

Image Classification for Industrial Equipment Defect Detection

as Part of the Application Requirements

for the Internship Position in
Machine Learning Internship

At

Eric Robotics

by

Omkar Jagtap

Guided By

Chirag Jhawar & Rohit Panikar



Abstract

model designed to categorize images of industrial machinery as either "defective" or "non-defective." The main objective was to develop a precise classification system capable of identifying defects, which can potentially be integrated into quality control processes within industrial environments. The project included preparing the dataset, training the model, and evaluating its performance using standard metrics such as accuracy, precision, and recall. This approach has the potential to enhance the efficiency of defect detection and decrease the need for manual inspections.

Objective

The project objective is to:

1. Develop a dataset containing images of industrial equipment labeled as "defective" or "non-defective."
2. Create and train a machine learning model to classify these images into the specified categories.
3. Evaluate model performance using classification metrics like accuracy, precision, and recall.

Introduction

Quality control plays a main role in industrial manufacturing, where ensuring that equipment is free from defects is crucial for safety and proper functioning. Conventional techniques for identifying defects in equipment are typically manual, labor-intensive, and exposed to human error.

In this project, we applied the Convolutional Neural Network(CNN) technique to streamline the classification of images of industrial equipment into defective and non-defective categories. This automation seeks to improve quality control procedures by offering a scalable, efficient, and precise approach to defect identification.

For this project, I utilized the MVTec Anomaly Detection (MVTec AD) Dataset, known for its effectiveness in identifying flaws in industrial items. This collection comprises high-resolution images of various kinds of industrial components, including some with visible defects.

I made several modifications to the dataset to align it with the requirements of our project:

1. **Labels:** Each picture was categorized as either "defective" (if it exhibited a defect) or "non-defective" (if it was without defects).
2. **Categories:** We arranged the images according to the type of industrial component, which enabled us to analyze defect trends for each category.

To ensure our model could generalize well to new data, I divided the dataset into three segments:

- **Training Set:** Used for instructing the model.
- **Validation Set:** Employed for assessing the model's effectiveness during training.
- **Testing Set:** Reserved for the final assessment to evaluate the model's performance in realistic scenarios.

Data Preprocessing

For this project, I preprocess data by using :

Resizing and Normalization: I resized each image to the same size so they would all fit into the model correctly. I also adjusted the pixel values to make sure they worked well with the model.

Augmentation: To help the model learn better and avoid getting overfitting, I used data augmentation. By creating slightly different versions of the images by rotating them, flipping them, or zooming in and out. specially for categories like (cable and nut .) These small changes made the model more adaptable and able to handle new data.

Model Selection

After reviewing various models for classifying images task, I chose a Convolutional Neural Network (CNN), which demonstrated excellent results in visual recognition tasks. CNNs are especially proficient at image classification since they can detect spatial features and patterns in images, which makes them ideal for recognizing defects.

Model Training

The model was trained on the prepared dataset using the following configuration to ensure effective learning and optimization :

- **Epochs:** 10 for each category
- **Learning Rate:** $1e-6$
- **Optimizer :** Adam with binary cross-entropy loss and accuracy as a metric

The dataset experienced a considerable class imbalance between "Non-Defective" and "Defective" items, resulting in the model favoring the majority class. This bias caused the model to predict solely the "Non-Defective" class, as indicated by the confusion matrix, leading to zero precision for the "Defective" class and inadequate recall, which diminished the model's effectiveness in detecting defective items.

To tackle this issue, methods such as data augmentation, oversampling, or implementing class weights could enhance predictions for the "Defective" class. Furthermore, the model exhibited indications of overfitting, likely due to limited variation in the training data and a small batch size, resulting in erratic updates and poor generalization. Increasing the batch size, incorporating dropout, or utilizing techniques like early stopping and regularization could help reduce overfitting and improve the model's performance.

Evaluation

The model's performance was assessed using the test set, with the following results:

- **Accuracy:** 0.66

accuracy			0.66	591
macro avg	0.33	0.50	0.40	591
weighted avg	0.43	0.66	0.52	591

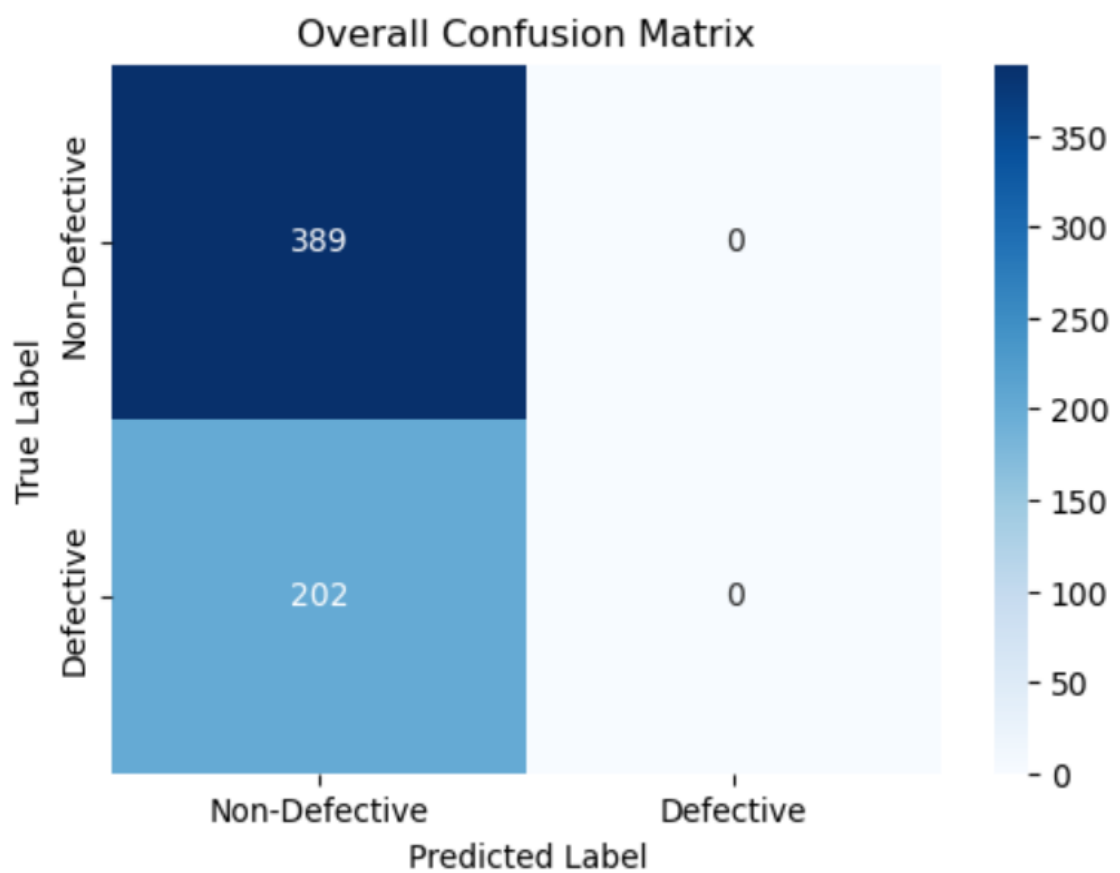


Fig 1 : confusion Matrix For Transistor

Classification Metrics :

- **Accuracy** measures the percentage of correctly classified images among all predictions.

- **Precision** reflects the percentage of correctly identified defective images among all images labeled as defective.
- **Recall** indicates the percentage of actual defective images that were correctly identified.
- **F1-Score** : provides a balanced measure of precision and recall, offering insight into overall model robustness.

Confusion Matrix

The confusion matrix showed the following breakdown:

- **True Positives (TP)**: Correctly identified defective images
- **True Negatives (TN)**: Correctly identified non-defective images
- **False Positives (FP)**: Non-defective images incorrectly classified as defective
- **False Negatives (FN)**: Defective images incorrectly classified as non-defective

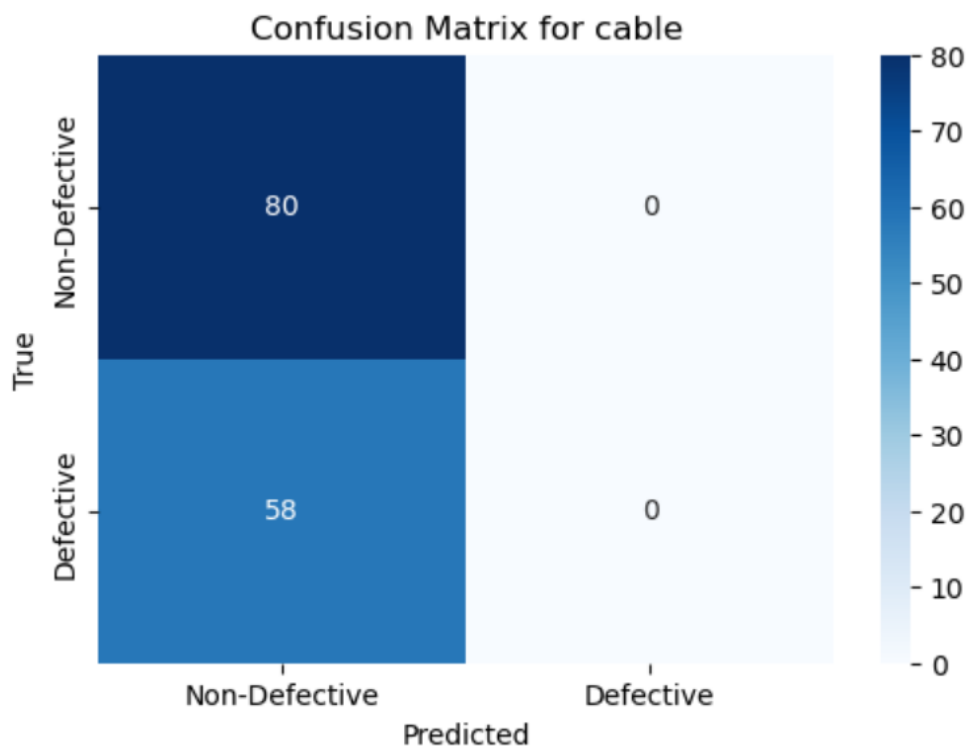


Fig 2 : confusion Matrix For Cable

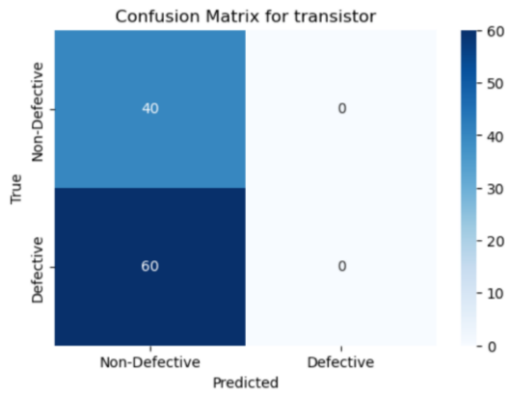


Fig 3 : confusion Matrix For Transistor

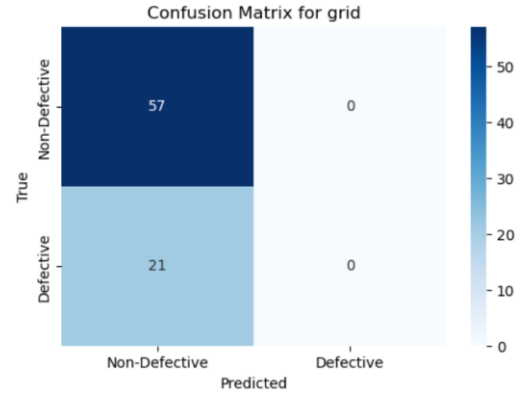


Fig 3 : confusion Matrix For Grid

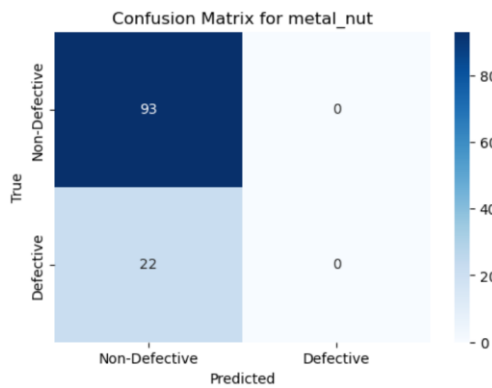


Fig 3 : confusion Matrix For Metal_nut

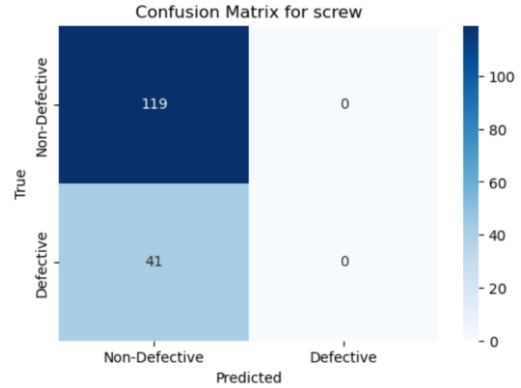


Fig 3 : confusion Matrix For Screw

Insights and Observations

Through this project, several key insights were gained:

- **Augmentation Importance:** Image augmentation enhanced model performance by reducing overfitting.
- **Defect Types:** Certain types of defects were more challenging to classify, suggesting that more training data or specialized models might be needed for these cases.
- **Class Imbalance:** If defective samples were limited, the model showed reduced recall. Addressing this with techniques like oversampling or loss function adjustments could improve results.

Conclusion

This project effectively showed a machine learning method for detecting defects in industrial settings, successfully distinguishing between images of defective and non-defective equipment. Automating the defect detection process allows industries to enhance efficiency and maintain consistency in their quality control efforts. Nevertheless, the model's effectiveness could be improved with more labeled data and optimized preprocessing methods, especially for addressing particular defects.

Future Scope

Further improvements to this project could include:

1. **Enhanced Dataset:** Gathering a broader and more diverse set of data to enhance model generalization.
2. **Specialized Models for Defect Types:** Developing models specific to each defect category to improve detection precision.
3. **Integration into Production:** Putting the model into a real-time inspection system to evaluate scalability and operational efficiency.