Garbage Classification With Data Image

Final Report

Weiyang Chen
University at Buffalo
wchen38@buffalo.edu

Song Lin
University at Buffalo
songlin@buffalo.edu

Shuo Qiang
University at Buffalo
shuoqian@buffalo.edu

ABSTRACT

Our motivation is to help the individual identify the different types of garbage to help protect the environment. In this project, we are planning to use the "Garbage Classification Data-set" from Kaggle and apply our machine learning knowledge to create three different machine learning models which are 'K-Nearest Neighbors' (KNN), 'Support Vector Machine' (SVM) and 'Convolution Neural Network' (CNN) and compare their performance to choose the best technology to identify different classes of garbage which include: cardboard, glass, metal, paper, plastic and trash.

1 INTRODUCTION

Everyone throws out a lot of different kinds of garbage every day. Different types of garbage's involve different treatments. However, a significant amount of waste didn't end up where it supposed to. According to the EPA [1], only 12.5% of e-waste in the US is recycled. Garbage is often piled up or buried, causing negative environmental impact such that: the spread of odor, contaminating the soil, toxic materials seep into groundwater and produce damage to the atmosphere. To prevent the harm misclassification of garbage bring to our life. We decide to come up with a machine learning model that has a high image classification ability to help the individual identify the different types of garbage to help protect the environment. Our idea is create an application that user can upload the image of garbage, and the model identify the type of garbage through image processing.

To tackle this problem, we first collected the data-set from the kaggle which contains 2778 images with labels that belongs to 6 different classes. And we build three different classifiers : KNN, SVM and CNN to predict the labels that when we upload the new images. Then we do comparison among them by their Accuracy Performance, we find that CNN has best performance in this problem among these three technology. It make us to pay more time on CNN that build a better model and using the model to implement prediction method for user interface.

2 DATA SET

2.1 Source

https://www.kaggle.com/asdasdasasdas/garbage-classification

2.2 Summary

It is the Garbage Classification Dataset contains 6 classifications: cardboard (450), glass (548), metal (450), paper(623), plastic (564) and trash(145).

Above images are the 6 different types of garbage. And we will add more images by our own. The reason we choose this data set is the data set contains large enough data. And with large data set will avoid problem like underfitting and overfitting during the training. Also the image is clean that Machine can easy to find the features from the image, It will reduce the time to converge.

2.3 Data Re-Sampling

In the Data set, each image is RGB image, but their size are kind different, So we need to unified image size.(In here we using matplotlib to resize each image to (300,300) RGB image when we read the image) And also some machine learning technology only accept grey color channel or some other types data, for example, CNN only accept a flat vector as input to training the model. So we need to transform the data into different types to different classifiers.

- For the KNN classifier, We used OpenCV2 Library that change the RBG image into Gray image, It will help us easy to calculate the different between training samples and our input image that want to get identify.
- For SVM classifier, It only accept 2D data, So we used reshape method for each image into 1D vector, and a 2D vector to store all the images from the data.
- For CNN classifier, It only accept a flat vector, So we used flatten Method for each image before training.

2.4 Training Samples and Testing Samples

The total number of Samples in the data set is 2778, We randomly selected 2200 Samples as training Samples and rest 578 Samples as testing Samples.

Supervised by Sreyasee DAS BHATTACHARJEE.

Project in Introduce to Machine Learning (CSE574),
2020.

3 METHOD

3.1 Programming Language

Python [3] : Python is a general-purpose language which has rich resource libraries,frameworks, flexibility, platform independence, and a wide community. It is strong expansion capabilities. Python also offers a concise and readable code. It can do a set of complex machine learning tasks and enable user to build prototypes quickly that allow you to test your product for machine learning purposes.

3.2 Library

- **Tensorflow.Keras** : Open Source Library to help us to build "Convolution Neural Network".
- **Sklearn** : Open Source Library to help to build "Muti-Classes Support Vector Machine".
- **OpenCV2** : Open Source Library to read the image and do the image processing.
- **Numpy** : Using to do matrix calculate in the programming.
- **Matplotlib** Open Source data analysis Library, To help us to manage the data.
- **Glob** : Using to load the data file from the local memory.
- **Random** : Using to shuffle the data set, to Help us to separate the data into training set and testing set.

3.3 K-Nearest Neighbors (K-NN)

Definition : KNN[6] is a supervised Machine Learning Algorithm that can be used to solve both classification and regression problems. The main idea is to compare the sample we want to predict with the existing samples that each already contains label. To calculate the different between predict sample and existing samples, and find k minimum different existing samples to vote to classify predict sample. (For example, we want to predict a image belong to dog or cat, then k = 3, we get two images are dog and rest one is cat, then we decide that the predict sample is dog). As the example, we saw in the Figure 1, by the K-NN algorithm, the result prediction should be red.

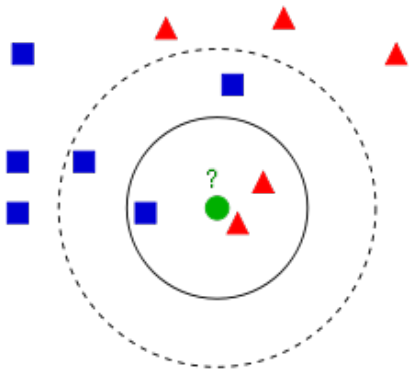

Figure 1: K-Nearest Neighbors [6]

Implement Detail :

Algorithm 1: K-Nearest Neighbors

Input: A set Training Data with Labels(T), A Predict Image (x), K value (k)
Output: Predict Label

for *Each training data t in T* **do**
  Calculate the different between x and t.
  Add different value into List (L).
Sort List(L) from small to big.
Find first (k) element in L as *new_list*.
Return the label with maximum appear times in *new_list*.

Calculate the different between two images : Each image in the data set is a 2D vectors with same size that contains same number of pixels. We can do calculate their absolute different by subtracting the corresponding pixels of the two images.

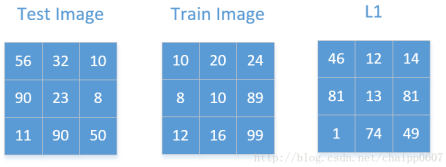

Figure 2: Absolute Different Between Two Images [4]

Accuracy Calculation : Input set of testing data, each testing data to make predict with whole set of training data. After finish all the prediction, calculate the accuracy with formula :

$$\frac{\#\ of\ Correct\ Predictions}{\#\ of\ Testing\ Data}$$

3.4 Support Vector Machine (SVM)

Definition : SVM[7] is also a supervised machine learning algorithm that used to analyze the data to make classification. In based SVM Model, It tries to maximize the margin between two categories by analyzing the training data, and after we input the predict sample, it make prediction by calculating the sample in the positive side or negative side.

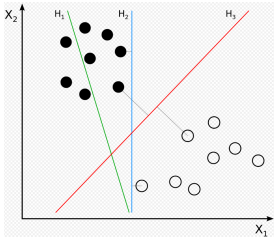

Figure 3: H3 separates them with the maximal margin [7]

Implement Detail: In this problem, the SVM maybe more complex, because we have 6 different classes. So, we implemented Multi-class SVM, which is using One vs. all classification. For each class, to do separation between current class and all other classes. After

we finish these steps for the six different classes, we will get six margins. And predict the upload image by comparing the location of upload image and margin direction.

