2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) | 979-8-3503-9874-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/SCEECS57921.2023.10063126

Segregation of Solid Municipal Waste Using Machine Learning

Abhijeet Pandey[1], Bhavi Khator[2], Dhruvi Agrawal[3], Danish Halim[4], and J Sathish Kumar[5]
Computer Science and Engineering
MNNIT Allahabad
p.abhijeetp94@gmail.com[1], *bhavikhator@gmail.com*[2], *dhruviagrawal2@gmail.com*[3], *danishhalim@gmail.com*[4], *sathish613@mnnit.ac.in*[5]

***Abstract*—As the world's population is growing, so is the generated waste. To process the trash efficiently, we must separate recyclable and organic waste from non-biodegradable waste. This paper discusses an approach based on machine learning, which classifies the images of garbage into two types, Organic and Recyclable. The algorithm uses a Convolutional Neural Network (CNN), a well-known deep-learning technique for models based on data sets consisting of images and pictures. The proposed model in this paper is DenseNet-169, which contains a total of 169 layers. Data augmentation and normalization are carried out to increase the number of training images and variable sizes of the images. Results were obtained by changing the train and test split, showing different accuracies corresponding to them. The results show that the model can accurately classify unknown images. The highest achieved accuracy was about 98.630%, with the proposed model.**

***Index Terms*—CNN, Neural Networks, Epochs, ReLU, Machine Learning, Waste Segregation, Classification, DenseNet, Deep Learning.**

I. Introduction

The economic and industrial activities in today's era of globalization and population growth have led to a worldwide problem of extreme relevance, i.e., a massive increase in the amount of waste produced. The amount of waste generated every year is increasing at an alarming rate. For instance, in 2016, the worldwide waste produced was around 2.02 billion tons, which is expected to increase to 2.59 billion tons in 2030, and 3.4 billion tons in 2050 [1]. What is more terrifying is that, out of all the garbage being produced, only 20% of it is properly recycled and reused [1]. If we look at some more statistics about the production of waste over the years-

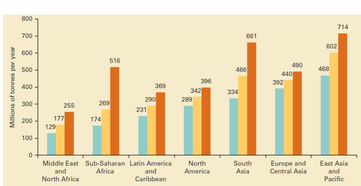

Fig. 1. Waste Generation Across Continents [2]

According to The World Bank, the world generates 2.01 billion tonnes of municipal solid waste (MSW) annually. Figure 1 illustrates how this problem of ever-increasing waste is a

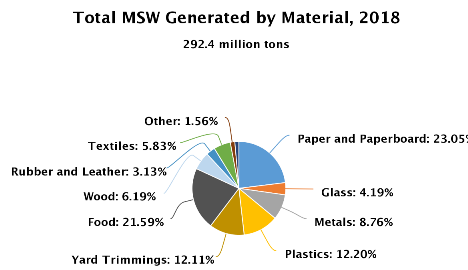

Fig. 2. Total MSW Generated Waste [2]

trend that does not show any signs of stopping. A conservative estimate shows that, at minimum, 33% of this waste is not handled in an environmentally safe manner. Different types of garbage must be acknowledged to understand how to improve the approach to managing MSW, which can be divided into recyclable and non-recyclable MSW. Figure 2 illustrates the break-up of the different waste types, with the slices in shades of blue depicting recyclable waste. Environmental Protection Agency(EPA) mentions that in 2018, the produced MSW was around 292.4 million tons in the USA itself and around 62 million tons in India. Talking about the scenario in India, out of 62 million tons, only around 12 million tons get treated, and other garbage is dumped in landfills or the ocean. The generation is expected to rise up to 200 million tons by 2041 [3].

From the statistics seen in the past few paragraphs, it can be concluded that the problem of garbage management is growing and needs special attention. A good amount of research is being done in the field of waste management to tackle this issue. We must act quickly to increase the amount of processed and recycled waste. The more amount of waste lies freely and untreated on earth, the less we will be able to utilize land resources. The unprocessed trash not only affects lands but also forms mountains beneath the surface of the sea and affects sea life. The first step toward boosting the amount of recycled garbage is to segregate it from other significant types of trash, including organic waste.

II. Related Work

In this section, This paper will explain some of the major works which have been done in this field and are related to the proposed task.

Mr. Wahid et al. [4] talked about using CNN to examine image data sets to classify the data into two different categories in their paper published in 2020. They were able to achieve a classification accuracy of 95.3125% using their model. The classes for segregation mentioned in their report were Digestible and Indigestible Waste. Their article divided the data set into three parts 80% of all the data is used as training data, 10% is used as validation data, and the remaining 10% is used for testing purposes. They used a model having 34 layers.

