AMERICAN INTERNATIONAL UNIVERSITY -BANGLADESH
Department of Computer Science
FACULTY OF SCIENCE and TECHNOLOGY

Program: CSE

Course Name: COMPUTER VISION AND PATTERN RECOGNITION

SECTION: I

Semester: FALL 2025-2026	**Date of Submission:** 01.12.25

MID ASSIGNMENT

……………………………………………………………………………………..

SUBMITTED TO

DR. DEBAJYOTI KARMAKER

…………………..………………………………………………………

SUBMITTED BY:

ID	NAME
22-49494-3	Aria Rahman

FOR FACULTY USE ONLY

Faculty comments: ………………………………………………………………

Signature: ………………………. **Date: ……………………….**

ASSIGNMENT 01

TITLE: A Comparison of Manhattan (L1) and Euclidean (L2) Distances with 5-fold Cross-Validation.

INTRODUCTION:

A k-Nearest Neighbors (k-NN) model was applied to classify images from animal_dataset containing three classes: panda, dog, and cat. Then, Compared the performance of two distance measures :Manhattan Distance (L1) & Euclidean Distance (L2). These were carried out using 5-fold cross-validation, and the accuracy for different values of K was recorded and plotted. The top 5 predictions were also displayed.

DATASET AND PREPROCESSING:

The dataset consisted of 300 images, which were converted to grayscale (32x32) to reduce dimensionality while preserving important features.

MODEL BUILDING AND EVALUATION:

The k-NN algorithm was used for classification. Two distance formulas were used:

- **Manhattan (L1)**: sum of absolute differences.
- **Euclidean (L2)**: square root of squared differences.

To evaluate the model more reliably, 5-fold cross-validation was used.
The dataset was split into 5 equal parts, and in each round:

- 4 folds: Training
- 1-fold: Validation

This was repeated 5 times for each K and average accuracy was then calculated.

RESULTS AND ANALYSIS:

The following observations were made:

- From the graph, the accuracy values for each K were compared using Manhattan (L1) and Euclidean (L2) distances.
- Manhattan accuracy stayed almost the same for most K values, starting around 38% and reaching up to 42%.
- Euclidean accuracy increased more as K became larger. It started at 33% and reached its highest accuracy of about 46% at K = 11.

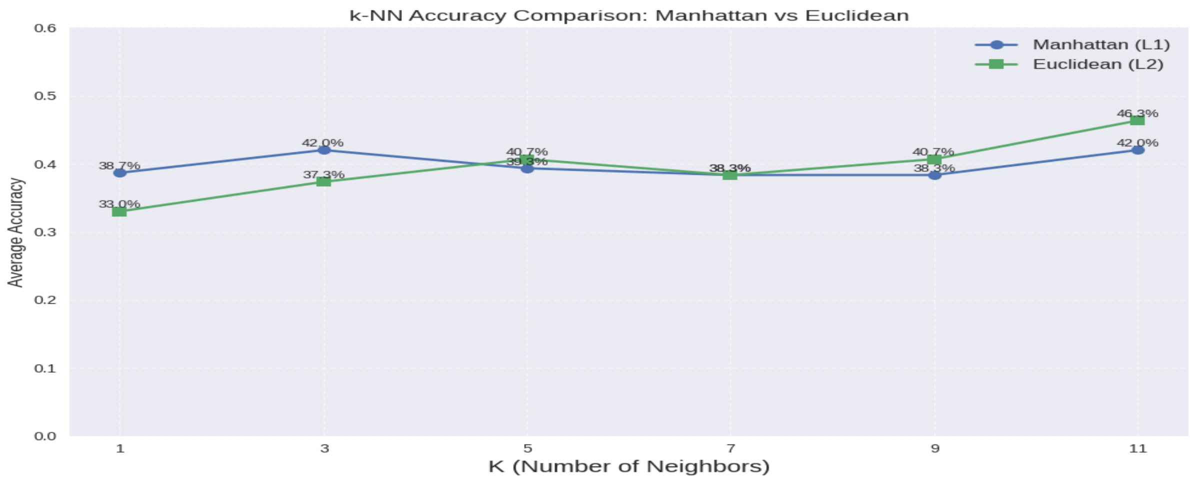

FIGURE 01: The accuracy for each value of K for both L1 and L2.

Overall, Euclidean distance performed better than Manhattan distance.

DISCUSSION:

The results showed that both distance measures. Euclidean (L2) performed slightly better than Manhattan (L1) as K increased. Manhattan accuracy stayed steady between 38.7% and 42%, while Euclidean accuracy improved from 33% to 46.3% at K=11. So, Euclidean distance was more effective for this dataset, especially for larger K values, whereas Manhattan distance remained stable but achieved lower overall accuracy.

