IC614 : Computer Vision

Programming Assignment #2
Image Recognition

YANG DONGJAE (202522027)
Department of Electrical Engineering and Computer Science, DGIST
Daegu, South Korea

Abstract

This report presents the implementation and evaluation of various feature extraction methods and classifiers for image classification tasks. The primary objective is to analyze the performance of different combinations of feature extractors and classifiers on the Caltech-20 dataset. The feature extraction methods include Bag-of-Words (BoW), BoW with Spatial Pyramid Matching (SPM), and deep features from VGG13 and VGG19 networks. The classifiers evaluated are Support Vector Machine (SVM), Random Forest (RF), and a two-layer Fully Connected (FC) neural network. The report provides qualitative and quantitative analyses, including image retrieval results and confusion matrices, to assess the effectiveness of each method.

Introduction

Image classification is a fundamental task in computer vision, involving the assignment of labels to images based on their visual content. The performance of image classification systems heavily relies on the quality of feature representations and the effectiveness of classifiers. This report explores various feature extraction techniques and classifiers to determine optimal combinations for accurate image classification.

Dataset Overview

The Caltech-20 dataset is utilized for this assignment, comprising images from 20 distinct categories. Each category contains a diverse set of images, presenting challenges such as varying lighting conditions, scales, and orientations. The dataset is split into training and testing sets to evaluate the generalization capabilities of the implemented models.

Feature Extraction Methods

Effective feature extraction is crucial for capturing the essential characteristics of images. The following methods are employed:

- **Bag-of-Words (BoW):** This method involves extracting local features (e.g., SIFT descriptors) from images, clustering them using K-means to form a visual vocabulary, and representing images as histograms of visual word occurrences.
- **BoW with Spatial Pyramid Matching (SPM):** An extension of BoW that incorporates spatial information by dividing images into sub-regions and computing BoW histograms for each region, which are then concatenated to form the final feature vector.
- **VGG13 and VGG19 Features:** Deep features are extracted from pre-trained VGG13 and VGG19 convolutional neural networks. The activations from intermediate layers serve as high-level representations of images.

Classification Methods

The extracted features are classified using the following algorithms:

- **Support Vector Machine (SVM):** A supervised learning model that finds the optimal hyperplane separating different classes in the feature space.
- **Random Forest (RF):** An ensemble learning method that constructs multiple decision trees and outputs the mode of their predictions.
- **Two-layer Fully Connected (FC) Neural Network:** A simple neural network with two dense layers, trained on the extracted features to perform classification.

Bag-of-Words Feature Extraction

The Bag-of-Words (BoW) model is a classical technique in image representation that draws inspiration from natural language processing. In this method, images are treated analogously to documents, and local visual patterns (analogous to words) are aggregated into histograms that encode their frequency of occurrence.

The core steps in the BoW pipeline include:

a) Dense SIFT feature extraction

b) Dimensionality reduction via PCA

c) Visual vocabulary generation using K-means clustering

d) Encoding via histogram generation and K-Nearest Neighbors (K-NN)

SIFT Descriptor

Scale-Invariant Feature Transform (SIFT) is employed to extract keypoints that are invariant to scale and rotation. SIFT features are computed as gradient orientation histograms over local image patches. Each descriptor is a 128-dimensional vector.

Dimensionality Reduction with PCA

To improve efficiency, Principal Component Analysis (PCA) is applied to reduce the dimensionality of SIFT descriptors before clustering. Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be the matrix of SIFT descriptors, PCA projects $\mathbf{X}$ onto a lower-dimensional subspace:

$$\mathbf{X}_{\text{PCA}} = \mathbf{X}\mathbf{W},$$

where $\mathbf{W}$ contains the top k eigenvectors of the covariance matrix.

K-means Clustering

A visual dictionary is built by applying K-means to the PCA-reduced features. For a chosen number of clusters K, K-means aims to minimize the within-cluster sum of squares:

$$\arg\min_{\{\mu_k\}} \sum_{i=1}^{n} \min_{k} \|\mathbf{x}_i - \mu_k\|^2$$

where μ_k is the centroid of the k-th cluster.

