AMERICAN INTERNATIONAL UNIVERSITY–BANGLADESH (AIUB)

FACULTY OF SCIENCE & TECHNOLOGY

Computer Vision & Pattern Recognition

FALL 2025-2026

Section: A

Assignment

Submitted By:

MD.SHARJEEL HASSAN KHAN
22-48915-3

Submitted To:

DR. DEBAJYOTI KARMAKER

Date: 01 December 2025

Assignment 1:
Image Classification: A Comparison of Manhattan (L1) and Euclidean (L2) Distances with 5-fold Cross-Validation

1. Introduction

This project focuses on classifying grayscale images from three categories cat, dog, and panda using the k-nearest neighbors (k-NN) algorithm. The goal is to compare two distance metrics, Manhattan (L1) and Euclidean (L2), to identify which performs better on the dataset. The images are resized and converted to grayscale to reduce complexity while keeping essential features.

2. Model Building and Evaluation

The model uses k-NN, where each image is represented as a 1024 dimensional feature vector obtained from a 32×32 grayscale image. Multiple K values are tested, and both L1 and L2 distances are used to measure similarity between images. To ensure reliable evaluation, 5-fold cross-validation is applied, allowing the model to be trained and validated across different subsets of data. For each K value and distance metric, accuracy is recorded and averaged across five folds to assess overall performance.

3. Results and Analysis

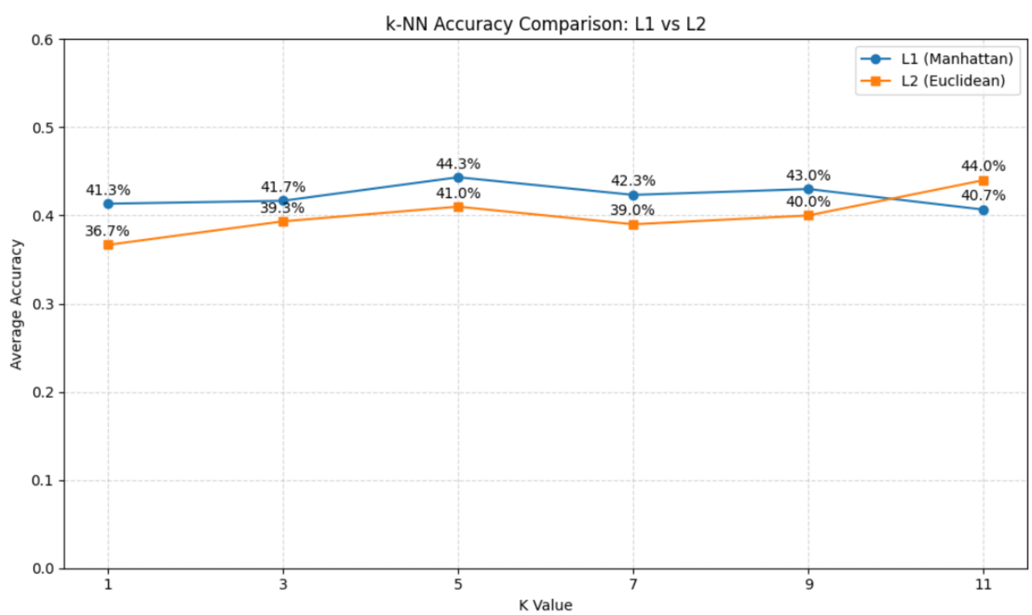

Figure: K-NN Accuracy Comparison

The accuracy trends show that L1 generally performs better than L2 for most K values, especially around K = 5, where the highest accuracy is observed. L2 distance shows slightly lower and more fluctuating performance across folds. The plotted graph clearly highlights how accuracy changes with different K values and demonstrates that the dataset responds more consistently to Manhattan distance. Overall, L1 achieves more stable and higher average accuracy than L2.

4. Discussion

The results suggest that Manhattan distance is more suitable for this dataset, likely because pixel intensity differences in grayscale images align well with L1's absolute difference calculations. The performance of L2 was slightly weaker, possibly due to its sensitivity to larger variations in pixel values. Increasing K beyond a certain point also reduced accuracy, indicating that too many neighbors can introduce noise. Further improvements could include feature extraction or dimensionality reduction techniques.

5. Top 5 Predictions

Five random test images were selected, and the k-NN model (K=3) predicted their classes using both L1 and L2 distances. The displayed results show the true label alongside predictions from both metrics. This visualization helps confirm the model's ability to classify new images and highlight cases where the two distance functions differ in prediction accuracy.

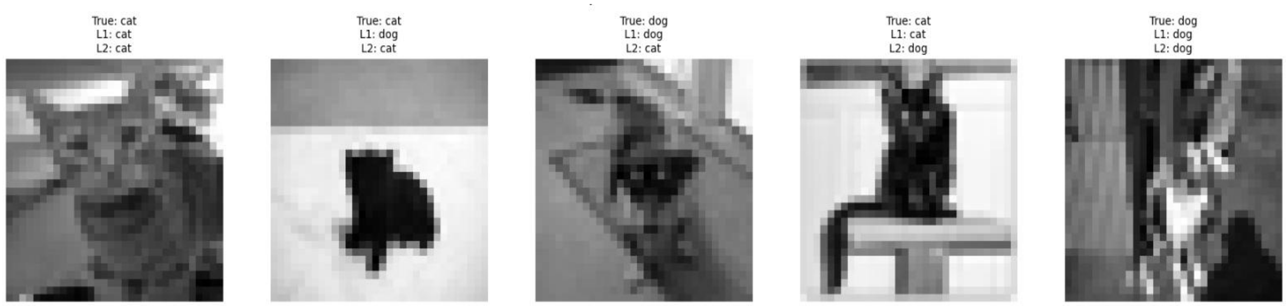

Figure: Top 5 Predictions

Assignment 2:
Implementation of a Three Hidden Layer Neural Network for Multi-Class Classification

Introduction:

This project focuses on implementing a neural network with three hidden layers for a five-class classification task. A synthetic dataset was generated, and the provided codebase was extended to support multi-class outputs using softmax activation and cross-entropy loss. The goal of the project was to modify the architecture for multi-class learning, train the model, evaluate its performance, and analyze the results.

Dataset Generation:

A synthetic dataset consisting of 3500 samples ,10 input features and 5 target classes were created using NumPy. To generate class labels, every two features were grouped and summed, and the class was assigned by selecting the pair with the highest sum. The dataset was shuffled and split into 75% training set (2625 samples) and 25% testing set (875 samples).
Class distribution remained balanced across all splits.

Model Evaluation

The neural network was evaluated using multiple metrics to assess its performance on the multi-class classification task. The model predicts class probabilities through the softmax output layer and is trained using cross-entropy loss. During training, both training and validation losses were monitored, and early stopping was applied to prevent overfitting when the validation loss did not improve for 50 consecutive epochs. After training, the model's predictions were compared with the true labels to calculate accuracy, as well as precision, recall, and F1-score for each class and in weighted form. A confusion matrix was also generated to visualize the distribution of correct and incorrect predictions across classes. This evaluation framework provides a detailed understanding of the network's strengths and weaknesses for the multi-class task.

Results and Analysis:

The model achieved a test accuracy of approximately 20.34%, which is close to random guessing for a five-class balanced dataset. During testing, the network predicted almost all samples as Class 0, resulting in high recall for that class but zero precision, recall, and F1-score for the remaining four classes.

