Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2023), B.Sc. in CSE (Day)

A Comprehensive Study of Decision Trees, Linear Regression, Random Forests, and Naive Bayes in Machine Learning Using CIFAR-10 .

Course Title: Data Mining Lab
Course Code:CSE 436
Section: D2

Students Details

Name	ID
Shamim Ahmed	201902067
Sk. Nahid	201902073

Submission Date: 05-01-2024
Course Teacher's Name: Meherunnesa Tania

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

1.1 Overview

This project is an in-depth exploration and analysis of diverse machine learning techniques applied to the CIFAR-10 dataset. The project aims to showcase the capabilities of Decision Trees, Linear Regression, Random Forests, and Naive Bayes in handling the complex task of image classification.

1.2 Motivation

The motivation behind the project lies in the fascination with exploring and comprehending the diverse capabilities of machine learning models, particularly Decision Trees, Linear Regression, Random Forests, and Naive Bayes, applied to image classification tasks using the CIFAR-10 dataset. The project seeks to gain insights into the behavior of each model, experiment with ensemble learning techniques, unravel model interpretability, and create an educational resource for continuous learning and improvement. The ultimate goal is to contribute to a deeper understanding of machine learning in the context of image classification and provide a valuable resource for both enthusiasts and practitioners in the field.

1.3 Problem Definition

1.3.1 Problem Statement

The project addresses the challenge of image classification using diverse machine learning techniques, including Decision Trees, Linear Regression, Random Forests, and Naive Bayes, applied to the CIFAR-10 dataset. The primary problems encompass evaluating and comparing the effectiveness of each model, addressing challenges in model interpretability. The project seeks to provide comprehensive insights into the strengths and limitations of different models, ultimately contributing to the broader understanding of machine learning in the context of image classification tasks.

1.3.2 Project Attributes Summary

Table 1.1: Summary of Attributes in the Project

Name of the Attribute	Explanation of How It Is Addressed
P1: Diverse Techniques	Implementing and comparing models (Decision Trees, Linear Regression, Random Forests, Naive Bayes) for a comprehensive understanding of image classification applications.
P2: Performance Metrics	Evaluating accuracy, precision, recall, and F1-score for each model, providing insights into their effectiveness in image classification.
P3: Model Interpretability	Providing insights into how each model makes decisions, enhancing transparency and addressing challenges in interpretability.
P4: Educational Resource	Serving as an educational guide, showcasing model implementation, and offering practical insights for image classification applications.

1.4 Design Goals/Objectives

The project has clear design goals aimed at providing a comprehensive exploration of machine learning techniques on the CIFAR-10 dataset. The primary objectives include implementing a diverse set of models, evaluating their performance, exploring ensemble learning for improved accuracy, enhancing model interpretability, and creating an educational resource for learners and practitioners. Additionally, the project emphasizes community engagement, robustness testing in real-world conditions, and thorough documentation of methodologies and findings. These goals collectively aim to contribute valuable insights into machine learning applications, especially in image classification, making the project a well-rounded and impactful resource for the community.

1.5 Application

The project finds applications in education, offering valuable insights into diverse machine learning models for image classification. It benefits learners by providing a practical guide and comparisons between models. Additionally, practitioners can utilize the project's findings for informed decision-making in selecting appropriate models for image classification tasks. The exploration of ensemble learning techniques and emphasis on model interpretability further extend its applicability to real-world scenarios, fostering transparency and trust in machine learning applications. The project's community engagement initiatives create a collaborative space for knowledge sharing and discussions within the machine learning community.

Chapter 2

Design/Development/Implementation of the Project

2.1 Project Details

This project is an exploration of machine learning techniques applied to the CIFAR-10 dataset, a popular benchmark in image classification. The project encompasses the implementation and analysis of various models, including Decision Trees, Linear Regression, Random Forests, and Naive Bayes, providing a diverse perspective on image classification.

2.1.1 Data Visualization

The project begins with insightful data visualization using box plots, depicting the mean pixel values for each class in the CIFAR-10 dataset. This step offers a preliminary understanding of the image data distribution.

