Import necessarry library dependencies

In [3]:
```
import tensorflow as tf
from tensorflow.keras import layers, models
from tensorflow.keras.applications import EfficientNetV2B0
from tensorflow.keras.applications.efficientnet import preprocess_input
from sklearn.metrics import confusion_matrix, classification_report
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

Import datasets

In [4]:
```
#Dataset paths
trainpath = r'/content/drive/MyDrive/modified-dataset/train'
validpath = r'/content/drive/MyDrive/modified-dataset/val'
testpath  = r'/content/drive/MyDrive/modified-dataset/test'
```

Understand the Data

In [5]:
```
# 1EXPLORE AND UNDERSTAND THE DATA
IMG_SIZE = (128, 128)
BATCH_SIZE = 32

datatrain = tf.keras.utils.image_dataset_from_directory(trainpath, shuffle=True
datavalid = tf.keras.utils.image_dataset_from_directory(validpath, shuffle=True
datatest  = tf.keras.utils.image_dataset_from_directory(testpath, shuffle=False

class_names = datatrain.class_names
print(f"Classes: {class_names}")
```

```
Found 2410 files belonging to 10 classes.
Found 300 files belonging to 10 classes.
Found 310 files belonging to 10 classes.
Classes: ['Battery', 'Keyboard', 'Microwave', 'Mobile', 'Mouse', 'PCB', 'Player',
'Printer', 'Television', 'Washing Machine']
```

Visualize samples

In [6]:
```
# Visualize samples
plt.figure(figsize=(10,10))
for images, labels in datatrain.take(1):
    for i in range(9):
        ax = plt.subplot(3,3,i+1)
```

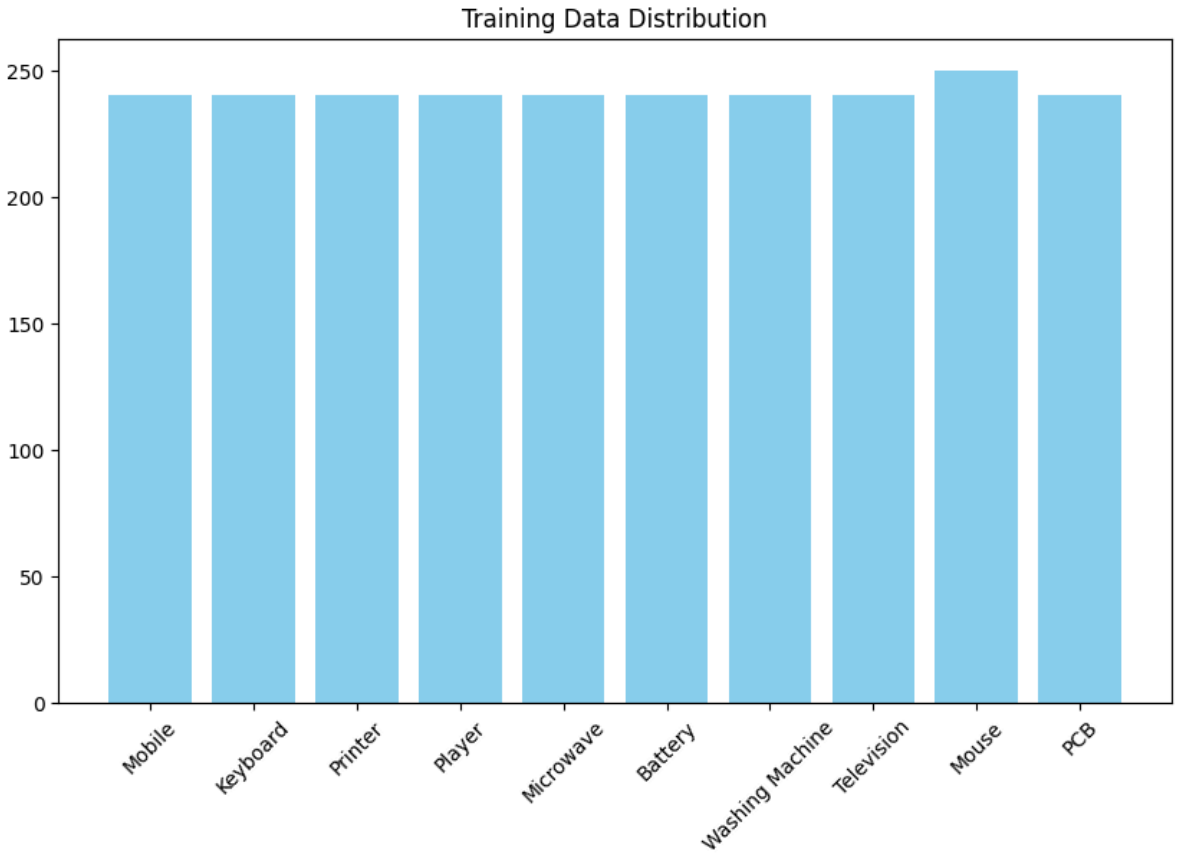
```
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
plt.show()
```

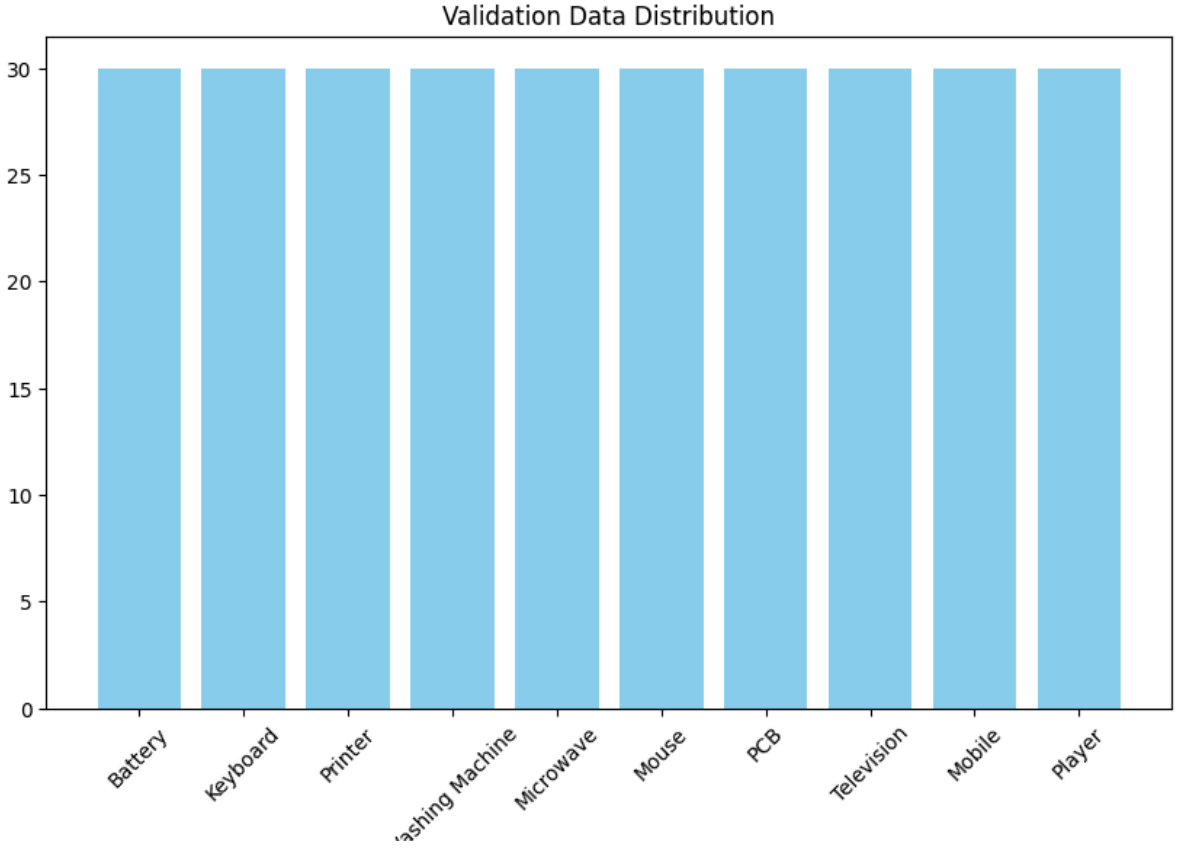
Plot class distribution

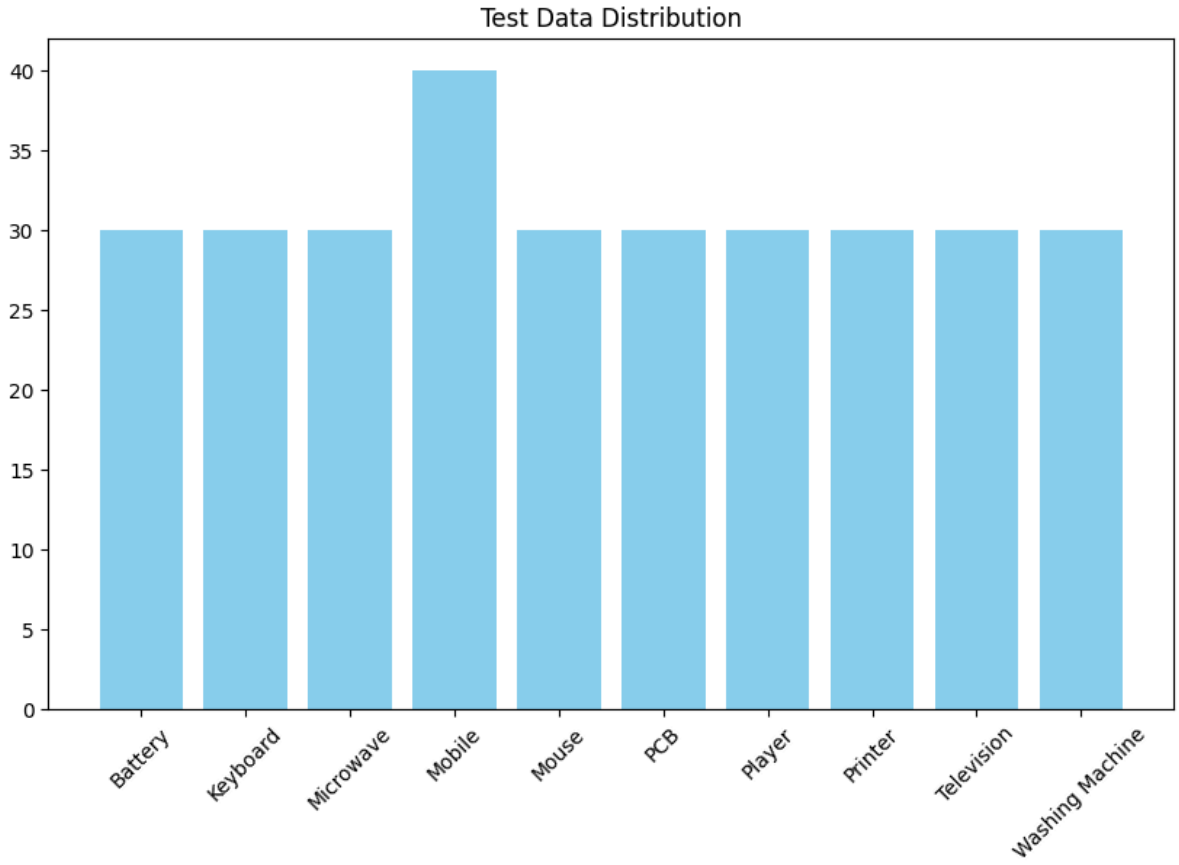
In [7]:

```
# Plot class distribution
def plot_class_distribution(dataset, title):
    counts = {}
    for _, labels in dataset:
        for label in labels.numpy():
            class_name = dataset.class_names[label]
            counts[class_name] = counts.get(class_name, 0) + 1
    plt.figure(figsize=(10,6))
    plt.bar(counts.keys(), counts.values(), color='skyblue')
    plt.title(title)
    plt.xticks(rotation=45)
```

```
    plt.show()

plot_class_distribution(datatrain, "Training Data Distribution")
plot_class_distribution(datavalid, "Validation Data Distribution")
plot_class_distribution(datatest, "Test Data Distribution")
```







DATA PREPROCESSING

In [8]:
```
# DATA PREPROCESSING / PREPARATION
data_augmentation = tf.keras.Sequential([
    layers.RandomFlip("horizontal"),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
])

datatrain = datatrain.prefetch(tf.data.AUTOTUNE)
dataValid = dataValid.prefetch(tf.data.AUTOTUNE)
datatest  = datatest.prefetch(tf.data.AUTOTUNE)
```

MODEL SELECTION USING EFFICIENTNET

In [9]:
```
# MODEL SELECTION
base_model = EfficientNetV2B0(input_shape=IMG_SIZE+(3,), include_top=False, wei
for layer in base_model.layers[:100]:
    layer.trainable = False

inputs = layers.Input(shape=IMG_SIZE+(3,))
x = data_augmentation(inputs)
x = preprocess_input(x)
x = base_model(x, training=False)
```

```
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dropout(0.2)(x)
outputs = layers.Dense(10, activation='softmax')(x)
model = models.Model(inputs, outputs)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-application
s/efficientnet_v2/efficientnetv2-b0_notop.h5
24274472/24274472 ━━━━━━━━━━━━━━━━━━━━ 0s 0us/step
```

MODEL TRAINING

In [10]:

```
# MODEL TRAINING
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
model.compile(optimizer=tf.keras.optimizers.Adam(1e-4),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

early_stop = EarlyStopping(monitor='val_loss', patience=3, restore_best_weights=
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=2)

history = model.fit(datatrain, validation_data=datavalid, epochs=15, callbacks=
```

```
Epoch 1/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 51s 183ms/step - accuracy: 0.2921 - loss: 2.0460 - val_
accuracy: 0.7967 - val_loss: 1.0489 - learning_rate: 1.0000e-04
Epoch 2/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 135ms/step - accuracy: 0.7852 - loss: 0.9561 - val_
accuracy: 0.8900 - val_loss: 0.4801 - learning_rate: 1.0000e-04
Epoch 3/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 21s 138ms/step - accuracy: 0.8765 - loss: 0.4865 - val_
accuracy: 0.9267 - val_loss: 0.2935 - learning_rate: 1.0000e-04
Epoch 4/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 20s 136ms/step - accuracy: 0.9105 - loss: 0.3234 - val_
accuracy: 0.9567 - val_loss: 0.2241 - learning_rate: 1.0000e-04
Epoch 5/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 20s 133ms/step - accuracy: 0.9268 - loss: 0.2303 - val_
accuracy: 0.9633 - val_loss: 0.1834 - learning_rate: 1.0000e-04
Epoch 6/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 9s 119ms/step - accuracy: 0.9395 - loss: 0.2093 - val_a
ccuracy: 0.9600 - val_loss: 0.1685 - learning_rate: 1.0000e-04
Epoch 7/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 125ms/step - accuracy: 0.9670 - loss: 0.1370 - val_
accuracy: 0.9700 - val_loss: 0.1556 - learning_rate: 1.0000e-04
Epoch 8/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 136ms/step - accuracy: 0.9681 - loss: 0.1139 - val_
accuracy: 0.9700 - val_loss: 0.1572 - learning_rate: 1.0000e-04
Epoch 9/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 20s 129ms/step - accuracy: 0.9606 - loss: 0.1249 - val_
accuracy: 0.9700 - val_loss: 0.1444 - learning_rate: 1.0000e-04
Epoch 10/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 9s 114ms/step - accuracy: 0.9818 - loss: 0.0712 - val_a
ccuracy: 0.9700 - val_loss: 0.1392 - learning_rate: 1.0000e-04
Epoch 11/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 134ms/step - accuracy: 0.9872 - loss: 0.0675 - val_
```

```
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 134ms/step - accuracy: 0.9872 - loss: 0.0673 - val_
accuracy: 0.9667 - val_loss: 0.1409 - learning_rate: 1.0000e-04
Epoch 12/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 21s 137ms/step - accuracy: 0.9892 - loss: 0.0500 - val_
accuracy: 0.9733 - val_loss: 0.1272 - learning_rate: 1.0000e-04
Epoch 13/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 9s 121ms/step - accuracy: 0.9865 - loss: 0.0515 - val_a
ccuracy: 0.9633 - val_loss: 0.1375 - learning_rate: 1.0000e-04
Epoch 14/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 145ms/step - accuracy: 0.9821 - loss: 0.0611 - val_
accuracy: 0.9600 - val_loss: 0.1519 - learning_rate: 1.0000e-04
Epoch 15/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 20s 135ms/step - accuracy: 0.9884 - loss: 0.0445 - val_
accuracy: 0.9700 - val_loss: 0.1400 - learning_rate: 5.0000e-05
```

MODEL TUNING AND OPTIMIZATION

In [11]:

```
# MODEL TUNING AND OPTIMIZATION (optional fine-tune)
for layer in base_model.layers[100:]:
    layer.trainable = True
model.compile(optimizer=tf.keras.optimizers.Adam(1e-5),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
model.fit(datatrain, validation_data=datavalid, epochs=5, callbacks=[early_stop
```

```
Epoch 1/5
76/76 ━━━━━━━━━━━━━━━━━━━━ 48s 179ms/step - accuracy: 0.9864 - loss: 0.0595 - val_
accuracy: 0.9700 - val_loss: 0.1295 - learning_rate: 1.0000e-05
Epoch 2/5
76/76 ━━━━━━━━━━━━━━━━━━━━ 18s 150ms/step - accuracy: 0.9882 - loss: 0.0549 - val_
accuracy: 0.9667 - val_loss: 0.1223 - learning_rate: 1.0000e-05
Epoch 3/5
76/76 ━━━━━━━━━━━━━━━━━━━━ 19s 133ms/step - accuracy: 0.9836 - loss: 0.0571 - val_
accuracy: 0.9700 - val_loss: 0.1313 - learning_rate: 1.0000e-05
Epoch 4/5
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 143ms/step - accuracy: 0.9875 - loss: 0.0465 - val_
accuracy: 0.9633 - val_loss: 0.1313 - learning_rate: 1.0000e-05
Epoch 5/5
76/76 ━━━━━━━━━━━━━━━━━━━━ 20s 140ms/step - accuracy: 0.9919 - loss: 0.0384 - val_
accuracy: 0.9667 - val_loss: 0.1310 - learning_rate: 5.0000e-06
```

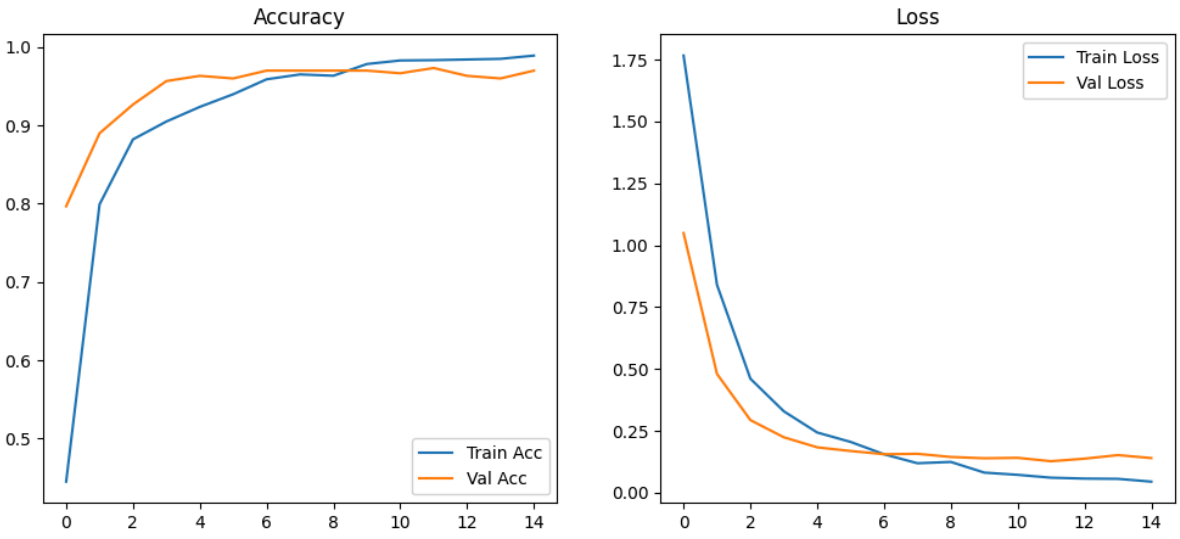
Out[11]: <keras.src.callbacks.history.History at 0x7e0c9268c550>

MODEL PERFORMANCE VISUALIZATION

In [12]:

```
# MODEL PERFORMANCE VISUALIZATION
plt.figure(figsize=(12,5))
plt.subplot(1,2,1)
plt.plot(history.history['accuracy'], label='Train Acc')
plt.plot(history.history['val_accuracy'], label='Val Acc')
plt.title('Accuracy')
plt.legend()
plt.subplot(1,2,2)
plt.plot(history.history['loss'], label='Train Loss')
```

```
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title('Loss')
plt.legend()
plt.show()
```



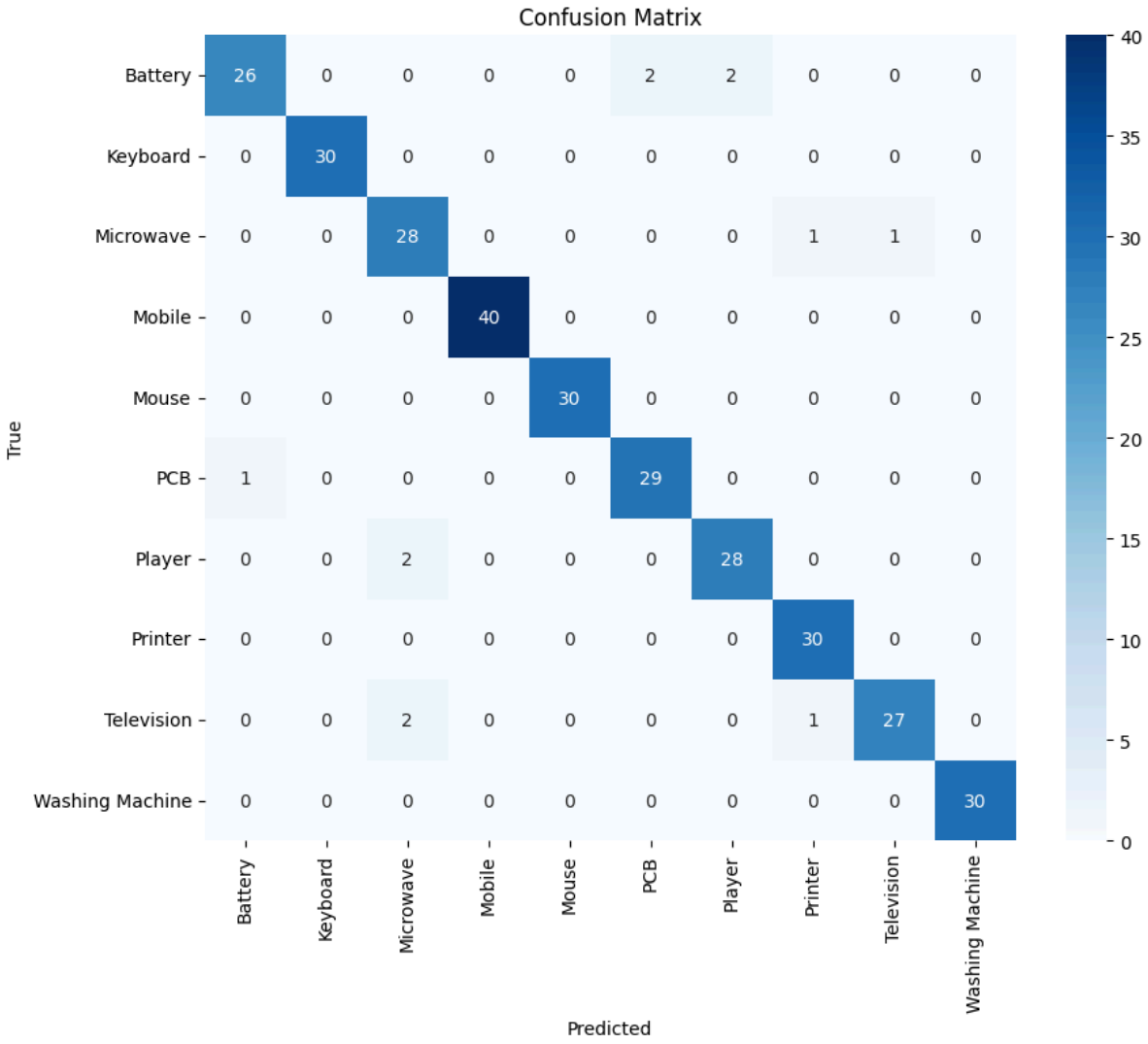
MODEL EVALUATION AND CONFUSION MATRIX

In [13]:

```
#  MODEL EVALUATION
test_loss, test_acc = model.evaluate(datatest)
print(f"Test Accuracy: {test_acc:.4f}, Test Loss: {test_loss:.4f}")

y_true = np.concatenate([y.numpy() for _, y in datatest], axis=0)
y_pred = np.argmax(model.predict(datatest), axis=1)

cm = confusion_matrix(y_true, y_pred)
print(classification_report(y_true, y_pred, target_names=class_names))

plt.figure(figsize=(10,8))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=class_names, yti
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")
plt.show()
```

```
10/10 ━━━━━━━━━━━━━━━━━━━━ 1s 135ms/step - accuracy: 0.9483 - loss: 0.1628
Test Accuracy: 0.9613, Test Loss: 0.1219
10/10 ━━━━━━━━━━━━━━━━━━━━ 7s 462ms/step
              precision    recall  f1-score   support

     Battery       0.96      0.87      0.91        30
    Keyboard       1.00      1.00      1.00        30
   Microwave       0.88      0.93      0.90        30
      Mobile       1.00      1.00      1.00        40
       Mouse       1.00      1.00      1.00        30
```

```
                    precision    recall  f1-score   support

             PCB       0.94      0.97      0.95        30
          Player       0.93      0.93      0.93        30
         Printer       0.94      1.00      0.97        30
      Television       0.96      0.90      0.93        30
Washing Machine        1.00      1.00      1.00        30

        accuracy                           0.96       310
       macro avg       0.96      0.96      0.96       310
    weighted avg       0.96      0.96      0.96       310
```



FINAL TESTING AND SAVE THE MODEL

In [14]:

```
# FINAL TESTING AND SAVE THE MODEL
model.save('efficientnetv2b0_ewaste_final.h5')
print("Keras model saved")
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `kera
s.saving.save_model(model)`. This file format is considered legacy. We recommend u
sing instead the native Keras format, e.g. `model.save('my_model.keras')` or `kera
s.saving.save_model(model, 'my_model.keras')`.
Keras model saved
```

Save TFLite

In [15]:
```
# Save TFLite
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_model = converter.convert()
with open('efficientnetv2b0_ewaste_final.tflite', 'wb') as f:
    f.write(tflite_model)
print("TFLite model saved")
```

```
Saved artifact at '/tmp/tmpz82u9cg5'. The following endpoints are available:

* Endpoint 'serve'
  args_0 (POSITIONAL_ONLY): TensorSpec(shape=(None, 128, 128, 3), dtype=tf.float3
2, name='keras_tensor_270')
Output Type:
  TensorSpec(shape=(None, 10), dtype=tf.float32, name=None)
Captures:
  138594976151120: TensorSpec(shape=(1, 1, 1, 3), dtype=tf.float32, name=None)
  138594976150928: TensorSpec(shape=(1, 1, 1, 3), dtype=tf.float32, name=None)
  138594976148240: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594976153232: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594976148816: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594976151504: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594976153424: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651242960: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651244112: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651245072: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651244304: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651243536: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594976148432: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651247184: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651246416: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651246608: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651246800: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651245648: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651249104: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651250064: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651249296: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651244496: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651251024: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651249488: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651251792: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651250256: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651248336: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651253328: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651252944: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651254288: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651253520: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651251408: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651253712: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651255632: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651256592: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651255824: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651254480: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651257552: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651257168: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594651258704: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
138594651257744: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651255440: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651258320: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651342032: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651341072: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651341840: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651340880: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651341264: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651343760: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651344720: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651343952: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651341456: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651344144: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651346064: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651347024: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651346256: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651344912: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651347984: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651347600: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651348944: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651348176: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651345872: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651349904: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651350096: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651348368: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651350480: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651346448: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651351632: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651352400: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651351248: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651351056: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651353360: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651352976: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651354320: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651353552: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651350288: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651355280: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651354896: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651356240: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651355472: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651345296: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651351824: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651353744: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651356816: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651522256: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651521488: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651521104: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651523600: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651522448: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651522640: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651523024: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651524944: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651525904: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651525136: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651523792: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651526864: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651526480: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651527824: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651527056: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651524752: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
138594651528784: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651528976: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651527248: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651529360: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651525328: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651530512: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651531280: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651530128: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651529936: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651530704: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651532624: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651533584: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651532816: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651531472: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651534544: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651534160: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651535504: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651534736: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651532432: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651536464: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651536080: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651537040: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651537232: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651536656: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575451152: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575450192: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594651533008: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575450384: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575452688: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575452304: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575453648: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575452880: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575450576: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575454608: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575454224: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575455568: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575454800: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575451536: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575456528: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575456720: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575454992: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575457104: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575453072: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575458256: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575459024: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575457872: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575457680: TensorSpec(shape=(), dtype=tf.resource, name=None)
138595886281552: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594976146896: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575450960: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594976147280: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594976147664: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575460752: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575460368: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575461712: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575460944: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575459600: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575462672: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575462864: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575461136: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575463248: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
138594575459216: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575464400: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575465168: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575464016: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575463824: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575456912: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838119056: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838118480: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575466320: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575464976: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838121552: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838121744: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838122320: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838123280: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838120592: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838119824: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838119440: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838119248: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838122704: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838121360: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838124240: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838125008: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838123472: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838121936: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838118672: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838126544: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838127504: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838126736: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838125968: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838128464: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838128080: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838129424: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838128656: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838126352: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838130384: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838130576: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838128848: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838130960: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838126928: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838132112: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838132880: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838131728: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838131536: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838132304: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838134416: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838134224: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838130768: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594838134032: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575762832: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575761488: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575763792: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575763024: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575762256: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575764752: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575764944: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575763216: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575765328: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575761872: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575766480: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575767248: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
138594575766096: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575765904: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575768208: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575767824: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575769168: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575768400: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575765136: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575770128: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575769744: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575771088: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575770320: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575761680: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575772048: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575772240: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575770512: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575772624: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575768592: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575773776: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575774544: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575773392: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575773200: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575773968: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575775888: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575776848: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575776080: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575774736: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575775696: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576024016: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576024592: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575775120: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594575776656: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576025552: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576025744: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576024400: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576026128: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576024208: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576027280: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576028048: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576026896: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576026704: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576027472: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576029392: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576030352: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576029584: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576028240: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576031312: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576030928: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576032272: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576031504: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576029200: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576033232: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576033424: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576031696: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576033808: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576029776: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576034960: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576035728: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576034576: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576034384: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576035152: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
138594576037072: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576038032: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576037264: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576035920: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576038992: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576039568: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576039376: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576038608: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576036880: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576039760: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576336272: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576335504: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576336464: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576334928: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576337616: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576338384: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576337232: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576337040: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576337808: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576339728: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576340688: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576339920: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576338576: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576341648: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576341264: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576342608: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576341840: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576339536: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576343568: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576343760: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576342032: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576344144: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576340112: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576345296: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576346064: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576344912: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576344720: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576345488: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576347408: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576348368: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576347600: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576346256: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576349328: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576348944: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576350288: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576349520: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576347216: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576346640: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576347792: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594576350864: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574599376: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574598992: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574598800: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574600720: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574599568: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574599760: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574600144: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574602064: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574603024: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574602256: TensorSpec(shape=(), dtype=tf.resource, name=None)
138594574600912: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
  138594574603984: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574603600: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574604944: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574604176: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574601872: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574605904: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574606096: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574604368: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574606480: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574602448: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574607632: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574608400: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574607248: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574607056: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574607824: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574609744: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574610704: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574609936: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574608592: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574608976: TensorSpec(shape=(), dtype=tf.resource, name=None)
  138594574613008: TensorSpec(shape=(), dtype=tf.resource, name=None)
TFLite model saved
```

Predictions on sample test images

In [16]:

```python
# Show predictions on sample test images
for images, labels in datatest.take(1):
    preds = model.predict(images)
    pred_classes = tf.argmax(preds, axis=1)
    for i in range(8):
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(f"True: {class_names[labels[i]]}, Pred: {class_names[pred_cla
        plt.axis("off")
        plt.show()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
```

True: Battery, Pred: Battery

True: Battery, Pred: Battery

True: Battery, Pred: Battery

True: Battery, Pred: Battery

True: Battery, Pred: Battery

True: Battery, Pred: PCB

True: Battery, Pred: Battery

Using CNN Model

In [17]:
```
# Normal CNN Model
normal_model = tf.keras.Sequential([
    tf.keras.layers.Rescaling(1./255, input_shape=IMG_SIZE+(3,)),  # Simple res
    tf.keras.layers.Conv2D(32, 3, activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(64, 3, activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(128, 3, activation='relu'),
    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

normal_model.compile(optimizer='adam',
                     loss='sparse_categorical_crossentropy',
                     metrics=['accuracy'])
normal_history = normal_model.fit(datatrain,
                                  validation_data=datavalid,
                                  epochs=15,
                                  callbacks=[early_stop, reduce_lr])
```

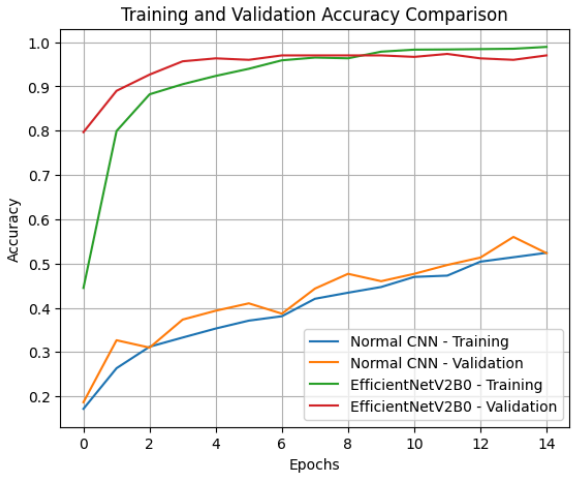
```
Epoch 1/15
/usr/local/lib/python3.11/dist-packages/keras/src/layers/preprocessing/tf_data_lay
er.py:19: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a laye
r. When using Sequential models, prefer using an `Input(shape)` object as the firs
t layer in the model instead.
  super().__init__(**kwargs)
76/76 ━━━━━━━━━━━━━━━━━━━━ 15s 139ms/step - accuracy: 0.1358 - loss: 2.2654 - val_
accuracy: 0.1867 - val_loss: 2.0728 - learning_rate: 0.0010
Epoch 2/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 7s 92ms/step - accuracy: 0.2451 - loss: 2.0675 - val_ac
curacy: 0.3267 - val_loss: 2.0186 - learning_rate: 0.0010
Epoch 3/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 93ms/step - accuracy: 0.3169 - loss: 1.9402 - val_a
ccuracy: 0.3100 - val_loss: 1.8574 - learning_rate: 0.0010
Epoch 4/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 96ms/step - accuracy: 0.3235 - loss: 1.8553 - val_a
ccuracy: 0.3733 - val_loss: 1.7331 - learning_rate: 0.0010
Epoch 5/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 6s 83ms/step - accuracy: 0.3490 - loss: 1.7900 - val_ac
curacy: 0.3933 - val_loss: 1.7310 - learning_rate: 0.0010
Epoch 6/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 7s 94ms/step - accuracy: 0.3539 - loss: 1.7752 - val_ac
curacy: 0.4100 - val_loss: 1.6510 - learning_rate: 0.0010
Epoch 7/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 93ms/step - accuracy: 0.3729 - loss: 1.7356 - val_a
ccuracy: 0.3867 - val_loss: 1.6837 - learning_rate: 0.0010
Epoch 8/15
```

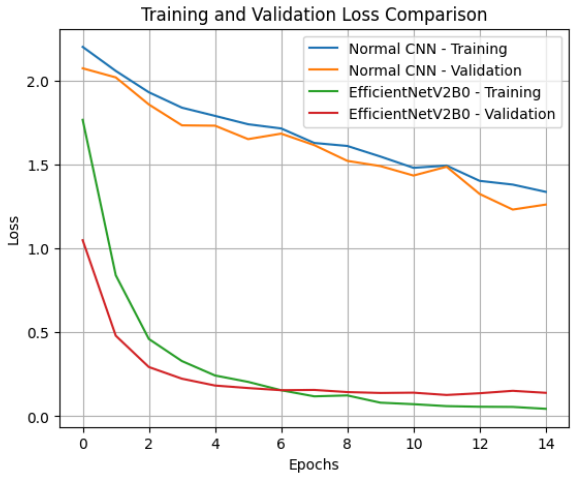
```
76/76 ━━━━━━━━━━━━━━━━━━━━ 7s 94ms/step - accuracy: 0.4131 - loss: 1.6530 - val_ac
curacy: 0.4433 - val_loss: 1.6153 - learning_rate: 0.0010
Epoch 9/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 7s 98ms/step - accuracy: 0.4366 - loss: 1.5812 - val_ac
curacy: 0.4767 - val_loss: 1.5214 - learning_rate: 0.0010
Epoch 10/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 7s 94ms/step - accuracy: 0.4337 - loss: 1.5523 - val_ac
curacy: 0.4600 - val_loss: 1.4903 - learning_rate: 0.0010
Epoch 11/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 7s 94ms/step - accuracy: 0.4535 - loss: 1.5012 - val_ac
curacy: 0.4767 - val_loss: 1.4341 - learning_rate: 0.0010
Epoch 12/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 87ms/step - accuracy: 0.4696 - loss: 1.4901 - val_a
ccuracy: 0.4967 - val_loss: 1.4864 - learning_rate: 0.0010
Epoch 13/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 91ms/step - accuracy: 0.4995 - loss: 1.4368 - val_a
ccuracy: 0.5133 - val_loss: 1.3244 - learning_rate: 0.0010
Epoch 14/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 8s 101ms/step - accuracy: 0.5181 - loss: 1.3933 - val_a
ccuracy: 0.5600 - val_loss: 1.2314 - learning_rate: 0.0010
Epoch 15/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 9s 88ms/step - accuracy: 0.5226 - loss: 1.3411 - val_ac
curacy: 0.5233 - val_loss: 1.2612 - learning_rate: 0.0010
```

Plot between CNN and EfficientNet

In [18]:

```python
import matplotlib.pyplot as plt

plt.figure(figsize=(14, 5))

# Accuracy Comparison
plt.subplot(1, 2, 1)
plt.plot(normal_history.history['accuracy'], label='Normal CNN - Training')
plt.plot(normal_history.history['val_accuracy'], label='Normal CNN - Validation
plt.plot(history.history['accuracy'], label='EfficientNetV2B0 - Training')
plt.plot(history.history['val_accuracy'], label='EfficientNetV2B0 - Validation'
plt.title('Training and Validation Accuracy Comparison')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)

# Loss Comparison
plt.subplot(1, 2, 2)
plt.plot(normal_history.history['loss'], label='Normal CNN - Training')
plt.plot(normal_history.history['val_loss'], label='Normal CNN - Validation')
plt.plot(history.history['loss'], label='EfficientNetV2B0 - Training')
plt.plot(history.history['val_loss'], label='EfficientNetV2B0 - Validation')
plt.title('Training and Validation Loss Comparison')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)

plt.show()
```





In [20]:
```
!pip install gradio tensorflow pillow
```

```
Requirement already satisfied: gradio in /usr/local/lib/python3.11/dist-packages
(5.31.0)
Requirement already satisfied: tensorflow in /usr/local/lib/python3.11/dist-packag
es (2.18.0)
Requirement already satisfied: pillow in /usr/local/lib/python3.11/dist-packages
(11.2.1)
Requirement already satisfied: aiofiles<25.0,>=22.0 in /usr/local/lib/python3.11/d
ist-packages (from gradio) (24.1.0)
Requirement already satisfied: anyio<5.0,>=3.0 in /usr/local/lib/python3.11/dist-p
ackages (from gradio) (4.9.0)
Requirement already satisfied: fastapi<1.0,>=0.115.2 in /usr/local/lib/python3.11/
dist-packages (from gradio) (0.115.12)
Requirement already satisfied: ffmpy in /usr/local/lib/python3.11/dist-packages (f
rom gradio) (0.6.0)
Requirement already satisfied: gradio-client==1.10.1 in /usr/local/lib/python3.11/
dist-packages (from gradio) (1.10.1)
Requirement already satisfied: groovy~=0.1 in /usr/local/lib/python3.11/dist-packa
ges (from gradio) (0.1.2)
Requirement already satisfied: httpx>=0.24.1 in /usr/local/lib/python3.11/dist-pac
kages (from gradio) (0.28.1)
Requirement already satisfied: huggingface-hub>=0.28.1 in /usr/local/lib/python3.1
1/dist-packages (from gradio) (0.33.0)
Requirement already satisfied: jinja2<4.0 in /usr/local/lib/python3.11/dist-packag
es (from gradio) (3.1.6)
Requirement already satisfied: markupsafe<4.0,>=2.0 in /usr/local/lib/python3.11/d
ist-packages (from gradio) (3.0.2)
Requirement already satisfied: numpy<3.0,>=1.0 in /usr/local/lib/python3.11/dist-p
ackages (from gradio) (2.0.2)
Requirement already satisfied: orjson~=3.0 in /usr/local/lib/python3.11/dist-packa
ges (from gradio) (3.10.18)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-package
s (from gradio) (24.2)
Requirement already satisfied: pandas<3.0,>=1.0 in /usr/local/lib/python3.11/dist-
packages (from gradio) (2.2.2)
Requirement already satisfied: pydantic<2.12,>=2.0 in /usr/local/lib/python3.11/di
st-packages (from gradio) (2.11.7)
Requirement already satisfied: pydub in /usr/local/lib/python3.11/dist-packages (f
rom gradio) (0.25.1)
Requirement already satisfied: python-multipart>=0.0.18 in /usr/local/lib/python3.
11/dist-packages (from gradio) (0.0.20)
```

```
Requirement already satisfied: pyyaml<7.0,>=5.0 in /usr/local/lib/python3.11/dist-
packages (from gradio) (6.0.2)
Requirement already satisfied: ruff>=0.9.3 in /usr/local/lib/python3.11/dist-packa
ges (from gradio) (0.11.13)
Requirement already satisfied: safehttpx<0.2.0,>=0.1.6 in /usr/local/lib/python3.1
1/dist-packages (from gradio) (0.1.6)
Requirement already satisfied: semantic-version~=2.0 in /usr/local/lib/python3.11/
dist-packages (from gradio) (2.10.0)
Requirement already satisfied: starlette<1.0,>=0.40.0 in /usr/local/lib/python3.1
1/dist-packages (from gradio) (0.46.2)
Requirement already satisfied: tomlkit<0.14.0,>=0.12.0 in /usr/local/lib/python3.1
1/dist-packages (from gradio) (0.13.3)
Requirement already satisfied: typer<1.0,>=0.12 in /usr/local/lib/python3.11/dist-
packages (from gradio) (0.16.0)
Requirement already satisfied: typing-extensions~=4.0 in /usr/local/lib/python3.1
1/dist-packages (from gradio) (4.14.0)
Requirement already satisfied: uvicorn>=0.14.0 in /usr/local/lib/python3.11/dist-p
ackages (from gradio) (0.34.3)
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages
(from gradio-client==1.10.1->gradio) (2025.3.2)
Requirement already satisfied: websockets<16.0,>=10.0 in /usr/local/lib/python3.1
1/dist-packages (from gradio-client==1.10.1->gradio) (15.0.1)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.11/dist-pa
ckages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.11/dist
-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=24.3.25 in /usr/local/lib/python3.11/d
ist-packages (from tensorflow) (25.2.10)
Requirement already satisfied: gast!=0.5.0,!=0.5.1,!=0.5.2,>=0.2.1 in /usr/local/l
ib/python3.11/dist-packages (from tensorflow) (0.6.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.11/di
st-packages (from tensorflow) (0.2.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.11/dist-
packages (from tensorflow) (18.1.1)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.11/dist
-packages (from tensorflow) (3.4.0)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.
4,!=4.21.5,<6.0.0dev,>=3.20.3 in /usr/local/lib/python3.11/dist-packages (from ten
sorflow) (5.29.5)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.11/di
st-packages (from tensorflow) (2.32.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packag
es (from tensorflow) (75.2.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.11/dist-packa
ges (from tensorflow) (1.17.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.11/dist-
packages (from tensorflow) (3.1.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.11/dist-pac
kages (from tensorflow) (1.17.2)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.11/di
st-packages (from tensorflow) (1.73.0)
Requirement already satisfied: tensorboard<2.19,>=2.18 in /usr/local/lib/python3.1
1/dist-packages (from tensorflow) (2.18.0)
Requirement already satisfied: keras>=3.5.0 in /usr/local/lib/python3.11/dist-pack
ages (from tensorflow) (3.8.0)
Requirement already satisfied: h5py>=3.11.0 in /usr/local/lib/python3.11/dist-pack
ages (from tensorflow) (3.14.0)
Requirement already satisfied: ml-dtypes<0.5.0,>=0.4.0 in /usr/local/lib/python3.1
1/dist-packages (from tensorflow) (0.4.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/
lib/python3.11/dist-packages (from tensorflow) (0.37.1)
```

```
lib/python3.11/dist-packages (from tensorflow) (0.37.1)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-package
s (from anyio<5.0,>=3.0->gradio) (3.10)
Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-pack
ages (from anyio<5.0,>=3.0->gradio) (1.3.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.11/dis
t-packages (from astunparse>=1.6.0->tensorflow) (0.45.1)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages
(from httpx>=0.24.1->gradio) (2025.6.15)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-pac
kages (from httpx>=0.24.1->gradio) (1.0.9)
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.11/dist-package
s (from httpcore==1.*->httpx>=0.24.1->gradio) (0.16.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages
(from huggingface-hub>=0.28.1->gradio) (3.18.0)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-pack
ages (from huggingface-hub>=0.28.1->gradio) (4.67.1)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.2 in /usr/local/lib/python3.11/d
ist-packages (from huggingface-hub>=0.28.1->gradio) (1.1.3)
Requirement already satisfied: rich in /usr/local/lib/python3.11/dist-packages (fr
om keras>=3.5.0->tensorflow) (13.9.4)
Requirement already satisfied: namex in /usr/local/lib/python3.11/dist-packages (f
rom keras>=3.5.0->tensorflow) (0.1.0)
Requirement already satisfied: optree in /usr/local/lib/python3.11/dist-packages
(from keras>=3.5.0->tensorflow) (0.16.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.1
1/dist-packages (from pandas<3.0,>=1.0->gradio) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-pack
ages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-pa
ckages (from pandas<3.0,>=1.0->gradio) (2025.2)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.1
1/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in /usr/local/lib/python3.11/
dist-packages (from pydantic<2.12,>=2.0->gradio) (2.33.2)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.
11/dist-packages (from pydantic<2.12,>=2.0->gradio) (0.4.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.
11/dist-packages (from requests<3,>=2.21.0->tensorflow) (3.4.2)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dis
t-packages (from requests<3,>=2.21.0->tensorflow) (2.4.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.11/dist-p
ackages (from tensorboard<2.19,>=2.18->tensorflow) (3.8)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/loca
l/lib/python3.11/dist-packages (from tensorboard<2.19,>=2.18->tensorflow) (0.7.2)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.11/dist-p
ackages (from tensorboard<2.19,>=2.18->tensorflow) (3.1.3)
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.11/dist-pack
ages (from typer<1.0,>=0.12->gradio) (8.2.1)
Requirement already satisfied: shellingham>=1.3.0 in /usr/local/lib/python3.11/dis
t-packages (from typer<1.0,>=0.12->gradio) (1.5.4)
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.11/
dist-packages (from rich->keras>=3.5.0->tensorflow) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.1
1/dist-packages (from rich->keras>=3.5.0->tensorflow) (2.19.1)
Requirement already satisfied: mdurl~=0.1 in /usr/local/lib/python3.11/dist-packag
es (from markdown-it-py>=2.2.0->rich->keras>=3.5.0->tensorflow) (0.1.2)
```

In [22]:

```
import gradio as gr
import tensorflow as tf
import numpy as np
```

```
import numpy as np
from PIL import Image
from tensorflow.keras.applications.efficientnet import preprocess_input

#  Load your saved Keras model
model = tf.keras.models.load_model('efficientnetv2b0_ewaste_final.h5')
class_names = [
    'Battery', 'Keyboard', 'Microwave', 'Mobile',
    'Mouse', 'PCB', 'Player', 'Printer',
    'Television', 'Washing Machine'
]

#  Prediction function
def classify_image(img):
    """Preprocess input image and make prediction."""
    img = img.resize((128, 128))
    img_array = np.asarray(img, dtype=np.float32)
    img_array = preprocess_input(img_array)
    img_array = np.expand_dims(img_array, axis=0)

    prediction = model.predict(img_array)
    index = np.argmax(prediction)
    class_name = class_names[index]
    confidence = prediction[0][index]

    return f"Predicted: {class_name} (Confidence: {confidence:.2f})"

#  Create Gradio Interface
iface = gr.Interface(
    fn=classify_image,
    inputs=gr.Image(type="pil"),
    outputs="text",
    title="E-Waste Classifier",
    description="Upload an image of an e-waste item for classification."
)

#  Launch the app
iface.launch(share=True)  # Enables a public link
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be bu
ilt. `model.compile_metrics` will be empty until you train or evaluate the model.
Colab notebook detected. To show errors in colab notebook, set debug=True in launc
h()
* Running on public URL: https://80de40e225bc483c3d.gradio.live

This share link expires in 1 week. For free permanent hosting and GPU upgrades, ru
n `gradio deploy` from the terminal in the working directory to deploy to Hugging
Face Spaces (https://huggingface.co/spaces)
```

Out[22]: