# 2ND MANDATORY ASSIGNMENT

# SUPPORT VECTOR MACHINES

Georgios Lazaridis

AEM: 4419

Aristotle University of Thessaloniki

December 2025

# 1 Introduction and Implementation

This project implements a Support Vector Machine (SVM) to solve the Multi-Class Classification problem on the **CIFAR-10** dataset. The goal is to compare the effectiveness of different SVM kernels (Linear vs. RBF) and evaluate their performance against a Feedforward Neural Network (MLP) baseline.

## 1.1 Task Definition and Data Pre-processing

Due to the high dimensionality of the CIFAR-10 images ($32 \times 32 \times 3$), specific pre-processing steps were required to make SVM training feasible:

- **Dataset Subset:** A stratified subset of **5,000 images** was used for training to manage computational load while maintaining class distribution.
- **Normalization:** Pixel values were flattened into vectors and standardized (StandardScaler) to have a mean of 0 and a standard deviation of 1.
- **Feature Extraction (PCA):** Principal Component Analysis (PCA) was applied to reduce dimensionality while retaining **90% of the variance**. This reduced the feature space from 3,072 to fewer than 100 components, significantly accelerating the SVM training process.

## 1.2 Algorithms Implemented

1. **SVM (RBF Kernel):** Utilizes the Radial Basis Function to map inputs into high-dimensional space, allowing for non-linear separation.
2. **SVM (Linear Kernel):** A baseline model attempting to separate classes with linear hyperplanes.
3. **MLP (Comparison):** A Multi-Layer Perceptron with one hidden layer (256 neurons) used as a benchmark for performance and training time.

# 2 Experimental Results

## 2.1 Performance Comparison Table

The experiments demonstrated a clear superiority of the non-linear kernel (RBF) over the linear approach. The Linear SVM failed to converge efficiently, resulting in excessive training times and poor accuracy.

Table 1: Experimental Results on CIFAR-10 (Test Set)

| **Model Architecture** | **Training Time (s)** | **Test Accuracy** | **Notes** |
|---|---|---|---|
| **SVM (RBF Kernel)** | $\sim 7.73$ | **43.91%** | Best Accuracy |
| MLP (1 Hidden Layer) | $\sim$ **9.08** | 40.03% | Fastest Training |
| SVM (Linear Kernel) | $\sim 433.96$ | 35.48% | Failed to Converge |

## 2.2 Visualizations


(a) Confusion Matrix - SVM (RBF) (b) Confusion Matrix - SVM (Linear)

Figure 1: Confusion Matrices comparing RBF and Linear Kernels.

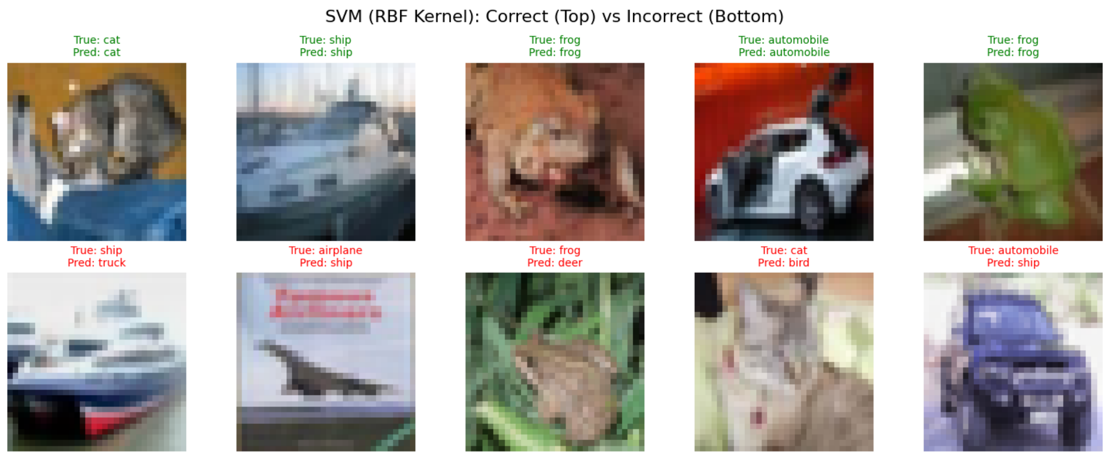


Figure 2: SVM (RBF): Characteristic examples of Correct (Green) vs. Incorrect (Red) classification.

# 3 Analysis and Discussion

## 3.1 SVM Kernel Comparison (Linear vs. RBF)

The comparison revealed a fundamental limitation of linear models in image recognition:

- The **Linear SVM** performed poorly (35.48%) and required over 400 seconds to train. This indicates that the raw pixel data (even after PCA) is not linearly separable. The algorithm struggled to find a hyperplane, resulting in slow convergence.
- The **RBF SVM** achieved the highest accuracy (43.91%) and trained much faster. The RBF kernel successfully handled the non-linear relationships between pixels by projecting them into a higher-dimensional feature space.

### 3.2 Comparison with Neural Networks (MLP)

The MLP (with one hidden layer) provided a competitive accuracy of 40.03% and was extremely fast to train (approx. 9 seconds).

- While the MLP was faster, the **SVM (RBF)** slightly outperformed it in accuracy on this specific subset.
- This suggests that for smaller datasets or reduced feature spaces (PCA), SVMs can be very effective and sometimes more stable than simple neural networks, which may require more data to generalize well.

### 3.3 Conclusion

The study confirms that non-linear methods are essential for image classification tasks. The combination of **PCA** for dimensionality reduction and **SVM with RBF kernel** proved to be a robust approach, outperforming the linear baseline and rivaling the simple MLP architecture.

## 4 Code Implementation

- **Libraries & Frameworks:** The implementation relies on the following Python libraries:
  - **Scikit-Learn:** Used for the core machine learning models (`SVC` for SVMs, `MLPClassifier` for the Neural Network), dimensionality reduction (`PCA`), and data scaling (`StandardScaler`).
  - **TensorFlow/Keras:** Utilized exclusively for efficiently loading the CIFAR-10 dataset (`cifar10.load_data`).
  - **Joblib:** Used for model serialization, allowing large models (like the RBF SVM) to be saved and loaded from the disk.
  - **Matplotlib & Seaborn:** Used to generate the Confusion Matrices and visual examples of classifications.
- **Efficiency:** The codebase utilizes a **Load or Fit** logic, implemented in `utils.py`. It checks for saved model files (e.g., `svm_rbf_cifar10.pkl`) in the `models/` directory to avoid retraining heavy models like the RBF SVM, thus optimizing experimental time.
- **Documentation:** The scripts automatically generate visual evidence, including Confusion Matrices and characteristic examples of correct/incorrect classification, fully documenting the behavior of both Linear and Non-Linear kernels for the purposes of the report.

## 5 GitHub Repository

The complete source code and necessary configuration files for reproducing the experiments are publicly available on the GitHub repository:

- `https://github.com/lazoulios/image-classification-svm-pca`
- **Contents:** The repository includes the `main.py` (training execution), `visualize.py` (graph generation), and `utils.py` (helper functions) scripts, the `requirements.txt` dependency list, and the `README.md` setup instructions.