Garbage classifier 12 Classes

Fulara Utkarsh(1372892) and Kumawat Mohitkumar(1453410)

***Abstract*—Sorting waste manually can be slow and prone to mistakes, making recycling less effective. This project introduces a smart waste management system that uses deep learning, specifically Convolutional Neural Networks (CNNs), to automatically classify waste into 12 categories. By analyzing images and recognizing different types of waste makes sorting easier. *But Can a CNN categorize different types of waste into multiple classes to simplify recycling?* To answer this, we developed a CNN-based solution that automates part of the recycling process.**

I. Intuition and Introduction to Convolutional Neural Networks (CNNs)

Humans have an incredible ability to recognize patterns instantly and subconsciously. When we look at an image, our brain can interpret it in multiple ways. For example, in Figure 1(a), if we focus on the right side of a particular optical illusion, we may see a person looking away, whereas if we focus on the left side, we perceive a person looking directly at us. This ability to recognize patterns is so fast that we are often unaware of it.

Consider another example in Figure 1(b): when we look at this image, our brain struggles to recognize what it represents. Is it two people, three people, or something else? Our brain gets confused because such shapes are uncommon in real life. This illustrates that our brain attempts to identify features and, based on those, classifies objects.

However, for a computer, image recognition is not automatic. Instead of understanding the image as humans do, computers process images as **matrices of numbers (pixel values)**. The challenge is to develop algorithms that allow machines to extract meaningful features from these numbers, just as the human brain does. Inspired by human perception, researchers developed **Convolutional Neural Networks (CNNs)**, a class of deep learning models designed to efficiently analyze visual data.

A. Assumptions for This Paper

Before diving into the details, we establish the following assumptions for clarity:

1) Image Representation:

- Images are represented as **grids of numbers**, where each element corresponds to a pixel value.
- To simplify explanations, we will focus on **black-and-white images**, meaning each pixel has a binary value (0 or 1).

2) Mathematical Prerequisites:

- Readers are expected to be familiar with **basic matrix operations**.
- To keep this paper concise, we will not explicitly explain every mathematical computation.

II. Dataset

The dataset used for our smart waste management project covers **12 distinct classes** of everyday waste materials: battery, biological, brown-glass, cardboard, clothes, green-glass, metal, paper, plastic, shoes, trash, and white-glass. These categories include Images of items like rotten produce, plastic containers, paper waste, shoes, and clothes. Each image is labeled under the correct category, allowing the system to learn how to identify and sort these items based on their visible features.

III. Image Pre-Processing

CNNs require fixed input dimensions to ensure consistency across the dataset. In this study, all images are resized to **128×128 pixels** to standardize input size and maintain compatibility with the CNN architecture. This resizing process helps prevent computational inefficiencies while preserving spatial information for effective feature extraction.

Raw images contain pixel values in the range **[0,255]**, which may lead to large activation values and slow convergence during training. To address this, **normalization** is applied by scaling all pixel values to the range **[0,1]** through division by 255. This transformation reduces numerical instability, accelerates training, and enhances model performance.

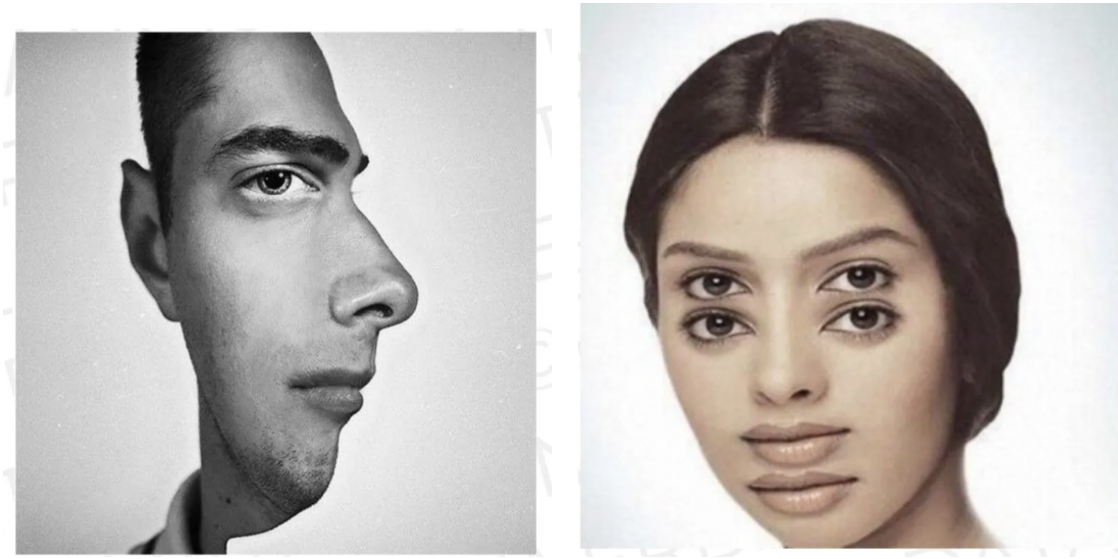
To improve dataset diversity and mitigate overfitting, **data augmentation** techniques are employed. These include **random rotations, horizontal and vertical flipping, zooming, brightness adjustments, and translations**. By introducing such variations, the CNN learns robust and generalized features rather than memorizing training samples, thereby improving real-world classification performance.

The dataset is partitioned into **75% training** and **25% validation** subsets. The training set is utilized for optimizing model parameters, while the validation set assesses the model's generalization ability on unseen data.

By implementing these image pre-processing techniques, the CNN becomes more resilient, effectively classifies different waste categories, and contributes to the advancement of **intelligent waste management systems**.

IV. Convolution

Convolutional Neural Networks (CNNs) are inspired by the way humans recognize objects by identifying patterns. However, computers require a method to detect these patterns, which is where **convolution** comes into play. Convolution is a fundamental mathematical operation in CNNs that enables pattern detection in images by applying **feature detectors**—also known as **filters** or **kernels**—to an image.



(a) Dual perception (b) Ambiguous facial Features

Fig. 1. Examples of pattern recognition and visual ambiguity. The left image can be interpreted in two ways depending on where one focuses, while the right image causes difficulty in feature recognition due to its unusual structure.

This operation extracts important features while reducing spatial dimensions, thereby preserving essential details. A CNN typically applies multiple filters in a single layer to capture different features. For instance, in our **Garbage Classifier Model**, we used a **3×3 feature map** with **32 filters** in the first layer. Each filter detects specific patterns, such as edges, textures, or object parts.

Early layers detect simple features like horizontal or vertical lines, while deeper layers capture complex structures such as curves and object shapes. For intuition, consider **Figure 2**, which shows an image of the Taj Mahal processed with an edge detection filter, highlighting structural features.

After convolution, the resulting feature maps are passed through non-linear activation functions such as ReLU or Swish, which introduce non-linearity into the model. This step is crucial because real-world images exhibit non-linear transformations. For instance, consider two surfaces—one red and one blue. If brightness increases, the intensity change does not follow a simple linear pattern; the red surface may respond differently than the blue surface. By applying non-linearity, CNNs can better capture complex patterns and hierarchical features, allowing deeper layers to learn more abstract representations.

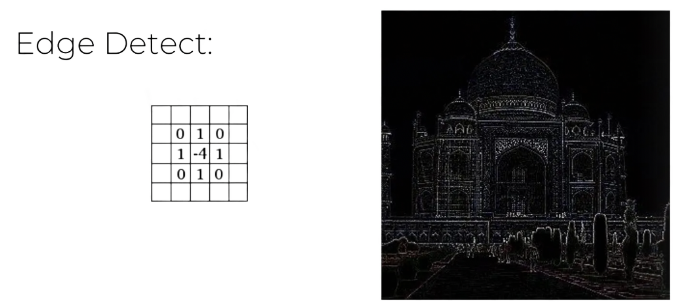


Fig. 2. Edge Detection filter on Tajmahal

V. Pooling in Convocutional Neural Network

Pooling is an operation in Convolutional Neural Networks (CNNs) that helps reduce the size of feature maps while preserving the most important information. This makes deep learning models more efficient and ensures they can recognize objects accurately, even if they appear in different positions or sizes. In our Smart Waste Classification System, pooling plays a vital role in improving performance by reducing the number of computations needed, preventing overfitting, and ensuring the system correctly identifies different types of waste. Since waste objects can be captured in various orientations, pooling helps the model remain flexible and accurate despite these variations. There are two main types of pooling: Max Pooling and Average Pooling. Max Pooling, which we use in our project, selects the highest pixel value in a given window, allowing the network to retain the most important features. Mathematically, this means taking the maximum value from a small section of the feature map and passing it forward while ignoring the rest. This helps in identifying key patterns like sharp edges, textures, and shapes—features that are especially important when distinguishing between waste materials like plastic, metal, or paper. On the other hand, Average Pooling takes the mean value of pixels in a window, which smooths out variations but can blur essential details, making it less useful for object classification. Since our waste classifier needs to pick up on specific details to differentiate between different types of materials, Max Pooling is the better choice.In our model, we apply MaxPooling2D(pool_size=(2,2), strides=2) at multiple layers to gradually reduce the size of the feature maps while keeping the most important information. For example, if an input image starts at 128×128 pixels, applying

Max Pooling reduces it to 64×64, then 32×32, and so on. This not only speeds up processing but also ensures that the model focuses on the most relevant characteristics of waste objects making classification more accurate. Additionally, pooling helps prevent overfitting by reducing unnecessary details and forcing the model to generalize better across different images. By integrating pooling into our Smart Waste Classification System, we make it more efficient, scalable, and capable of real-time waste classification.

VI. Flattening

After applying the pooling layer in our model, the output is fed into a fully connected artificial neural network. However, since the pooled layers consist of multiple matrices (feature maps), we need to convert them into a single, one-dimensional array before passing them to the network. This is where the flattening step comes in. Flattening transforms the multi-dimensional feature maps into a single vector, which can then be used as input for the fully connected layer. This serves as a crucial step before the final classification stage. For intuition, consider **Fig. 3**, which shows an image in the form of a flattened matrix before being fed into the neural network.

VII. Fully Connected Artificial Neural Network

Now, after all the steps—such as convolution, pooling, and flattening—we add a fully connected artificial neural network (ANN) to our model. The output from the flattening layer serves as the input to this ANN. But what is the purpose of this ANN if the previous steps have already detected patterns? The answer is that the ANN helps to combine extracted features into higher-level attributes, allowing the model to generalize better and improve its predictions. For example, in our case, we use an ANN with two layers: the first layer has 512 neurons, and the final layer has 12 neurons, corresponding to our 12 classes. During the training phase, the model classifies images based on all these steps. The output is then compared to the actual labels using a loss function, which provides a numeric measure of how optimal the predictions are. To improve the model, a process called backpropagation is used to adjust parameters—this includes everything from the weights in convolutional filters and feature detectors to the pooling matrices and the weights of the fully connected layers. To visualize all these layers, see **Figure 4**, which represents the architecture of CNN. The figure illustrates the sequence of convolution, subsampling (pooling), and fully connected layers in a **Convolutional Neural Network (CNN)** designed for digit recognition. Each layer in the figure shows a feature map, which is created by applying a filter (pattern detector) across the whole image. The same filter is used at every position, ensuring the CNN can recognize the same pattern no matter where it appears in the image.

In the following sections, we will explain backpropagation and loss functions in detail to build a solid understanding of these fundamental concepts.

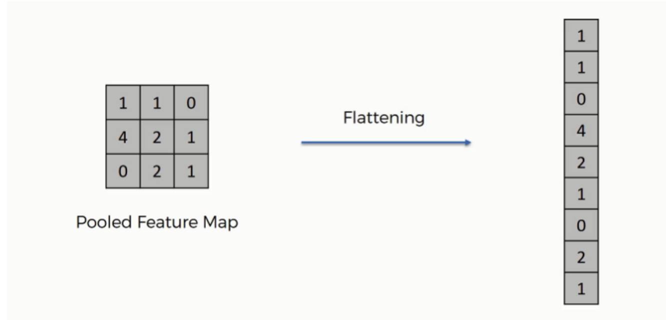


Fig. 3. Flattening of an Image

VIII. Loss Function and Backpropagation in Smart Waste Classification

In the development of our smart waste classification model, the concepts of **loss function** and **backpropagation** play an essential role in ensuring accurate learning and performance improvement. The **loss function** quantifies how far the predicted output is from the actual class labels, allowing the model to adjust its internal parameters to minimize classification errors over successive training iterations.

For this multi-class classification task, we employ the **categorical cross-entropy loss function**, as it is specifically designed for classification problems with multiple categories, such as plastic, metal, glass, and organic waste. This function calculates the difference between the predicted probability distribution and the true class labels, penalizing incorrect predictions. Mathematically, the loss function is given by:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{1}$$

where:

- C is the number of classes (12 in our case),
- y_i represents the actual class label (one-hot encoded),
- $\hat{y}_i$ is the predicted probability for class i.

Once the loss function computes the error, **backpropagation** comes into play to optimize the model parameters. Backpropagation is an iterative process that updates the neural network's weights by computing the gradient of the loss function with respect to each parameter. The process follows three main steps:

1) **Forward Pass:** The input image passes through the Convolutional Neural Network (CNN), and a prediction is made.
2) **Gradient Computation:** The loss function calculates the error, and gradients are computed using the **chain rule of differentiation**.
3) **Weight Update:** The computed gradients adjust the weights using an optimization algorithm, such as **AdamW**, to minimize the loss function iteratively.

The weight update rule is expressed as:

$$w^{(t+1)} = w^{(t)} - \eta \frac{\partial L}{\partial w} \tag{2}$$

where:

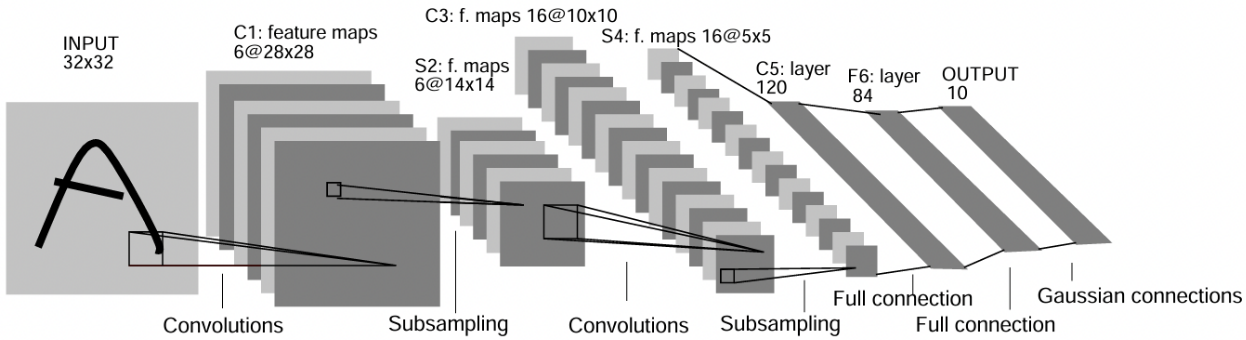


Fig. 4. Architecture of LeNet Subsampling Convolutions Subsampling Full connection Full connection a Convolutional Neural Network here for digits recognition Each plane is a feature map ie a set of units whose weights are constrained to be identical

- $w^{(t)}$ represents the current weight,
- η is the learning rate, controlling the magnitude of the update,
- $\frac{\partial L}{\partial w}$ is the gradient of the loss function concerning the weight.

By continuously adjusting the weights through backpropagation, the model improves its ability to extract key features such as textures, colors, and edges of waste materials. This iterative learning process ensures that our classifier effectively distinguishes between various waste types with high accuracy, making it a reliable tool for smart waste management.

IX. Conclusion

In this research, we presented a smart waste classification system using convolutional neural networks to automate the categorization of waste into 12 distinct classes. Throughout the project, we followed a well-structured process—including image pre-processing, normalization, and data augmentation—to ensure that the model could effectively learn to differentiate among the 12 waste categories. By utilizing Covolution, pooling and a fully connected Neural-network, we were able to extract critical features from raw images, while the integration of a categorical cross-entropy loss function with backpropagation optimized the model's performance.

Although the system performed well on the current dataset, we recognize that there is still room for improvement. Future work could focus on expanding the dataset and adding data from sensors (weight, chemical composition), which could make waste sorting easier and enhance the automation of waste management processes by reducing manual sorting errors, and contribute to more sustainable recycling practices. Overall, this project has not only deepened our understanding of CNNs and their practical applications but also laid a solid foundation for future research in Artificial Intelligence.

X. Acknowledgments

This project has been a great starting point for us in our AI journey. Before working on this, AI and AI agents felt like a mystery to us. But after completing this project, we now have a better understanding of how these fascinating tools work and the math behind them.

We would like to thank our university and our professor, Mr. Gian Luca, for giving us the chance and freedom to take this course and work on our own ideas. His lectures helped us understand the concept of convolutional neural networks, which inspired us to create a garbage classifier for our project.

Although there is still room for improvement, this project has given us a strong foundation. Moving forward, we are excited to explore these concepts more deeply and continue learning about artificial intelligence.

XI. References

References

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition,". [Online]. Available: http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf

[2] J. Wu, "Introduction to Convolutional Neural Networks, Nanjing University, China, May 1, 2017. [Online]. Available: https://cs.nju.edu.cn/wujx/paper/CNN.pdf

[3] SuperDataScience Team, "Artificial Intelligence A-Z: Learn How To Build An AI," *Udemy*. [Online]. Available: https://www.udemy.com/course/artificial-intelligence-az/.

[4] 3Blue1Brown, Neural Networks,*YouTube*. [Online]. Available: https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi.