International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

AI-Driven Real-Time Gear Classification for Automotive Manufacturing

Michael Mesfin Tadesse, Luo Jian You, Qiao Hua
Zhou Zhi Qiang , Rizwan-ul-Haque Syed
Zhejiang Shuanghuan Driveline Co., Ltd Zhejiang, China

Yun Xiang
Institute of Cyberspace Security, Zhejiang University of Technology ZJUT
BinJiang Institute of Artificial Intelligence, Zhejiang University of Technology ZJUT Zhejiang, China

Abstract

Recent advancements in deep learning have significantly enhance visual inspection systems across different industrial applications. This study introduces a real-time gear classification framework built on MobileNetV2 and a task-specific training pipeline tailored for high-throughput production environments. The core of this research is a custom SH dataset (Zhejiang Shuanghuan Driveline Co., Ltd), featuring up to 50,000 high-resolution RGB images. This dataset spans 50 gear categories, all recorded under a wide range of diverse factory-floor conditions. The proposed pipeline incorporates robust data augmentation to enhance model generalization and address class imbalance. Transfer learning from ImageNet-pretrained weights enables efficient domain adaptation, while a customized classification head and progressive layer unfreezing support fine-grained recognition. Experimental results show that MobileNetV2 achieved 92% classification accuracy with sub-20 ms inference latency, outperforming deeper architectures such as VGG16 and ResNet-50 in real-time scenarios. The system was deployed on industrial-grade hardware, achieving real-time inference with a throughput exceeding 7,200 gears per hour. It maintained robust performance under varying lighting conditions and gear orientations, demonstrating strong generalization in real-world factory environments. This work underscores the effectiveness of lightweight convolutional neural network (CNN) architectures, transfer learning, and data-centric training strategies in building scalable, high-performance inspection systems aligned with Industry objectives.

Keywords: transfer learning; deep learning; convolutional neural networks; data augmentation; real-time inspection; gear categorization ; artificial intelligence ; smart manufacturing.

1 Introduction

As the impact of artificial intelligence (AI) continues to reshape and expand its influence across critical industries, breakthroughs advances in computer vision (CV) and deep learning (DL) have transformed domains such as autonomous driving [1], medical diagnostics [2], microchip defect detection [3], and smart manufacturing [4]—enabling machines to interpret complex visual data with unprecedented accuracy and efficiency. At the core of these breakthroughs, convolutional neural networks (CNNs) have emerged as highly effective tools, leveraging their multi-layered architectures to learn hierarchical features directly from raw image data. This capability has enabled CNNs to consistently surpass traditional handcrafted methods in both accuracy and generalization [5, 6].

Image classification plays a key role in computer vision uses. It backs up jobs like recognizing faces, monitoring videos for security, and controlling quality in industry settings. In manufacturing areas, CNNs show real potential. They boost how dependable inspections get and cut back on errors from people. This happens thanks to their full end-to-end learning setup [7]. That said, setting up real-time systems for classifying images in factory spots stays pretty tough. Things like shifts in object shapes, light levels, surface feels, and extra noise in the back all create issues. This hits hard in focused areas such as checking gears. There, tiny visual changes across gear kinds make it hard to classify them right.

Back in the day, inspections in manufacturing leaned on standard machine learning ways and hands-on sorting steps [8–11]. Those approaches needed features that workers built by hand, covering stuff like forms, measurements, and surface patterns. They held up fine in steady, controlled spots. Still, they fell short on flexibility and strength when factory conditions kept changing around. The downsides of those older methods really pushed the move to more advanced deep learning tools.

Even with what CNNs can do, using them for industry tasks like sorting gears has faced limits. There just is not enough labeled data for training, and shifting to new domains proves tricky [12]. Open datasets tend to stay out of reach or do not fit well. Reasons include keeping things secret, owning the data outright, and how unique factory images turn out [13–15]. To push past those hurdles, we put together the SH Gear Dataset. It holds detail images taken right from active production lines. This collection mirrors actual on-the-ground situations. It works great for training aimed at specific jobs. Pair it with transfer learning from CNNs pretrained on ImageNet [16], plus heavy data boosting [17], and targeted adjustments [18]. That mix lets models learn well from small but spot-on data sets.

Transfer learning forms the base of how we handle this.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

It means locking the bottom layers in place to keep basic visual traits like edges and patterns intact. Then, we tweak the top layers to pick up on patterns tied to the job at hand. This way cuts down on overlearning issues and speeds up how fast training settles in. It shines when you have tight computing power or few marked examples [19]. From there, we built a setup for classifying gears in real time. It fits right into production factory spaces. The whole thing blends transfer learning, slim CNN bases, and a flexible app tuned for edge devices.

In building this setup, we picked MobileNetV2 as the main structure [20]. Its small size and smart separable convolutions by depth make it perfect for hardware in factories. To check how it does, we ran tests against VGG16 [21]. That one pulls out features in deep layers step by step. We also compared it to ResNet-50 [22]. This model uses leftover links to steady training in thick networks. Earlier work points out the balances these models strike [23]. It comes down to how accurate they are versus how quick they run in live, edge-style uses.

This framework was the first AI-driven visual inspection system put into use at the SH Manufacturing Facility. At that time, deep learning was not widely used in similar gear manufacturing settings. At that time, the combination of lightweight CNN architectures, transfer learning, and hardware that could be deployed on the edge made it possible to close the gap between research-grade models and live production needs. This work is an early operational reference point for AI integration in precision gear inspection, and its results show the state of the art at a key time in the adoption of AI in industry. The field has come a long way since then.

To bolster robustness against diverse gear appearances and make our system work well with types of gears and lighting we use many ways to change the pictures we train it with. We also use a way to train the system, where we freeze some parts and then slowly make it better so it can learn the special things about the gears we are looking at. We have a set of pictures called the SH Gear Dataset with over 50 types of gears taken in different factory conditions. This helps us train and test the system. We use this system on the Early Production Containment lines at the SH Manufacturing Facility. It can correctly identify gears 92 percent of the time. Can look at 7,200 gears, in one hour. This is ten times faster than people can do it who can only look at 720 gears in one hour. The system can also work in time because it only takes 20 milliseconds to make a decision and it can work on special computers with Intel Core i7 processors that are used in factories.

Further enhancements in robustness and deployability stem from hyperparameter tuning, model compression, and a real-time interface designed using TensorFlow, OpenCV, and PySimpleGUI. Containerized deployment ensures scalability, maintainability, and seamless integration across production sites, aligning with Industry 4.0 objectives of automation, traceability, and smart analytics.

Overview and Contributions This work integrates recent advances in deep learning with practical deployment strategies to deliver a scalable, efficient, and interpretable solution for automated gear classification in industrial environments. The system is designed for real-time operation, optimized for edge deployment, and built with flexibility to accommodate diverse gear types and imaging conditions. The following contributions reflect both technical innovation and deployment readiness:

- **Development of the SH Gear Dataset:** A specialized dataset was constructed, comprising over 40,000–50,000 high-resolution RGB images of gears captured at multiple resolutions (244×244 to 720×720 pixels) to ensure fine-detail visibility and capture subtle visual differences. Acquired under authentic factory-floor conditions, the dataset provides a robust foundation for training and validating deep learning models in real-world industrial environments.

- **Transfer Learning with ImageNet-Pretrained CNNs:** MobileNetV2, pretrained on ImageNet, was adapted to leverage generalized low-level features such as edges and textures. This transfer learning strategy enabled robust performance across varying gear types and imaging conditions, despite the moderate dataset size.

- **Quantitative Benchmarking Across Architectures:** A comparative evaluation of MobileNetV2, ResNet-50, and VGG16 demonstrated that MobileNetV2 provides the best trade-off between accuracy and speed—achieving 92% accuracy with a 20 ms inference time—making it well-suited for real-time industrial deployment. Deeper models offered only marginal accuracy improvements at the expense of significantly higher computational costs.