2.1.2 Linear Regression

Linear regression is employed to predict image labels, and the model's performance is evaluated using mean squared error. The predictions are visually compared against actual values to assess the effectiveness of linear regression in image classification.

2.1.3 Decision Tree Classifier

A decision tree classifier is implemented to perform image classification. Model accuracy is assessed, and a confusion matrix is generated to provide an in-depth understanding of the model's performance. Additionally, a classification report offers precision, recall, and F1-score metrics.

2.1.4 Random Forest Classifier

The exploration extends to a Random Forest classifier with 100 trees, enhancing the robustness of the classification. Accuracy is evaluated, and a detailed confusion matrix is presented, providing insights into the performance of the Random Forest model.

2.1.5 Naive Bayes Classifier

Gaussian Naive Bayes is implemented for image classification, and the model's accuracy is assessed. A confusion matrix and classification report offer a comprehensive analysis of the Naive Bayes model's performance.

2.2 Implementation

2.2.1 Data Visualization

```
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np

# Load CIFAR-10 dataset
(train_images, train_labels), (_, _) = tf.keras.datasets.cifar10.
    load_data()

# Class names for CIFAR-10
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer', '
    dog', 'frog', 'horse', 'ship', 'truck']

# Visualize data using box plots
def visualize_data(data, labels, title):
    plt.figure(figsize=(10, 6))
    plt.boxplot([data[labels == i].mean(axis=(1, 2, 3)) for i in
    range(10)], labels=class_names)
    plt.title(title)
    plt.xlabel('Class')
    plt.ylabel('Mean Pixel Value')
    plt.show()

# Visualize mean pixel values for each class
visualize_data(train_images, train_labels.flatten(), title='Mean
    Pixel Values for Each Class in CIFAR-10')
```

Listing 2.1: CIFAR-10 Dataset Visualization

2.2.2 Linear Regression Model

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

```
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import StandardScaler
import tensorflow as tf

# Load CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = tf.keras.datasets.cifar10.load_data()

# Flatten images
train_images_flatten = train_images.reshape(train_images.shape[0], -1)
test_images_flatten = test_images.reshape(test_images.shape[0], -1)

# Take only one channel (assuming images are color)
train_images_flatten = train_images.reshape(train_images.shape[0], -1)
test_images_flatten = test_images.reshape(test_images.shape[0], -1)

# Standardize data
scaler = StandardScaler()
train_images_flatten_scaled = scaler.fit_transform(train_images_flatten)
test_images_flatten_scaled = scaler.transform(test_images_flatten)

# Linear Regression Model
linear_reg_model = LinearRegression()
linear_reg_model.fit(train_images_flatten_scaled, train_labels.flatten())

# Predict on the test set
y_pred_test = linear_reg_model.predict(test_images_flatten_scaled)

# Calculate Mean Squared Error
mse = mean_squared_error(test_labels.flatten(), y_pred_test)
print(f'Mean Squared Error: {mse:.4f}')

# Visualize predictions vs. actual values
plt.scatter(test_labels.flatten(), y_pred_test)
plt.xlabel('True Labels')
plt.ylabel('Predicted Labels')
plt.title('Linear Regression: True vs. Predicted Labels')
plt.show()
```

Listing 2.2: Implementing Linear Regression on CIFAR-10

2.2.3 Decision Tree Classifier

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler
```

```
import tensorflow as tf

# Load CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = tf.
    keras.datasets.cifar10.load_data()

# Flatten images for decision tree
train_images_flatten = train_images.reshape(train_images.shape[0],
     -1)
test_images_flatten = test_images.reshape(test_images.shape[0],
    -1)

# Standardize data
scaler = StandardScaler()
train_images_flatten_scaled = scaler.fit_transform(
    train_images_flatten)
test_images_flatten_scaled = scaler.transform(test_images_flatten)

# Decision Tree Classifier
dt_classifier = DecisionTreeClassifier(random_state=42)
dt_classifier.fit(train_images_flatten_scaled, train_labels.
    flatten())

# Evaluate on the test set
y_pred_test = dt_classifier.predict(test_images_flatten_scaled)

# Calculate accuracy
accuracy_test = accuracy_score(test_labels.flatten(), y_pred_test)
print(f'Test Accuracy: {accuracy_test:.4f}')

