```
import tensorflow as tf  # Core TensorFlow library
from tensorflow.keras import layers, models, optimizers, callbacks  # Layers, model creation, optimizers, and training callbacks
from tensorflow.keras.models import Sequential, load_model  # For sequential model architecture and loading saved models
from tensorflow.keras.applications import EfficientNetV2B0  # Pretrained EfficientNetV2B0 model for transfer learning
from tensorflow.keras.applications.efficientnet import preprocess_input  # Preprocessing function specific to EfficientNet
import numpy as np  # Numerical operations and array handling
import matplotlib.pyplot as plt  # Plotting graphs and images
import seaborn as sns  # Plotting graphs and images
from sklearn.metrics import confusion_matrix, classification_report  # Evaluation metrics for classification models
import gradio as gr  # Web interface library to deploy and test ML models
from PIL import Image  # For image file loading and basic image operations
```

```
testpath= C:\Users\M.Shiva Kumar\Downloads\E-Waste classification dataset (3).zip\modified-dataset'
trainpath=C:\Users\M.Shiva Kumar\Downloads\E-Waste classification dataset (3).zip\modified-dataset'
validpath =C:\Users\M.Shiva Kumar\Downloads\E-Waste classification dataset (3).zip\modified-dataset'
```

```
print(len(datatrain.class_names))
class_names = datatrain.class_names
print(class_names)
```

```
# Set the size of the entire figure (width=10, height=10 inches)
plt.figure(figsize=(10, 10))


# Take one batch from the dataset and iterate over the images and labels
for images, labels in datatrain.take(1):
  # Display the first 12 images from the batch
  for i in range(12):
      # Create a 4x3 grid of subplots and select the (i+1)th position
      ax = plt.subplot(4, 3, i + 1)


      # Display the image; convert the tensor to a NumPy array and ensure correct type
      plt.imshow(images[i].numpy().astype("uint8"))


      # Set the title of the subplot to the class name of the image
      plt.title(class_names[labels[i]])
```

```
# Remove axis ticks and labels for clarity
plt.axis("off")
```