- **Optimized and Scalable Real-Time Edge Deployment:** The system employs a quantized and pruned MobileNetV2 model, delivering sub-20 ms latency on an Intel Core i7-10710U-based Industrial Touch Panel PC, with a throughput of 7,200 classifications per hour—a tenfold improvement over manual inspection. The modular hardware setup, using off-the-shelf components (e.g., Logitech Brio camera, LED ring light, embedded PC), enables rapid adaptation to new gear types with minimal retraining. Containerized deployment ensures scalability and consistency across sites.

- **Industrial-Grade Integration and Deployable Application:** A full-stack application was developed for factory-floor deployment on Industrial Touch Panel PCs, integrating image acquisition, real-time inference, operator feedback, and automated logging. This end-to-end solution ensures high usability, operational reliability, and compliance with ISO 9001 standards, facilitating seamless integration into existing manufacturing workflows.

This paper is organized as follows: Section 2 reviews related work on computer vision and deep learning for manufacturing object detection and classification. Section 3 presents the methodology, including the SH Gear Dataset, CNN architectures (e.g., MobileNetV2), and evaluation metrics. Section 4 describes the deployment of the real-time gear classification system on the production line, focusing on integration and application-level implementation. Section 5 provides a comprehensive evaluation of the deployed

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

AI-based system across four Early Production Containment (EPC) lines, analyzing performance, scalability, and limitations. Section 6 concludes the study by summarizing the key findings and discussing their broader implications for intelligent manufacturing within the Industry 4.0 framework.

2 Background

Ensuring the Quality of production parts remains one of the primary cornerstones of industrial manufacturing. In the past, this burden was left to human analysis or conventional rule-based algorithms that stemmed from manually extracted features including the Gray Level Co-Occurrence Matrix (GLCM) and Local Binary Patterns (LBP) [24, 25]. Although some accuracy was achieved by conventional techniques, addressing actual manufacturing variabilities of varying illumination levels, complex geometry, and minute details on the surfaces remained a challenge for conventional techniques that are prevalent in a typical manufacturing environment.

As Industry4.0 emerged, smart manufacturing has embraced data-centric paradigms to foster productivity, flexibility, and product quality. Initial application cases of machine learning (ML) showed potential in automating inspection through the identification of statistical patterns in sensor and image data [26–28]. Yet, conventional ML approaches would depend on hand-crafted or pre-defined feature sets, which had difficulty in representing the complexity of real-world manufacturing environments—e.g., changing lighting, surface texture, and complicated defect geometries [29, 30]. This drawback has motivated the transition to deep learning (DL), a subfield of ML that enables hierarchical feature representation directly from raw data [31, 32]. In particular, convolutional neural networks (CNNs) have emerged as the prevailing method for visual inspection tasks because of their capacity for automatically acquiring spatial features at various levels of abstraction [6, 33]. CNNs circumvent hand-engineered features and have demonstrated state-of-the-art performance in a wide range of applications such as image classification, defect analysis, and image segmentation [34–36]. Collectively, these advances have made DL the foundation of contemporary industrial inspection systems, allowing for greater accuracy, robustness, and scalability than conventional ML approaches.

Owing to the limitations of the available data, augmentation techniques of data using transformations and generating images synthetically using GANs have been explored [37]. In addition to that, transfer learning has been recognized as a widely applicable fix. This method uses pre-trained models for images with huge datasets like ImageNet [16] to be fine-tuned to the target dataset with few examples for domain-specific problems [38, 39], which requires less computation and less training time.

Among early CNN architectures, VGG16 has been a popular choice for gear inspection and defect detection due to its simplicity and uniform 16-layer design [40]. Its structured convolutional blocks enable consistent extraction of spatial and textural features essential for differentiating between gear types and identifying surface anomalies. Several studies have successfully applied VGG16-based transfer learning for industrial inspection, achieving strong classification accuracy on limited datasets [41, 42]. However, the architecture's high computational cost and large parameter count make it less practical for real-time production environments, especially on embedded or resource-limited hardware [43].

ResNet-50 introduced the concept of residual learning, allowing much deeper networks to train effectively by mitigating vanishing gradient issues [22]. Its skip connections facilitate feature reuse and hierarchical abstraction, making it highly effective for complex industrial visual tasks such as gear surface evaluation and component classification [44, 45]. The model demonstrates excellent generalization across varying lighting conditions and gear orientations. Nonetheless, its higher computational demand and longer inference time make it less suitable for edge-based or latency-sensitive deployments without specialized GPU acceleration [39, 46].

MobileNetV2 [20], EfficientNet [43], and Inception [47] CNN architectures with reduced sizes of weights have appeared promising for the classification of gear in real-time machinery due to their efficient computation [48]. Training of CNN architectures for gears was performed with relevant transformations of the images [17, 35]. However, similar to other applications of image recognition, issues of inter-class similarity and environmental noise like motion blurs and reflection [49] pose significant challenges for

Recently, there have been improvements to the concept of transfer learning by allowing the freezing of convolutional layers to focus on general representation learning while fine-tuning the classification head for a specific task [50, 51]. Methods including dropout and checkpoint optimization can also be used to improve generalization performance for imbalanced classes or few-shot learning scenarios [52, 53].

In this situation, hybrid designs that integrate CNNs with autoencoders [53], or Object Detection methodologies like YOLO [54], have been able to expand capabilities from classification to include defect location in a realtime fashion. This can also allow for Multitask Learning. In addition to that, Explanatory AI techniques like Grad-CAM [55] have been investigated for improving interpretability and standard compliance, including ISO 9001 [56]

In addition to standard gear inspection pipeline operations of segmentation and surface anomaly detection, this paper specifically targets gear classification. This becomes a complex issue due to the fine-grained characteristics of gears that involve curvature and wear of the gear's surface. Prior art involved grayscale thresholding techniques and edge detection algorithms. However, current state-of-the-art techniques utilize CNN architectures of U-Net/Mask R-CNN for better feature extraction despite varying manufacturing settings [35, 57, 58].

In response, multimodal inspection systems combining RGB, thermal, and vibration sensors have been developed to increase fault detection robustness [59]. Cloud-based and edge-deployable solutions are gaining traction, enabling centralized model retraining and real-time data analysis—hallmarks of Industry 4.0 [60]. Despite the improvement of models themselves, the quality of the training

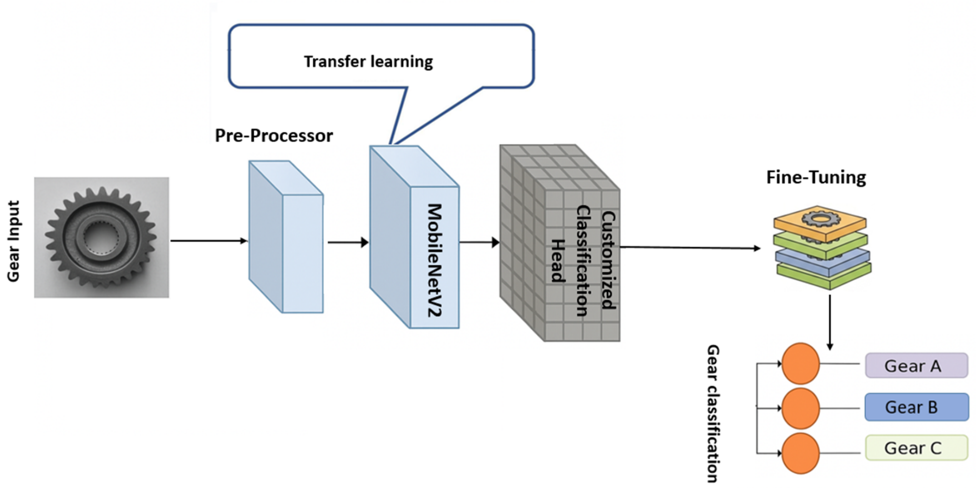


Figure 1: Gear classification pipeline overview.

dataset remains a key issue. Datasets of gear inspections may involve class imbalance problems, varying illumination conditions, and the lack of adequate defective samples due to costly annotation processes [17]. Solutions to the problems include simulating defective samples, domain adaptation learning techniques, and semi-supervised learning learning methodologies. Notably, a lack of standard public datasets remains a key issue to reproducibility of results across the current state of research works [35, 51].

In general, deep learning is changing from being separate visual modules to being parts of smart, connected manufacturing systems. Visual inspection is becoming more and more combined with signals like acoustic, thermal, and vibration data to make predictive maintenance and overall production monitoring easier [61, 62]. These trends show that there is a need for inspection solutions that can be scaled, work in real time, and be understood. This study fills that gap by creating a MobileNetV2-based gear classification system that is optimized for use in industry.

3 Methodology

This section presents the methodology for developing an AI-based gear classification m odel t ailored f or high-throughput manufacturing applications. The approach focuses on dataset construction, model design, training strategies, and optimization techniques to ensure robust and efficient classification performance under realistic visual conditions.

CNN form the core of the classification p ipeline, with transfer learning employed to leverage pretrained knowledge and reduce the need for extensive labeled data. Advanced preprocessing and data augmentation techniques are applied to increase the model's resilience to variations in gear appearance, lighting, and orientation.

As illustrated in Figure 1, input gear images are first preprocessed (including resizing, normalization, and augmentation), then passed through a MobileNetV2 backbone initialized with ImageNet weights. A customized classification head—comprising global average pooling, dense, and dropout layers—is appended and fine-tuned to adapt the model to gear-specific features. The final output layer produces multi-class predictions (e.g., Gear A, Gear B, Gear C), enabling accurate and scalable gear type classification in production environments.

3.1 Gear Dataset Composition, Imaging, and Augmentation

A custom, domain-specific gear image dataset was curated to facilitate the training and evaluation of deep learning models for real-time classification. It comprises high-resolution RGB images spanning 50 gear classes—each represented by approximately 1,000 to 1,200 images—including spur, bevel, helical, and worm gears commonly used in automotive systems, industrial machinery, and precision mechanical equipment.

3.1.1 Gear Imaging Strategy and Visual Variability Management

Image acquisition was conducted directly within the manufacturing environment under a variety of real-world operating conditions. A Logitech Brio 8.5 MP camera [63] was mounted in a fixed top-down position above a vibration-isolated inspection table and paired with an LED ring light with it to cut down on shadows, reflections, and uneven lighting. We manually turned and moved the gears under the camera. This way, we could get shots of all their different sides and angles, which was important for making sure our system could sort them correctly.

To deal with issues like glare and patchy light, our camera setup had controlled lighting and steady mounts. We also made sure to include some rare or unusual gears in our samples, just to make our data more varied. Every picture came with lots of details, like the gear's type, what it's made of, the light conditions, camera settings, and when it was taken. This helped us keep track of everything and run tests more easily when we were training and checking our model.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Figure 2 showcases representative gear images collected during the actual sampling process under real manufacturing conditions, highlighting the system's ability to perform fine-grained visual classification. In the top row (Category Class 1), you see gears TYPE-A through TYPE-D. They show how different gears, even those meant for different things or shaped differently, can look surprisingly alike. For instance, TYPE-A and TYPE-B have almost the same center and overall size. And TYPE-C and TYPE-D have very similar tooth patterns, making them hard to tell apart just by looking. To sort these correctly, our model needs to spot tiny details like the shape of the bevels, how symmetrical the edges are, and slight differences in the tooth spacing. The bottom row (Category Class 2) presents a similar situation within its own group. Here, TYPE-1 and TYPE-2 gears come from the same family but look almost identical from the outside, even though they do different jobs. These kinds of similarities within a group can easily cause mistakes in sorting or putting things together, especially in fast-moving production lines like those for Early Production Containment (EPC). This visual challenge really shows why we need advanced deep learning models. They can pick up on small structural differences that old inspection methods just can't see.

To visually categorize and differentiate similar gear types, the system employed an adaptive resolution approach. Through successive trials with a noticeable number of experiments at the standard resolution of 224×224 for each of the different gear types, it became apparent that loss of subtle geometric and surface detail - such as fine tooth profiles or edge contours or surface texture patterns - is common at lower resolutions, thereby impacting accuracy in classification. Because more precise, fine-scale features are necessary to reliably categorize gear types with similar shape or even finish, lower-resolution inputs risk underrepresenting these fine-scale features. For all gear types with demonstrated levels of visual similarity the resolution was increased, in this instance providing inputs at up to 720×720. The result of increasing the input resolution to higher levels is that the convolutional layers block were able to capture important fine edges, textures, and contours that would lead to a more accurate categorization of each of the different regular category or classes.

The choice of resolution was guided by a preprocessing analysis based on inter-class similarity metrics: feature embeddings were first extracted using a lightweight CNN on standard-resolution images, and pairwise distances between categories were calculated. Categories with low inter-class distances—indicating strong visual resemblance—were automatically processed at higher resolutions, while distinctly different types remained at 224×224 pixels to conserve computational resources. By combining empirical trials, similarity-based assessment, and selective scaling, the system ensured that each gear was analyzed at the most appropriate resolution, improving classification accuracy for challenging categories while maintaining efficiency suitable for real-time industrial deployment.

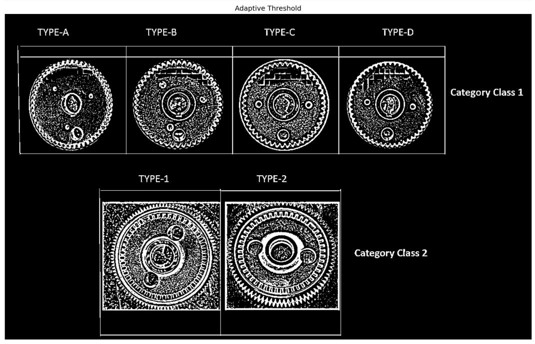


Figure 2: Sample input image used for gear classification.

3.1.2 Data Augmentation Pipeline

A custom data augmentation pipeline was developed using Python and OpenCV to synthetically expand the SH gear dataset, simulating real-world variations without requiring additional manual data collection [17, 64]. Dynamic transformations—each applied with a probability of 0.3—included rotation, flipping, scaling, contrast adjustment, noise injection, and zooming. These augmentations introduce realistic variability in gear appearance, thereby enhancing the model's generalization capability and robustness. Figure 3 presents representative examples of these applied transformations, demonstrating their contribution to improved classification performance under diverse imaging conditions.

- **Horizontal and vertical flipping** (30%) for orientation variations [21].
- **Rotation** ($\pm 30°$ in $5°$ increments) for angular diversity [65].
- **Random cropping** (80–90% of the frame size) for partial views [66].
- **Brightness and contrast** adjustments ($\pm 20\%$, $\pm 15\%$) for lighting fluctuations [67].
- **Scaling and Zoom:** Random scaling in the range of 0.8–1.2× simulated zoom effects and varying gear-to-camera distances.
- **Noise Addition:** Low-intensity Gaussian noise and synthetic defect injection for sensor noise and defect robustness [68].

3.1.3 Preprocessing and Validation

Prior to training, all images were normalized to the [0,1] range to align with standard CNN input requirements. Adaptive histogram equalization was applied cautiously to each RGB channel to enhance contrast under uneven lighting while preserving pretrained feature distributions during transfer learning. Instead of fixed resizing, models were trained using both native and upscaled inputs, with interpolation maintaining aspect ratios where necessary. The dataset was split into training, validation, and test subsets in an 80:10:10 ratio using stratified sampling to ensure balanced class representation despite varying sample

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

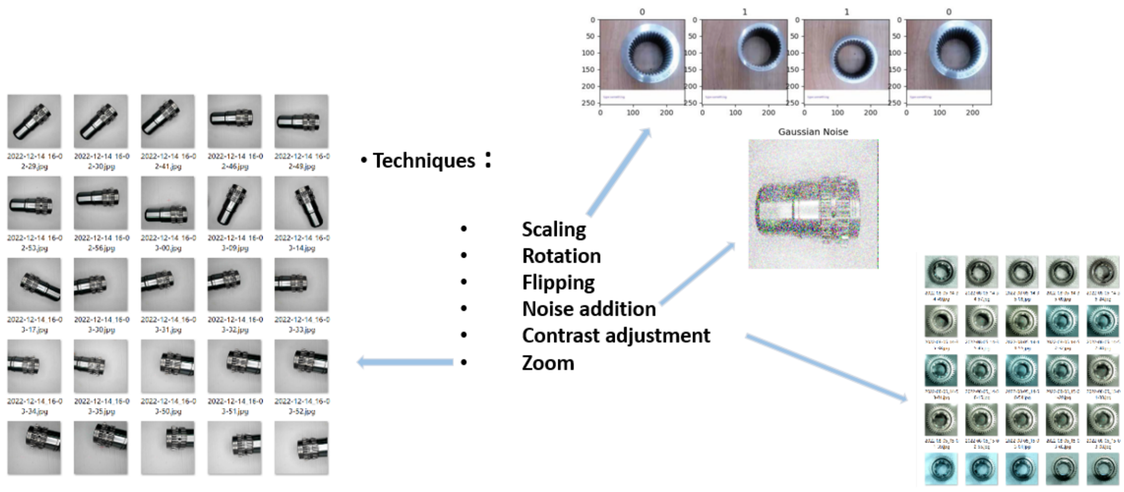


Figure 3: Data Augmentation Applied to Gear Images.

counts.Approximately 2% of images identified as damaged, unreadable, low-quality, or near-duplicate/repetitive were removed following automated and manual quality checks, and a subset underwent manual annotation verification to correct inconsistencies. These preprocessing and validation steps ensured a clean, balanced dataset that supported robust and reproducible model training.

3.2 Model Design and Training

The design and training of the gear classification model are centered on a transfer learning framework that adapts pretrained CNN architectures—most notably MobileNetV2—to the gear inspection domain. The pipeline consists of three key components: transfer learning strategy, model architecture, and training optimization.

3.2.1 Transfer Learning Strategy

Transfer learning was employed to adapt pretrained CNN architectures—particularly MobileNetV2—to the gear classification domain. Leveraging ImageNet-pretrained weights [34], the models retained foundational features such as edges and textures, reducing training time and improving generalization in limited-data scenarios [38, 39].

Initially, all convolutional layers were frozen to preserve general visual knowledge. A custom classification head tailored for gear recognition was appended. This head included average pooling, flattening, a fully connected ReLU-activated dense layer, dropout (rate = 0.5), and a softmax output layer with 40 output classes. After training the classification head alone for 20 epochs using the Adam optimizer (learning rate = 0.001), the deeper layers of MobileNetV2 were progressively unfrozen and fine-tuned with a reduced learning rate (0.0001). This hybrid transfer learning approach allowed the model to adapt efficiently to domain-specific features without overfitting.

3.2.2 Pipeline Architecture and Model Design

The proposed gear classification system incorporates MobileNetV2 as its backbone, chosen for its efficiency and suitability for edge deployment. The pipeline consists of the following components:

- **Input and Preprocessing:** RGB gear images, captured using a Logitech Brio camera, are resized, normalized, and augmented using random zoom, shift, and rotation operations. The system supports multiple resolutions.

- **Feature Extraction Backbone:** MobileNetV2 uses depthwise separable convolutions and residual bottlenecks to extract hierarchical features. Initial layers are frozen during early training to retain general patterns learned from ImageNet.

- **Classification Head:**
  - Global Average Pooling
  - Flatten layer
  - Dense layer with 128 ReLU-activated units
  - Dropout (0.5)
  - Dense softmax output layer with classes

- **Fine-Tuning:** Deeper layers of MobileNetV2 are gradually unfrozen and trained using a lower learning rate to specialize the model for gear-specific characteristics.

Alternative backbones such as VGG16 and ResNet-50 were explored for comparative evaluation. Both followed a similar strategy of freezing convolutional bases and appending custom heads. While informative, these architectures were ultimately not deployed due to their higher computational requirements compared to MobileNetV2.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

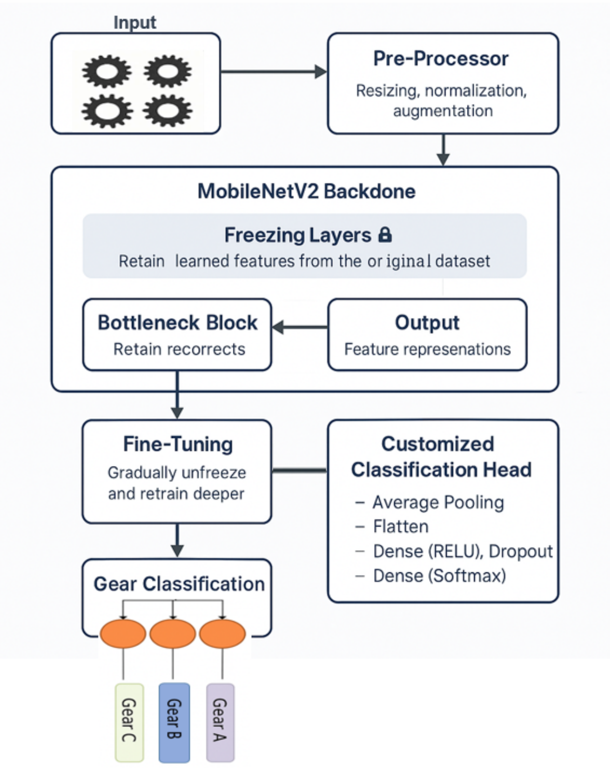


Figure 4: Model design and training pipeline for real-time gear classification.

3.2.3 Training Procedure and Optimization

To ensure stable, efficient, and generalizable training, the following optimization strategies were applied:

- **Data Augmentation:** A Keras-based augmentation pipeline generated 10 synthetic variants per training image through random rotations, flips, shifts, and zooming.
- **Learning Rate Scheduling:** An adaptive learning rate was used, starting at 0.001 with a decay factor of INIT_LR/EPOCHS, and further reduced during fine-tuning.
- **Model Checkpointing:** The model's weights were saved whenever validation loss improved, ensuring retention of the best-performing model:

```
ModelCheckpoint(fname,
monitor='val_loss',
mode='min',
save_best_only=True,
verbose=1)
```

- **Regularization:** Dropout (0.5) was applied to the classification head, and L2 weight decay was used to control model complexity.
- **Early Stopping:** Training was terminated if validation accuracy did not improve for 10 consecutive epochs, preventing overfitting and saving computational resources.
- **Layer Freezing Strategy:** The progressive unfreezing technique allowed early layers to retain generic features while deeper layers adapted to domain-specific patterns, striking a balance between generalization and specialization.

Together, these techniques formed a robust training strategy that ensured high classification performance while maintaining real-time suitability for industrial deployment. Figure 4 illustrates the architecture of the gear classification pipeline using transfer learning with MobileNetV2. The system begins with raw gear image inputs, which are first processed through a pre-processing block involving resizing, normalization, and augmentation to enhance robustness. The images are then passed to the MobileNetV2 backbone, where early layers are frozen to retain generic visual features learned from ImageNet. The feature representations are processed through bottleneck blocks and forwarded to a customized classification head consisting of global average pooling, flattening, a ReLU-activated dense layer with dropout, and a softmax output layer. A fine-tuning stage is employed, where deeper layers are progressively unfrozen and retrained to adapt to gear-specific features. The final output layer classifies the gears (e.g., Gear A, Gear B, Gear C) based on learned representations, enabling accurate and efficient real-time classification suitable for deployment in industrial environments.

