Işık Özsoy 21703160
Utku Kalkanlı 21602325
Yahya Mahmoud Ahmed Elnouby Mohamed 21801332
Mohammed Sameer Yaseen 21801331
Oğuz Orhun Tüzgen 21802313
Section: 2
TA: İlayda Beyreli

CS 464 - INTRODUCTION TO MACHINE LEARNING: Progress Report
Computer Vision-Based Garbage Classification

1. Introduction and Background Information

Currently, garbage is classified into recycling categories manually. However, performing that process manually is time and energy-consuming as well as economically inefficient. Given that the process is a repetitive one and is based on object detection and identification, we thought it would be more time and economy-efficient if that process was done automatically. The aim of the project is to find an efficient way to automate classifying trash. Therefore, a computer vision approach will be used to provide predictions regarding objects' materials based on the images in our training datasets and their labels. We are going to predict the category of an object and classify it as a battery, organic material, glass (brown, green, or white), plastic, metal, cardboard, paper, or trash using Garbage Classification dataset [1,2].

Garbage Classification dataset [2] is an image dataset for classifying household garbage. This dataset was chosen since it sorted the garbage into 12 classes while the other dataset [1] consists of approximately 6 classes. The dataset with 12 classes consists of 15,150 images from 12 different classes of household garbage; paper, cardboard, biological, metal, plastic, green-glass, brown-glass, white-glass, clothes, shoes, batteries, and trash as mentioned while the second one has approximately 2500 images. The datasets have good variety, large quantities, and notes which makes the implementation of the model easier. More data leads to more accurate results, therefore it is crucial to have a large amount of data which is considered while choosing the dataset. Considering this, if the current data is not enough, we are going to combine Garbage Classification dataset with other datasets.

BATTERY


BIOLOGICAL


SHOES

Figure 1: The representation of the Garbage Classification dataset

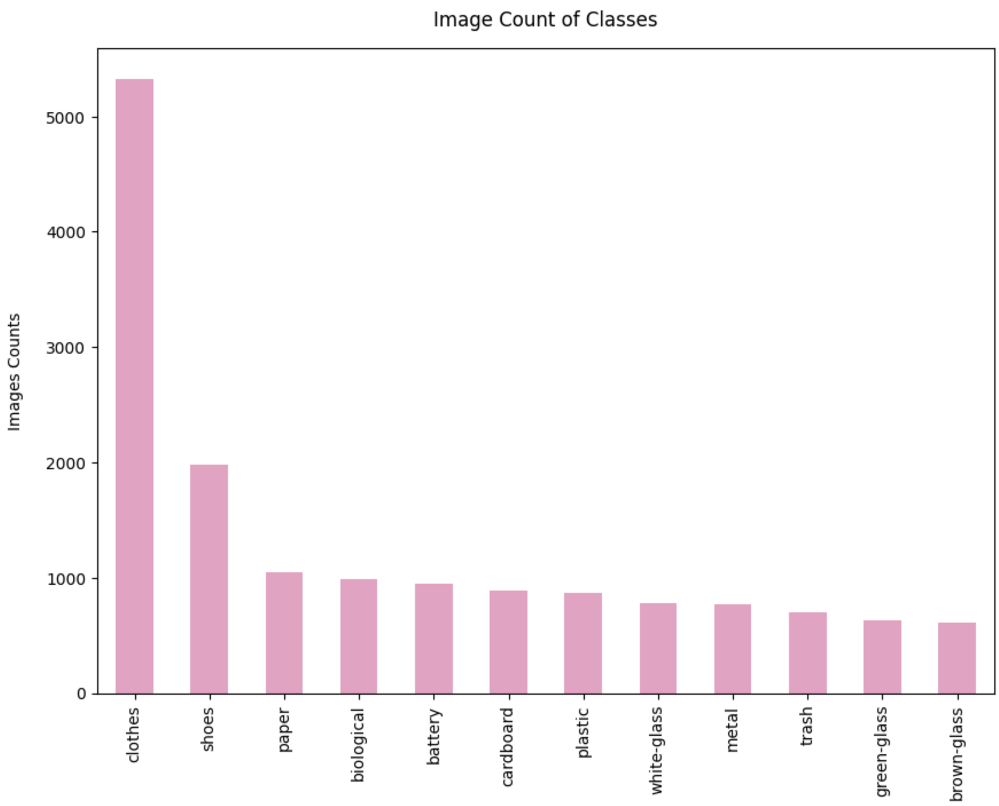

Figure 2: Bar chart that demonstrates image count of classes for Garbage Classification dataset [2]

The project is based on garbage classification by using transfer learning. Transfer learning is a technique whereby a neural network model focuses on using the knowledge learnt from the previous problems and applying it to other similar problems. In our project, the knowledge gained while learning to detect plastics could apply when trying to recognize papers. Another benefit of transfer learning is the decrease in the training time for a convolutional neural network model.

Convolutional Neural Network (CNN) is a specialized state of multilayer network for detecting the geometric shape in image processing [4]. The role of the CNN is to reduce the images into a form which is easier to process, while maintaining all the features which are critical for getting a good prediction [5]. A CNN extracts features from images instead of manually doing it. This increases accuracy of learning models in computer vision. We are going to benefit from this feature in our project.

If you flatten an image and provide each image pixel as an independent neighborhood, you lose so much important information such as edges or shape of an object. In a block image where we preserve spatial structure, neighbors of pixels maintain that important information. Because of that, we choose to implement our learning algorithm with CNN, which exploits the locality of features to guide the learning of features. In CNN, by cutting the image to small patches rather than processing the full image at a time, we decrease the complexity and the weights to learn. This is done by convolving our input image and a filter. We will use the power of local receptive fields of an image where neighbor pixels are activated and processed together.

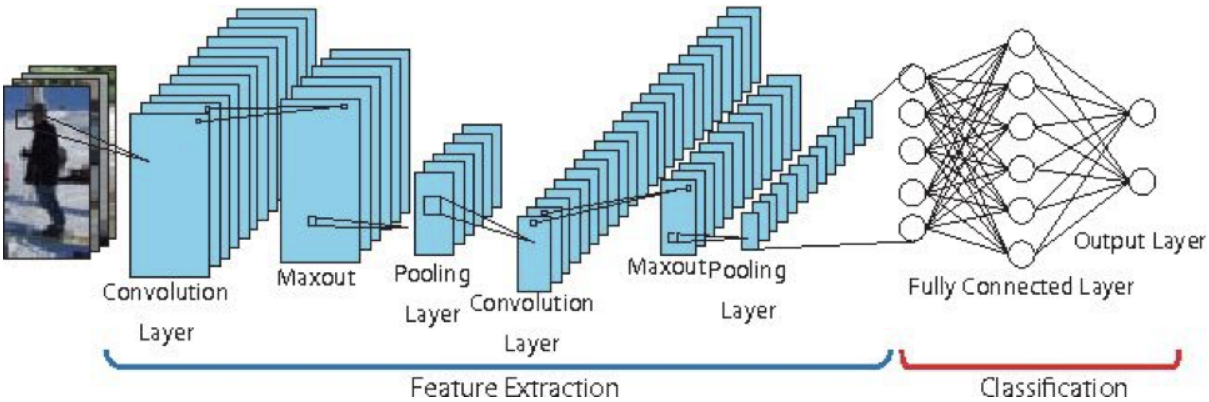

Figure 3. Architecture of Convolutional Neural Network

General architecture of our CNN shows how the action flow will be, Convolution layers will be followed by maxout and pooling layers, where we will use Max Pooling. Max Pooling chooses the maximum value from the output of feature maps(output of convolution). By choosing the maximum one, we basically summarize the convolution operation in each layer and downsampling our volume.

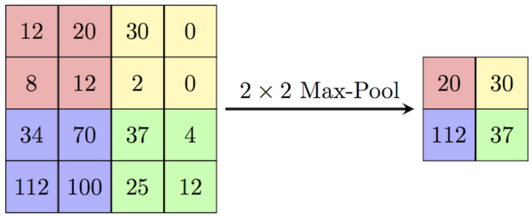

Figure 4. Max Pooling

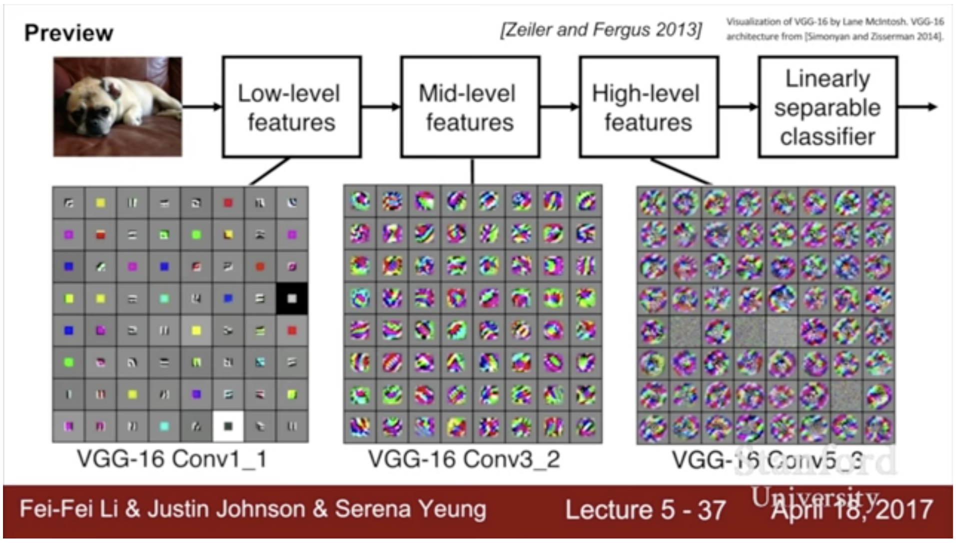

Figure 5. Visualization of Filters

Our CNNs will learn the most optimal filters for recognizing specific objects and patterns in our garbage images. In fact, it will learn in each layer by multiple filters. The filters in early layers represent low level features like edges, at the mid level more complex features like corners and at high level things will resemble concepts and blobs. In figure 5, each grid represents a neuron and what the input would look like that maximizes the activation of that neuron.

At the end, when we get our last convolutional layer output, we will use a fully connected layer to combine all of these convolutional outputs and use Softmax Function to convert it to a probability distribution. We will use the ReLU function(Rectified Linear Unit) as our activation function.

2. What We Have Done

Obtaining Data

We have searched for a dataset featuring a large number of images across multiple categories for garbage. We've structured these data into two main folders, namely train and test. The test folder contains approximately 30% of the total images of each category. Additionally each folder contained a number of subfolders equal to the number of classes we have. The reason for making such a structure is to utilize a PyTorch built in function called 'ImageFolder' which is able to read all the data only by providing a path to the parent folder.

Image processing

Since each image has a different size, we have resized all images to 256X256, and transformed them into tensors. We then have splitted the data into validation and training sets. The split was performed randomly, with 200 items saved for validation and the rest for training. For now, we have only used the dataset with 6 classes [1] and we calculated the accuracy by using train, validation and test split method.

GPU Utilization

To be able to speed up the training process, we needed to use the GPU. For that end, we used PyTorch built-in function 'DataLoader' with a batch size of 32. A batch is a portion of the data that will be sent to the GPU and used for training at each time. The reason for using batches is that we don't want to load all the data to the GPU at once since its memory might not be able to handle the load.

Transfer Learning

Transfer learning is a machine learning technique which utilizes previously trained models- for different purposes- in order to make better predictions for a case specific problem. Since the very fundamentals of computer vision problems are quite similar, we can use a state-of-the-art computer vision model such as ResNet and increase its depth to adapt to our use case specifics. In this case we used ResNet50 as the base model, which contains 50 layers of neural network. The reason we use ResNet rather than alexNet or vgg16 etc. is because it requires less computational power while providing one of the best accuracies(>%75). We have then added the last layer to make the predictions for our specific case of garbage detection.

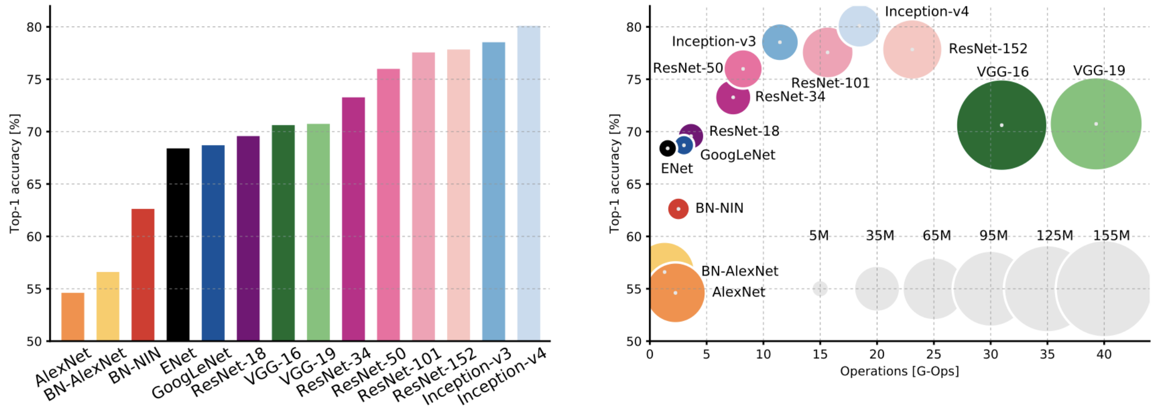

Figure 6: Comparison of Different CNN Architectures [6]

Training

We used the following hyperparameters for training the model: 16 epochs, Adam optimization function, and 5.5e-5 learning rate. Using these hyperparameters, we could achieve 91.41% accuracy on the validation set. Additionally, looking at figure 6, which illustrates the increase in the accuracy per epoch, we notice that the accuracy increases up to around 94.5% but it keeps fluctuating between 90% and 95%. This means that based on our model and data this is the best we can do, and the accuracy will hardly increase with increasing number of epochs.

Overfitting

To get a better understanding of the accuracy of our model, we have plotted the loss values per an epoch for each of the test and validation data. We would expect the curves for both of the test and validation losses to be decreasing and close to each other. If the training curve decreased initially, and started increasing while the validation curve kept decreasing, that is an indication that some overfitting has happened. Based on figure 7, we notice that both of the curves are decreasing and close to each other; hence, overfitting is minimal in our model.

Results

By testing the model on training data we have achieved an accuracy of 84.93%, which is actually pretty good given our limited dataset. In order to increase the accuracy, we are planning to combine datasets and use k-Fold Cross Validation which will be explained detailly in the “What Remains To Be Done” section.

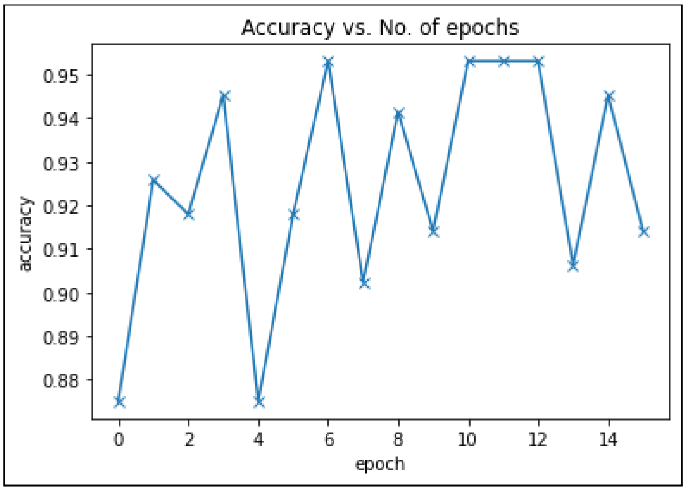

Figure 7: Accuracy per epoch

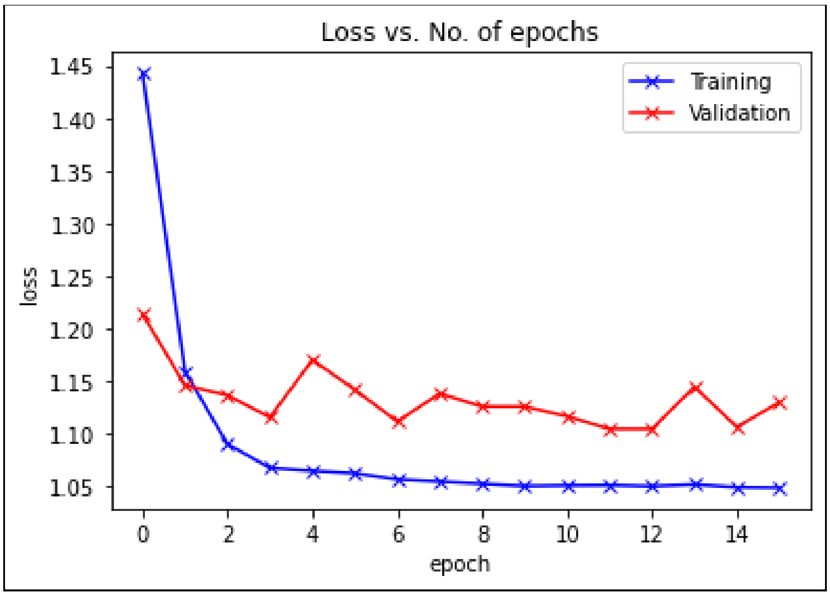

Figure 8: loss value per epoch using Training and validation data

Implementation of Libraries

In this project, we are planning to use the PyTorch library, in addition to Pandas, NumPy and sklearn.model_selection libraries. We will use Pandas and NumPy for low level matrix operations. We are planning to use the ResNet50 model with transfer learning. While implementing the function that splits Garbage Classification dataset into three folds, we have used pandas, numpy and sklearn libraries.

For implementing the transfer learning technique which utilizes previously trained models, currently we are using PyTorch.torchvision.models. However, we might later on use Keras python library. The advantage of using Keras is that it provides access to many developed top performing pretrained models with different architectures for image recognition.

3. What Remain To Be Done

Data Augmentation

Data augmentation is a powerful technique for utilizing smaller datasets. This technique generates randomly transformed images from the original images from the dataset [7]. Therefore we can generate some more variety in the existing dataset. By utilizing this technique we want to achieve a better accuracy score with our modest-sized dataset.

K-Fold Cross Validation

As mentioned in section 2, we have used the smaller dataset that consists of approximately 2500 images and train, validation, split method for model selection. However, we want to use k_Fold Cross Validation as a model selection method. We thought that it might be more appropriate since we have a limited amount of data, rather than splitting our dataset as a train-validation-test to use our dataset more efficiently. In cross validation, we will split our dataset into 3 folds, and we plan to perform 3 runs on our dataset, each run corresponds to training model on (k-1) splits and testing on 1 split in rotation. We plan to determine performance metrics (MSE, misclassification error rate, etc.) based on the results across the iterations. We have already written the function that splits the data into three folds, however, we have not used it in our experiments yet.

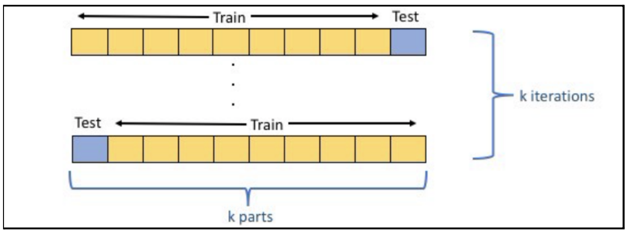

Figure 9: The division of train and test sets using K-fold cross validation technique

4. Description of the Division of Work

Team Members	Tasks Done	Remaining Tasks
Işık Özsoy	Investigation of possible issues of Garbage Classification dataset	Implementation of k-Fold Cross Validation Combining datasets
Utku Kalkanlı	Research on transfer learning and how CNN is used on classifying images. Research on CNN architectures.	Combining ResNet transfer learning and k-Fold Cross Validation.
Yahya Mohamed	CNN research to understand how it will be used, implementing some models using the CNN approach to classify the garbage	Research on Ensemble Learning Applying data augmentation

Mohammed Yaseen	Research on state of the art garbage classification solutions	Examining existing projects for the implementation stage
Oğuz Tüzgen	Investigation of new datasets to increase the scope of the model Python library research	Optimization for ML techniques to increase the accuracy and the efficiency

Table 1: Distribution of work

5. References

[1] "Garbage classification," Kaggle, 24-Nov-2018. [Online]. Available: https://www.kaggle.com/asdasdasasdas/garbage-classification. [Accessed: 27-Oct-2021].

[2] M. Mohamed, "Garbage classification (12 classes)," Kaggle, 24-Jan-2021. [Online]. Available: https://www.kaggle.com/mostafaabla/garbage-classification. [Accessed: 27-Oct-2021].

[3] Sangminwoo, "RecycleNet/Data/dataset-resized at master · sangminwoo/recyclenet," *GitHub*. [Online]. Available: https://github.com/sangminwoo/RecycleNet/tree/master/data/dataset-resized. [Accessed: 27-Nov-2021].

[4] U. Ozkaya and L. Seyfi, "Fine-tuning model comparisons on Garbage Classification for Recyclability," *NASA/ADS*. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2019arXiv190804393O/abstract. [Accessed: 26-Nov-2021].

[5] S. Saha, "A comprehensive guide to Convolutional Neural Networks‐the eli5 way," *Medium*, 17-Dec-2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed: 26-Nov-2021].

[6]. Alfredo Canziani & Eugenio Culurciello, Adam Paszke, "An Analysis Of Deep Neural Network Models For Practical Applications". *arXiv preprint:* arXiv:1605.07678, 2017.

[7] V. Lyashenko, "Data Augmentation in python: Everything you need to know," *Data Augmentation in Python: Everything You Need to Know*, 12-Nov-2021. [Online]. Available: https://neptune.ai/blog/data-augmentation-in-python. [Accessed: 27-Nov-2021].