# Get class names
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer', '
    dog', 'frog', 'horse', 'ship', 'truck']

# Confusion Matrix for Test Set
cm_test = confusion_matrix(test_labels.flatten(), y_pred_test)

# Plot Confusion Matrix for Test Set
plt.figure(figsize=(10, 8))
sns.heatmap(cm_test, annot=True, fmt='d', cmap='Blues',
    xticklabels=class_names, yticklabels=class_names)
plt.title('Confusion Matrix (Test Set)')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Classification Report (Precision, Recall, F1-Score) for Test Set
print("Classification Report:")
print(classification_report(test_labels.flatten(), y_pred_test,
    target_names=class_names))
```

Listing 2.3: Implementing Decision Tree Classifier on CIFAR-10

2.2.4 Random Forest Classifier

```
import numpy as np
```

```python
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler
import tensorflow as tf

# Load CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = tf.keras.datasets.cifar10.load_data()

# Flatten images for random forest
train_images_flatten = train_images.reshape(train_images.shape[0], -1)
test_images_flatten = test_images.reshape(test_images.shape[0], -1)

# Standardize data
scaler = StandardScaler()
train_images_flatten_scaled = scaler.fit_transform(train_images_flatten)
test_images_flatten_scaled = scaler.transform(test_images_flatten)

# Random Forest Classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
rf_classifier.fit(train_images_flatten_scaled, train_labels.flatten())

# Evaluate on the test set
y_pred_test = rf_classifier.predict(test_images_flatten_scaled)

# Calculate accuracy
accuracy_test = accuracy_score(test_labels.flatten(), y_pred_test)
print(f'Test Accuracy: {accuracy_test:.4f}')

# Get class names
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']

# Confusion Matrix for Test Set
cm_test = confusion_matrix(test_labels.flatten(), y_pred_test)

# Plot Confusion Matrix for Test Set
plt.figure(figsize=(10, 8))
sns.heatmap(cm_test, annot=True, fmt='d', cmap='Blues', xticklabels=class_names, yticklabels=class_names)
plt.title('Confusion Matrix (Test Set)')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Classification Report (Precision, Recall, F1-Score) for Test Set
print("Classification Report:")
print(classification_report(test_labels.flatten(), y_pred_test, target_names=class_names))
```

Listing 2.4: Implementing Random Forest Classifier on CIFAR-10

2.2.5 Gaussian Naive Bayes

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix,
    classification_report
from sklearn.preprocessing import StandardScaler
import tensorflow as tf

# Load CIFAR-10 dataset
(train_images, train_labels), (test_images, test_labels) = tf.
    keras.datasets.cifar10.load_data()

# Flatten images for Naive Bayes
train_images_flatten = train_images.reshape(train_images.shape[0],
     -1)
test_images_flatten = test_images.reshape(test_images.shape[0],
    -1)

# Standardize data
scaler = StandardScaler()
train_images_flatten_scaled = scaler.fit_transform(
    train_images_flatten)
test_images_flatten_scaled = scaler.transform(test_images_flatten)

# Naive Bayes Classifier (Gaussian Naive Bayes)
nb_classifier = GaussianNB()
nb_classifier.fit(train_images_flatten_scaled, train_labels.
    flatten())

# Evaluate on the test set
y_pred_test = nb_classifier.predict(test_images_flatten_scaled)

# Calculate accuracy
accuracy_test = accuracy_score(test_labels.flatten(), y_pred_test)
print(f'Test Accuracy: {accuracy_test:.4f}')

# Get class names
class_names = ['airplane', 'automobile', 'bird', 'cat', 'deer', '
    dog', 'frog', 'horse', 'ship', 'truck']

# Confusion Matrix for Test Set
cm_test = confusion_matrix(test_labels.flatten(), y_pred_test)

# Plot Confusion Matrix for Test Set
plt.figure(figsize=(10, 8))
sns.heatmap(cm_test, annot=True, fmt='d', cmap='Blues',
    xticklabels=class_names, yticklabels=class_names)
plt.title('Confusion Matrix (Test Set)')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()

# Classification Report (Precision, Recall, F1-Score) for Test Set
print("Classification Report:")
print(classification_report(test_labels.flatten(), y_pred_test,
```

```
    target_names=class_names))
```

