CIFAR-10 Object Recognition using SVM and CNN

Big Data Coursework

Asmi Saluja 2311895

Swansea University
Prifysgol Abertawe

DECEMBER 15, 2025

Department of Computer Science Adran Gyfrifidureg

1. Introduction

Object recognition represents a fundamental challenge in computer vision with applications spanning autonomous vehicles, medical diagnostics, and content filtering systems . This study addresses multi-class image classification using a 10-class subset of the CIFAR dataset, comprising 10,000 training images and 1,000 testing images. Each 32×32 pixel RGB image depicts one of ten object categories: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, or truck. The low resolution combined with high intra-class variability (different poses, lighting conditions) and inter-class similarity (particularly among animal categories) creates significant classification challenges.

We implement and compare two different approaches throughout our study. First, a traditional machine learning approach combining Histogram of Oriented Gradients (HOG) feature extraction with Support Vector Machines (SVM) , enhanced by Principal Component Analysis (PCA) for dimensionality reduction. Second, a deep learning approach utilizing Convolutional Neural Network (CNN) that learns hierarchical representations directly from raw pixels. While the coursework specification references CIFAR-100 benchmarks (39.43% for 20 classes, 24.49% for 100 classes), our 10-class implementation operates in a different performance regime.

The SVM achieved 47.3% test accuracy while the CNN demonstrated superior performance at 55.30%, validating the effectiveness of learned representations over hand-crafted features. This report , presents the analysis of the dataset, application of the methods used, results including confusion matrix analysis.

2. Method

2.1 Dataset analysis

We start by loading the datasets and printing their shapes to understand the dataset
The CIFAR-10 dataset was provided as 4D NumPy arrays with dimensions (32, 32, 3, N), where N represents the number of samples. Initial preprocessing involved transposing the arrays to (N, 32, 32, 3) for compatibility with standard machine learning libraries. After feature extraction we standardize the dataset to ensure zero mean and unit variance across all features. Following this we perform PCA (Principal Component Analysis) to reduce dimensionality to reduce overfitting and decrease computational requirements.

For the CNN approach, pixel values were normalized to the range [0, 1] by dividing by 255, ensuring stable gradient descent during training. The dataset contained 10,000 training samples and 1,000 testing samples distributed across 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

2.2 Traditional Machine Learning Pipeline

Feature Extraction with HOG: HOG features capture local shape information by computing gradient orientations within spatial regions . Images were converted to grayscale to reduce dimensionality while preserving structural information. The HOG configuration employed pixels per cell =(3,3) and cells per block =(2,2) using which we convert the image to a hog image(edges visible). In our study instead of looping through the dataset as it requires more computational power and is time consuming we can change the image index to visualise one picture at a time(not the best solution but helps understand the dataset).

Standardization and PCA: Standard Scaler normalized HOG features to zero mean and unit variance, ensuring equal contribution to SVM distance calculations and preventing large-magnitude features from dominating the RBF kernel. PCA subsequently reduced dimensionality from 2,916 to 300 components.

2.3 Classification Algorithms

2.3.1 Support Vector Machine

SVM Classification: We start by creating the Support Vector Machine that uses an RBF kernel (separates image classes using curved boundaries instead of straight lines). C=10 controls how strict the model is, while gamma=0.001 creates broader decision regions preventing overfitting. Overall we train a non-linear Support Vector Machine with an RBF kernel on standardised image features in order to learn a decision boundary that separates the different image classes. We then ensure the labels are a one-dimensional integer array and make sure that the number of samples matches the number of labels to ensure that features and labels have matching lengths to avoid errors.
To sum it up we find the accuracy of the model and make a classification report of the model.
We also use confusion matrix for the model that shows the predicted label the actual label with the class names, makes it easier to read and understand.

2.3.2 Convolutional Neural Network Architecture

The CNN comprises of three convolutional blocks with progressively increasing filter depth (32-64-128), implementing hierarchical feature learning from low-level edges to high-level semantic features. Each convolutional block contains two Conv2D layers with 3×3 kernels, ReLU activation, and same padding, followed by Batch Normalization after each Conv2D layer to stabilize training and accelerate convergence, then MaxPooling2D with 2×2 pooling for spatial down sampling and translation invariance, and finally Dropout with a rate of 0.25 to prevent co-adaptation of neurons.

The first block with 32 filters extracts low-level features like edges and corners from the 32×32 inputs, reducing spatial dimensions to 16×16. The second block with 64 filters learns mid-level patterns such as textures and simple shapes, reducing to 8×8. The third block with 128 filters captures high-level semantic features and object parts, producing 4×4×128 feature maps.

The fully connected layers process the flattened feature vector of 2,048 dimensions through a Dense layer with 512 units using ReLU activation, Batch Normalization, and Dropout of 0.5, then another Dense layer with 256 units with the same configuration, and finally an output layer with 10 units and SoftMax activation that produces class probability distributions. After setting up the architecture we train the CNN model on images and labels. The model sees the dataset 20 times as (epochs = 20), lesser epochs takes less time . We set aside 10 % of the data to check performance during training and the model updates weights after every 32 samples.

We then evaluate the test loss and the test accuracy based on the testing image and testing label. We also predict the results using a confusion matrix and classification report containing precision, recall, f1 -score, support.

3. Results

3.1 Evaluation Metrics

Model performance was assessed using:

- Accuracy: Percentage of correctly classified samples.
- Classification report containing precision, recall, f1 -score, support.
- Confusion Matrix: Visualization of prediction patterns and misclassification trends.

3.2 Support Vector Machine Performance

The SVM classifier achieved an accuracy of 47.30% on the test set. The classification report shows varying performance across different classes such as horse with 70% precision (best) to bird with 31 % (worst) precision.

The confusion matrix reveals that the SVM struggles most with distinguishing between visually similar animal categories (bird, cat, deer, dog). This is expected as HOG features, while effective for capturing shape information, may not adequately capture the subtle differences between similar object categories at low resolution.

3.3 Convolutional Neural Network Performance

The CNN achieved a test accuracy of 55.30% , representing improvement over the SVM approach. The classification report shows varying performance across different classes such as ship with 83% precision (best) to bird with 31 % (worst) precision.

The confusion matrix for the CNN shows more balanced performance across classes compared to the SVM. The network appears to distinguish better between similar categories, likely due to its ability to learn complex feature hierarchies automatically.

3.4 Comparative analysis

Measure	**SVM**	**CNN**
Accuracy	47.3%	55.30%
Feature type	Handcrafted	Learned
Training time	Moderate	Slower

The CNN outperforms the SVM. However, the SVM approach has advantages in interpretability and training speed. HOG features are well-understood and visualizable, while CNN learned features are more abstract.
This study compared traditional machine learning (SVM with HOG and PCA) against deep learning (CNN) for CIFAR-10 object recognition. The SVM achieved 47.3% through tuned hyperparameters and dimensionality reduction, while the CNN reached 55.30% via hierarchical feature learning. Both methods substantially exceed random baseline (10%).

4. Conclusion

Throughout this report we compared two machine learning approaches for object recognition on the CIFAR 10 dataset:

1. SVM with Hog features – A traditional approach using handcrafted features, requires less computational resources, works well with limited data and has a faster training time.

2. Convolutional Neural Network – A deep learning approach with automatic feature learning, higher accuracy through hierarchical feature learning, that works better with more data, automatically learns relevant features from raw pixels and better captures spatial relationships in images.

The SVM achieved 47.3% through tuned hyperparameters and dimensionality reduction, while the CNN reached 55.30% via hierarchical feature learning. Both methods substantially exceed random baseline (10%).Both methods successfully classified the images in the dataset though CNN demonstrated superior accuracy but higher cost of computational requirements.

4.1 Critical Analysis

This study reveals some limitations that affect the performance of our classification systems such as usage of 5 epochs instead of more due to computational restrictions that leads to underfitting. The training loss continued to decrease steadily, which is clear evidence the network had not converged and could benefit from additional training.

No data augmentation was applied, which is standard practice for image classification. The CIFAR-10 training set contains only 1,000 images per class (10,000 total), which is relatively small for training deep networks. Data augmentation techniques would particularly help with classes showing high variance in appearance, such as cats and dogs. The truck classification failure might be partially due to insufficient exposure to truck variations during training.

4.2 Improvement Strategies

Extended training with callbacks: Train the CNN for 50-100 epochs with early stopping (patience=10 epochs) and ReduceLROnPlateau (factor=0.5, patience=5). This would allow proper convergence and likely resolve the truck classification failure while preventing overfitting through monitoring validation loss.

Data augmentation: Implement Image Data Generator with horizontal flips, rotation range, width/height shifts, and zoom range. This would effectively triple the training data diversity and improve generalization, particularly for underperforming classes.