Mr. Susanth et al. [5] talked about using four different models, ResNet50, DenseNet169, VGG16, and AlexNet, to segregate between various classes of image data sets consisting of pictures of garbage. Their paper talks about segregating the images into different types of image classes, such as glass, paper, plastic, metal, and cardboard, in which most of the classes come under the category of solid and recyclable waste. They observed that the maximum accuracy was achieved when using the DenseNet169 Model, around 94%. Also, the accuracy of AlexNet50 was observed to be very close to DenseNet169.

Srinilta et al. [6] have done a similar task of segregating the garbage images into different classes such as General, Hazardous, Recyclable, and Compostable waste. Their paper used four distinct pre-trained models, i.e., VGG-16, ResNet-50, MobileNet V2, and DenseNet-121, to classify 9200 municipal solid waste images. They have used 70% of all the images for training the model and kept the other 30% for testing it in real-world scenarios. They also used the 10-fold cross-validation technique to test the accuracy over 30 epochs. They achieved a maximum classification accuracy of 94.86% using the ResNet-50 Model.

Mr. Adedeji et al. [7] used a 50-layer pre-train ResNet-50 CNN model and SVM on an image dataset containing solid waste images. Their paper involved segregating solid waste into four classes, i.e., glass, metal, plastic, and paper. Their model achieved a maximum classification frequency of 87% using the discussed model.

Mindy Yang and Gary Thung [8] collected hand-clicked images for building a data set. Their paper mentioned that their data set includes about 400 pictures for each category. They divided the images into six classes for classification purposes. They used two machine learning models, Support Vector Machine (SVM) and CNN. On their data set, SVM performed better than CNN. Using SVM, they achieved a maximum accuracy of 63% while CNN was just able to obtain an accuracy of 23%.

It has been observed from the previous few paragraphs that the maximum obtained classification accuracy is 95% by far. So the following sections explain the approach that will be followed to try to increase classification accuracy.

III. Proposed Work

In the previous sections, some problems faced due to the lack of unprocessed data have been discussed along with some of the past works with similar objectives. Now this paper will discuss the approach proposed in the following sections to build a model that will classify all the images containing waste items into two primary categories: **Organic** and **Recyclable** Waste. But before talking about the model, we will first discuss the approach for creating a clean data set on which the model will be trained, and it will learn to classify the provided images based on its attribute.

A. Data set

The first step toward developing a good machine-learning model is to have a good data set in which all the major attributes are clearly visible and can be used by the machine to interpret and differentiate between various types of images. First of all, an adequate amount of images are required to train the model well. If the number of training images is low, then the model will not be able to learn properly. This will result in lesser testing accuracy. It will mean that the model is not able to fit properly. The data set that this paper has used to train the model is the Waste Classification data by Sashaank Sekar [9]. The database is split into two categories Training Data(85%) and Test Data(15%). Which in numbers contain 22564 training images and 2513 test images.
The database is as below:

TABLE I
Data set Division

	Organic	Recyclable
Train	12565	9999
Test	1401	1112

While training the model with the given data, one more part is created from the training data set, called the validation data set. 15% of the training data is taken out for the validation data set.

B. Data Augmentation

Although 22,500 images in the data set can look like a considerable number, it is relatively small to train a model containing many layers. This paper has used the DenseNet-169 model for the proposed approach, a predefined model containing 169 layers. So the requirement of images to train this model is very high. In these situations, the technique of data augmentation comes into the picture.

Data Augmentation can increase the number of images quite significantly. In this technique, some of the features are modified in different photos. Some of the changes are horizontal flip, vertical flip, zooming, re-scaling, rotation, and resizing the images. This generates more images from existing pictures, which ends up making a large data set. It also increases the variety in the data set, such that the data set does not end up getting monotonous. A monotonous data set leads to the overfitting of the model. Overfitting is a phenomenon where the model gives high accuracy on the training data set by making changes that are specific to the training data set. When encountered with any other data set, it performs poorly because the model is not generic. Data augmentation helps in preventing the problem of overfitting.

The augmentation strategy which is followed in the proposed model is represented in the table II.

