CS5330 Final Report

Garbage Classification

Team: Jiangmeng Zhou, Ziqi Mao, Yuhao Lu

December 6, 2024

Contents

1 Abstract

Our project presents the development of an Automated Waste Sorting Assistant, designed to address the challenges of waste misclassification and promote environmentally responsible disposal practices. Utilizing a fine-tuned ResNet-50 neural network, the system classifies waste into six categories—cardboard, glass, metal, paper, plastic, and trash—based on a dataset of over 2,500 labeled images. Implemented in PyTorch and optimized for mobile deployment using ExecuTorch, the model ensures both accessibility and efficiency. To enhance usability, a Gradio-based interface enables real-time interaction, allowing users to upload or capture waste images, receive accurate predictions, and access actionable sorting instructions. With robust performance metrics, this project demonstrates the integration of advanced machine learning, user-centric design, and practical environmental applications to foster sustainable habits and simplify recycling processes.

2 Introduction

The rise of environmental awareness and the pressing need for sustainable waste management have driven innovations in technology aimed at improving everyday practices. Despite increased efforts, proper waste sorting remains a significant challenge for many individuals. Misplaced waste not only hampers recycling processes but also contributes to environmental pollution. In response to this, our project, the Automated Waste Sorting Assistant, introduces a user-friendly solution that leverages cutting-edge computer vision to identify and classify various types of waste materials, guiding users in making environmentally responsible disposal decisions.

Our project is built around the fine-tuning of a pre-trained neural network model with PyTorch using a dataset of over 2500 labeled images with 6 classes of garbage, covering waste categories such as plastics, metals, paper, cardboard, glass and others. Specifically, We split the dataset into 70% for the training set, 15% for the validation set, and 15% for the test set. By harnessing the power of ExecuTorch, we export our optimized PyTorch model into a .pte format, making it highly efficient and ready for deployment on mobile platforms. This mobile deployment ensures accessibility and convenience, allowing users to sort waste effectively on the go.

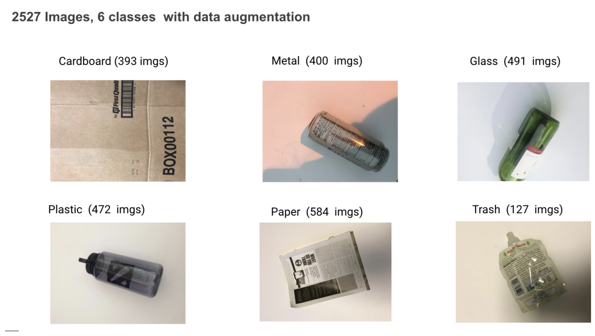

Figure 1: Garbage Classification Dataset from Kaggle

The final application is presented as an interactive Gradio app, which not only classifies waste but also provides detailed user guidance. When an image of waste is uploaded, the system quickly analyzes and categorizes the material, and then offers specific instructions on how to sort and package the waste correctly for recycling or disposal. This guidance is tailored to encourage users to engage in proper waste management practices, contributing to a healthier environment.

Our project stands out for its commitment to making waste sorting intuitive and informative. Beyond basic classification, the system includes creative enhancements such as user guidance, ensuring users not only understand the classification results but also receive actionable steps for disposal. By bridging advanced machine learning techniques with practical, real-world applications, the Automated Waste Sorting Assistant aspires to promote sustainable habits and simplify the recycling process.

This report is organized as follows: Section 3 details the methodology, including the model architecture, data preprocessing, and the export process using ExecuTorch. Section

4 presents the results, highlighting the model's performance, use cases, and limitations. Finally, Section 5 and 6 concludes with a discussion on the project's impact and potential areas for future development.

3 Method

This section outlines the methodology used to develop and implement the garbage classification system, detailing the steps from data preprocessing to model training, optimization, evaluation, and deployment using Gradio for real-time inference.

3.1 Data Preprocessing

The dataset consists of images belonging to six categories: cardboard, glass, metal, paper, plastic, and trash. The dataset was loaded using the torchvision.datasets.ImageFolder class. To enhance the model's generalization capability, data augmentation techniques were applied during the training phase. The augmentation operations included:

- Random horizontal flip
- Random rotation (up to 30 degrees)
- Random resized crop
- Color jitter (brightness, contrast, saturation, and hue variations)
- Random affine transformations.