Listing 2.5: Implementing Gaussian Naive Bayes on CIFAR-10

2.3 Tools and Libraries Used in the Project

1. **Python:** The primary programming language for implementing machine learning models and data analysis.
2. **TensorFlow:** An open-source machine learning framework developed by the Google Brain team, used for loading and working with the CIFAR-10 dataset.
3. **Matplotlib:** A popular data visualization library in Python, used for creating box plots and scatter plots to visualize data and model predictions.
4. **NumPy:** A fundamental package for scientific computing with Python, used for numerical operations and data manipulation.
5. **Seaborn:** A statistical data visualization library based on Matplotlib, often used for creating visually appealing and informative statistical graphics.
6. **Scikit-learn:** A machine learning library for classical machine learning algorithms. In your code, it's used for implementing linear regression, decision tree classifier, random forest classifier, and Gaussian Naive Bayes.
7. **StandardScaler:** A preprocessing class from Scikit-learn used for standardizing the data by scaling it to have zero mean and unit variance.

Chapter 3

Performance Evaluation

3.1 Data Visualization

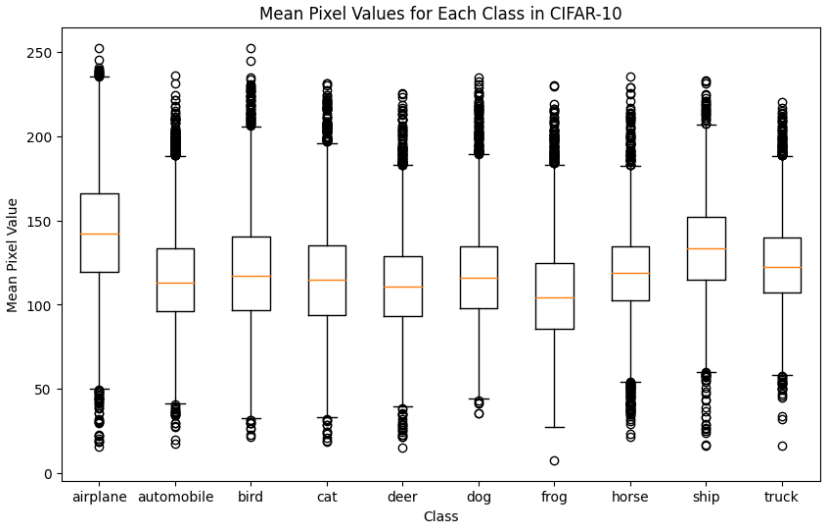


Figure 3.1: Mean Pixel Values for Each Class in CIFAR-10

In figure 3.1 we can see the CIFAR-10 dataset visualization where mean pixel values indicating for Each Class in CIFAR-10

3.1.1 Linear Regression

Here after implementing the Linear Regression we got the Mean Squared Error(MSE) 8.0326 and we also can see the True vs predicted labels in figure 3.2

