ML-Project Report.

Team members:

Name	ID
Ali Mohsen	20221106
Sherif Mahmoud	20221080
Mostafa Mohamed	20221153
Youssef Waleed	20221206
Noureldin Yacer	20220462

1. Introduction

This report offers an in-depth evaluation on two kinds of classification models: Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN), coupled with two ways of feature extraction: traditional and deep learning-inspired convolutional neural networks (CNNs). This is aimed at understanding the performance trade-off, efficiency, and applicability of the models for real-world applications related to image classification tasks. This report is based on an experiment performed on an existing set with data of 5,476 instances across six categories.

2. Feature Extraction Methods

Two major methods of feature extraction:

Multiple descriptor methods were considered, including:

1) Local Binary Patterns (LBP) - texture analysis.

2) Gray-Level Co-occurrence Matrix (GLCM) for structural textures.

3) Color histograms in the HSV color space for color distribution.

4) Color Moments for statistical color characterization.

5) Histogram of Oriented Gradients (HOG) for shape and edge information.

This set provides a 2,303-dimensional feature space for each image, which reflects multiple visual properties through proper feature engineering.

2.2 Deep Learning Features

The features were extracted employing a pre-trained ResNet50 model, initialized by ImageNet weights. The network was truncated before the final classification layer, yielding a 2,048-dimensional feature vector per image. The features are representative of complex, high-level semantic patterns learned by large-scale visual data.

3. Experimental Setup

The data was split into training, validation, and testing sets following the conventional split of 60-15-25. Both models were tested under similar circumstances:

1) SVM Parameters: radial basis function kernel with fixed hyperparameters (C=10 and gamma='scale'), and probability estimates set to True (probability=True).

2) K-NN Setup: Using Cosine distance measure, set k to 5, and voting to be weighted towards distance (weights='distance') to handle high-dimensional spaces better.

3) Evaluation Metrics: Accuracy, training time, confusion matrix analysis, and scores from 5-fold cross validation.

4. Results and Analysis

4.1 Performance with Handcrafted Features

Metric	SVM	k-NN
Test Accuracy	91.09%	85.03%
Validation Accuracy	87.83%	81.18%
Training Time	315.56 seconds	2.15 seconds

SVM performed better with the hand-designed feature set, with a testing accuracy of 91.09% compared to 85.03% for k-NN. The margin of improvement of 6.06% represents the strength of SVM in handling moderately high-dimensional space with possibly correlated variables. The confusion matrices also suggest better class balance for SVM with less off-diagonal error penalties. SVM, however, took 146 times longer to train compared to k-NN due to its computational complexity.

4.2 Performance with CNN Features

Metric	SVM	k-NN
Test Accuracy	98.54%	97.44%
Validation Accuracy	97.32%	96.15%
Training Time	109.16 seconds	5.12 seconds

Both classifiers performed well on CNN-based features, with 98.54% test accuracy on SVM and 97.44% on k-NN. This performance difference decreased to 1.1%, which emphasizes that CNN-based features provide a representation where distance-based classifiers work well even with high-dimensional data. Moreover, both classifiers showed significantly faster training times on CNN-based features than on handcrafted features, with 65% less training time on SVM and marginally higher times on k-NN because of its lazy learning mechanism.

4.3 Comparative Analysis

Aspect	**Handcrafted Features**	**CNN Features**
Dimensionality	2,303 dimensions	2,048 dimensions
SVM Accuracy	91.09%	98.54%
k-NN Accuracy	85.03%	97.44%
SVM Training Time	315.56s	109.16s
k-NN Training Time	2.15s	5.12s
Interpretability	High (explicit features)	Low (learned representations)
Feature Extraction Time	Moderate	High (requires GPU)

5. Discussion

5.1 Feature Representation Impact

The great improvement in accuracy achieved through CNN features (7.45% for SVM, 12.41% for k-NN) again verifies the superiority of learned feature representations over carefully hand-crafted features in this classification problem. Learned features not only possess stronger discriminative capability but also encourage more orthogonal feature representations, as indicated by the faster SVM training time on features learned by CNNs with similar dimensionality.

5.2 Classifier Characteristics

Support Vector Machines consistently demonstrated superiority over other algorithms on both feature types, and in particular, on hand-crafted features, its margin maximization strategy succeeded well in handling the diverse feature space. This sensitivity to feature quality is apparent from training times, taking 315.56 seconds for hand-crafted features, whereas for CNN features, it took 109.16 seconds, indicating that features derived from CNN lead to a separable space, resulting in fast convergence.

K-Nearest Neighbors showed strong performance on CNN features, defying the traditional notion that distance-related approaches become less efficient in high-dimensional space. This analysis suggests that, given well-organized spaces, the curse of dimensionality can be avoided. k-NN, however, needs to be considered from the perspective of its faster training time, while the computations involve linear costs for prediction.

5.3 Computational Trade-offs

The analysis reveals a multi-dimensional trade-off space:

1) Accuracy vs Training Time: SVM gives better accuracy but requires more time for training, especially when dealing with manual features.

2) Training Time vs. Inference Time: k-NN has very low training time complexity but high inference costs, while SVM is the reverse.

3) Feature Engineering vs. Feature Learning: The former requires expertise, tuning, and is interpretable, but CNN features, although requiring computation for extraction, perform better.

6. Conclusion

The results from the comparative analysis show that both feature representation and choice of classifier affect system performance. Features created from deep learning (ResNet50) perform much better than traditional designed features using both classifiers, with accuracy rates of more than 97% compared with 85-91% using traditional features. Although SVM produces more accurate results, another promising choice is k-NN, especially with CNN features, with training times much faster than SVM. The choice of which of these to use depends upon the requirements of the applications, such as the desired standards of accuracy, computational prerequisites, levels of interpretability, and implementational factors. Current vision recognition systems stand to gain from considering feature extraction and classification as an entity, with transformations from deep learning models when computational feasibility allows.