Scene Classification with Bag of Visual Words and Spatial Pyramids

Contents

1. Introduction

Scene classification is a fundamental problem in computer vision, where the goal is to assign a semantic category to an image based on its visual content. Unlike object recognition, scene recognition relies more heavily on global appearance and texture patterns rather than the presence of a single dominant object. This project implements a classical scene classification pipeline using local feature descriptors, bag-of-visual-words representations, and supervised learning methods.

The system is built around three main ideas. First, local image descriptors (SIFT) are used to capture invariant visual patterns. Second, these descriptors are quantized into a visual vocabulary using k-means clustering, allowing images to be represented as histograms of visual word occurrences. Third, classification is performed using several models, including a nearest neighbor baseline, a linear support vector machine, and a chi-square kernel SVM combined with a spatial pyramid representation. The spatial pyramid is used to incorporate coarse spatial information, which is otherwise lost in standard bag-of-words models.

2. Methodology

2.1 Dataset

The experiments in this project use a standard scene classification dataset consisting of color images grouped into multiple semantic scene categories, such as Bedroom, Coast, Forest, Highway, and Street. The dataset is organized into separate training and test splits, with images stored in class-specific directories.

The training set contains 100 images per class, while the test set contains a variable number of images per class. All images are used as provided, without manual annotation beyond their scene category labels inferred from the directory structure.

Only the training split is used to learn the visual vocabulary. Both training and test splits are encoded using the learned vocabulary and evaluated using the same feature representation and classifiers.

2.2 Feature Extraction

For each image in the dataset, SIFT keypoints and descriptors are extracted using OpenCV's SIFT implementation. Images are converted to grayscale prior to feature extraction. The extracted descriptors are stored per image as NumPy files, along with their spatial locations and image dimensions. This intermediate representation allows feature extraction to be decoupled from later stages of the pipeline.

Only the training set descriptors are used for learning the visual vocabulary, ensuring that no information from the test set leaks into the model during training.

2.3 Visual Vocabulary Construction

A visual vocabulary is constructed by clustering a random subset of SIFT descriptors from the training set using k-means clustering. In this project, the vocabulary size is fixed to **K = 400**, based on empirical evaluation

during development. The resulting cluster centers represent visual words and are stored for reuse during image encoding.

2.4 Image Encoding

Each image is encoded as a histogram of visual word occurrences by assigning each SIFT descriptor to its nearest visual word.

Two encoding schemes are used:

Plain Bag of Visual Words

A single global histogram is computed for each image. The histogram is L1-normalized to account for varying numbers of descriptors per image.

Spatial Pyramid Bag of Visual Words

To incorporate spatial information, a two-level spatial pyramid is used:

- Level 0: a single global histogram (1×1)
- Level 1: a 2×2 grid over the image

Histograms from all regions are concatenated, resulting in a feature vector of length **5 × K**. The final feature vector is L1-normalized.

3. Classification Models

Three classification approaches are evaluated.

3.1 Nearest Neighbor Classifier

A nearest neighbor classifier is used as a baseline. Images are compared using Euclidean distance between their bag-of-words histograms. This model provides a simple reference point but does not learn a discriminative decision boundary.

3.2 Linear Support Vector Machine

A linear SVM is trained using the plain bag-of-words representation. This model learns a discriminative classifier while maintaining computational efficiency. It serves as a stronger baseline than nearest neighbor while still operating on global histograms.

3.3 Chi-Square Kernel SVM with Spatial Pyramid

The final model uses a chi-square kernel SVM trained on spatial pyramid bag-of-words features. The chi-square kernel is well suited for histogram-based representations, as it emphasizes relative differences

between bins rather than absolute magnitudes. Kernel values are precomputed for both training and test data. This model represents the most expressive configuration evaluated in the project.

