CIFAR-10 Image Classification: KNN vs. CNN

A Project from the Aygaz Yapay Zekaya Giriş Bootcamp

Emre Baytimur - 22/06/2024

Outline

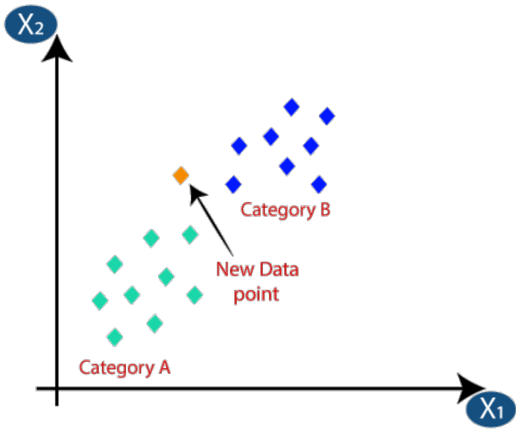
- Aim of the Project
- Materials and Methods
- Model Results
- Comparison
- References

CIFAR-10 Image Classification

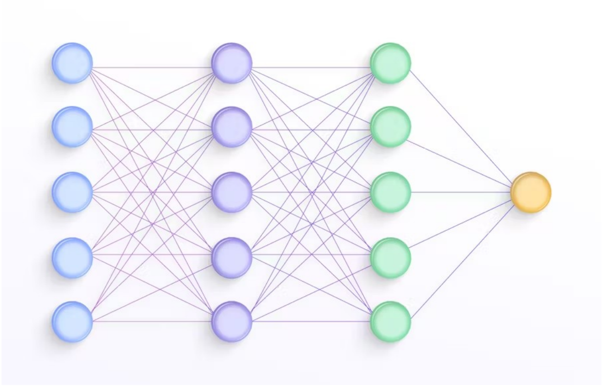
The current project tries to;

- Classify Images in CIFAR-10
- Compare the performance of two different machine learning approaches: K-Nearest Neighbors (KNN) and Convolutional Neural Networks (CNNs) on CIFAR10

KNN



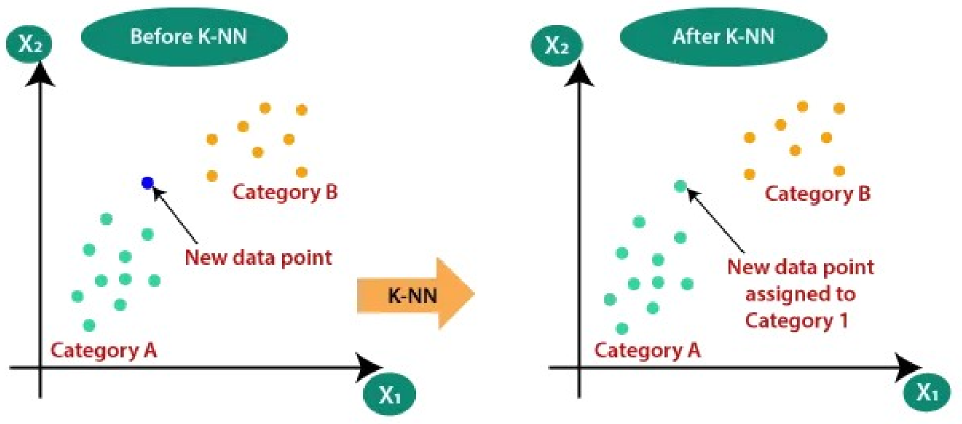
CNNs



KNN vs CNNs

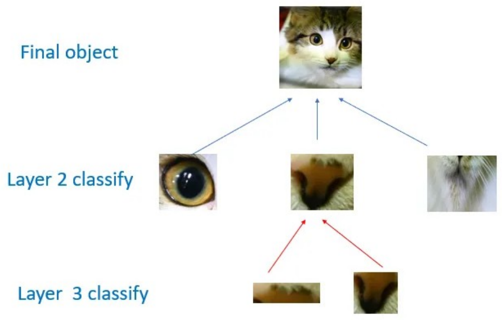
- KNN

KNN is a machine learning algorithm that utilizes the k closest (distance) examples in the training data for a new data point to make classifications or predictions. It assigns the most common class (classification) or average value (regression) to the new point.



- CNNs

CNNs are multi-layered artificial neural networks that learn to identify complex patterns in images by analyzing them through a series of filters. The network consist of neurons each examining specific parts of an image and working together to classify/recognize the object



Hierarchy of how CNN splits images in a layer by layer

CIFAR10 Dataset

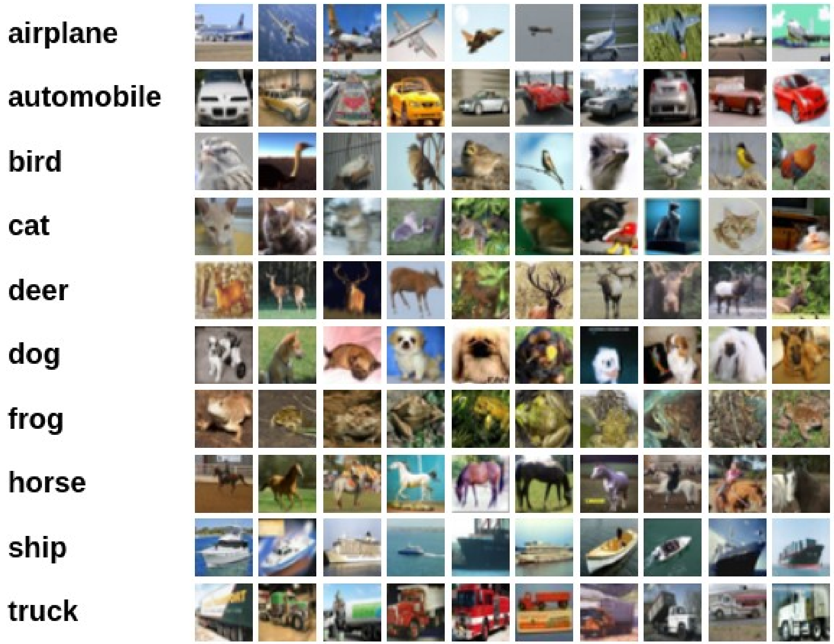
- 60000 images

Training set
- 50000 images

Test set
- 10000 images

Each Image: 32x32 pixels

10 classes, 6000 images per class



KNN Results

Model parameters	5 neighbors	10 neighbors	15 neighbors
Minkowski	**Accuracy**: 0.3398 **F1 Score**: 0.3260 **Precision**: 0.4304 **Recall**: 0.3398	**Accuracy**:0.3386 **F1 Score**: 0.3244 **Precision**: 0.4587 **Recall**: 0.3386	**Accuracy**: 0.3405 **F1 Score**: 0.3232 **Precision**: 0.4607 **Recall**: 0.3405
Manhattan	**Accuracy**: 0.3770 **F1 Score**:0.3724 **Precision**: 0.4571 **Recall**: 0.3770	**Accuracy**: 0.3810 **F1 Score**: 0.3748 **Precision**: 0.4692 **Recall**: 0.3810	**Accuracy**: 0.3733 **F1 Score**: 0.3662 **Precision**: 0.4662 **Recall**: 0.3733
Euclidean	**Accuracy**: 0.3398 **F1 Score**: 0.3260 **Precision**: 0.4304 **Recall**: 0.3398	**Accuracy**:0.3386 **F1 Score**: 0.3244 **Precision**: 0.4587 **Recall**: 0.3386	**Accuracy**: 0.3405 **F1 Score**: 0.3232 **Precision**: 0.4607 **Recall**: 0.3405