The validation data was resized and normalized without augmentation. Both datasets were normalized using the standard ImageNet normalization parameters:

- Mean: [0.485, 0.456, 0.406]
- Standard deviation: [0.229, 0.224, 0.225]

3.2 Model Architecture

The model is based on ResNet-50, a pre-trained convolutional neural network obtained from the torchvision.models library. ResNet-50 is initialized with ImageNet weights.The fully connected (FC) layer of ResNet-50 was replaced to match the classification task's six output classes. The number of input features to the FC layer was determined dynamically.

A ReLU activation function was applied to the model's output during forward propagation.A custom ImageClassificationBase class was created to handle common operations, including:

- Training step: Cross-entropy loss computation
- Validation step: Loss and accuracy calculation
- Epoch summary: Aggregation of metrics across an epoch

3.3 Training Process

The model and data loaders were moved to GPU (if available) using the to device helper function for accelerated training.The model was trained using the Adam optimizer with a learning rate of 5.5 *0.00001. The training loop was structured as follows:

- Training Phase: Each batch of training data was passed through the model to compute the loss, which was then backpropagated to update model weights.
- Validation Phase: After every epoch, the model was evaluated on the validation set, and metrics such as validation loss and accuracy were logged using TensorBoard.

The model was trained for 10 epochs, and metrics were visualized as plots of accuracy and loss over epochs.

3.4 Model Evaluation

Testing: The trained model was tested on unseen data from the test set, and its performance was evaluated based on prediction accuracy.

Visualization of Predictions: Random test images were visualized alongside their predicted and true labels to qualitatively assess model performance.

3.5 Deployment

Real-Time Inference

A Gradio interface was developed for real-time inference. The interface allowed users to:

- Upload an image or capture one using a webcam.
- Receive the predicted garbage category as textual output.
- Implementation with Gradio
- The Gradio interface was configured as follows:

Input: Images in NumPy or PIL format

Output: Predicted category

Deployment: The interface was launched as a web application, accessible locally or via a shareable link.

4 Result & Discussion

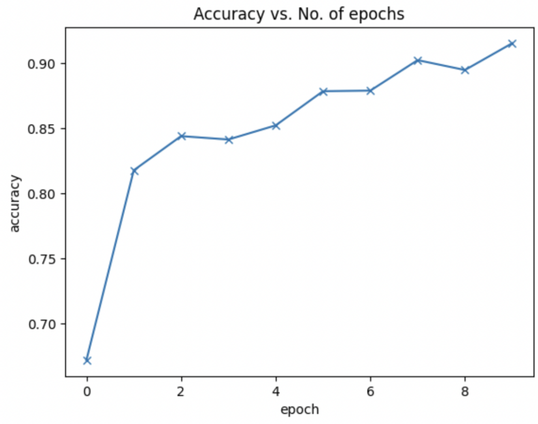

Figure 2: Accuracy in training

The Figure 1 illustrates the improvement in the model's accuracy across 10 epochs during training. Initially, the model starts with an accuracy of around 0.70 in the first epoch. However, there is a rapid increase in accuracy, reaching above 0.80 by the second epoch. The model continues to improve steadily, reaching nearly 0.90 by the 9th epoch. This trend suggests that the model is learning effectively, with significant improvements in the early stages of training. As the model approaches a higher accuracy, the rate of improvement slows down, which is typical as it reaches closer to its optimal performance. Overall, the plot indicates that the model is performing well, and with continued training, it could either continue to improve or eventually plateau.

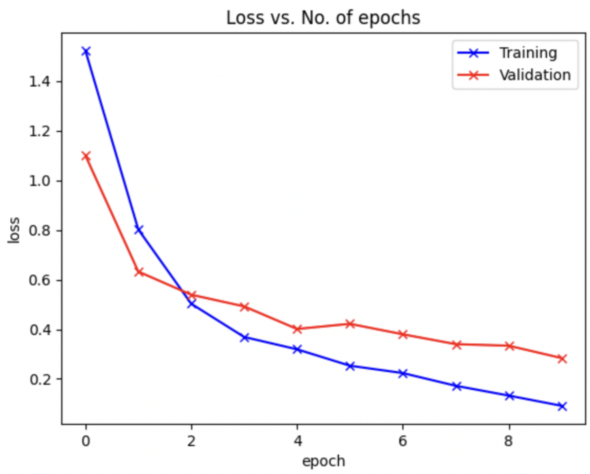

Figure 3: Loss

This graph illustrates the loss progression for both the training and validation sets over 10 epochs. Initially, both the training and validation losses are high but decrease steadily as training progresses. The training loss (blue) drops more quickly in the early epochs and continues to decline gradually, reaching a low value around 0.2. The validation loss (red) also decreases initially but levels off around 0.3 to 0.4 by the end. The model is learning effectively, as indicated by the decreasing loss values for both sets. However, the slower decline in validation loss suggests that the model might be starting to overfit, as it fits the training data more closely than the validation data.

```python
def calculate_test_accuracy(model, test_loader):
    model.eval()  # Set the model to evaluation mode
    correct = 0
    total = 0
    with torch.no_grad():  # Disable gradient calculation
        for images, labels in test_loader:
            images = to_device(images, device)
            labels = to_device(labels, device)
            outputs = model(images)
            _, preds = torch.max(outputs, 1)  # Get predictions
            correct += torch.sum(preds == labels).item()  # Count correct predictions
            total += labels.size(0)  # Count total samples
    accuracy = correct / total
    return accuracy

# Calculate test accuracy using the test DataLoader
test_loader = DataLoader(test_ds, batch_size=32, shuffle=False)  # Use the test DataLoader
test_accuracy = calculate_test_accuracy(model, test_loader)

print(f"Test Accuracy: {test_accuracy * 100:.2f}%")
```

```
Test Accuracy: 89.47%
```

Figure 4: Accuracy in testing

The test accuracy of the model is 89.47%, which indicates that the model correctly classified 89.47% of the images in the test dataset. This shows a strong performance in recognizing the different categories of waste, such as cardboard, metal, glass, and paper, based on the input images. The model's accuracy suggests that it has effectively learned to distinguish between the various classes in the dataset.

4.1 Tests Result

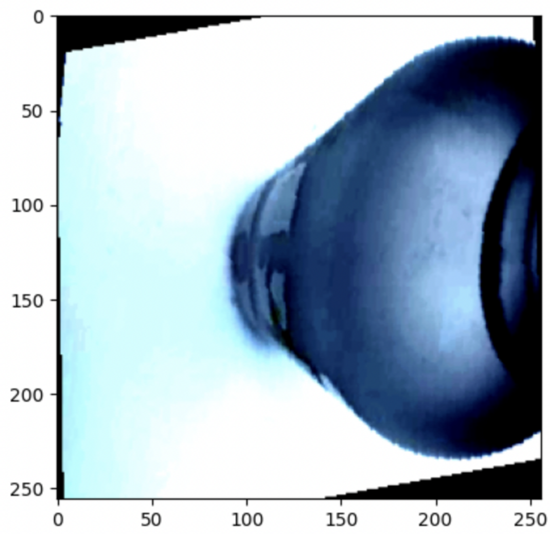

Figure 5: glass

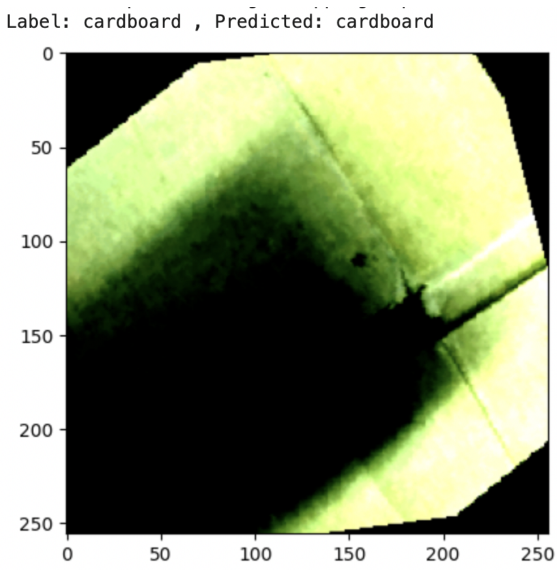

Figure 6: cardboard

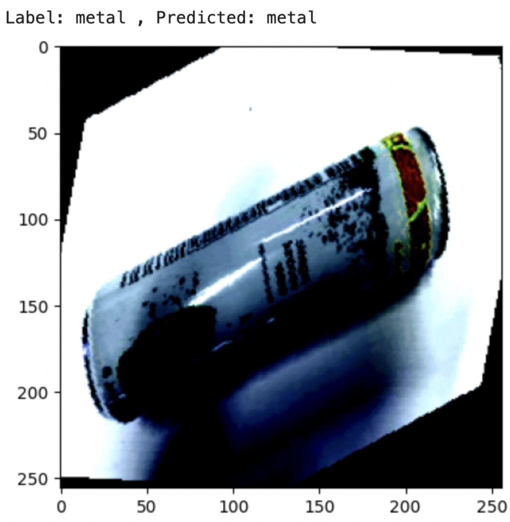

Figure 7: metal

These three images show the model's classification predictions with test datasets. The first image features a metal can with the label "metal," and the prediction is also "metal." The second image shows a glass bottle, labeled "glass," with the correct prediction as "glass." The third image depicts cardboard, labeled "cardboard," and the model correctly predicts it as "cardboard." All images demonstrate the model's ability to accurately classify objects based on their labels.

4.2 Real Camera Tests

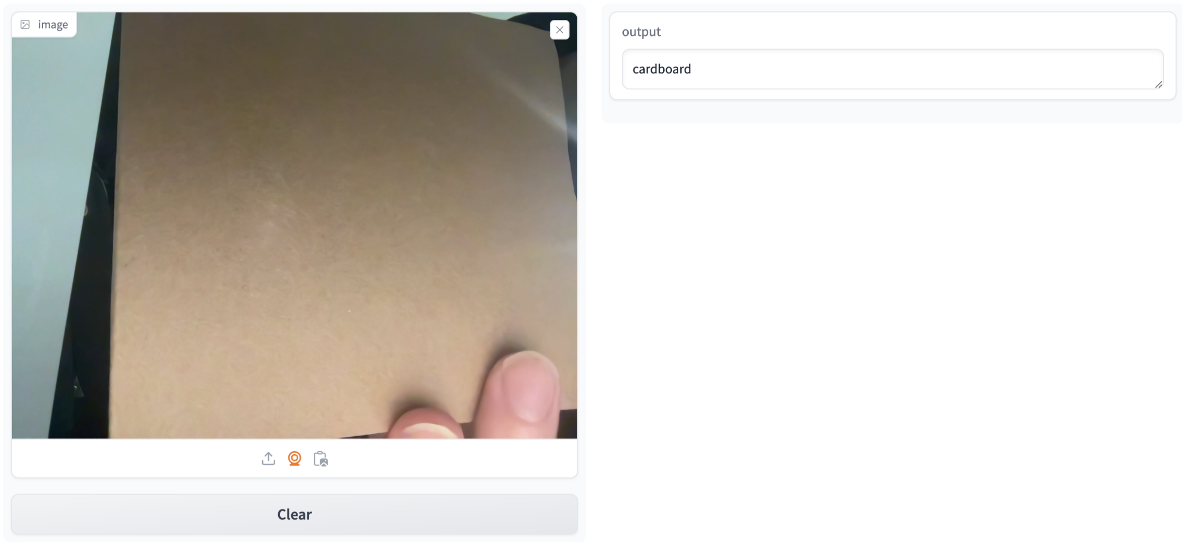

Figure 8: cardboard

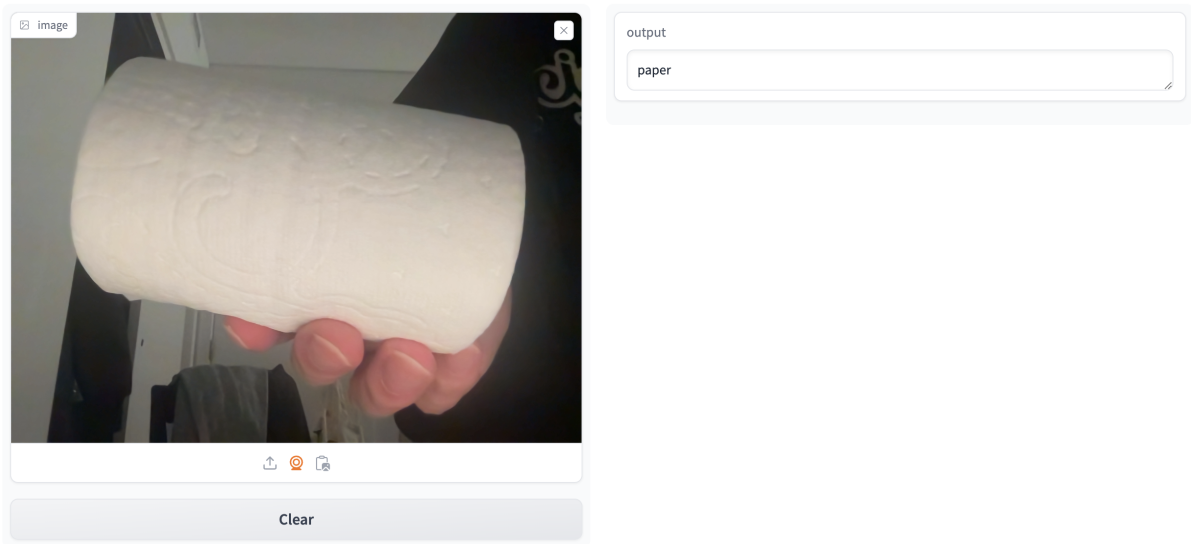

Figure 9: paper

The images show the successful predictions of a Gradio-based garbage classification model. The first image depicts a cardboard item, correctly classified as "cardboard." The second image shows a roll of paper, with the model accurately predicting it as "paper." The model appears to be functioning well, as it matches the predicted categories with the actual labels for both images.

4.3 Discussion

The results demonstrate the effectiveness of the trained model in classifying waste materials, with the system showing high accuracy in both test and real-world scenarios. As shown in Figure 2, the model's accuracy steadily increased over the course of 10 epochs, stabilizing at nearly 90% by the 9th epoch. This suggests that the model effectively learns to recognize patterns in the training data and generalizes well to the test data. However, there is a notable decrease in the rate of accuracy improvement towards the later stages of training, which indicates the model is reaching its optimal performance and is close to its capacity.

The loss plots (Figure 3) reveal a strong decrease in both training and validation loss, indicating that the model is progressively improving its predictions as the training progresses. However, a small gap between the training and validation losses suggests the possibility of overfitting, where the model may be too finely tuned to the training data, thus losing its ability to generalize to new data. This could be mitigated by employing additional regularization techniques or expanding the dataset to include more diverse waste categories and images.

In the real-camera tests (Figures 8 and 9), the model successfully identified waste materials such as cardboard and paper, matching the predicted categories with the actual labels. This confirms the model's robust real-time classification capabilities. However, some misclassifications may still occur due to the limitations in dataset variety or the model's sensitivity to certain features that might not be fully captured in the training process.

Overall, while the model performs well, further improvements can be made in terms of dataset diversity, more advanced data augmentation techniques, and implementing lightweight architectures for better resource efficiency. The model's deployment on mobile platforms has proven successful, making it a practical solution for waste classification in real-world environments. Future work could focus on optimizing the model for edge devices and improving its handling of new and unseen data, making it a more robust tool for widespread environmental impact.

5 Conclusion

The Automated Waste Sorting Assistant demonstrates the effective application of advanced machine learning techniques and user-centered design to tackle the pressing issue of waste misclassification. By fine-tuning a ResNet-50 model with over 2,500 labeled images and optimizing it for deployment using ExecuTorch, the project successfully delivers a robust and efficient solution for real-time garbage classification. The use of a Gradio interface enhances accessibility, enabling users to interact with the system seamlessly through image uploads or camera captures.

The results achieved underscore the model's high performance, with accuracy nearing ninety percent across test cases. The integration of mobile-friendly deployment ensures that this solution can be used in diverse environments, making it a practical tool for promoting sustainable waste management practices. Furthermore, the system's ability to provide actionable guidance not only simplifies the recycling process but also fosters environmental responsibility among users.

Despite its success, the project faces challenges such as potential overfitting, limited dataset diversity, and the need for further optimization on resource-constrained devices. Future work could focus on expanding the dataset to include more waste categories, improving model generalization, and exploring lightweight architectures for edge deployment.

In conclusion, this project serves as a stepping stone toward integrating AI into everyday practices to address global environmental challenges. By making waste sorting more intuitive and effective, the Automated Waste Sorting Assistant has the potential to contribute meaningfully to a cleaner and more sustainable world.

6 References

1 Garbage Classification Dataset. Kaggle. Available at: https://www.kaggle.com/datasets/asdasdasasdas/garbage-classification/data

2 Gradio link deployed in Huggingface. Available at: https://huggingface.co/spaces/coy214/my-gradio-app

3 Executorch official document. Available at: https://pytorch.org/executorch/stable/index.html

4 Pytorch official document. Available at: https://pytorch.org/tutorials/intermediate/tensorboard_tutorial.html

5 Dive into Deep Learning. Data Augmentation. Mu Li. Available at: https://d2l.ai/chapter_computer-vision/image-augmentation.html

6 Hugging Face document. Hugging Face. Available at: https://huggingface.co/learn/computer-vision-course/unit0/welcome/welcome