TABLE II
DATA AUGMENTATION STRATEGY

Horizontal Flip	True
Vertical Flip	True
Rotation Range	10°
Zoom Range	0.4
Rescale Image	1.0/255

Along with data augmentation, another technique is used to deal with the issue of images of different sizes. This technique is called Normalization, where the values of each pixel are scaled to a standard scale which is small and can be processed efficiently. It helps in improving the performance and increasing the accuracy of the model.

C. Model

Till now, the generation, cleaning, and populating of the data set have been discussed. Although data is a vital part of the process, selecting the best model to train on the given data is also essential. If you have a good data set but use an irrelevant model on that data, the model will not produce good results. So it is equally important to select a suitable learning model.

This paper has used machine learning, specifically deep learning, at the core of our model. Deep learning is a subset of machine learning that deals with the areas involving neural networks. Deep learning is mainly used on data sets that involve many input features and attributes. It involves multiple layers, which progressively process the higher-level features of the data. Diving further, the proposed model is based on Convolutional Neural Networks (CNN), which are known for their ability to process data involving pictures and images.

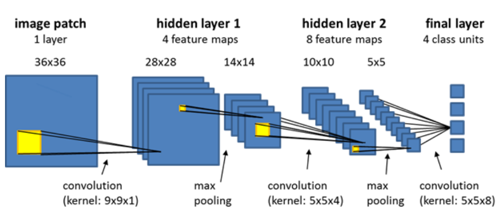

Fig. 3. CNN Architecture [10]

CNN (also called ConvNet) is a deep-learning algorithm primarily used for image and picture data sets and analysis. First, the model takes an image as the input, then assigns weights to various attributes or objects in the photo, segregating one picture from another. CNNs are famous because they require minimal pre-processing compared to most other classification models. A CNN model consists of three major types of unique hidden layers: 1) Convolutional Layer 2) Pooling Layer 3) Fully Connected Layer.

The main task of the Convolutional Layers is to extract attributes from an image with the help of different filters. A filter can be thought of as a mask or a small matrix that is filled using random values, it is generally a matrix of small dimensions. These matrices are used to extract features or detect patterns which are then fed into the next layer as the input. The output generated by these layers is generally a feature map represented by a matrix.

The pooling layer is used to summarize a region's feature from the whole feature map, which is obtained by the convolutional layer. This layer uses a two-dimensional window of tiny size (generally 2*2). This window is iterated throughout the feature map to select a cumulative feature from a small region. The criteria depend on the type of pool used. The two major types of pools are Max Pool and Average Pool. The maximum value from the window is selected in the max pool, and in the average pool, the average value in the whole window is chosen for the output matrix. The output from this layer is passed into the next layer as input.

Finally, the Fully Connected Layer is the layer that carries out the actual classification task. The outputs from the pooling layer, which are the shrank version of the real images, are taken and then converted to a single vector. This vector is used for comparison, i.e., it is compared with different vectors obtained from the trained images. Which then classifies the image to generate the output.

For the base model, **DenseNet169** is used. DenseNet contains a total of 169 layers connected in a feed-forward way, where each layer is connected to every other layer. In other words, the output from a layer is passed to every successor layer, and every layer gets input from every predecessor layer. This is the reason why every layer gets lossless features even after giving through a large number of layers. Also, in DenseNet, the number of parameters gets reduced due to the reuse of functions. This is why DenseNets can be a good choice when dealing with images having many features. Because the loss of attributes in DenseNets is minimal, DenseNets can produce good results. One of the significant drawbacks of DenseNets is that these are very complex, so if the data is not clean, the model can take a reasonable amount of time to train, and if the data set size is low, it can result in overfitting.

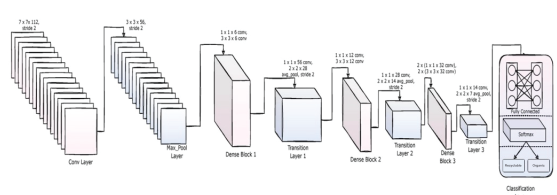

Fig. 4. DenseNet-169 Architecture Flowchart [11]

Along with the basic model implementation, the approach in this paper also implemented some additional features, which can increase the model's efficiency significantly. Two of these features or callbacks are:

Early Stopping is a technique of stopping the model when there is no improvement in already achieved accuracy. This helps reduce the probability of unnecessary extra fitting of the model and saves the model from situations like overfitting. In the end, it improves the performance of the model.

Checkpointing is a process of saving the progress of the model at a defined interval, It includes saving the current states, current weights, and current attributes of different neural networks in the model at any given point in time. Checkpointing makes the model fault-tolerant, i.e., if, due to any issue, the model stops in the middle, without checkpointing, the whole process needs to be restarted, and all the states and progress will be lost. If the model checkpoints at regular intervals, the progress will be restored, and the model will recover to its previously saved states.

D. Model Characterstics

As discussed in section III-C, the base model for the proposed algorithm is DenseNet-169. Some additional layers are added on top of this base model. These layers are in the following order:

1) Dropout, 2) Flatten, 3) Batch Normalization, 4) Dense Layer (With ReLU Activation Function), 5) Batch Normalization, 6) Dropout, 7) Dense Layer (With ReLU Activation Function), 8) Batch Normalization, 9) Dropout, 10) Dense Layer (With ReLU Activation Function), 11) Dropout, 12) Dense Layer (Output Layer) (With Softmax Activation Function).

A Dropout layer, to avoid overfitting, randomly sets input units to 0 with a specified probability during the training process.

A Flatten layer is used to change the dimensions of input data, it collapses a multidimensional input into a single-dimensional output, that's why called Flatten layer.

A Batch Normalization Layer is used to normalize the input data into mini-batches so that the model can be more stable and trained efficiently.

A Dense Layer is a unique type of layer that contains an Activation Function for the neuron. It removes linearity from the data so that the model can be appropriately trained. The activation function used in the discussed model is the Rectified Linear Unit (ReLU) Activation Function.

$$ReLU : g(x) = max(0, x) \quad (1)$$

The final dense layer is the output layer which will categorize the image into one of the two classes. For this layer, the softmax activation function is used.

$$\sigma(z)_i = \frac{e^{-B_{z_i}}}{\sum_{j=1}^{K} e^{-B_{z_j}}} \quad (2)$$

The AUC (Area Under Curve) metric calculates the model's performance. AUC metric is very similar to accuracy but slightly better when the model is used for classification. The more the AUC, the better the classification results.

The **Error Function** used for the proposed model is the Categorical Crossentropy Function, which is one of the best and most used error functions for a classification model. When the data set includes multi-class data, i.e., data with two or more classes, then this loss function is used.

$$H(x) = -\Sigma_x p(x) log(p(x)) \quad (3)$$

where p(x) is the probability distribution of a random variable x.

Adam Optimizer is used to optimize the model on the given error function, which is Categorical Crossentropy. Based on first and second-order moments, The Adam optimizer is a stochastic gradient descent model.

- Number of Epochs: 10
- Training Split: 70
- Validation Split: 15
- Testing Split: 15

E. Final Model

After compiling all the information from this section's previous subsections, the final model is obtained. The Structure of the final model can be seen in table III and Figure 5.

TABLE III
FINAL MODEL

Layer	Param #	Output Shape
DenseNet169	12642880	(None, 5, 5, 1664)
Dropout	0	(None, 5, 5, 1664)
Flatten	0	(None, 41600)
Batch Normalization	166400	(None, 41600)
Dense	208005000	(None, 5000)
Batch Normalization	20000	(None, 5000)
Dropout	0	(None, 5000)
Dense	5001000	(None, 1000)
Batch Normalization	4000	(None, 1000)
Dropout	0	(None, 1000)
Dense	500500	(None, 500)
Dropout	0	(None, 500)
Dense (Output)	1002	(None, 2)

Model: Sequential
Total params: 226,340,782
Trainable params: 213,602,702
Non-trainable params: 12,738,080

IV. EXPERIMENTAL SETUP

In the previous sections, details about the proposed model and algorithms have been discussed. Now, the point of discussion will be implementing the proposed work with the help of programming to realize the proposed work.

Python was chosen as the primary programming language for implementing the machine learning algorithm. Python is one of the most widely used programming languages that supports various libraries just for machine learning. Out of these libraries, **TensorFlow and Keras** are used in the implementation discussed in this paper. TensorFlow is a machine

