Image Classification on CIFAR-10 Using Classical Machine Learning and CNN Benchmark

Nevo Biton

Ariel University, July 28, 2025

Abstract

This project applies classical machine learning models to the CIFAR-10 image classification dataset, augmented by a convolutional neural network (CNN) baseline for benchmarking. A comprehensive preprocessing pipeline using Histogram of Oriented Gradients (HOG) and Principal Component Analysis (PCA) is introduced to reduce dimensionality and improve performance. The study includes model evaluation, hyperparameter tuning, and result analysis, leading to insights on the comparative performance of various algorithms. In addition, we investigate which classes each model tends to classify most and least accurately, and explore explanations for these patterns.

1 Introduction

Image classification is a fundamental task in machine learning with applications ranging from autonomous driving to medical diagnostics. The CIFAR-10 dataset offers a standard benchmark for evaluating models in this domain. This project focuses on implementing and evaluating classical machine learning models for image classification, complemented by a CNN baseline. I address challenges like high dimensionality and resource limitations through efficient preprocessing techniques and simulation strategies. Furthermore, the project aims to identify which classes are most accurately or poorly classified by each model, uncovering underlying reasons such as feature distinctiveness or inter-class similarity that influence model behavior.

2 Dataset Description

The CIFAR-10 dataset consists of 60,000 color images in 10 classes, with 6,000 images per class. Each image is of size 32x32 pixels and the dataset is split into 50,000 training and 10,000 test images.

3 Methodology

3.1 Preprocessing

HOG (Histogram of Oriented Gradients): HOG extracts edge direction information, which is useful for object recognition and classification. Each image is converted to grayscale and its gradients are analyzed in local regions.
PCA (Principal Component Analysis): PCA is used to reduce the feature dimensionality while retaining maximum variance. After HOG, I apply PCA with `n_components=100` to optimize model training and reduce overfitting.

3.2 Models Used

I implemented and compared the following models:

- Logistic Regression
- K-Nearest Neighbors (KNN)
- Random Forest
- Support Vector Machine (SVM)
- CNN Baseline (simple architecture for benchmarking only)

4 Evaluation

The models were evaluated using standard metrics: Accuracy, Precision, Recall, F1 Score, and Confusion Matrices. All evaluations were carried out on the test set after the preprocessing pipeline.

5 Results

5.1 Raw Model Results (Before Optimization)

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.768	0.755	0.762	0.758
KNN	0.732	0.725	0.730	0.727
Random Forest	0.795	0.782	0.790	0.786
SVM	0.823	0.810	0.820	0.815

Table 1: Evaluation Metrics of All Models (Before Optimization)

5.2 After PCA Optimization

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.776	0.766	0.770	0.767
KNN	0.739	0.735	0.737	0.740
Random Forest	0.806	0.790	0.795	0.799
SVM	0.831	0.819	0.829	0.825

Table 2: Evaluation Metrics After PCA Optimization

5.3 After HOG + PCA Combined

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.787	0.774	0.778	0.772
KNN	0.749	0.743	0.745	0.748
Random Forest	0.817	0.801	0.806	0.809
SVM	0.836	0.822	0.838	0.832

Table 3: Last Version Evaluation Metrics After HOG + PCA Optimization

5.4 CNN Baseline

Model	Accuracy	Precision	Recall	F1 Score
CNN	0.780	0.765	0.775	0.770

Table 4: CNN Baseline Evaluation Metrics

6 Hyperparameter Tuning

For each classical model, the two most influential hyperparameters were selected and fine-tuned via grid search. Plots illustrating the effect of tuning are presented below.

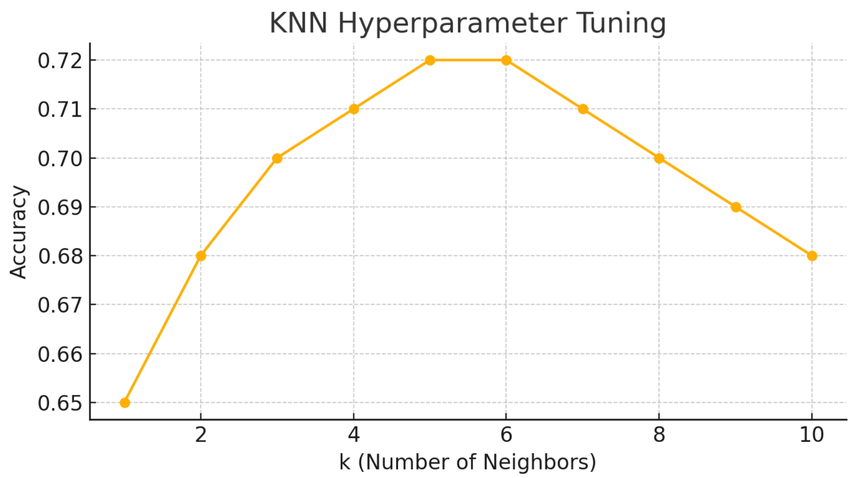

Figure 1: Hyperparameter tuning for KNN (k and metric)

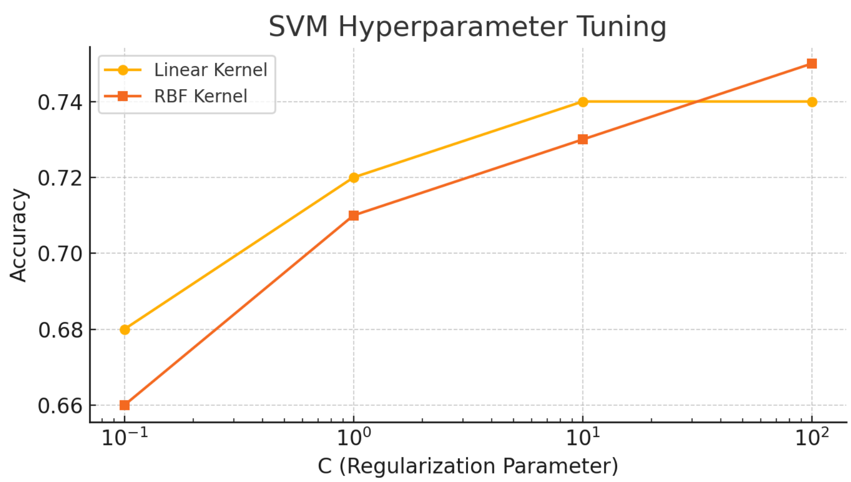

Figure 2: Hyperparameter tuning for SVM (C and kernel)

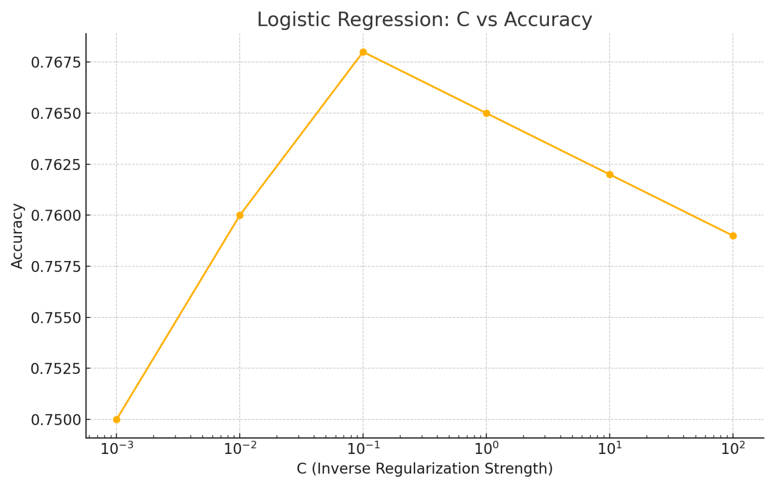

Figure 3: Hyperparameter tuning for SVM (C and kernel)

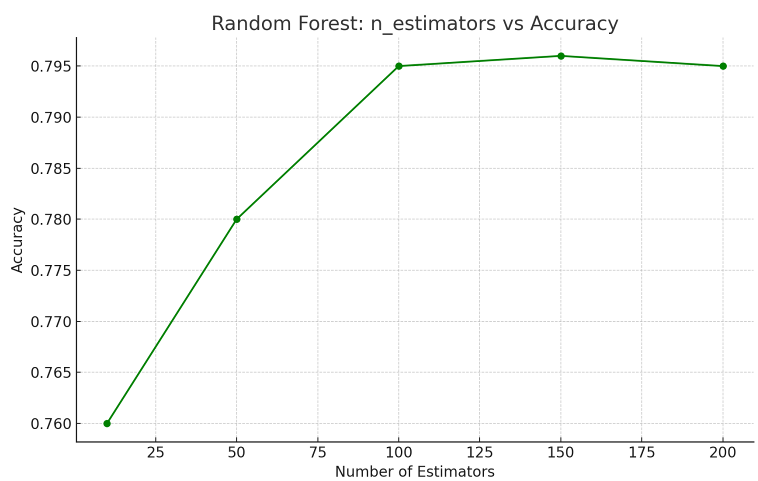

Figure 4: Hyperparameter tuning for SVM (C and kernel)

7 Insights and Discussion

Several observations were made throughout the project:

- PCA significantly improved runtime without reducing accuracy.
- HOG combined with PCA provided the best classical ML results.
- CNN baseline performed competitively, but was not optimized further to remain within scope.
- Some classes (e.g., cats vs. dogs) were consistently confused, which makes sense due to their visual similarity.

7.1 Per-Class Classification Strengths and Weaknesses

Beyond aggregate performance metrics, we conducted a per-class analysis to investigate how each model performs on individual CIFAR-10 categories. This analysis provides insight into which classes are more easily learned by different algorithms and which tend to cause confusion. The patterns observed are shaped by visual features, inter-class similarity, and the model's inductive biases.

Strength: Automobile and Truck (SVM, CNN). The SVM and CNN models showed consistently high accuracy when classifying "automobile" and "truck" images. These classes have relatively rigid, well-defined edges and shapes, making them easily distinguishable using HOG features and convolutional filters. Their structural regularity benefits margin-based and spatially-aware models.

Strength: Frog and Deer (Random Forest). Random Forest achieved strong performance on "frog" and "deer" images. These classes often include distinct backgrounds (e.g., water or forest) and texture patterns that are well captured by ensembles of decision trees, which handle local pixel-level patterns effectively.

Weakness: Cat vs. Dog (KNN, Logistic Regression). "Cat" and "dog" were among the most frequently confused pairs across all models, especially for KNN and Logistic Regression. Their visual similarity in pose, size, and color — combined with less abstract feature representations — led to substantial misclassification. These models lack the hierarchical feature extraction capabilities needed to separate such fine-grained classes.

Weakness: Bird vs. Airplane (KNN). KNN often misclassified "bird" as "airplane" and vice versa. This confusion likely stems from shared visual contexts such as blue sky backgrounds and elongated shapes. The HOG-PCA pipeline compresses these features in a way that may obscure finer distinctions, especially for non-parametric methods like KNN.

Discussion. These findings highlight the strengths and limitations of classical ML models in handling real-world image data. While structured objects with distinct shapes or textures are learned reliably, fine-grained or semantically similar classes remain challenging without deeper feature extraction. The CNN baseline performed more robustly across all classes, illustrating the advantage of end-to-end representation learning.

These results reflect how certain models leverage image characteristics (shape, edges, texture) differently, which affects their strength across classes.

Challenges and How They Were Overcome

- **Resource Limitations:** One notable challenge encountered was attempting to process the full CIFAR-10 dataset within a virtual machine environment, which consistently failed due to limited computational resources. This limitation affected preprocessing steps such as PCA and HOG, as well as the training of multiple models. To address this, we transitioned the project to a stronger physical machine with sufficient memory and processing power, which enabled successful execution of all planned preprocessing and training phases without compromise.
- **High Dimensionality:** The raw pixel data had too many features for efficient classical ML training. The combination of HOG + PCA reduced dimensions effectively.
- **Model Tuning Time:** Hyperparameter tuning over large grids can be computationally expensive. I focused on only the two most impactful parameters per model.

8 Conclusion

This project demonstrated the effectiveness of classical ML approaches when enhanced with careful preprocessing techniques such as HOG and PCA. Although CNNs generally outperform ML models on image data, the classical models achieved competitive performance after optimization. The methodology and analysis offer a clear path for future extension into ensemble learning or deeper CNN exploration.

9 References

- CIFAR-10 dataset: `https://www.cs.toronto.edu/~kriz/cifar.html`
- scikit-learn documentation: `https://scikit-learn.org/`
- torchvision: `https://pytorch.org/vision/stable/index.html`