WASTE IMAGE CLASSIFICATION WITH DEEP LEARNING TECHNIQUES

George Liu
Student# 1006849341
georgedev.liu@mail.utoronto.ca

Kathy Lin
Student# 1007620166
kath.lin@mail.utoronto.ca

Viet Minh Nguyen
Student# 1007503989
nvmichael.nguyen@mail.utoronto.ca

Chris Shih
Student# 1008298272
chris.shih@mail.utoronto.ca

ABSTRACT

This report will outline the progress made on the "Waste Image Classification with Deep Learning Techniques" project. It will include a description of the project, team member responsibilities, and any notable contributions and milestones completed by the time of this report, which include baseline models, data preprocessing, and initial model iterations.

—-Total Pages: 6

1 BRIEF PROJECT DESCRIPTION

With the rapid modernization and growth of the world's population, waste and pollutant generation has increased dramatically. The World Bank (Bank, 2022) estimates that the world's annual waste generation is poised to break 3.88 billion tonnes in 2050 - a 73% increase from 2020. Recycling has always been a great option to help us beat this statistic, but with a shockingly low recycling rate of 27% for discarded glass and 8% for discarded plastic (Cho, 2020), there is a clear need for improvement. Cho (2020) mentions a contamination risk if waste is deposited into the wrong bin or sent to the wrong facility, preventing large batches of materials from being recycled.

This project will seek to minimize this risk by leveraging deep-learning models for image classification. The neural network will be able to process an input image of a piece of waste and reliably classify the waste into five material classes — glass, plastic, paper, metal, and trash (non-recyclable materials). With this information, large batches of waste can now be separated and sent to their respective processing facilities. A sketch of the project idea is illustrated in Figure 1.

Deep learning, more specifically with Convolutional Neural Networks (CNNs), has become a popular choice for solving image classification problems. With the introduction of convolutional layers, the tedious task of extracting features from images (as seen in more traditional ML algorithms) can be done during the training itself, reducing overhead by allowing the model to determine the defining features for each image on its own. Additionally, with highly accurate and complex pre-built neural networks being so widely accessible, utilizing transfer learning becomes a possibility. All of these factors contributed to the decision to train a CNN for this task.

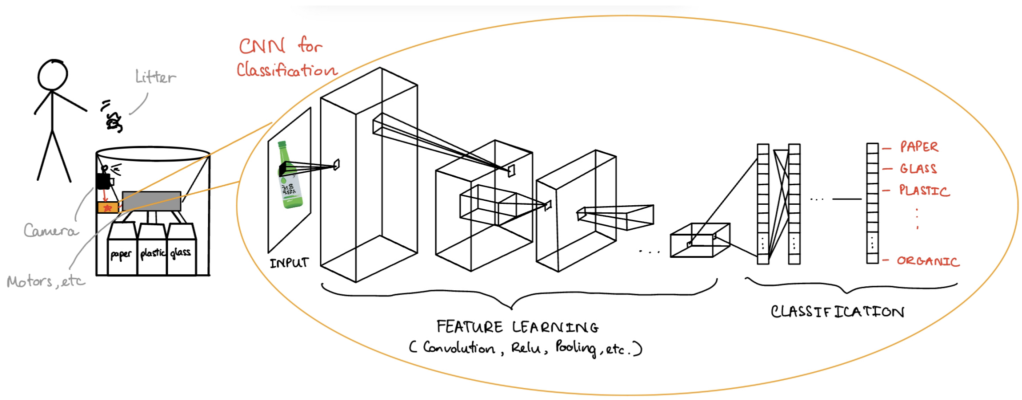

Figure 1: Project idea sketch

2 Individual Contributions and Responsibilities

2.1 Completed Tasks and How We Have Worked Together

We have finished data processing including data augmentation, trained and tested the baseline model, and started training the primary model.

Our team hosts a weekly online meeting on Thursday at 6 pm to review each other's progress and address any challenges encountered. Our team is using a group chat to communicate everything that is related to the project and provide regular updates on assigned tasks to make sure everyone understands what is going on in the team. Communication channels are split into task assignments, project resources, formal and informal messaging, etc. to avoid miscommunications. Additionally, all source codes are shared between all members through Google Colab, while files are stored in a group Google Drive folder.

Each member has communicated their progress clearly and has completed assigned tasks. Details about the completed tasks are listed below (PP stands for Project Plan and PR stands for Progress Report).