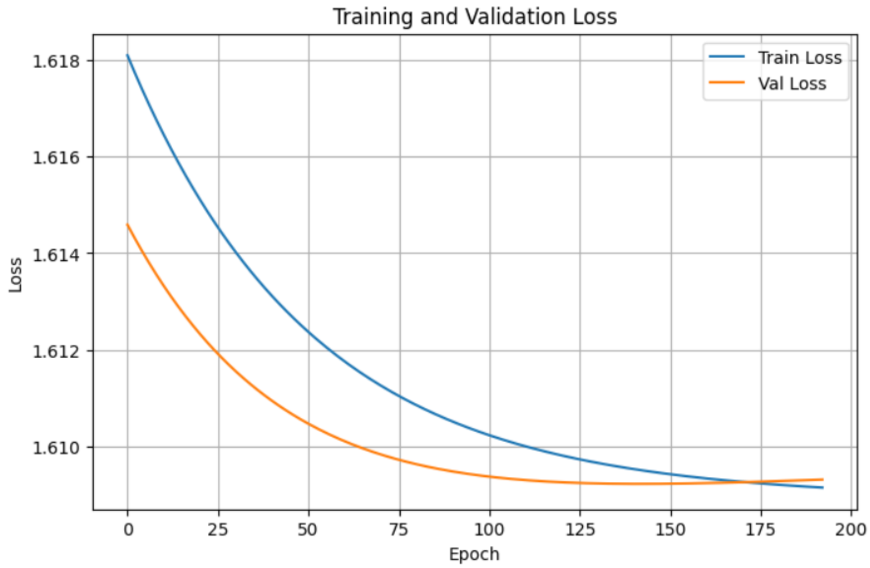

Figure 1: Training and Validation Loss Curve

Figure 1 shows the training and validation loss curves over the epochs. The curves indicate that the model stopped improving early, which likely triggered early stopping. This suggests that the network struggled to learn meaningful class boundaries, possibly due to the use of sigmoid activations, vanishing gradients, and the non-linear nature of the synthetic dataset.

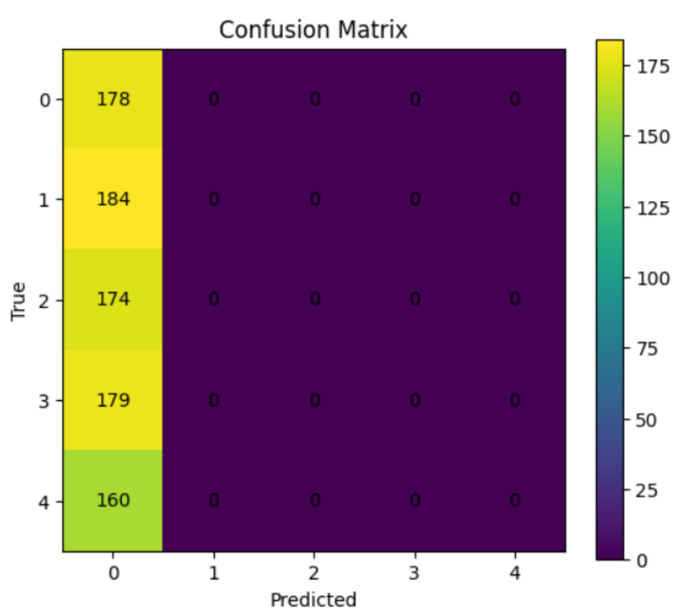

Figure 2: Confusion Matrix

Figure 2 presents the confusion matrix for the test set. The confusion matrix confirms that most predictions were assigned to Class 0, highlighting the model's difficulty in distinguishing between the five classes. Despite these limitations, the experiment successfully implemented multi-class classification with softmax outputs, cross-entropy loss, and evaluation metrics.

Conclusion

In conclusion, the project successfully demonstrated how to build and train a three-hidden-layer neural network for multi-class classification. While the model's accuracy remained low, the work provided a clear understanding of data generation, forward and backward propagation, and multi-class evaluation. The challenges faced highlight the need for better activation functions, initialization, or optimization methods to improve learning. Overall, the objectives of the project were met, and the experiment offered useful insights into neural network behavior and limitations.