Import necessarry library dependencies

In [1]:

```
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 [2]:

```
#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 [3]:

```
# 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 [4]:

```
# 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)
```

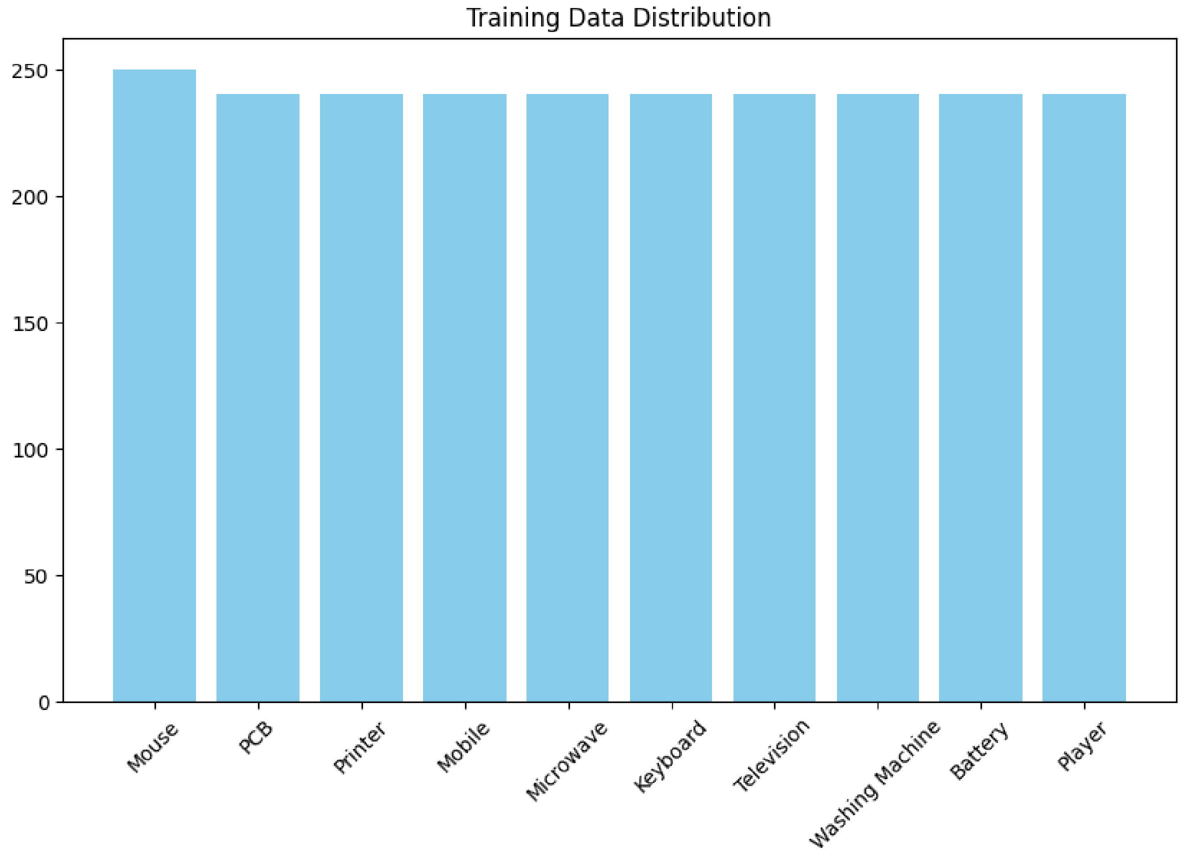
```
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
plt.show()
```

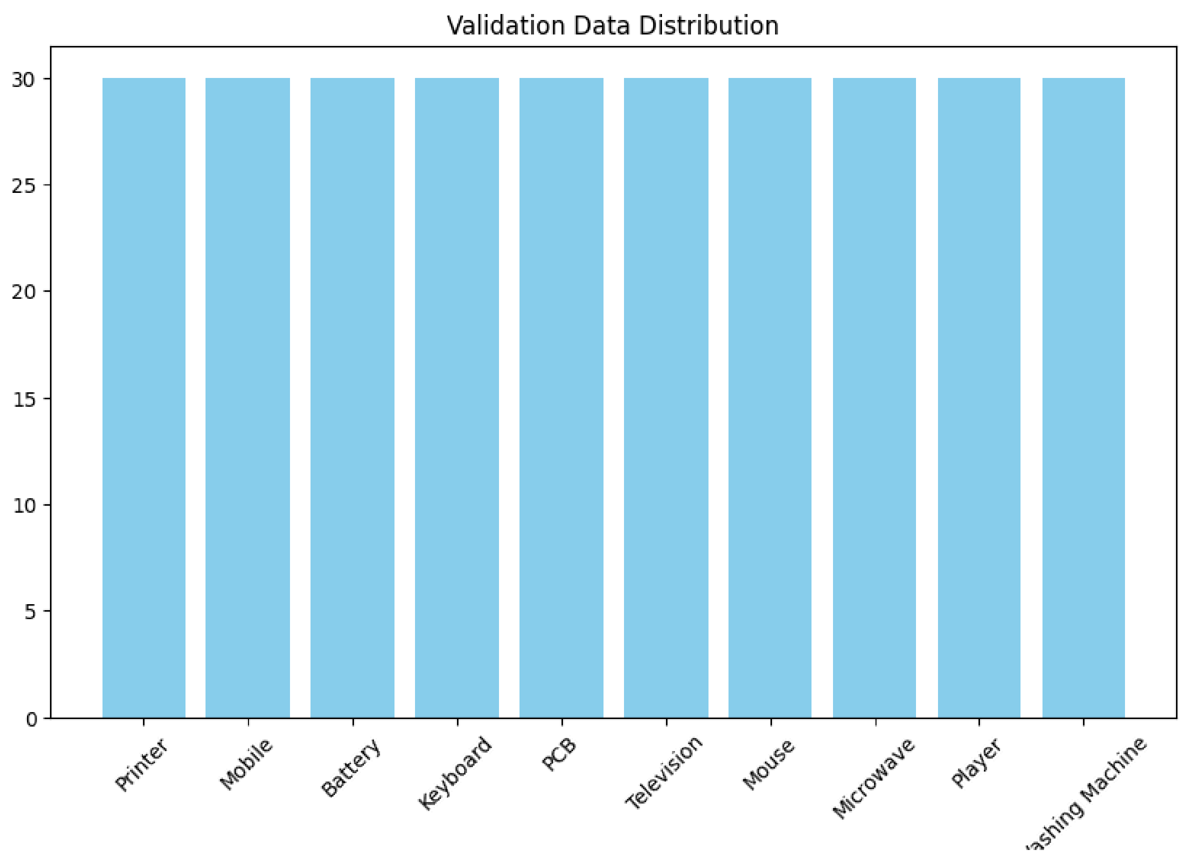
Plot class distribution

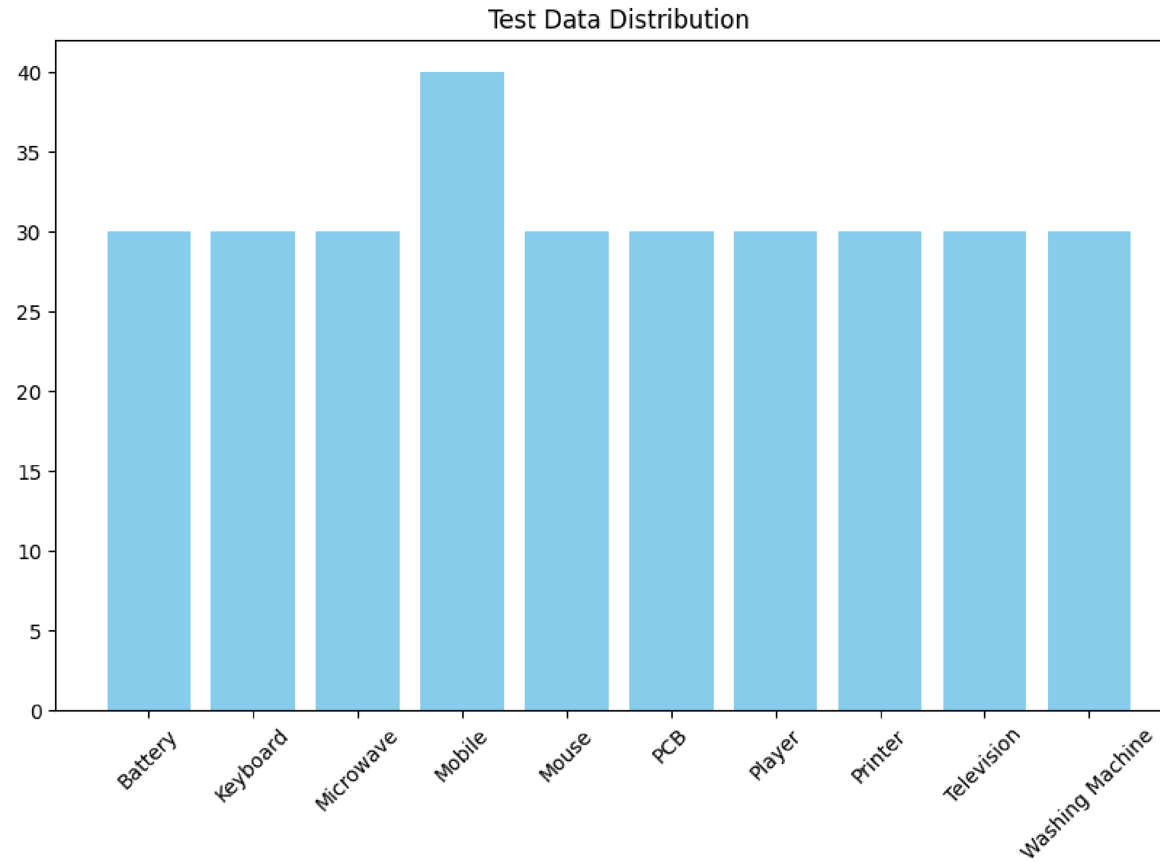
In [5]:

