Garbage Classification with Machine Learning

AI & Data Science Graduation Project

Prepared by:
Mohamed Mahdy Mohamed Abd El-Monem
Menna Khaled Saad
Rawan Mohamed Taha
Hagar Essam
Mohamed Abdelwahed Mohamed

To:
Eng. Mostafa Badr

Table of Contents

1. Project Overview

The Garbage Classification project aims to leverage machine learning techniques to develop an automated system that categorizes waste materials into classes such as paper, plastic, metal, and glass. Utilizing publicly available datasets, the project delivers an end-to-end pipeline from data preprocessing to deployment. This contributes to the growing demand for efficient waste management by enabling accurate classification systems, suitable for integration into smart recycling solutions.

2. Milestone 1: Data Collection and Preprocessing

2.1 Data Sources

- Describe where the image data will be obtained:
  - Web scraping (Bing Image Crawler).
  - Existing datasets (if any).
  - Manual collection (if applicable).

2.2 Data Collection Procedure

- Explain the steps taken to gather the data:
  - Keywords used for web scraping.
  - Number of images collected per class.
  - Handling of data source limitations.

2.3 Data Preprocessing

- Detail the steps taken to prepare the images for model training:
  - Resizing.
  - Normalization.
  - Data augmentation (if used).
  - Splitting into training, validation, and test sets.
  - Handling class imbalance.

2.4 Tools and Technologies

- List the tools used:
  - Python libraries (Pillow, OpenCV).
  - Web scraping tools (icrawler).

2.5 Deliverables

- Specify the expected output of this milestone:
  - A structured dataset of preprocessed images.
  - Scripts for data collection and preprocessing.
  - Documentation of the data collection process.

3. Milestone 2: Advanced Data Analysis and Feature Engineering

3.1 Exploratory Data Analysis (EDA)

- Describe the analysis performed on the image data:
  - Visualization of sample images from each class.
  - Distribution of classes in the dataset.
  - Image size variations.
  - Other relevant image properties (e.g., color distribution).

3.2 Feature Engineering (If Applicable)

- Describe any feature engineering techniques used. For CNNs, this might be minimal, but could include:
  - PCA for dimensionality reduction (if needed).
  - Feature extraction from pre-trained models (if used).

3.3 Statistical Analysis

- Describe any statistical tests performed
  - Kolmogorov-Smirnov test
  - Wasserstein Distance

3.4 Visualization Tools

- List the tools used for visualization:
  - Matplotlib.
  - Seaborn.
  - Plotly.

3.5 Deliverables

- Specify the expected output of this milestone:
  - EDA report with visualizations.
  - Documentation of feature engineering techniques.
  - Justification for chosen data preprocessing and feature engineering methods.

4. Milestone 3: Model Development and Optimization

4.1 Model Selection

- Describe the machine learning models considered:
  - CNN-RNN hybrid model.
  - Other models (if explored): CNN, ANN, GoogLeNet
- Justify the choice of the CNN-RNN model.

4.2 Model Architecture

- Provide a detailed description of the CNN-RNN architecture:

        - CNN backbone (EfficientNet).
        - RNN layers (GRU).
        - Feature refinement layers.
        - Classifier layers.
        - Diagram of the model architecture.

4.3 Training Procedure

- Explain the model training process:
    - Loss function (Cross-Entropy Loss).
    - Optimizer (AdamW).
    - Learning rate schedule (OneCycleLR).
    - Batch size.
    - Number of epochs.
    - Use of early stopping.
    - Data augmentation during training.
    - Mixed precision training.
    - Gradient clipping

4.4 Hyperparameter Tuning

- Describe the hyperparameter tuning process:
    - Tool used (Optuna).
    - Hyperparameters tuned (learning rate, dropout, etc.).
    - Search space for each hyperparameter.
    - Objective function (validation accuracy).
    - Pruning techniques (if used).
    - Number of trials.

4.5 Evaluation Metrics

- Define the metrics used to evaluate model performance:
    - Accuracy.
    - Precision.
    - Recall.
    - F1-score.
    - Confusion matrix.
    - ROC AUC.
    - Balanced Accuracy

4.6 Results and Analysis

- Present the model's performance on the validation and test sets.
- Analyze the results:
    - Strengths and weaknesses of the model.
    - Confusion matrix analysis.
    - Impact of different hyperparameters.
    - Comparison with other models (if applicable).

4.7 Tools and Technologies

- List the tools used:
  - PyTorch.
  - Optuna.
  - MLflow (for experiment tracking).

4.8 Deliverables

- Specify the expected output of this milestone:
  - Trained CNN-RNN model.
  - Documentation of the model architecture and training process.
  - Hyperparameter tuning report.
  - Evaluation results and analysis.

5. Milestone 4: Deployment and Monitoring

5.1 Deployment Platform

- Specify where the model will be deployed (e.g., local server, cloud platform).
  From the code, it seems like a local deployment using Dash.

5.2 Web Application Development

- Describe the development of the web application:
  - Framework used (Dash).
  - UI design and features.
  - Image upload functionality.
  - Integration of the trained model.
  - Display of classification results.

5.3 Deployment Procedure

- Explain the steps to deploy the application.

5.4 Monitoring and Maintenance

- Describe how the system's performance will be monitored after deployment.
- Discuss plans for model updates and maintenance.

5.5 User Documentation

- Describe the documentation provided to the users.

5.6 Tools and Technologies

- List the tools used:
  - Dash.
  - Other deployment-related tools (e.g., Docker).

5.7 Deliverables

- Specify the expected output of this milestone:
  - A deployed and functional web application.
  - User documentation.
  - Deployment scripts and instructions.
  - Monitoring plan.

6. Milestone 5: Final Documentation and Presentation

6.1 Final Documentation

- Compile all previous milestone deliverables into a comprehensive final report.
- Ensure the documentation is clear, concise, and well-organized.

6.2 Presentation

- Prepare a presentation summarizing the project:
  - Project goals and objectives.
  - Methodology used.
  - Key findings and results.
  - Demonstration of the web application.
  - Conclusion and future work.

6.3 Deliverables

- Specify the expected output of this milestone:
  - Final project report.
  - Presentation slides.

7. Conclusion

- Summarize the project's achievements and outcomes.
- Reflect on the project's success in meeting the initial goals and objectives.
- Discuss any challenges encountered and lessons learned.

8. Future Improvements

- Outline potential future improvements to the system:
  - Improve model accuracy with more data or advanced architectures.
  - Expand the number of garbage categories.
  - Implement real-time classification from video streams.
  - Integrate with robotic sorting systems.
  - Deploy on a cloud platform for scalability.
  - Add user feedback mechanisms.
  - Explore different front-end frameworks.