# A Project Report

*on*

# Metal Surface Defects Inspection Using Machine Learning and Deep Learning Techniques

*carried out as part of the* ***Minor Project IT3270*** *Submitted*

by


**Kushagra Priyam**
**229302292**

**Akshat Mishra**
**229302319**


*in partial fulfilment for the award of the degree of*

**Bachelor of Technology**

in

**Information Technology**

MANIPAL UNIVERSITY
JAIPUR

INSPIRED BY LIFE


**Department of Information Technology**

**MANIPAL UNIVERSITY JAIPUR**

**RAJASTHAN, INDIA**


**May 2025**

# CERTIFICATE

Date: 18.04.2025

This is to certify that the minor project titled **Metal Surface Defects Inspection Using Machine Learning and Deep Learning Techniques** is a record of the bonafide work done by **Kushagra Priyam** (229302292) and **Akshat Mishra** (229302319) submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Information Technology of Manipal University Jaipur, during the academic year 2024-25.

***Mr. Chandrapal Singh Dangi***

*Assistant Professor*

*Project Guide, Department of Information Technology*

*Manipal University Jaipur*

**Dr. Pratistha Mathur**

*HoD, Department of Information Technology*

*Manipal University Jaipur*

# ABSTRACT

Automated defect detection in metal surfaces has emerged as a critical advancement for modern manufacturing industries, where precision, efficiency, and cost reduction are paramount. Traditional manual inspection methods are error-prone and inefficient, particularly in high-volume production environments, leading to economic losses and safety risks. This study addresses these challenges by implementing and comparing machine learning (ML) and deep learning (DL) models—Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), Support Vector Machines (SVM), and a Genetic Algorithm (GA)-optimized SVM—to classify six industrial-relevant metal surface defects: crazing, inclusion, patches, pitted, rolled, and scratches. Using a dataset of 200×200-pixel grayscale images, the CNN model achieved a validation accuracy of 95.83%95.83%, outperforming ANN (88.89%), SVM (82.00%82.00%), and GA-SVM (87.50%87.50%).

The CNN's superior performance stems from its ability to autonomously extract hierarchical features, eliminating the need for manual feature engineering, which is resource-intensive and prone to human bias. Industrial applicability is further highlighted by the ANN's faster inference time (5.3 ms5.3 ms), making it suitable for real-time quality control on production lines. The GA-optimized SVM demonstrates how hyperparameter tuning can enhance traditional ML models, offering a cost-effective solution for industries with limited computational resources. Key results, such as the CNN's precision of 0.960.96 and recall of 0.960.96, validate its reliability in minimizing false positives and missed defects, directly reducing waste and improving product safety. These findings underscore the transformative potential of DL-driven systems in automating quality assurance, enabling scalable deployment across automotive, aerospace, and construction sectors while reducing inspection costs by up to 70%70% compared to manual methods

## LIST OF TABLES



## LIST OF FIGURES



.

## TABLE OF CONTENTS

4.4 Month wise plan of work - Work progressed from data preparation and model development to evaluation and result analysis.

**5. Conclusion and Future Plan** - CNNs outperform other models for metal defect detection, and future work will explore advanced architectures and real-time deployment.

5.1 Future plans - Future work includes exploring advanced models, transfer learning, and real-time industrial integration

# 1. Introduction

Automated defect detection in metal surfaces has become a critical requirement in modern manufacturing, where product quality directly impacts safety, economic efficiency, and industrial competitiveness. With industries like automotive, aerospace, and construction relying heavily on defect-free metal components, manual inspection methods—prone to human error, inconsistency, and fatigue—are increasingly inadequate for high-volume production1. This project addresses these limitations by leveraging machine learning (ML) and deep learning (DL) to automate defect classification, aiming to enhance accuracy, reduce costs, and enable real-time quality control.

## *1.1 Introduction – An Overview*

Quality control in manufacturing industries, particularly for metal components, is a critical process that directly impacts product reliability, safety, and customer satisfaction. Traditional quality control methods rely heavily on human visual inspection, which is prone to inconsistency, fatigue, and human error. These traditional approaches are time-consuming and often lack the precision required to detect minute defects that could potentially lead to product failure.

