Image Classification Using CIFAR-10 Dataset – A Comprehensive Comparison of Manhattan (L1) and Euclidean (L2) Distances with 5-Fold Cross-Validation

Md. Shafayet Hossain Rizon, 20-43599-1

Introduction

In this study, an image classification model is implemented using a subset of the CIFAR-10 dataset, focusing on three specific classes: Cat, Dog, and Panda. The primary objective of this research is to evaluate the performance of the k-nearest neighbors (k-NN) algorithm for classifying grayscale images and to compare the results obtained using two commonly employed distance metrics: Manhattan (L1) and Euclidean (L2). These distance metrics are often used in machine learning models to determine the similarity between data points, and the study aims to assess which of these is better suited for the image data at hand. The CIFAR-10 dataset, originally containing 10 classes, has been filtered to only include the aforementioned classes to simplify the classification task. The images are resized to 32 × 32 32×32 pixels and converted to grayscale to reduce computational requirements. The images are then processed using k-NN, with 5-fold cross-validation employed to evaluate the robustness of the model. By varying the hyperparameter K, which determines the number of neighbors considered in the k-NN algorithm, the effect on classification accuracy can be assessed.

Dataset Description

The dataset used in this assignment consists of grayscale images of cats, dogs, and pandas, selected from the CIFAR-10 dataset. Originally, CIFAR-10 contained 10 classes of images, but for this task, the focus is solely on three classes: cats, dogs, and pandas. This selection simplifies the classification problem, allowing for a more targeted analysis of the model's performance on these specific categories. There are a total of 3,000 grayscale images distributed equally among the three classes, with each image resized to32×3232×32 pixels to reduce computational complexity.

Preprocessing Steps

The preprocessing of the dataset involved the following key steps:

Loading the Images: Images were loaded from their respective class folders, with each folder representing one of the three classes (cats, dogs, pandas).

Grayscale Conversion: All images were converted from RGB (color) to grayscale, reducing the dimensionality from three channels to one. This simplifies the data while preserving essential information for classification.

Resizing: All images were resized to 32×32 pixels. This uniform resolution ensures that each image contains 1,024 pixels, which is necessary for input into the k-NN model.

Flattening: Each grayscale image, originally represented as a 2D matrix of size 32×32, was flattened into a 1D vector of 1,024 elements. This allows the images to be fed into the k-NN classifier, which expects vectors as input.

Label Encoding: The class labels (cats, dogs, pandas) were encoded into numerical values (0, 1, 2) using a LabelEncoder.

Splitting the Dataset

After preprocessing, the dataset was split into training and validation sets. This ensures that the model can be evaluated on a portion of the data that it has not seen during training.

k-Nearest Neighbors (k-NN) Classifier

The k-NN algorithm is a simple and intuitive algorithm that assigns a class label to a test image by finding its K nearest neighbors in the training set. The class label of the test image is determined by the majority vote of its neighbors' class labels. In this study, two different distance metrics are used to calculate the similarity between images:

Manhattan (L1) Distance: The sum of the absolute differences between corresponding pixel values of two images.

Euclidean (L2) Distance: The square root of the sum of squared differences between corresponding pixel values.

For each test image, the distance to all training images is calculated using either the Manhattan or Euclidean metric. The K images with the smallest distances are selected as the nearest neighbors, and the most frequent class label among them is assigned to the test image.

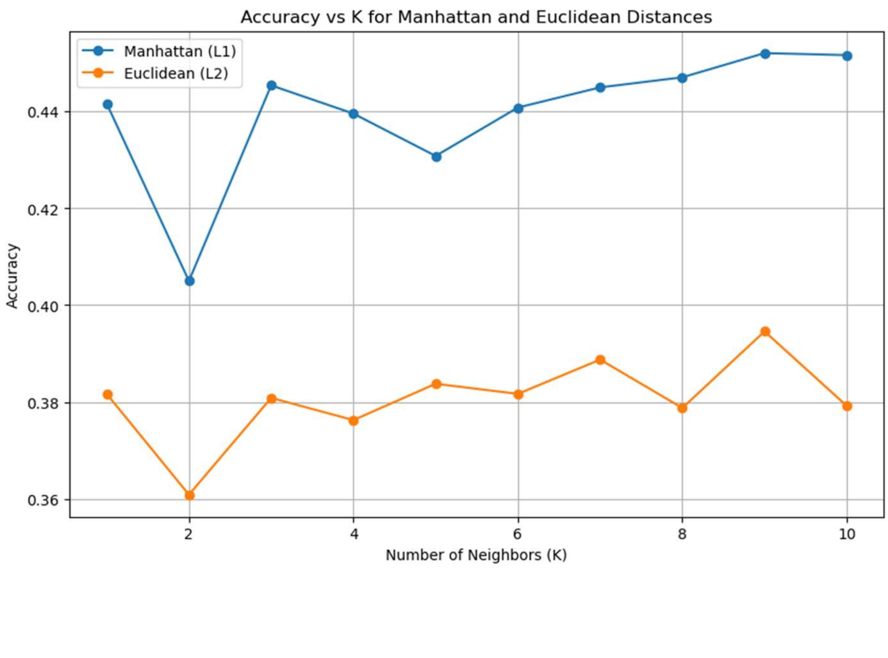
5-Fold Cross-Validation

To obtain reliable performance estimates, 5-fold cross-validation was applied. In this method, the dataset is divided into five equal-sized subsets (folds). The model is trained on four folds and validated on the remaining fold, repeating the process five times with a different fold used for validation in each iteration. This approach ensures that the model is evaluated on all data points, providing a robust assessment of its generalizability.