4 Real-Time System Deployment

This section details the deployment of the proposed AI-based gear classification system within an industrial production environment, focusing on hardware-software integration, edge deployment setup, real-time pipeline implementation, and operator-oriented application design. The solution was developed as a modular, containerized desktop application tailored for execution on industrial-grade computing platforms within Early Production Containment (EPC) lines.

Built in Python using TensorFlow, OpenCV, and PySimpleGUI, the application integrates a pre-trained MobileNetV2 model adapted for classifying 50 gear types. The system runs locally on Intel Core i7-based Industrial Touch Panel PCs and includes mechanisms for operator interaction, gear image acquisition, real-time classification, and structured data logging. Containerization with Docker ensures consistency, scalability, and ease of maintenance across multiple production sites.

4.1 System Integration and Hardware Deployment

To support real-time gear classification at the Early Production Containment (EPC) lines, a compact and robust vision inspection station was developed and deployed. The standalone system includes a vibration-isolated table, a high-resolution Logitech Brio camera, a concentric custom LED ring light with diffuser, and an industrial-grade All-in-One

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Touch Panel PC—forming a reliable edge-computing unit suitable for continuous factory operation.

The camera is mounted on a vibration-dampened frame and calibrated to capture consistent top-down gear images, with adjustable exposure, focus, and white balance to accommodate varying surface finishes and geometries. The lighting system ensures uniform illumination while minimizing glare and shadows, critical for revealing fine structural details. Initial calibration procedures addressed overexposure and reflection artifacts by adjusting brightness and angle settings, thereby stabilizing imaging quality and reducing preprocessing overhead.

The inference engine, implemented in TensorFlow and optimized for edge performance, runs directly on the Touch Panel PC powered by an Intel Core i7-10710U processor. The PySimpleGUI-based user interface allows operators to monitor classification results, adjust resolution or threshold parameters, and control the inspection flow via touchscreen input. Integration into existing production workflows was conducted in collaboration with plant engineers to ensure ergonomic design, optimal hardware placement, and seamless inspection alignment.

This hardware-software integration enables efficient operation under high-throughput conditions while maintaining accuracy and usability in real-world factory settings.

4.2 Real-Time Recognition Pipeline

The real-time pipeline processes continuous image streams using OpenCV for frame capture, resizing, normalization, and color space conversion. Each processed frame is passed to the MobileNetV2 model for classification. The model's lightweight structure supports low-latency inference, suitable for high-speed production lines.

Classification results, including confidence scores, are displayed on-screen and overlaid onto live video. For frames with low-confidence predictions (below a set threshold), the system flags results for optional manual verification. The interface design balances automation with human oversight, providing visual feedback while allowing intervention when necessary. Ongoing optimizations ensure stable inference performance under real-world conditions.

4.3 Data Management and Traceability

The system includes a structured logging framework to support traceability and operational oversight. Classification metadata—such as timestamps, predicted labels, confidence scores, and operator actions—is recorded using the `openpyxl` library and stored in local Excel-based logs.

To support long-term scalability and fault tolerance, the system includes scheduled backup routines, periodic autosaving, and configuration logging. A monitoring dashboard provides operators with access to real-time operational metrics including gear throughput, alert frequency, and confidence distribution. The data management infrastructure facilitates downstream integration with enterprise quality systems and supports further analytics and process refinement.

This deployment establishes a production-ready infrastructure for intelligent gear inspection, integrating AI models with industrial hardware and workflows. Performance benchmarks and evaluation results are presented in Section 5.

Figure 5.1 presents the gear classification results alongside real-time data logging. Each entry includes a timestamp, the predicted gear class, the associated confidence score, and the final inspection status (OK or NG), enabling traceable and actionable insights during production.

5 Discussion and Performance Analysis

This section provides a comprehensive evaluation of the deployed AI-based real-time gear classification system across four Early Production Containment (EPC) lines at the SH Manufacturing Facility. Utilizing a MobileNetV2 model trained on the SH Gear Dataset (Section 3.1), the system achieves 92% classification accuracy with an operational throughput of 7,200 gears per hour—representing a tenfold improvement over manual inspection (720 gears/hour) and a 15–20% increase in accuracy. The analysis covers operational impact, throughput and reliability, scalability, hardware efficiency, and comparative model performance, offering insights into the system's deployment readiness and industrial viability.

Operational Efficiency and System Impact: The real-time classification system delivers 20 frames per second throughput with visual feedback via a PySimpleGUI interface, enabling swift and accurate gear sorting. This leads to an estimated 65–70% reduction in sorting time, while also reducing operator fatigue and improving decision consistency. The touch-enabled GUI is designed for ease of use and rapid adoption on the factory floor. Structured logging through `openpyxl` ensures traceability and supports quality audits.

Operator feedback has highlighted the interface's usability and the system's responsiveness under normal load, though minor latency under peak conditions suggests future iterations could benefit from GUI optimization.

The system's robust performance across varied gear types and operational conditions demonstrates its adaptability for broader intelligent inspection tasks, including weld seam evaluation, surface defect detection, and casting quality monitoring. This flexibility positions it as a scalable AI solution for high-precision manufacturing environments.

Throughput Enhancement and Error Mitigation: Compared to manual sorting—which is constrained by human fatigue and variability—the proposed system provides consistent performance with significantly higher throughput and accuracy. The model automatically flags low-confidence predictions (confidence < 0.8) for operator review, supporting a hybrid human-in-the-loop approach that minimizes classification errors without sacrificing speed.

The optimized inference pipeline maintains stable performance even under peak operational loads. Continuous performance logging enables trend analysis and guided retraining, ensuring sustained accuracy and supporting long-term reliability in high-volume production environments.

Scalability and Hardware Optimization The system's modular architecture enables scalable deployment with

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

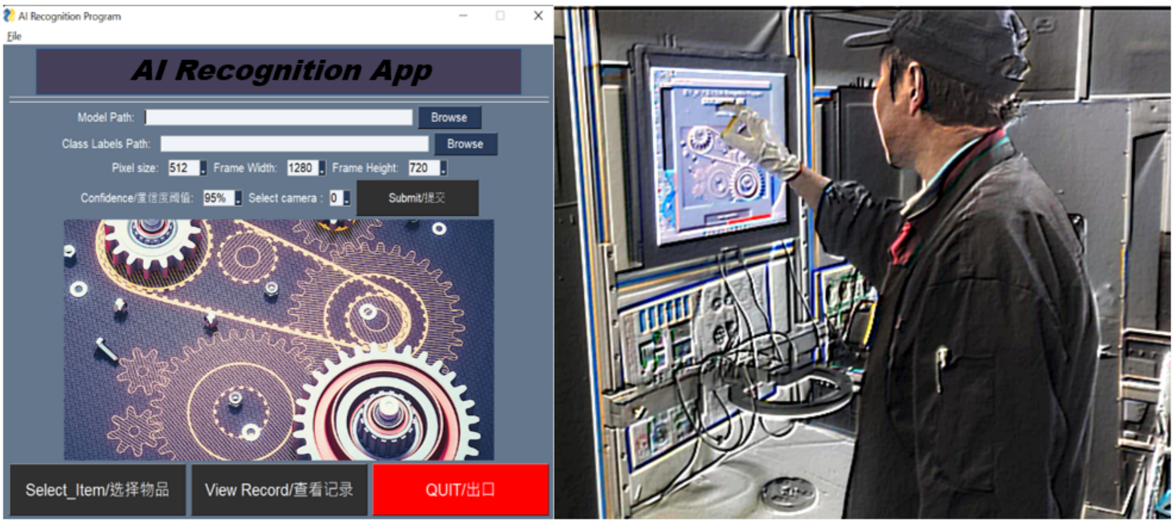
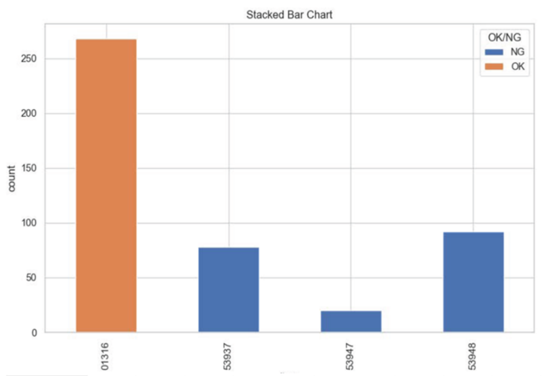


Figure 5: Real-time deployment of the gear classification system on an industrial touch panel PC, featuring a user interface (UI) for operator interaction, monitoring, and quality assurance.



