Engineering and Technology Journal e-ISSN: 2456-3358

Volume 11 Issue 04 April-2026, Page No.-9515-9519

DOI: 10.47191/etj/v11i04.22, I.F. – 8.595

Intelligent Real-Time Quality Inspection in Manufacturing Using Artificial Intelligence

Atul Deshmukh[1], Bhushan Vidhale[2], Pankaj Chandankhede[3], Pooja Yenurkar[4]

[1,2,3]Faculty, G H Raisoni College of Engineering, Nagpur, India

[4] Faculty, G H Raisoni college of Engineering and Management, Nagpur, India

ABSTRACT: To realize ambitious goals, such as zero-defect manufacturing in precision manufacturing in Industry 4.0, smart and autonomous quality control has to replace manual and error-prone quality control. We present Smart Inspect, a system that utilizes artificial intelligence (AI) to autonomously detect, classify, and segment defects with extremely high speed and accuracy. Unlike existing solutions, Smart Inspect fits smoothly into existing production lines, enabling manufacturers to scale inspection throughput with their production needs.

The Smart Inspect model is based on an advanced computer vision pipeline. The first step of the pipeline consists of the acquisition and preprocessing of each image. Several standard operations are available through OpenCV to achieve uniformity within our model's input. The array processing operations are performed using the array-processing library NumPy, and the defect detection is enabled and powered by a deep learning model created with the TensorFlow deep learning framework. We employ MobileNetV2, a compact convolutional neural network, and we utilize transfer learning to take advantage of the feature extraction capabilities of pre-trained networks on the task of manufacturing defect detection. Transfer learning improves the time required to train the model, and the number of images required. Its classification head is designed to prevent overfitting by applying a GlobalAveragePooling2D layer and a Dense layer with softmax activation function, allowing it to classify several kinds of surface defects and anomalies. To satisfy the real-time demands of the production environment, the architecture of the system is threaded. The threading separates the image capture and the model inference, so they can be run concurrently, and also prevents the inspection system from becoming a manufacturing bottleneck. JSON is also used to produce structured logs that show the identified defect and metadata associated with it (defect type, defect location, high-precision timestamp, etc.). This creates a clear and auditable history of quality control, which can be analyzed to improve the production process. Smart Inspect makes measurement and inspection automatic, allowing manufacturers to eliminate human inspection errors and increase quality assurance rates. It reduces subjectivity and inspection fatigue for speed and efficiency. It provides a highly accurate, low-cost, scalable, and flexible solution to manufacturers on their adventure to a zero-defect strategy and those wishing to differentiate their products through quality assurance, against manufacturers in an increasingly data-driven era.

I. INTRODUCTION

In today's precision manufacturing industry, product quality is constant if companies keep it, a state that is both vital and difficult. Methods for manual inspection that are customary are often slow and also inconsistent; they are prone to human error, so they are unsuitable for production in high volume. To address this matter, we developed SmartInspect: AI-Driven Quality Control for Precision Manufacturing, a system that detects any defects automatically through leveraging artificial intelligence with computer vision.

In this paper integrates OpenCV with NumPy for image pre-processing and analysis. MobileNetV2 with TensorFlow works as the base of the deep learning model so it can classify defects. The architecture incorporates GlobalAveragePooling2D and Dense layers for extracting features efficiently also to predict accurately. For handling system operations, logging, parallel execution as well as real-time monitoring, Python modules such as OS, time, date time, threading, and JSON get employed. SmartInspect combines these technologies in order to provide a scalable and cost-effective solution for when it improves inspection speed. Human intervention is minimized, and reliability within precision manufacturing environments is improved.

II. RELATED WORK

The transformation of manufacturing systems under Industry 4.0 has significantly increased the demand for intelligent, automated quality inspection techniques. Traditional manual inspection methods are not only time-consuming but also inconsistent and prone to human error, making them unsuitable for high-speed production environments. Consequently, researchers have explored Artificial Intelligence (AI) and computer vision-based approaches to enable real-time, accurate, and scalable quality monitoring.

Early work by Yang et al. [1] provided a comprehensive survey of deep learning techniques for defect detection, emphasizing the superiority of Convolutional Neural Networks (CNNs) over traditional machine learning approaches. Their study demonstrated that CNNs can automatically extract hierarchical features such as edges, textures, and patterns, thereby eliminating the need for handcrafted feature engineering. Subsequent research has focused on improving defect detection accuracy and robustness. A systematic review in Engineering Applications of Artificial Intelligence [2] highlighted that deep learning models, including ResNet and U-Net architectures, achieve high accuracy in detecting surface defects across multiple industries. However, these models often require large labeled datasets and high computational resources, which can limit their deployment in real-time industrial environments. Further advancements were reported in CNN-based visual inspection systems [3], where supervised learning techniques were widely used for classification and segmentation tasks. The study also explored unsupervised and semi-supervised learning methods for anomaly detection, particularly in scenarios with limited labeled data. These approaches improve flexibility but may suffer from lower precision compared to fully supervised models.

Recent studies [4] have introduced advanced AI techniques such as Generative Adversarial Networks (GANs), vision transformers, and few-shot learning models. These methods address challenges like data scarcity and model generalization. However, issues such as model interpretability, reliability, and real-time deployment constraints remain critical challenges for industrial adoption. In addition, AI-driven quality control systems integrating computer vision and deep learning have shown significant improvements in inspection speed and accuracy [5]. Lightweight architectures like MobileNetV2, combined with transfer learning, have gained popularity due to their ability to perform real-time inference with limited computational resources. These models are particularly suitable for embedded and edge-based manufacturing systems. Despite these advancements, there is still a need for integrated systems that combine real-time processing, scalability, and robust defect detection. The proposed SmartInspect system addresses these challenges by leveraging lightweight deep learning models, multithreaded execution, and structured data logging for efficient and scalable quality monitoring.

Ref	Method/ Approach	Key Technique	Advantages	Limitations
[1]	Deep Learning Survey	CNN-based models	High accuracy, automatic feature extraction	Requires large datasets
[2]	Surface Defect Detection	ResNet, U-Net	Robust detection, high precision	High computational cost
[3]	CNN-based Inspection	Supervised & Unsupervised Learning	Flexible for different data scenarios	Lower accuracy in unsupervised cases
[4]	Advanced AI Models	GANs, Transformers	Handles data scarcity, improved generalization	Complex, not easily deployable in real-time

III. METHODOLOGY

The proposed system follows along with a multi-stage pipeline that also integrates computer vision, deep learning, and real-time execution mechanisms. We designed that methodology to ensure robustness as well as reproducibility. Also, the methodology complies under modern industrial automation practices.

OpenCV (cv2) captures in real time high-resolution product images from a production line camera or test dataset. The acquisition module performs: Frame capture at predefined intervals. Distinguishing between each frame with the help of the data time library and its time stamps. Using OS for controlling the files organization in directories. Serializing metadata into json files for future reference. This guarantees that each image coming into the system has an auditable identity for post-model verification and error analysis. Data Pre-processing All images are modulated by structured pre-processing in order to maintain consistency throughout the pipeline.

Pre-processing stages include: Resizing and Cropping Images are resized to 224×224 pixels (input of the MobileNetV2 model). Cropping the irrelevant backgrounds of the product surface is removed.

Pixel intensities are rescaled inside the [0, 1] range with NumPy arrays because that ensures consistent input distributions for neural network stability.

Rotation, flipping, brightness variation. It wanted to simulate conditions of a real factory and also lighting and orientation that varied. These steps reduce noise. The model's generalization capability is able to be improved. ImageNet pre-trains MobileNetV2 which is a fast convolutional neural network system backbone. It extracts hierarchical attributes fulfilling this function. Early layers are with low-level edges also textures. Mid-level (patterns, shapes). High-level defect regions exist within. Anomalies also occur. Instead of flattening, a GlobalAveragePooling2D layer compresses feature maps into a dense vector. This operation ensures compact feature representation, and it reduces model parameters for prevention of overfitting.

Features extracted have dense fully connected layers appended. They convert feature embeddings to separate groups. These discrete classes do include Pass, Defect-Type A, and also Defect-Type B. For hidden layers use ReLU for output use Softmax or Sigmoid Activation Functions. For cases involving multi-class classification a loss function involves categorical cross-entropy. Adam: Fast convergence optimizer. Training Strategy: Fine-tune last few MobileNetV2 layers so weights pre-trained adapt for manufacturing dataset. Abstract features get transformed by this module for actionable inspection decisions. A decision logic layer will receive the final inference results. Classify as Pass/Reject if confidence is over the threshold. Flag the product for the purpose of manual inspection in the event confidence is ambiguous. Results are logged by JSON reports. The reports also contain:

- Timestamp
- Predicted class
- Confidence score
- Processing latency

In addition to providing an inspection trail for upcoming audits, this ensures transparency. Execution in Real Time and Parallelization. Near real-time operation is required in manufacturing environments. The system uses threading for concurrent tasks to solve this:

Thread 1: Takes pictures with the camera.

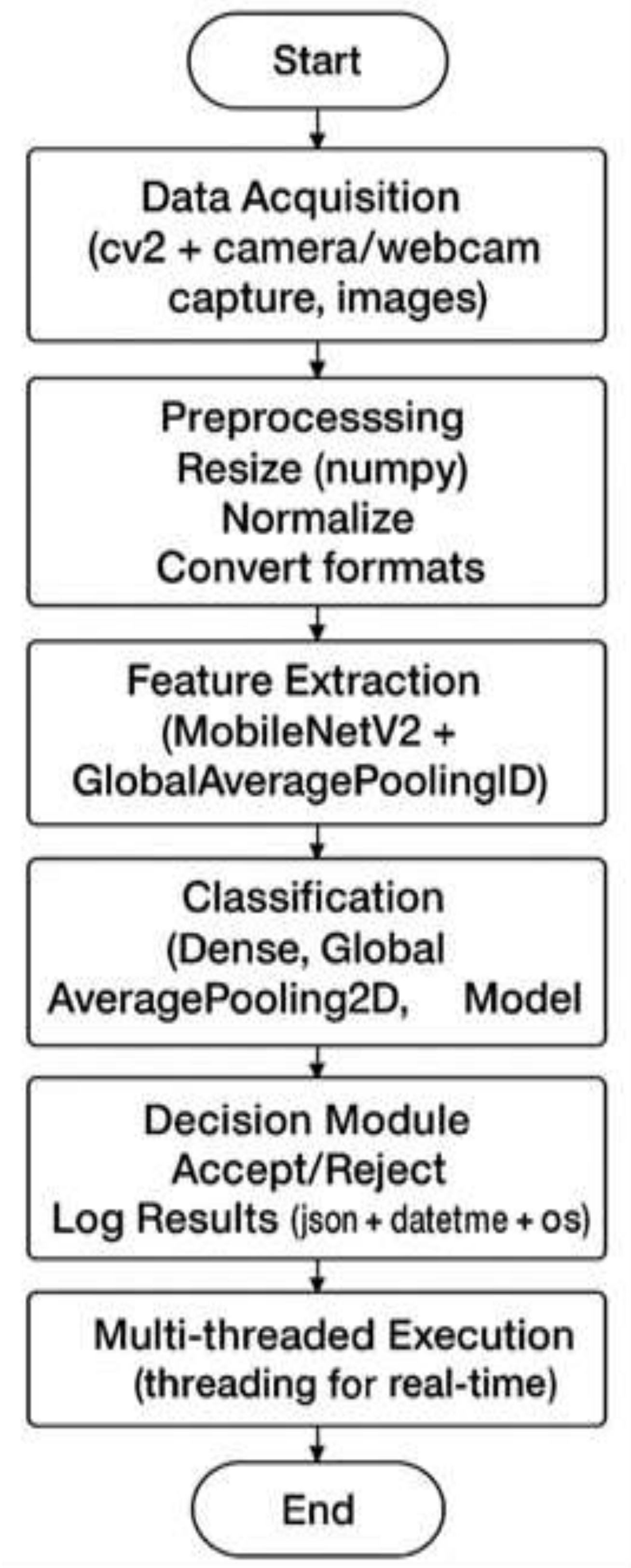
Thread 2: Image pre-processing.

Thread 3: Uses the TensorFlow model to perform inference.

Thread 4: Updates the monitoring dashboard and logs outcomes.

By minimizing bottlenecks, this pipeline allows for continuous inspection without stopping production lines. A specialized monitoring interface offers: Real-time display of product status using cv2.imshow(). Automated archiving of results in JSON/CSV formats.

Weekly or batch reporting utilizing operating system utilities for organized directory management. The dual-mode (real-time and offline analytics) renders the system suitable for prolonged factory implementation.



The first step involves the initialization of the system, where all the hardware like cameras and sensors and software components are set up. A camera/webcam and OpenCV (cv2) capture images of the manufacturing line. OpenCV (cv2) is a library that completes computer vision tasks. Real-time images or video frames of the product are collected. These images are the input for further processing. It provides for continuous inspection of products as they move through the production process. The task of pre-processing is to make the data more accurate and consistent before it is fed to the AI model. This includes:

Resizing with NumPy: Image dimensions resize to fit the model input.

Normalization: Scaling the pixel values to a consistent range, such as from 0 up to 1. The image formats are converted. Examples are RGB to grayscale or tensor formats needed by the neural network. It is also useful for transforming the images into the appropriate form for feature extraction. Here, the AI model relies on MobileNetV2 and Global Average Pooling to recognize trends or features in each image. This system uses visual features such as shapes, edges, or textures. MobileNetV2 is a small deep learning model that can be run in real time. This involves converting complex images into numerical values. Afterwards, the processed features are fed into a classification network for categorization. The network consists of Dense and GlobalAveragePooling2D layers. The AI model then assesses the product and assigns it a category or label such as "defective". This is the part of the AI system that makes decisions.

It makes decisions based on the classifier's predictions: If the product meets quality standards, accept it Reject the defective product.

JSON format stores all results also. Datetime and operating system (OS) trace and report within fields. Multi-threading allows many of the functions in the system, such as capturing, processing, and logging, to happen simultaneously to achieve real-time performance. This makes it suitable for continuous inspections in high-speed manufacturing and high-volume production lines.

This process continues until the inspection is complete, or is cycled into an online quality control phase, with the option of using the inspection results to analyze quality, retrain models, or create quality reports.

IV. RESULTS

V. CONCLUSIONS

In this project, SmartInspect: AI-Driven Quality Control for Precision Manufacturing, we successfully developed an automated inspection system for the reason that we leveraged computer vision along with deep learning techniques. Integrating OpenCV for image preprocessing, NumPy for efficient data manipulation, and TensorFlow with a MobileNetV2 backbone improved with Dense and GlobalAveragePooling2D layers allowed the system to show strong feature extraction plus classification capabilities. Multithreading, datetime, and JSON-based logging enabled real-time monitoring, process synchronization, and efficient data management.

For industrial quality control tasks, the proposed framework validates greatly reduced human error, improved accuracy, and improved efficiency. The project contributes toward the advancement of smart manufacturing. It presents a scalable AI solution for defect detection plus quality assurance that is lightweight and cost-effective.

REFERENCES

1. J. Yang et al., "Using deep learning to detect defects in manufacturing: A comprehensive survey," Materials, 2020.
2. "Deep learning approaches for surface defect detection," Engineering Applications of AI, 2024.
3. "CNN-based visual defect detection," Computers in Industry, 2023.
4. "AI-enabled defect detection: Challenges and future trends," Advanced Engineering Informatics", 2026.
5. N. R. Mitta, "AI for automated quality control," Journal of Deep Learning Applications, 2024.
6. S. Sundaram and A. Zeid, "Artificial Intelligence-Based Smart Quality Inspection for Manufacturing," Micromachines, vol. 14, no. 3, Art. no. 570, 2023.
7. S. Wang, "An Intelligent Quality Control Method for Manufacturing Processes Based on a Human–Cyber–Physical Knowledge Graph," Science Direct, 2024.