2.1.1 Tasks Chris Has Completed

- PP Abstract, Introduction, Background and Related Work: do research on related work and write the sections (completed on Oct. 13)
- Coding Baseline Model Architecture and Testing: code the baseline model, train the model, and test it to get a quantitative and qualitative result (completed on Nov. 3)
- PR Brief Project Description, Notable Contribution (Baseline Model): describe the project and the baseline model and discuss the testing result (completed on Nov. 3)

2.1.2 Tasks George Has Completed

- PP Data Processing, Architecture, Baseline Model: clearly document the data processing procedure and baseline and primary model to be used (completed on Oct. 13)
- Coding Data Processing: code data preprocessing part and data augmentation part (completed on Nov. 3)
- PR Notable Contribution (Data Processing): describe how data processing was done (completed on Nov. 3)

2.1.3 Tasks Michael Has Completed

- PP Illustrations, Risk Register: draw a figure that illustrates the project and describe how we handle contingencies (completed on Oct. 13)

- Coding Primary Model Training: code training part and plotting functions to get one qualitative and quantitative result (completed on Nov. 3)
- PR Notable Contribution (Primary Model Result): describe the training and testing result (completed on Nov. 3)

2.1.4 Tasks Kathy Has Completed

- PP Ethical Consideration, Project Plan, Colab Link: document ethical concerns of the project and project plan thoroughly (completed on Oct. 13)
- Coding Primary Model Architecture: code the primary model (feature extraction using transfer learning and classifier) (completed on Nov. 3)
- PR Individual Contributions and Responsibilities, Notable Contribution (Primary Model Architecture): describe what each member has completed and the architecture of the primary model (completed on Nov. 3)

2.2 Updated Task Distributions

The updated version of task distributions are listed in table 1 (FR stands for Final Report and VP stands for Video Presentation).

Table 1: Tasks, assigned person, and internal deadlines

Task	Assigned Person	Internal Deadline
Training the Primary Model	Michael	Nov. 15
Collect Test Data	All	Nov. 15
Test the Primary Model	Kathy	Nov. 22
FR Introduction, Background and Related Work	Chris	Nov. 29
FR Baseline Model	Chris	Nov. 29
FR Illustration, Quantitative Results, Qualitative Results	Michael	Nov. 29
FR Data Processing, Discussion	George	Nov. 29
FR Architecture, Evaluate model on new data	Kathy	Nov. 29
FR Ethical Considerations	Kathy	Nov. 29
FR Project Difficulty / Quality	Chris	Nov. 29
VP Problem	Michael	Nov. 29
VP Data Processing	George	Nov. 29
VP Baseline Model	Chris	Nov. 29
VP Primary Model	Kathy	Nov. 29
VP Results	Chris	Nov. 29
VP Discussion	Kathy	Nov. 29

3 Notable Contribution

3.1 Data Processing

The data we used for the project consists of the two datasets from Mohamed (2021) and Chang (2018).

To increase the diversity of our each examples and ensure a balanced class distribution, the two data sources were combined, de-duplicated, and organized into classes of paper, glass, metal, plastic, and trash. Any images that did not belong to these classes were removed. After processing, we obtain the following distribution for each class as shown in Table 2.

Table 2: Number of examples in each class after each data processing step

Dataset	Paper	Glass	Metal	Plastic	Trash	Misc	Total
After Merging Datasets	2938	2512	1179	1347	834	9323	18133
After Cleaning	1472	2023	718	782	697	0	5692
After Balancing	680	680	680	680	680	0	3400

To clean the dataset, we removed 4,904 duplicate files across the two data sources, some of which were due to exact matches inside the individual dataset, and others due to shared images between the two sources. Image sizes were center-cropped to 224x224 to standardize image sizes. For images smaller than 224x224, the images were upscaled and cropped to 224x224. Select examples are shown for clean and raw samples as shown in Figure 2.

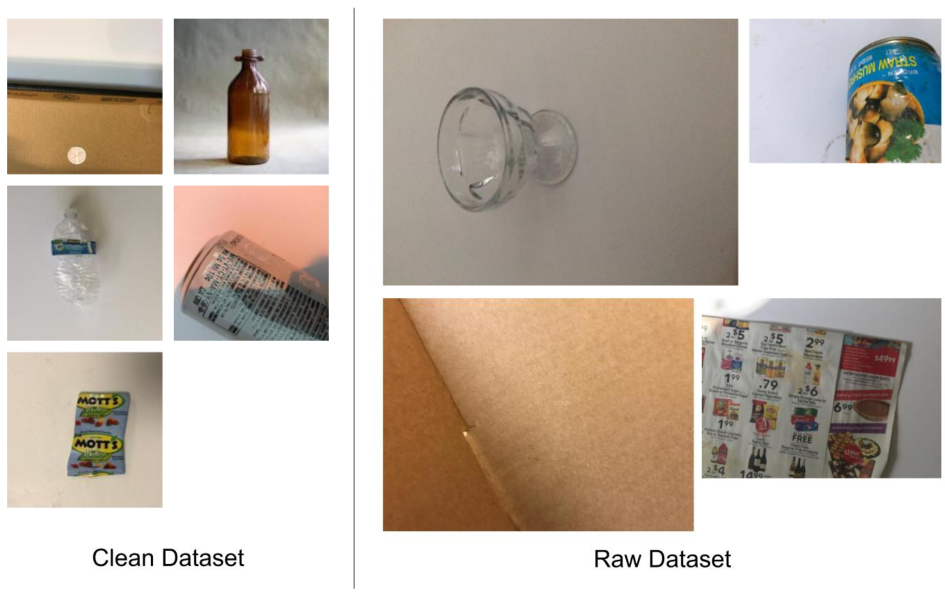

Figure 2: Clean vs. data samples

Since the differences between class distributions in each class is substantial, we balanced each class by not including data from larger classes so that each class has equal amounts of data.

To remove visually similar images, we attempted to use perceptual hashes which map similar images to similar 256-bit hashes, as implemented by the Linux package findimagedupes (Chin). However, since each image in the dataset were too similar to each other, we were unable to get useful visually similar image de-duplication as the system results in 238 unique images from a starting dataset of over 5,000 images, even with the smallest possible threshold. This results in many false positives as shown by the identified duplicates in Figure 3. Note that each row represents a group of images that have the same perceptual hash.

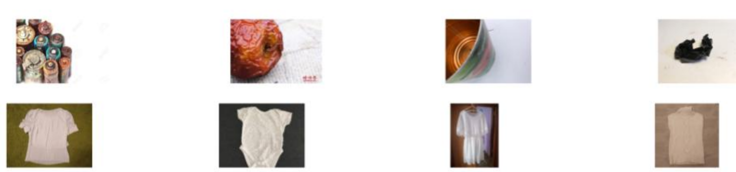

Figure 3: False duplicates as identified by findimagedupes

To avoid contamination of testing data with training data, the team will collect and prepare a new dataset whereby each member will photograph 30 images of garbage in order to have 24 test images for each class in total. This will then be processed by center-cropping the images to 224x224 and organized into their proper labels.

3.2 Baseline Model

The baseline model chosen for this project will be a traditional Support Vector Machine (SVM) machine learning algorithm, utilizing previously extracted image features to classify images. This algorithm utilizes the Scikit-Learn library, which provides simple APIs for a wide range of machine learning algorithms, including SVMs.

Due to the large size of the dataset and the lack of parallelization and GPU support in Scikit-Learn, the SVM was trained on a subset of our data, with only 150 images for each of the five classes (glass, paper, plastic, metal, trash). To further improve training times, the image data was compressed to 150x150x3, instead of the 224x224x3 size that was used to train the CNN. The image features were extracted by compressing the 3D image matrices into a single 1D array of pixel values.

Utilizing the Pandas library, these arrays were mapped to their corresponding labels, split into training and test sets, and processed by the SVM classifier. The baseline model achieved an accuracy of 54.7%. The confusion matrix for the testing data on our five classes is shown below (Figure 4):

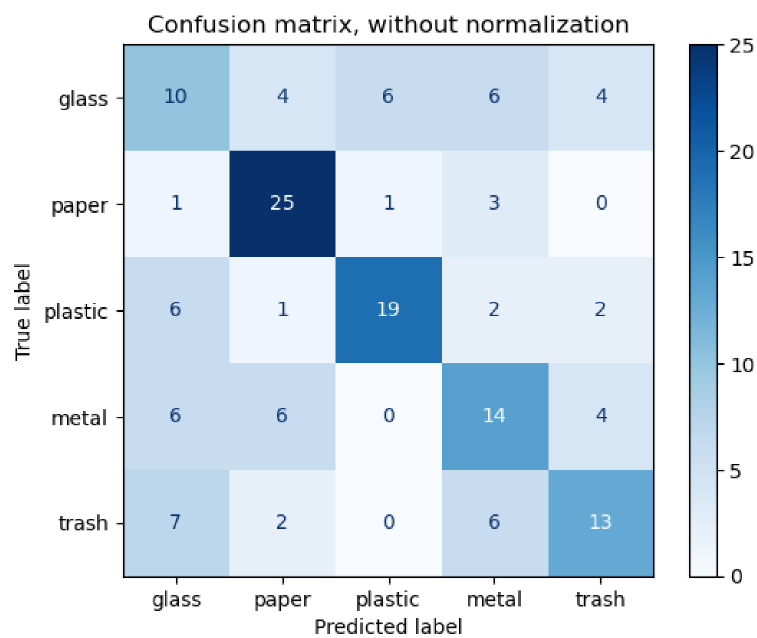

Figure 4: Confusion matrix for the SVM baseline model

This SVM classifier performs much better than a random process, which should theoretically result in a 20% accuracy. However, it still makes a considerable amount of errors during classification. As seen from the matrix, it predicts glass, plastic, metal, and trash quite poorly, but it manages to predict paper to a reasonable degree. This may be due to the nature of our dataset. A lot of the garbage in the four classes that the SVM struggled with had lots of colourful graphics and logos printed on them, as seen in Figure 5. Our naive method of extracting features could have reduced its ability to differentiate between the classes — for example, we have no method of extracting edges or any other defining characteristics. We may need to improve the robustness of our feature extraction methods in order to increase our accuracy for the baseline.

Figure 5: Similarly colorful graphics seen in multiple classes

One of the notable challenges we faced when training the SVM was, as mentioned before, finding a way for the model to train in a reasonable amount of time. With 150,528 input variables per image (224x224x3), the SVM would take substantial time to derive a hyperplane that can separate our five classes. Additionally, with no GPU support in Scikit-Learn, we needed to shrink our dataset and our images to compensate for this training overhead. This discrepancy between our baseline and our CNN datasets will be accounted for during evaluation further on in the project timeline.

3.3 Primary Model

We chose transfer learning to train our model since we only have 3,400 images in our cleaned dataset. We first pass images with dimension $224 \times 224 \times 3$ into various choices of pre-trained model, including: AlexNet, Inception, VGG16, VGG19, ResNet18, ResNet152. Due to time constraints and limited GPU-time, the team elected to limit testing to AlexNet for the initial experimentation. Our plan is to discover more hyperparameters for all models, and we will select the best one for our final result.

The AlexNet pre-trained model generates a tensor with shape $(256 \times 6 \times 6 \times 64)$, representing extracted features of an image. Given these extracted features, we will train our Classifier layers, which consist of 2 layers, each input and output size being $(256 \times 6 \times 6 \times 64)$ and (64×5). We use ReLU as the activation function. The last layer has 5 neurons which correspond to the 5 classes (Paper, Glass, Metal, Plastic, and Trash).

We tested our model with the AlexNet pre-trained model. The training/validation error/loss is recorded in Figure 6 and 7 respectively. As we can observe, the training error and loss approaches zero, meaning that the model is capable of learning the classification task. However, the validation error and loss stops decreasing over time, which could be indicative of overfitting. Therefore, our team will need to further improve the model by either trying out different pre-trained models, tweaking hyperparameters, or implementing regularization techniques to improve the performance.

Figure 6: Train vs Validation Error

Figure 7: Train vs Validation Loss

References

The World Bank. Solid waste management, 2022. URL https://www.worldbank.org/en/topic/urbandevelopment/brief/solid-waste-management#:~:text=In%202020%2C%20the%20world%20was,3.88%20billion%20tonnes%20in%202050. Accessed on October 11, 2023.

C Chang. Garbage classification, 2018. URL https://www.kaggle.com/datasets/asdasdasasdas/garbage-classification/data. Accessed on October 13, 2023.

Jonathan Chin. findimagedupes. URL https://github.com/jhnc/findimagedupes. Accessed on November 3, 2023.

Renee Cho. Recycling in the u.s. is broken. how do we fix it?, 2020. URL https://news.climate.columbia.edu/2020/03/13/fix-recycling-america/. Accessed on October 11, 2023.

Mostafa Mohamed. Garbage classification (12 classes), 2021. URL https://www.kaggle.com/datasets/mostafaabla/garbage-classification/data. Accessed on October 13, 2023.