4. Experimental Results

4.1 Classification Accuracy

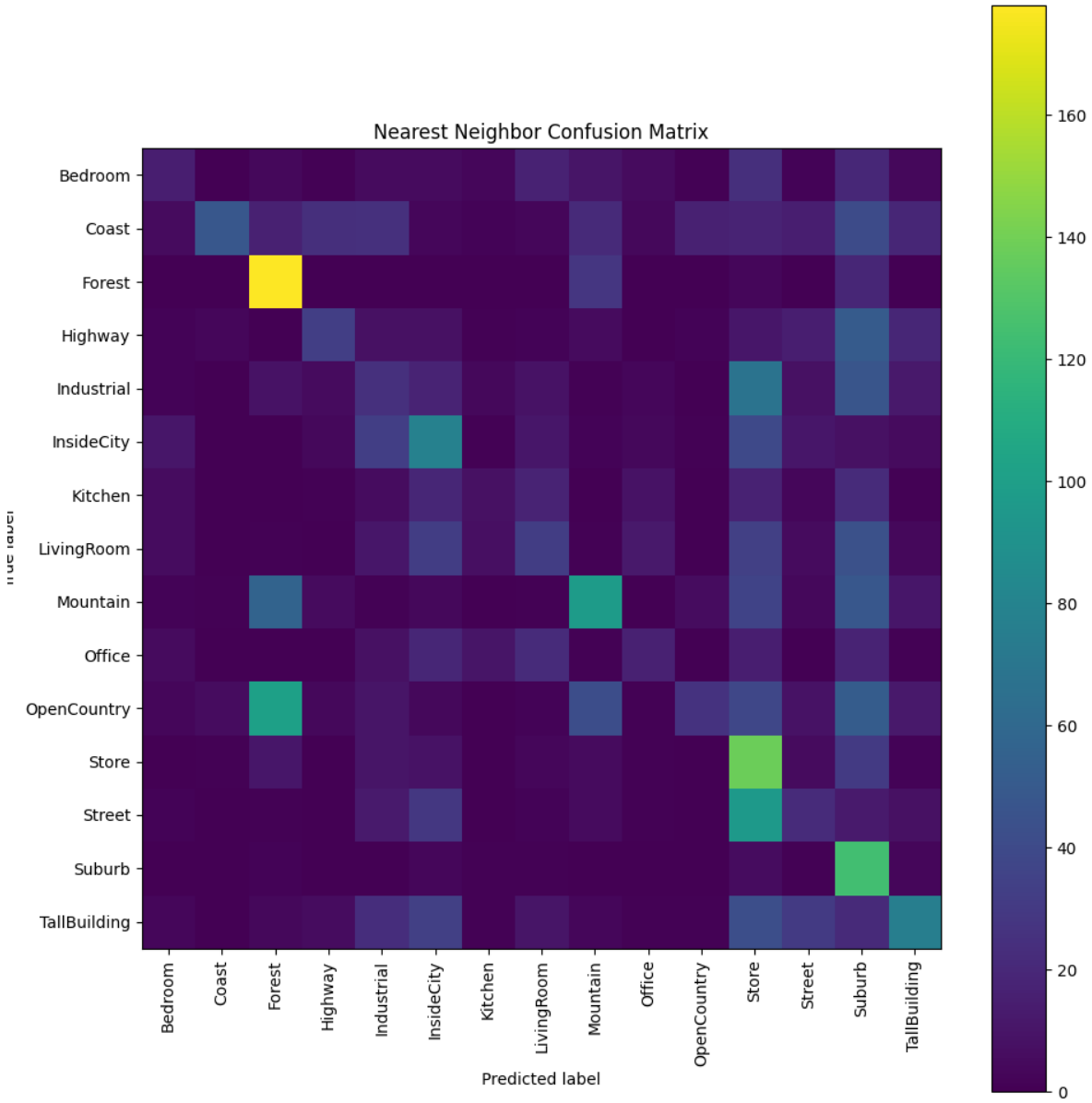
All experiments are conducted using a vocabulary size of **K = 400**. The following accuracies are obtained on the test set:

Model	**Feature Representation**	**Accuracy**
Nearest Neighbor	BoW	0.307
Linear SVM	BoW	0.426
Chi-square SVM	Spatial Pyramid BoW	0.610