TOP 5 PREDICTIONS:

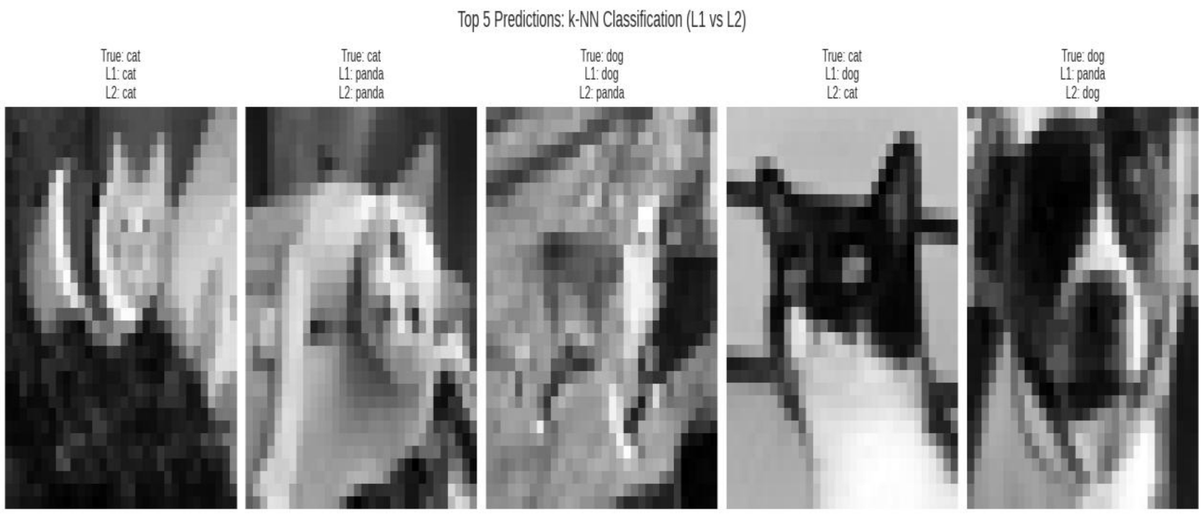

FIGURE 02: The TOP 5 prediction for L1 & L2.

From the top 5 prediction samples, it was seen that the first image was correctly classified as a cat by both L1 and L2. In the second image, both distance methods made incorrect predictions. For the third image, L1 correctly identified the dog while L2 predicted the wrong class. In the fourth image, L2 correctly predicted the cat while L1 made an error. Finally, in the fifth image, L2 again gave the correct dog prediction, while L1 was incorrect. Overall, these examples show that L2 produced more correct predictions than L1 in these samples, which supports the overall trend seen in the accuracy results.

ASSIGNMENT 02

TITLE: Implementation of a Three Hidden Layer Neural Network for Multi-Class Classification.

DATA GENERATION:

A synthetic dataset containing 3,500 samples with 10 features each was generated. Five distinct classes were ensured by summing consecutive pairs of features and assigning each sample to the class with the largest sum. The dataset was shuffled randomly to prevent any ordering bias and split into training (75%) and testing (25%) subsets. This preprocessing ensured that the input features were suitable for training a neural network for multi-class classification.

NN IMPLEMENTATION:

A neural network with three hidden layers was implemented for multi-class classification as:

- Input layer: 10 neurons; hidden layers: 64, 32,16 neurons; output layer: 5 neurons
- Activations: Sigmoid in hidden layers, SoftMax in output layer.
- Weights were randomly initialized from a standard normal distribution.
- Backpropagation was adapted for multi-class using one-hot encoding.
- Training: Gradient descent with learning rate 0.01.

TRAINING & EVALUATION:

- The network was trained over multiple epochs with early stopping to prevent overfitting.
- Each epoch involved a forward pass for predictions and backpropagation to update weights.
- Training metrics were plotted to visualize trends in loss, accuracy, and weighted precision, recall, and F1-score.

RESULT & ANALYSIS:

- Training accuracy increased and loss decreased steadily, showing effective learning.
- Per-class precision, recall, and F1-score, along with weighted averages, were tracked, handling class imbalances well.
- The confusion matrix revealed occasional misclassifications due to overlapping features.

CONCLUSION:

The multi-class neural network was successfully implemented, trained, and evaluated. The network demonstrated strong classification performance on a synthetic five-class dataset. Challenges were encountered in adapting the backpropagation algorithm for multi-class classification. These were resolved by deriving and verifying the gradient formulas. Hyperparameters, including learning rate, hidden layer sizes, and early stopping patience, were also tuned to stabilize training.

Overall, it was observed that careful weight initialization, appropriate activation functions, and early stopping contributed significantly to the stability and accuracy of the model.