RecycleAI

Materials Recycling using Machine Learning

Data-X: Fall 2017 | Final Project Report
By: Wesley Jin, Kenny Kang, Elias Orellana, Umesh Thillaivasan, Nate Tran

Objective: Classify waste objects into different recycling streams using machine learning

Introduction:

We aim towards a vision of a world without trash or waste, our project aims to train a computer vision system to classify waste objects in order to make the process of sorting of waste quick and economical - allowing for *all* waste streams to be processed for reusable and recyclable materials. We were able to achieve 86% accuracy on 6 waste classes (paper, plastic, glass, cardboard, metal, and trash) with limited data using convolutional neural networks. With more data and resources - we are confident we can build this up to an industrial-grade product.

Industry Mentor

A proof-of-concept model - "ReycleAI" - was developed by our project mentor Gerry Pesavento. We used this existing project for inspiration for our final product and improved on the execution. We were also provided a small database of labeled waste objects, but it was not very comprehensive for real use.

The Problem

Waste management is a growing problem across the globe. Landfills are quickly reaching capacity and tons of recyclable natural resources are being wasted. Environmental effects include groundwater leaching, demolition of environments to make space for landfills, and emission of greenhouse gases.There is a lot of potential to improve the current state of waste streams - according to the US Environmental Protection Agency, more than 40% of trash that ends up at landfills is recyclable. Trash put into "Landfill" bins goes straight to the landfill - with no processing in between. Additionally, waste placed in recycling bins are processed at a recycling plant - where both humans and machines sort out the materials once again. We believe our project has the potential to disrupt both of these current practices - bringing data and machine learning to a data-less industry.

Methodology

Overall System Architecture

The overall architecture of the RecycleAI system consists of a feedback loop. This allows Recycle AI to continuously update the weights of our machine learning model overtime for

improvement on the classification task (Figure 1 - appendix). This is done in a three-steps process:

1. An initial machine learning model is built using labeled material images to classify into 6 classes. This step will be described further in the following sections of this report.
2. The hardware for the vision system is built using a Raspberry Pi, a Raspberry Pi camera, and other supporting materials (LED/infrared light, motors, etc. - see Figure 2 - appendix) This system will take pictures of individual objects and will use the model loaded in the Raspberry Pi to predict the class of material it belongs to. If the prediction probability is greater than a certain threshold (currently set at .98), the object will be classified and stored in a database then moved into an appropriate bin. If not, the images of the object will be sent to a different database for further processing. A prototype of the system was built by our mentor and was improved upon with some modification from our team.
3. Images of objects that are not classified in step (2) will be stored in the cloud storage Amazon S3, and will be crowd-sourced via the RecycleAI website to identify the correct label. These labeled images will be added into the original image database, and the model will be re-trained with these images in step(1).

Tools & Resources

The following tools were utilized to develop the full RecycleAI systems:

- Raspberry Pi and Raspberry Pi camera
- Keras w/ TensorFlow backend - neural network models
- Jupyter Notebook - code development
- Floydhub.com - train using GPUs
- MySQL - database management
- Amazon S3 - cloud storage for images and website hosting
- Github & Google Drive - version control and collaboration

Data

Our material images database is consisted of 2527 images pulled from Trashnet database (reference 2) and 111 images generated by RecycleAI. We reclassify the images into 6 different classes: cardboard, paper, plastic, metal, glasses, and trash. We split the database into 75% training set and 25% validation set.

In order to enhance the database for our deep learning model, we perform several steps of images augmentation, including horizontal flipping, rotating, random cropping, and color jittering. This is both to add more images to the dataset as well as to prevent overfitting of CNN models.

Models

We developed four machine learning models. Validation accuracy (VA) for each of our model approaches are listed below.

1. Basic sequential CNN, used as baseline model (VA = 64%)
2. VGG16 pretrained model for feature extraction used to train SVM, Logistic Regression, Random Forest, and Decision Tree models (VA = 42%, 53%, 40%, 47% respectively)
3. Gerry's (project mentor) pre-trained model: extracting image features using InceptionV3 and train a CNN to classify images (VA = 77.6%)
4. Fine-tuned InceptionV3 model (VA = 86.76%)

* InceptionV3 & VGG16 are models trained on the ImageNet dataset. We used the weights from this model due to limited data & resources for training a robust model from scratch.

Sequential CNN(#1):

This was a baseline model to see if we were able to train a model from scratch we built a convolutional neural network with three convolutional layers with 'relu' activation and max pooling for dimensionality reduction, we added three dense layers after flattening the output of the convolutions. We achieved a %64 accuracy, and decided that it wasn't feasible to train with the amount of data we had.

