Faiza Abdullah

Lab 05

ITAI 1378 Comp Vision-Artificial Intelligence

Professor: Anna Devarakonda

Decoding Images:

A Reflective Journey into SVM-Based Classification with CIFAR-10

INTRODUCTION:

In this reflective journal, I document my learning and insights from the lab on image classification using the Support Vector Machine (SVM) algorithm with the CIFAR-10 dataset. The CIFAR-10 dataset, consisting of 60,000 32x32 color images across 10 classes, is a widely used benchmark in machine learning for image classification tasks. This lab aimed to implement a basic machine learning algorithm, SVM, to classify these images. This journal reflects on my understanding of the SVM algorithm, the data preparation steps, model training, evaluation, and the challenges faced during the process.

REFLECTION JOURNEY:

Understanding SVM and Its Application in Image Classification

Support Vector Machine (SVM) is a supervised machine learning algorithm that can be used for both classification and regression tasks. In image classification, SVM works by finding the optimal hyperplane that separates the data into different classes with the maximum margin. The algorithm is particularly effective in high-dimensional spaces, making it suitable for image data, where each pixel can be considered a feature[1].

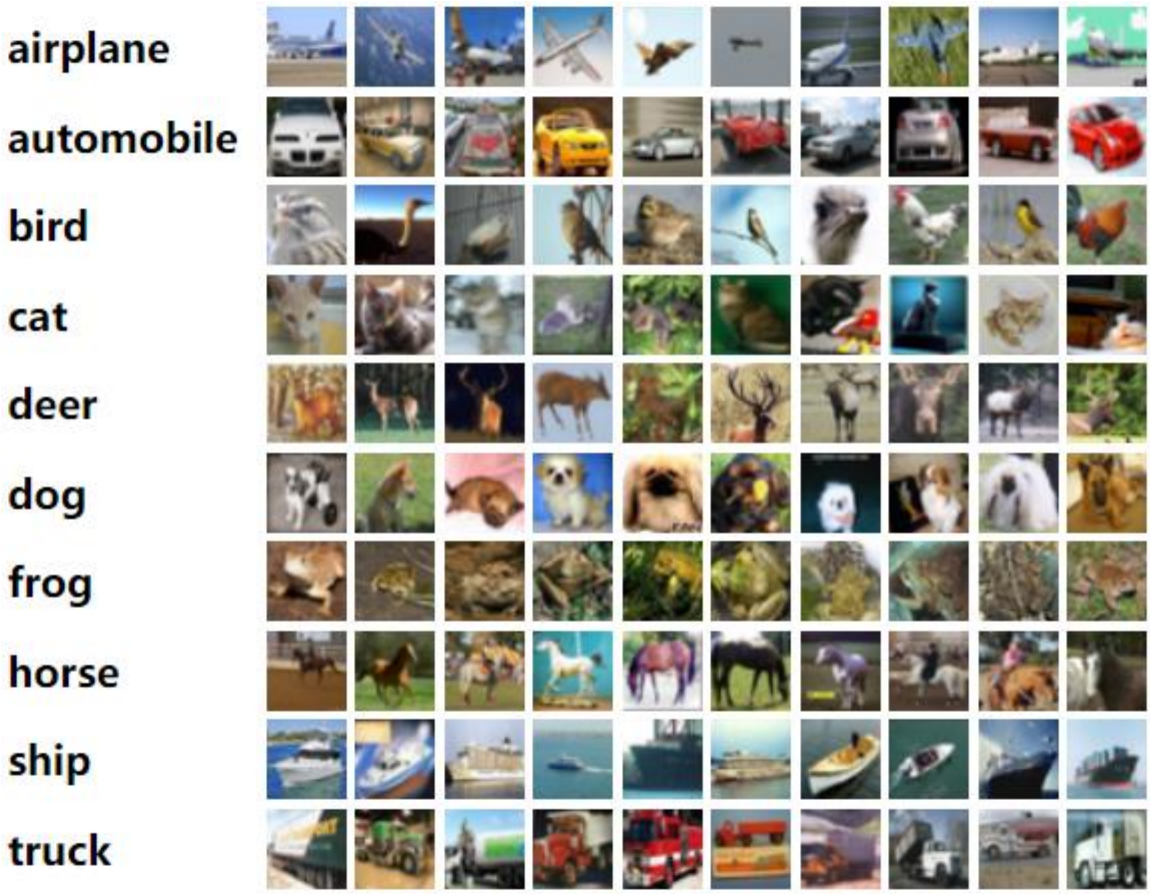
During the lab, I learned that SVM can handle both linear and non-linear classification through the use of different kernel functions. For this lab, we used a linear kernel, which is