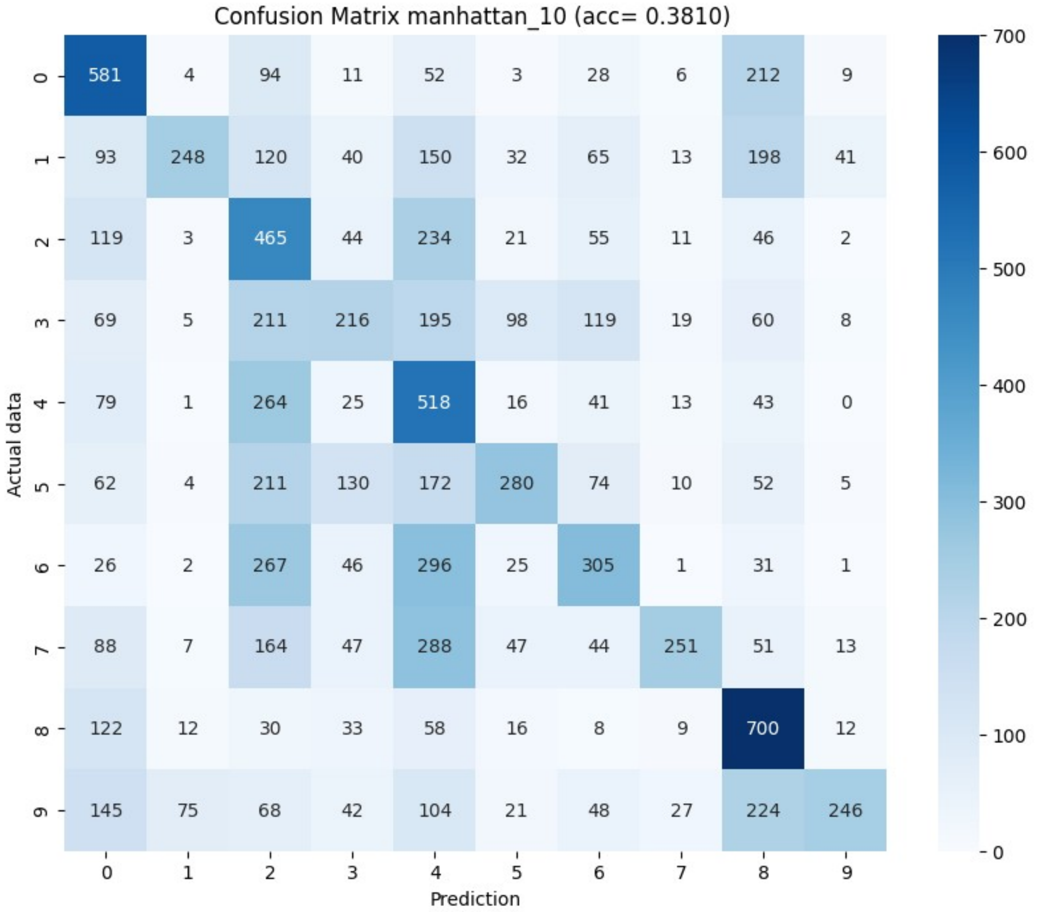
***Accuracy**:* The percentage of correct predictions made by the model.
***F1 Score**:* Combined measure of precision and recall
Precision: Proportion of *True Positives.*
Recall: The proportion of actual positive cases that the model correctly identified as positive – avoiding false positives.

KNN Results

- Overall, the best model was *manhattan_10*

- Manhattan distance achieved the best results among the tested metrics (k = 10)

- However, accuracy remains below 0.4 (3810), indicating a poor performance



CNN Model

Layer (type)	Output Shape	Param #
conv2d_8 (Conv2D)	(None, 30, 30, 32)	896
dropout_6 (Dropout)	(None, 30, 30, 32)	0
conv2d_9 (Conv2D)	(None, 28, 28, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 14, 14, 64)	0
dropout_7 (Dropout)	(None, 14, 14, 64)	0
conv2d_10 (Conv2D)	(None, 12, 12, 128)	73,856
max_pooling2d_5 (MaxPooling2D)	(None, 6, 6, 128)	0
flatten_4 (Flatten)	(None, 4608)	0
dense_5 (Dense)	(None, 64)	294,976
dropout_8 (Dropout)	(None, 64)	0
dense_6 (Dense)	(None, 10)	650

