Mid-Sem Report of PRML- PROJECT

**Object-recognition model**

TEAM MEMBER’s NAME ROHAN REGAR (B22CS072)

ARNAVA SRIVASTAVA (B22AI009) BUDDHAV GARG (B22EE019) DHEERAJ DHAKER (B22EE023) HARSHIT BARWAR (B22EE030)

Github Repository link

[**RohanRegar/prml\_object\_recognition**](https://github.com/RohanRegar/prml_object_recognition)

# Objective

The objective of this project is to evaluate and compare machine learning (ML) techniques for object recognition tasks using the CIFAR-10 dataset. Object recognition in natural scenes has diverse applications, and this project aims to assess the efficacy of various ML approaches in this context. The project involves exploring and analyzing different ML techniques to identify their performance, strengths, and weaknesses in object recognition.

# Dataset

The CIFAR-10 dataset comprises of 10 classes: airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each class contains 6,000 images split into a training set of 50,000 images and a test set of 10,000 images. The images are low-resolution, making this dataset suitable for quick experimentation with machine learning models.

Each image comprises of 32X32 pixels in 3 Channels (R,G,B)

# Preprocessing Steps for CIFAR-10 Dataset

 A subset of images from the CIFAR-10 training set was visualized using a subplot, offering insight into the dataset's content and distribution, aiding in initial data exploration and understanding.

 Subsequently, the dataset underwent normalization by flattening the feature array and applying min-max normalization, converting pixel values from the original 0-255 range to a normalized range of 0.0-1.0. This standardized representation ensured consistent data formatting, facilitating improved model convergence and performance during training and evaluation. Both the training and test sets were subjected to this normalization process, establishing a uniform preprocessing pipeline for subsequent machine learning tasks.

# Early Results

We have imported the following four classifier and evaluation metrics from Scikit- Learn Library

**CLASSIFIER**

KNN

Decision Tree

Random Forest

Naive Bayes

**ACCURACY**

0.33

0.26

0.46

0.29

**PRECISION**

0.43

0.26

0.46

0.31

**RECALL**

0.33

0.26

0.46

0.29

**F1-SCORE**

0.32

0.26

0.46

0.27

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CLASSIFIER | ACCURACY | PRECISION | RECALL | F1-SCORE |
| KNN | 0.33 | 0.43 | 0.33 | 0.32 |
| Decision Tree | 0.26 | 0.26 | 0.26 | 0.26 |
| Random Forest | 0.46 | 0.46 | 0.46 | 0.46 |
| Naïve Bayes | 0.29 | 0.31 | 0.29 | 0.27 |

# Proposed Approaches:

 Convolutional Neural Networks (CNNs): CNNs have demonstrated state-of-the-art performance in image classification tasks. We will experiment with different hyperparameters, optimization techniques, and regularization methods to fine-tune the models.

Specifically, we will explore varying the number of layers, filter sizes, and activation functions in the convolutional and pooling layers.

**Enhancements to Classical Machine Learning Algorithms:**

1. **K-Nearest Neighbors (KNN): We have initially experimented with KNN with k=5, but we can further explore different values of k to find the optimal value for our dataset. Additionally, we can investigate different distance metrics such as Manhattan distance, Euclidean distance, and cosine similarity to measure the similarity between data points.**
2. **Decision Trees: For decision trees, we can explore different splitting criteria such as Gini impurity and entropy to construct more informative trees. We can also experiment with tree pruning techniques to prevent overfitting and improve generalization. Additionally, we can adjust parameters like the maximum depth of the tree to optimize the performance of the decision tree classifier.**
3. **Random Forest: Random forests are an ensemble of decision trees, so similar enhancements as decision trees can be applied. Additionally, we can experiment with the number of trees in the forest and the maximum number of features considered for splitting at each node.**
4. **Naive Bayes: Although Naive Bayes is a simple probabilistic classifier, we can experiment with different distribution assumptions such as Gaussian, multinomial, and Bernoulli to handle different types of data distributions.**

We can also explore feature selection techniques to identify and select the most relevant features for classification, which can potentially improve the performance of the traditional classifiers.

By exploring these enhancements to classical machine learning algorithms, we aim to compare their performance with more advanced techniques like CNNs and identify the most suitable approach for object recognition on the CIFAR-10 dataset.