Fabric Detection in Real-World Manufacturing Using YOLOv5-Transformer Models

Makara Mao[1] **Keovichear Ouk**[1] **Hongly Va**[2] **Min Hong**[3]
[1]Department of Information Technology and Engineering, Royal University of Phnom Penh, Cambodia
[2]Department of Computer Science, Cambodia Academy of Digital Technology, Cambodia
[3]Department of Computer Software Engineering, Soonchunhyang University, Asan-si, 31538, Republic of Korea
mao.makara@rupp.edu.kh

Abstract

Detecting fabric defects, especially in textiles with complex textures, presents significant challenges due to the intricate nature of fabric patterns. Among various object detection methods, the YOLO algorithm is renowned for its real-time performance and accuracy. By treating object detection as a single regression problem, YOLO predicts bounding boxes and class probabilities from an entire image in one pass. This paper proposes a novel approach for fabric detection using the YOLOv5-Transformer model, which integrates Transformer architecture to enhance defect detection in textiles. YOLOv5, a fully convolutional neural network, strikes an optimal balance between speed and accuracy in end-to-end detection tasks. Leveraging the latest advancements in deep learning, YOLO achieves high detection speeds without significantly compromising precision, making it ideal for real-world applications. Our proposed YOLOv5-Transformer model surpasses other YOLOv5 variants, achieving an accuracy of 82.9%, representing a 5.6% improvement over YOLOv5s and YOLOv5n and a 2.7%–3.2% improvement over other versions YOLOv5m, YOLOv5l, YOLOv5x. Comparative performance metrics are also presented, including processing time on GPU, precision, recall, and F1 score.

Keywords: *YOLOv5, Object Detection, Fabric Dataset, Textile Materials.*

1 Introduction

With the rapid growth of artificial intelligence (AI) technology, there has been increasing interest in applying AI solutions to various industrial challenges, particularly fabric manufacturing. The textile industry faces significant challenges in fabric quality control due to the intricate patterns, textures, and designs involved [1]. Traditionally, manual inspection has been used for defect detection, but it is labor-intensive, time-consuming, and prone to human error. To overcome these limitations, computer vision techniques have emerged as powerful tools for automating fabric defect detection, significantly improving accuracy and efficiency [2]. Among these techniques, object detection algorithms like You Only Look Once (YOLO) have gained prominence due to their real-time performance and ability to handle large-scale data [3]. YOLO's strength lies in treating object detection as a single regression problem, predicting object locations and classifications in a single pass through the network. However, detecting fabric defects remains challenging due to textiles' complex textures and subtle irregularities [4].

Convolutional neural networks (CNNs) have become the dominant model in computer vision, excelling in tasks like image classification, object detection, and semantic segmentation [5]. Existing CNN-based object detection models can be divided into one-stage and two-stage detectors.

One-stage detectors, such as the YOLO family of models and single-shot multi-box detectors (SSD), treat object detection as a straightforward regression problem. These models achieve fast inference speeds by directly predicting bounding boxes and class labels in a single step, making those ideal for real-time applications [6]. However, it may sacrifice some accuracy, especially when dealing with minor or overlapping objects. Two-stage detectors, such as Faster R-CNN [7], offer higher accuracy by separating the detection process into two stages: the region proposal network (RPN) identifies candidate object regions,

and a second stage refines these proposals and classifies the objects [8][9]. While more accurate, two-stage models are often slower and more computationally intensive.

This paper presents the application of The development of the YOLOv5-Transformer model, which combines YOLOv5's object detection capabilities with the Transformer's multi-scale information fusion, and a detailed comparison of performance metrics, including processing time on (CPU and GPU), precision, recall, F1 score, and frames per second (FPS) across different models.

2 Related Work

Object detection is a critical task in computer vision that involves both localizing objects within an image and classifying them into predefined categories. The algorithm typically returns bounding boxes, confidence scores, and class labels for detected objects [8]. Recent advances in deep learning have led to the development of highly efficient and accurate object detection models, categorized primarily into one-stage and two-stage detectors.