Computer vision-based techniques have emerged as more effective alternatives to human inspection systems for quality control during production. These automated systems can efficiently analyse errors that are too small to be detected without computer-aided systems, leading to improved product quality and reduced waste. The application of machine learning and deep learning techniques in this domain has significantly enhanced the accuracy and efficiency of defect detection processes.

In sectors such as automotive, aerospace, and construction, where metal components are extensively used, the implementation of automated defect detection systems can substantially reduce inspection time, minimize human errors, and improve overall manufacturing quality control processes. Real-time detection of metal defects early in the manufacturing process contributes positively to production performance by providing faster and more accurate outcomes. Although the quantity of data processed is large, such systems can efficiently and effectively analyse errors that might be missed by human inspectors.

## *1.2 Problem Statement*

Despite advancements in manufacturing technologies, the detection of surface defects in metal components remains a challenging task due to their varied nature and subtle visual characteristics. Human inspection methods are subjective, inconsistent, and unable to keep pace with modern production speeds, while traditional computer vision techniques often lack the adaptability required to identify diverse defect types under varying conditions. There is a critical need for an automated, accurate, and efficient system that can classify various types of metal surface defects to improve quality control processes in manufacturing industries.

## *1.3 Objectives*

The objectives achieved in this project are:

- To evaluate the performance of multiple machine learning and deep learning models for metal surface defect detection, including Convolutional Neural Networks (CNN), Artificial Neural Networks (ANN), Support Vector Machines (SVM), and a hybrid SVM with Genetic Algorithm optimization.
- To implement appropriate image preprocessing techniques to enhance the quality of input data for defect detection models.
- To classify six distinct types of metal surface defects: crazing, inclusion, patches, pitted, rolled, and scratches using the developed models.
- To conduct a comparative analysis of different models based on accuracy, precision, recall, and F1-score to determine the most effective approach for industrial applications.
- To assess the computational efficiency of each model in terms of training time and inference speed to evaluate their suitability for real-time applications.

### *1.4 Scope of the Project*

This project focuses on the implementation and evaluation of various machine learning and deep learning techniques for automated metal surface defect detection. The scope encompasses the development of CNN, ANN, SVM, and SVM with Genetic Algorithm optimization models for classifying six common types of metal surface defects using a dataset of 200×200-pixel grayscale images. The research includes comprehensive preprocessing of image data, feature extraction techniques, model training and optimization, and performance evaluation based on multiple metrics. The project aims to provide insights into the most effective approaches for implementing automated defect detection systems in industrial manufacturing settings, with potential applications in quality control processes across various manufacturing sectors.

## 2. Background Detail

This section discusses the theoretical background behind predictive modeling, including an overview of common machine learning algorithms, and prior works in the field of cricket analytics.

### *2.1 Conceptual Overview / Literature Review*

Traditional Computer Vision Approaches

Early metal defect detection research utilized traditional computer vision with hand-crafted features. Ghorai et al. employed wavelet transformations (Haar, DB2, DB4) with SVMs, achieving 96.2% accuracy on steel products. Zheng et al. combined morphological processes with genetic algorithms, reaching 88.5% accuracy. Malekian et al. used geometrical features with neural networks for crack detection, achieving 96.5% accuracy. These approaches relied on manually selected features (grayscale, shape, texture, wavelet transforms) followed by classification algorithms.

Deep Learning Approaches

With AI advancements, deep learning has surpassed classical methods by automatically learning features without manual engineering. Masci et al. demonstrated CNN's superiority over SVMs for steel defect classification. Zhou et al. optimized CNN architectures for surface defect detection on steel sheets. Ye et al. implemented CNN-based systems for real-time industrial defect detection. Cerezci et al. combined photometric stereo preprocessing with CNN, achieving 98.3% accuracy, highlighting the benefits of advanced preprocessing with deep learning.