```
# 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 [6]:

```
# 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 [7]:

```
# 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 [9]:

```
# 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 ━━━━━━━━━━━━━━━━━━━━ 52s 180ms/step - accuracy: 0.2598 - loss: 2.1305 - val_
accuracy: 0.7967 - val_loss: 1.1384 - learning_rate: 1.0000e-04
Epoch 2/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 137ms/step - accuracy: 0.7977 - loss: 0.9845 - val_
accuracy: 0.9033 - val_loss: 0.5048 - learning_rate: 1.0000e-04
Epoch 3/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 20s 123ms/step - accuracy: 0.8681 - loss: 0.4941 - val_
accuracy: 0.9333 - val_loss: 0.3257 - learning_rate: 1.0000e-04
Epoch 4/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 140ms/step - accuracy: 0.9202 - loss: 0.3001 - val_
accuracy: 0.9300 - val_loss: 0.2432 - learning_rate: 1.0000e-04
Epoch 5/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 20s 130ms/step - accuracy: 0.9412 - loss: 0.2134 - val_
accuracy: 0.9500 - val_loss: 0.1960 - learning_rate: 1.0000e-04
Epoch 6/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 125ms/step - accuracy: 0.9487 - loss: 0.1866 - val_
accuracy: 0.9433 - val_loss: 0.1804 - learning_rate: 1.0000e-04
Epoch 7/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 123ms/step - accuracy: 0.9540 - loss: 0.1712 - val_
accuracy: 0.9533 - val_loss: 0.1624 - learning_rate: 1.0000e-04
Epoch 8/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 142ms/step - accuracy: 0.9671 - loss: 0.1273 - val_
accuracy: 0.9500 - val_loss: 0.1550 - learning_rate: 1.0000e-04
Epoch 9/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 19s 123ms/step - accuracy: 0.9703 - loss: 0.1106 - val_
accuracy: 0.9500 - val_loss: 0.1546 - learning_rate: 1.0000e-04
Epoch 10/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 9s 121ms/step - accuracy: 0.9827 - loss: 0.0785 - val_a
ccuracy: 0.9567 - val_loss: 0.1511 - learning_rate: 1.0000e-04
Epoch 11/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 134ms/step - accuracy: 0.9785 - loss: 0.0719 - val_
```

```
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 134ms/step - accuracy: 0.9783 - loss: 0.0719 - val_
accuracy: 0.9600 - val_loss: 0.1248 - learning_rate: 1.0000e-04
Epoch 12/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 133ms/step - accuracy: 0.9772 - loss: 0.0771 - val_
accuracy: 0.9567 - val_loss: 0.1306 - learning_rate: 1.0000e-04
Epoch 13/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 131ms/step - accuracy: 0.9863 - loss: 0.0611 - val_
accuracy: 0.9633 - val_loss: 0.1198 - learning_rate: 1.0000e-04
Epoch 14/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 123ms/step - accuracy: 0.9917 - loss: 0.0386 - val_
accuracy: 0.9600 - val_loss: 0.1251 - learning_rate: 1.0000e-04
Epoch 15/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 142ms/step - accuracy: 0.9923 - loss: 0.0323 - val_
accuracy: 0.9700 - val_loss: 0.1191 - learning_rate: 1.0000e-04
```