Ensemble Models(#2):

Another approach we took to classify our images, was using an ensemble of different models. Using a pre trained model allowed us to featurize our data to input into other classification models. This is useful since we don't have to worry about overfitting a deep network with the small amount of data we had, instead using the the layers that are optimized to find gradually higher level features we are able to feed them in to a simpler model.

We chose *VGG16* with weights used on the Imagenet classification task, to extract the features from our images. We used and compare the performance of the following models: *Support Vector Machine, Logistic Regression, Decision Tree, and Random Forest*

This approach turned out to not perform well, with SVM perform at %42; logistic regression at %55 and %53. On the other hand, decision trees and Random forest were overfitting on the training set with scores of %100 and %40 and %99 and %47 respectively.

Inception V3 Transfer Learning (#3):

For this approach we took a Inception V3 pre-trained model weights to use to featurize the images. This time, we used another neural network with fully connected layers to train on the features and we were able to improved to a 77.6% validation set score.

Fine-Tuning Inception V3 (#4):

A key obstacle of this project is the lack of data we possessed with around 2,000 unique images, 10,000 if data augmentation is included. Therefore training a machine learning model from scratch, especially a convolutional neural network would not be optimal with such little data. Therefore we decided to use a model that already has amazing performance in object classification, Google's InceptionV3, and fine tune it to our dataset.

This approach is split into two steps. The first was to train a fully connected layer separately from the dataset, then to replace this layer with Inception V3's top layer and train the

entire model, including the intermediate weights of the pre-trained model. While the first step isn't as intuitive as the second, it is necessary in order to take advantage of the pretrained weights of the Inception model. If you were to place an untrained model at the top of Inception, the large gradients produced during back propagation would extend to the weights of the pre-trained model and ruin them With the limited dataset, this accuracy is not expected to be amazing as validation hardly surpassed 70%, however it still helps enormously in the final outcome. In order to avoid this problem, we must train the final layer separately first. During the second step, we selectively train the model on the later layers of Inception as these deal with high level features of an image. We choose an initial learning rate of 0.0001 and train over 30 epochs. We found slightly higher accuracy training the last half of the layers (86.76%) opposed to the last quarter (~84%).

Regarding improving this model, the final training accuracy was approximately (99.2%). This gap between training and validation accuracy typically implies that the model needs to be better at generalizing new data. This can be done in two ways, adding regularization or adding more training data. While there was some regularization added in the form of dropout layers, this model would benefit significantly more from more training data.

Challenges

The overall success of our project did not come without challenges along the way. Our mentor, Gerry Pesavento, provided us a starting point to the project by explaining his original progress, and donating the hardware and software he used. His work has since been dismantled and the links broken, so our first task was understanding his system and trying to rebuild his hardware and software system. This was challenging as there were so many complex components to understand such as the Raspberry Pi and Pi Camera, the Prediction and Training model, AWS S3, mySQL, and PHP and the website.

Other challenges was trying to build a model with very limited data as there were very few resources to build a model with. After we overcame that challenge, we simplified our task by creating a binary classification model of Recycling or Not; however, this did not add value as we were unable to identify what the model predicted the object was. We increased the classes to 6 in order to add utility and better intelligence to classification. Now our model would be distinguish between metal and plastic, or trash and glass, and therefore add needed value to ultimately classifying and making a decision about the object.

Finally, some objects could appear similar to other objects in appearance, but different in material composition which is important to distinguish. These instances (like plastic that looked like glass) would occasionally fool the model and give a false prediction.

Future works

Moving forward, we would like to collect more data and label them to train our model more accurately and improve our neural net models. We would also look to fine-tune our model more accurately.

We would also like to add additional classes to the model to incorporate waste streams like e-waste, hazardous waste, and specific types of metals and plastics. Since this solution would be ultimately for industrial applications, we would like to work on multiple object classification in one image to accommodate mixed waste streams.

Finally, we would like to utilize additional sensors like infrared and hyperspectral sensors to better identify material types without being fooled by physical appearance.

Conclusion

We were able to improved on our initial prototype by addressing problems such as data integration/augmentation, model selection and choosing better target classes for our classification task. However there are a lot of work and research to be done in the future, including improving our hardware and also our algorithms. First of all we need to acquire more data to have a more robust algorithms; with more data our model will be able to improve recursively. Other aspects of this projects include making the interface functional and easy to operate. In order for our model to learn more we need human input to correct on its mistakes. We also have ideas to improve the system by integrating more sensors such as a scale, and a hyperspectral camera to better recognize the materials that we are processing. Lastly, we envision a system that can perform in an industrial setting and can be deployed at the processing facilities to classify large streams of materials in a fast manner.

Team Member Contributions:

Elias Orellana - neural network model development
Kenny Kang - neural network model development
Nate Tran - neural network model development, AWS, Raspberry Pi
Umesh Thillaivasan - mentor contact, php / web interface, AWS, Raspberry Pi
Wesley Jin - data cleaning, graphs & visualizations, research, final report

Reference

1. https://archive.epa.gov/epawaste/nonhaz/municipal/web/html/
2. http://cs229.stanford.edu/proj2016/poster/ThungYang-ClassificationOfTrashForRecyclabilityStatus-poster.pdf
3. https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html
4. https://www.raspberrypi.org/blog/tag/raspberry-pi-user-guide/

Appendix

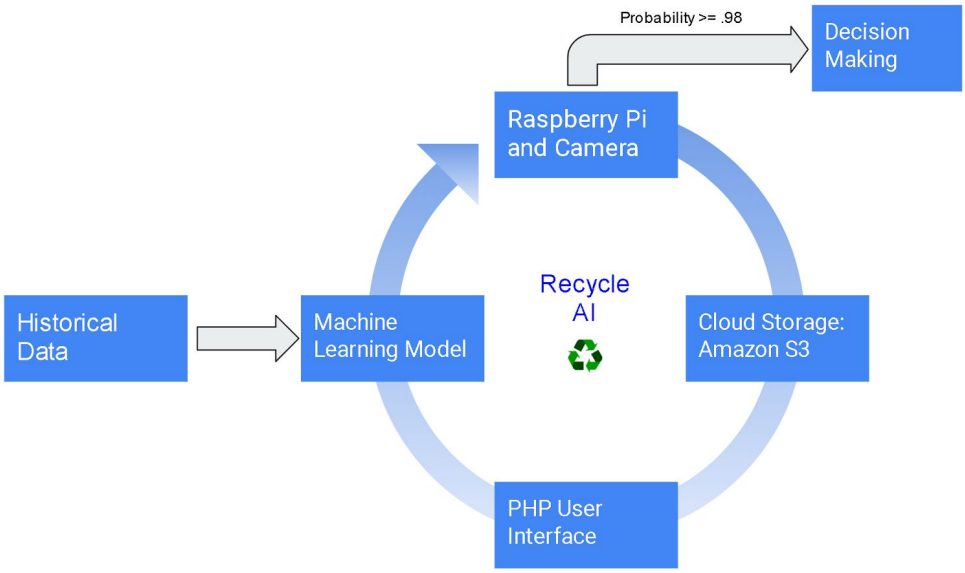

Figure 1: System Architecture - Our system follows a feedback loop that generate new data and update the model on its own

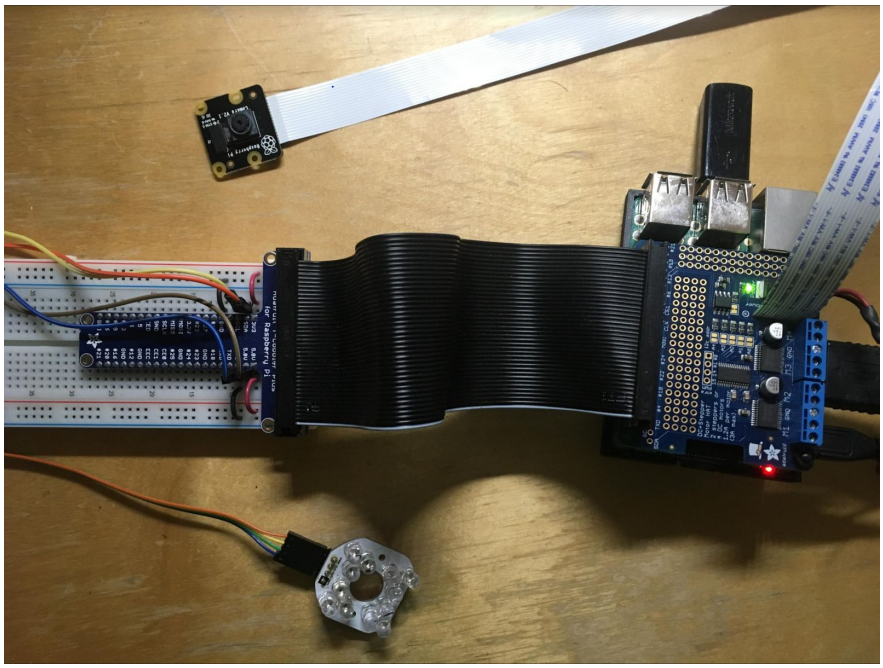

Figure 2: The vision system is consisted of a Raspberry Pi, the PCB, a Pi Camera, an LED/infrared lightsource, and a motor