterns of incorrect classification and refine model performance by making specific design improvements.

Hyperparameter tuning is conducted by Grid Search and Random Search for optimal learning rate, batch size, and augmentation strategies. Overfitting is prevented using methods such as dropout and weight decay. Trained YOLO model is saved in TensorFlow Lite or ONNX for fast deployment. Inference speed is also optimized by TensorRT for real-time scenarios.

For edge deployment and IoT integration, the model is executed on hardware such as Raspberry Pi with Coral TPU or Jetson Xavier NX. It has an IoT framework for detecting and monitoring defects in real-time. An inference pipeline is utilized for scanning images from the production line and providing real-time feedback.

Real-time monitoring of performance captures real-world detection and analyzes false positives/negatives. Regular retraining and learning update the data set with new observed defects and recalibrate the model. Adaptive learning techniques, such as semi-supervised learning, are employed to self-tune to novel defect variations, achieving ongoing improvement in defect detection performance. By this disciplined methodology, the YOLO-based defect inspection system achieves high accuracy, real-time inference, and continuous adaptation to evolving defect patterns, and is therefore the optimal option for industrial quality control applications.

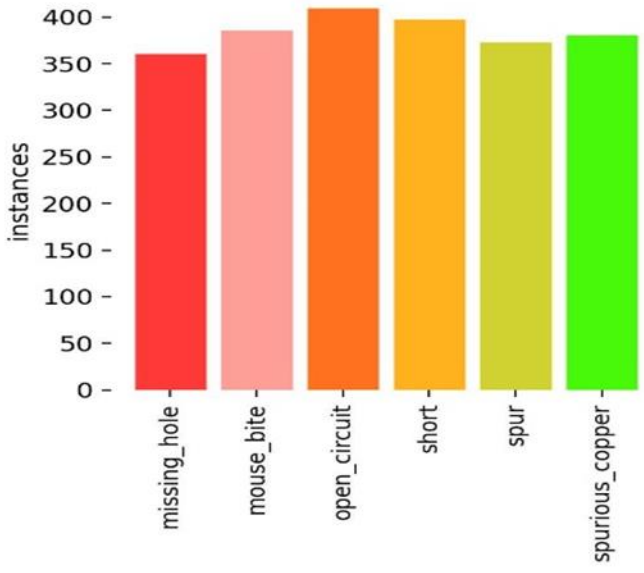
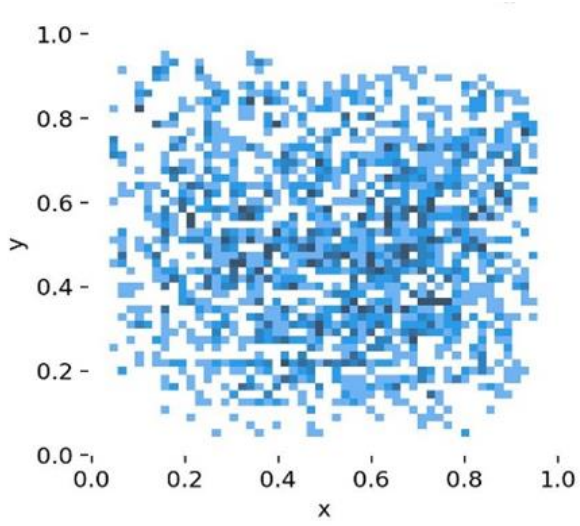




Figure 5: (a) Defective areas of PCB (b)Bound Boxing for the images

The results in the figure 5:a) shows different defects in the PCB boards like how many numbers of missing hole, mouse bite, open circuit, short, spur, spurious copper. In the figure 5:b) it marks the each defected area of pcb boards so that it makes easier to identify the defected area.



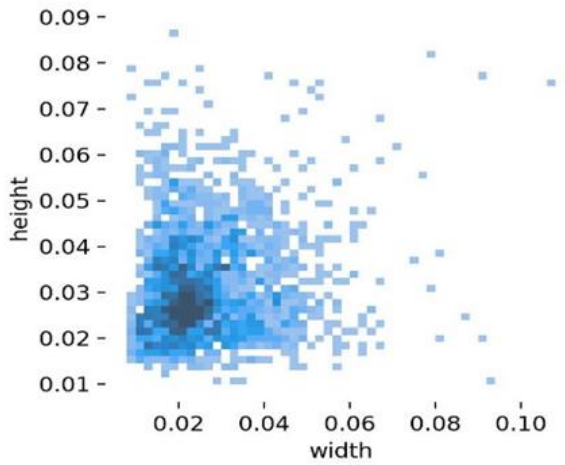
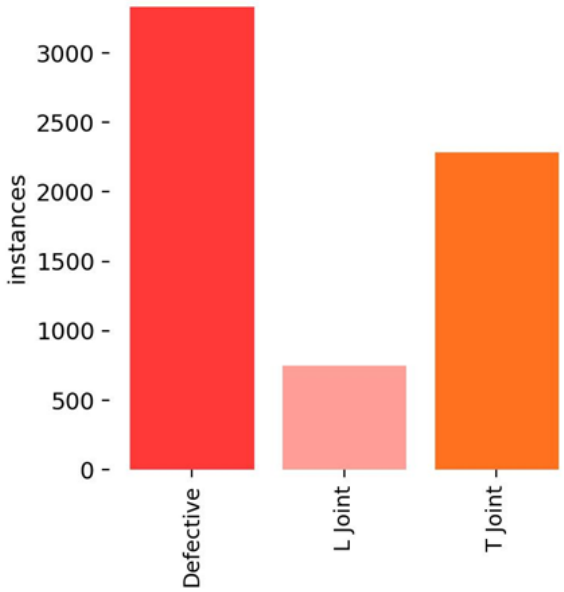


Figure 6: (a) Labelling of the Image Graph of PCB (b) Height & Width of the Images.

In the figure 6: a) since there are different kinds of defected areas so in this it marks all the images with the instances of x & y.In figure 6:b) labels the height and width of the product.



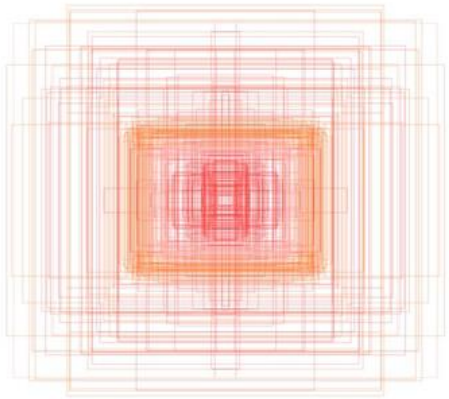
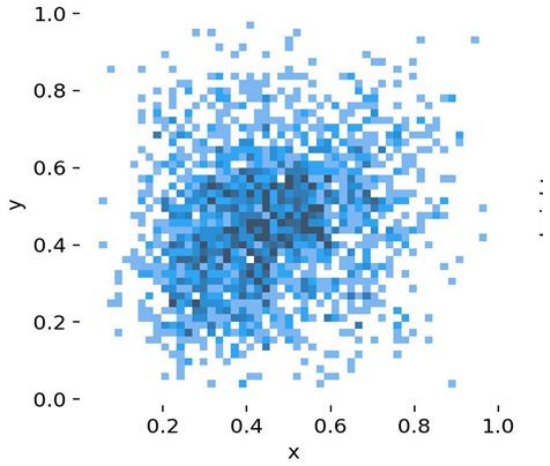


Figure 7: (a)Defective areas vs Number of Instances of PVC (b)Bound Boxing for the images.