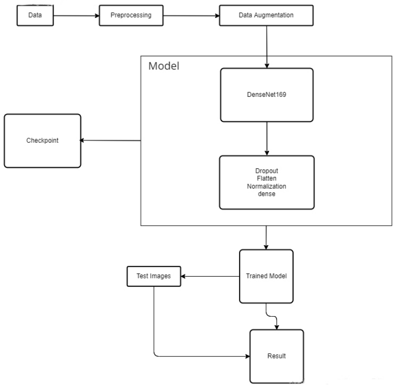

Fig. 5. Proposed Model

learning library by Google, which is also open-source. Keras is a subset of TensorFlow, which deals with the field of Deep Learning. These libraries include multiple power pre-defined models and mathematical tools for the tasks like Model Implementation, Data Augmentation, Image Conversion, and many more. Being open source, their libraries are continuously improved by the vast community. Programming was done on **Google Colab** [12], which is an online Jupiter Notebook where multiple people can collaborate simultaneously. It works on a remote run-time environment in the server of google. That is why Colab does not load the client's computer and gives high performance. Colab is a great tool to work on a research project as it allows multiple mathematical libraries to be preinstalled in the run-time and enables the user to install additional libraries. Google Colab can directly mount with **Google Drive** so that the data can also be accessed from the cloud. The data set was imported from **Kaggle**, a web-based data science environment, which enables its users to publish and access data to build and train a model. The data set described in the section III-A is also taken from a repository in Kaggle.

V. RESULTS

This paper has discussed the details of the proposed model and its implementation using python and TensorFlow on Google Colab [12]. The model is run on the cleaned and augmented data set. The structure of the additional layers was the same, but the base model was changed to compare different pre-trained models, including Logistic Regression, VGG16, and DenseNet169. Out of these models, DenseNet169 performed the best and gave the best AUC of 98.630% in the validation data set. The error and AUC curves for each epoch of the DenseNet Model can be seen in figure 6 & 7, respectively.

Figure 8 shows the comparison curve between the DenseNet169 and VGG16 models.

Additionally, to test the DenseNet169 model more thoroughly, another 15% of the data was divided for testing the model after the training and validation were done, Which gave

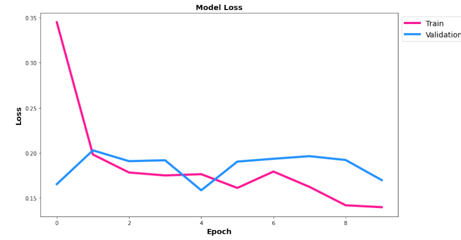

Fig. 6. Error vs. Epoch Number Curve

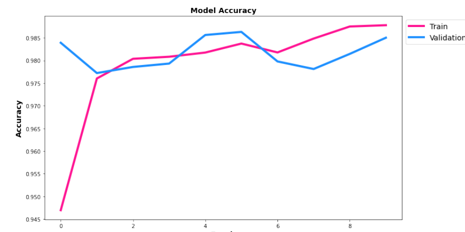

Fig. 7. Accuracy vs. Epoch Number Curve

a total of 2513 images. The accuracy obtained from the testing data was 96.80%. Figure 9 shows the confusion matrix for the testing data to better visualize the achieved results. A confusion matrix in an $N \times N$ matrix when the data includes N output or target classes. The columns include the Actual value, and the rows represent the Predicted value by the model. A confusion matrix is beneficial in measuring the performance of a classification model. In the matrix (figure 9), the "Organic" Waste class is represented as 0, and the "Recyclable" Waste class is 1.

The model's classification report can be generated using

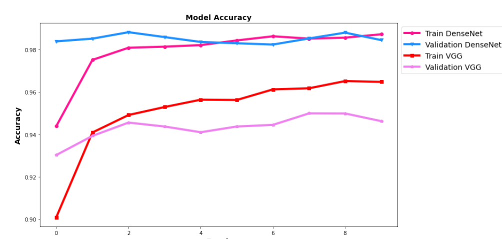

Fig. 8. VGG16 vs DenseNet169 Accuracy Comparison

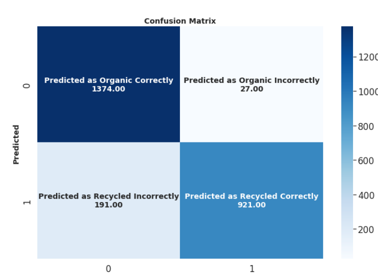

Fig. 9. Confusion Matrix for DenseNet169

the confusion matrix. It is vital in measuring the accuracy of predictions made by the Model. The Organic Class is taken as the positive class, and the Recyclable class as the negative class. Fig. 10 shows the classification report for DenseNet169. True Positive (TP) = Correctly Predicted Images of Organic (Positive) class, True Negative (TN) = Correctly Predicted Images of Recyclable (Negative) class, False Positive (FP) = Incorrectly Predicted Images of Organic (Positive) class, False Negative (FN) = Incorrectly Predicted Images of Recyclable (Negative) class.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1\_Score = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{6}$$

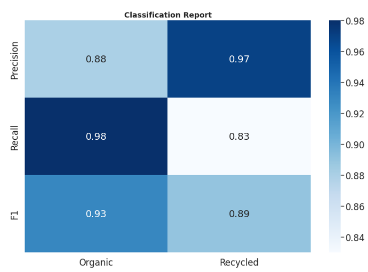

Fig. 10. Classification Report for DenseNet169

The DenseNet169 model performed much better than Logistic Regression and VGG16 models. Table IV depicts the accuracy comparison of these three models.

TABLE IV
ACCURACY COMPARISON OF DIFFERENT MODELS

Model	Accuracy
Logistic Regression	93.00%
VGG16	96.10%
DenseNet169	98.63%

VI. CONCLUSION

Deep Learning algorithm DenseNet169 is best suited for classifying solid waste into two waste classes, i.e., Organic and Recyclable waste. The implementation of the models was carried out using python programming language, with the help of TensorFlow and Keras libraries. This paper compares the accuracy of the DenseNet model with the performance of Logistic Regression and the VGG16 Models. The model which used Logistic Regression achieved a maximum classification accuracy of 93% in the validation data set. Additionally, the VGG16 implementation performed better than Logistic Regression, reaching a maximum accuracy of 96%. The DenseNet169 model achieved a maximum validation accuracy (AUC) of 98.630% with a 0.152 loss value. The loss function used for logistic regression was Binary Crossentropy, and for VGG16 and DenseNet169 Models, Categorical Crossentropy was used as the loss function. Adam Optimizer was used to optimize the models over the given loss function, which is a stochastic gradient descent algorithm. Some additional layers were added on top of the base models to generate the desired outputs. Finally, it is concluded that after cleaning and populating the default data set, the model with DenseNet-169 as the base model produced the best results on the data by achieving maximum frequency and producing the slightest error compared to the other models.

For future aspects, we have decided to implement the model on the hardware level, where smart bins will segregate different municipal waste into separate compartments inside the container according to the categorized classes so they can be processed differently.

REFERENCES

[1] Statista, "Global waste generation - statistics & facts," 2022. [Online]. Available: https://www.statista.com/topics/4983/waste-generation-worldwide/#topicOverview

[2] E. P. Agency, "National overview: Facts and figures on materials, wastes and recycling," 2019. [Online]. Available: https://www.epa.gov/facts-and-figures-about-materials-waste-and-recycling/national-overview-facts-and-figures-materials

[3] Statista, "Msw generated across india from 2001 to 2041," 2021. [Online]. Available: https://www.statista.com/statistics/1009110/india-msw-generation-amount/

[4] M. W. Rahman, R. Islam, A. Hasan, N. I. Bithi, M. M. Hasan, and M. M. Rahman, "Intelligent waste management system using deep learning with iot," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, pp. 2072–2087, 2020.

[5] G. S. Susanth, L. J. Livingston, and L. A. Livingston, "Garbage waste segregation using deep learning techniques," in *IOP Conference Series: Materials Science and Engineering*, vol. 1012. IOP Publishing, 2021, p. 012040.

[6] C. Srinilta and S. Kanharattanachai, "Municipal solid waste segregation with cnn," in *proceedings of 5th International conference on engineering, applied sciences and technology (ICEAST)*. IEEE, 2019, pp. 1–4.

[7] O. Adedeji and Z. Wang, "Intelligent waste classification system using deep learning convolutional neural network," *Procedia Manufacturing*, vol. 35, pp. 607–612, 2019.

[8] M. Yang and G. Thung, "Classification of trash for recyclability status," *CS229 project report*, vol. 2016, no. 1, p. 3, 2016.

[9] S. Sekar, "Waste classification data," Kaggle, 2021. [Online]. Available: https://www.kaggle.com/datasets/techsash/waste-classification-data/

[10] Trimble, "Using deep learning models / convolutional neural networks," 2021. [Online]. Available: https://docs.ecognition.com/v10.0.2/eCognition_documentation/User%20Guide%20Developer/8%20Classification%20-%20Deep%20Learning.htm

[11] A. Vulli, P. Naga Srinivasu, S. K. S. Madipally, J. Shafi, J. Choi, and M. F. Ijaz, "Fine-tuned densenet-169 for breast cancer metastasis prediction using fastai and 1-cycle policy," *Sensors*, vol. 22, 04 2022.

[12] A. Pandey, "Google colab," 2022. [Online]. Available: https://colab.research.google.com/drive/1tAEvCPitp372QeSUecWbLtKQl8klYXeW?usp=sharing