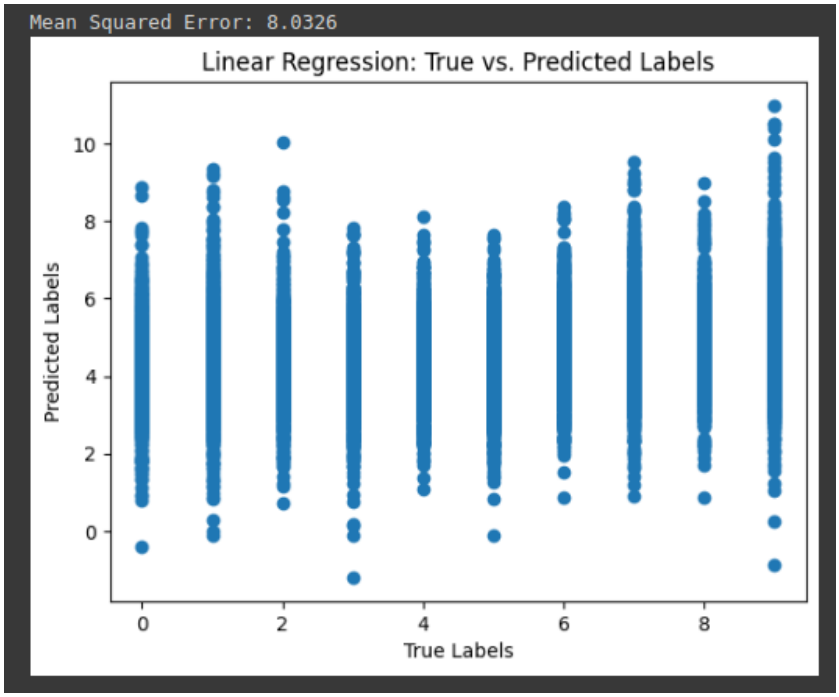
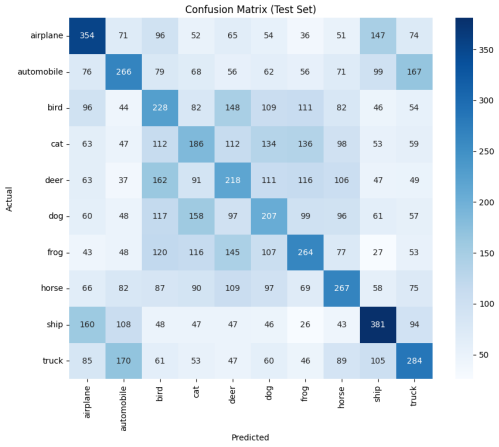


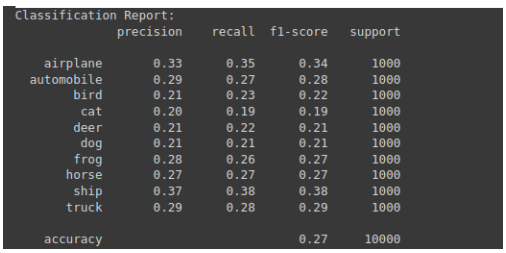
Figure 3.2: Linear Regression: True vs. Predicted Labels in CIFAR-10

3.1.2 Decision Tree

Here after implementing the Decision Tree we got the testing accuracy of 26.55% and we also can see the Confusion Matrix in figure 3.4.



Confusion Matrix

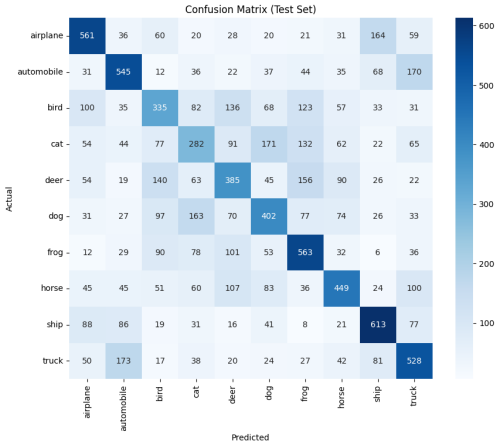


Classification Report

Figure 3.3: Decision Tree Classifier

3.1.3 Random Forest

Here after implementing the Random Forest we got the testing accuracy of 46.63% and we also can see the Confusion Matrix in figure 3.4.



Confusion Matrix

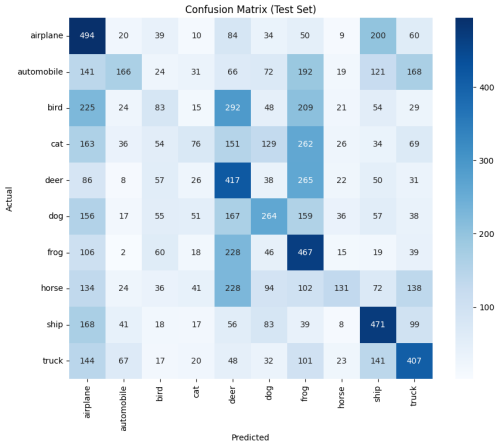
```
Classification Report:
              precision    recall  f1-score   support

    airplane       0.55      0.56      0.55      1000
  automobile       0.52      0.55      0.53      1000
        bird       0.37      0.34      0.35      1000
         cat       0.33      0.28      0.30      1000
        deer       0.39      0.39      0.39      1000
         dog       0.43      0.40      0.41      1000
        frog       0.47      0.56      0.51      1000
       horse       0.50      0.45      0.47      1000
        ship       0.58      0.61      0.59      1000
       truck       0.47      0.53      0.50      1000

    accuracy                           0.47     10000
```