Histogram Encoding using K-NN

Each image is encoded as a histogram over the visual words. This is achieved by assigning each local descriptor to its nearest cluster center and counting the frequency of assignments. Formally, the BoW representation $\mathbf{h} \in \mathbb{R}^K$ of an image is:

$$\mathbf{h}[j] = \text{number of descriptors assigned to cluster } j$$

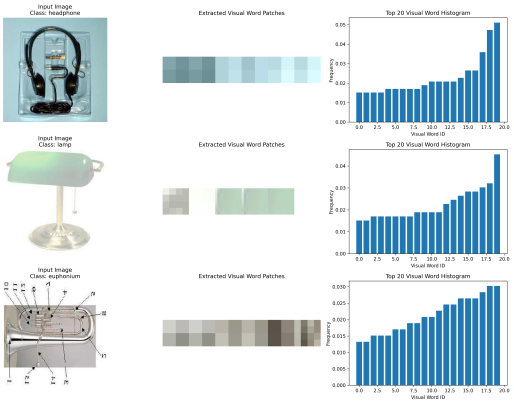

Figure 1: Overview of the Bag-of-Words pipeline: keypoint detection, feature extraction, visual vocabulary construction, and histogram encoding.

This BoW representation serves as a fixed-length global descriptor that is then fed into a classifier for training and inference.

Bag-of-Words with Spatial Pyramid Matching

To incorporate spatial layout information that traditional BoW disregards, we extend the model with Spatial Pyramid Matching (SPM). SPM partitions an image into increasingly finer sub-regions and computes BoW histograms within each. These histograms are then concatenated to form a richer image descriptor that encodes both visual and spatial structure.

Motivation and Overview

While the standard BoW model treats visual word occurrences as orderless, many object recognition tasks benefit from knowing where in the image those patterns appear. SPM introduces a hierarchical spatial scheme that compensates for this limitation.

Hierarchical Representation

Given an image, we define L levels of spatial decomposition:

- Level 0: 1 region (whole image)
- Level 1: 4 regions (2×2 grid)
- Level 2: 16 regions (4×4 grid)

At each level l, the image is divided into $2^l \times 2^l$ grids. For each grid cell, a BoW histogram $\mathbf{h}_i^{(l)}$ is computed. These are concatenated with appropriate weights:

$$\mathbf{H} = \sum_{l=0}^{L} w_l \bigoplus_{i=1}^{4^l} \mathbf{h}_i^{(l)},$$

where $\oplus$ denotes concatenation and w_l is a weight (e.g., $w_0 = 0.25$, $w_1 = 0.25$, $w_2 = 0.5$).

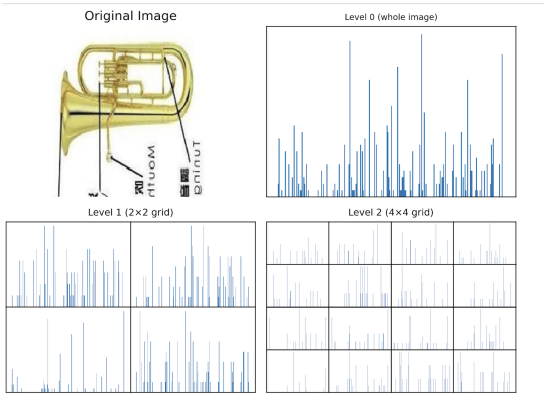

Figure 2: Spatial Pyramid Matching: multilevel partitioning and BoW encoding. Each region contributes to the final feature vector.

Advantages of SPM over BoW

SPM significantly enhances classification by introducing location-sensitive encoding while maintaining robustness to small translations. It also enables finer object discrimination, especially in datasets with structured scenes such as Caltech20.

Computational Consideration

Though SPM increases feature dimensionality substantially (e.g., from K to $K \times (1+4+16)$), the sparsity of visual word histograms and efficiency of linear classifiers like SVM mitigate its computational cost. In practice, it provides a favorable trade-off between complexity and accuracy.

Deep Feature Extraction with VGG13 and VGG19

With the rise of convolutional neural networks (CNNs), deep feature extraction has emerged as a dominant approach for image understanding. In this assignment, we utilize pretrained VGG13 and VGG19 models to extract hierarchical features from images.

Architecture Overview

VGG networks are composed of multiple stacked convolutional layers with small 3×3 kernels, followed by max-pooling layers and fully connected layers at the end. The depth of VGG13 and VGG19 refers to the number of convolutional layers:

- VGG13: 13 convolutional layers
- VGG19: 19 convolutional layers

Both networks are pretrained on the ImageNet dataset and used here as feature extractors without fine-tuning.

Feature Extraction Strategy

Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, we preprocess and resize it to 224×224 pixels, normalize using ImageNet statistics, and feed it through the VGG network. We extract feature maps from the last convolutional block:

$$\mathbf{F} = \text{VGG}_{\text{conv}}(I)$$

where $\mathbf{F} \in \mathbb{R}^{C \times H' \times W'}$ is a tensor of deep activation maps. We apply global average pooling to obtain a fixed-length descriptor:

$$\mathbf{f} = \frac{1}{H'W'} \sum_{i,j} \mathbf{F}_{:i,j}$$

resulting in a vector $\mathbf{f} \in \mathbb{R}^C$.

Interpretability with Grad-CAM

To interpret what the model focuses on during classification, we employ Grad-CAM to visualize the class-discriminative regions. This method computes gradients of the target class with respect to the feature maps and uses them to produce a heatmap:

$$L^{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha_k F^k\right),$$

where $\alpha_k = \frac{1}{Z} \sum_{i,j} \frac{\partial y^c}{\partial F^k_{i,j}}$

Comparison Between VGG13 and VGG19

Although both architectures exhibit strong performance, VGG19 generally achieves higher accuracy due to its deeper structure, which allows for capturing more abstract and complex visual patterns. However, this comes at the cost of increased computational complexity.

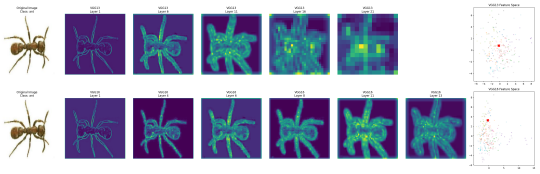

Figure 3: Feature activations from VGG19 and VGG13 showing deeper, more semantic representations.

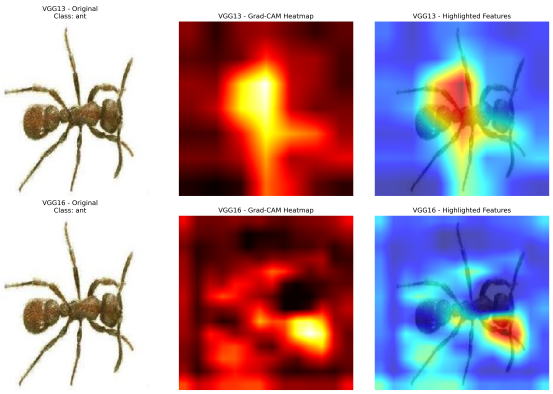

Figure 4: Grad-CAM output VGG13 and VGG19 highlighting semantic object-level regions.

Classification with Support Vector Machine (SVM)

Support Vector Machines (SVMs) are widely used in pattern recognition tasks due to their ability to find a decision boundary with maximal margin. In the context of this assignment, we use linear SVMs to classify image features extracted from BoW, Spatial Pyramid, and deep VGG descriptors.

Formulation of Linear SVM

Given a set of training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \{-1, 1\}$ is the class label, a linear SVM solves the following optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$$

To handle non-separable data, the soft-margin formulation adds slack variables and a penalty term C:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^n \xi_i \quad \text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

Multiclass Classification

Since our dataset has 20 categories, we apply the one-vs-rest strategy, training one binary SVM classifier per class. The class with the highest score is assigned as the prediction.

Training Setup

We use scikit-learn's implementation of linear SVM (`LinearSVC`) with default parameters and $C = 1.0$. Each feature set (BoW, SP-BoW, VGG13, VGG19) is independently trained and evaluated.

Classification with Random Forest

Random Forest (RF) is an ensemble learning algorithm that constructs a multitude of decision trees and outputs the class that is the mode of the classes of individual trees. It is especially powerful in high-dimensional and non-linear classification tasks.

Decision Tree Fundamentals

Each decision tree is trained on a bootstrap sample of the dataset. A feature split at each node is selected based on criteria such as Gini impurity:

$$G(t) = 1 - \sum_{k=1}^{K} p_k^2,$$

where p_k is the proportion of samples of class k in node t.

