E-Waste Segregation using Random Forest

1. Title Page
Project: E-Waste Segregation using Random Forest
Author: Akshata venugopal
Institution: VIT BHOPAL
Date: 25th november

---

2. Abstract
(Short summary of problem, approach, dataset, and key results.)

---

3. Introduction
- Motivation: electronic waste volume, hazards, need for automated sorting.
- Objective: build a simple image classifier to separate common e-waste items.

---

4. Literature Review (short)
- Mention baseline approaches: SVM, Random Forest, CNNs; conclude transfer learning is state-of-the-art for images.

---

5. System Requirements
Functional
- Train model from folder-structured dataset.
- Predict class for an input image.
- Save/load model and label encoder.

Non-functional
- Performance: inference within ~1s (on CPU).
- Maintainability: modular code.
- Usability: notebook interface with clear steps.
- Reliability: deterministic random state.

---

6. Dataset
- Source: uploaded archive (local path: `/mnt/data/extracted/modified-dataset`).
- Structure: `train/`, `test/`, `val/` with class subfolders.

- Number of classes: N (list classes).
- Images per class: mention counts (replace with actual numbers from your run).

---

7. Methodology
- Preprocessing: resize to 128×128, convert grayscale to RGB, remove alpha channel, normalize.
- Feature extraction: flatten pixel values to a 1-D vector.
- Model: Random Forest with 200 trees.
- Training: default hyperparameters except `n_estimators=200`, `random_state=42`.
- Evaluation: classification report, confusion matrix, accuracy.

---

8. Implementation
- Include code snippets or note: notebook cells used for each step.
- File structure overview.

---

9. Results
- Overall accuracy: (paste accuracy after running)
- Per-class precision/recall/f1: (paste classification report)
- Confusion matrix: (insert plot screenshot)

---

10. Discussion
- Analyze which classes are confused and why (e.g., similar appearance).
- Limitations: small dataset, RF loses spatial info.
- Remedies: data augmentation, larger dataset, transfer learning.

---

11. Future Work
- Replace flattened-features RF with transfer learning (MobileNet/ResNet + fine-tuning).
- Deploy as Flask API or mobile app for live sorting.
- Add more classes & balanced data.

---

12. Conclusion
- Short summary of results, feasibility, and next steps.

---

13. References
- Cite datasets, scikit-learn, scikit-image, any papers you referenced.