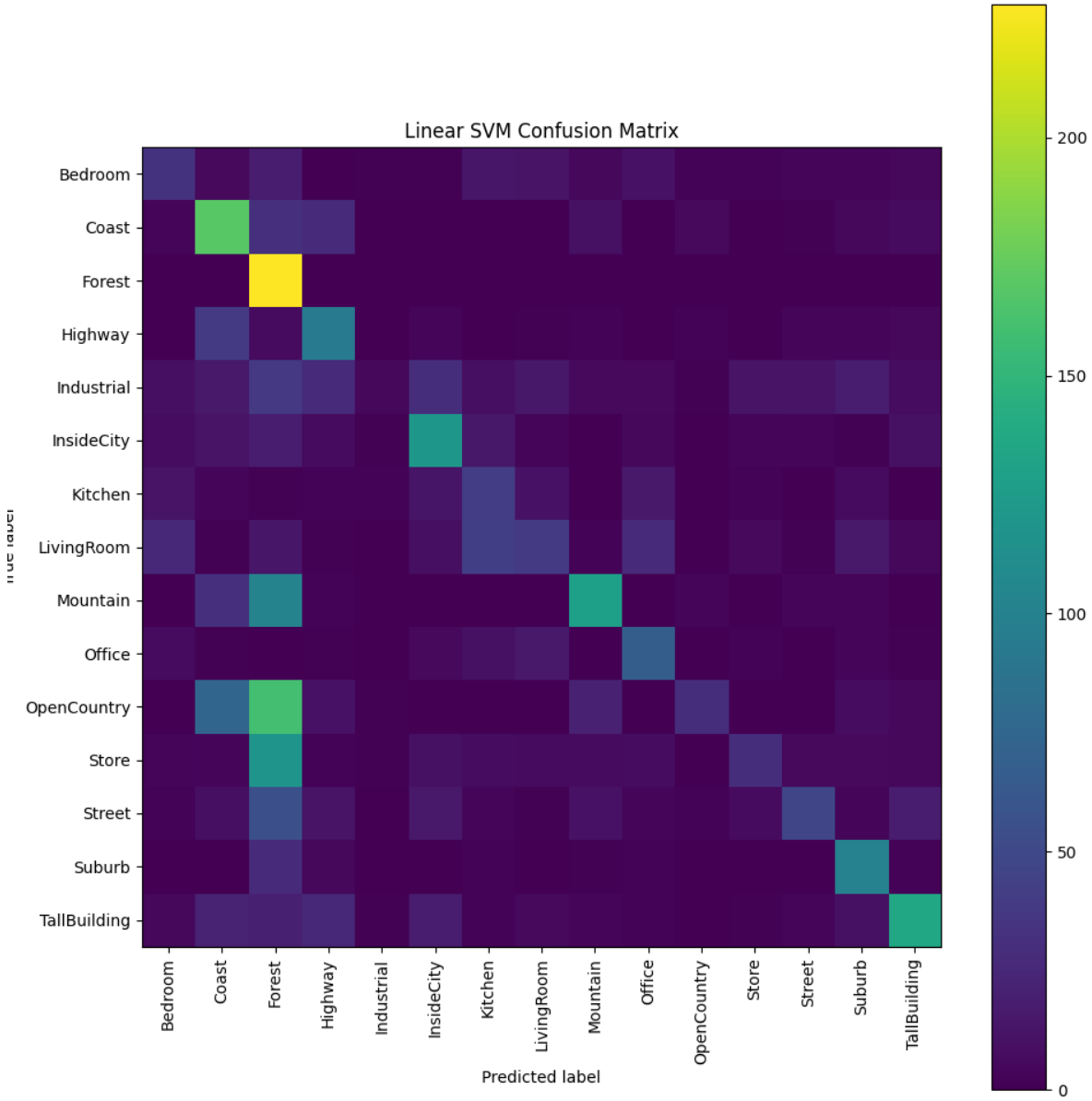
The results show a clear progression in performance as more expressive models and representations are used.