Timestamp	Class_Names	Confidence	counts	OK/NG	Result_Inde	timestamp	Screen_count	OK_Item_Count	NG_Item_Count
2023-07-01 14:27:36	01316	99	1	OK	1	2023-07-01 14:27:53	1	1	0
2023-07-01 14:27:36	53948	97	1	NG	4	2023-07-01 14:28:27	2	1	1
2023-07-01 14:27:36	53947	98	1	NG	3	2023-07-01 14:28:46	3	1	2
2023-07-01 14:27:36	01316	99	29	OK	1	2023-07-01 14:29:08	4	2	2
2023-07-01 14:27:36	53937	96	1	NG	2	2023-07-01 14:29:19	5	2	3
2023-07-01 14:27:36	53937	97	4	NG	2	2023-07-01 14:29:50	6	2	4

Figure 6: Gear classification results with real-time data logging, displaying timestamps, predicted classes, confidence scores, and final inspection status (OK/NG).

minimal effort. For example, classification was successfully extended to five additional gear types using only 1,000 labeled images and a few extra training epochs. MobileNetV2 demonstrates superior efficiency compared to deeper models such as VGG16 and ResNet-50, excelling in training speed, memory consumption, and inference latency.

Figure 7 presents memory usage and inference latency for MobileNetV2, ResNet-50, and VGG16 across input resolutions from 224×224 to 720×720 pixels. As illustrated on the left, memory consumption increases with input size for all models; however, MobileNetV2 consistently requires the least memory, followed by VGG16 and ResNet-50, highlighting its suitability for resource-constrained environments. Correspondingly, MobileNetV2 achieves the lowest inference latency across all resolutions(as shown on the right), reinforcing its advantage for real-time applications. Conversely, VGG16 incurs the highest latency, especially at larger input sizes, which may restrict its use in latency-sensitive scenarios.

Deployed on an Intel Core i7-10710U Industrial Touch Panel PC, the system operates with approximately 30% lower energy consumption compared to GPU-based setups while maintaining stable performance under varying production loads. Its ability to handle different image resolutions (ranging from 224×224 to 720×720 pixels) without compromising stability affirms its robustness across diverse operational conditions. Operator feedback suggests that refining retraining protocols could further streamline adaptation when introducing new gear types.

Performance Evaluation Methodology: The evaluation framework, illustrated in Fig. 7, outlines the full classification pipeline—from real-time image capture to output monitoring. The system was assessed using both the SH Gear Dataset test set (1,000 images across 30–48 gear types) and live production data. Metrics evaluated include accuracy, precision, recall, latency, and throughput, offering a holistic view of system performance under realistic industrial conditions.

Quantitative Results and Model Comparison:

MobileNetV2 exhibited a well-balanced performance, achieving 92% accuracy, 0.91 precision, 0.90 recall, and a cross-entropy loss of 0.25, with an inference latency of only 20 ms per image. As shown in Fig. 8, VGG16 attained a slightly higher accuracy of 94%, but at the cost of increased latency (50 ms), while ResNet-50 reached 93% accuracy with even longer delays. Although these deeper models provided marginal accuracy gains, their higher inference times constrained their practicality for sustained real-time deployment.

Although the proposed model achieves an inference latency of under 20 ms per image, the overall system throughput of approximately 500 ms per gear represents the complete operational cycle of the industrial inspection process

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

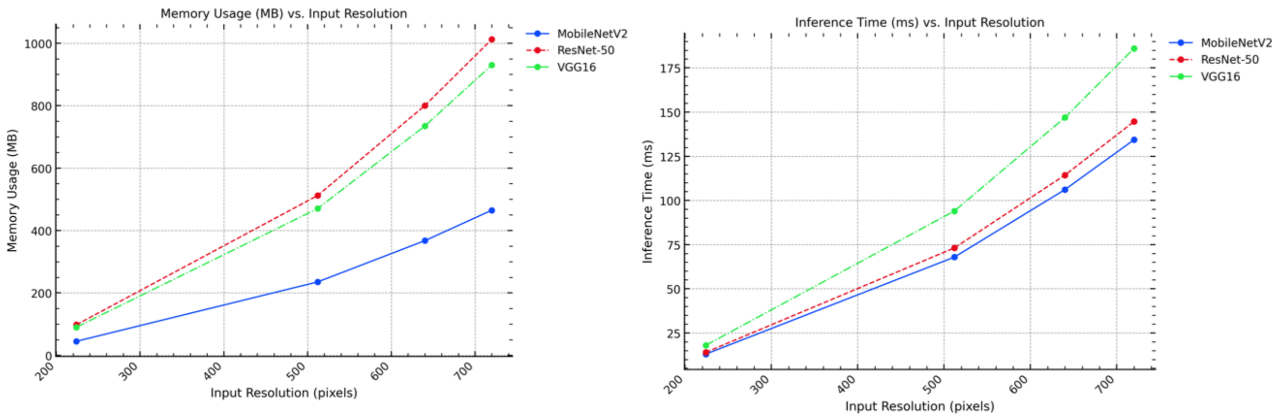


Figure 7: Model performance across input resolutions showing memory usage and inference time.

rather than the neural network's computation time alone. This end-to-end duration includes multiple stages: mechanical positioning of the gear, camera triggering and autofocus stabilization, lighting adjustment, image capture and transfer, preprocessing, inference, result visualization, and data logging with traceability verification.

In practical production environments, these non-algorithmic processes dominate the total cycle time. The 20 ms inference ensures that the vision subsystem does not become a bottleneck, maintaining real-time responsiveness even when synchronized with slower mechanical and sensor components. Consequently, the measured 500 ms cycle reflects integrated system timing, not computational delay.

VGG16 and ResNet-50, the higher-latency models, consume a greater slice of the processing window and thus are more susceptible to queue overflow, dropped frames, or synchronization failure at high-load operations. MobileNetV2's slender architecture, however, delivers uniformly low and deterministic latency to ensure smooth and stable performance across common industrial PCs or edge devices with limited GPU resources.

Furthermore, MobileNetV2 offers a balanced and deployment-ready solution. Its modest computational and memory requirements enable direct deployment on commodity industrial hardware without utilizing dedicated accelerators or making substantial infrastructure modifications. In contrast, heavier architectures are typically founded upon GPU-based processing, which can introduce cumulative timing delays across thousands of inspection cycles and reduce operational responsiveness.

MobileNetV2 also benefited from a false positive rate below 2%, supported by robust preprocessing techniques such as glare removal and contrast normalization that increased classification stability across varying light and surface conditions. The experiments for VGG16 and ResNet-50 started with their corresponding pretrained convolutional bases with added custom classification heads—VGG16's final convolutional blocks were selectively fine-tuned with smaller batch sizes for stable convergence, while ResNet-50's base layers were frozen to leverage residual learning.

While the deeper models were valuable in the experiment interpretation context, MobileNetV2 appeared to be the most appropriate option when considering the implications of inference, hardware availability, and relevance to real-time applications, as the most optimized for edge-based industrial equipment classification.

Comparison with Traditional Methods: Manual sorting methods yielded 75–80% accuracy and processed only 720 gears per hour, significantly lower than the proposed system. The AI-based approach not only improves both speed and accuracy but also introduces automatic traceability, reducing dependency on manual logging and improving consistency. Unlike VGG16 and ResNet-50, which are less suited to real-time demands due to latency, MobileNetV2 delivers a viable industrial solution that aligns with modern manufacturing efficiency standards.