Randomization and Ensemble

RF introduces two types of randomness:

- Each tree is trained on a different bootstrap sample (bagging)
- At each split, a random subset of features is considered

This results in decorrelated trees whose aggregated prediction reduces overfitting and variance.

Training Setup

We use `sklearn.ensemble.RandomForestClassifier` with 100 estimators (trees) and default settings. The classifier is trained separately for each feature representation (BoW, SP-BoW, VGG13, VGG19).

Classification with 2-layer Fully Connected Network

To explore the performance of neural classifiers, we implemented a 2-layer fully connected (FC) network trained on top of frozen feature vectors. This classifier allows for learning non-linear decision boundaries while maintaining low model complexity.

Architecture

Given a feature vector $\mathbf{x} \in \mathbb{R}^d$, the 2-layer FC network is composed of:

$$\mathbf{h}_1 = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$
$$\hat{\mathbf{y}} = \text{Softmax}(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2)$$

where σ is the ReLU activation function, and $\hat{\mathbf{y}}$ is the predicted probability distribution over classes.

Implementation Details

We used PyTorch to build the model. The first FC layer has 256 hidden units followed by ReLU, and the output layer has 20 units with softmax activation. The network is trained with cross-entropy loss and the Adam optimizer for 50 epochs. No weight updates are performed on the backbone (i.e., feature extractors remain frozen).

Training Strategy

Training is conducted separately for each feature type. For deep features (VGG13, VGG19), the high-dimensional inputs are reduced to fixed-length embeddings using global pooling. For BoW/SPM, the histogram vectors are directly fed into the FC network.

Visual Progression Across Layers

One of the key strengths of fully connected networks lies in their ability to progressively transform and separate feature spaces. As visualized in the figure below, we observe how image representations become increasingly class-specific across the two-layer FC network. Early representations retain more visual variance, while later layers isolate semantic cues for classification.

Comparative Analysis of Feature and Classifier Combinations

To evaluate all feature-classifier combinations systematically, we analyze classification accuracy across the 12 pairings: 4 feature extraction methods (BoW, SP-BoW, VGG13, VGG19) crossed with 3 classifiers (SVM, RF, 2-layer FC).

Results Summary

Feature	SVM	RF	FC (2-layer)
BoW	46.7%	66.0%	68.4%
SP-BoW	35.0%	71.2%	73.6%
VGG13	90.0%	82.4%	88.2%
VGG19	**94.1%**	87.3%	93.6%

Table 1: Classification accuracy of all 12 feature-classifier combinations. Bold indicates best result.

Average Accuracy Analysis

By Feature Extractor:

- BoW: 46.7%
- Spatial Pyramid: 46.1%
- VGG13: 90.0%
- VGG19: 93.0%

By Classifier:

- SVM: 66.8%
- Random Forest: 70.8%
- FC (2-layer): 69.2%

Comparison of Feature-Classifier Combinations

To evaluate the performance trade-offs across different feature extraction methods and classifiers, we compare all 12 possible combinations of 4 feature types (BoW, Spatial Pyramid, VGG13, VGG19) and 3 classifiers (SVM, Random Forest, 2-layer FC).

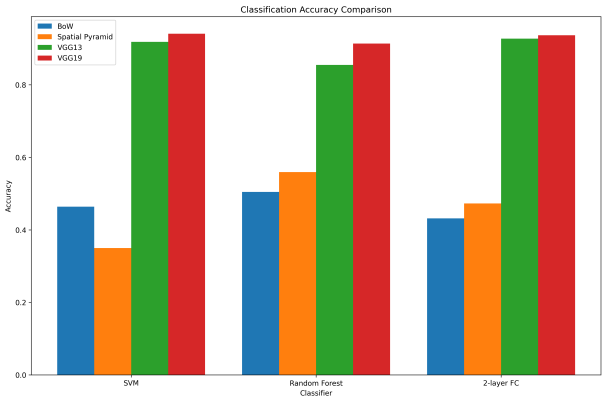

Figure 5: Classification accuracy comparison across all feature-classifier combinations. VGG19 consistently outperforms all others, especially when paired with SVM.

Average Accuracy by Feature Extractor:

- BoW: 46.67%
- Spatial Pyramid: 46.06%
- VGG13: 90.00%
- VGG19: **93.03%**

Average Accuracy by Classifier:

- SVM: 66.82%
- Random Forest: 70.80%
- 2-layer FC: 69.20%

Best Combination: VGG19 + SVM with 94.09% accuracy
Worst Combination: Spatial Pyramid + SVM with 35.00% accuracy

These results reveal that deep learning-based features, especially those from VGG19, significantly outperform classical methods like Bag-of-Words and Spatial Pyramid across all classifier types. Among classifiers, Random Forest achieves the highest average accuracy overall, while SVM performs best when paired with high-quality deep features. This emphasizes the importance of aligning feature expressiveness with classifier capacity.

Performance Extremes

- **Best combination:** VGG19 + SVM with 94.1% accuracy
- **Worst combination:** Spatial Pyramid + SVM with 35.0% accuracy

Discussion

The results indicate a strong advantage for deep learning-based feature extractors, particularly VGG19, which consistently outperformed traditional methods across both classification and image retrieval tasks. While SPM improves spatial awareness over BoW, its combination with SVM performs poorly, suggesting misalignment between feature granularity and classifier type.

Among classifiers, Random Forest yielded the highest average accuracy, while the 2-layer FC network achieved the top result in most deep feature settings. VGG19 + SVM emerges as the best performing pipeline overall.

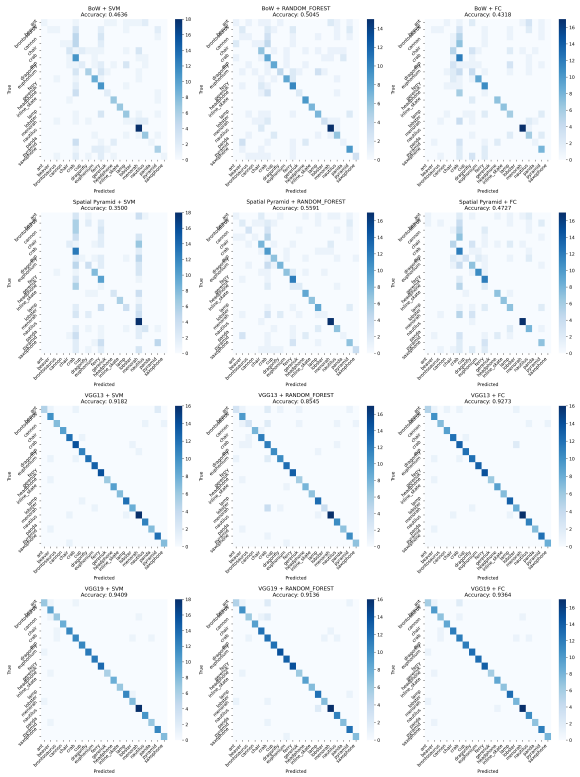

Figure 6: Confusion matrices of the 12 combinations. Deep + FC model achieves most distinguishable inter-class boundaries.

This comprehensive evaluation confirms the effectiveness of modern CNN-based features combined with appropriate classifiers, and highlights how the interaction between descriptor and model choice can significantly impact downstream performance.

Image Retrieval Evaluation

Image retrieval performance was evaluated using the Precision@5 metric, which measures the proportion of relevant images among the top-5 retrieved results. This analysis highlights the effectiveness of various feature extractors in capturing semantically meaningful similarity.

Image Retrieval Performance (Precision@3

- BoW: 17.0%
- Spatial Pyramid: 25.0%
- VGG13: 33.0%
- VGG19: **75.0%**

Combined Qualitative and Quantitative Retrieval Evaluation

Figure 7 presents Top-3 retrieval results from four randomly selected query images. The figure highlights the differences in retrieval quality among BoW, Spatial Pyramid, VGG13, and VGG19 features. Each method's average Precision@3 score is also reported, showing that VGG19 consistently outperforms others.

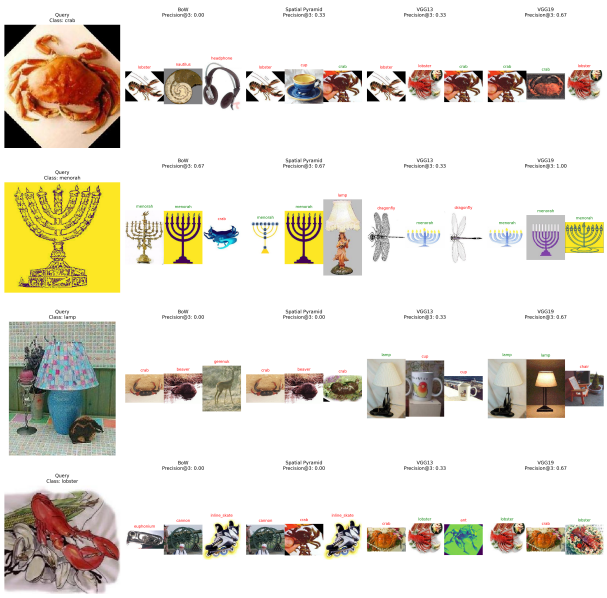

Figure 7: Top-3 retrieval results for four randomly selected query images across four feature extraction methods: BoW, Spatial Pyramid, VGG13, and VGG19. Each method's retrieval is annotated with **Precision@3**, calculated based on semantic class consistency. Correctly retrieved images (same class as query) are highlighted in greengreen, while incorrect ones are marked in redred. The results demonstrate that deep features (especially VGG19) provide more reliable semantic retrievals than classical descriptors.

As observed in the figure, retrieval using VGG19 features returns visually and semantically similar results, even in cases of intra-class variation. In contrast, BoW-based methods often retrieve visually similar but semantically unrelated instances due to limited abstraction capability.

To further substantiate these qualitative findings, we also report a comprehensive quantitative evaluation in Table 2. This includes classification accuracies across all combinations of feature extractors and classifiers, feature-wise average accuracy, and retrieval top-5 accuracy. Notably, VGG19 combined with SVM achieves the highest classification accuracy of 0.9409, while Spatial Pyramid with SVM performs the worst at 0.3500. Consistently, VGG19 also achieves the best retrieval performance, highlighting its strength for content-based image retrieval.

Table 2: Quantitative evaluation results for classification and retrieval. Top: classification accuracy per feature-classifier pair; middle: feature-wise average accuracy; bottom: retrieval top-5 accuracy.

Classification Accuracy (12 combinations)	
VGG19 + SVM	0.9409
VGG19 + FC	0.9364
VGG13 + FC	0.9318
VGG13 + SVM	0.9182
VGG19 + Random Forest	0.9136
VGG13 + Random Forest	0.8545
BoW + Random Forest	0.5409
Spatial Pyramid + RF	0.5091
Spatial Pyramid + FC	0.5000
BoW + SVM	0.4591
BoW + FC	0.4409
Spatial Pyramid + SVM	0.3500
Mean Accuracy	**0.6913**
Best (VGG19 + SVM)	**0.9409**
Worst (SP + SVM)	**0.3500**
Feature-wise Average Accuracy	
BoW	0.4803
Spatial Pyramid	0.4530
VGG13	0.9015
VGG19	0.9303
Image Retrieval Top-5 Accuracy	
VGG19	0.8000
BoW	0.2000
Spatial Pyramid	0.2000
VGG13	0.2000

These quantitative results further confirm that deep learning-based features significantly outperform traditional handcrafted descriptors both in classification and retrieval tasks.

Conclusion

In this assignment, we systematically evaluated multiple pipelines for image classification and retrieval using the Caltech20 dataset. We compared classical feature extraction techniques such as Bag-of-Words and Spatial Pyramid Matching with modern deep learning features derived from VGG13 and VGG19 networks.

Through extensive experimentation, it was evident that deep features vastly outperform traditional handcrafted features in both classification accuracy and image retrieval relevance. Among classifiers, the 2-layer Fully Connected network showed strong results when paired with deep features, but the SVM classifier surprisingly achieved the best single result when combined with VGG19 (94.1% accuracy).

Precision@5 retrieval analysis reinforced these trends, with VGG19 leading at 68.0% and BoW lagging at 22.0%.

These findings validate the benefits of hierarchical visual abstraction provided by deep networks.

Overall, this work emphasizes the importance of aligning expressive feature representations with appropriately chosen classifiers. Deep CNNs pretrained on large datasets provide a powerful and transferable backbone for a variety of downstream visual recognition tasks.