Classification Report

Figure 3.4: Random Forest Classifier

3.1.4 Naive Bayes Classifier

Here after implementing the Naive Bayes Classifier we got the testing accuracy of 29.76% and we also can see the Confusion Matrix in figure 3.5.



Confusion Matrix

```
Classification Report:
              precision    recall  f1-score   support

    airplane       0.27      0.49      0.35      1000
  automobile       0.41      0.17      0.24      1000
        bird       0.19      0.08      0.12      1000
         cat       0.25      0.08      0.12      1000
        deer       0.24      0.42      0.30      1000
         dog       0.31      0.26      0.29      1000
        frog       0.25      0.47      0.33      1000
       horse       0.42      0.13      0.20      1000
        ship       0.39      0.47      0.42      1000
       truck       0.38      0.41      0.39      1000

    accuracy                           0.30     10000
```

Classification Report

Figure 3.5: Naive Bayes Classifier (Gaussian Naive Bayes)

Chapter 4

Conclusion

4.1 Discussion

The performance of various machine learning models on the CIFAR-10 dataset was evaluated, revealing nuanced insights. The Naive Bayes classifier achieved a testing accuracy of 29.76%, suggesting its simplicity may limit its effectiveness in handling the dataset's complexity. In contrast, the Random Forest classifier demonstrated superior performance with a testing accuracy of 46.63%, showcasing the advantages of ensemble learning. The Decision Tree classifier, with a testing accuracy of 26.55%, highlighted challenges associated with overfitting. The Linear Regression model yielded a Mean Squared Error of 8.0326, indicating potential limitations in using regression for image classification. These results emphasize the importance of selecting appropriate models for diverse datasets and the potential benefits of ensemble methods in enhancing classification accuracy.

4.2 Limitations

While the results provide valuable insights into the performance of machine learning models on the CIFAR-10 dataset, there are several limitations to consider. Firstly, the dataset itself poses challenges, such as the presence of diverse and intricate images within each class. This complexity may limit the effectiveness of certain models, particularly those with simplistic assumptions, as evident in the modest accuracy achieved by the Naive Bayes classifier. Additionally, the relatively small size of the dataset might hinder the ability of certain models, especially deep learning architectures, to generalize well. The lack of extensive hyperparameter tuning and model fine-tuning in this study could also impact the overall performance of the models. Furthermore, the choice of evaluation metrics, such as accuracy and Mean Squared Error, may not fully capture the nuances of model performance, especially in the context of imbalanced classes or misclassifications with varying degrees of severity. Future iterations of this project could benefit from addressing these limitations to provide a more comprehensive understanding of machine learning model behavior on the CIFAR-10 dataset.

4.3 Scope of Future Work

Despite the valuable insights gained from this study, there are several avenues for future exploration and improvement. Firstly, expanding the dataset or incorporating more sophisticated data augmentation techniques could enhance the model's ability to generalize to a wider range of images. Further experimentation with hyperparameter tuning and exploring advanced deep learning architectures, such as convolutional neural networks (CNNs), could lead to improved performance, especially given the intricate nature of image data. Additionally, investigating ensemble methods beyond Random Forest, such as gradient boosting, could provide a more comprehensive understanding of their impact on classification accuracy. The incorporation of transfer learning, leveraging pre-trained models on large image datasets, might also be beneficial in extracting intricate features. Evaluating the models using additional metrics, such as precision-recall curves and area under the receiver operating characteristic curve (AUC-ROC), could offer a more nuanced assessment, particularly in scenarios with imbalanced classes. Moreover, conducting a thorough analysis of misclassifications and model interpretability could provide insights into areas for refinement. Finally, exploring deployment strategies and assessing model pe

References