[1] https://www.geeksforgeeks.org/support-vector-machine-algorithm/

simpler and computationally less expensive, especially given the size of the CIFAR-10 dataset. However, an alternative approach using the Radial Basis Function (RBF) kernel was also explored. The RBF kernel, while more effective in capturing non-linear relationships, introduces higher computational complexity and may not be feasible for large-scale datasets.

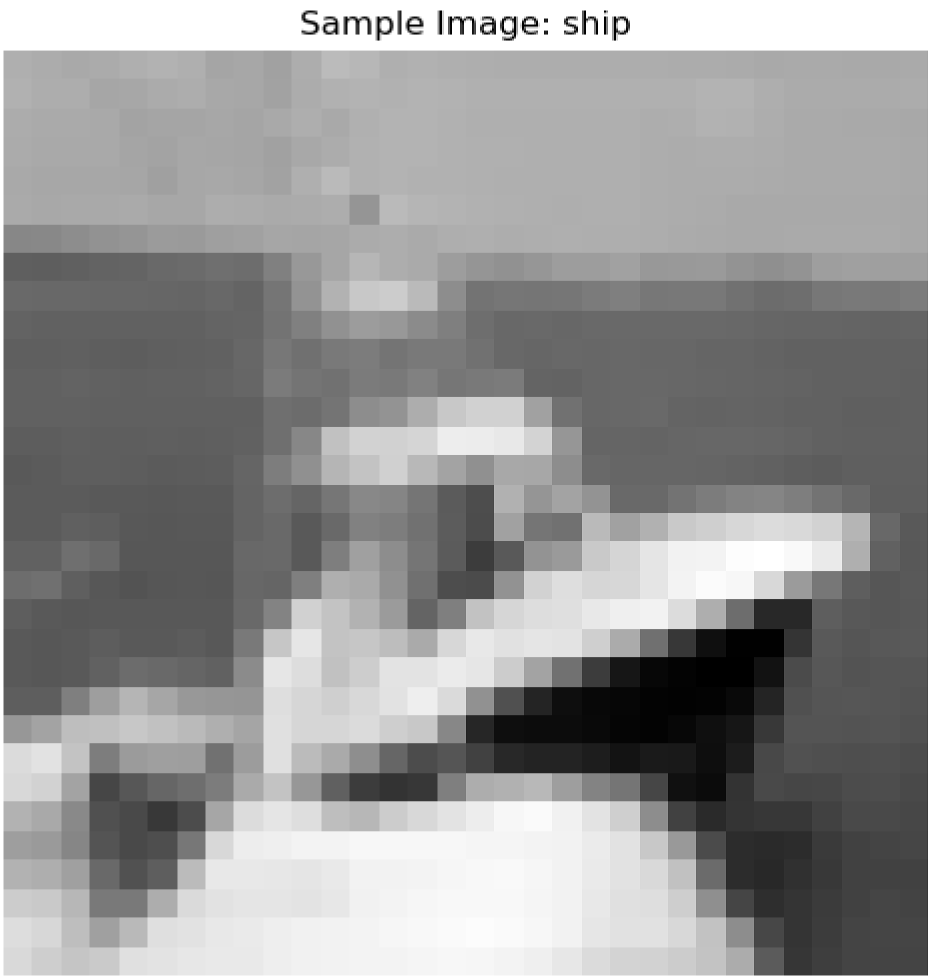
Data Preparation Steps

For the CIFAR-10 dataset, this involved loading the dataset, visualizing some images, and converting the images to grayscale. Converting the images to grayscale reduced the dimensionality of the data, making it easier to process without losing significant information. Below is an example of the CIFAR-10 images before preprocessing[2]:



[2] https://paperswithcode.com/dataset/cifar-10

After converting the images to grayscale, we flattened them into 1D arrays, which is necessary for SVM, as it requires the input data to be in a 2D format (samples x features). Below is a visualization of grayscale images:




```
Training set size: (15000, 1024)
Testing set size: (3000, 1024)
```

Additionally, standard scaling techniques were employed to normalize the data, ensuring optimal performance of the SVM classifier. The dataset was then split into training and testing sets, with 80% of the data used for training and 20% for testing. This split is crucial for evaluating the model's performance on unseen data.

Model Training and Evaluation

Training the SVM model involved fitting the model to the training data. Given the size of the dataset, training the SVM took a considerable amount of time, even with a linear kernel. The computational cost of training SVM on large datasets became a notable challenge. To mitigate this, a strategic sampling approach was employed, where carefully selected subsets of the dataset maintained a representative class distribution while improving training efficiency.

An SVM classifier with a linear kernel is trained on the flattened training data (X_train_flat, y_train_subset.ravel()), the model's performance is evaluated using accuracy and a classification report, which includes precision, recall, and F1-score for each class, as well as the overall accuracy.

```
# Train an SVM classifier
model = SVC(kernel='linear')
model.fit(X_train_flat, y_train_subset.ravel())

# Predict on the test set
y_pred = model.predict(X_test_flat)

# Evaluate the model
print("Accuracy:", accuracy_score(y_test_subset, y_pred))
print("Classification Report:\n", classification_report(y_test_subset, y_pred, target_names=chosen_classes))
```

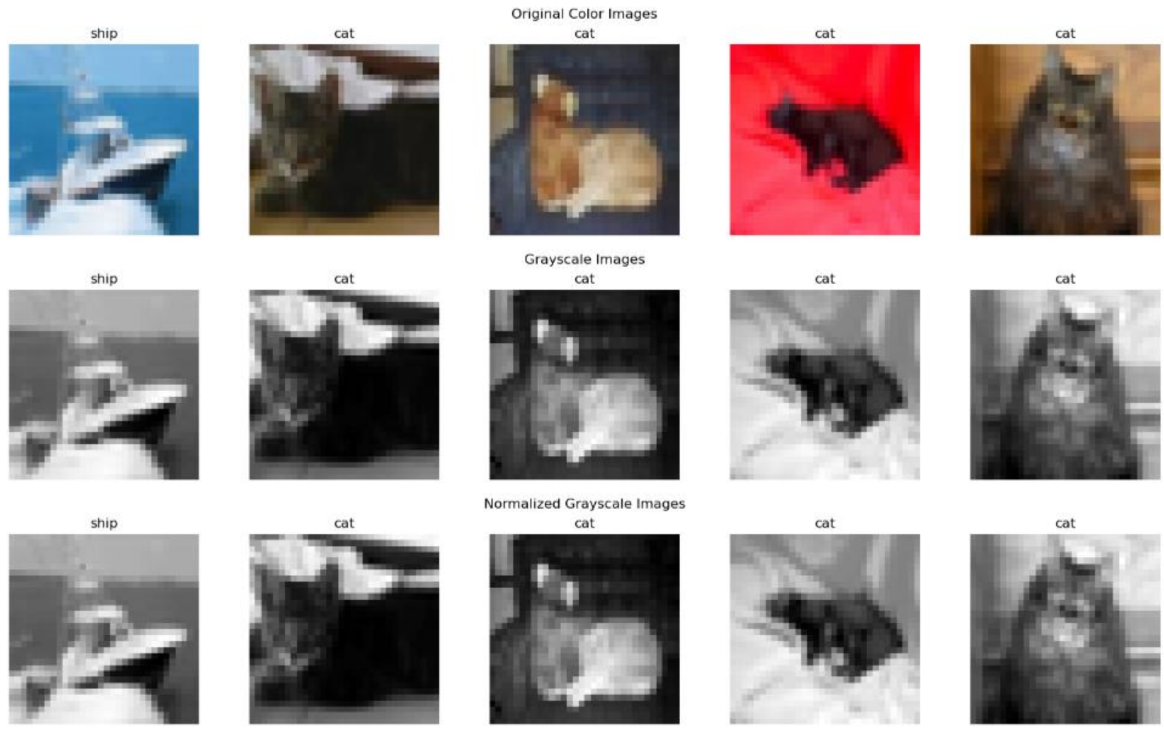
```
Accuracy: 0.547
Classification Report:
              precision    recall  f1-score   support

         cat       0.48      0.48      0.48      1000
         dog       0.49      0.48      0.49      1000
        ship       0.66      0.68      0.67      1000

    accuracy                           0.55      3000
   macro avg       0.54      0.55      0.55      3000
weighted avg       0.54      0.55      0.55      3000
```

The model achieves an overall accuracy of 54.7%, which means it correctly classifies approximately 55% of the test samples. The report shows the performance metrics for each class ("cat," "dog," and "ship"). The "ship" class has the highest precision, recall, and F1-score, indicating that the model performs better at recognizing ships compared to cats and dogs.

We also visualized some predictions alongside their true labels to analyze misclassifications. Certain image categories were classified more accurately than others, prompting an investigation into feature importance and kernel selection.



Challenges and Insights

One of the main challenges faced during the lab was the computational cost of training the SVM on the entire dataset. To address this, we could have used a subset of the data or employed more efficient algorithms like Stochastic Gradient Descent (SGD) for training. The RBF kernel was considered to improve performance but came at a significant computational cost.

Additionally, model performance varied across different image categories, with some classes being more easily distinguishable than others.

The lab provided valuable insights into the trade-offs between model complexity and computational efficiency. It also underscored the importance of choosing the right algorithm for the task at hand. Future implementations might benefit from exploring alternative approaches such as hierarchical classification or ensemble methods to enhance scalability and accuracy.

Responses to Lab Questions

What is the purpose of converting images to grayscale in this lab?

Converting images to grayscale reduces the dimensionality of the data, making it easier to process and reducing computational costs. It also simplifies the model by reducing the number of features, which can be beneficial when using simpler algorithms like SVM.

Why is it important to split the dataset into training and testing sets?

Splitting the dataset into training and testing sets is crucial for evaluating the model's performance on unseen data. The training set is used to train the model, while the testing set is used to assess how well the model generalizes to new data.

What are the limitations of using a linear SVM for image classification?

Linear SVMs assume that the data is linearly separable, which is often not the case with complex image data. This can lead to lower accuracy, as seen in this lab. Non-linear SVMs, such as those using the RBF kernel, or other algorithms like CNNs may be more suitable for such tasks.

Critical Analysis & Referencing

The lab provided a hands-on introduction to image classification using SVM, but it also highlighted the limitations of using simple models for complex tasks. While SVM is a powerful algorithm, its computational cost and inability to capture non-linear relationships in image classification make it less favorable than deep learning-based approaches like CNNs.

CONCLUSION:

This lab was an excellent introduction to the challenges of image classification and the application of SVM in this context. While the model's performance was not exceptional, the lab provided valuable insights into the data preparation process, model training, and evaluation. It also highlighted the importance of choosing the right algorithm for the task and the trade-offs involved in model selection. This experience has deepened my understanding of machine learning and has motivated me to explore more advanced techniques in the future.

CITATIONS:

CIFAR-10 Dataset: https://www.cs.toronto.edu/~kriz/cifar.html

https://eagleonline.hccs.edu/courses/278598/files/70587246?module_item_id=18927450

https://www.geeksforgeeks.org/support-vector-machine-algorithm/

https://paperswithcode.com/dataset/cifar-10