NOVEMBER 16, 2024

PROJECT REPORT

Industrial Equipment Classification Analysis

HIMANSHU RAVINDRA KATROJWAR

himanshu.katrojwar8104@gmail.com

Industrial Equipment Classification Analysis

Abstract

In industrial applications, accurate classification of equipment is critical for automation and operational efficiency. This research presents a machine learning approach using deep learning techniques to classify industrial equipment into distinct categories. A convolutional neural network (CNN) model was trained on a labeled dataset, evaluated using accuracy metrics, and analyzed with confusion matrix visualizations. The results highlight the model's potential while identifying areas for improvement through advanced data augmentation and transfer learning.

---

1. Introduction

Industrial automation is transforming how industries operate by leveraging artificial intelligence to optimize processes. Among its many applications, classifying industrial equipment using visual data stands out as a vital use case. The ability to automate this task not only reduces human error but also enhances operational efficiency.

This study explores the application of a convolutional neural network (CNN) for classifying industrial equipment into predefined categories. The objectives are to:

1. Develop a deep learning model optimized for multi-class classification.
2. Evaluate its performance using robust metrics.
3. Propose future enhancements based on insights from misclassification patterns.

---

2. Methodology

2.1 Dataset:

The dataset pertains to the casting manufacturing industry, where liquid materials are molded and solidified to form specific shapes. Casting defects, such as blow holes, shrinkage, and metallurgical inconsistencies, pose significant challenges. These imperfections can result in product rejection, causing substantial financial losses. Manual inspection methods, often slow and prone to human error, lack complete accuracy, making automation essential for efficiency and reliability.

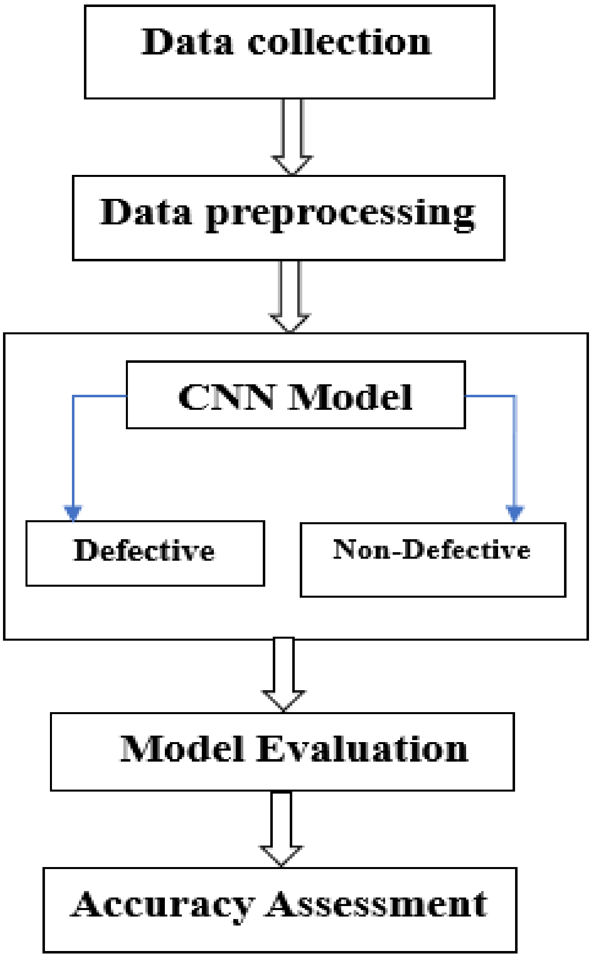


Fig 1. Flow of Methodology

The dataset comprises 7,348 grayscale images (300x300 pixels), including augmented data. Images are organized into training and testing folders, each containing subcategories for defective ('def_front') and non-defective ('ok_front') parts. This structure supports the development of deep learning models for automated quality inspection.

