Scene Recognition Report

Your Team Members' Names and ECS IDs

April 21, 2025

1 Introduction

This report describes the detailed implementations of three different approaches to the scene recognition task provided in COMP6223 Coursework 3. Each approach, its classifier, feature extraction methods, parameter selections, and functions used are thoroughly explained.

2 Run 1: Tiny Image with KNN Classifier

For Run 1, we implemented a simple k-nearest neighbour (KNN) classifier with the tiny image feature representation. This approach involves the following detailed steps:

1. **Image Preprocessing (Function: create_tiny_image):** Images were centrally cropped into a square, resized to a fixed 16x16 resolution using the PIL library, and flattened into a one-dimensional vector. Each feature vector was normalized to have zero mean and unit length to improve robustness.
2. **Training Data Preparation (Function: load_training_data):** Loaded images were converted into tiny image vectors and labeled according to their respective classes.
3. **KNN Classifier Training (Function: train_knn_classifier):** A KNN classifier was trained with Euclidean distance as the metric. Various values of k were tested, with $k = 3$ chosen based on validation performance.
4. **Testing and Prediction (Function: predict_and_save):** The trained KNN model was used to classify the test images, and predictions were saved in the required format.

3 Run 2: Bag-of-Visual-Words with Linear Classifiers

Run 2 utilized a bag-of-visual-words (BoVW) representation based on dense pixel patch sampling and linear classifiers:

1. **Patch Extraction (Function: extract_patches):** Images were divided into overlapping 8x8 patches sampled every 4 pixels.
2. **Patch Normalization (Function: normalize_patches):** Each patch was mean-centered and normalized to unit length.
3. **Vocabulary Construction (Function: build_vocabulary):** Normalized patches from training images were clustered into 500 visual words using K-Means clustering.
4. **Feature Extraction (Function: extract_bow_features):** Images were represented as histograms of visual word occurrences.
5. **Linear Classifier Training (Function: train_linear_classifiers):** Logistic regression classifiers were trained using a one-vs-all strategy.
6. **Testing and Prediction (Function: predict_and_save):** Predictions were made by selecting the class with the highest classifier confidence.

4 Run 3: Dense SIFT with SVM Classifier

In Run 3, a dense Scale-Invariant Feature Transform (SIFT) method combined with an SVM classifier was employed:

1. **Dense SIFT Extraction (Function: extract_dense_sift):** SIFT descriptors were densely extracted from multiple scales (1.0, 0.75, and 0.5) with an 8-pixel step size using OpenCV.
2. **Visual Vocabulary Creation (Function: build_vocabulary):** Descriptors were clustered into 500 visual words using OpenCV's implementation of K-Means.
3. **Feature Representation (Function: extract_bow_features):** Histograms of visual words represented each image.
4. **Feature Standardization (Function: StandardScaler from sklearn):** Feature histograms were standardized to have zero mean and unit variance.
5. **Classifier Training (Function: train_svm_classifier):** A Support Vector Machine with a radial basis function (RBF) kernel was trained. Cross-validation was used to optimize hyperparameters.
6. **Testing and Prediction (Function: predict_and_save):** Predictions on test images were generated and stored accordingly.

5 Individual Contributions

- Member 1: Implementation and validation of Run 1, including image preprocessing and KNN tuning.
- Member 2: Developed Run 2 by extracting patches, constructing visual vocabularies, and training logistic regression classifiers.
- Member 3: Dense SIFT feature extraction and visual vocabulary construction for Run 3.
- Member 4: SVM classifier training, feature standardization, and validation for Run 3.

6 Conclusion

We implemented and evaluated three distinct methods for scene classification ranging from simple tiny image features with KNN classification to advanced dense SIFT descriptors with a non-linear SVM. Run 3 demonstrated the most advanced features and sophisticated classifier, providing potentially superior recognition performance.