

Material Stream Identification (MSI) Comprehensive Technical Report

1. Introduction

This report presents the design, implementation, and evaluation of a **Material Stream Identification (MSI)** system based on classical machine learning techniques. The main objective of the project is to accurately classify different material types from image data while preserving **real-time performance**, **low computational cost**, and **ease of deployment**. To achieve this, multiple feature extraction strategies and classifiers were explored and compared.

2. Feature Extraction Methods

2.1 Local Binary Patterns (LBP)

Local Binary Patterns (LBP) is a texture-based feature extraction method that captures local spatial structures in grayscale images. It is computationally lightweight, robust to illumination variations, and therefore well suited for real-time applications.

In this project, **uniform LBP** was applied using **P = 8 neighboring pixels** and a **radius R = 1**. This configuration produces a compact histogram of **10 bins**, resulting in a feature vector of size 10 that effectively represents texture information with minimal computational overhead.

2.2 CNN-Based Features (EfficientNet-B0 and ResNet-18)

Pretrained Convolutional Neural Networks (CNNs) were also evaluated as feature extractors to capture higher-level semantic information from images.

- **EfficientNet-B0**, pretrained on ImageNet, was used by removing its final classification layer and extracting the resulting feature vector. These features were mainly used with the **SVM classifier**.
- **ResNet-18**, also pretrained on ImageNet, was similarly used as a feature extractor by discarding its final classification layer. The extracted features were primarily used with the **k-NN classifier**.

Although CNN-based features significantly improved classification accuracy due to their strong representational power, they introduced higher computational cost and increased deployment complexity compared to handcrafted features.

2.3 Hybrid CNN + LBP Features

A hybrid feature representation was explored by concatenating CNN features with LBP features, resulting in a **1290-dimensional feature vector**. This approach was designed to combine the strengths of both methods:

- **CNN features** capture high-level semantic and structural information.
- **LBP features** focus on fine-grained texture and local spatial patterns at a low computational cost.

The motivation behind this hybrid setup is that texture information provided by LBP complements the semantic features extracted by CNNs. Several pretrained CNN models were tested, and the selected configuration achieved the best accuracy. Compared to traditional handcrafted features such as RGB or HSV histograms, CNN-based features significantly improved performance, and incorporating LBP further boosted accuracy with only a minor increase in computational overhead.

3. Classifiers

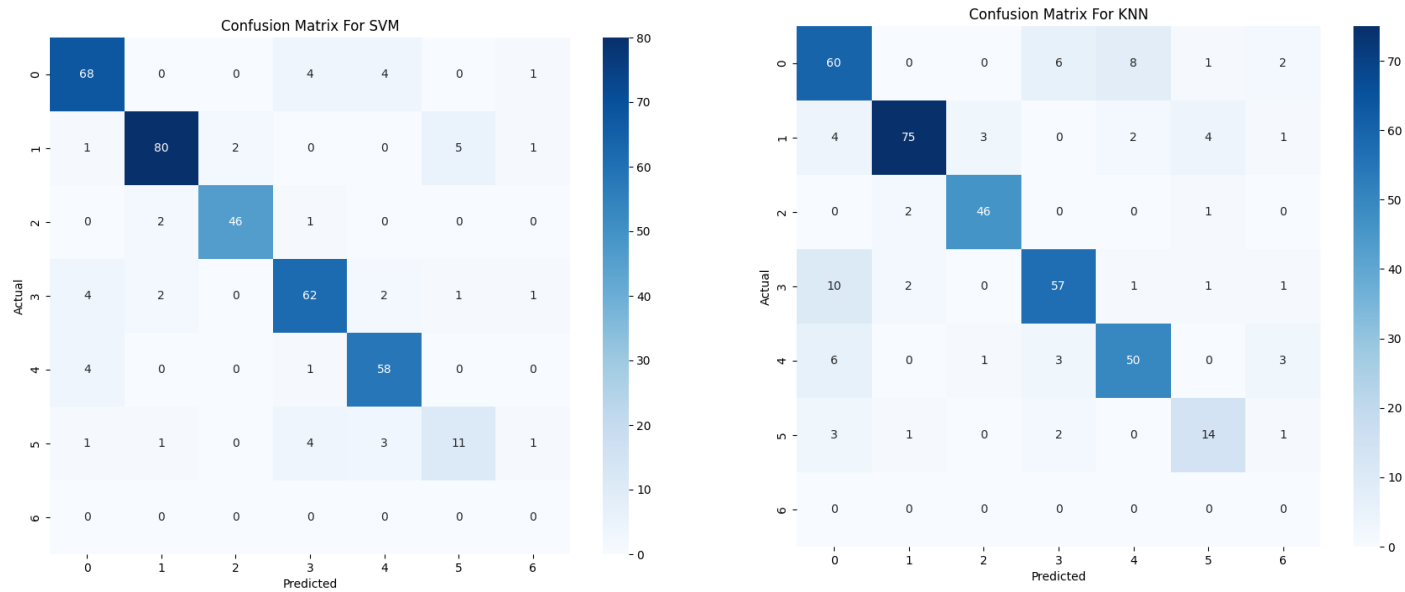
3.1 Support Vector Machine (SVM)

Support Vector Machines (SVMs) are well suited for classification tasks involving high-dimensional feature spaces. This makes SVM particularly effective when used with **EfficientNet-B0 features (1280 dimensions)**. Hyperparameter tuning was performed to select the optimal values for **C**, **gamma**, and **kernel type**, ensuring the best trade-off between accuracy and generalization.

3.2 k-Nearest Neighbors (k-NN)

The k-Nearest Neighbors (k-NN) algorithm is a simple, instance-based classifier that was evaluated for comparison purposes. While easy to implement, its inference time grows with the size of the dataset, making it less suitable for real-time systems. However, it performed reasonably well when used with **ResNet-18 features (512 dimensions)**. Hyperparameter tuning was applied to determine the optimal number of neighbors (**k / n_neighbors**).

4. Performance Comparison



Accuracy: 0.876010781671159					Accuracy: 0.8140161725067385				
Classification Report:					Classification Report:				
	precision	recall	f1-score	support		precision	recall	f1-score	support
0	0.87	0.88	0.88	77	0	0.72	0.78	0.75	77
1	0.94	0.90	0.92	89	1	0.94	0.84	0.89	89
2	0.96	0.94	0.95	49	2	0.92	0.94	0.93	49
3	0.86	0.86	0.86	72	3	0.84	0.79	0.81	72
4	0.87	0.92	0.89	63	4	0.82	0.79	0.81	63
5	0.65	0.52	0.58	21	5	0.67	0.67	0.67	21
6	0.00	0.00	0.00	0	6	0.00	0.00	0.00	0
accuracy			0.88	371	accuracy			0.81	371
macro avg	0.74	0.72	0.73	371	macro avg	0.70	0.69	0.69	371
weighted avg	0.88	0.88	0.88	371	weighted avg	0.84	0.81	0.82	371

5. Discussion

The experimental results show that **CNN-based and hybrid feature representations** achieved the highest classification accuracy. However, this performance gain comes at the cost of increased computational requirements and a stronger dependency on deep learning frameworks. These factors limit their practicality for **lightweight or resource-constrained deployments**, where simplicity, speed, and ease of integration are critical considerations.

6. Conclusion

Based on experimental evaluation and the imposed system constraints, the **hybrid CNN + LBP feature representation combined with an SVM classifier** was selected for final deployment. This configuration provided a strong balance between accuracy and efficiency, delivering **reliable and consistent performance** while remaining feasible for practical implementation.