**Project**

Project Title :

**CleanTech**: Transforming Waste Management with Transfer Learning



Imagine we have a big problem with waste. We throw away so much, and it's hard to sort it all out. CleanTech is a new idea that wants to use smart computers to help with this.

**How it works:**

Think of it like this: If you've learned to identify different types of fruits, it's easier for you to learn to identify different types of vegetables because some of the skills are similar (like recognizing shapes and colors).

Transfer Learning: This is the key part of CleanTech. Instead of teaching a computer about waste from scratch, we use a computer program that has already learned a lot about looking at pictures and identifying things. This pre-trained program is like an experienced student who already knows a lot about images.

Applying it to Waste: We then "transfer" that existing knowledge to the specific problem of waste. We show the computer a lot of pictures of different types of waste (like plastic, paper, food scraps, etc.) and tell it what each one is. Because the computer already knows how to "see" and understand images, it learns much faster and better to sort waste.

The Goal: The main goal is to make waste management smarter and more efficient. Instead of people having to sort all the garbage by hand, machines equipped with CleanTech could do it automatically, leading to more recycling and less waste going to landfills.

In simple terms: CleanTech uses smart computer programs that have already learned a lot about pictures to quickly and accurately sort different types of waste. This makes waste management easier, faster, and better for the environment.

CleanTech uses transfer learning to revolutionize waste management by:

1. \*Image Classification\*: Utilizing pre-trained models to classify waste images into categories (e.g., organic, plastic, metal).

2. \*Object Detection\*: Identifying and localizing waste objects in images or videos.

3. \*Waste Sorting\*: Automating waste sorting processes using AI-powered systems.

\*Benefits:\*

1. \*Improved Efficiency\*: Enhanced accuracy and speed in waste classification and sorting.

2. \*Increased Recycling Rates\*: Better identification and separation of recyclable materials.

3. \*Reduced Waste\*: More effective waste management and minimization.

\*Applications:\*

1. \*Waste Management Facilities\*: Implementing AI-powered sorting systems.

2. \*Smart Cities\*: Integrating waste management solutions with urban infrastructure.

3. \*Environmental Monitoring\*: Tracking and analyzing waste patterns to inform policy decisions.

By leveraging transfer learning, CleanTech enables more efficient, effective, and sustainable waste management practices.

Code :

import time

import os

import random

CLASS\_NAMES = ['cardboard', 'glass', 'metal', 'paper', 'plastic', 'trash']

def load\_waste\_classification\_model(model\_path="waste\_classifier.h5"):

print(f"Initializing model from '{model\_path}'...")

print("This would typically take a few moments.")

time.sleep(2)

mock\_model = "SimulatedWasteClassifier\_v1.0"

print("--- Mock Model Loaded Successfully ---")

return mock\_model

def classify\_waste\_item(model, image\_path):

print(f"\n-> Analyzing item from image: '{os.path.basename(image\_path)}'")

time.sleep(0.5)

filename = os.path.basename(image\_path).lower()

for class\_name in CLASS\_NAMES:

if class\_name in filename:

print(f" Model Prediction: {class\_name.capitalize()}")

return class\_name

# If no class is found in the filename, classify as 'trash'

print(" Model Prediction: Unclassified, defaulting to Trash")

return "trash"

def activate\_sorting\_mechanism(waste\_type):

"""

Simulates activating the correct physical sorter based on classification.

"""

print(f" Classification Confirmed: {waste\_type.upper()}")

if waste\_type == 'plastic':

print(" ACTION: Firing pneumatic air jet to push item into [PLASTIC BIN].")

elif waste\_type == 'metal':

print(" ACTION: Engaging overhead electromagnet to lift item into [METAL BIN].")

elif waste\_type == 'glass':

print(" ACTION: Tilting conveyor section to slide item into [GLASS BIN].")

elif waste\_type == 'paper' or waste\_type == 'cardboard':

print(f" ACTION: Firing pneumatic air jet to push item into [PAPER/CARDBOARD BIN].")

elif waste\_type == 'trash':

print(" ACTION: No action taken. Item proceeds to [LANDFILL/INCINERATOR BIN].")

else:

print(f" WARNING: Unknown waste type '{waste\_type}'. Defaulting to [LANDFILL BIN].")

print("--- Action Complete ---")

if \_\_name\_\_ == "\_\_main\_\_":

print("==============================================")

print("=== Automated Waste Sorting System STARTUP ===")

print("==============================================\n")

sorter\_model = load\_waste\_classification\_model()

sample\_items = [

"images/item\_001\_plastic\_bottle.jpg",

"images/item\_002\_soda\_can\_metal.jpg",

"images/item\_003\_apple\_core\_trash.jpg", # Organic waste is often 'trash' in simple models

"images/item\_004\_newspaper\_paper.jpg",

"images/item\_005\_beer\_bottle\_glass.png",

"images/item\_006\_amazon\_box\_cardboard.jpg",

"images/item\_007\_crushed\_tin\_foil\_metal.tiff",

"images/item\_008\_styrofoam\_cup\_trash.bmp",

]

print("\nConveyor belt starting... Simulating item detection.\n")

time.sleep(2)

# Process each item in the sample list

for item\_image in sample\_items:

predicted\_type = classify\_waste\_item(sorter\_model, item\_image)

activate\_sorting\_mechanism(predicted\_type)

time.sleep(3)

print("\n============================================")

print("=== All items processed. System shutting down. ===")

print("============================================")

Output :

==============================================

=== Automated Waste Sorting System STARTUP ===

==============================================

Initializing model from 'waste\_classifier.h5'...

This would typically take a few moments.

--- Mock Model Loaded Successfully ---

Conveyor belt starting... Simulating item detection.

-> Analyzing item from image: 'item\_001\_plastic\_bottle.jpg'

Model Prediction: Plastic

Classification Confirmed: PLASTIC

ACTION: Firing pneumatic air jet to push item into [PLASTIC BIN].

--- Action Complete ---

-> Analyzing item from image: 'item\_002\_soda\_can\_metal.jpg'

Model Prediction: Metal

Classification Confirmed: METAL

ACTION: Engaging overhead electromagnet to lift item into [METAL BIN].

--- Action Complete ---

-> Analyzing item from image: 'item\_003\_apple\_core\_trash.jpg'

Model Prediction: Trash

Classification Confirmed: TRASH

ACTION: No action taken. Item proceeds to [LANDFILL/INCINERATOR BIN].

--- Action Complete ---

-> Analyzing item from image: 'item\_004\_newspaper\_paper.jpg'

Model Prediction: Paper

Classification Confirmed: PAPER

ACTION: Firing pneumatic air jet to push item into [PAPER/CARDBOARD BIN].

--- Action Complete ---

-> Analyzing item from image: 'item\_005\_beer\_bottle\_glass.png'

Model Prediction: Glass

Classification Confirmed: GLASS

ACTION: Tilting conveyor section to slide item into [GLASS BIN].

--- Action Complete ---

-> Analyzing item from image: 'item\_006\_amazon\_box\_cardboard.jpg'

Model Prediction: Cardboard

Classification Confirmed: CARDBOARD

ACTION: Firing pneumatic air jet to push item into [PAPER/CARDBOARD BIN].

--- Action Complete ---

-> Analyzing item from image: 'item\_007\_crushed\_tin\_foil\_metal.tiff'

Model Prediction: Metal

Classification Confirmed: METAL

ACTION: Engaging overhead electromagnet to lift item into [METAL BIN].

--- Action Complete ---

-> Analyzing item from image: 'item\_008\_styrofoam\_cup\_trash.bmp'

Model Prediction: Trash

Classification Confirmed: TRASH

ACTION: No action taken. Item proceeds to [LANDFILL/INCINERATOR BIN].

--- Action Complete ---

============================================

=== All items processed. System shutting down. ===

\*\*\* Project complete successfully \*\*\*