Trash Classification

Mohammed Al Saleem 188301
King Abdullah University of Science and Technology, CEMSE

May 13, 2023

Abstract

This report investigates three methods for image classification: Convolutional Neural Networks (CNNs), k-Nearest Neighbor classifier (KNN), and CLIP. The experiments were conducted on a small dataset of images of six classes: paper, cardboard, glass, metal, plastic, and trash.
Data augmentation techniques, such as MixUp, are used in image classification to increase the size and diversity of a training dataset. This can improve the performance and generalization of machine learning models, especially when the amount of available training data is limited.
Hopefully, this project will help to reduce pollution and improve the current recycling process. By using the methods in this project to classify garbage into recycling categories, the waste process can be more efficient, resulting in positive environmental and economic effects.

1 Introduction

The issue of trash has been increasing in the last years, leading to pollution (land, water, and air), as well as the consumption of toxic waste by animals, and increasing the number of landfills.
To ensure sustainability, recycling has become essential, and the current recycling process needs improvement. By using images classification methods to classify garbage into recycling categories, the waste process can be more efficient. Which will lead to a reduction in waste, and resulting in positive environmental and economic effects.
The classification problem considering receiving an image of a trash piece with white background and classifying it into one of recycling material categories. We use CNN, KNN, and CLIP to classify and predict the image into one of the six categories of trash classes. Also, we will use Mixup which is a data augmentation method to get more images.

2 Related Work

In the field of image classification, there have been numerous research projects that have utilized support vector machines and neural networks. One notable CNN architecture for image classification is AlexNet, which won the 2012 ImageNet Large Scale Visual Recognition Challenge. AlexNet was a significant breakthrough in the field, as it demonstrated the efficacy of deep learning in image classification tasks. Since then, CNNs have become an increasingly popular approach in image classification.
KNN clustering is another algorithm that can be used for image classification. In this approach, each image is represented as a vector of features, and the algorithm groups similar images together based on their distance in feature space. To classify a new image, KNN calculates the distance between the image's feature vector and the feature vectors of the labeled images in the training set.
CLIP (Contrastive Language-Image Pre-Training) is a recent breakthrough in image classification. It is a multimodal learning model that can classify images based on text inputs, such as textual descriptions or labels. During training, the model learns to associate words with the visual features that are commonly associated with them. This allows the model to classify images based on text inputs by comparing the visual features of an image with the semantic meaning of the text input.
MixUp data augmentation is a recent development in image classification that has shown promising results. This method generates virtual training examples by linearly interpolating pairs of images and their labels. This technique encourages the classifier to learn more robust and generalizable decision boundaries.

3 Data Description

The dataset contains images of recycled objects across six classes:'

- 501 glass (426 train / 75 test)
- 594 paper (504 train / 90 test)
- 403 cardboard (343 train / 60 test)
- 482 plastic (410 train / 72 test)
- 410 metal (348 train / 62 test)
- 137 trash (other) (110 train / 27 test)

The total: 2527 images

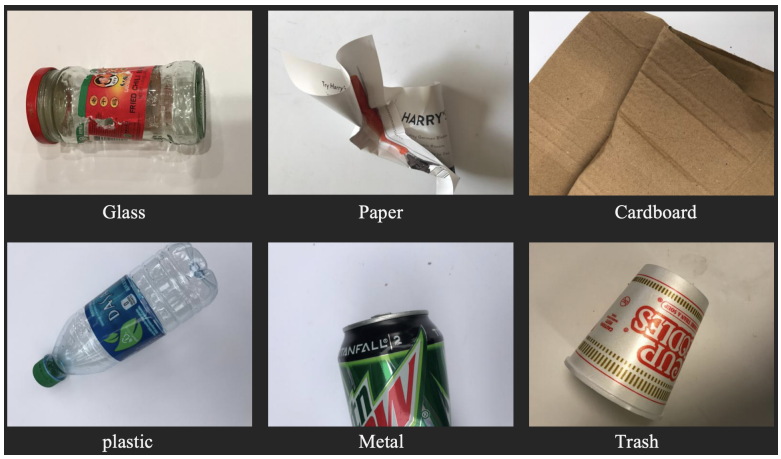

Figure 1: Some images from the data.

- The dataset containing pictures have been resized down to 512 x 384.
- The images taking using a white posterboard as a background.
- The lighting and pose for each photo are different, which introduces variation in the dataset.

4 Methods

4.1 MixUp

Data augmentation techniques are used in image classification to increase the size and diversity of a training dataset by applying various transformations and modifications to the original images. This can improve the performance and generalization of machine learning models, especially when the amount of available training data is limited (like in our case, the size of each class is relatively small). Some common data augmentation techniques include flipping, rotation, scaling, cropping, and adding noise.
Mixup is another data augmentation technique used in image classification that combines pairs of images and their labels to create new training examples. The technique involves taking a weighted average of two images (xi, xj) and their corresponding labels (yi, yj) to create a new image and label. The weights are drawn from a Beta distribution, which helps to smooth the distribution of training examples and improve the generalization of the model.
But in the project, MixUp is used in a more straightforward way. Where it is applied to each class separately to increase the number of images of each class, and the uniform distribution is used determine the weighted average of two images. Therefore, the MixUp applied to images and it take the class label directly.

$$\hat{x} = \lambda x_i + (1 - \lambda) x_j$$
$$\hat{y} = \lambda y_i + (1 - \lambda) y_j$$

Figure 2: MixUp Equation.

4.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a popular type of deep learning architecture used in image classification tasks. They consist of multiple layers that perform different operations on input images including:

1. Convolutional layers: These layers extract features by applying a set of learnable filters to the image. Each filter detects a certain feature in the image, like edges or corners, and the output of the layer is a set of feature maps that represent the presence of these features in the image parts. The operation done by sliding a filter over the image and computing the dot product between the filter and the overlapping pixels. 3x3 kernel, stride 1 and with padding.

2. Activation layers: These layers introduce non-linearity into the network by applying a non-linear activation function. In this project we will use ReLU to speed up training.

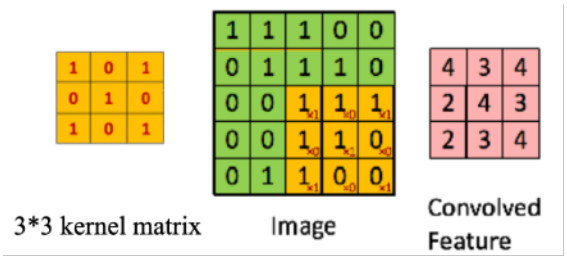

Figure 3: The kernel of convolutional layer.

$$F(x) = max(0, x) \tag{1}$$

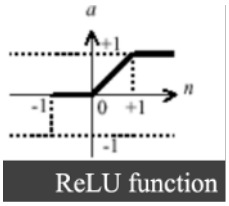

Figure 4: ReLU function.

3. Pooling layers: These layers downsamples the output of the convolutional layers by taking the maximum or average value over small regions of the feature maps. Which reduce the dimensionality of the feature maps and increase the efficiency of the network. In this project we can use the module max_pooling2d with a size of 2×2 and stride of 2 and no padding.

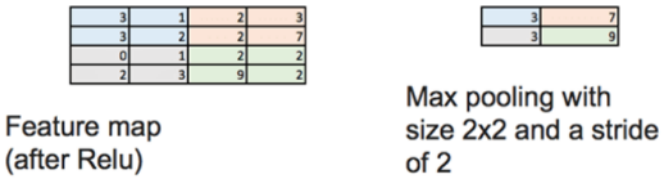

Figure 5: Max pooling.

4. Fully connected layers: These layers perform classification by taking the output of the previous layer and mapping it to a set of output classes. Each neuron in the layer is connected to all the neurons in the previous layer. The final fully connected layer uses a SoftMax activation function, which outputs a probability from 0 to 1 for each predictable classification label in the model. The

dropout regularization is added to the hidden layer nodes to reduce over-fitting. During training, the weights of the filters in the convolutional layers and the

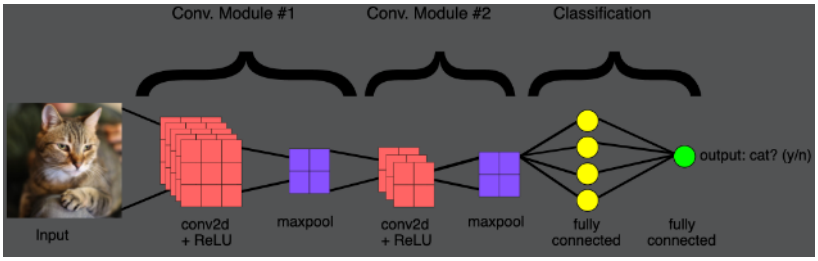

Figure 6: CNN layers.

fully connected layers are learned to minimize the loss function. The class with the highest probability is then selected as the predicted label for the image.

4.3 k-Nearest Neighbor classifier (KNN)

KNN (k-Nearest Neighbors) is a non-parametric clustering algorithm that relies on the distance between feature vectors (in our case, are the color histograms the raw RGB pixel of the images).
KNN depends on the neighbor observations with independent variables to predict. The number of neighbors which will be calculated represented by 'k', calculating the distance between neighbors can be done in many ways, the famous one is Minkowski distance. When the power parameter (p) equal to 2 it called "Euclidean Distance", and when it equal to 1 it called "Manhattan Distance". The distance has an inverse relationship with the weight of neighbors.

$$\text{Minkowski} \quad \left(\sum_{i=1}^{k} (|x_i - y_i|)^q \right)^{1/q}$$

Figure 7: Minkowski distance formula.

Then the k closest images in the training set are then used to determine the class of the new image.

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} w_i f(x_i)}{\sum_{i=1}^{k} w_i}$$
$$\text{where} \quad w_i \equiv \frac{1}{d(x_q, x_i)^2}$$

x: features, f: class, w: weight, d: distance

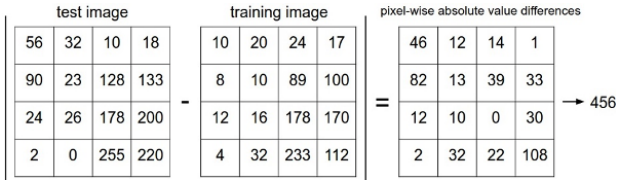

Figure 8: calculating distance example.

The used parameter that used in this project are n_neighbors=i, p = 1, algorithm = 'brute'
The brute algorithm is the brute-force algorithm to compute nearest neighbors by searching through all possible pairwise distances. Also, p = 1 means that the used distance is Manhattan Distance.

4.4 CLIP

CLIP (Contrastive Language-Image Pre-training) is a method for image classification that was created by researchers at OpenAI, and was first introduced in January 2021.
CLIP uses a combination of natural language processing and computer vision techniques. It involves pre-training a neural network on a large dataset of images and associated natural language captions, and then fine-tuning the network on a smaller dataset of labeled images to perform a specific task. During pre-training, the network learns to associate images with their corresponding captions by maximizing the similarity between them in a contrastive learning framework. CLIP pre-trains an image encoder and a text encoder to predict which images were paired with which texts in OpenAI dataset. During fine-tuning, the network is trained to classify images based on their labels (zero-shot classifier). The network is initialized with the weights learned during pre-training, and the weights are fine-tuned to minimize the loss.

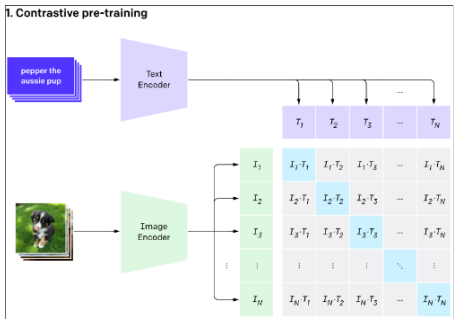

Figure 9: Contrastive pre-training.

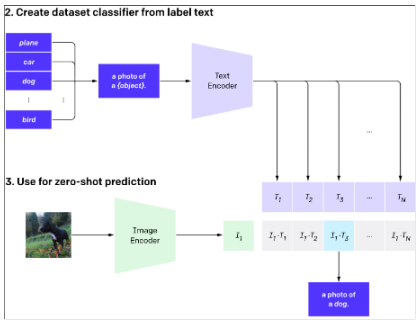

Figure 10: Create dataset classifier and use for zero-shot prediction.

5 Experiments / Results

5.1 MixUp

Here we wanted to increase the number of images in each class to 150%. To do that a loop is generated with the half length of each class. Then randomly choose two images within the class, and generate a random mixing weight following Unif(0.2, 0.8).
Finally, mix the images using the given weight by the equation in Figure 2.
So, the new size of the dataset is 3789. 891 paper, 604 cardboard, 751 glass, 615 metal, 723 plastic, 205 trash.

5.2 CNN

After preprocessing data and splitting them to train and test and convert them to tensors on DataLoader with batch size equal to 16, the CNN model is defined with the following layers (and Dropout = 0.2):

- Layer 0: Input image of size 128x128
- Layer 1: Convolution with 128 filters, size 3, stride 2, padding 1
- Layer 2: Max-Pooling with a size 3x3 filter, stride 2
- Layer 3: Convolution with 256 filters, size 3, stride 1, padding 1
- Layer 4: Max-Pooling with a size 3x3 filter, stride 2
- Layer 5: Convolution with 256 filters, size 3, stride 1, padding 1
- Layer 6: Convolution with 128 filters, size 3, stride 1, padding 1
- Layer 7: Max-Pooling with a size 3x3 filter, stride 2
- Layer 8: Adaptive-Pooling with a size 64x64 filter

- Layer 9: Fully Connected with 1024 neurons
- Layer 10: Fully Connected with 512 neurons
- Layer 11: Fully Connected with 6 neurons

The used loss function is the categorical cross-entropy, and the used optimizer is SGD (lr=0.001, momentum=0.5)

The results:

- Test Loss: 0.743851
- Test Accuracy of paper: 86% (115/133)
- Test Accuracy of cardboard: 64% (58/90)
- Test Accuracy of glass: 83% (89/107)
- Test Accuracy of metal: 76% (70/92)
- Test Accuracy of plastic: 71% (77/108)
- Test Accuracy of trash: 53% (16/30)
- **Test Accuracy (Overall): 75% (425/560)**

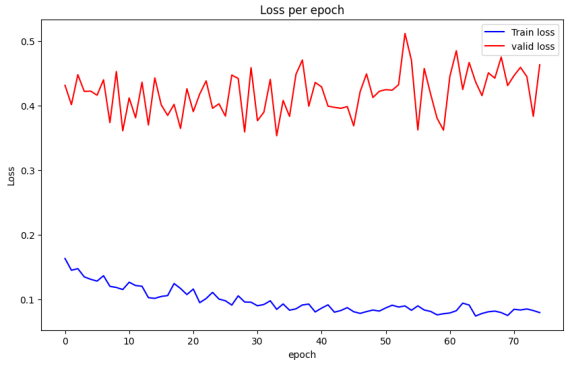

Figure 11: Loss per epoch

5.3 KNN

Similarly, after preprocessing data and splitting them to train and test, the model runs on different values of k, and we get the following accuracies:

In Figure 12, we can see that k=1, 2 returns the same and the best accuracy 78.816%. Depending on k = 2, we get the following precision results: cardboard 0.80%, glass 0.79%, metal 0.78%, paper 0.92%, plastic 0.69%, trash 0.68%,

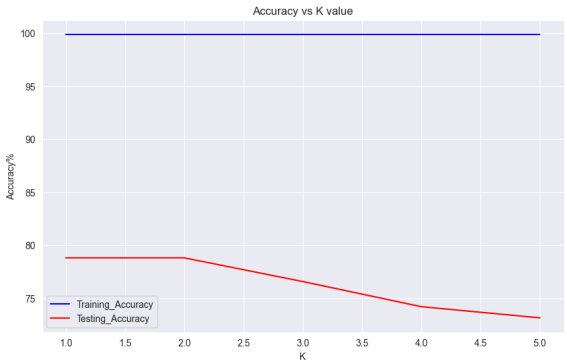

Figure 12: Accuracy vs. k

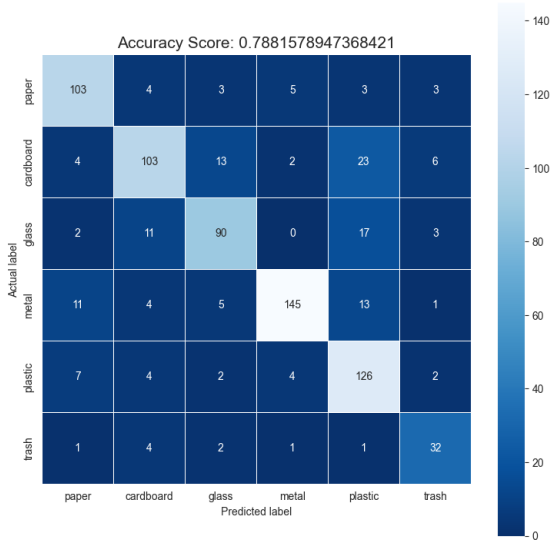

Figure 13: confusion matrix

5.4 CLIP

The used model here is the 'ViT-B/32', here noticed that the trash class (others) cause lower accuracy (about 60%), so the model was applied to the other main five classes and returned the following results:

- paper: 83.73%
- cardboard: 90.07%
- glass: 58.72%
- metal: 33.01%
- plastic: 91.01%
- **total accuracy: 71.31%**

6 Conclusion

In this report, three methods for image classification were explored: Convolutional Neural Networks (CNNs), k-Nearest Neighbor classifier (KNN), and CLIP. This is after using the MixUp data augmentation technique to create new image examples.
The experiments conducted in this report showed that the KNN model achieved the highest accuracy of 78.82% on the test set for k=1 and k=2, followed by the CNN model with an accuracy of 75% on the test set. CLIP achieved an accuracy of 71.31% on the test set for the main five classes. However, the accuracy for the trash class was relatively low for all models.
Overall, data augmentation techniques and deep learning architectures like CNNs can significantly improve the performance and generalization of image classification models. KNN is a simple and effective algorithm for image classification, especially when the size of the dataset is small. CLIP is a promising method that combines natural language processing and computer vision techniques, and it could provide better results with good finetuning.
In conclusion, the CNN model achieved the best results for image classification in this project, and further improvements can be made by exploring other data augmentation techniques and deep learning architectures.

7 References

https://medium.com/analytics-vidhya/image-classification-techniques-83fd87011cac

https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks

https://www.guru99.com/convnet-tensorflow-image-classification.html

https://paperswithcode.com/method/mixup

https://openai.com/research/clip

https://cs231n.github.io/classification/

https://github.com/garythung/trashnet