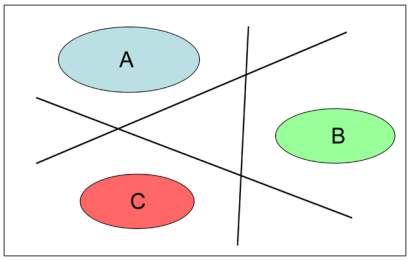

Figure 4: Diagram of 3 classes SVM and applied one vs. all classificaion[2]

Accuracy Calculation : After we got 6 margins by training the training data set. We need to input the a set of testing data, for each testing data to make predict with Multi-Class SVM model, and calculate the accuracy with formula :

$$\frac{\#\ of\ Correct\ Predictions}{\#\ of\ Testing\ Data}$$

3.5 Convolutional Neural Network (CNN)

Definition : A Convolutional Neural Network is a Deep Learning Algorithm. It can accept the images as input and to determine the different among the images. So people usually use CNN to solve image recognition. Our problem is to do the Garbage Classification, as we input a garbage image, we want the machine to determine which classes this garbage belongs to.

Implement Detail :

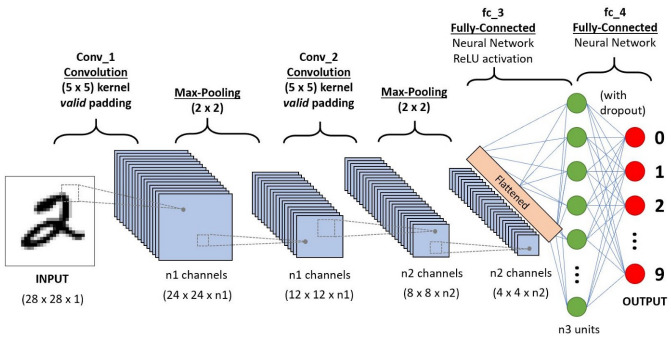

Figure 5: A CNN sequence to classify handwritten digits[5]

As the image above shows how the Convolution Neural Network working on a input image. It has three main layers which are convolution layer, pooling layer and fully-connection layer.

Convolution Layer : it use Kernel to implement convolution effect that features of images become obvious and if we scan whole image which after changed by convolution layer we can get features from the image as many as possible.

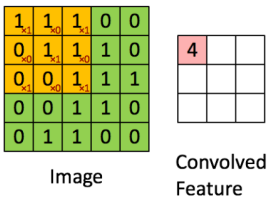

Figure 6: Example of Convolution working with Kernel.[5]

Pooling Layer, it use to reduce size of convolved Feature image by using pooling method ("Max-Pooling or Avg-Pooling") or principal component analysis (PCA) method. After reducing size, we will keep the most effective features and ineffective features will be removed during reducing size, and it will also help the model saving the time during the training.

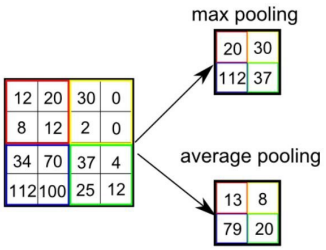

Figure 7: Example of two different pooling methods.[5]

So by keep convolution and pooling. Image features are highlighted, for now, we can call it feature map, and it helps Machine easier to learn the image and make classification.

Fully-Connected Layers : it is used to make classification, and it only accept a flat vector as input data, so after Convolution layers and pooling layers, we need to transform the feature map from matrix into a flat vector by Flatten method.

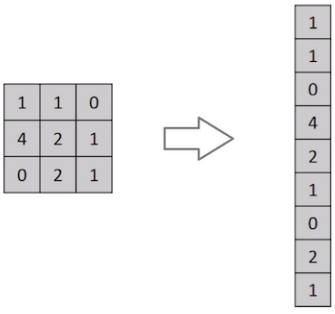

Figure 8: Flattening of a 3x3 image matrix into a 9x1 vector[5]

The reason it need to flatten the matrix is after the convolution and pooling layers, we get the feature map matrix, we need to integrate feature map as a 1D value and pass the value into activation function that machine can predict the label by activation function's result.

4 EXPERIMENTAL RESULT

4.1 K-Nearest Neighbors (K-NN)

K value effect for the classification accuracy:

K value selection is a hard problem for the K-NN, So first we want to see how K value effect the classification accuracy, and it will help us to pick a good K value. In the same training and testing data set, we test the accuracy of K-NN by changing the K value.

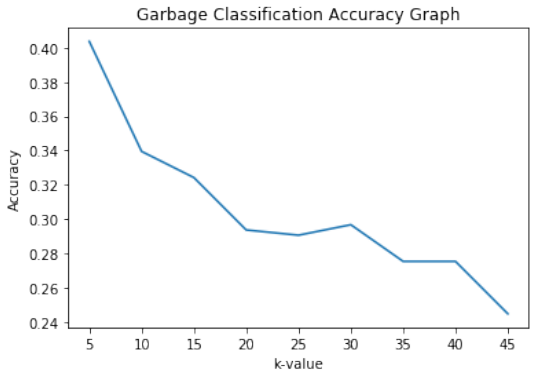

Figure 9: K-NN accuracy graph with K-value changing

In the Figure 9 shows when k-value goes up, the accuracy will go down.

K-NN Performance:

We shuffle the training data and testing data 10 times, and apply the different data into KNN with K value = 5 to check the accuracy performance in the KNN model.

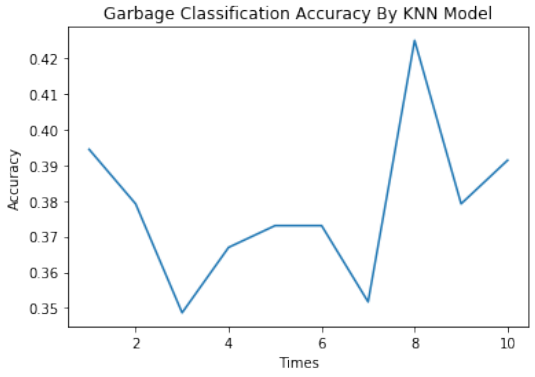

Figure 10: K-NN Accuracy with K value equal to 5 - 10 times

When the training and testing data are different, it's accuracy fluctuates between 0.35 and 0.40. So, K-NN accuracy is around 38% in this problem

4.2 Multi-Classes Support Vector Machine (SVM)

Kernel Trick :

We used image as input data that has higher-dimension. In the SVM, we need to apply kernel trick to de-dimension the data. In the experimental, We used 3 different kernels which are Linear, Polynomial and Gaussian, and compare their performance to determine the best kernel in this problem.

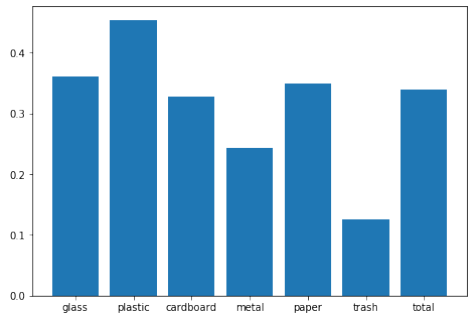

Figure 11: SVM Accuracy with Linear Kernel

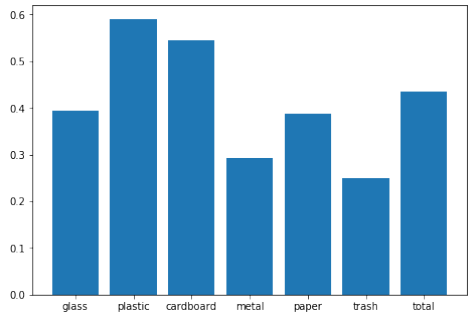

Figure 12: SVM Accuracy with Polynomial Kernel

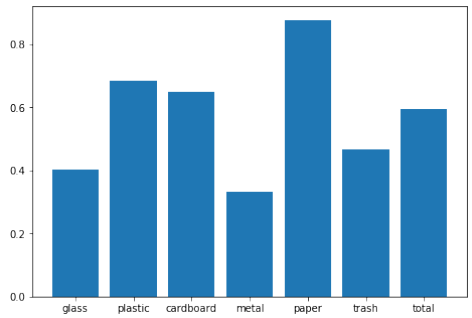

Figure 13: SVM Accuracy with Gaussian Kernel

It clearly to see by above three images that Gaussian Kernel has best performance among them in this problem. Under the same training and testing data, the model was built with Gaussian Kernel has Significantly higher accuracy than the other two models.

SVM Performance :

We apply Gaussian Kernel into SVM Model and trained model 10 times with different training set and testing set.

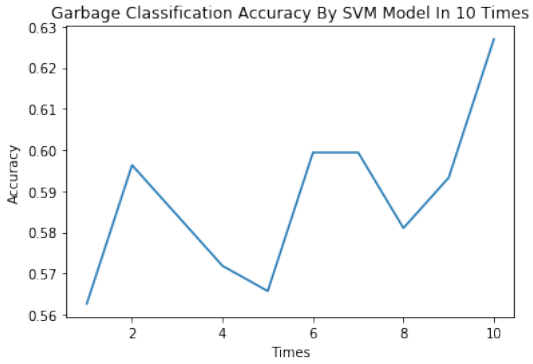

Figure 14: SVM Accuracy with Gaussian Kernel - 10 times

As Figure 14 shows, When the training and testing data are different, it's accuracy fluctuates between 0.56 and 0.63. So, SVM accuracy is around 60% in this problem.

4.3 Convolutional Neural Network (CNN)

First Model :

Building with 8 layers (Input data : 300 * 300 * 3 image) :

- 2D convolution layer - 64 filters (3,3)
- 2D AveragePooling - (2,2)
- 2D convolution layer - 32 filters (3,3)
- 2D AveragePooling - (2,2)
- 2D convolution layer - 32 filters (3,3)
- 2D AveragePooling - (2,2)
- Fully-Connected Layer with ReLU activation function
- Fully-Connected Layer with Softmax activation function

Runs 30 Epochs, each epoch take 100 steps.

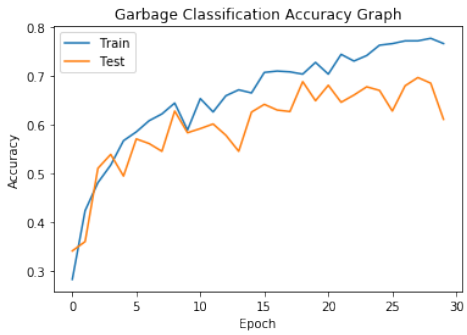

Figure 15: First Model Accuracy Graph

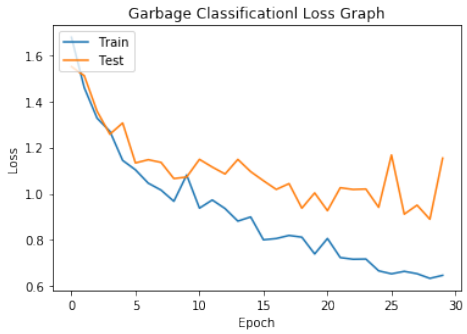

Figure 16: First Model Loss Graph

Second Model :

Building with 8 layers (Input data : 300 * 300 * 3 iamge)

- 2D convolution layer - 64 filters (3,3)
- 2D MaxPooling - (2,2)
- 2D convolution layer - 32 filters (3,3)
- 2D MaxPooling - (2,2)
- 2D convolution layer - 32 filters (3,3)
- 2D MaxPooling - (2,2)
- Fully-Connected Layer with ReLU activation function
- Fully-Connected Layer with Softmax activation function

Runs 30 Epochs, each epoch take 100 steps.

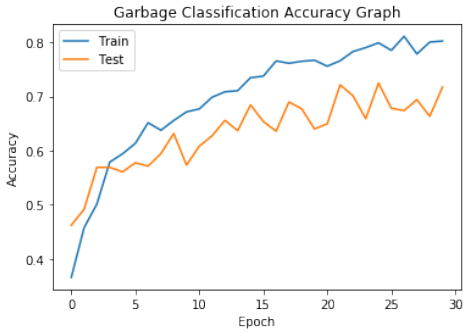

Figure 17: Second Model Accuracy Graph

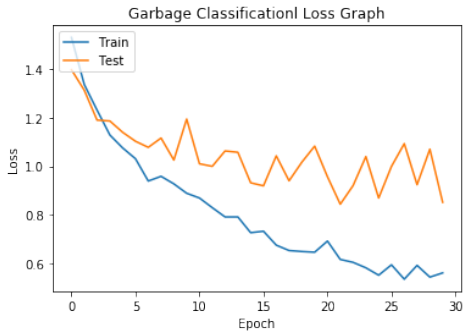

Figure 18: Second Model Loss Graph

Third Model :
Building with 8 layers (Input data : 300 * 300 * 3 iamge)

- 2D convolution layer - 64 filters (3,3)
- 2D MaxPooling - (2,2)
- 2D convolution layer - 32 filters (3,3)
- 2D MaxPooling - (2,2)
- 2D convolution layer - 32 filters (3,3)
- 2D GlobalMaxPooling
- Dropout - 25%
- Fully-Connected Layer with ReLU activation function
- Fully-Connected Layer with Softmax activation function

Runs 100 Epochs, each epoch take 100 steps.

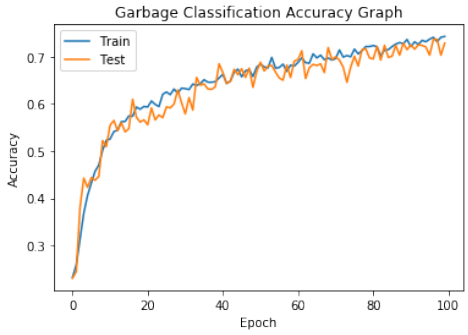

Figure 19: Third Model Accuracy Graph

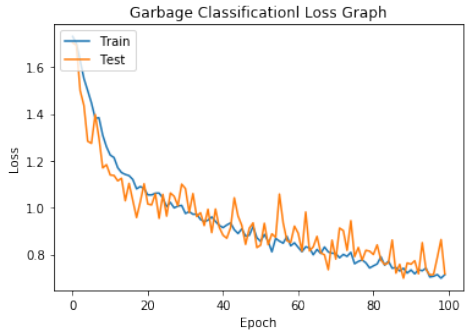

Figure 20: Third Model Loss Graph

Fourth Model :
Building with 8 layers (Input data : 300 * 300 * 3 iamge)

- 2D convolution layer - 64 filters (3,3)
- 2D MaxPooling - (2,2)
- 2D convolution layer - 32 filters (3,3)
- 2D MaxPooling - (2,2)
- 2D convolution layer - 32 filters (3,3)
- 2D MaxPooling - (2,2)
- Fully-Connected Layer with ReLU activation function
- Dropout - 25%
- Fully-Connected Layer with Softmax activation function

Runs 200 Epochs, each epoch take 71 steps.

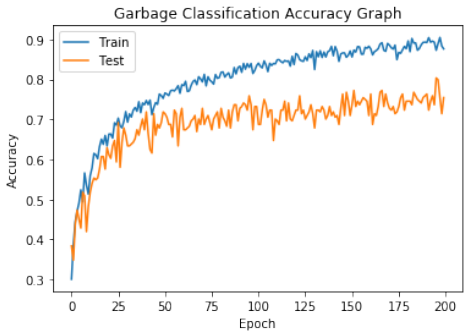

Figure 21: Fourth Model Accuracy Graph

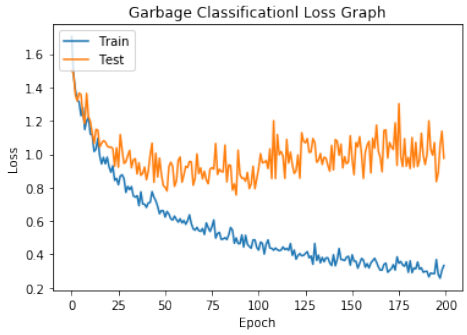

Figure 22: Fourth Model Loss Graph

CNN Performance :
As the first two model images shows that the two models have not overfitting yet, and precision accuracy has the potential to rise. By comparing the two models, We can also see that MaxPooling has better performance than AvgPooling in this problem. As last model image we can see that although the Training Accuracy is increasing, Validation Accuracy stay the same after 100 epochs, it is same to the loss value. So the CNN Accuracy is around 75%. It is better than KNN and SVM technology.

5 USER INTERFACE IMPLEMENT

5.1 Technology Select

Model : CNN (Has highest accuracy among three technology)
Platform : Python with Tkinter library

5.2 UI Layout

Figure 23: UI Layout

In the middle of the UI, there shows a image. If we click the button in the bottom named 'Open', Then it will ask us to select a image file from your computer, and middle image will change to the image you selected, and it will also send the image into CNN model to let model predict the classification and return the result as text below the image.

5.3 How to use

First : Click the 'Open' button and it will ask you to select a image from your computer.

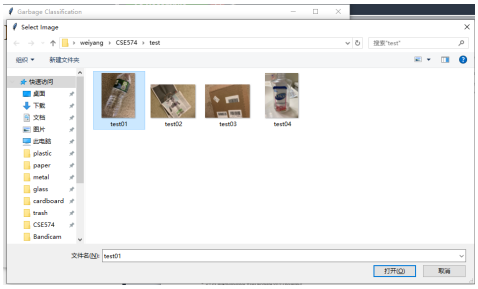

Figure 24: Select Image

Second : After Select the image the result will show in the main screen as text information.

Figure 25: Showing the result

6 CHALLENGE

6.1 Training Time

During the CNN training, each epoch usually take 300 sec to finish, and in one training period, we trained at least 100 epoch which take at least 8 hours for one model.

6.2 Hardware Facilities

Training CNN require the computer's GPU, If your computer GPU is not powerful enough, you may not be able to use your computer to train CNN model. I advise to use cloud computers from big companies like google colab or AWS or UB CCR. But when if training time too long, You are at risk of being interrupted.

6.3 Lack of data

2778 Images are not really enough for the CNN training. It limits the increase in accuracy. And also, the data is too single that all the image's background is white. It will cause if the background is no longer white when we test, the accuracy will no longer accurate.

7 CONCLUSION

We implement the three different image classification method.

- KNN implemented with around 40% Accuracy
- SVM implemented with around 60% Accuracy
- CNN implemented with around 75% Accuracy

In Conclusion, CNN has the best performance among three of them, and has potential to improve its accuracy.

8 FUTURE WORK

For the future work, we will collect more garbage image data, and focus on the CNN improvement. I believe if we get large enough data set, our model Accuracy will be much better even with different background colors or different places.

REFERENCES

[1] U.S. Environmental Protection Agency. [n. d.]. Municipal Solid Waste. https://archive.epa.gov/epawaste/nonhaz/municipal/web/html/
[2] Ben Aisen. 2006. A Comparison of Multiclass SVM Methods. https://courses.media.mit.edu/2006fall/mas622j/Projects/aisen-project/
[3] Angela Beklemysheva. [n. d.]. Why Use Python for AI and Machine Learning. https://steelkiwi.com/blog/python-for-ai-and-machine-learning/
[4] chaibubble. [n. d.]. Learning KNN (1) Image classification and KNN principle. https://blog.csdn.net/chaipp0607/article/details/77915298
[5] Sumit Saha. [n. d.]. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.
[6] Wikipedia. [n. d.]. k-nearest neighbors algorithm — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
[7] Wikipedia. [n. d.]. Support-vector machine — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Support-vector_machine