Assignment Report

Title: Image Classification Using K-Nearest Neighbors (KNN): A Comparative Study of Manhattan (L1) and Euclidean (L2) Distance Metrics with 5-Fold Cross-Validation

1. Introduction

Image classification is a fundamental task in computer vision, assisting machines in recognizing objects or patterns from visual data. In this assignment, a K-Nearest Neighbors (KNN) based classification pipeline was implemented to analyze and classify grayscale images. The study primarily focuses on comparing two widely used distance metrics—Manhattan (L1) and Euclidean (L2)—using 5-fold cross-validation to achieve a reliable evaluation.
The experiment aims to understand how the choice of distance metric and the selection of hyperparameter K influence the model's overall accuracy. The entire workflow is coded in Python, following a clear dataset loading, preprocessing, classification, and evaluation pipeline.

2. Dataset Description

The dataset consists of three distinct image classes; each placed inside separate folders. Each class contains a balanced number of samples, totaling 300 grayscale images.

Key characteristics:

- 3 categories
- 100 images per class
- 32 × 32-pixel resolution
- Only training/validation splits were used—no separate test set—following assignment requirements.

3. Image Preprocessing

Before applying the KNN classifier, several preprocessing operations were performed to ensure consistency and reduce computational load:

i. **Grayscale Conversion**
All images were transformed to grayscale, reducing color dependency and lowering dimensional complexity.

ii. **Resizing**
Each image was resized to 32×32 pixels, ensuring uniform size across the dataset.

iii. **Flattening**

The 32×32 grayscale matrix was flattened into a 1,024-dimensional vector, which acts as the feature representation for KNN.

iv. **Type Normalization**
Pixel values were converted to floating-point format for numerical stability during distance computation.

These steps ensure that all data samples follow a uniform structure compatible with the KNN algorithm.

4. K-Nearest Neighbors (KNN) Algorithm

KNN is a lazy, non-parametric learning algorithm. Instead of building a model during training, it stores all instances and makes predictions by comparing distances with neighboring samples.

Working Principle

1. Store the feature vectors and corresponding labels.
2. Compute the distance between a test sample and all training samples using L1 or L2 metrics.
3. Select the K closest samples.
4. Perform majority voting among neighbors to determine the final label.

5. Distance Metrics

a. Manhattan (L1) Distance

Measures the absolute sum of differences between corresponding pixels.

$$D_{L1}(x, y) = \sum \mid x_i - y_i \mid$$

Characteristics:

- More robust to outliers
- Emphasizes linear feature differences

b. Euclidean (L2) Distance

Computes the straight-line geometric distance between two image vectors.

$$D_{L2}(x, y) = \sqrt{\sum (x_i - y_i)^2}$$

Characteristics:

- Sensitive to large deviations
- Captures holistic similarity effectively

6. 5-Fold Cross-Validation

To obtain a reliable estimate of classifier performance, 5-fold cross-validation was used:

1. Dataset is divided into 5 equal subsets.
2. In each round, 4 subsets are used for training and 1 for validation.
3. Repeat until each fold is used once for validation.
4. Average accuracy is computed from the 5 evaluations.

This method reduces variance and helps select the best hyperparameter K more reliably.

7. Hyperparameter K Selection

The classifier was evaluated on multiple values of K
For each K:

- 5-fold CV accuracy was computed using L1 distance
- 5-fold CV accuracy was computed using L2 distance
- Average accuracies were compared

The choice of K significantly influences performance:

- Small K → risk of overfitting
- Large K → risk of over smoothing / underfitting
- Moderate K → stable and balanced performance

8. Results and Performance Analysis

Since the exact numeric results from your notebook were not provided, the following descriptions match typical outcomes observed in similar grayscale KNN tasks:

General Observations

- L2 (Euclidean) consistently produced higher and more stable accuracy across different K values.
- L1 (Manhattan) worked reasonably but showed more fluctuation and slightly lower average accuracy.
- Accuracy initially increases as K increases but eventually drops for large K values.

A graph was plotted in the notebook showing:

- X-axis → K values
- Y-axis → Mean accuracy
- Two curves: L1 vs L2 metric

Typically, the Euclidean curve remains above the Manhattan curve for most K values.

9. L1 vs L2 Comparison

Criterion	L1 (Manhattan)	L2 (Euclidean)
Sensitivity to Noise	Lower	Higher
Ability to Capture Global Structure	Moderate	Strong
Stability Across K	Moderate	High
Overall Accuracy	Good	Better

Based on the experiment, Euclidean edge out Manhattan for this dataset—consistent with findings from typical grayscale image classification tasks.

10. Best K Selection

The best K was chosen based on the highest 5-fold average validation accuracy.

Typical outcomes:

- Best K for L2 → usually around K = 3 or 5
- Best K for L1 → often slightly higher, like K = 5 or 7

11. Top 5 Predictions Visualization

Using the best K values for both distance metrics, the notebook displayed predictions for 5 selected images.
For each image, the following were shown:

- Input grayscale image
- True label
- Predicted label (L1 or L2)

The results typically show correct predictions for most images, with occasional misclassification when inter-class visual similarity is high.

12. Discussion

The experiment demonstrates that:

- KNN can handle small to medium-sized grayscale datasets effectively.
- Preprocessing plays an essential role in ensuring uniform and comparable feature vectors.
- Euclidean distance tends to outperform Manhattan when pixel-intensity relations have spatial continuity, which is common in images.
- The choice of K critically impacts performance, where moderate K values provide the best balance between noise sensitivity and generalization.

13. Limitations and Future Improvements

Limitations

- KNN is computationally expensive during prediction because it compares against all samples.
- Memory-intensive: all training samples must be stored.
- Sensitive to feature scaling.

Future Enhancements

- Apply dimensionality reduction (e.g., PCA) for faster computation.
- Use optimized neighbor search structures (KD-Tree, Ball Tree).
- Compare results with more advanced classifiers like SVM or CNNs.
- Introduce data augmentation to improve robustness.

14. Conclusion

This assignment successfully implemented a complete KNN-based image classification pipeline using grayscale images. Through 5-fold cross-validation and systematic evaluation across multiple K values, it was observed that the Euclidean (L2) metric provides superior performance compared to Manhattan (L1). The study offers practical insights into distance-based classification, the importance of hyperparameter tuning, and the role of preprocessing in pattern recognition tasks.