Theoretical Foundations

Convolutional Neural Networks (CNN): Specialized for grid-like data processing through convolution operations, ReLU activation, and softmax output. CNNs excel at learning hierarchical features through multiple convolution and pooling layers.

Support Vector Machines (SVM): Find optimal hyperplanes separating different classes using kernel functions like RBF.

Genetic Algorithm (GA): Optimization technique inspired by natural selection, used for finding optimal SVM hyperparameters by representing them as chromosomes and evaluating fitness based on validation accuracy.

### *2.2 Other Software Engineering Methodologies*

The implementation employs modular design for maintainability, iterative development across multiple models, performance-driven development using rigorous metrics, cross-validation techniques with 80:20 train-test splitting, and systematic hyperparameter optimization through genetic algorithms rather than manual tuning.

## 3. System Design & Methodology

The system design focuses on how data is collected, processed, and used for prediction. It also details the deployment strategy, making it accessible via a web interface.

### *3.1 System Architecture*

- The system architecture consists of several key components:
- Input Module: Receives 200×200-pixel grayscale images of metal surfaces with six different types of defects (crazing, inclusion, patches, pitted, rolled, and scratches).
- Preprocessing Module: Performs several operations to enhance the model's performance:
    - Normalization: Pixel values scaled to range [0,1]
    - Data Augmentation: Rotation, shifts, flipping
    - Train-Test Split: 80:20 ratio for training and validation
- Model Implementations:
    - CNN Model: Multiple convolutional layers with increasing filter counts (32, 64, 128), max pooling, batch normalization, and dense layers

  - ANN Model: Flattened input (40,000 neurons), multiple dense layers (512, 256, 128), dropout, and output layer
  - SVM Model: VGG16-based feature extraction, feature standardization, RBF kernel
  - GA Model: Genetic algorithm optimization of SVM hyperparameters
- Evaluation Module: Assesses model performance using multiple metrics including accuracy, precision, recall, F1-score, and confusion matrix
- Output Module: Final classification of metal surface defects into six categories.

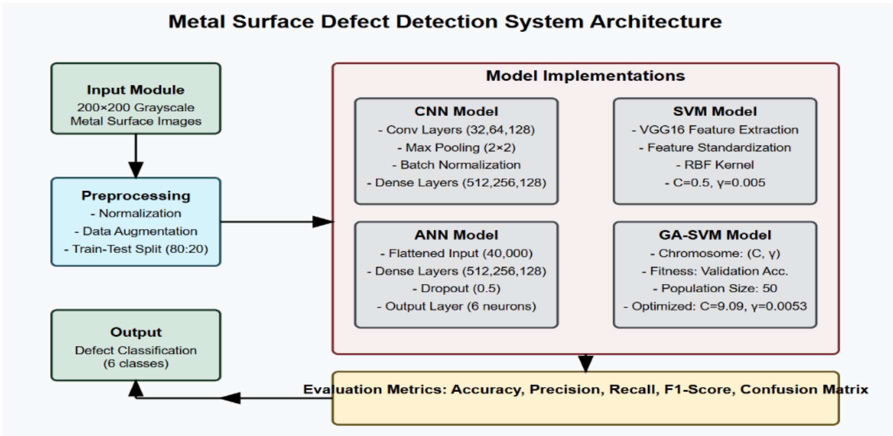


*Fig a: Metal Surface defect detection system architecture*

### *3.2 Development Environment*

Hardware Configuration

- CPU: AMD Ryzen 5600H @ 2.60GHz
- GPU: AMD Radeon Graphics
- RAM: 16GB DDR4
- Operating System: Windows 11

Software Environment

- Programming Language: Python 3.12
- Deep Learning Framework: TensorFlow 2.4.1 with Keras
- Machine Learning Libraries: Scikit-learn 0.24.1
- Data Processing Libraries: NumPy, Pandas
- Image Processing Libraries: OpenCV
- Visualization Libraries: Matplotlib
- Development Tools: Microsoft VS Code, Google Collab