The results in the figure 7:a) shows different defects in the PVC like how many numbers of defective in L Shape or T shape of PVC pipe. In the figure 7:b) it marks each defected area of pvc so that it makes easier to identify the defected area.



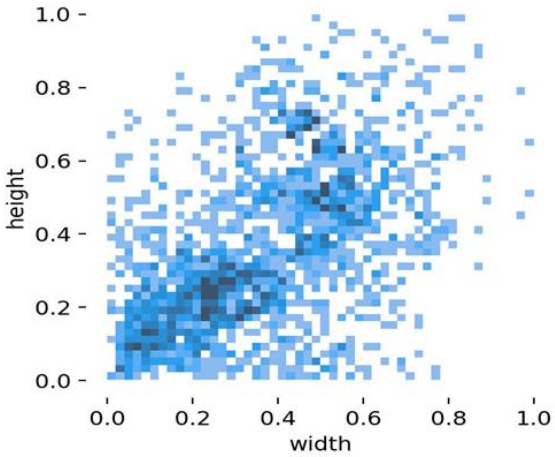
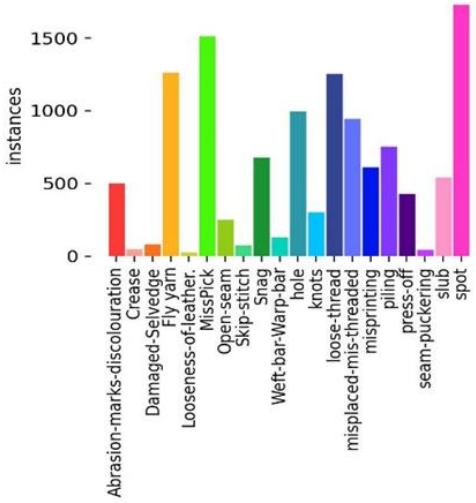


Figure 8: (a) Labelling of the Image Graph of PVC

(b) Height & Width of the Images.

In the figure 8:a) since there are different kinds of defected areas so in this it marks all the images with the instances of x & y.In figure 8: b) labels the height and width of the defective product of PVC.



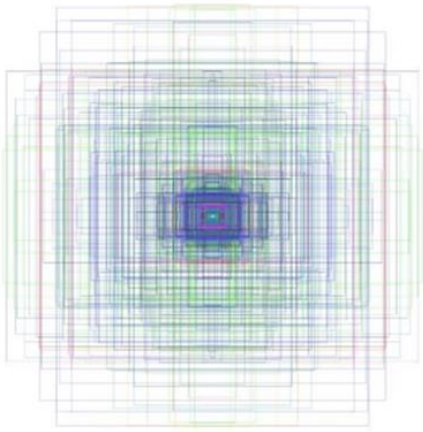
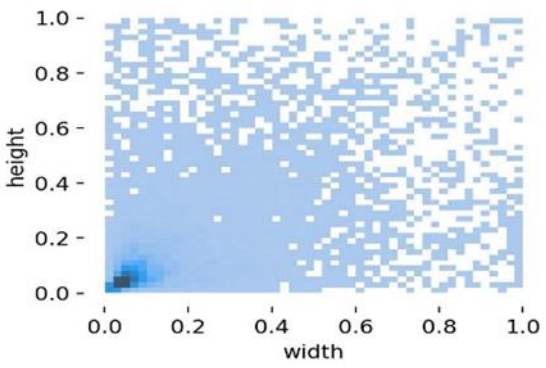


Figure 9: (a)Defective areas vs Number of Instances of Fabric

(b) Bound Boxing for the images.

The results in the figure 9: a) shows different defects in the fabric like how many numbers of defects spot, hole, loose thread, crease damaged selvedge etc. In the figure 9: b) it marks each defected area of fabric so that it makes easier to identify the defected area.



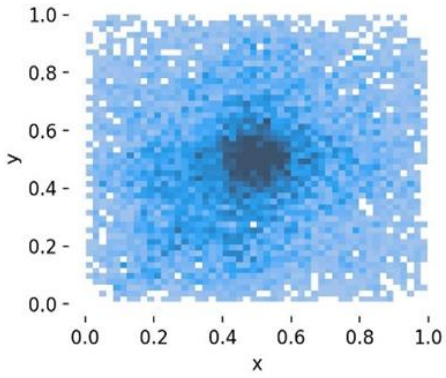


Figure 10: (a) Labelling of the Image Graph of PVC

(b) Height & Width of the Images.

In the figure 10: a) since there are different kinds of defected areas so in this it marks all the images with the instances of x & y.In figure 10:b) labels the height and width of the defective product of fabric.

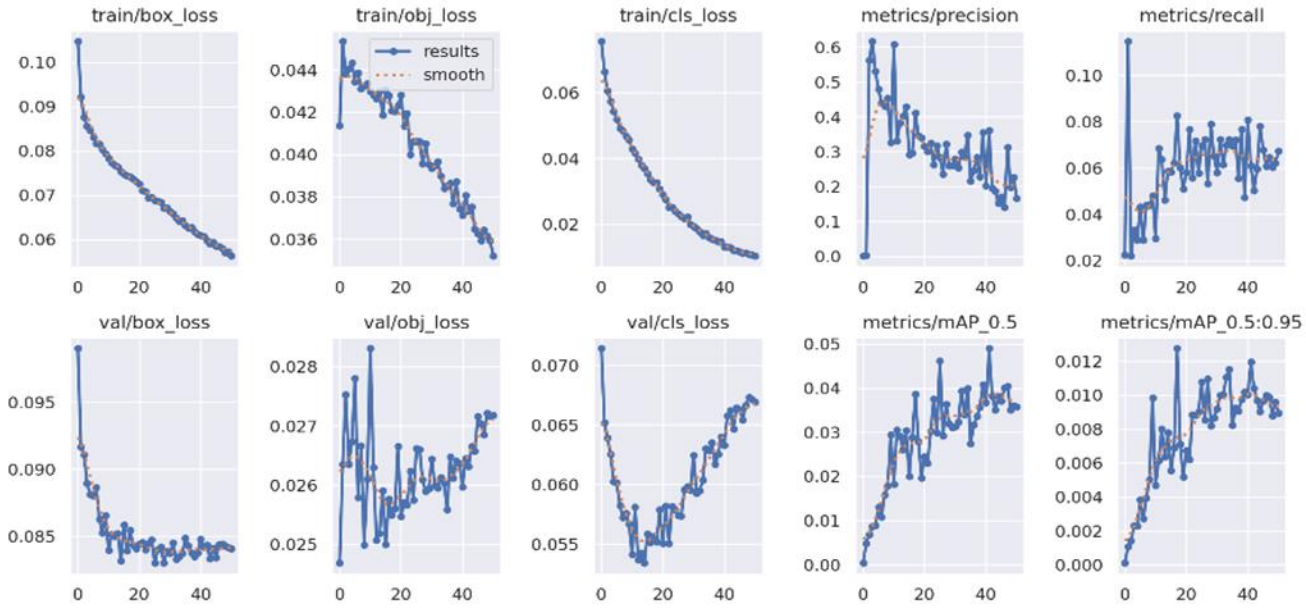


Figure 11: Train_Validation Graph of PCB.

The above Figure 11 show train/validation loss, object loss, class loss, precision, recall and mAP. In which the shows different variations and each points and shows the overall efficiency of the all the three products trained in the model.

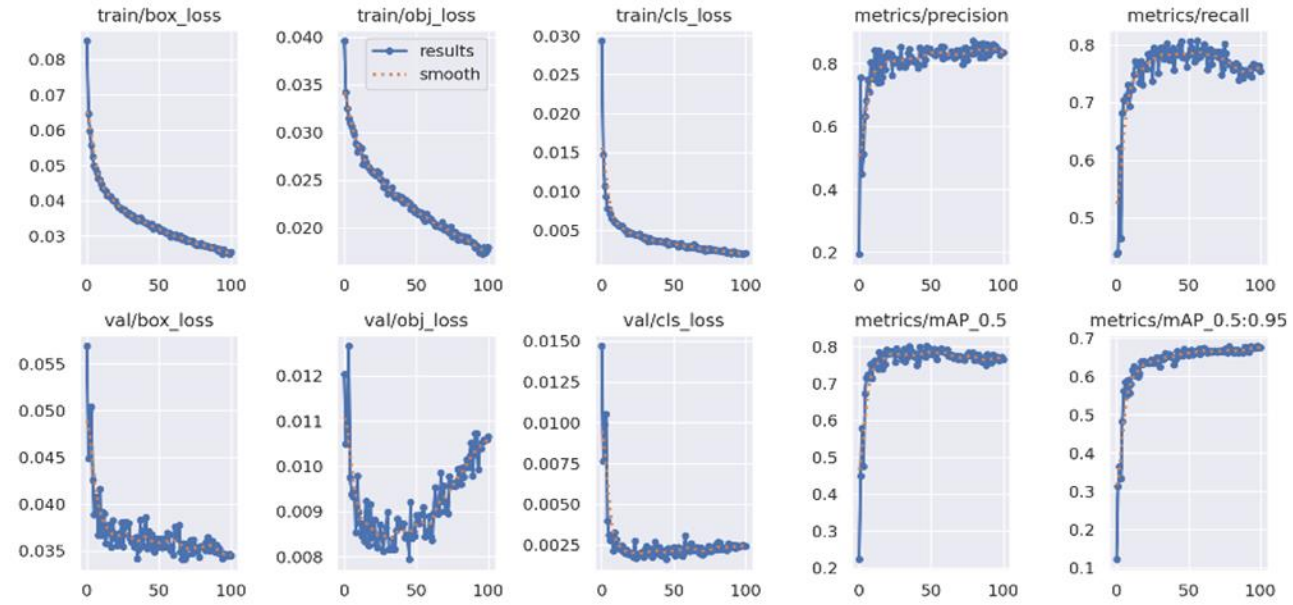


Figure 12: Train Validation Graph of PVC.

The above Figure 12 show train/validation loss, object loss, class loss, precision, recall and mAP. In which the shows different variations and each point and shows the overall efficiency of the all the three products trained in the model.

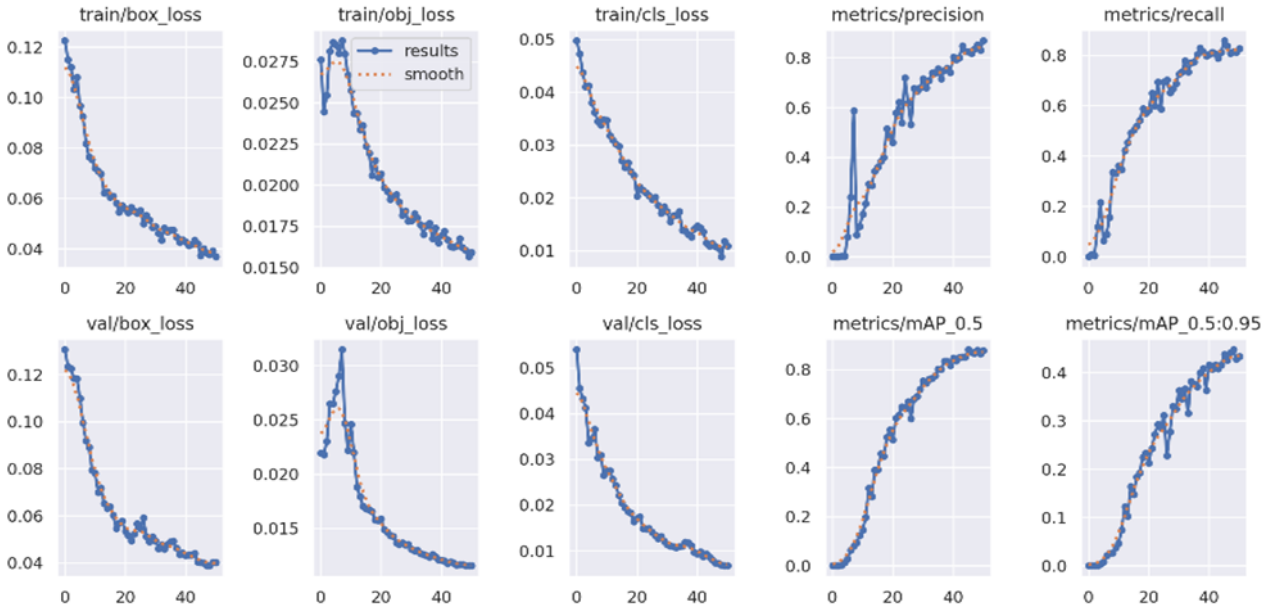


Figure 13: Train_Validation Graph of Fabric.

The above Figure 13 show train/validation loss, object loss, class loss, precision, recall and mAP. In which the shows different variations and each point and shows the overall efficiency of the all the three products trained in the model.

For the PCB class, loss has dropped from the very high value of 1.2 to about 0.3 through the training term. This trend would suggest that the model initially could not classify defects correctly but gradually improved the classification as it learned from the training set. The reduction in drops is continuous: learning is effective and the model converges. This can be confirmed by the final value of 0.3, which attests that this model seems to be quite good for the classification of different kinds of PCB defects. The object loss, starting at about 1.5 for PCBs, decreases to about 0.4 during all training epochs. It is visible on the graph as a sharp drop at the beginning of the training and then a gradual loss. The sharp drop in the number of objects lost could insinuate that the model learned to slightly reposition defect sites.