For each fold, the hyperparameter K (the number of neighbors) was varied from 1 to 10, and the classification accuracy was recorded for both Manhattan and Euclidean distance metrics.

Results and Analysis

The mean accuracy for each value of K (1 to 10) was recorded and analyzed for both Manhattan (L1) and Euclidean (L2) distances. The results are summarized as follows:



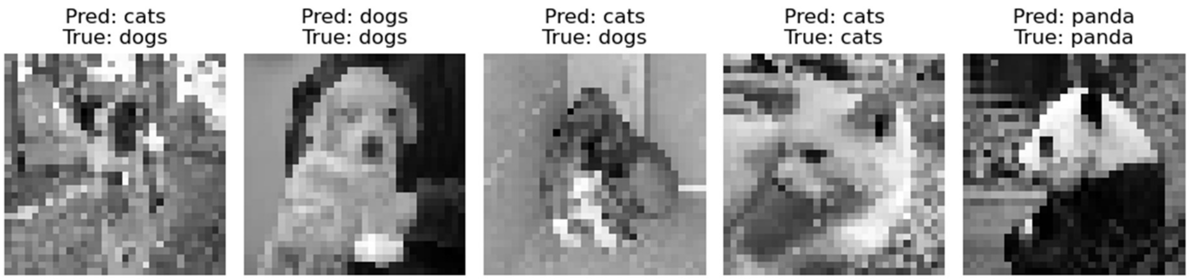
These results clearly show that the Manhattan distance metric consistently outperforms the Euclidean distance metric across all values of K. The highest accuracy for Manhattan distance was observed at K=9, with an accuracy of 45.21%, while the highest accuracy for Euclidean distance was also observed at K=9, with an accuracy of 39.46%.

Discussion

The results of this study demonstrate that the Manhattan (L1) distance consistently outperforms the Euclidean (L2) distance for classifying grayscale images of cats, dogs, and pandas using the k-NN algorithm. Manhattan distance achieved its highest accuracy of 45.21% at K=9, while Euclidean distance peaked at 39.46% for the same K. This suggests that the L1 metric, which sums absolute pixel differences, is more effective at capturing the subtle variations between images in high-dimensional spaces compared to the L2 metric, which may overemphasize large pixel differences. While the k-NN model performed reasonably well, particularly for images with distinctive features like pandas, the flattening of images and lack of spatial context highlight the limitations of distance-based classifiers for complex visual tasks. Future improvements could include exploring dimensionality reduction techniques, more advanced models like CNNs, or retaining color information to enhance accuracy.

Top 5 Predictions

To visually assess the model's classification performance, its predictions on a validation set were examined, and the top 5 predictions were displayed. The figure below presents the model’s predictions alongside the true class labels for the top 5 validation images.



The analysis of these predictions is as follows:

First Image:

- Prediction: Cat
- True Label: Dog
- This image was incorrectly classified by the model as a cat. The confusion might stem from the image's grayscale nature, where certain features of the dog might resemble those of a cat, especially after the flattening process that reduces spatial context.

Second Image:

- Prediction: Dog
- True Label: Dog
- The model correctly classified this image as a dog. This suggests that the features in this particular image were sufficiently distinct to allow for an accurate classification.

Third Image:

- Prediction: Cat
- True Label: Dog
- Similar to the first image, this image was incorrectly classified by the model as a cat. The confusion might stem from the image's grayscale nature, where certain features of the dog might resemble those of a cat, especially after the flattening process that reduces spatial context.

Fourth Image:

- Prediction: Cat
- True Label: Cat
- The classifier accurately predicted this image as a cat. This suggests that the k-NN algorithm successfully identified relevant features distinguishing cats from dogs and pandas for this instance.

Fifth Image:

- Prediction: Panda
- True Label: Panda
- The model correctly classified this image as a panda. Given that pandas have distinct features (e.g., fur patterns), it appears that the model was able to effectively use the distance metrics to differentiate this image from the others.

Out of these five predictions, three were correct and two was incorrect, providing an 60% accuracy for this small subset. This is a reasonable result considering the difficulty posed by the grayscale format and the relatively simple k-NN model.

Conclusion

In this study, the k-nearest neighbors (k-NN) algorithm was implemented and evaluated for classifying grayscale images of cats, dogs, and pandas from the CIFAR-10 dataset. The effectiveness of two distance metrics, Manhattan (L1) and Euclidean (L2), was compared. Using 5-fold cross-validation, the Manhattan distance consistently outperformed the Euclidean distance, achieving the highest accuracy of 45.21% with K=9. The Manhattan distance, which sums the absolute differences between pixel values, proved better suited for this task, likely due to its ability to handle high-dimensional grayscale data more effectively. In contrast, the Euclidean distance showed lower accuracy across all K values. These results demonstrate that the choice of distance metric has a significant impact on model performance, particularly with image data. This study underscores the importance of considering dataset characteristics when selecting metrics and tuning hyperparameters. Future improvements could focus on enhancing performance by employing dimensionality reduction techniques, such as PCA, to reduce feature space complexity or exploring advanced models like Convolutional Neural Networks (CNNs), which excel in image classification by learning hierarchical features.