INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 7,854 Volume 12, issue 12 (2025)

REAL-TIME MULTI-CAMERA TEXTILE DEFECT DETECTION USING YOLO-BASED DEEP LEARNING AND TENSORFLOW LITE ON CPU PLATFORMS

Termiz State University of Engineering and Agrotechnologies
Qarshiyev B. E.,

Kholliyev S. R.

Abstract: Automated quality inspection is a fundamental requirement in modern textile manufacturing. Manual inspection techniques are increasingly unable to satisfy industrial demands due to subjectivity, low inspection speed, and high labor costs. This paper presents a complete end-to-end system for real-time textile defect detection based on deep learning, optimized for deployment on CPU-only platforms. A YOLO-based object detection model is trained on textile defect datasets and subsequently converted into TensorFlow Lite (TFLite) format for efficient inference. The system supports images, video streams, live camera feeds, and simultaneous multi-camera processing. A lightweight web-based interface is implemented using Streamlit, providing visualization, logging, statistical analysis, and CSV export functionalities.

Extensive optimization techniques, including input resolution reduction, inference rate limiting, and asynchronous multi-camera execution, are employed to achieve real-time performance without GPU acceleration. Experimental evaluation confirms the system's efficiency, scalability, and practical applicability in industrial textile quality control.

Keywords: Textile defect detection, deep learning, YOLO, TensorFlow Lite, real-time inspection, multi-camera systems, computer vision, Streamlit.

Introduction

Textile manufacturing is one of the most quality-sensitive industrial domains. Even small defects such as broken yarns, stains, or weaving irregularities can significantly reduce the commercial value of fabrics. As production speeds increase, the reliance on traditional manual inspection becomes problematic due to operator fatigue, inconsistency, and limited scalability. Recent progress in computer vision and deep learning has enabled automated inspection systems capable of learning complex visual patterns directly from data. Convolutional Neural Networks (CNNs) have demonstrated superior performance in detecting defects across various materials. However, most existing solutions require powerful GPUs, which are expensive and impractical for many textile enterprises. This research focuses on designing a **GPU-independent, CPU-optimized real-time textile defect detection system**, suitable for deployment in real industrial settings.

Related Work

Early textile inspection systems relied on classical image processing approaches such as: Gray-level co-occurrence matrices (GLCM), Gabor filters, Fourier and wavelet transforms, Statistical texture descriptors. While computationally efficient, these methods are sensitive to noise, lighting conditions, and fabric variations. Deep learning-based approaches have significantly improved robustness and accuracy. Two-stage detectors (Faster R-CNN) offer high precision but insufficient speed, while single-stage detectors such as SSD and YOLO provide real-time performance. YOLO models, in particular, process the entire image in a single forward pass, making them ideal for time-critical applications. TensorFlow Lite enables deployment of trained deep learning models on constrained devices by applying graph optimization and reduced-precision arithmetic.

Let an input textile image be represented as:

$$I \in \mathbb{R}^{H \times W \times 3}$$

The task is to detect a set of defects:

$$D = \{(b_i, c_i, s_i)\}_{i=1}^{N}$$

where

- $b_i = (x_1, y_1, x_2, y_2)$ denotes bounding box coordinates,
- c_i is the defect class,
- $s_i \in [0, 1]$ is the confidence score.

The system must satisfy:

- real-time inference constraints;
- CPU-only execution;
- scalability to multiple cameras;
- robustness under varying fabric textures.

System Architecture The system includes data acquisition, YOLO model training, TensorFlow Lite conversion, and real-time inference. The proposed architecture consists of the following stages:

1. Data acquisition and annotation
2. YOLO-based model training
3. Model export (ONNX)
4. TensorFlow Lite conversion
5. CPU-optimized inference engine
6. Multi-camera processing module
7. Web-based visualization and logging interface

Model Training

A YOLO-based deep learning architecture is trained using labeled textile images with augmentation. A YOLO-based architecture is trained using labeled textile defect images. The

INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 7,854 Volume 12, issue 12 (2025)

loss function is defined as:

$$\mathcal{L} = \mathcal{L}_{box} + \mathcal{L}_{obj} + \mathcal{L}_{cls}$$

where bounding box regression, objectness confidence, and classification losses are jointly optimized. Data augmentation techniques such as scaling, flipping, and color jittering are applied to enhance generalization.

Model Optimization

The trained model is converted to TensorFlow Lite for efficient CPU inference. ONNX Export The trained YOLO model is exported into ONNX format to ensure framework independence. TensorFlow Lite Conversion The ONNX model is converted to TensorFlow and then to TFLite using FP32 and FP16 formats: **FP32** ensures higher numerical stability. **FP16** reduces memory and improves inference speed.

7. Real-Time Inference with TensorFlow Lite

Given an input frame I_t, TFLite inference computes:

$$D_t = f_{\text{TFLite}}(I_t)$$

Inference frequency is controlled by:

$$\Delta t \geq \frac{1}{FPS_{max}}$$

to avoid CPU overload.

Experiments

Experiments show stable performance at 8–15 FPS on CPU-only hardware.

8. Multi-Camera Processing

Each camera stream is processed in a separate execution thread:

Algorithm 1: Multi-Camera Inference

arduino Копировать код

```arduino
for each camera Ci in Cameras:
    start thread Ti
    while Ti is active:
        capture frame
        if inference interval elapsed:
            run TFLite inference
        update shared frame buffer
```

Frames from multiple cameras are merged into a grid layout for visualization.

Streamlit is used to implement a lightweight UI enabling: selection of input mode (image, video, camera); real-time visualization; confidence threshold adjustment; FPS regulation; detection logs; CSV export. Experiments were conducted on a CPU-only laptop: Intel-based processor, No GPU acceleration Results:

- Average FPS: **8–15**
- Multi-camera support: **up to 3 cameras**
- Stable inference under continuous operation

Conclusion

The proposed system demonstrates that reliable textile defect detection is feasible without GPU acceleration. The results confirm that real-time textile inspection is feasible without GPUs. Compared to manual inspection, the system offers improved consistency, traceability, and scalability. This study presented an optimized real-time textile defect detection system using YOLO and TensorFlow Lite. The combination of deep learning, CPU optimization, and a web-based interface provides a practical and cost-effective industrial solution. Future improvements include:

- Raspberry Pi and edge deployment
- Android application with TFLite
- Automatic defect severity grading
- Integration with ERP and MES systems
- Self-adaptive learning pipelines

References:

[1] Redmon, J., et al., *You Only Look Once: Unified, Real-Time Object Detection*, CVPR, 2016.
[2] Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y., *YOLOv4*, arXiv, 2020.
[3] Howard, A. et al., *MobileNets*, arXiv, 2017.
[4] TensorFlow Lite Documentation, Google, 2024.
[5] Xie, X., *A Review of Defect Detection Methods for Textile Fabrics*, Journal of Textile Research.
[6] Goodfellow, I., Bengio, Y., Courville, A., *Deep Learning*, MIT Press.