Discussion and Implications: The results confirm that MobileNetV2 strikes an ideal balance between speed, accuracy, and efficiency, making it a robust choice for real-time, scalable gear classification. The system's design supports easy retraining and deployment, energy-efficient operation, and broad adaptability across gear types and related components. Its structured logging and hybrid feedback mechanism enable both traceability and continuous improvement.

Among pretrained architectures, ImageNet-based MobileNetV2 demonstrated the best trade-off between accuracy and inference speed for real-time gear classification. The use of transfer learning—via early layer freezing followed by selective fine-tuning—enabled rapid convergence and strong generalization to domain-specific features.

In summary, MobileNetV2, trained with ImageNet weights and optimized through staged fine-tuning and augmentation, offers the best trade-off for deployment in gear classification. It provides a scalable, sustainable solution for high-throughput, low-latency inspection and can serve as a blueprint for expanding AI-driven quality control across broader industrial domains.

6 Conclusion and Future Work

This study presents the successful design, deployment, and evaluation of a real-time AI-based gear classification system optimized for high-throughput industrial settings. By in-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

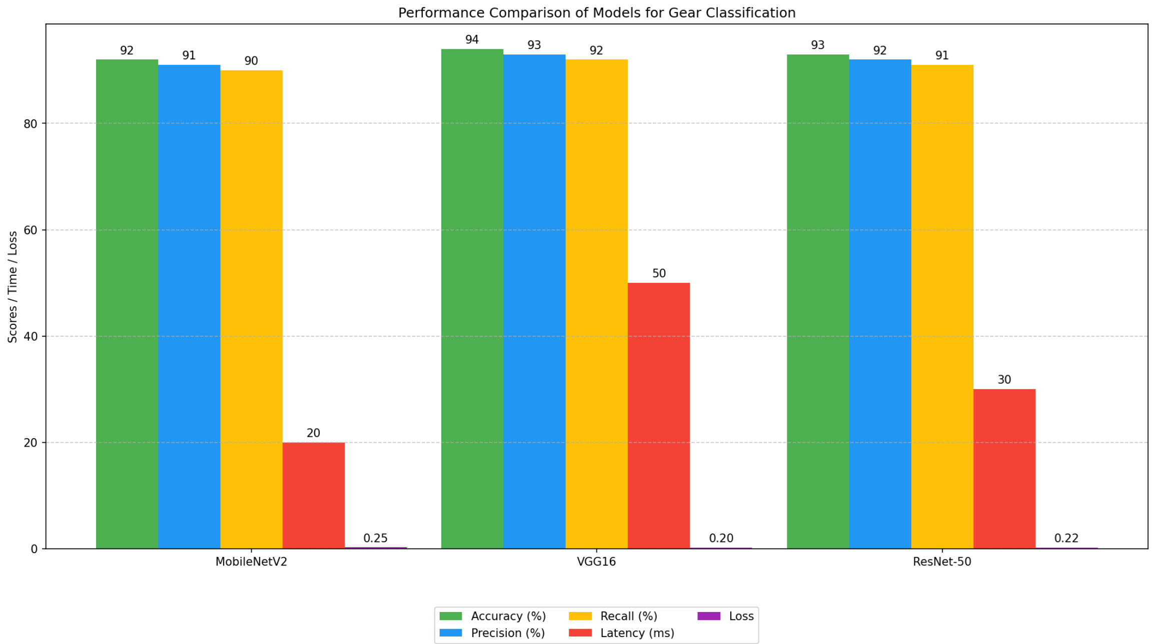


Figure 8: Performance comparison of MobileNetV2, VGG16, and ResNet-50 in gear classification.

tegrating the lightweight and efficient MobileNetV2 architecture into a modular, edge-deployable software–hardware pipeline, the system achieved substantial gains in accuracy, responsiveness, and scalability. Its implementation on Early Production Containment (EPC) lines at the SH Manufacturing Facility resulted in a tenfold increase in throughput and significantly improved inspection reliability, effectively modernizing legacy quality control processes.

MobileNetV2 was chosen for its strong trade-off between precision and computational efficiency,transfer-learning consistency, hardware capability, and efficiency. outperforming more resource-intensive models such as VGG16 and ResNet-50 in latency-critical environments. The system's seamless integration into industrial workflows—featuring a responsive user interface and automated ISO-compliant data logging—supports scalable deployment and real-time traceability.

Achieving 92% classification accuracy and processing over 7,000 gears per hour, the system establishes a strong foundation for further development. The following subsections outline enhancements aimed at improving model performance, expanding cross-domain applicability, and strengthening infrastructure to meet evolving manufacturing demands.

Enhancing Model Accuracy and Robustness. While MobileNetV2 delivers excellent performance for real-time gear classification, future work will focus on addressing edge cases such as glare-induced errors, fine-grained gear similarities, and varying surface finishes. Ensemble methods combining lightweight and deeper models (e.g., ResNet-50 or VGG16) may improve feature representation, while attention mechanisms—both spatial and channel-based—can help focus on relevant visual patterns such as gear teeth or micro-wear features.

Additional improvements will involve advanced preprocessing techniques (e.g., adaptive histogram equalization, reflection suppression) and data augmentation across diverse lighting and orientation scenarios. Transitioning to higher-resolution imaging (e.g., 1024×1024 pixels) will enable defect-level recognition, such as detecting cracks, abrasions, or dimensional inconsistencies—thereby expanding the system's role toward predictive maintenance. Automated hyperparameter tuning (e.g., via Bayesian optimization) will be employed to improve training efficiency and consistency.

Cross-Domain Deployment and Adaptability. The system's modular architecture and transfer learning-based training strategy make it adaptable beyond gear classification. Future extensions will target other critical inspection tasks such as weld seam defect detection, surface crack classification, casting flaw identification, and real-time monitoring of heat-affected zones or machined finishes. These applications will benefit from specialized datasets and task-specific fine-tuning.

To support rapid domain adaptation, a cloud-based retraining pipeline is planned enabling remote dataset curation, automated model optimization, and deployment with minimal downtime. This platform aims to reduce retraining and integration time for new use cases to under five hours. Field validation with industry partners in automotive, aerospace, and electronics manufacturing will assess generalizability and support wider adoption across smart factory ecosystems.

Hardware Optimization and Software Improvements. Although current deployment on Intel Core i7-10710U–based industrial PCs supports robust performance, deeper model variants remain constrained by latency requirements. Upgrading to AI-accelerated edge platforms

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

such as NVIDIA Jetson Orin (capable of up to 200 TOPS) will enable real-time deployment of larger architectures with sub-30 ms inference latency—without compromising throughput or responsiveness.

The focus of the future development of the software will be on replacing the current Excel-based logging system with a more advanced database (like SQLite or PostgreSQL) that allows for greater accuracy of data and scalability for inspecting over ten thousand parts a day. In addition to that, improvements to performance will be made through the use of asynchronous task management, and background data processing will provide a much more responsive user interface that will not crash under the load of high-volume operations. Future versions will also have the ability to connect remotely and integrate with the existing company infrastructure, which will allow for centralized monitoring and diagnostics, as well as updates to all systems, across multiple locations from a single point of control. These advanced elements are intended to provide effective solutions not only for predictive maintenance and operational tracking but also to provide a future-proof solution with optimised energy efficiency, horizontal scalability from an expanded product line, and fast adaptability to the rapidly changing needs of today's manufacturing environments for years to come.

To sum it up, this project strongly indicates what's possible with a flexible, real-time AI system for sorting gears, which could change industrial inspections for good. We developed and put this system into action between 2023 and 2024, making it the very first time AI was used for gear inspection at our site. This was at a time when deep learning was just starting to catch on in similar factories. Considering how fast AI has moved since then, our work here serves as both a useful example and an early guide for others looking to use AI in manufacturing. Looking ahead, we'll concentrate on making the models even tougher, allowing them to work in different areas, and making sure the hardware and software fit together perfectly. This will help make our system a core part of intelligent, automated quality control in modern manufacturing.

