1. Dataset Analysis

1.1 Overview of Training Dataset

The training dataset contains 50,000 images with 10 kinds of labels, each representing a different category of objects as listed:

Label	Count	Category
0	5010	Plane
1	5012	Car
2	5038	Bird
3	5007	Cat
4	4995	Deer
5	4993	Dog
6	4955	Frog
7	5000	Horse
8	5020	Ship
9	4970	Truck

1.2 Visualizations

We present the first picture in each category here, indicating a brief understanding of the image content.

2. Classifier Exploration

2.1 Methodology

For each classifier, we used an 80:20 split of the training dataset into training and validation subsets. Data preprocessing included converting images into grayscale and RGB formats, followed by applying Principal Component Analysis (PCA) for dimensionality reduction. The implementation details, results, and optimization strategies for the three classifiers are presented in Sections 2.2 to 2.4.

2.2 Classifier 1: Support Vector Machine (SVM)

Description:

Support Vector Machine (SVM) is a supervised learning algorithm for classification tasks. It works by finding a hyperplane that separates classes in a high-dimensional feature space. For non-linear data, SVM uses kernel functions (e.g., Radial Basis Function) to map data into a higher-dimensional space for better separability.

Performance:

1. Grayscale Data (PCA n=100):
    a. Accuracy: 48%
    b. Macro F1-Score: 48%
    c. Best-performing classes: 1, 7, 8 (F1-scores: 0.57, 0.56, 0.57)
    d. Challenging classes: 3, 4 (F1-scores: 0.34, 0.39)
2. RGB Data (PCA n=100):
    a. Accuracy: 54%
    b. Macro F1-Score: 54%
    c. Best-performing classes: 1, 7, 8 (F1-scores: 0.61, 0.62, 0.66)
    d. Challenging classes: 3, 4 (F1-scores: 0.39, 0.44)

Optimization

Hyperparameters: The model was optimized with a grid search, tuning the penalty parameter C=10, kernel rbf, and γ=0.001.

Dimensionality Reduction: PCA was applied to reduce dimensionality while preserving key features.

Observations

The SVM classifier performed better on RGB data, leveraging richer feature information. PCA effectively reduced dimensionality without significant loss of accuracy.

2.3 Classifier 2: K-Nearest Neighbors (KNN)

Description

K-Nearest Neighbors (KNN) is an instance-based learning algorithm that classifies data points based on the majority class of their nearest neighbors in feature space.

Performance

1. Grayscale Data (PCA n=100):
    a. Accuracy: 35%
    b. Macro F1-Score: 34.7%
    c. Best-performing classes: 9, 1, 7 (Precision: 63%, 59.5%, 57.3%)
    d. Challenging classes: 4, 2, 6 (Precision: 23.4%, 28.3%, 30.3%)

Optimization

Hyperparameters: Grid search was used to tune the number of neighbors (n=5), distance metric (minkowski), and weights (distance).
Dimensionality Reduction: PCA reduced data dimensions while retaining 90% of the variance.

Observations

Despite hyperparameter tuning and PCA adjustments, accuracy remained around 35%. KNN struggled with high-dimensional data and performed less effectively compared to other classifiers.

2.4 Classifier 3: Decision Tree

Description

The Decision Tree classifier splits data into subsets based on feature values, maximizing information gain at each split. It is interpretable and effective for smaller datasets but can overfit without proper constraints.

Performance

1. RGB Data (PCA n=100):
    a. Accuracy: 29.76%
    b. Macro F1-Score: 29%
    c. Best-performing classes: 8, 0 (Precision: 41%, 40%)
    d. Challenging classes: All other classes
2. RGB Data (PCA n=300):
    a. Accuracy: 30.20%
    b. Macro F1-Score: 30%
    c. Best-performing classes: 0, 8 (Precision: 42%, 40%)
    d. Challenging classes: All other classes

Optimization

Hyperparameters: Grid search optimized max depth (max_depth=10), minimum samples per split (min_samples_split=5), and minimum samples per leaf (min_samples_leaf=10).
Dimensionality Reduction: PCA reduced dimensionality from 3072 features to fewer components (e.g., n=100, 300).

Random Forest

By creating random forest, the performances increased significantly from 30% to 41%:

Performance of Random Forest

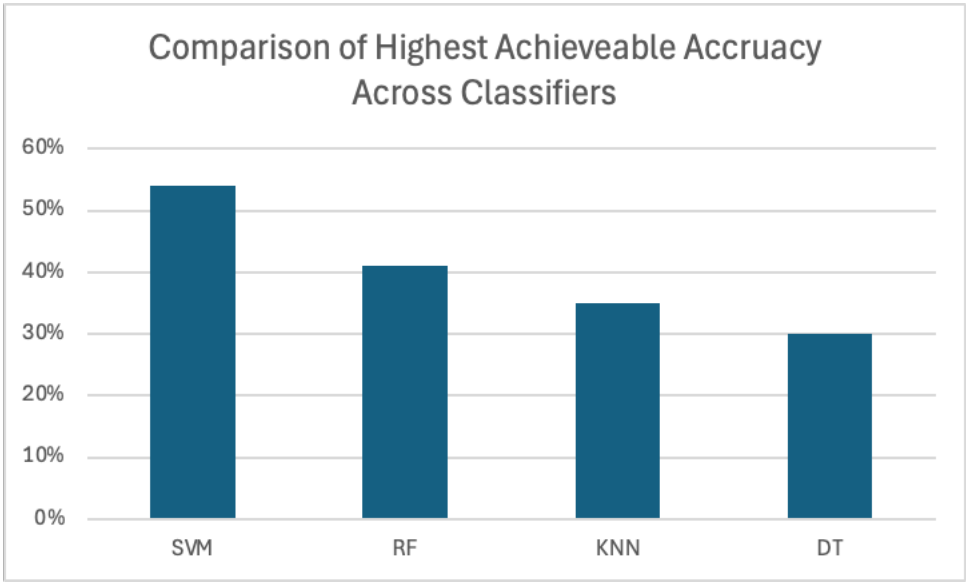
a. Accuracy: 41.43%
b. Macro F1-Score: 41%
c. Best-performing classes: 7 (Precision: 52%)
d. Challenging classes: All other classes

Observations

Hyperparameter tuning resulted in minimal accuracy improvement. Ensemble learning with Random Forest significantly improved accuracy from 30% to 41%, demonstrating its effectiveness over standalone decision trees.

2.5 Comparison and Analysis

2.5.1 Summary of Classifier Performance



SVM = Support Vector Machine
RF = Random Forest (Ensemble Learning Technique based on Decision Tree)
KNN = K-Nearest Neighbours
DT = Decision Tree

The classification accuracy results indicate that SVM performed the best with an accuracy of 54%, followed by KNN at 35% and Decision Tree (DT) at 30%.

2.5.2 Analysis

SVM's superior performance suggests that the dataset likely has well-defined class boundaries that SVM's hyperplane-based approach effectively captures. This result highlights SVM's strength in handling high-dimensional data and separating complex patterns.

KNN, with a significantly lower accuracy of 35%, might have struggled due to its reliance on proximity-based decisions, which can be sensitive to noisy or overlapping data. The dataset may have a structure that makes it difficult for KNN's majority-vote mechanism to differentiate classes effectively, particularly if the data distribution is uneven or lacks clear clusters.

DT, with the lowest accuracy of 30%, indicates its limitation in capturing complex patterns in the dataset. Decision Trees are prone to overfitting, especially on smaller datasets or those with noisy or imbalanced features. This simplicity, while useful for interpretability, may have contributed to its relatively poor performance compared to the other classifiers.

3. Final Solution Description

3.1 Final Pipeline Overview

Data Splitting:

- The dataset was split into training (80%) and validation (20%) sets to balance model training and evaluation effectively.

Preprocessing and Feature Representation:

1. Grayscale Images:
    - Images were resized to 32×32 and converted to grayscale. Each image was flattened into a vector of 1024 features, creating a uniform representation for the SVM.
    - Normalization was applied using StandardScaler to standardize feature ranges, ensuring uniform contributions to the model.
2. RGB Images:
    - A second approach used RGB images to leverage additional color channel information. Each 32×32 RGB image was flattened into a vector of 3072 features.
    - These features were also normalized to maintain consistency across input variables.

Dimensionality Reduction with PCA:

- Grayscale Data:
    - PCA reduced the dimensionality from 1024 features to nnn principal components, tested with n=100n=100n=100 and n=300n=300n=300.
    - PCA retained the essential variance, simplifying the feature space and improving computational efficiency.
    - Observations:
        - PCA with n=100: Accuracy of 48%, Macro F1-Score of 48%.
        - PCA with n=300: Accuracy of 47%, Macro F1-Score of 47%. The negligible improvement showed that 100 components were sufficient to capture relevant patterns.
- RGB Data:
    - PCA reduced the dimensionality of the 3072-feature RGB data to 100 components, significantly reducing noise and computational costs while retaining key variance.

Classifier Selection:

- Support Vector Machine (SVM):
    - An SVM model with a linear kernel was initially implemented, achieving limited accuracy (~31%) on grayscale images.
    - After recognizing the limitations of linear kernels in handling non-linear class boundaries, the SVM was optimized with a Radial Basis Function (RBF) kernel. Tuned parameters included:
        - Kernel: RBF for non-linear decision boundaries.
        - Regularization (C): Set to 10 to balance margin maximization and misclassification penalties.

- Gamma: Set to 0.001 to control the influence of individual data points.

3.2 Highlights

SVM Model with Improvements:

1. Grayscale Data (SVM with PCA):
   - PCA with n=100:
     - Accuracy: 48%, Macro F1-Score: 48%.
     - Best-performing classes: 1, 7, and 8 (F1-scores: 0.57, 0.56, and 0.57).
     - Challenging classes: 3 and 4 (F1-scores: 0.34 and 0.39).
   - PCA with n=300:
     - Accuracy: 47%, Macro F1-Score: 47%.
     - Similar class-level performance with minor variations in recall and F1-scores.
2. RGB Data (Final SVM with PCA and Hyperparameters):
   - Using the optimized RBF kernel and PCA (n=100), the final model leveraged the richer feature set of RGB images.
   - Validation Results:
     - Overall Accuracy: 54%, Macro F1-Score: 54%.
     - Best-performing classes remained consistent (1, 7, and 8), with F1-scores around 0.61–0.66.
     - Challenging classes (e.g., 3 and 4) showed modest improvement compared to the grayscale model, with F1-scores of 0.39 and 0.44.
3. Observations:
   - The additional color channel information in RGB images provided subtle but noticeable performance gains compared to grayscale data.

Challenges and Solutions:

1. Grid Search Complexity:
   - Hyperparameter tuning through exhaustive grid search was computationally expensive, taking ~16 hours. This was mitigated by narrowing the search range and using initial parameter estimates ($C = 10, \gamma = 0.001$).
2. Limited Class-Level Improvement:
   - While overall metrics improved, some classes remained challenging due to overlaps in feature space. Future iterations could address this with advanced techniques like oversampling or weighted loss functions.

3.3 Final Deciding Model:

- The RGB-based SVM model with RBF kernel, PCA (n=100n=100n=100), and tuned hyperparameters was selected as the final solution. This model balanced computational efficiency with accuracy, achieving 54% validation accuracy and balanced F1-scores across classes.