Machine Learning Project

TA: Farah

GitHub repo URL : nourhan279/machine_learning_proj

NAME	ID
Nourhan Mohamed Ahmed	20220368
Sarah Ibrahim Abd El Sattar	20220150
Mohamed Hussien Aly	20220286
Ahmed Mohamed Abass	20220035
Ahmed Yasser Ahmed	20220046

Material Stream Identification System

1. Introduction

This project presents an Automated Material Stream Identification System based on classical machine learning techniques combined with feature extraction. The system performs processing starting from raw image acquisition to real-time material classification using a live camera feed.

The main focus of this project is to demonstrate mastery of the complete machine learning pipeline, including data augmentation, feature extraction, classifier training, performance evaluation, and system deployment.

2. Dataset Description

The provided dataset contains images belonging to six primary material classes, in addition to a negative class (Unknown). The classes are:

ID	Class Name	Description
0	Glass	Bottles and jars made of silicate-based materials
1	Paper	Thin cellulose-based materials
2	Cardboard	Multi-layered rigid cellulose materials
3	Plastic	Polymer-based materials such as bottles and films
4	Metal	Metallic objects such as aluminum cans
5	Trash	Non-recyclable or contaminated waste
6	Unknown	Out-of-distribution or low-confidence samples

Images are organized into class-specific folders. The dataset was split into training (80%) and validation (20%) sets using stratified sampling to preserve class distributions.

3. Data Augmentation

Data augmentation was applied to the training dataset. The augmentation process increased the dataset size by approximately 60% .

Applied Augmentation Techniques:

- **Rotation:** Random rotation between -20° and +20°
- **Horizontal Flipping:** Simulates different object orientations

- **Brightness Adjustment:** Random intensity scaling to simulate lighting variations
- **Zooming:** Random center cropping followed by resizing

These techniques help the model become invariant to orientation, scale, and illumination changes commonly encountered in real-world scenarios.

4. Image Preprocessing

Before feature extraction, all images were standardized through the following preprocessing steps: - Resizing all images to 224 × 224 pixels - Ensuring all images are in RGB color format (3 channels) - Saving processed images in a unified directory structure

This step ensures compatibility with the chosen feature extraction model and consistency across the dataset.

5. Feature Extraction

5.1 MobileNetV2 Feature Extractor

A pre-trained MobileNetV2 convolutional neural network was used as a fixed feature extractor.

Key configuration details: - Pre-trained on ImageNet - `include_top = False` - `pooling = 'avg'` (Global Average Pooling)

This configuration converts each input image into a 1280-dimensional feature vector, significantly reducing dimensionality while preserving semantic information.

Advantages of MobileNetV2 Features:

- Computationally efficient
- Strong generalization due to ImageNet pretraining
- Suitable for real-time applications

6. Classifier Implementation

6.1 Support Vector Machine (SVM)

- **Kernel:** Radial Basis Function (RBF)
- **Regularization Parameter (C):** 10
- **Feature Scaling:** StandardScaler

The RBF kernel allows the SVM to model non-linear decision boundaries in the high-dimensional feature space.

Unknown Class Handling (SVM):

The SVM rejection mechanism is based on the `decision_function`. If the maximum confidence score is below a predefined threshold (0.6), the sample is classified as Unknown.

6.2 k-Nearest Neighbors (k-NN)

- **Number of Neighbors (k):** 2
- **Distance Metric:** Euclidean
- **Weighting Scheme:** Distance-based
- **Feature Scaling:** StandardScaler

k-NN is a non-parametric classifier that makes decisions based on proximity in feature space.

Unknown Class Handling (k-NN):

rejection is implemented using a distance-based threshold. If the distance to the nearest neighbor exceeds a learned threshold, the sample is labeled as Unknown.

7. Performance Evaluation

7.1 Dataset Statistics

- Training samples: 3690
- Validation samples: 923
- Feature vector size: 1280

7.2 Validation Results

k-NN Classifier:

- Validation Accuracy (without Unknown): 96.86%
- Validation Accuracy (with Unknown mechanism): 96.86%

The k-NN classifier achieved strong performance across all six primary classes, with slightly lower precision for the Trash class due to its visual diversity.

SVM Classifier:

- Validation Accuracy: 97.51%

The SVM achieved the highest overall accuracy and more stable class-wise performance compared to k-NN.

8. Classifier Comparison

Aspect	SVM	k-NN
Accuracy	Higher (97.5%)	Slightly Lower (96.9%)
Inference Speed	Fast	Slower for large datasets
Memory Usage	Low	High (stores all samples)
Unknown Handling	Confidence-based	Distance-based
Scalability	High	Limited

Overall, the SVM classifier was selected as the final model due to its superior accuracy, efficiency, and scalability.

9. Real-Time System Deployment

The best-performing SVM model was integrated into a real-time application using OpenCV.

Deployment Features:

- Live camera input
- Frame skipping to maintain high FPS
- On-the-fly MobileNetV2 feature extraction
- Real-time prediction display with confidence score
- FPS counter for performance monitoring

The system operates efficiently in real time and demonstrates robust classification performance .

10. Conclusion

The comparative analysis showed that SVM outperforms k-NN in terms of accuracy, efficiency, and deployment suitability. The implemented rejection mechanisms further enhance system robustness by handling unknown and out-of-distribution inputs.