# Project: Material Stream Identification System

## Project Instructions:

- The **maximum** number of students in a team is **5** and the **minimum is 3**.

- Team members must be from the **same lab** (or have the same TA).

- All team members must understand **all** parts of the project.

- **No late submission** is allowed.

- Only **one** team member should upload a **zip** file following the **naming convention:** *TAName_Group_ID1_ID2_ID3_ID4*

- A **penalty** will be **imposed for violating** any of the assignment rules **or missing** any deliverable.

- **Cheaters will get ZERO** and no excuses will be accepted as per the attached **"Plagiarism Scope"** document.

## 1. Introduction

The efficient and automated sorting of post-consumer waste is a critical bottleneck in achieving circular economy goals. This project challenges students to develop an **Automated Material Stream Identification (MSI) System** using fundamental Machine Learning (ML) techniques. It emphasizes mastery of the entire ML pipeline: **Data Preprocessing**, **Feature Extraction**, **Classifier Training,** and **Performance Evaluation.**

## 2. Project Objectives

The primary goal is the end-to-end implementation of a feature-based vision system.

a) **Data Augmentation and Feature Extraction:** Design and implement a pipeline to convert raw images into a fixed-size, numerical feature vector. This conversion from pixel space to feature space is essential for all subsequent classification steps.

b) **Classifier Implementation:** Implement and train **two** distinct foundational ML classifiers capable of classifying the feature vectors into **seven distinct material categories** (six primary classes + one negative class):
   - **Variant A: Support Vector Machine (SVM)** classifier.
   - **Variant B : k-Nearest Neighbors (k-NN)** classifier.

c) **Architecture Comparison:** Analyze and **report** on the trade-offs between the two classifiers (**SVM vs k-NN**) and the different chosen feature extraction methods.

d) **Robust Classification:** Try to achieve a minimum validation accuracy of 0.85 across the six primary classes.

e) **System Deployment:** Integrate the best-performing model into a functional application that processes **live camera frames** in real-time, and displays the classification result.

# 3. Data and Material Classes

## 3.1. Redefined Material Classes

The model must classify all input images into one of the following seven classes based on the features extracted from the image.

| ID | Common Name | Description |
|---|---|---|
| 0 | Glass | Items made of amorphous solid materials, primarily silicates (e.g., bottles, jars). |
| 1 | Paper | Thin materials made from pressed cellulose pulp (e.g., newspapers, office paper). |
| 2 | Cardboard | Heavy-duty structured material composed of multiple layers of cellulose fiber. |
| 3 | Plastic | Items made from high-molecular-weight organic compounds (e.g., water bottles, film). |
| 4 | Metal | Items made of elemental or compound metallic substances (e.g., aluminum cans, steel scrap). |
| 5 | Trash | Miscellaneous non-recyclable or contaminated waste (e.g., organic matter, food packaging). |
| 6 | Unknown | This class is mandatory. It should represent out-of-distribution items or blurred inputs. |

## 3.2. Dataset Availability and Structure

Students are required to use the **attached dataset** for training and validation. The dataset is structured as follows:

- It contains a dedicated and representative set of image examples for the first six defined classes.
- Images are organized into separate folders corresponding to their class labels.

# 4. Technical Requirements

## 4.1. Data Augmentation

Students **must** apply data augmentation techniques to the provided dataset to artificially increase the training sample size by a minimum of **30%**. This is a mandatory step to improve model generalization and robustness against variations in lighting, orientation, and scale. Students **must** select and justify the augmentation techniques used (e.g., rotation, flipping, scaling, and color jitter) in their technical report.

## 4.2. Feature Extraction (Image to Vector Conversion)

Students **must** define and implement a methodology to convert the raw 2D or 3D image data into a 1D numerical feature vector (a fixed-length list of numbers). This step is crucial and requires students to search and justify their choice of feature descriptors.

## 4.3. Model Architecture and Implementation

- **Frameworks:** Students should use appropriate **libraries** capable of implementing the required ML models, with a strong focus on utilizing fundamental algorithms.

- **Support Vector Machine Requirement:** The SVM classifier must be designed to accept the extracted feature vector as input. Students must select and justify the optimal **architecture elements.**

- **k-Nearest Neighbors Requirement:** The k-NN classifier must be designed to accept the extracted feature vector as input, and the weighting scheme (e.g., uniform, distance-based) for the classifier.

## 4.4. Data Handling: Handling the "Unknown" Class (ID 6)

Students must implement a rejection mechanism tailored to each model, as the system should only classify items it is confident about.

## 4.5. System Deployment

Integrate the best-performing model into a **functional application** that processes **live camera frames** in real-time, and displays the classification result.

# 5. Deliverables and Evaluation

## 5.1. Submission Checklist

a) **Source Code Repository:** A well-documented Git repository containing all code required to:
   - Implement the data preparation/preprocessing pipeline
   - Train and save the SVM and K-NN models
   - Run the final real-time classification application using a live camera feed
b) **Trained Model Weights:** The final saved classifier files (e.g., using Python serialization or model-specific file formats). **This is the file submitted for the competition** *(refer to section 6 for more details about the competition).*
c) **Comprehensive Technical Report (PDF):** A formal document including a section comparing the chosen feature extraction methods and classifier performance.

## 5.2. Evaluation Criteria

| Criterion | Weight | Details |
|---|---|---|
| Feature extraction and data autgmentation | 4 marks | Quality, appropriateness, and justification of the chosen feature vector methodology. Demonstration of the image-to-vector conversion process.<br>Making all class counts nearly the same size, e.g., 500, using data augmentation. |
| Theoretical understanding | 3 marks | Depth of explanation of the feature descriptor and classifier properties, SVM kernel choice, k-NN choices, etc. |
| **Competition Score (Hidden Test Set)** | **2 marks** | **Grade component directly tied to the final rank achieved on the Hidden Test Set Leaderboard.** |
| System Deployment (Real-Time) | 3 marks | Evaluation of the live camera application's stability and real-time processing speed. |

## 6. Hidden Data and Competition Mechanism *(Critical Requirement)*

- **Hidden Test Set (The Competition):** The primary purpose of the small, highly challenging **hidden test set** is for the final competition ranking. Students will submit their **best-performing** trained model file for evaluation against this unseen, private dataset.
- The highest accuracy achieved on the Hidden Test Set will receive the top rank and maximum points for the **competition score** evaluation criterion. We may measure based on the accuracy, for example if there are 10 test cases and 8 correct predictions, you may get 8/10 × 2 marks.
- **A private leaderboard** may be maintained and updated during the evaluation period, displaying the accuracy achieved by each team.