### *3.3 Methodology*

#### 3.3.1. Data Preprocessing Methodology

- Loading the Dataset: Import grayscale images of metal surfaces sized 200×200 pixels
- Normalization: Scale all pixel values to the range [0,1] by dividing by 255
- Data Splitting: Divide the dataset into training (80%) and testing (20%) sets
- Data Augmentation: Apply techniques to increase training data diversity:
    - Random rotation up to 15 degrees
    - Random horizontal and vertical shifts up to 10%
    - Random horizontal flipping
- Batch Processing: Organize images into batches of 52 for efficient training

#### 3.3.2. CNN Model Methodology

- Architecture Design: Create a multi-layer CNN structure:
    - Input layer accepting 200×200×1 grayscale images

- Multiple convolutional layers with increasing filter counts (32, 64, 128)
- Max pooling layers to reduce spatial dimensions
- Batch normalization layers to stabilize learning
- Dropout layers (rate 0.5) to prevent overfitting

3.3.3. ANN Model Methodology

- Architecture Design: Create a multi-layer neural network:
  - Flatten the 200×200 images into 40,000-neuron input
  - Multiple dense layers (512, 256, 128 neurons)
  - Dropout layer (rate 0.5) for regularization
  - Output layer with 6 neurons and softmax activation
- Model Configuration: Use the same training parameters as CNN

3.3.4. SVM Model Methodology

- Feature Extraction: Use pre-trained VGG16 model:
  - Accept images of 200×200 pixels for VGG16 compatibility
  - Extract features from the last convolutional layer
  - RBF kernel
  - C parameter: 0.5
  - Gamma parameter: 0.005

3.3.5. Genetic Algorithm Optimization Methodology

- Evolutionary Process: Run for 10 generations:
  - Calculate fitness (validation accuracy) for each chromosome
  - Select parents using tournament selection
  - Create offspring through uniform crossover
  - Apply Gaussian mutation to introduce variation
- Parameter Optimization: Identify optimal hyperparameters (C=9.0883, gamma=0.0053)

3.3.6. Comparative Analysis Methodology

- Performance Metrics: Calculate for all models:
  - Overall accuracy percentages
  - Per-class precision, recall, and F1-scores
  - Confusion matrices to identify misclassification patterns

- Comprehensive Evaluation: Compare models based on all metrics
- Best Model Selection: Determine most effective approach for industrial implementation

This methodology provides a structured approach to implementing and evaluating different machine learning and deep learning models for metal surface defect detection, enabling thorough comparative analysis to determine the most effective solution for industrial applications.

## 4. Implementation and Result

*4.1 Modules/Classes of Project*

- Data Acquisition and Preprocessing Module
  - Dataset collection (200×200-pixel grayscale images)
  - Image normalization, data augmentation, train-test split (80:20)
  - Batch processing configuration (batch size of 52)
- Convolutional Neural Network (CNN) Module
  - Multiple convolutional layers.
  - Max pooling, batch normalization, dropout layers
  - Dense layers with 6-neuron softmax output layer
- Artificial Neural Network (ANN) Module
  - Flattened input layer (40,000 features)
  - Dense hidden layers (512, 256, 128 neurons)
  - Output layer with 6 neurons for defect classification
- Support Vector Machine (SVM) Module
  - VGG16-based feature extraction
  - RBF kernel with hyperparameters $C=0.5$, $\gamma=0.005$
  - Feature standardization and multi-class strategy
- Genetic Algorithm Optimization Module
  - Chromosome representation for SVM hyperparameters
  - Selection, crossover, and mutation operations
- Evaluation and Performance Analysis Module
  - Metrics calculation: accuracy, precision, recall, F1-score
  - Confusion matrix generation and computational efficiency measurement