MODEL TUNING AND OPTIMIZATION

In [10]:

```
# 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 ━━━━━━━━━━━━━━━━━━━━ 47s 181ms/step - accuracy: 0.9923 - loss: 0.0303 - val_
accuracy: 0.9633 - val_loss: 0.1201 - learning_rate: 1.0000e-05
Epoch 2/5
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 134ms/step - accuracy: 0.9875 - loss: 0.0374 - val_
accuracy: 0.9633 - val_loss: 0.1144 - learning_rate: 1.0000e-05
Epoch 3/5
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 138ms/step - accuracy: 0.9941 - loss: 0.0353 - val_
accuracy: 0.9633 - val_loss: 0.1188 - learning_rate: 1.0000e-05
Epoch 4/5
76/76 ━━━━━━━━━━━━━━━━━━━━ 19s 121ms/step - accuracy: 0.9895 - loss: 0.0435 - val_
accuracy: 0.9633 - val_loss: 0.1256 - learning_rate: 1.0000e-05
Epoch 5/5
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 135ms/step - accuracy: 0.9950 - loss: 0.0286 - val_
accuracy: 0.9700 - val_loss: 0.1140 - learning_rate: 5.0000e-06
```

Out[10]:

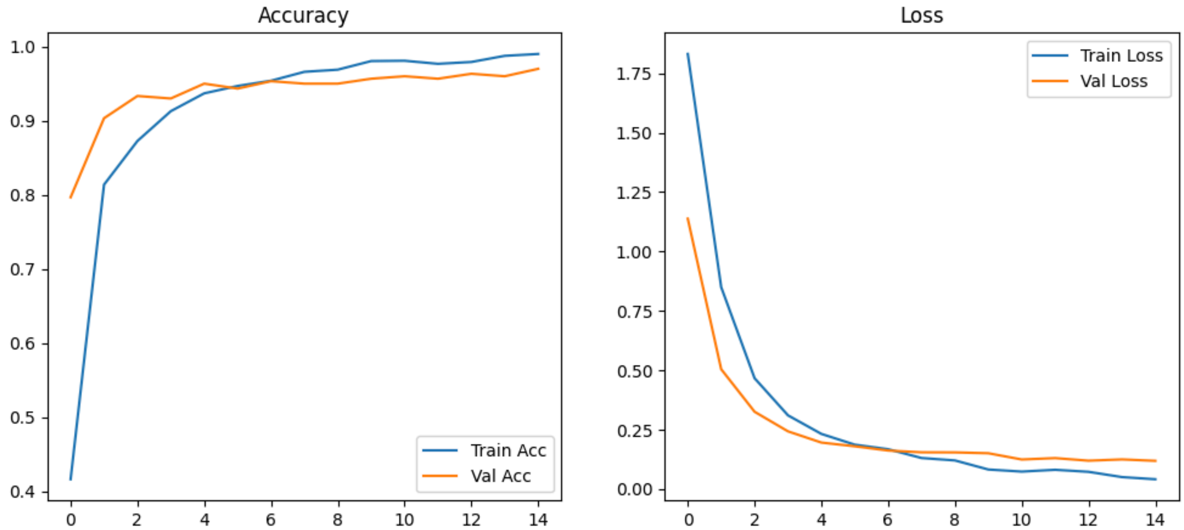
```
<keras.src.callbacks.history.History at 0x7d08ecd52850>
```

MODEL PERFORMANCE VISUALIZATION

In [11]:

```
# 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()
```



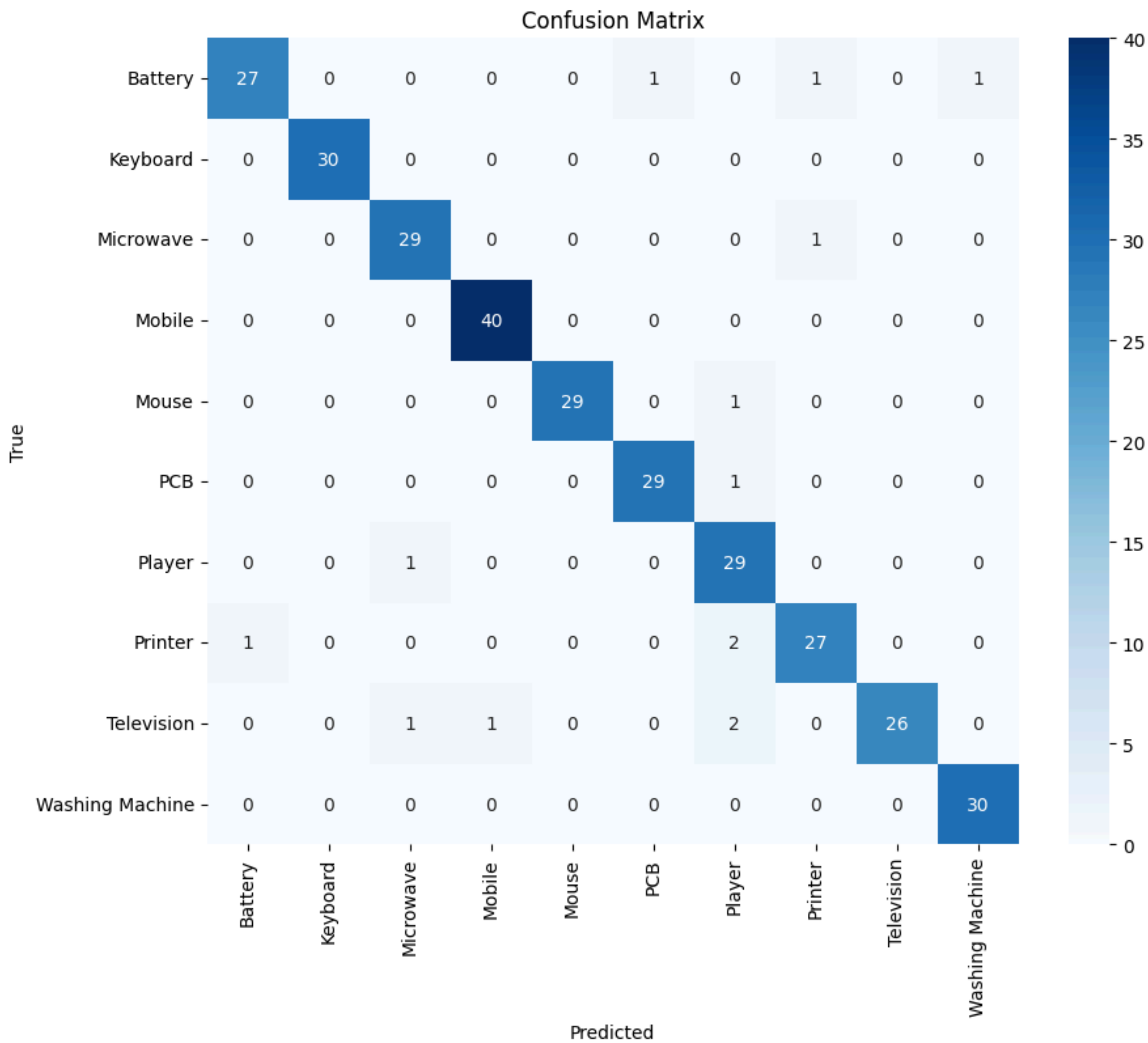
MODEL EVALUATION AND CONFUSION MATRIX

In [12]:

```
#  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, ytic
plt.xlabel("Predicted")
plt.ylabel("True")
plt.title("Confusion Matrix")
plt.show()
```

```
10/10 ━━━━━━━━━━━━━━━━━━━━ 1s 128ms/step - accuracy: 0.9555 - loss: 0.1602
✅ Test Accuracy: 0.9548, Test Loss: 0.1317
10/10 ━━━━━━━━━━━━━━━━━━━━ 7s 393ms/step
              precision    recall  f1-score   support

     Battery       0.96      0.90      0.93        30
    Keyboard       1.00      1.00      1.00        30
   Microwave       0.94      0.97      0.95        30
      Mobile       0.98      1.00      0.99        40
       Mouse       1.00      0.97      0.98        30
```

```
          Mouse       1.00      0.97      0.98        30
            PCB       0.97      0.97      0.97        30
         Player       0.83      0.97      0.89        30
        Printer       0.93      0.90      0.92        30
     Television       1.00      0.87      0.93        30
Washing Machine       0.97      1.00      0.98        30

       accuracy                           0.95       310
      macro avg       0.96      0.95      0.95       310
   weighted avg       0.96      0.95      0.95       310
```



FINAL TESTING AND SAVE THE MODEL

In [18]:

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

```
Keras model saved
```

Save TFLite

In [19]:

```python
# 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/tmpodr7su2v'. 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:
  137479627293328: TensorSpec(shape=(1, 1, 1, 3), dtype=tf.float32, name=None)
  137479627293136: TensorSpec(shape=(1, 1, 1, 3), dtype=tf.float32, name=None)
  137480611166160: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627293712: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627296208: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627295824: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627295440: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627296016: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627297360: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627298320: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627297552: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627296784: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627297744: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627301008: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627300048: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627300432: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627300240: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627298128: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627302928: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627303888: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627303120: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627299472: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627304848: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627303312: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627305616: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627304080: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627302160: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627307152: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627306768: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627308112: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627307344: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627305232: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627304464: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627620816: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627621776: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627308304: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627620624: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627622736: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627622352: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627623696: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627622928: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627621584: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627621008: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479627624848: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
137479627625808: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627625040: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627623504: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627624272: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627626960: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627627920: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627627152: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627625616: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627627344: TensorSpec(shape=(), dtype=tf.resource, name=None)
137480611162704: TensorSpec(shape=(), dtype=tf.resource, name=None)
137480611164432: TensorSpec(shape=(), dtype=tf.resource, name=None)
137480611166352: TensorSpec(shape=(), dtype=tf.resource, name=None)
137480611164816: TensorSpec(shape=(), dtype=tf.resource, name=None)
137480611165008: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627629456: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627630224: TensorSpec(shape=(), dtype=tf.resource, name=None)
137480611165776: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627626384: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627628112: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627631184: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627629648: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627631568: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627630416: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627632720: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627633488: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627632336: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627632144: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627634448: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627634064: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627635408: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627634640: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627631376: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627636368: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627635984: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628030032: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627634832: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479627632912: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628031568: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628031760: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628031184: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628032144: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628030992: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628033296: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628034064: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628032912: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628032720: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628033488: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628035408: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628036368: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628035600: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628034256: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628037328: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628036944: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628038288: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628037520: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628035216: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628039248: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628039440: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628037712: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628039824: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628035792: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
137479628035792: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628040976: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628041744: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628040592: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628040400: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628041168: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628043088: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628044048: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628043280: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628041936: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628045008: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628044624: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628046160: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628045200: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628042896: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628045392: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479628042320: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625867536: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625868496: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625867728: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625869648: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625870416: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625869264: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625869072: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625871376: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625870992: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625872336: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625871568: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625868304: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625873296: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625872912: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625874256: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625873488: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625868880: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625875216: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625875408: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625873680: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625875792: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625871760: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625876944: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625877712: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625876560: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625876368: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625877136: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625879056: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625880016: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625879248: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625877904: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625880976: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625880592: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625881936: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625881168: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625878864: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625882896: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625882128: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625882512: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625878288: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625879440: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626081296: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626080336: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626081488: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
137479626080528: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626083984: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626085136: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626084368: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626084752: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626084560: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626083216: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626083408: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626086480: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626085712: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626084176: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626087440: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626087632: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626085904: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626088016: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626083600: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626089168: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626089936: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626088784: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626088592: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626085520: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626091472: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626092432: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626091664: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626090896: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626093392: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626093008: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626094352: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626093584: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626091280: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626095312: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626095504: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626093776: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626094928: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626095888: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626392208: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626392784: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626091088: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626096272: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626391824: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626394128: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626395088: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626394320: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626392976: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626396048: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626395664: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626397008: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626396240: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626393936: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626397968: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626398160: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626396432: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626398544: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626394512: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626399696: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626400464: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626399312: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626399120: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626401424: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626401040: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
137479626402384: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626401616: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626398352: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626403344: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626402960: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626404304: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626403536: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626393360: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626405264: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626405456: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626403728: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626405840: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626401808: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626399888: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626405648: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626406608: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479626406416: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624687696: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624688848: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624689808: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624689040: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624688080: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624690768: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624690384: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624691728: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624690960: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624688656: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624692688: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624692880: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624691152: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624693264: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624689232: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624694416: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624695184: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624694032: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624693840: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624694608: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624696528: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624697488: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624696720: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624695376: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624698448: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624698064: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624699408: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624698640: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624696336: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624700368: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624700560: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624698832: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624700944: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624696912: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624702096: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624702864: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624701712: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624701520: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624702288: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625097872: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625098448: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624703440: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479624700752: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625099408: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
137479625099408: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625099024: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625100368: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625099600: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625097296: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625101328: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625101520: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625099792: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625101904: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625097488: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625103056: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625103824: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625102672: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625102480: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625103248: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625105168: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625106128: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625105360: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625104016: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625107088: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625106704: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625108048: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625107280: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625104976: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625109008: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625109200: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625107472: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625109584: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625105552: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625110736: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625111504: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625110352: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625110160: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625110928: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625112848: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625113232: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625109392: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625112656: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625572432: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625573008: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625574544: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625573776: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625573584: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625575504: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625575696: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625573968: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625576080: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625573200: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625577232: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625578000: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625576848: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625576656: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625577424: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625579344: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625580304: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625579536: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625578192: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625581264: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625580880: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625582224: TensorSpec(shape=(), dtype=tf.resource, name=None)
137479625581456: TensorSpec(shape=(), dtype=tf.resource, name=None)
```

```
  137479625579152: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625583184: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625583376: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625581648: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625583760: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625579728: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625584912: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625585680: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625584528: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625584336: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625585104: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625587024: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625587984: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625587216: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625585872: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479625586832: TensorSpec(shape=(), dtype=tf.resource, name=None)
  137479624263248: TensorSpec(shape=(), dtype=tf.resource, name=None)
TFLite model saved
```

Predictions on sample test images

In [15]:

```
# 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()
```

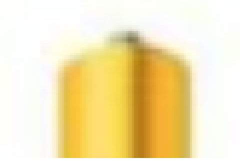
```
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
```

True: Battery, Pred: Battery

True: Battery, Pred: Printer

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 [16]:

```
# Normal CNN Model
normal_model = tf.keras.Sequential([
    tf.keras.layers.Rescaling(1./255, input_shape=IMG_SIZE+(3,)),  # Simple resc
    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])
```

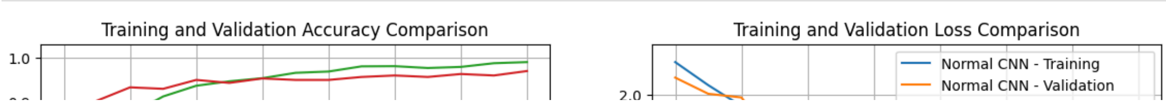
```
Epoch 1/15
/usr/local/lib/python3.11/dist-packages/keras/src/layers/preprocessing/tf_data_lay
er.py:19: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a laye
r. When using Sequential models, prefer using an `Input(shape)` object as the firs
t layer in the model instead.
  super().__init__(**kwargs)
76/76 ━━━━━━━━━━━━━━━━━━━━ 13s 126ms/step - accuracy: 0.1479 - loss: 2.2522 - val_
accuracy: 0.2133 - val_loss: 2.1031 - learning_rate: 0.0010
Epoch 2/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 6s 79ms/step - accuracy: 0.2411 - loss: 2.0616 - val_ac
curacy: 0.2833 - val_loss: 2.0063 - learning_rate: 0.0010
Epoch 3/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 87ms/step - accuracy: 0.3085 - loss: 1.9544 - val_a
ccuracy: 0.2500 - val_loss: 1.9843 - learning_rate: 0.0010
Epoch 4/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 84ms/step - accuracy: 0.3229 - loss: 1.8475 - val_a
ccuracy: 0.3400 - val_loss: 1.7361 - learning_rate: 0.0010
Epoch 5/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 96ms/step - accuracy: 0.3549 - loss: 1.7743 - val_a
ccuracy: 0.4300 - val_loss: 1.6915 - learning_rate: 0.0010
Epoch 6/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 6s 79ms/step - accuracy: 0.3797 - loss: 1.7258 - val_ac
curacy: 0.4267 - val_loss: 1.6392 - learning_rate: 0.0010
Epoch 7/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 7s 96ms/step - accuracy: 0.4125 - loss: 1.6638 - val_ac
curacy: 0.4700 - val_loss: 1.6006 - learning_rate: 0.0010
Epoch 8/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 9s 82ms/step - accuracy: 0.4091 - loss: 1.6414 - val_ac
curacy: 0.4433 - val_loss: 1.6227 - learning_rate: 0.0010
Epoch 9/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 91ms/step - accuracy: 0.4209 - loss: 1.6662 - val_a
```

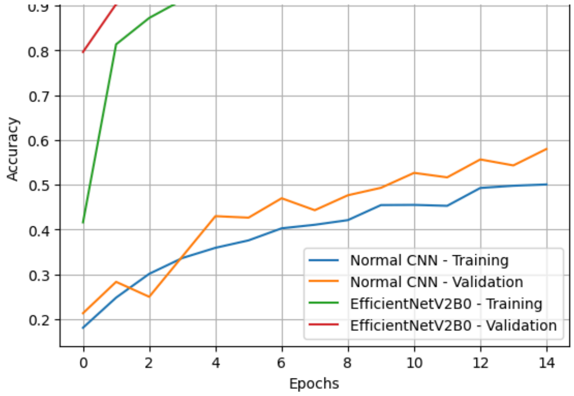
```
ccuracy: 0.4767 - val_loss: 1.4897 - learning_rate: 0.0010
Epoch 10/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 10s 91ms/step - accuracy: 0.4577 - loss: 1.5337 - val_a
ccuracy: 0.4933 - val_loss: 1.4748 - learning_rate: 0.0010
Epoch 11/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 7s 98ms/step - accuracy: 0.4625 - loss: 1.4887 - val_ac
curacy: 0.5267 - val_loss: 1.4806 - learning_rate: 0.0010
Epoch 12/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 6s 83ms/step - accuracy: 0.4660 - loss: 1.4985 - val_ac
curacy: 0.5167 - val_loss: 1.4915 - learning_rate: 0.0010
Epoch 13/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 89ms/step - accuracy: 0.4701 - loss: 1.4690 - val_a
ccuracy: 0.5567 - val_loss: 1.3467 - learning_rate: 5.0000e-04
Epoch 14/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 11s 95ms/step - accuracy: 0.5041 - loss: 1.3938 - val_a
ccuracy: 0.5433 - val_loss: 1.3434 - learning_rate: 5.0000e-04
Epoch 15/15
76/76 ━━━━━━━━━━━━━━━━━━━━ 6s 81ms/step - accuracy: 0.5215 - loss: 1.3725 - val_ac
curacy: 0.5800 - val_loss: 1.2799 - learning_rate: 5.0000e-04
```

Plot between CNN and EfficientNet

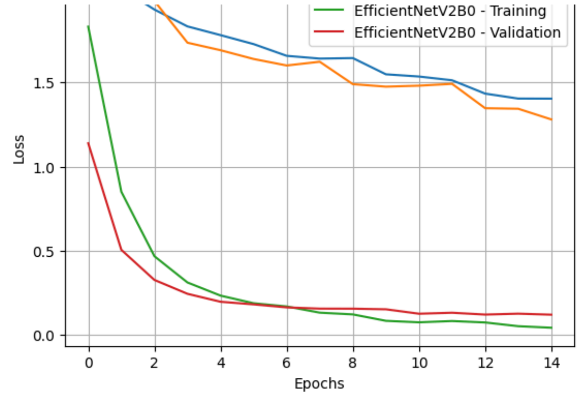
In [17]:

```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()
```





Accuracy
Epochs
Normal CNN - Training
Normal CNN - Validation
EfficientNetV2B0 - Training
EfficientNetV2B0 - Validation



Loss
Epochs
EfficientNetV2B0 - Training
EfficientNetV2B0 - Validation