In addition to real-world data, simulation datasets have become essential for training and evaluating these models, particularly when obtaining diverse or labeled real-world data is challenging. Simulated datasets offer controlled environments to replicate varying conditions, such as lighting changes, object occlusions, and different viewpoints, ensuring robust model performance across scenarios [9].

Processing these datasets involves data augmentation techniques, including scaling, rotation, and noise addition, which help improve model generalization. Proper pre-processing tasks, including normalization and converting annotations into formats such as COCO or YOLO, ensure compatibility with detection frameworks. Leveraging simulations accelerates algorithm development and provides a cost-effective way to test object detectors under complex or rare conditions without relying entirely on physical data collection [10].

One-stage detectors, such as YOLO and single-shot multi-box detectors (SSD), prioritize speed by directly predicting object locations and classifications from the image in a single network pass [11]. YOLOv5, in particular, is a widely used model for its high speed and relatively high accuracy. It uses convolutional neural networks (CNNs) to extract spatial features from the image, enabling fast real-time detection [12].

YOLOv5 employs a feature pyramid network (FPN) to detect objects at different scales, making it suitable for multi-scale object detection tasks. It divides the image into grid cells, predicting bounding boxes and confidence scores for each cell. Despite its speed, YOLOv5 may encounter difficulties with small or occluded objects due to its reliance on local image features and relatively limited receptive field in the CNN layers.

Another one, two-stage detectors like Faster R-CNN are designed for higher accuracy, though they may be slower compared to one-stage detectors [13]. These models generate region proposals (potential object locations) and then refine those proposals through a second-stage classification and bounding box regression.

The advantage of two-stage detectors lies in their ability to provide more precise localization, as the second stage refines the predictions made in the first stage. While two-stage detectors typically offer higher accuracy, they are computationally more expensive due to the added complexity of region proposal generation and refinement [14].

The proposed YOLOv5-Transformer model builds upon the advantages of the YOLOv5 architecture and Transformer networks. By combining the fast inference speed of YOLOv5 with Transformers enhanced global context capture ability, this model aims to balance speed and accuracy, particularly for detecting torn fabric defects [15].

YOLOv5 Backbone: The backbone consists of convolutional layers that extract hierarchical features from the input image. These features are passed through neck architecture, like the FPN, which can detect objects at different scales. Transformer Integration: The Transformer network is integrated into the detection pipeline to enlarge the receptive field beyond the local region CNNs covers. Transformers excel at modeling long-range dependencies in the image, enabling the model to effectively capture regional and global features [16].

This is especially important for fabric defect detection, where defects might span across large areas or appear at multiple scales. Two-Stage Refinement: After the initial predictions

by the YOLOv5 backbone, a second stage refines the predictions by leveraging the global context modeled by the Transformer.

This refinement stage helps improve the accuracy of bounding box localization, especially for challenging cases like overlapping or minor defects on complex fabric textures [17].

3 Methodology

In this section, we present a comprehensive overview of the model architecture of YOLOv5 in Figure 1. Our innovative model integrates YOLOv5 with a Transformer model to enhance accuracy and speed without compromising efficient object detection performance.

Specifically, we have incorporated Transformer layers into the YOLOv5 backbone to augment the model's capacity to capture global contextual information and long-range dependencies within the image.

This hybrid architecture combines the rapid, one-stage detection of YOLOv5 with the attention mechanisms of Transformers, enabling the model to discern objects in intricate, crowded scenes better. By integrating the Transformer model, our objective is to achieve enhanced precision in detection, particularly in demanding scenarios, while capitalizing on the real-time capabilities of YOLOv5.

Transformers have become increasingly popular in computer vision due to their capability to capture global context and long-range dependencies, making them particularly valuable for object detection.

The Transformer model divides an image into small patches, each flattened into a token and passed through multiple self-attention layers. Through this process, the self-attention mechanism computes relationships between patches, enabling the model to comprehend global dependencies and focus on different parts of the image.

Multi-head attention is applied in parallel to capture diverse aspects of the image, and the results are then processed through feed-forward layers with layer normalization to stabilize the training process. This approach assists the model in accurately predicting object classes and bounding boxes using learnable object queries.

In our integrated architecture, the features extracted from YOLOv5's backbone are segmented into patches and passed through Transformer layers to enhance the model's capacity to capture global context. The Transformer outputs are combined with YOLOv5's feature pyramid network (FPN) to enhance multi-scale detection, enabling the model better to detect small, occluded, or overlapping objects.

This hybrid approach combines the speed and efficiency of YOLOv5 with the attention-driven accuracy of the Transformer, resulting in a powerful and context-aware object detection model.

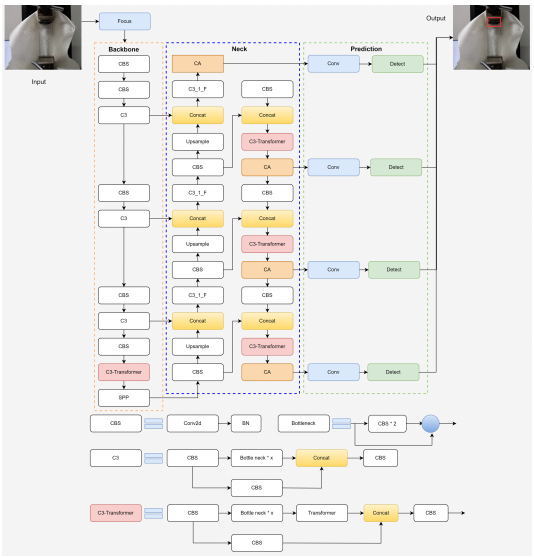

Figure 1. The architecture of the Transformer model.

3.1 Image Labeling

We utilized LabelMe, a versatile graphical image annotation tool depicted in Figure 2, to meticulously create polygonal annotations for object detection tasks. LabelMe's flexibility makes it well-suited for preparing datasets for different YOLO versions. By manually tracing polygons around objects in the images, we could accurately annotate object boundaries, which is crucial for training precise YOLOv5 models. These detailed annotations enable the model to detect objects with great accuracy. Once the annotations were completed, LabelMe generated JSON files containing the annotation data, which we converted into the required format for YOLOv5.

LabelMe's integration with Python also facilitated the seamless incorporation of data augmentation techniques. We utilized scripts to apply transformations such as rotation, scaling, and color adjustments directly to the annotated im-

ages and their corresponding JSON files. This process created an augmented dataset, enhancing the model's robustness and generalization capabilities. This integration has simplified the process of preparing and augmenting the dataset, ultimately contributing to the exceptional performance of YOLOv5 in various object detection scenarios.

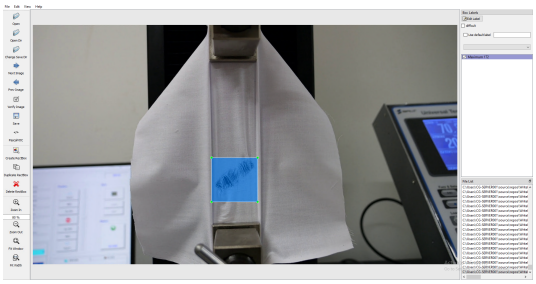

Figure 2. Labeling on Image by using the LabelMe tool.

3.2 Dataset

The dataset discussed in this section, captured using the LUMIX GH6 camera, is a crucial resource for fabric category experimentation. It consists of 7,620 images, the dataset was divided into 80% for training, 10% for validation, and 10% for testing to ensure balanced model evaluation, each categorized into one of five distinct fabric groups: Cotton Fabric Plain, Fabric Wide Hanbok Fabric Nobang DTP, Cotton Yarn-Dyed Check Stripe Plain Fabric, Hanbok Fabric, and Cotton Blend Plain Fabric. This rich diversity in fabric types facilitates comprehensive training and testing of object detection models on YOLOv5-Transformer.

3.3 Data Augmentation

To enhance the diversity and robustness of the fabric dataset, we employed a comprehensive set of data augmentation techniques, as illustrated in Figure 3. These techniques included horizontal and vertical flips to mirror the images, random rotations within a range of -45 to +45 degrees, Gaussian blurring to simulate out-of-focus conditions, and brightness adjustments varying from -22% to +22% for different lighting scenarios.

Additionally, we introduced random noise to mimic real-world imaging variations and darkened images to simulate low-light environments. We converted images to grayscale to focus on texture and pattern rather than color. Cropping was also applied to emphasize different parts of the fabric, aiding the model in learning from diverse perspectives. Our experiment focused on the transformations within the green polygons highlighted in Figure 3. Specifically, we examined random rotations and brightness adjustments to simulate real-world variations and enhance the model's accuracy in classifying fabrics under diverse conditions.

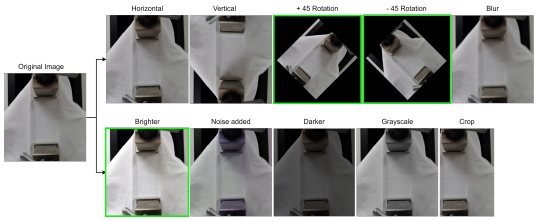

Figure 3. The techniques for applying the data augmentation.

4 Results and Discussion

In this section, we discuss the fabric prediction model based on YOLOv5-Transformer. We used a batch size of 32 for training to balance computational efficiency and model accuracy. A weight decay of 0.0005 was also employed to regularize the model and mitigate overfitting by penalizing large weights. We standardized the input image size to 320 x 320 pixels.

Training was carried out over 100 epochs to allow the model ample iterations to learn from the data. A learning rate of 0.0001 was chosen to ensure steady and reliable convergence. We opted for the stochastic gradient descent (SGD) optimizer due to its effectiveness in large-scale machine-learning tasks. We configured eight workers to parallelize data loading and expedite the training process.

Table 1. Experiment environment.

Component	Specification
Operating System	Windows 10 (64-bit)
Python Version	3.12.4
PyTorch Version	2.3.1
CUDA Toolkit	11.8
cuDNN Version	8.9.7
CPU	AMD Ryzen 9 5900X
GPU	NVIDIA GeForce RTX 4060
RAM	128 GB DDR4

In our training experiments, we compared the time performance of YOLOv5 and YOLOv5-Transformer on both CPU and GPU, as shown in Table 2. In addition to inference time, we now include the number of floating-point operations (GFLOPs) and model parameters for fair comparison.

The proposed YOLOv5-Transformer model maintains a moderate computational cost (43.3 GFLOPs, 20.4 M parameters), which is lower than YOLOv5m (47.9 GFLOPs, 21.2 M) and substantially lighter than YOLOv5x (203.8 GFLOPs, 86.7 M). Despite the integration of Transformer layers, our model achieved faster run-time on both CPU and GPU.

This efficiency improvement can be attributed to two main factors: (1) the Transformer block effectively enhances feature representation without significantly increasing convolutional complexity, thereby reducing redundant computations during feature extraction; and (2) the optimized feature fusion and reduced number of convolutional layers in the neck lead to fewer sequential operations, improving parallel processing efficiency on GPUs.

Consequently, YOLOv5-Transformer achieves a 69.61% CPU and 57.34% GPU speed improvement over YOLOv5x, while maintaining high accuracy. These results confirm that our design enhances computational efficiency without sacrificing model precision.

Table 3 compares the performance of various YOLOv5 variants and the proposed YOLOv5-Transformer model based on GPU latency and frames per second (FPS).

The proposed YOLOv5-Transformer achieves a high mean average precision (mAP) of 82.9%, representing an improvement of 2.2–4.4% over other YOLOv5 models, while maintaining a competitive FPS of 100.33 and GPU latency of 3183 ms.

Although the YOLOv5-Transformer has higher GFLOPs (43.3) than YOLOv5s (15.8) and YOLOv5n (4.1), its FPS remains comparable. This seemingly counterintuitive result arises because FLOPs alone do not fully reflect real-time performance. The Transformer block enhances feature representation without sub-stantially increasing memory transfer or kernel launch overhead, allowing more efficient parallelization on GPUs.

Moreover, our model employs optimized tensor operations and fewer sequential convolutional layers in the neck, reducing pipeline bottlenecks. As a result, the GPU utilization is higher and inference time remains stable, even with moderately increased computational complexity.

5 Conclusions

In this paper, we propose the YOLOv5-Transformer algorithm for detecting torn paths on fabric datasets from a material testing machine captured by a camera. This work aims to improve the existing YOLOv5 algorithm.

Our approach combines a convolutional network and a Transformer to design a new model within YOLOv5 and validate several improved measures to enhance YOLOv5's performance in fabric detection. Specifically, our proposed YOLOv5-Transformer module integrates the local observation capabilities of ConvNext and the global analysis capabilities of the Transformer, making a more significant contribution to improving detection accuracy compared to the original YOLOv5 module.

Additionally, we integrate the YOLOv5-Transformer module to reduce interference from background information, allowing the network to focus more effectively on valuable areas and further enhance detection accuracy. The proposed model achieved a 4.4% higher mAP compared to YOLOv5s on the fabric dataset.

While the YOLOv5-Transformer achieves the highest mAP, further exploration of other variants, fine-tuning hyperparameters, and incorporating advanced feature aggregation techniques could further improve accuracy while maintaining efficiency, surpassing other comparative to other existing papers such as Teacher Network, Improved YOLOv5s, FD-YOLOv5, and YOLOv5. The YOLOv5-Transformer demonstrated a 4.4% higher mAP compared to YOLOv5s, a 2.2% improvement over YOLOv5l, and a 2.6% increase over YOLOv5x. These results reflect the robustness of our proposed algorithm.

Furthermore, we plan to investigate the potential of our model on more complex datasets, such as 3D objects and multiple objects in a single image, using two-stage detectors. We also plan to expand the fabric dataset by adding more categories for each type of fabric. Additionally, we aim to compare our model with more versions of

Table 2. Comparison results of the model's performance by using CPU and GPU.

Models	GFLOPs	Parameters (M)	CPU (ms)	GPU (ms)
YOLOv5s	15.8	7.2	28,960	1,988
YOLOv5n	4.1	1.9	13,081	1,749
YOLOv5m	47.9	21.2	68,405	3,121
YOLOv5l	107.7	46.5	130,105	4,506
YOLOv5x	203.8	86.7	212,994	4,629
YOLOv5-Transformer	43.3	20.4	66,121	3,003

Table 3. Comparison of YOLO models and our proposed model on the fabric dataset using GPU.

Models	GFLOPs	P (%)	R (%)	mAP0.5 (%)	mAP (%)	GPU (ms)	FPS
YOLOv5s	15.8	99.7	100	99.5	78.5	1,988	101.88
YOLOv5n	4.1	99.7	100	99.5	78.5	1,749	100.82
YOLOv5m	47.9	99.9	100	99.5	80.7	3,121	84.28
YOLOv5l	107.7	99.8	100	99.5	80.5	4,506	110.92
YOLOv5x	203.8	99.9	100	99.5	80.3	4,629	102.91
YOLOv5-Transformer	43.3	99.9	100	99.5	82.9	3,183	100.33

the YOLO family and work on improving frames per second (FPS) performance for real-time systems.

Acknowledgment

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (NRF-2022R1I1A3069371), was funded by BK21 FOUR (Fostering Outstanding Universities for Research) (No.: 5199990914048).

References

[1] D. Zheng, "Pattern-driven color pattern recognition for printed fabric motif design," *Color Research & Application,* pp. 207–221, 2021.

