PRML Mid-Term Report
Object-recognition using CIFAR-10

Abhishek Yadav, Amol Gaur, Binay Suman, Omprakash Nain
Tanshina Mangliya, Shahil Sharma, Shivanshu Verma.

B22ES020, B22CS008, B22CS019, B22AI062
B22EE067, B22CS048, B22ES010.

Contents

Abstract

This mid-project report presents our progress in exploring traditional machine learning techniques for object recognition using the CIFAR-10 dataset. We outline the problem statement, introduce the dataset, and explore some traditional machine learning techniques. Our initial exploration involved exploring baseline models such as KNN, SVM, PCA, ANN, Decision Trees and analyze their performance systematically. Through this project, we aim to gain insights into the effectiveness of traditional ML techniques for real-world object recognition tasks.

1 Introduction

Object recognition in natural scenes is a fundamental problem in computer vision with extensive applications in fields such as autonomous driving, surveillance, and augmented reality. The objective of this project is to compare various machine learning techniques for object recognition using the CIFAR-10 dataset.

2 Problem Statement

The objective of this project is to explore traditional machine learning techniques for object recognition in natural scenes. Given the CIFAR-10 dataset, which consists of 60,000 32x32 color images in 10 classes, our goal is to develop models that can accurately classify these images into their respective categories. The emphasis is on rigorously exploring classical machine learning techniques while adhering to the provided guidelines.

3 Dataset

The CIFAR-10 dataset comprises images belonging to ten classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each image is 32x32 pixels with three color channels (RGB). The dataset is split into training and test sets, with 50,000 images in the training set and 10,000 images in the test set.

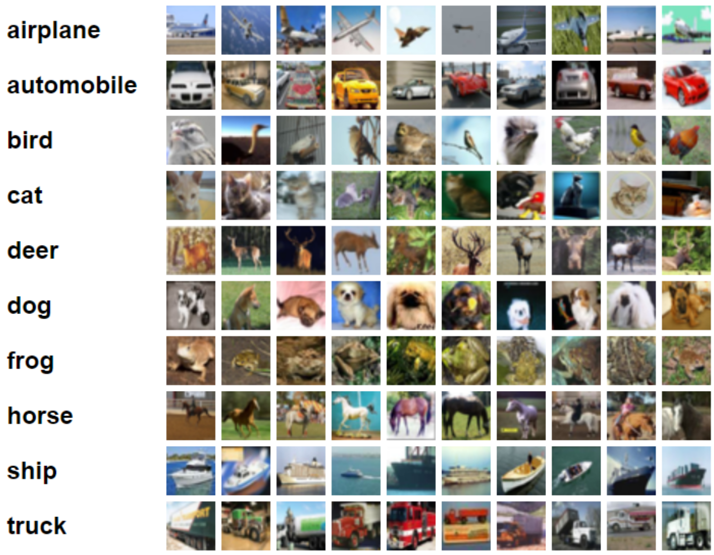

Figure 1: Example of CIFAR-10 Dataset.

4 Proposed Approaches

For this project, we plan to implement the few of following machine learning techniques:

4.1 K-Nearest Neighbors (KNN)

K-Nearest Neighbors is a simple and intuitive classification algorithm. It classifies objects based on the majority class among the k-nearest neighbors of a given data point in the feature space. In the context of object recognition, KNN calculates the distance between the test image and all images in the training set. It then selects the k nearest neighbors and assigns the class label based on the majority class among these neighbors.

Implementaion: To implement KNN for object classification using the CIFAR-10 dataset, we need to calculate the distance between images. Typically, the Euclidean distance metric is used. We then select the value of k, the number of neighbors to consider, and classify the test image based on the majority class among these neighbors.

Pros:

- Easy to understand and implement.
- No training phase, as the entire training dataset serves as the model.
- Can perform well with small datasets.

Cons:

- Slow at test time, especially with large datasets.
- Sensitive to irrelevant features or noisy data.
- Not suitable for high-dimensional data without dimensionality reduction.

4.2 Support Vector Machine (SVM):

Support Vector Machines is a powerful supervised learning algorithm used for classification tasks. It finds the hyperplane that best separates the classes in the feature space. In the context of object recognition, SVM aims to find the optimal decision boundary that maximizes the margin between different classes.

Implementation: To implement SVM for object classification, we need to preprocess the CIFAR-10 dataset, extract relevant features, and train the SVM model. Features can be extracted using techniques like Histogram of Oriented Gradients (HOG) or Convolutional Neural Networks (CNNs). SVM then learns the decision boundary between classes based on these features.

Pros:

- Effective in high-dimensional spaces.
- Versatile, as different kernel functions can be used for various decision boundaries.
- Memory efficient, as it uses a subset of training points for decision function (support vectors).

Cons:

- Computationally intensive, especially with large datasets.
- Requires careful selection of kernel and tuning of hyperparameters.
- Not suitable for multi-class classification without extensions like one-vs-all or one-vs-one.

4.3 Decision Trees

Decision Trees are versatile supervised learning models that recursively partition the feature space into regions. Each internal node represents a feature, each branch represents a decision based on that feature, and each leaf node represents a class label. In object recognition, decision trees make decisions based on the values of features extracted from images.

Implementation: To implement decision trees for object classification, we need to preprocess the CIFAR-10 dataset and extract relevant features. Decision trees are then trained using these features, and during testing, the test images are passed through the tree to reach a leaf node, which determines the class label.

Pros:

- Reduces dimensionality while preserving most of the variance.
- Speeds up learning algorithms and reduces overfitting.
- Removes correlated features, improving model performance.

Cons:

- Assumes linear relationships between variables.
- May not perform well if data is not Gaussian distributed.
- Does not consider class labels, which can result in loss of discriminative information.

4.4 Principal Component Analysis (PCA)

Principal Component Analysis is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional space while preserving most of the original variance. In the context of object recognition, PCA can be used to reduce the dimensionality of the image data, making it easier for subsequent classifiers to handle.

Implementation: To implement PCA for object classification, we first flatten the images in the CIFAR-10 dataset into feature vectors. We then perform PCA to reduce the dimensionality of these feature vectors while retaining most of the variance. The reduced-dimensional feature vectors are then fed into a classifier like SVM or ANN for object classification.

Pros:

- Easy to interpret and visualize.
- Robust to outliers and missing values (with appropriate handling).
- Automatically selects relevant features.

Cons:

- Prone to overfitting, especially with deep trees.
- Instability: small changes in the data can result in a different tree structure.
- Biased towards features with more levels.

4.5 Artificial Neural Networks (ANN)

Artificial Neural Networks are a class of machine learning models inspired by the structure and function of the human brain. ANN consists of interconnected neurons organized in layers, including an input layer, one or more hidden layers, and an output layer. In object recognition, ANN learns to map input images to their corresponding class labels through the process of training.

Implementation: To implement ANN for object classification, we preprocess the CIFAR-10 dataset and feed the flattened image pixels as input to the neural network. The network is then trained using backpropagation and gradient descent to minimize the classification error. During testing, the trained network predicts the class label of unseen images.

Pros:

- Ability to capture complex nonlinear relationships in data.
- Can learn hierarchical representations of features.
- Can be used for various tasks such as classification, regression, and even unsupervised learning.

Cons:

- Requires large amounts of data for training, especially for deep architectures.
- Computationally intensive, especially during training with large networks.
- Prone to overfitting, especially with limited training data and complex architectures.

5 References

- CIFAR-10 dataset: https://www.cs.toronto.edu/~kriz/cifar.html

****End of Report****