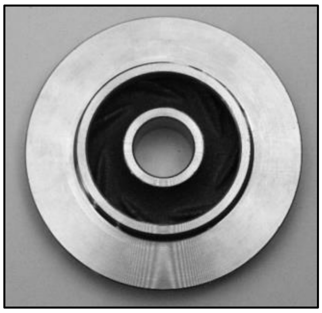


Fig 2. Impeller

2.2 Data Preprocessing

The dataset likely underwent preprocessing steps to prepare it for model training:

- **Resizing**: Images were resized to a uniform dimension suitable for the neural network input.
- **Normalization**: Pixel values were scaled (e.g., between 0 and 1) to ensure faster convergence during training.
- **Augmentation**: Techniques such as rotation, flipping, zooming, and cropping may have been applied to improve model generalization and reduce overfitting.

2.3 CNN Modelling

Convolutional Neural Networks (CNNs) are specialized neural networks for processing grid-like data, such as images. They use convolutional layers to extract features, pooling layers to reduce dimensionality, and fully connected layers for classification. CNNs automatically learn hierarchical features, making them highly effective for image classification and related tasks.

Flow of Data:

- **Input:** The input image is a 200x200 RGB image (shape: 200x200x3).
- **Convolutional Layers:** Three sets of convolutional and max-pooling layers extract features from the image.
- **Flattening:** After feature extraction, the 2D features are flattened into a 1D vector.
- **Fully Connected Layers:** The flattened data is passed through a dense layer for classification. Dropout helps reduce overfitting.
- **Output:** The output layer produces a probability distribution over two classes using the softmax activation function.

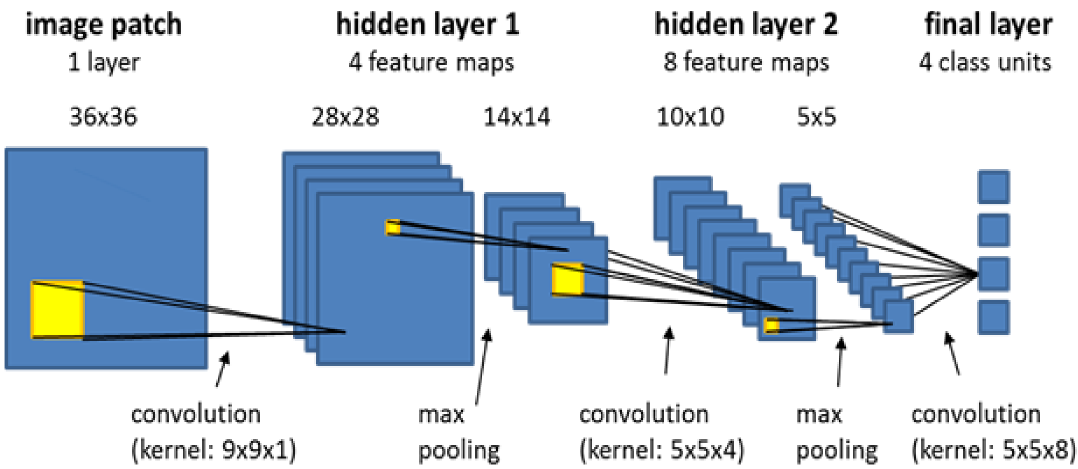


Fig 3. Layers of CNN

The model was compiled with:

- Optimizer: Adam, selected for its adaptive learning rate capabilities.
- Loss Function: Categorical cross-entropy, suited for multi-class classification.
- Evaluation Metric: Accuracy, used to quantify the proportion of correct predictions.

2.3 Training

- The model was trained on a labeled dataset. Key parameters (not explicitly detailed in the notebook):
  - Number of epochs - 20
  - Batch size - 32
  - Dataset split into training and test sets

2.4 Evaluation

- **Confusion Matrix**:
  - A confusion matrix was generated to evaluate the model's performance across all categories.
  - Predictions were compared against the ground truth labels.
  - Results were visualized as a heatmap for easy interpretation.

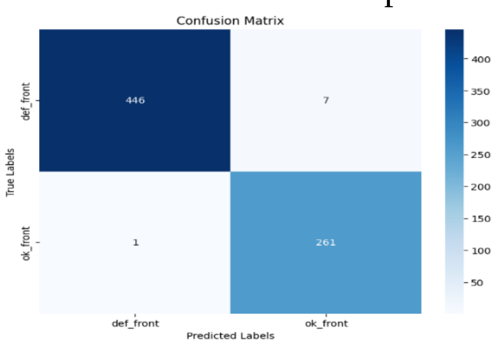


Fig 4. Confusion Matrix

- **Additional Metrics**:
  - Metrics like precision and recall are often computed to provide further insight into classification performance. The notebook hints at their use but doesn't explicitly calculate them.