4.2 Confusion Matrix Analysis

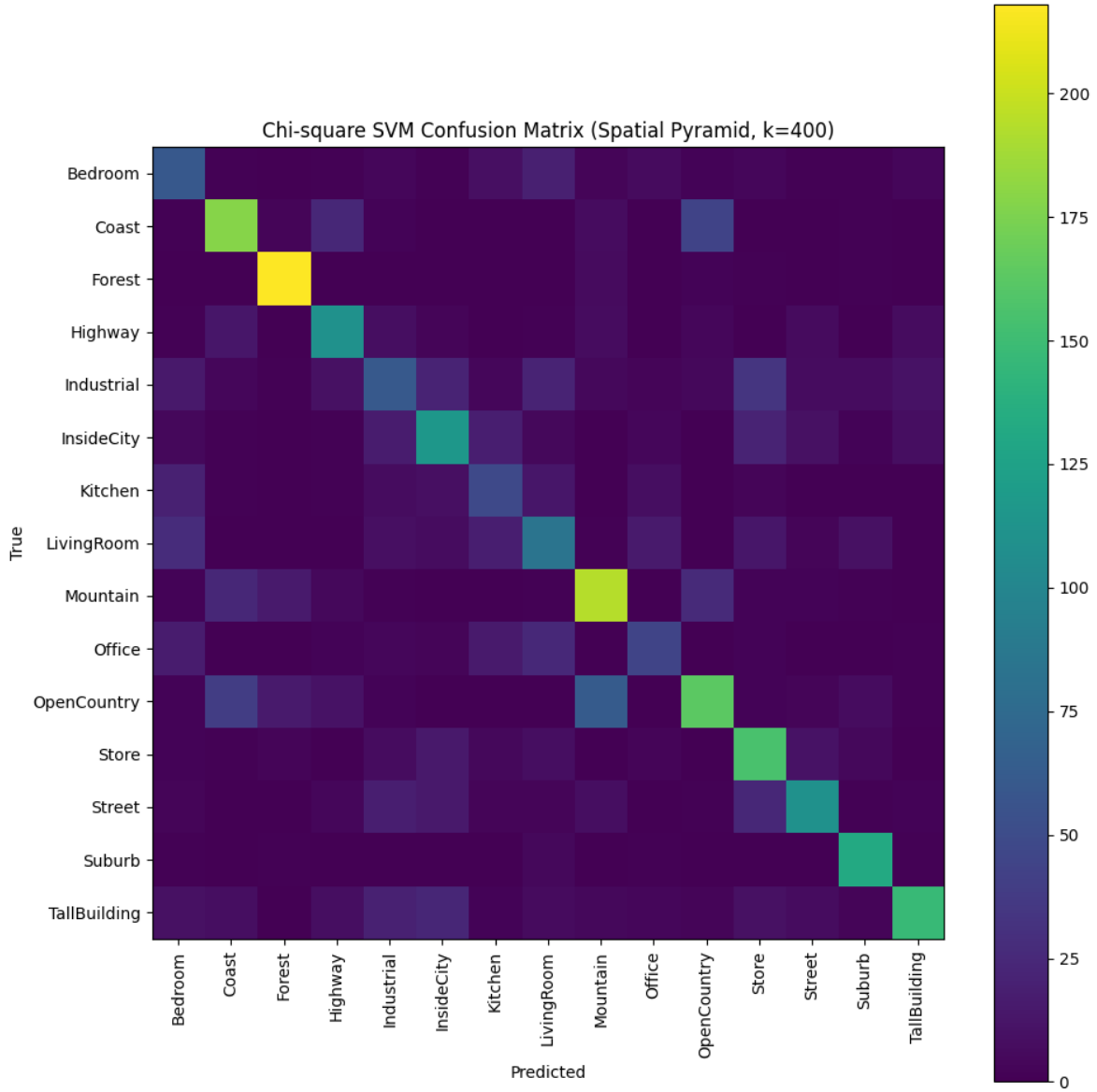
Confusion matrices are generated for all three classifiers.



The nearest neighbor model exhibits widespread confusion between visually similar scene categories, particularly among outdoor environments. This behavior is expected due to the lack of learned decision boundaries.



