CNN-Based Defect Detection in Manufacturing Images

# Progress Report

**Group Member:** Evan Musick

## Introduction

Automated defect detection in manufacturing is essential for quality control and reducing production costs. Traditional methods require extensive human intervention, which can be time-consuming, subjective, and prone to errors. This project implements a Convolutional Neural Network (CNN) approach to classify and detect defects in steel surface images. The NEU-DET dataset is being used, which contains images of six different types of steel surface defects: crazing, inclusion, patches, pitted surface, rolled-in scale, and scratches. By leveraging deep learning techniques, specifically CNNs, we aim to develop a model that can accurately identify these defects with minimal human intervention.

## Approach

### Model Architecture

The project utilizes a transfer learning approach with a pre-trained ResNet18 model as the backbone. ResNet18 was chosen for its proven effectiveness in image classification tasks while maintaining computational efficiency. The architecture consists of:

1. **Base Network**: Pre-trained ResNet18 model with weights from ImageNet

2. **Modifications**:

* The final fully connected layer has been replaced to output 6 classes (corresponding to the 6 defect types)
* The input expects RGB images of size 224×224 pixels

3. **Implementation Details**:

* The model preserves the feature extraction capabilities of ResNet18
* Only the final classification layer is fully retrained, while earlier layers maintain their pre-trained weights

### Dataset

The NEU-DET (Northeastern University Defect Detection) dataset is being used for this project. This dataset contains:

1. **Content**: Steel surface defect images with six different defect types

2. **Structure**:

* Training set: Located in NEU-DET/train/images
* Validation set: Located in NEU-DET/validation/images
* Each image is accompanied by XML annotation files

### Preprocessing and Data Augmentation

Several preprocessing and augmentation techniques are applied to improve model performance:

1. **Image Resizing**: All images are resized to 224×224 pixels to match the input requirements of the ResNet18 model

2. **Data Augmentation** (applied only to training data):

* Random horizontal flipping
* Random rotation (up to 10 degrees)

3. **Normalization**: Images are normalized using ImageNet mean and standard deviation values (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])

## Evaluation Plan

### Dataset Splitting

The dataset is organized as follows:

1. **Training Set**: Used for model training

2. **Validation Set**: Split into:

* Validation subset: Used for hyperparameter tuning and early stopping
* Test subset: Used for final model evaluation

3. **Split Ratio**: The validation set is further split with a configurable ratio (default 0.5), meaning 50% for validation and 50% for testing

### Evaluation Metrics

The model will be evaluated using the following metrics:

1. **Accuracy**: Overall classification accuracy

2. **Classification Report**: Precision, recall, and F1-score for each defect class

3. **Loss Curves**: Training and validation loss over epochs to analyze convergence and potential overfitting

### Hyperparameters

The following hyperparameters are being tuned:

1. **Learning Rate**: Default is 0.001, with experiments planned for 0.0005 and 0.0001

2. **Batch Size**: Default is 32, with experiments for 16 and 64

3. **Number of Epochs**: Default is 15, with early stopping based on validation accuracy

4. **Optimizer**: Adam optimizer is currently used, with potential experiments using SGD

### Comparison with Traditional Methods

For comprehensive evaluation, the CNN approach is being compared with traditional machine learning methods:

1. **Feature Extraction**:

* HOG (Histogram of Oriented Gradients)
* LBP (Local Binary Patterns)

2. **Classifiers**:

* SVM (Support Vector Machine)
* Random Forest

This comparison will provide insights into the advantages of deep learning over traditional approaches for this specific defect detection task.

## Current Progress

The implementation of the CNN model is complete, including the data loading pipeline, model architecture, training loop, and evaluation scripts. Initial training runs have been performed, and the model is being fine-tuned based on validation performance. The traditional machine learning comparison is also implemented and ready for comparative analysis.

## Next Steps

1. Complete hyperparameter tuning to optimize model performance
2. Perform comprehensive evaluation on the test set
3. Analyze results and compare with traditional methods
4. Document findings and prepare final report and presentation