References

[1] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," arXiv preprint arXiv:1604.07316, 2016.

[2] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[3] H. Liu and et al., "Micronet: A lightweight cnn for real-time defect detection in semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 33, no. 3, pp. 379–388, Aug 2020.

[4] J. Lee, B. Bagheri, and H. A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.

[5] X. Zhao and et al., "A comprehensive review of deep learning in visual inspection and quality control," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 1, pp. 123–139, Jan 2024.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[7] Z. Li, X. Zhang, Y. Zhang, and W. Wang, "Application of deep convolutional neural networks in detecting surface defects on industrial products," *Sensors*, vol. 19, no. 3, p. 496, 2019.

[8] A. Smith, B. Johnson, and C. Lee, "Machine learning for industrial inspection: A review of classical and modern approaches," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 6, pp. 6789–6801, 2023.

[9] X. Zhang, Y. Li, and Z. Wang, "Classical machine learning techniques for automated visual inspection in manufacturing," *Journal of Manufacturing Systems*, vol. 68, pp. 123–135, 2023.

[10] H. Chen, Q. Liu, and M. Zhao, "Enhancing industrial quality control with classical statistical learning methods," *International Journal of Production Research*, vol. 61, no. 15, pp. 5102–5118, 2024.

[11] K. Patel, R. Kumar, and S. Gupta, "Evaluating manual sorting techniques in automotive component inspection," *Journal of Manufacturing Processes*, vol. 97, pp. 345–356, 2023.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[13] Y. Cao, X. Jiang, D. Zhang, Y. Lin, and M. Gong, "A comprehensive survey on deep learning-based methods for industrial surface defect detection," *Engineering Applications of Artificial Intelligence*, vol. 100, p. 104187, 2021.

[14] F. Yan, Z. Yang, M. Xu, and X. Liang, "Industrial inspection ai: Challenges, datasets, and algorithms," *Journal of Manufacturing Systems*, vol. 62, pp. 933–947, 2022.

[15] R. Tan, M. Li, Y. T. Lim, and K. C. Tan, "Ai for manufacturing: Challenges and opportunities," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2261–2270, 2021.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Miami, FL, USA, 2009, pp. 248–255.

[17] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 27, 2014, pp. 3320–3328.

[19] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Factors of transferability for a generic convnet representation," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2015, pp. 27–28.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4510–4520.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[23] A. G. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1314–1324.

[24] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, 2000.

[25] A. K. Jain, "Statistical pattern recognition: A review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[26] J. Lee, B. Bagheri, and H. A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing Letters*, vol. 3, pp. 18–23, 2015.

[27] A. Smith, B. Johnson, and C. Lee, "Machine learning for industrial inspection: A review of classical and modern approaches," *IEEE Transactions on Industrial Informatics*, vol. 19, pp. 6789–6801, 2023.

[28] M. A. Khan, R. Hassan, and M. Ali, "Machine learning for industry 4.0: A systematic review using deep learning-based topic modelling," *Sensors*, vol. 22, no. 22, p. 8641, 2022.

[29] H. Liu *et al.*, "Micronet: A lightweight cnn for real-time defect detection in semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 33, pp. 379–388, 2020.

[30] C. Zhang, R. Liu, and Y. Wang, "Using deep learning to detect defects in manufacturing: A comprehensive survey and current challenges," *Materials*, vol. 13, no. 24, p. 5755, 2020.

[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[32] P. Bergmann, K. Batzner, D. Sattlegger, and C. Steger, "Deep learning for unsupervised anomaly localization in industrial images: A survey," *arXiv preprint*, 2022.

[33] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105. [Online]. Available: https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[35] L. Zhang, T. Zhou, and Y. Wang, "Gear surface defect detection based on lightweight cnns and attention mechanisms," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4632–4643, 2022.

[36] X. Zhao *et al.*, "A comprehensive review of deep learning in visual inspection and quality control," *IEEE Transactions on Industrial Informatics*, vol. 20, pp. 123–139, 2024.

[37] A. Antoniou, A. Storkey, and H. Edwards, "Data augmentation generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[38] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2020.

[39] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pp. 270–279, 2018.

[40] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[41] L. Xu and W. Zhang, "Surface defect classification of gears based on transfer learning with vgg16," *Journal of Manufacturing Processes*, vol. 56, pp. 621–630, 2020.

[42] J. Zhang, Y. Li, and W. Chen, "Gear surface defect recognition using deep convolutional neural networks and transfer learning," *Applied Sciences*, vol. 11, no. 5, p. 2348, 2021.

[43] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019.

[44] J. Li, Y. Zhao, and Q. Xie, "Deep learning-based industrial visual inspection: A survey," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5797–5810, 2021.

[45] Y. Wang, J. Sun, and F. Liu, "Automated visual inspection of mechanical components using resnet-based transfer learning," *IEEE Access*, vol. 10, pp. 58 760–58 771, 2022.

[46] H. Lin and L. Zhao, "Real-time surface defect detection for industrial manufacturing using lightweight cnns," *Sensors*, vol. 20, no. 16, p. 4633, 2020.

[47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[48] A. Canziani, A. Paszke, and E. Culurciello, "Analysis of deep neural network models for practical applications," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, 2016, pp. 1–8.

[49] W. Li, Q. Zhang, and X. Liu, "Gear surface defect detection based on attention-guided convolutional neural network," *Applied Sciences*, vol. 11, no. 2, p. 586, 2021.

[50] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021.

[51] J. Li, S. Qin, F. Liu *et al.*, "A review on deep learning-based fault diagnosis in manufacturing," *Journal of Manufacturing Systems*, vol. 62, pp. 654–678, 2022.

[52] M. Gulzar *et al.*, "Gear image classification using convolutional neural networks," *Journal of Mechanical Systems*, 2023.

[53] K. Sridhar *et al.*, "Hybrid cnn-autoencoder model for multi-gear classification," *Engineering Applications of Artificial Intelligence*, 2023.

[54] A. Mamat *et al.*, "Yolo-based gear defect detection in automotive assembly," *Expert Systems with Applications*, 2023.

[55] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[56] International Organization for Standardization, "Iso 9001:2015 quality management systems — requirements," 2015.

[57] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2015, pp. 234–241.

[58] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[59] Y. Sun, T. Liu, and J. Wang, "Multimodal deep learning for industrial visual inspection: A survey," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 6970–6982, 2022.

[60] H. Wang, S. Wang, B. Liu, and Y. Zhang, "A review on deep learning techniques for industrial surface defect inspection," *Neurocomputing*, vol. 489, pp. 1–19, 2022.

[61] W. Zhang, Y. Yang, and P. Chen, "Deep learning-enabled predictive maintenance for industrial machinery: A review," *Journal of Manufacturing Systems*, vol. 60, pp. 703–719, 2021.

[62] A. Mosavi, E. Salwana, and T. Rabczuk, "Industrial applications of ai: A review of machine learning methods for fault detection in predictive maintenance," *Engineering Applications of Artificial Intelligence*, vol. 95, p. 103894, 2020.

[63] Logitech, "Logitech brio ultra hd for business – usb 4k 90° autofokus," Suprag.ch. [Online]. Available: https://www.suprag.ch/de/shop/product/prod/102711-ptz-und-webcams/775269000000-logitech-brio-ultra-hd-for-business-usb-4k-90-autofokus/, accessed: May 28, 2025.

[64] G. Bradski, "The opencv library," *Dr. Dobb's Journal of Software Tools*, vol. 25, no. 11, pp. 120–125, 2000.

[65] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 818–833.

[66] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning (ICML)*, 2015, pp. 448–456. [Online]. Available: https://arxiv.org/abs/1502.03167

[67] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[68] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html