[2] Q. Liu, C. Wang, Y. Li, M. Gao, and J. Li, "A fabric defect detection method based on deep learning," *IEEE Access,* vol. 10, pp. 4284–4296, 2022.

[3] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: Challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications,* vol. 82, pp. 9243–9275, 2023.

[4] M. M. Khodier, S. M. Ahmed, and M. S. Sayed, "Complex pattern Jacquard fabrics defect detection using convolutional neural networks and multispectral imaging," *IEEE Access,* vol. 10, pp. 10653–10660, 2022.

[5] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object detection," in *Computer Vision,* Cham, Switzerland: Springer International Publishing, pp. 875–883, 2021.

[6] P. Adarsh, P. Rathi, and M. Kumar, "YOLO v3-Tiny: Object detection and recognition using one stage improved model," in *Proc. 6th Int. Conf. Advanced Computing and Communication Systems (ICACCS),* Coimbatore, India, pp. 687–694, 2020.

[7] W. Zhang, Q. Zhu, Y. Li, and H. Li, "MAM Faster R-CNN: Improved Faster R-CNN based on malformed attention module for object detection on X-ray security inspection," *Digital Signal Processing,* vol. 139, p. 104072, 2023.

[8] E. Kim, J. Lee, H. Jo, K. Na, E. Moon, G. Gweon, B. Yoo, and Y. Kyung, "SHOMY: Detection of small hazardous objects using the You Only Look Once algorithm," *KSII Transactions on Internet & Information Systems,* vol. 16, pp. 1–16, 2022.

[9] S. Ros, P. Tam, I. Song, S. Kang, and S. Kim, "A survey on state-of-the-art experimental simulations for privacy-preserving

federated learning in intelligent networking," *Electronic Research Archive,* vol. 32, pp. 1333–1364, 2024.

[10] S. Teng, J.-Y. Kim, S. Jeon, H.-W. Gil, J. Lyu, E. H. Chung, K. S. Kim, and Y. Nam, "Analyzing optimal wearable motion sensor placement for accurate classification of fall directions," *Sensors,* vol. 24, p. 6432, 2024.

[11] A. Kumar, Z. Zhi-Jie, and H. Lyu, "Object detection in real time based on improved single shot multi-box detector algorithm," *EURASIP Journal on Wireless Communications and Networking,* vol. 1, p. 204, 2020.

[12] N. O. Lynda, N. A. Nnanna, and M. M. Boukar, "Remote sensing image classification for land cover mapping in developing countries: A novel deep learning approach," *Int. J. Comput. Sci. & Netw. Security,* vol. 2, pp. 214–222, 2022.

[13] H. Wang and N. Xiao, "Underwater object detection method based on improved Faster R-CNN," *Applied Sciences,* vol. 13, p. 2746, 2023.

[14] D. Demetriou, P. Mavromatidis, P. M. Robert, H. Papadopoulos, M. F. Petrou, and D. Nicolaides, "Real-time construction demolition waste detection using state-of-the-art deep learning methods: Single-stage vs two-stage detectors," *Waste Management,* vol. 167, pp. 194–203, 2023.

[15] Z. Zhao and Y. Zhao, "YOLOv5s-Transformer: Improved YOLOv5 network for real-time detection of cigarette smoking based on image processing," in *Proc. 4th Int. Seminar on Artificial Intelligence, Networking and Information Technology (AINIT),* Nanjing, China, pp. 672–675, 2023, IEEE.

[16] Z. Zhang, Z. Lei, M. Omura, H. Hasegawa, and S. Gao, "Dendritic learning-incorporated vision transformer for image recognition," *IEEE/CAA Journal of Automatica Sinica,* vol. 11, pp. 539–541, 2024.

[17] K. S. Kumar and M. R. Bai, "LSTM-based texture classification and defect detection in a fabric," *Measurement: Sensors,* vol. 26, p. 100603, 2023.