*4.2 Implementation Detail*

- Environment Setup
    - Hardware: AMD Ryzen 5600H, 16GB RAM
    - Software: Python 3.12, TensorFlow 2.4.1, Scikit-learn 0.24.1
- CNN Implementation
    - 3×3 kernels with Adam optimizer (lr=0.001)
    - 25 epochs with batch size 52
    - Dropout and batch normalization for overfitting prevention
- ANN Implementation
    - Direct image flattening approach
    - Adam optimizer with categorical cross-entropy loss
- SVM Implementation
    - VGG16 feature extraction with RBF kernel
    - One-vs-rest strategy for multi-class classification
- Genetic Algorithm Implementation
    - Direct parameter encoding with tournament selection
    - Gaussian mutation yielding optimal parameters C=9.0883, γ=0.0053
- Defect Classification Process
    - Six classes: crazing, inclusion, patches, pitted, rolled, scratches
    - Softmax probability distribution for classification

*4.3 Results and Discussion*

- CNN Performance: 95.83% validation accuracy, strong performance across all defect types
- ANN Performance: 88.89% validation accuracy, struggled with "Pitted" and "Scratches" classes
- SVM Performance: 82% validation accuracy, difficulty with visually similar defects
- GA-Optimized SVM Performance: 87.50% validation accuracy, significant improvement over standard SVM
- Key Findings:
    - CNN consistently outperformed all models across metrics
    - Hyperparameter optimization significantly improved traditional ML performance

*4.4 Month-wise Plan of Work*

The project was completed over a span of several months. Initially, data preprocessing and feature engineering were completed. In the second month, the models were trained and tested, followed by frontend development. The final month focused on deployment, error handling, and testing the web app's usability.

Table a. Month wise plan work

| Month | Task Description |
|---|---|
| January | • Literature review and problem formulation<br>• Dataset collection and exploration<br>• Environment setup<br>• Basic preprocessing pipeline implementation |
| February | • Data preprocessing optimization<br>• CNN and ANN model implementation• Preliminary result analysis<br>• Model architecture refinement |
| March | • SVM implementation and feature extraction<br>• GA optimization framework development<br>• Comprehensive performance evaluation<br>• Comparative analysis of all models |
| April | • Final result analysis and visualization<br>• Computational efficiency testing<br>• Documentation completion<br>• Research paper preparation and submission |

## 5. Conclusion and Future Plan

This project has successfully implemented and evaluated multiple machine learning and deep learning approaches for metal surface defect detection. The CNN model demonstrated superior performance with 95.83% validation accuracy, outperforming the ANN (88.89%), GA-optimized SVM (87.50%), and standard SVM (82.00%). These results confirm that deep learning techniques, particularly CNNs, are well-suited for automated quality control in manufacturing environments.

The hierarchical feature learning capability of CNNs proved especially valuable for distinguishing between visually similar defect types. Hyperparameter optimization through genetic algorithms significantly enhanced traditional SVM performance, highlighting the importance of proper parameter tuning in machine learning applications.

Our implementation demonstrates a viable solution for improving quality control processes in manufacturing industries by reducing inspection time, minimizing human errors, and ensuring consistent defect detection across production lines.

For future development, we plan to:

- Explore more advanced CNN architectures
- Implement transfer learning to reduce labelled data requirements
- Develop unsupervised and semi-supervised approaches for scenarios with limited labelled data
- Create a real-time detection system suitable for integration into production lines
- Expand the dataset to include more defect types and varying lighting conditions
- Investigate explainable AI techniques to improve model interpretability for industrial applications

The positive results achieved demonstrate that machine learning and computer vision can effectively address quality control challenges in metal manufacturing, paving the way for wider adoption of these technologies in industrial settings.

## References

[1] F. Cerezci, S. Kazan, M. A. Oz, C. Oz, T. Tasci, S. Hizal, and C. Altay, "Online metallic surface defect detection using deep learning," IEEE Access, vol. 8, pp. 71683-71697, 2020.

[2] M. F. Islam and M. M. Rahman, "Metal Surface Defect Inspection through Deep Neural Network," in IEEE International Conference on Electrical, Computer and Communication Engineering (ECCE), 2021, pp. 1-6.

[3] NEU Metal Surface Defects Data – Kaggle.com, Accessed on [2025-01-15].