Total Layers: 9
3 Convolutional Layers (Conv2D)
2 Pooling Layers (MaxPooling2D)
1 Flatten Layer
2 Dense Layers
1 Output Layer (Dense)

Layer Descriptions

Convolutional layers --> learn features.
Pooling layers --> downsample the data.
Dropout layers --> prevent overfitting.
Flatten layers --> *reshapes feature maps*
Dense layers --> reasoning for classificatior

CNN Model

Gradual Increase in Complexity in Convolutional Layers

Increasing in the number of neurons (32, 64, 128) in the convolutional layers (progressive learning)

Max Pooling for Downsampling

Lowering the computational cost by reducing image size while preserving important features

Dropout for Regularization

Dropout layers to prevent overfitting by forcing the model to rely on generalization

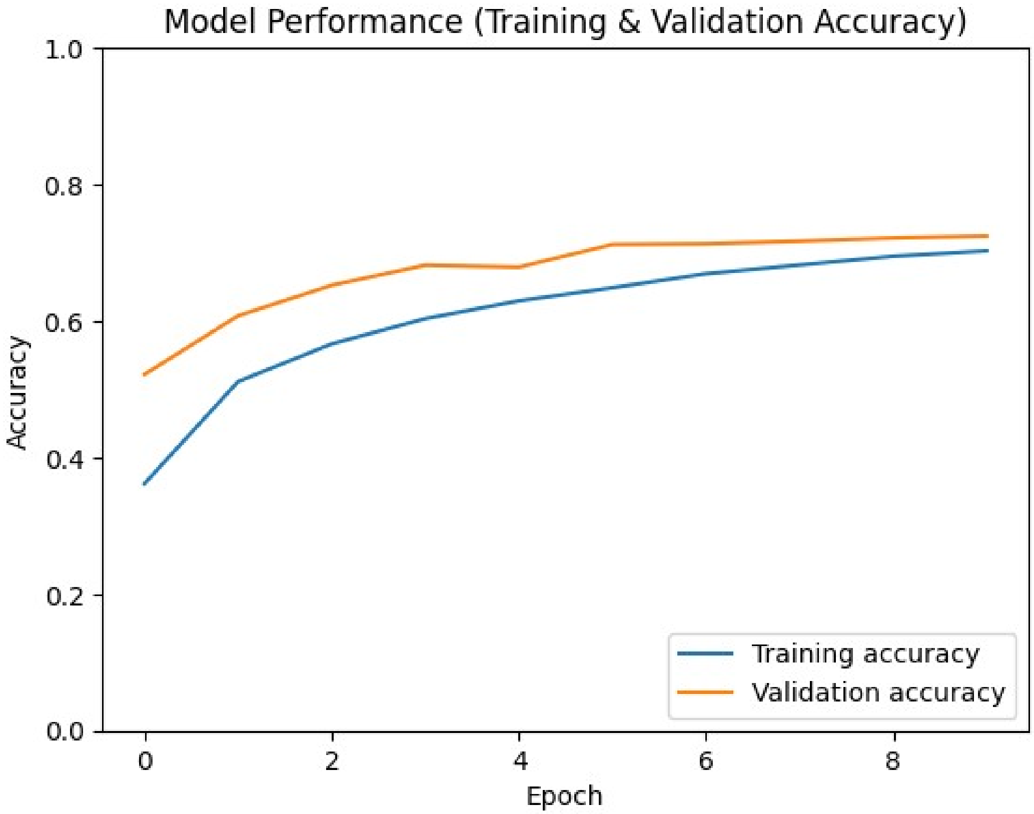
ReLU Activation

Introduces non-linearity, allowing the model to learn more complex relationships between features

Softmax Output

Assign probabilities for each class in the CIFAR-10 dataset, let the model picks the most likely one

CNN Model Performance

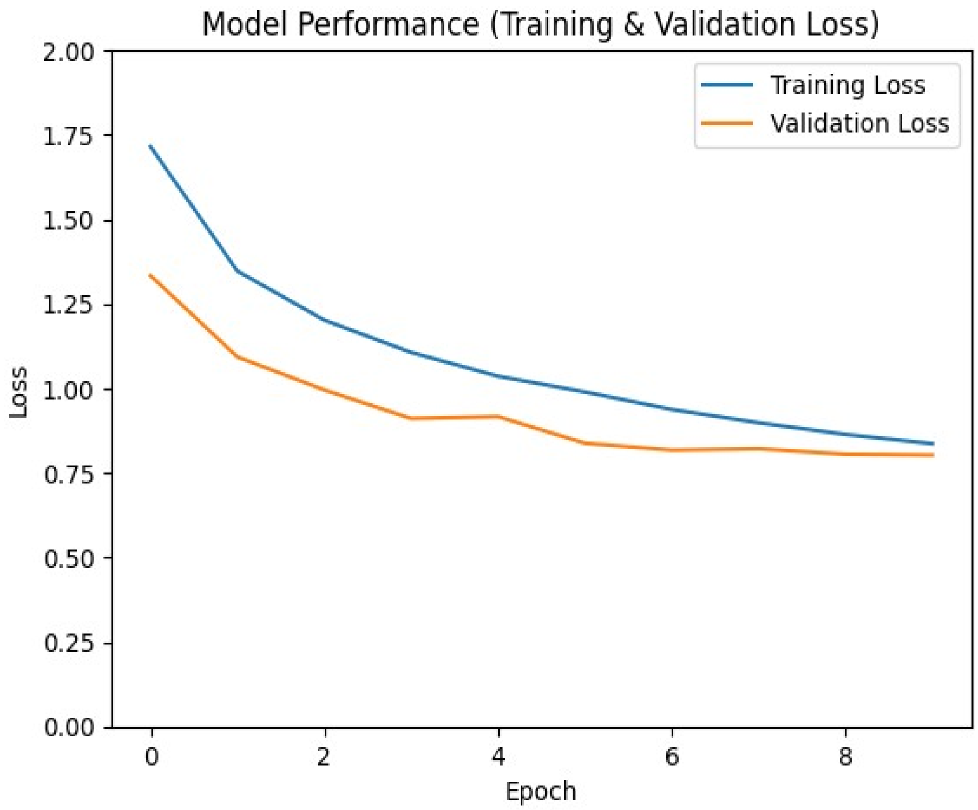


Accuracy reached to *0.7040*
Validation accuracy reached to *0.7242*

The model shows good signs of generalizability!

Validation accuracy is consistently higher than **training accuracy,** which suggests the model is learning effectively and classifies well the test data.

CNN Model Performance



Loss: 0.8306 (at the final epoch)
Validation Loss: 0.8026 (at the final epoch)

Both **training loss** and **validation loss** are decreasing over training! Also, The **validation loss** is continously lower than the **training loss!**

Our CNN model might be learning and generalizing well.

Comparsion and Discussion

- Traditional ML methods may perform worse than DL methods on CIFAR10 classification task
- Our CNN performed better in terms of the loss and accuracy when compared to KNN
- Thus, CNN is a better choice for image classification tasks
- Overall, CNN model is trained successfully and it is learning from the training data and performing better than our all KNN models.

References

- Ai, S. E. (n.d.). What is a Convolutional Neural Network?. What is a convolutional neural network? https://skyengine.ai/se/skyengine-blog/125-what-is-a-convolutional-neural-network
- Harshi, H. (2021, September 21). Convolutional Neural Network. Medium. https://medium.com/analytics-vidhya/cnn-32fbc1706a89
- Krizhevsky, A., Nair, V., & Hinton, G. (n.d.). The CIFAR-10 dataset. CIFAR-10 and CIFAR-100 datasets. https://www.cs.toronto.edu/~kriz/cifar.html
- What is the K-nearest neighbors algorithm?. IBM. (2021, October 4). https://www.ibm.com/topics/knn