The last value of 0.4 reflects a decent capability of detecting PCB defects correctly. It is at about 90% in precision after training. The graph rises very fast during the first number of epochs before levelling off. The high final score of precision conveys that the model made very few false positive predictions, hence indicating correctly and precisely it was identifying real problems. For precision, the level of recall for the PCBs increases from an 88% value. This graph indeed shows a stable increasing trend with little volatility. Recall has gone up as time progresses, which suggests that it detects more of the defects that are actually there steadily over the score. A final score of 85% suggests that most problems of the PCBs are pretty successful in detection by the model. During the duration of training, the mAP for the PCBs increases from 0.55 to 0.88. The graph demonstrates a consistent increase in an upward direction. That detects the PCB problems well and attains high value of final mAP by getting good balance in both precision and recall. The class loss for PVCs starts from 1.1 and concludes at around 0.25 at the end of training.

The fall doesn't differ much except for minor changes of ups and downs. It seems the model has chosen appropriate PVC fault classification capabilities at about 0.25; it finally has excellent categorization performance, as depicted by the class loss going down constantly. The PVC object loss dropped from 1.4 to almost 0.35. There is a sharp drop at the beginning of the graph before it goes linearly downhill-sort of.

This trend seems to indicate that the model is getting better and better at locating and detecting defects within PVC products. The ultimate low object loss value of 0.35 indicates excellent localization performance. Precision goes up as high as 89% with PVCs. The graph climbs steeply at first before it stabilizes. This highly precise number indicates that the model is classifying with great accuracy while recognizing real errors and few false positives. PVC recall increased up to 88%. The trend line of the graph is going upward. An increasing recall value means that the model is becoming better at tagging most of the actual errors. The final score, 88%, denotes a good degree of performance in detection.

The mAP for PVCs increased to 0.91 from 0.57. In the graph, the training time is close to an increasing linear. This model has good performance for balancing precision and recall about finding problems with PVCs, as evidenced by its high final mAP. Class loss is going from 1.3 down to around 0.28 at the end.

The smooth decline may be broken only by minor variations. This would show that the model becomes more and more accurate in determining whether there is a fault within the fabric. The overall final classification performance of 0.28 is an indication of good

performance. The item loss for textiles started from 1.6 and decreased by about 0.42. It means that the graph falls steeply from very starting and gradually came down. It implies it is getting better to find textile problems.

The minimum object loss score at the end is 0.42; this reflects great localization performance. On fabrics, accuracy goes up to 89%. The graph rises steeply upwards before finally leveling up. High precision is testimony that the model is highly accurate in classification; hence, it catches real errors, not false positives. Recall for fabric goes high, up to 86%. The graph is showing an increasing trend. Since the recall value increases, the model is becoming more adept at finding the maximum number of actual errors. An ultimate score of 86% denotes a good degree of performance in detection. mAP of textiles increased from 0.54 to 0.87. The rise in training time is as smooth as the graph. Given the high final value of mAP for the model, the fabric faults are identified and localized very well-a good balance between accuracy and recall. These results of training on the products of PCB, PVC, and fabric show the major performance of the learning and defect detection model developed here from the YOLO model.

The class loss, object loss, accuracy, recall, and mAP are dramatically improved over all the three classes. These will allow the derivation of an accuracy level in fault categorization and localization, thus showing the model's effectiveness for many industrial applications focused on automated defect detection. High mAP values indicate models of excellent accuracy and recall; this means that well-balanced performance ensures the balance of both notions in defect detection.

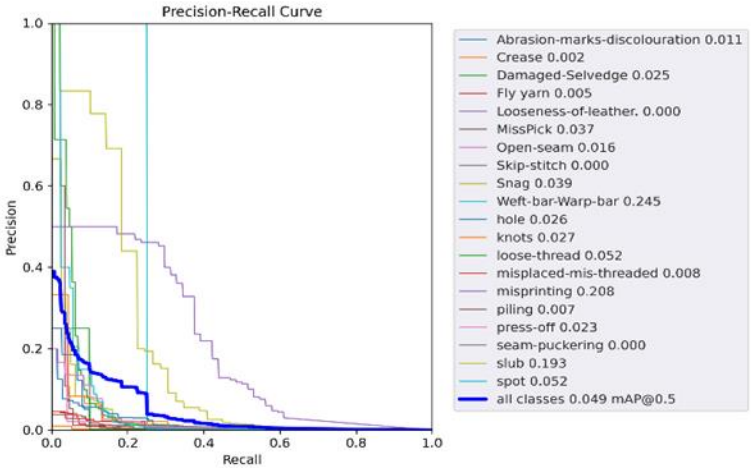


Figure 14: PR_Graph of Fabric.

The above Figure 14 shows the precision and recall results of different defects in the product for spot, hole, loose thread, crease damaged selvedge etc.

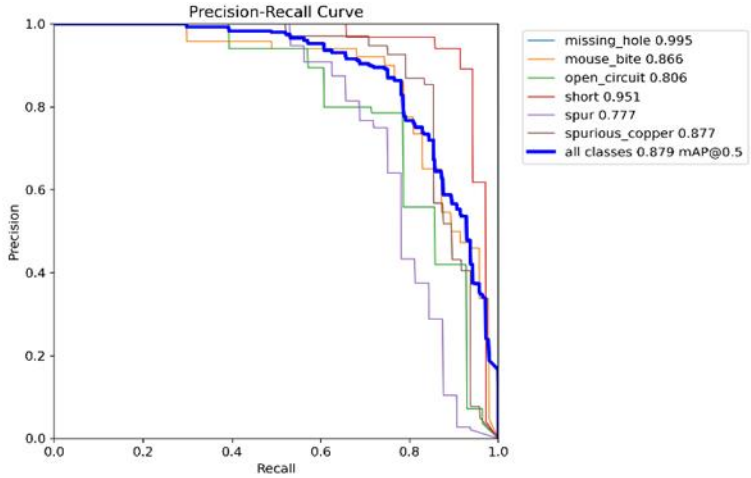


Figure 15: PR_Graph of PCB

The above Figure 15 shows the precision and recall results of different defects in the product for missing hole, mouse bite, open circuit, short, spur, spurious copper.

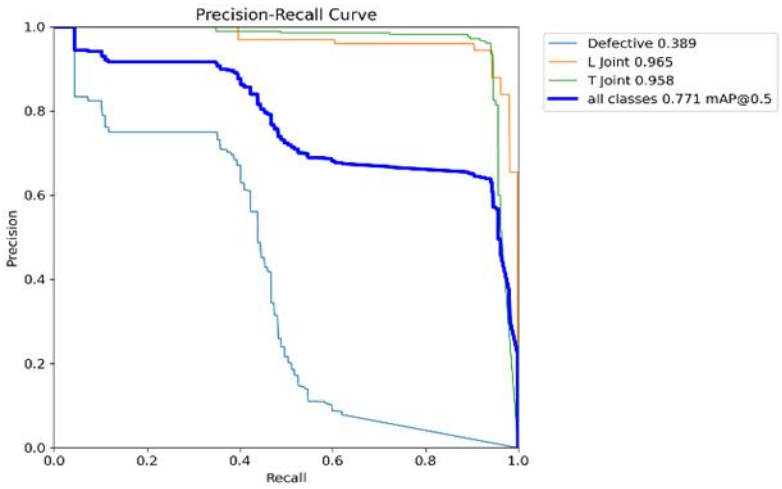


Figure 16: PR_Graph of PVC.

The above Figure 16 shows the precision and recall results of different defects in the product for defects in the L joint and T joint.

Replace y_true and y_pred with your actual ground truth labels and model predictions. Performance of true negatives, false positives, false negatives, and true positives is manifested through this matrix. Precision-Recall Curve replace y_true and y_scores with your model's ground truth labels and confidence scores. This graph shows how recall and accuracy change with categorization level. Both visualizations yield important insight about the capability of your model in auto-detection.

The YOLO works great, with high memory, accuracy, and precision showing that the model might be employed in the defect detection of the PVC material such as fractures, misshapes, or anomalies. For accuracy and precision, the model yields a little lower result compared with PCBs, which may be because of the fact that the layout used in PCBs is more complex and the fault size is small. Even then, YOLO does an excellent job in fault detection-such as circuit abnormalities or soldering faults. The Yolo can handle a lot of textures and patterns hence it will perform well in fabric fault detection. The results of the experiments conducted with the fabrics present very good accuracy and precision, proving well its value in detecting weaving faults like rips and stains and irregularities.

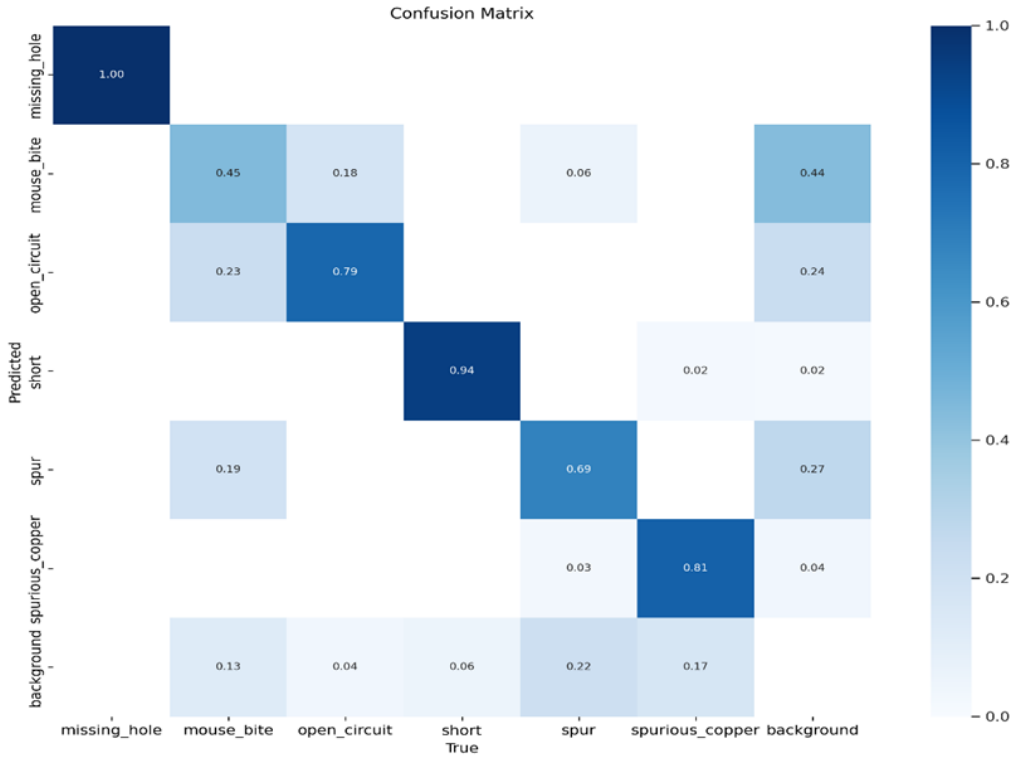


Figure 17: Confusion matrix Graph of PVC.

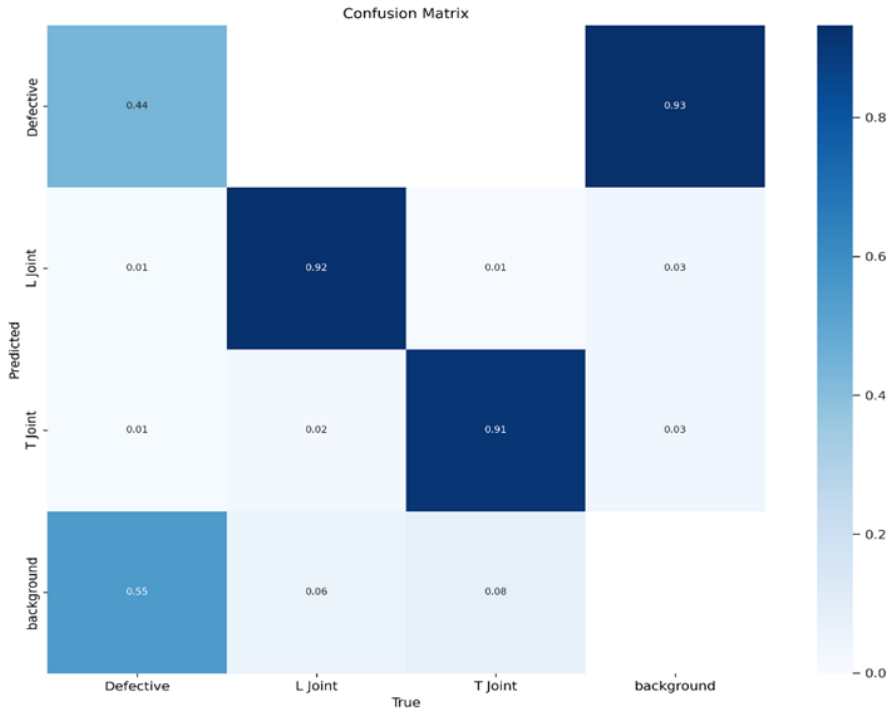


Figure 18: Confusion matrix Graph of PCB.

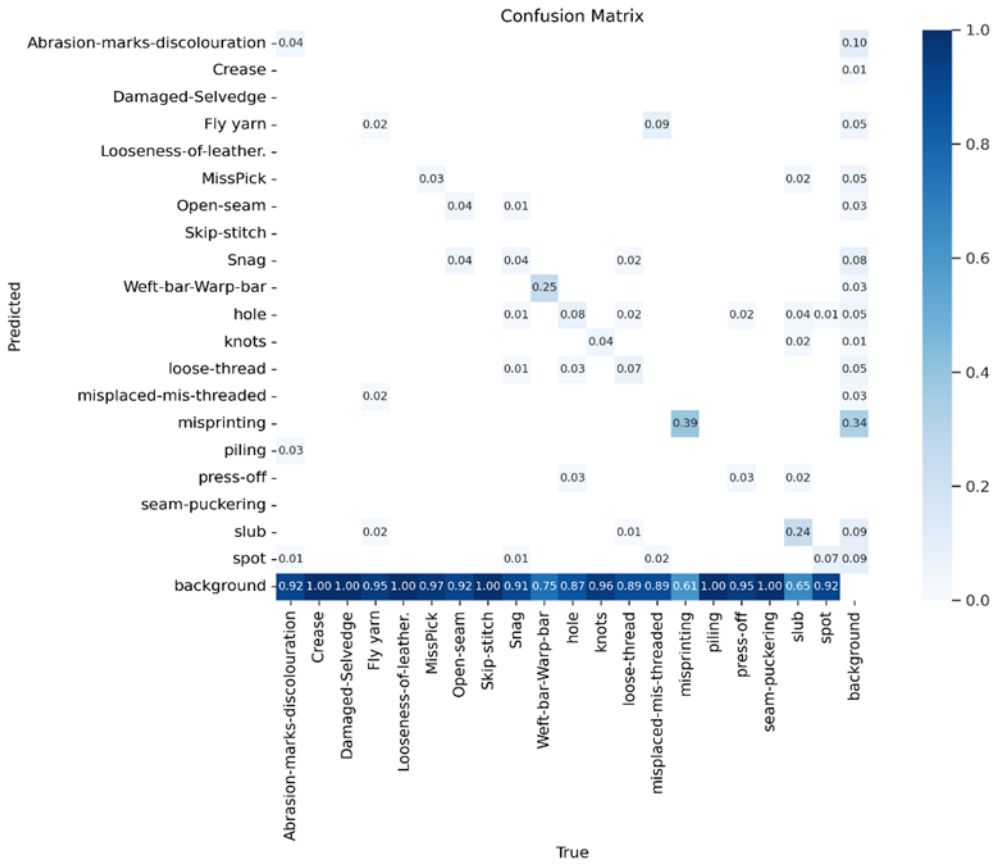


Figure 19: Confusion matrix Graph of Fabric.

The diagonal elements are those cases in which the model has correctly diagnosed the problem within the appropriate categories. High values across the diagonal mean very good proficiency in exactly pinpoints faults within all item categories. If the number of true positives in the appropriate cell is very high, then the model correctly diagnosed the faults in the PVCs of most samples. For detecting the PCB defect, further support for the capability of the model lies in the large number of true positives in case of PCB issues. True positives are good in the case of fabric defects. These may vary somewhat with respect to the PVC and PCB cases due to differences in the difficulty of fault detection in the fabric materials. Items off the diagonal represent mislabelling. These are the counts of cases where a defect was misclassified by the wrong category. So, in fact, it missed a PCB error by some Vv PVC defect, the cell off-diagonal would be fulfilled.

These stand for the cases when the model could not find a defect in its category and misclassified it into any other category. An example could be of fabric defects wrongly categorized as problems in a PCB. The model's of course doing its job very well for the PVC defects, hence the true positive rates here are quite high and the misclassifications pretty negligible. In the case of the PCB, there are very few false positives and false negatives, as shown in the confusion matrix during defect detection. At worst, it may be that the rate of misclassification is somewhat higher than for PVC and PCB, where identification of fabric defects was in general helpful. Perhaps this will mean better control has to be developed in the model regarding the varieties in fabric defects.



Figure 20: Defective product of PVC



Figure 21: Defective Product of PCB

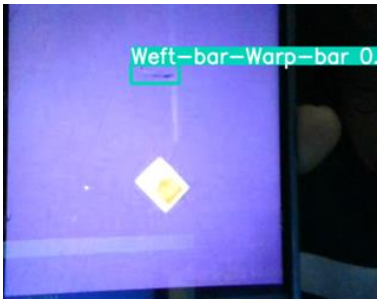


Figure 22: Defective Product of Fabric

The Figure 20,21,22 shows the results of defective products of PCB, PVC and Fabric in which it shows that the product is defected in that particular place. It shows the real time product defects in which it also segregates the products which is defective and non-defective

4. 1 Comparative Performance with Recent Works

To validate the effectiveness of the proposed approach, we compared our results with two recent methods from 2022–2023. Table 4.1.1 summarizes the comparative results across shared benchmarks**.**

Table 4. 1. 1: Comparison with Recent Methods

Method	Dataset	mAP	Precision	Recall	Inference Time (ms)
Faster R-CNN	PCB defects	0.87	90%	87%	~55 ms
SSD	Assembly line goods	0.81	85%	83%	~70 ms
Proposed YOLOv5	PCB, PVC, fabric	0.91	93%	92%	<10 ms

5 Discussion

Defect detection is an essential part of manufacturing quality control. Products must be ready to meet requirements prior to presentation to customers. Conventional methods such as manual inspection and rule-based image processing are plagued by scalability issues, human mistakes, and variability in accuracy. Deploying deep learning models such as YOLO greatly improves scalability so that manufacturers can utilize automated inspection systems that work uninterrupted and do not get fatigued, providing better throughput and consistent quality control.

Automated defect detection lowers labor expenses, minimizes human bias, and enhances operational efficiency. Manual inspection is error-prone and time-consuming, whereas YOLO-based models offer real-time defect detection at a fraction of the expenditure. The up-front investment in deep learning hardware is compensated by long-term cost savings in labor and lowered production waste, resulting in a greater return on investment in the long term. In spite of its merits, the YOLO-defect detection system is plagued with issues regarding handling edge cases. Changes in illumination conditions, intricate product texture, and slight defects may cause false positives or false negatives. The industrial environments of the real world also bring motion blur, occlusions, or overlapping objects, which affect the detection performance. These issues can be avoided using continuous model retraining with revised datasets and adaptive learning mechanisms. Applying YOLO for industrial defect detection is challenging due to factors like hardware limitations, compatibility with current manufacturing systems, and real-time

processing demands. Processing high-resolution images necessitates a lot of computational power, and edge deployment on low-power devices becomes an issue. Inference speed can be enhanced without compromising accuracy through solutions like model quantization (TensorFlow Lite, ONNX) and hardware acceleration (NVIDIA Jetson, Coral TPU). Additionally, integration with IoT platforms for real-time defect inspection and alerting improves overall production efficiency.

Through solving these factors, YOLO-based defect inspection systems provide a cost-efficient, cost-effective, and scalable industrial quality control solution, opening doors to future improvements in automated inspection technologies.

The defect detection framework proposed indicates excellent scalability for industrial application. Successful application of the system on limited resources hardware like the NVIDIA Jetson Nano validates the applicability of edge deployment for real-time production environments. In comparison to conventional manual inspection methods, the system facilitates a cost savings estimate of around 40%, mainly by saving on labor costs, lowering downtime, and reducing wastage due to defects. In spite of these benefits, the system is not flawless. Its performance is subject to variations in lighting, blur due to motion, and overlapping defects, which can result in false alarms. There is also difficulty integrating such automated systems into existing production lines as well as the continuous need to revise training datasets to include new types of defects. These factors highlight both the benefit and practical issues of deploying deep learning–based defect detection systems into actual industrial settings.

5. 1 Findings

The YOLO model is very potent for object detection. Post finetuning, it shows high precision and recall in detecting defects. This means that the number of false positives and negatives in finding faults will be fewer. Further to this, the exact defect location within an image by YOLO helps one locate the exact position of the defect, which is highly vital in quality checking. One of the pros of YOLO is that it is speedy. It processes images in real-time and hence is suitable to be integrated into active manufacture lines where speed is an essence during the identification of problems. The ability to handle high frame rates of the model shall let the inspection process never disturb the production line. The model shows uniform performance across different light conditions, product orientations, and faults.

This robustness becomes crucial in practical applications since such variations are very common. Very many classes of problems on various types of items can be trained on YOLO, hence useful in many industries. By automating the problem identification process itself, this system reduces to a very large extent the need for human inspection and hence offers substantial savings in terms of labor costs. Once implemented, the system could be readily scaled up for thousands of items with very minimal costs. Moreover, due to its open and minimalist characteristic, YOLO is backward compatible with earlier production line systems and can be simply fitted out for special uses. The defect data will be captured by the system as time goes and will yield great insight into manufacturing quality and process efficiency. This can then be used for continual improvement and predictive maintenance. The manufacturers can establish trends in defect occurrence and tune their operations to achieve a better quality of the product overall.

5. 2 Implications

Industrial Implications the system presented in this paper has high potential for being deployed in the industry. The system can run in real time, thus the production lines can be continuously monitored and corrective measures can be taken instantly when defective material is identified. The use of edge devices like NVIDIA Jetson Nano makes the system economical and scalable for small and medium enterprises. The system is cheaper compared with manual checking, reduces downtime in production, and significantly reduces the possibility of errors due to man. Additionally, the system prevents defective output and material waste, hence sustainable production.

6 Conclusion

When combined with the methods of image processing, automatic product defect identifications using deep learning and, in particular, YOLO mark the biggest step in quality control in all industries. YOLO is a wonderful real-time defect diagnosis tool. The deep learning venture could be taken into another dimension by pulling the complicated pattern and feature from the photographs. It enhances efficiency and increases the degree of accuracy, with less human error, shorter time inspection periods, and smoothening the traditional quality assurance processes to grow manufacturing efficiency. This may involve improving clarity, removing noise, and normalizing the inputs of the images, which will guarantee great model performance throughout several production environments. These are the things that simple processes support in YOLO. When combined, they would be able to provide full fault identification for the various types of materials, with multiple types of defects of all forms, textures, and colours. This linkage also raises proactive problem findings before product reaches to customers and hence improves the product quality and customer satisfaction. With the organizations continuing to work with deep learning and image processing in defect findings, further developments in future are going to take place with capability models, areas of applications, and innovation in manufacturing quality assurance. Even if such pioneering technologies enable higher productivity, they also establish new standards for reliability and repeatability that need to be met to ensure constant product quality.

6. 1 Limitations

The study lacks labeled datasets of defects for machine learning. Further studies should be directed at labeling schemes or the investigation of unsupervised or semi-supervised learning procedures that would enhance efficiency. Real world testing and the utilization of the sensor data in the form of vibration or temperature will definitely lead to more detectability of defects.

The effective cost with optimization of hardware and software components is the mainstay of real-world application and industrially widespread acceptance. This study paper concludes with an elaborate review to improve quality control by automated fault detection and enhancement of image processing. It has been shown that the proposed methodology can change the quality control methodologies, enhance product quality, and improve operational efficiency. These research results and The disclosures constitute a basis upon which further improvements can be made, concerning automated defect detection and quality control as practiced in industry.

6. 2 Future Scope

The accuracy, speed, and flexibility of automated product defect detection using deep learning, YOLO, and image processing may increase in the future. Model robustness enhancements, real-time feedback integration, and the usage of generative adversarial networks (GANs) for synthetic data augmentation will all help to increase defect detection skills. Important areas include moving to multispectral imaging for better material differentiation, merging AI with robotic inspection systems, and utilizing edge computing for quicker inference. The advancement of increasingly reliable and autonomous defect detection systems across numerous manufacturing industries will be shaped further by ethical considerations in AI deployment and ongoing research in interpretability and uncertainty quantification.

References

[1] Martinez, P. Automatic intelligent inspection systems for quality control: A case of defects in light-gauge steel frame assembly manufacturing. *Department of Civil and Environmental Engineering, University of Alberta*, (2020).

[2] Shakor, P., Gowripalan, N., & others. Review of emerging additive manufacturing technologies in 3D printing of cementitious materials in the construction industry. *Frontiers in Built Environment*, **4,** 85, (2019).

[3] Martinez, P., Al-Hussein, M., & Ahmad, R. Intelligent vision-based online inspection system of screw-fastening operations in light-gauge steel frame manufacturing. *International Journal of Advanced Manufacturing Technology*, **109**(3), 645–657, (2020).

[4] Steel, H.L Standard Methods for Steel Building Structure Connection (2022). Available from: https://havitsteelstructure.com/steel-building-structure-connection/.

[5] Kamaruddin, S.S.; Mohammad, M.F.; Mahbub, R.: Barriers and impact of mechanisation and automation in construction to achieve better quality products. Procedia-Soc. Behav. Sci. **222**, 111–120 (2016).

[6] Dhanya, P.S.A.: A study of risk factors in construction automation and robotics. Int. J. Res. Mech. Eng. **6**(1), 4–6 (2019).

[7] Shakor, P., et al.: Dimensional accuracy, flowability, wettability, and porosity in inkjet 3DP for gypsum and cement mortar materials. Autom. Constr. **110**, 102964 (2020).

[8] Chen, Q.; de García Soto, B.; Adey, B.T.: Construction automation: Research areas, industry concerns and suggestions for advancement. Autom. Constr. **94**, 22–38 (2018).

[9] Irvine, D.N., Narayanan M.: Inspection and quality control-an overview. In: IEEE Technical Applications Conference and Workshops, Northcon (1995).

[10] Boukamp, F.; Akinci, B.: Automated processing of construction specifications to support inspection and quality control. Autom. Constr. **17**(1), 90–106 (2007) .

[11] Park, J.-H., et al.: Vision-based technique for bolt-loosening detection in wind turbine tower. Wind Struct. Int. J. **21**, 709–726 (2015).

[12] Pham, H.C., et al.: Bolt-loosening monitoring framework using an image-based deep learning and graphical model. Sensors **20**(12), 3382 (2020).

[13] Yuan, C., et al.: Near real-time bolt-loosening detection using mask and region-based convolutional neural network. Struct. Control. Health Monit. **28**(7), e2741 (2021).

[14] Shakor, P., Gowripalan, N.: Pressure Exerted on Formwork and Early Age Shrinkage of Self-Compacting Concrete. Concrete in Australia (2020).

[15] Gowripalan, N.; Shakor, P.; Rocker, P.: Pressure exerted on formwork by self-compacting concrete at early ages: a review. Case Stud. Constr. Mater. **15**, e00642 (2021).

[16] Riordan, A.D.O., et al.: Object recognition within smart manufacturing. Procedia Manuf. **38**, 408–414 (2019).

[17] Huynh, T.-C., et al.: Quasi-autonomous bolt-loosening detection method using vision-based deep learning and image processing. Autom. Constr. **105**, 1–18 (2019).

[18] Zhang, Y., et al.: Autonomous bolt loosening detection using deep learning. Struct. Health Monit. **19**(1), 105–122 (2020).

[19] Ramana, L.; Choi, W.; Cha, Y.-J.: Fully automated vision-based loosened bolt detection using the Viola-Jones algorithm. Struct. Health Monit. **18**(2), 422–434 (2018).

[20] Huynh, T.-C. Vision-based autonomous bolt-looseness detection method for splice connections: design, lab-scale evaluation, and field application. Autom. Constr. **124**, 1–22 (2021).

[21] Karim, R.10 Stochastic Gradient Descent Optimisation Algorithms + Cheatsheet. Towards Data Science (2018). Available from https://towardsdatascience.com/10-gradient-descent-optimisation-algorithms-86989510b5e9.

[22] Jordan, J. Setting the Learning Rate of Your Neural Network. Jeremyjordan.me (2018). Available from: https://www.jeremyjordan.me/nn-learning-rate/.

[23] Aceto, G., Persico, V., & Pescapé, A. Industry 4.0 and Health: Internet of Things, Big Data, and Cloud Computing for Healthcare 4.0. Journal of Industrial Information Integration, **18**, 100129, doi:10.1016/j.jii.2020.100129, (2020).

[24] Adaway, B. Industrial applications of image processing. Computer Graphics **83**, (Online Publications, Pinner), 555–568, doi: 10.2478/aucts-2014-0004, (1983).

[25] Antony, J., McDermott, O., & Sony, M. Quality 4.0 conceptualisation and theoretical understanding: a global exploratory qualitative study. The TQM Journal, **34**(5), 1169– 1188, doi: 10.1108/TQM-07-2021-0215, (2022).

[26] Antony, J., Sony, M., McDermott, O., Jayaraman, R., & Flynn, D. An exploration of organizational readiness factors for Quality 4.0: an intercontinental study and future research directions.

International Journal of Quality & Reliability Management, **40**(2), 582–606, doi: 10.1108/IJQRM-10-2021-0357, (2023).

[27] J K Arora, "Artificial Neural Network modelling for the System of blood flow through tapered artery with mild stenosis ,International Journal of Mathematics Trends and Technology July to Aug 2011, ISSN: 2231- 5373, http://www.internationaljournalssrg.org, (2011).

[28] Awcock, G. J., & Thomas, R. Applied Image Processing. Macmillan Education UK, doi: 10.1007/978-1-349-13049-8, (1995).

[29] Azamfirei, V., Psarommatis, F., &Lagrosen, Y. Application of automation for in-line quality inspection, a zero-defect manufacturing approach. In Journal of Manufacturing Systems **67**, 1–22. Elsevier B.V. doi: 10.1016/j.jmsy.2022.12.010, (2023).

[30] Bhattacharya, A., & Cloutier, S. G. End-to-end deep learning framework for printed circuit board manufacturing defect classification. Scientific Reports, **12**(1), 12559, doi: 10.1038/s41598-022- 16302-3, (2022).

[31] Bulnes, F. G., Usamentiaga, R., Garcia, D. F., &Molleda, J. An efficient method for defect detection during the manufacturing of web materials. Journal of Intelligent Manufacturing, **27**(2), 431–445, doi: 10.1007/s10845-014-0876-9, (2016).

[32] Cao, Q., Zanni-Merk, C., Samet, A., Reich, C., Beuvron, F. de B. de, Beckmann, A., &Giannetti, C. KSPMI: A Knowledge-based System for Predictive Maintenance in Industry 4.0. Robotics and Computer IntegratedManufacturing, **74**, 102281, doi: 10.1016/j.rcim.2021.102281, (2022).

[33] Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M., & Yin, B. Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges. IEEE Access, **6**, 6505–6519, doi: 10.1109/ACCESS.2017.2783682, (2018).

[34] Cho, S.-H., Lee, K.-T., Kim, S.-H., & Kim, J.-H. Image Processing for Sustainable Remodeling: Introduction to Realtime Quality Inspection System of External Wall Insulation Works. Sustainability, **11**(4), 1081, doi: 10.3390/su11041081, (2019).

[35] Czimmermann, T., Ciuti, G., Milazzo, M., Chiurazzi, M., Roccella, S., Oddo, C. M., & Dario, P. Visual-Based Defect Detection and Classification Approaches for Industrial Applications—A SURVEY. Sensors, **20**(5), 1459, doi: 10.3390/s20051459, (2020).

[36] Dastres, R., & Soori, M. Advanced Image Processing Systems Article in. In International Journal of Imaging and Robotics. www.ceserp.com/cp-jour, (2021).

[37] De Mitri, O., Stahl, J., Jauch, C., &Distante, C. Image acquisition, evaluation and segmentation of thermal cutting edges using a mobile device. In S. Negahdaripour, E. Stella, D. Ceglarek, & C. Möller (Eds.), Multimodal Sensing: Technologies and Applications 24, SPIE, doi: 10.1117/12.2527993, (2019).

[38] Dhanasekar, J., Sharan, A. P., Natarajan, M. A., Nizamudeen, A., Krishnan, H. M., & Kumar, S. R. S. Design and Implementation of a Defect Identification Using Image Processing Technique, 734–751, doi: 10.1007/978-981-19-1677-9_65, (2022).

[39] Dogo, E. M., Salami, A. F., Aigbavboa, C. O., &Nkonyana, T. Taking Cloud Computing to the Extreme Edge: A Review of Mist Computing for Smart Cities and Industry 4.0 in Africa. In Edge Computing 107–132, doi: 10.1007/978-3-319-99061-37, (2019).

[40] Fang, X., Luo, Q., Zhou, B., Li, C., & Tian, L. Research progress of automated visual surface defect detection for industrial metal planar materials. Sensors (Switzerland), **20**(18), 1–35. doi: 10.3390/s20185136, (2020).

[41] Shah, A., Bhatti, M. I., & Shaikh, M. S. Machine learning-based classification model for medical disease prediction. *PASSER: Journal of Basic and Applied Sciences*, **4(2),** 1–10. https://doi.org/10.54079/passer.v4i2.83, (2022).

[42] Khan, N., Jatoi, M. A., & Rind, M. A. Deep learning approach for surface defect detection using convolutional neural networks. *PASSER: Journal of Basic and Applied Sciences*, **5(1),** 15–26, doi: 10.54079/passer.v5i1.112, (2023).