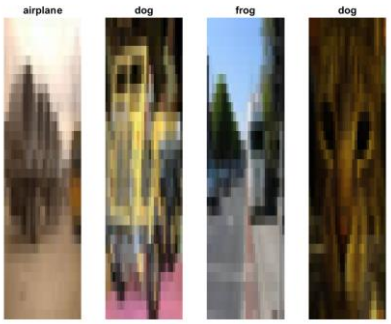
A study on image classification

Introduction

Image classification involves training models to automatically identify and categorise segments of an image. This coursework focuses on this aspect of machine learning, evaluating models using the CIFAR-10 dataset, an image dataset consisting of 60,000 colour images grouped into 10 classes, representing different objects such as vehicles. This project aims to use this data set and various image classification models such as decision trees and support vector machine multiclass (SVM multiclass) and K nearest neighbour (K-NN) using Euclidean distance and cosine distance, which shall be trained and then evaluated.

Data and Preparation



Each image in the dataset is 32x32x3 pixels. To prepare the data for training and testing, it is converted into a double ensuring compatibility with MATLAB's functions. The data variable represents the image set with its labels. To understand the images, a 4x1 subplot of randomly selected images is displayed here along with their corresponding labels. As the dataset is very large and training and testing this across all 4 of the selected models would take a considerable amount of time, a random set of 3 classes is selected (18000 images, 6000 per class). This set is then split 50/50 into a training and testing set. Images are then restructured into a 1x3072 vector, ensuring compatibility with the image classification models.

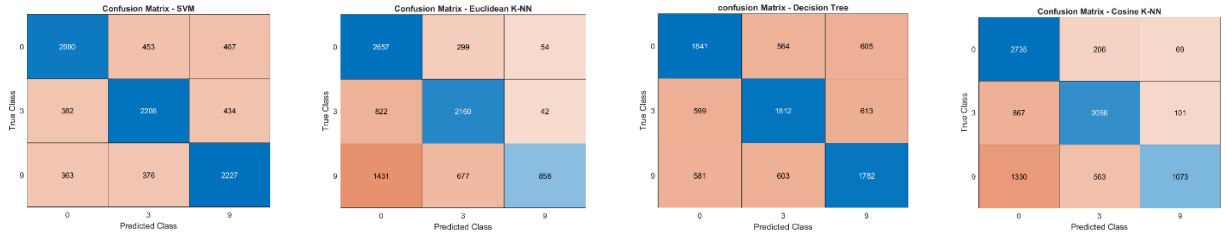
Methodology

This study uses K-NN Euclidean and cosine distance metrics (these models were built from scratch), SVM multiclass and decision tree. Each model is trained, tested and evaluated. K-NN predicts the class of an image by identifying majority class among its k closest neighbours in the training data. In this study we set k to 5. Euclidean distance measures the straight-line distance between image vectors. Cosine distance calculates similarity based on the angle between vectors. SVM is a one vs all strategy, suitable for multiclass classification. Decision trees recursively partition the feature space to separate attributes belonging to different classes. Both the decision tree and the SVM models were MATLAB premade functions. Each technique is evaluated using performance metrics including accuracy, time and a confusion matrix.

Results

The results in this study are held in the data files cw1.mat. SVM outperformed all other models in accuracy (72.5%) as well as the fastest total time of 3.9 seconds with decision tree coming in last (60.39%). The K-NN techniques tended to have exponentially greater execution times, with Euclidean distance coming in at 1108.88 seconds. SVM also had the most balanced confusion matrix, unlike decision tree's which showed significant miscalculations.

Classification technique	Accuracy (%)	Total Time (seconds)
K-NN (cosine Distance)	65.16	634.12
K-NN (Euclidean Distance)	63.06	1108.88
SVM Multiclass	72.50	3.90
Decision Trees	60.39	16.47



Conclusion

Based on the results across these models in this study, I would recommend SVM for this data set. It produces the most accurate and balanced classifications with the least miscalculations. SVM is designed to handle multiclass problems such as this one due to the dataset used. However, SVM does struggle with scalability, so a larger dataset could create a longer computation time.