**Image Classification for Basic Defect Detection**

**Author:**  
**Syed Adil Shah**

**Start Date:** 02-06-2025  
**End Date:** 04-06-2025

**Date of Submission: 04-06-2025**  
**Total Duration:** 3 days

### ****1. Objective****

The objective of this project was to build a binary image classification model to distinguish between **defective** and **non-defective** manufactured parts using computer vision and deep learning techniques. This task involved applying Convolutional Neural Networks (CNNs) and understanding fundamental concepts like data preprocessing, model training, evaluation, and augmentation.

### ****2. Dataset****

For this project, a subset of the **CIFAR-10** dataset was used. Only two classes were selected to simulate a binary classification task:

Class 0: Labeled as **Non-defective**

Class 1: Labeled as **Defective**

The dataset was divided into:

Training set: 10,000 images (5,000 per class)

Testing set: 2,000 images (1,000 per class)

### ****3. Methodology****

#### ****Data Preprocessing****

Loaded raw CIFAR-10 data manually using Python’s pickle module.

Extracted only two relevant classes for the binary classification.

Normalized image pixel values to the range [0, 1].

#### ****Model Architecture****

**Image Classification for Basic Defect Detection**

A custom Convolutional Neural Network (CNN) was designed using Keras:

3 convolutional layers with ReLU activations

MaxPooling for dimensionality reduction

Dropout layers for regularization

Dense layers for classification

#### ****Training****

Categorical Crossentropy was used as the loss function.

Optimizer: Adam

Batch size: 64

Epochs: 10

#### ****Data Augmentation****

Implemented using ImageDataGenerator to:

Introduce variability in training data

Prevent overfitting

Simulate real-world variations (flips, rotations, zoom)

### ****4. Evaluation & Results****

The model was evaluated using:

**Accuracy**: ~93% on test data

**Precision & Recall**: Good balance between precision and recall

**Confusion Matrix**: Clear separation between defective and non-defective classes

The results were satisfactory for a basic defect detection use case.

### ****5. Conclusion****

This project successfully demonstrated the workflow of a deep learning-based image classification task. The CNN model achieved reliable accuracy within a short time frame (3 days), and the use of augmentation improved generalization. The project met the learning goals of understanding image data handling, model design, training, and performance evaluation.

### ****6. Learnings****

Gained hands-on experience with manual data loading using pickle.

Learned how to use TensorFlow/Keras for defining and training CNNs.

Practiced using ImageDataGenerator for on-the-fly data augmentation.

Learned how to evaluate models using classification metrics and visualize results.