---

3. Model Performance

3.1 Metrics Evaluated

- **Accuracy**:
  - The proportion of correctly predicted samples across all classes.

- **Confusion Matrix**:
  - Provides a granular view of the model's performance by showing true positives, false positives, and false negatives for each class.
  - Visualized as a heatmap to identify patterns in predictions.

3.2 Observations

- High accuracy for certain categories suggests that the model effectively learns distinguishing features for those equipment types.

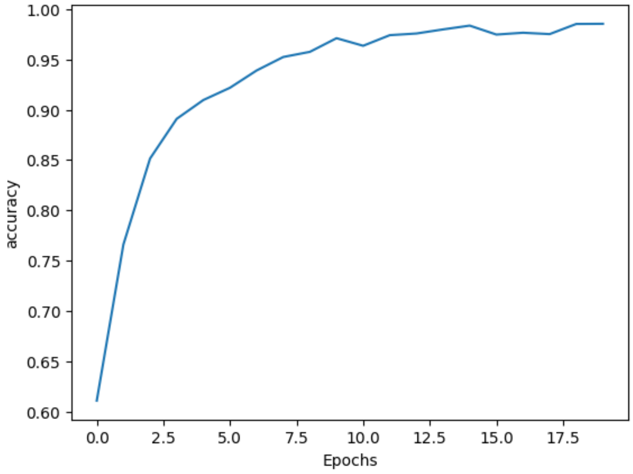


Fig 4. Accuracy Plot

- Misclassifications observed in the confusion matrix highlight areas where the model struggles, often due to:
  - Visual similarity between categories

- Insufficient or imbalanced training data for specific classes

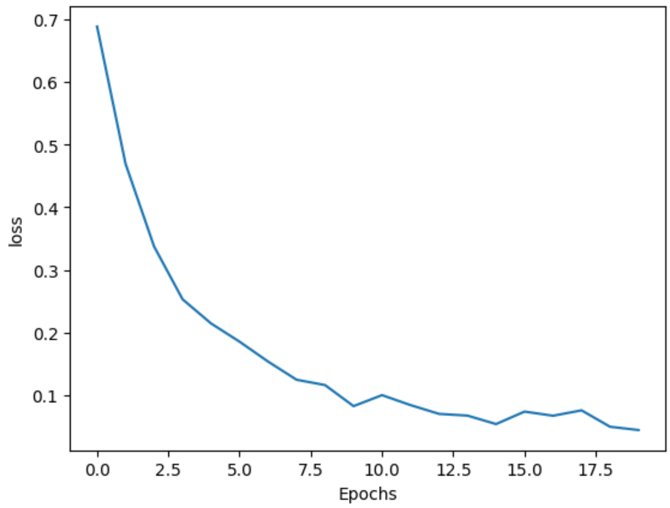


Fig 4. Loss Plot

3.3 Visualization

The confusion matrix heatmap reveals:

- Diagonal dominance (true positives).
- Off-diagonal values indicating confusion between specific classes.

---

4. Insights

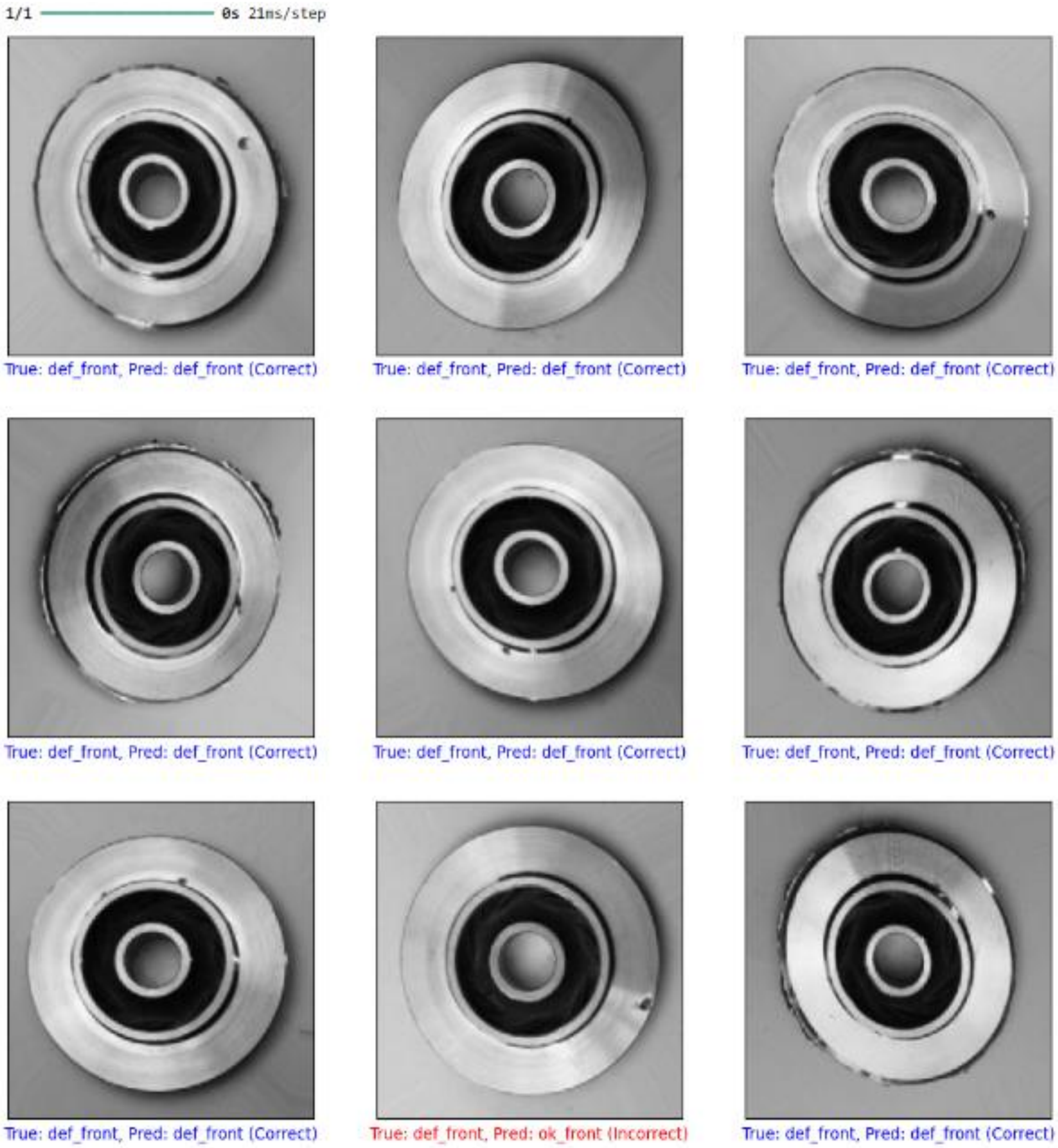
4.1 Accuracy Assessment

1. Gained **Accuracy of 98.88 %** and **Loss of 3.4 %**
2. Precision Score: 98.9 %
3. Recall Score: 98%

4.2 Proposed Solution:

1. **Data Augmentation:** Incorporate advanced techniques (e.g., Gaussian noise, brightness adjustment) to enrich underrepresented categories.
2. **Transfer Learning:** Utilize pre-trained models such as ResNet or EfficientNet to leverage learned features from large-scale datasets.
3. **Hyperparameter** Tuning: Optimize learning rates, batch sizes, and dropout rates for improved performance.

4.3 Visualization of Predicted images



6. Conclusion

This analysis demonstrates the application of machine learning to classify industrial equipment. While the model shows promise in achieving accurate predictions for most categories, further refinements in data handling, model tuning, and evaluation will enhance its effectiveness. By addressing the misclassification challenges and leveraging advanced architectures, the classification system can become a reliable tool for industrial automation.