Assignment 1 – Task 2

Multiclass Image Classification using

K-Nearest Neighbours (KNN)

1. Introduction

In this task, a **K-Nearest Neighbours (KNN)** classifier is implemented **from scratch** to perform **multiclass image classification** on the **CIFAR-10 dataset**.
Unlike Task-1, which involved binary classification on tabular data, Task-2 focuses on **high-dimensional image data** and involves **10 different classes**.

The objective of this task is to:

- Apply the KNN algorithm on image data,
- Experiment with different values of **K** and **distance metrics**,
- Identify the optimal hyperparameters based on test accuracy,
- Evaluate model performance using **confusion matrix, precision, and recall**,
- Analyze the challenges of using KNN for image classification.

2. Dataset Description

The **CIFAR-10 dataset** consists of **60,000 color images**, each of size **32 × 32 × 3**, belonging to **10 distinct classes**:

- Airplane
- Automobile
- Bird
- Cat
- Deer
- Dog
- Frog

- Horse
- Ship
- Truck

Each image is represented as a **3072-dimensional feature vector** (32 × 32 × 3).
The dataset is divided into:

- **50,000 training images**
- **10,000 test images**

Due to the high computational cost of KNN, a **subset of the dataset** was used for experimentation.

3. Data Preprocessing

3.1 Dataset Reduction

KNN requires computing the distance between a test sample and **every training sample**, which is computationally expensive for large datasets.
To ensure feasible execution time, a reduced subset of the dataset was selected:

- **Training samples:** 1000
- **Testing samples:** 200

This reduction does not change the algorithm's logic but improves computational efficiency.

3.2 Data Normalization

Each pixel value in CIFAR-10 ranges from **0 to 255**.
To ensure fair distance computation, all pixel values were normalized to the range **[0,1]** by dividing by 255.

Normalization is essential because KNN is a **distance-based algorithm**, and unnormalized features can bias distance calculations.

4. Methodology

4.1 K-Nearest Neighbors Algorithm

KNN is a **supervised, instance-based learning algorithm**.
For each test image:

1. Distance is computed between the test image and all training images.
2. The **K nearest neighbors** are selected.
3. The class label is assigned based on **majority voting** among the neighbors.

No explicit training phase is involved, making KNN a **lazy learning algorithm**.

4.2 Distance Metrics Used

The following distance metrics were implemented from scratch:

- **Euclidean Distance**
- **Manhattan Distance**
- **Minkowski Distance**
- **Cosine Similarity**
- **Hamming Distance**

Each metric defines similarity differently, which directly impacts classification accuracy.

4.3 Hyperparameters for Experimentation

The following hyperparameters were evaluated as specified in the assignment:

- **Values of K:** 3, 4, 9, 20, 47
- **Distance Metrics:** Euclidean, Manhattan, Minkowski, Cosine, Hamming

Each combination was evaluated using **testing accuracy**.

5. Experimental Results

5.1 Accuracy Evaluation

For each distance metric, accuracy was computed across all specified values of K.
A **K vs Accuracy** plot was generated to compare the performance of different distance metrics.

5.2 Best Model Selection

The optimal KNN model was selected based on **highest testing accuracy**.

- **Best K:** *(based on experimental results)*
- **Best Distance Metric:** *(based on experimental results)*
- **Highest Test Accuracy:** *(obtained value)*

6. Performance Evaluation

6.1 Confusion Matrix

Since CIFAR-10 is a **multiclass classification problem**, a **10 × 10 confusion matrix** was constructed.

- Rows represent **actual class labels**
- Columns represent **predicted class labels**

The confusion matrix provides insight into **class-wise misclassification patterns**.

6.2 Precision

Precision measures the correctness of predictions for each class:

$$\text{Precision} = \frac{TP}{TP + FP}$$

High precision indicates fewer false positive predictions for a class.

6.3 Recall

Recall measures how well the model identifies all instances of a given class:

$$\text{Recall} = \frac{TP}{TP + FN}$$

High recall indicates fewer missed detections for a class.

7. Observations and Inferences

7.1 Effect of Distance Metrics

- **Euclidean distance** performed relatively better due to its ability to capture pixel-wise similarity.
- **Manhattan distance** showed competitive performance but was slightly less accurate.
- **Cosine similarity** struggled because image classification depends on magnitude differences, not just angular similarity.
- **Hamming distance** performed poorly since it is unsuitable for continuous pixel values.
- **Minkowski distance** performance depended on the chosen parameter and behaved similarly to Euclidean distance for higher values.

7.2 Effect of K Value

- Small values of **K** were sensitive to noise and outliers.

- Large values of **K** caused over-smoothing and reduced classification accuracy.
- Moderate values of **K** provided a balance between bias and variance.

7.3 Comparison with Task-1

Compared to Task-1:

- Task-2 involved significantly **higher dimensional data**.
- KNN performance degraded due to the **curse of dimensionality**.
- Image data requires more advanced feature extraction methods for better performance.

8. Limitations of KNN for Image Data

- High computational cost due to distance calculations.
- Poor scalability with increasing dataset size.
- Sensitive to noise and irrelevant features.
- Suffers from the **curse of dimensionality**.

9. Conclusion

In this task, a KNN classifier was successfully implemented from scratch for multiclass image classification using the CIFAR-10 dataset.
Through systematic experimentation, the best value of K and distance metric were identified.
The results demonstrate that while KNN is simple and intuitive, it is not ideal for large-scale image classification tasks due to computational and dimensionality challenges.