Sure! We'll modify the model to classify waste into three categories: paper, plastic, and others. Here’s an updated implementation to capture images using a webcam, classify them, and control servo motors based on the classification.

### Step 1: Prepare the Dataset

Ensure your dataset includes images labeled as `paper`, `plastic`, and `other`.

### Step 2: Train the Model on a Powerful Machine

Train your model to classify images into three categories: `paper`, `plastic`, and `other`.

```python

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.applications import MobileNetV2

from tensorflow.keras.models import Model

from tensorflow.keras.layers import Dense, GlobalAveragePooling2D

# Load and preprocess the dataset

train\_datagen = ImageDataGenerator(rescale=1.0/255, validation\_split=0.2)

train\_generator = train\_datagen.flow\_from\_directory(

'path\_to\_dataset',

target\_size=(224, 224),

batch\_size=32,

class\_mode='categorical',

subset='training'

)

validation\_generator = train\_datagen.flow\_from\_directory(

'path\_to\_dataset',

target\_size=(224, 224),

batch\_size=32,

class\_mode='categorical',

subset='validation'

)

# Load the MobileNetV2 model pre-trained on ImageNet

base\_model = MobileNetV2(weights='imagenet', include\_top=False)

x = base\_model.output

x = GlobalAveragePooling2D()(x)

x = Dense(1024, activation='relu')(x)

predictions = Dense(3, activation='softmax')(x) # 3 classes: paper, plastic, other

model = Model(inputs=base\_model.input, outputs=predictions)

# Freeze the base model layers

for layer in base\_model.layers:

layer.trainable = False

# Compile the model

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

# Train the model

model.fit(train\_generator, validation\_data=validation\_generator, epochs=10)

# Save the model

model.save('waste\_classifier\_model.h5')

```

### Step 3: Convert the Model to TensorFlow Lite

```python

import tensorflow as tf

# Load the model

model = tf.keras.models.load\_model('waste\_classifier\_model.h5')

# Convert the model

converter = tf.lite.TFLiteConverter.from\_keras\_model(model)

tflite\_model = converter.convert()

# Save the TensorFlow Lite model

with open('waste\_classifier\_model.tflite', 'wb') as f:

f.write(tflite\_model)

```

### Step 4: Load and Run the TensorFlow Lite Model on Raspberry Pi with Webcam

1. \*\*Capture Image Using Webcam\*\*:

- Use OpenCV to capture images from the webcam.

- Classify the images into `paper`, `plastic`, and `other`.

```python

import cv2

import tensorflow as tf

import numpy as np

from PIL import Image

# Load the TFLite model and allocate tensors

interpreter = tf.lite.Interpreter(model\_path="waste\_classifier\_model.tflite")

interpreter.allocate\_tensors()

# Get input and output tensors

input\_details = interpreter.get\_input\_details()

output\_details = interpreter.get\_output\_details()

# Function to preprocess the image

def preprocess\_image(img):

img = cv2.resize(img, (224, 224))

img = np.array(img, dtype=np.float32)

img = np.expand\_dims(img, axis=0) # Add batch dimension

img /= 255.0 # Normalize to [0, 1]

return img

# Function to classify the waste

def classify\_waste(img):

img = preprocess\_image(img)

interpreter.set\_tensor(input\_details[0]['index'], img)

interpreter.invoke()

output\_data = interpreter.get\_tensor(output\_details[0]['index'])

return np.argmax(output\_data) # Returns the class index

# Initialize webcam

cap = cv2.VideoCapture(0)

if not cap.isOpened():

print("Error: Could not open webcam.")

exit()

while True:

# Capture frame-by-frame

ret, frame = cap.read()

if not ret:

print("Error: Failed to capture image.")

break

# Display the resulting frame

cv2.imshow('frame', frame)

# Press 'q' to quit the webcam feed

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# Classify the current frame

class\_index = classify\_waste(frame)

print("Class Index:", class\_index)

# Add your servo control logic here based on class\_index

if class\_index == 0:

set\_servo\_angle(90) # Paper

elif class\_index == 1:

set\_servo\_angle(0) # Plastic

else:

set\_servo\_angle(180) # Other

# When everything is done, release the capture and close windows

cap.release()

cv2.destroyAllWindows()

```

### Step 5: Connect to Servo Motors

Use the GPIO pins on the Raspberry Pi to control the servo motors for the sorting mechanism.

```python

import RPi.GPIO as GPIO

import time

# Set up GPIO pins

servo\_pin = 17

GPIO.setmode(GPIO.BCM)

GPIO.setup(servo\_pin, GPIO.OUT)

# Set up PWM for servo motor

pwm = GPIO.PWM(servo\_pin, 50) # 50 Hz

pwm.start(0)

# Function to rotate the servo motor to a specific angle

def set\_servo\_angle(angle):

duty = angle / 18 + 2

GPIO.output(servo\_pin, True)

pwm.ChangeDutyCycle(duty)

time.sleep(1)

GPIO.output(servo\_pin, False)

pwm.ChangeDutyCycle(0)

# Example usage

set\_servo\_angle(90) # Rotate to 90 degrees

```

### Integrate Image Classification with Sorting Mechanism

Combine the image classification and servo control.

```python

while True:

# Capture frame-by-frame

ret, frame = cap.read()

if not ret:

print("Error: Failed to capture image.")

break

# Display the resulting frame

cv2.imshow('frame', frame)

# Press 'q' to quit the webcam feed

if cv2.waitKey(1) & 0xFF == ord('q'):

break

# Classify the current frame

class\_index = classify\_waste(frame)

print("Class Index:", class\_index)

# Control the servo based on the classification

if class\_index == 0:

set\_servo\_angle(90) # Paper

elif class\_index == 1:

set\_servo\_angle(0) # Plastic

else:

set\_servo\_angle(180) # Other

# When everything is done, release the capture and close windows

cap.release()

cv2.destroyAllWindows()

```

### Conclusion

By following these steps, you can create a Raspberry Pi-compatible AI model that detects and sorts waste into three categories: paper, plastic, and others. This involves setting up a webcam for image capture, preprocessing the images, classifying them using a TensorFlow Lite model, and integrating the results with a servo motor control to sort the waste.