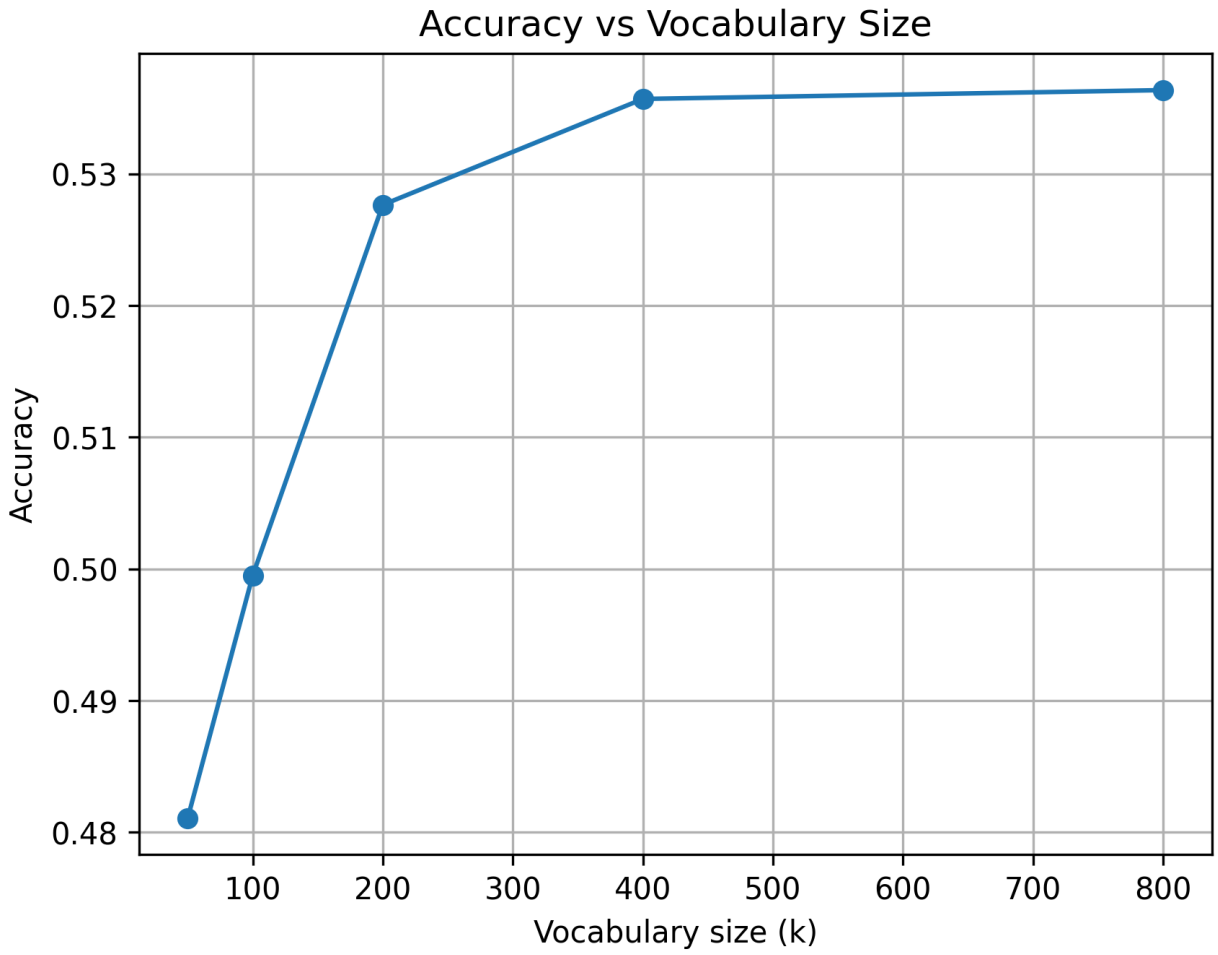
The linear SVM reduces confusion across many classes, demonstrating the benefit of discriminative learning even with a simple global representation.



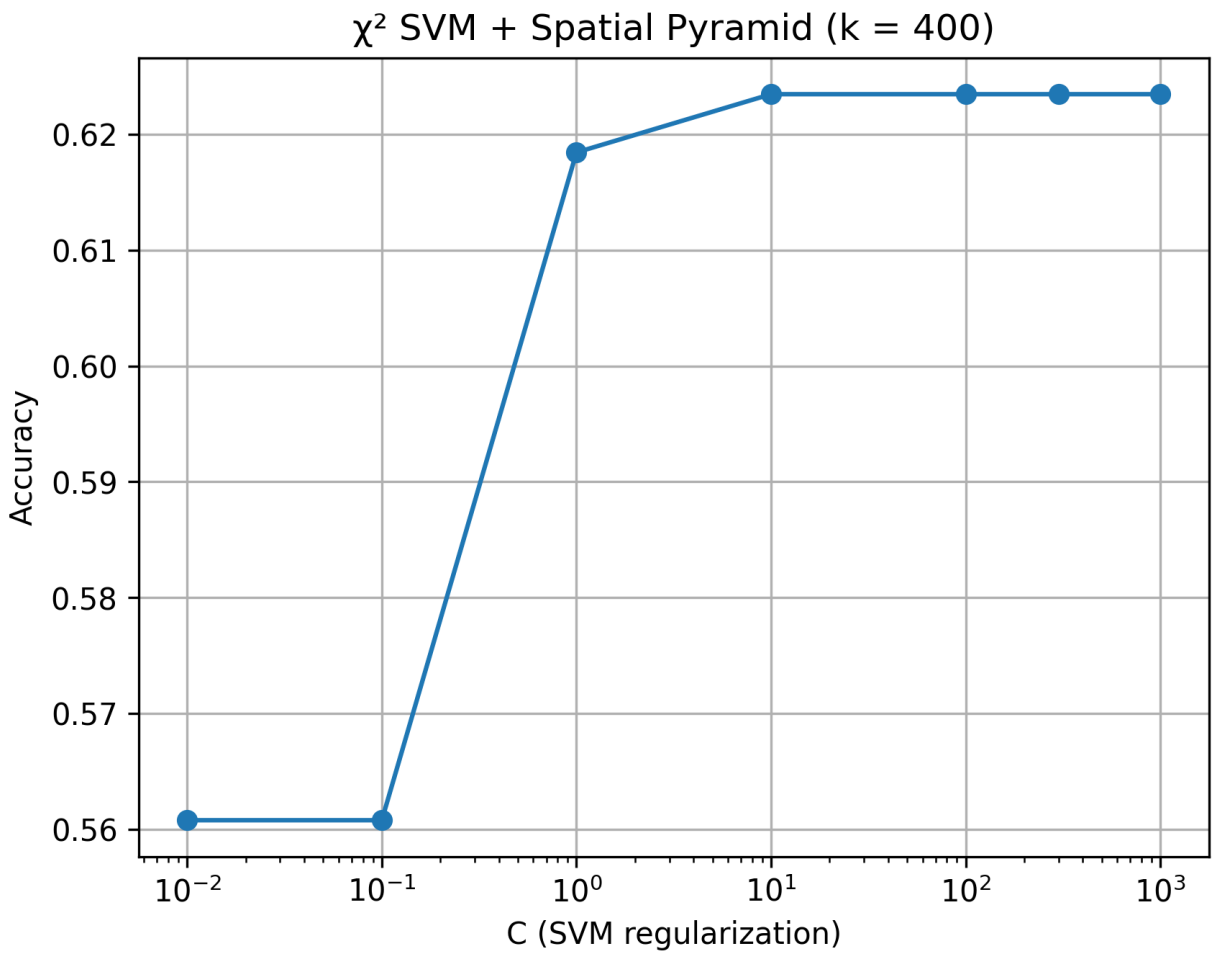
The chi-square SVM with spatial pyramid yields the cleanest confusion matrix overall. Improvements are especially noticeable in categories where spatial layout is informative, such as indoor scenes. Remaining errors are primarily concentrated among classes with similar global structure and texture.

4.3 Hyperparameter Analysis

Two hyperparameter sweeps are performed during development.



The first sweep evaluates different vocabulary sizes (K), showing that performance improves with larger vocabularies up to a point before saturating. Based on this analysis, **K = 400** is selected as a trade-off between performance and computational cost.



The second sweep evaluates the regularization parameter C for the SVM. Results indicate that performance is relatively stable across a range of C values, suggesting that the final model is not overly sensitive to this parameter. **C =10** was chosen

These sweeps are used to guide model optimization but are not part of the final evaluation.

5. Discussion

The experimental results demonstrate that incorporating spatial information and an appropriate kernel significantly improves scene classification performance. While plain bag-of-words representations capture local appearance, they discard spatial layout, which is critical for distinguishing many scene categories. The spatial pyramid partially addresses this limitation by preserving coarse spatial structure.

Additionally, the chi-square kernel provides a better similarity measure for histogram-based features than linear distance metrics. The high training accuracy observed for the kernel SVM suggests some degree of overfitting, but test performance remains substantially higher than that of the baseline models.

6. Conclusion

This project implements a complete scene classification pipeline based on SIFT features, bag-of-visual-words representations, and supervised learning. Experimental results show that a chi-square kernel SVM combined with a spatial pyramid representation significantly outperforms simpler baselines. These findings are consistent with established results in classical computer vision literature.

Future work could explore denser feature extraction, deeper spatial pyramids, or learned feature representations to further improve performance.

7. References

1. Forsyth, D. and Ponce, J. (2012). Computer Vision: A Modern Approach, 2nd edition. Pearson Education Limited.
2. Stockman, G. and Shapiro, L. G. (2001). Computer Vision. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition.
3. OpenCV Developers. OpenCV Documentation. https://docs.opencv.org/