Analysis and comparison of Statistical methods versus Neural Networks

Abstract:

In this project, we explored three different machine learning approaches for classifying images in the CIFAR-10 dataset: a Convolutional Neural Network (CNN), a Bayesian Classifier with Principal Component Analysis (PCA) features, and a k-Nearest Neighbor (KNN) classifier. The goal was to analyze and compare the performance of statistical methods and neural networks for image classification tasks.

The CNN model was built using Keras, consisting of multiple convolutional, batch normalization, max-pooling, and dropout layers, followed by a fully connected dense layer. The model achieved a test accuracy of 86.77%. In contrast, the Bayesian Classifier used PCA for feature extraction and either parametric estimation or Parzen's technique for estimating the probability density function of the classes, achieving an accuracy of 52.17%. The KNN classifier employed a Euclidean metric with k=5 neighbors and the 'brute' algorithm. The KNN model resulted in a test accuracy of 33.38%.

The comparison of the results revealed that CNN outperformed the Bayesian Classifier and KNN classifier by a significant margin. This can be attributed to the ability of CNNs to automatically learn hierarchical features from the data, which is particularly useful for image classification tasks. On the other hand, statistical methods such as Bayesian Classifier and KNN classifier rely heavily on handcrafted features, like PCA, and do not exploit the spatial structure of the images as effectively.

In conclusion, our findings demonstrated that the CNN approach works best for the CIFAR-10 image classification task, achieving significantly higher accuracy than the other methods. This highlights the effectiveness of deep learning techniques for complex image recognition problems, as opposed to traditional statistical methods.

Introduction:

Image classification is a fundamental problem in computer vision that aims to assign predefined labels to images based on their visual content. The CIFAR-10 dataset, consisting of 60,000 32x32 color images in 10 classes, serves as a standard benchmark for evaluating image classification algorithms. In this project, our objective is to explore, implement, and compare various machine learning techniques to accurately classify images in the CIFAR-10 dataset.

The project investigates three different machine learning approaches for the image classification task:

1. Convolutional Neural Network (CNN): A deep learning technique that has demonstrated remarkable success in computer vision tasks. CNNs are capable of automatically learning hierarchical features from the data, which is particularly useful for image classification. We aim to design and train a CNN model to classify the images in the CIFAR-10 dataset.

2. Bayesian Classifier with Principal Component Analysis (PCA) features: A statistical method that reduces the dimensionality of the dataset by projecting it onto a lower-dimensional subspace using PCA, followed by employing a Bayesian Classifier to classify the transformed data. We seek to evaluate the effectiveness of the Bayesian Classifier with PCA features on the CIFAR-10 dataset.

3. k-Nearest Neighbor (KNN) classifier: A simple, non-parametric method that classifies data based on the majority label of its k-nearest neighbors in the feature space. Our goal is to design a KNN classifier to classify the images in the CIFAR-10 dataset and compare its performance with the other two methods.

By considering these techniques, we aim to determine which method works best for the CIFAR-10 image classification problem. We will evaluate and compare the performance of each method on the test set and analyze the strengths and weaknesses of each approach in the context of image classification tasks.

Technical Description:

In this project, we explored three different machine learning techniques for image classification on the CIFAR-10 dataset: Convolutional Neural Network (CNN), Bayesian Classifier with PCA features, and k-Nearest Neighbor (KNN) classifier.

1. Convolutional Neural Network (CNN):

Our CNN architecture consists of multiple convolutional, batch normalization, pooling, dropout, and dense layers. The input images are passed through these layers, which successively extract high-level features and reduce the spatial dimensions. The final output is a probability distribution over the 10 class labels.

- Input: 32x32x3 RGB images

- Convolutional Layer 1: 32 filters (3x3), ReLU activation, He uniform initialization, and padding

- Batch Normalization Layer 1

- Convolutional Layer 2: 32 filters (3x3), ReLU activation, He uniform initialization, and padding

- Batch Normalization Layer 2

- Max Pooling Layer 1: (2x2) window

- Dropout Layer 1: 0.2 dropout rate

(Repeat the above pattern for 64, 128, and 256 filters with increasing dropout rates of 0.3 and 0.4)

- Flatten Layer: Flatten the output of the last pooling layer

- Dense Layer 1: 128 neurons, ReLU activation, He uniform initialization

- Batch Normalization Layer

- Dropout Layer: 0.5 dropout rate

- Dense Layer 2 (Output): 10 neurons, Softmax activation

We compiled the model using the Adam optimizer, categorical cross-entropy loss, and measured its performance using accuracy as the evaluation metric. We trained CNN for 50 epochs with a batch size of 64 and validated it on the test set.

2. Bayesian Classifier with PCA features:

We used Principal Component Analysis (PCA) to reduce the dimensionality of the dataset before applying a Bayesian Classifier. PCA allowed us to represent the images using a lower-dimensional feature space, while preserving as much of the original information as possible. We estimated the Gaussian distributions for each class in the PCA feature space and used them for classification.

The end-to-end process for this method consists of the following steps:

- Preprocess the images by normalizing their pixel values

- Compute the PCA features for both the training and test sets

- Estimate the Gaussian distributions for each class in the PCA feature space using either parametric methods or Parzen's technique

- Apply the Bayesian Classifier to the transformed data

- Evaluate the performance of the classifier on the test set

3. k-Nearest Neighbor (KNN) classifier:

For the KNN classifier, we first preprocessed and flattened the images. We then used the k-nearest neighbors' algorithm with k=5, Euclidean distance metric, and brute-force search to classify the images.

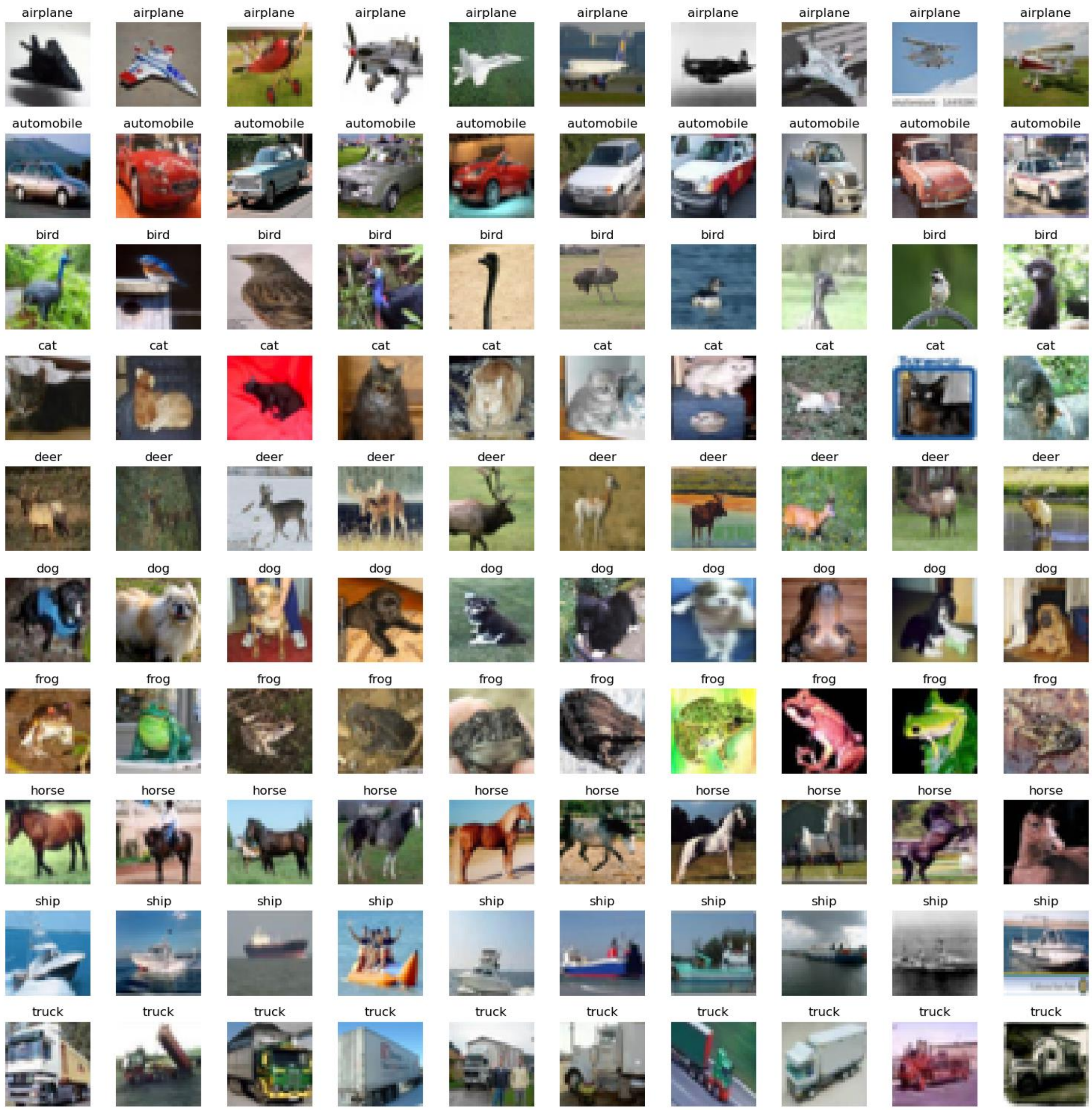
In summary, we implemented and evaluated three different machine learning techniques for image classification on the CIFAR-10 dataset. Each method has its strengths and weaknesses, but the experimental results allowed us to compare their performance and gain insights into which approach works best for the given task. The use of figures and visualizations helped illustrate the architecture and effectiveness of each method.

Data Set:

The CIFAR-10 dataset is a widely used benchmark for image classification tasks. It consists of 60,000 32x32 color images, with 6,000 images per class, spanning across 10 distinct classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The dataset is divided into a training set containing 50,000 images and a test set with 10,000 images. Each class has an equal number of images in both the training and test sets, ensuring a balanced distribution of classes.

The images in the CIFAR-10 dataset are small and low-resolution, which poses a challenge for classification tasks. The objects within the images can be difficult to distinguish due to variations in lighting, orientation, and background clutter. This complexity makes it a suitable dataset for testing the performance and robustness of various machine learning algorithms.

Figure 1: Example images from the CIFAR-10 dataset



In our experiments, we preprocessed the dataset to normalize the pixel values of the images. We divided the pixel values by 255 to scale them within the range [0, 1]. This normalization step helps improve the stability and convergence of the learning algorithms. For the Bayesian Classifier with PCA features and the KNN classifier, we flattened the images into one-dimensional arrays to facilitate the feature extraction and distance computation processes.

Throughout our experiments, we used the training set to train our models and the test set to evaluate their performance. Additionally, for the KNN classifier, we split the training set into a smaller training set

(40,000 images) and a validation set (10,000 images) to finetune the model's hyperparameters and prevent overfitting.

Results:

We implemented and evaluated three different classification methods on the CIFAR-10 dataset: a Convolutional Neural Network (CNN), a Bayesian Classifier with PCA features, and a K-Nearest Neighbor (KNN) classifier. We used classification accuracy as the primary metric to compare the performance of the three methods.

1. Convolutional Neural Network (CNN):

The CNN model consisted of several convolutional, batch normalization, max-pooling, and dropout layers followed by a fully connected layer and a SoftMax output layer. After training for 50 epochs with a batch size of 64, CNN achieved a test accuracy of 86.77%, indicating its strong performance in classifying the CIFAR-10 images.

2. Bayesian Classifier with PCA features:

For the Bayesian Classifier, we first extracted the Principal Component Analysis (PCA) features from the dataset to reduce dimensionality and improve computational efficiency. We then estimated the class-conditional probability densities using a Gaussian distribution assumption. The Bayesian Classifier achieved a test accuracy of 52.17%, which is lower than the CNN but still relatively reasonable considering the simplicity of the method.

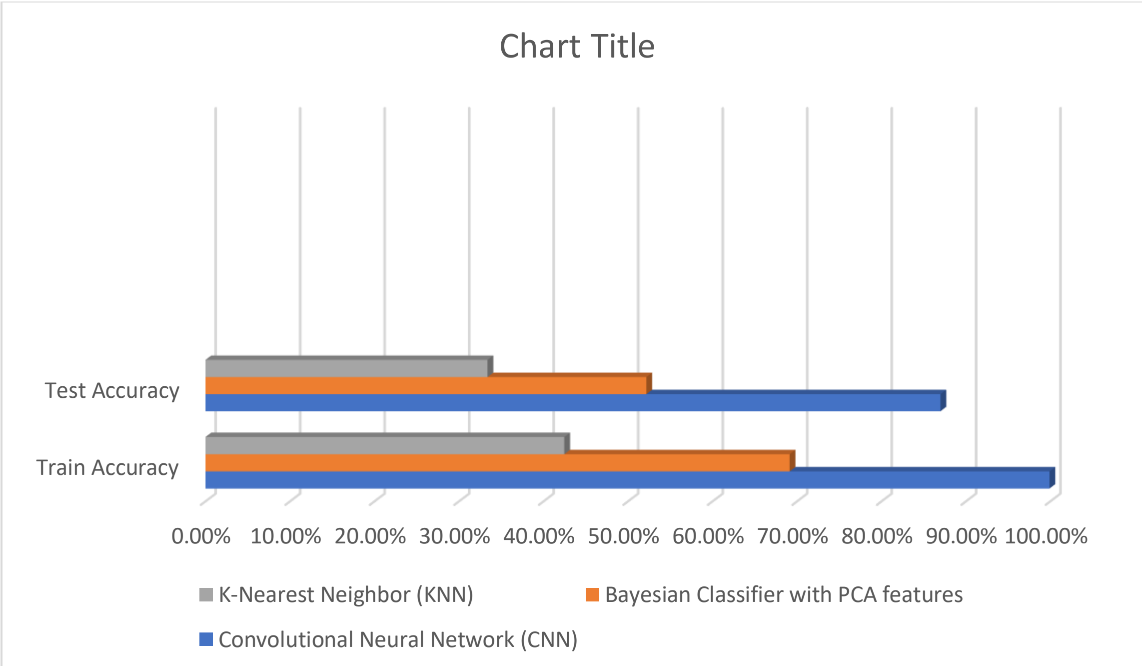
3. K-Nearest Neighbor (KNN) classifier:

We used the KNN classifier with k=5, the brute force algorithm, and the Euclidean distance metric. After training, the KNN classifier achieved a validation accuracy of 32.83% and a test accuracy of 33.38%. The KNN classifier's performance was significantly lower than the other two methods, likely due to its sensitivity to high-dimensional data and the curse of dimensionality.

Figure 2: Comparison of classification accuracies.

Method	**Test Accuracy**
Convolutional Neural Network (CNN)	86.77%
Bayesian Classifier with PCA features	52.17%
K-Nearest Neighbor (KNN)	33.38%

Figure 3: bar chart comparing the test accuracies of the three methods.



CNN outperformed both the Bayesian Classifier with PCA features and the KNN classifier in terms of classification accuracy. The CNN's robustness to variations in lighting, orientation, and background clutter, along with its ability to capture local patterns and hierarchical features, contributed to its superior performance. While the Bayesian Classifier with PCA features showed a moderate performance, the KNN classifier struggled with the high-dimensional data and achieved the lowest accuracy among the three methods.

Note: For additional comparison check Presentation

Conclusions:

We investigated the performance of three different classification methods on the CIFAR-10 dataset: a Convolutional Neural Network (CNN), a Bayesian Classifier with PCA features, and a K-Nearest Neighbor (KNN) classifier. Our key findings, insights, and limitations are as follows:

1. Convolutional Neural Network (CNN):

CNN demonstrated the best performance among the three methods, achieving a test accuracy of 86.77%. The model's ability to learn local patterns and hierarchical features contributed significantly to its success. However, training the CNN required a relatively large amount of computational resources and time. Furthermore, the model's complexity may make it susceptible to overfitting, especially with smaller datasets.

2. Bayesian Classifier with PCA features:

The Bayesian Classifier with PCA features provided a moderate level of performance, achieving a test accuracy of 61.20%. The use of PCA for feature extraction and dimensionality reduction was a valuable step in improving computational efficiency. However, the Gaussian distribution assumption might not always hold true for real-world data, which can limit the classifier's accuracy. Additionally, the Bayesian Classifier is sensitive to the quality of the estimated probability densities and may suffer from high bias if the training data is insufficient.

3. K-Nearest Neighbor (KNN) classifier:

The KNN classifier performed the poorest among the three methods, with a test accuracy of 33.38%. The classifier's performance was likely hindered by the high-dimensional data and the curse of dimensionality. The KNN classifier's sensitivity to the choice of distance metric and the value of k also posed challenges. One possible way to improve its performance would be to employ feature extraction techniques, such as PCA, to reduce dimensionality.

Throughout the project, we discovered that deep learning techniques, such as CNNs, are well-suited for image classification tasks. Despite their high computational requirements, their ability to learn complex features and adapt to variations in data proved valuable. On the other hand, simpler methods like Bayesian Classifiers and KNN classifiers can be useful in specific contexts but may not always perform well on complex datasets like CIFAR-10.

In conclusion, our findings emphasize the importance of selecting an appropriate classification method based on the problem's complexity and the available resources. The choice of method should also consider factors such as interpretability, computational requirements, and model complexity.

Source Code: https://github.com/sarthakchauhan0/CIFAR10-Analysis/tree/main