In [1]:

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [7]:

```
train_data_dir = '/Users/3gendatasystemsprivatelimited/Desktop/PlasticScan Final'
```

In [8]:

```
num_classes = 4
```

In [9]:

```
image_size = (224, 224)
batch_size = 32
```

In [10]:

```
datagen = ImageDataGenerator(
    rescale=1.0/255.0,  # Normalize pixel values to [0, 1]
    rotation_range=20,  # Randomly rotate images
    width_shift_range=0.2,  # Randomly shift images horizontally
    height_shift_range=0.2,  # Randomly shift images vertically
    horizontal_flip=True,  # Randomly flip images horizontally
    zoom_range=0.2  # Randomly zoom in on images
)
```

In [11]:

```
train_generator = datagen.flow_from_directory(
    train_data_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'  # Set class_mode to 'categorical'
)
```

```
Found 27 images belonging to 4 classes.
```

In [12]:

```
model = keras.Sequential([
    layers.Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes, activation='softmax')  # Update output units to num_cl
])
```

In [13]:

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'
```

In [50]:

```
model.fit(train_generator, epochs=20)
```

```
Epoch 1/20
1/1 [==============================] - 1s 925ms/step - loss: 9.3069 -
accuracy: 0.3704
Epoch 2/20
1/1 [==============================] - 1s 543ms/step - loss: 4.0476 -
accuracy: 0.2222
Epoch 3/20
1/1 [==============================] - 1s 525ms/step - loss: 2.1757 -
accuracy: 0.2222
Epoch 4/20
1/1 [==============================] - 1s 530ms/step - loss: 1.5402 -
accuracy: 0.2963
Epoch 5/20
1/1 [==============================] - 1s 547ms/step - loss: 1.4374 -
accuracy: 0.1852
Epoch 6/20
1/1 [==============================] - 1s 532ms/step - loss: 1.3908 -
accuracy: 0.1852
Epoch 7/20
1/1 [==============================] - 1s 538ms/step - loss: 1.3800 -
accuracy: 0.1852
Epoch 8/20
1/1 [==============================] - 1s 536ms/step - loss: 1.3731 -
accuracy: 0.4074
Epoch 9/20
1/1 [==============================] - 1s 545ms/step - loss: 1.3543 -
accuracy: 0.4074
Epoch 10/20
1/1 [==============================] - 1s 580ms/step - loss: 1.3403 -
accuracy: 0.3704
Epoch 11/20
1/1 [==============================] - 1s 534ms/step - loss: 1.3086 -
accuracy: 0.3704
Epoch 12/20
1/1 [==============================] - 1s 531ms/step - loss: 1.2819 -
accuracy: 0.3704
Epoch 13/20
1/1 [==============================] - 1s 528ms/step - loss: 1.2579 -
accuracy: 0.4074
Epoch 14/20
1/1 [==============================] - 1s 531ms/step - loss: 1.2569 -
accuracy: 0.4815
Epoch 15/20
1/1 [==============================] - 1s 530ms/step - loss: 1.1977 -
accuracy: 0.4074
Epoch 16/20
1/1 [==============================] - 1s 525ms/step - loss: 1.1853 -
accuracy: 0.4815
Epoch 17/20
1/1 [==============================] - 1s 519ms/step - loss: 1.1228 -
accuracy: 0.5185
Epoch 18/20
1/1 [==============================] - 1s 527ms/step - loss: 1.1925 -
accuracy: 0.4815
Epoch 19/20
1/1 [==============================] - 1s 524ms/step - loss: 1.1298 -
accuracy: 0.4815
Epoch 20/20
1/1 [==============================] - 1s 536ms/step - loss: 1.1204 -
accuracy: 0.5185
```

Out[50]:

```
<keras.src.callbacks.History at 0x28bd6fe20>
```

In [51]:

```
model.save('/Users/3gendatasystemsprivatelimited/Desktop/HDPE model')
```

```
INFO:tensorflow:Assets written to: /Users/3gendatasystemsprivatelimite
d/Desktop/HDPE model/assets

INFO:tensorflow:Assets written to: /Users/3gendatasystemsprivatelimite
d/Desktop/HDPE model/assets
```

In [52]:

```
pip install opencv-python
```

```
Requirement already satisfied: opencv-python in ./anaconda3/lib/python
3.10/site-packages (4.8.0.76)
Requirement already satisfied: numpy>=1.21.4 in ./anaconda3/lib/python
3.10/site-packages (from opencv-python) (1.23.5)
Note: you may need to restart the kernel to use updated packages.
```

In [18]:

```
import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

In [19]:

```
model = keras.models.load_model('/Users/3gendatasystemsprivatelimited/Desktop/HDPE m
```

In [53]:

```
img_path = "/Users/3gendatasystemsprivatelimited/Desktop/test img/HDPE.jpeg"
img = cv2.imread(img_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (224, 224))
```

In [54]:

```
img = img / 255.0
img = np.expand_dims(img, axis=0)
```

In [55]:

```
probs = model.predict(img)
```

```
WARNING:tensorflow:5 out of the last 6 calls to <function Model.make_p
redict_function.<locals>.predict_function at 0x28b94e830> triggered t
f.function retracing. Tracing is expensive and the excessive number of
tracings could be due to (1) creating @tf.function repeatedly in a loo
p, (2) passing tensors with different shapes, (3) passing Python objec
ts instead of tensors. For (1), please define your @tf.function outsid
e of the loop. For (2), @tf.function has reduce_retracing=True option
that can avoid unnecessary retracing. For (3), please refer to http
s://www.tensorflow.org/guide/function#controlling_retracing (https://w
ww.tensorflow.org/guide/function#controlling_retracing) and https://ww
w.tensorflow.org/api_docs/python/tf/function (https://www.tensorflow.o
rg/api_docs/python/tf/function) for  more details.

WARNING:tensorflow:5 out of the last 6 calls to <function Model.make_p
redict_function.<locals>.predict_function at 0x28b94e830> triggered t
f.function retracing. Tracing is expensive and the excessive number of
tracings could be due to (1) creating @tf.function repeatedly in a loo
p, (2) passing tensors with different shapes, (3) passing Python objec
ts instead of tensors. For (1), please define your @tf.function outsid
e of the loop. For (2), @tf.function has reduce_retracing=True option
that can avoid unnecessary retracing. For (3), please refer to http
s://www.tensorflow.org/guide/function#controlling_retracing (https://w
ww.tensorflow.org/guide/function#controlling_retracing) and https://ww
w.tensorflow.org/api_docs/python/tf/function (https://www.tensorflow.o
rg/api_docs/python/tf/function) for  more details.

1/1 [==============================] - 0s 43ms/step
```

In [56]:

```
plt.imshow(img[0])
plt.title(f'HDPE Probability: {probs[0][0]:.2f}')
plt.axis('off')
plt.show()
```

TESTING WITH PVC IMAGE

In [57]:

```
 import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
import cv2
```

In [58]:

```
model = keras.models.load_model('/Users/3gendatasystemsprivatelimited/Desktop/HDPE m
```

In [59]:

```
img_path = "/Users/3gendatasystemsprivatelimited/Desktop/test img/PVC6.jpeg"
img = cv2.imread(img_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (224, 224))
```

In [60]:

```
img = img / 255.0
img = np.expand_dims(img, axis=0)
```

In [61]:

```
probs = model.predict(img)
```

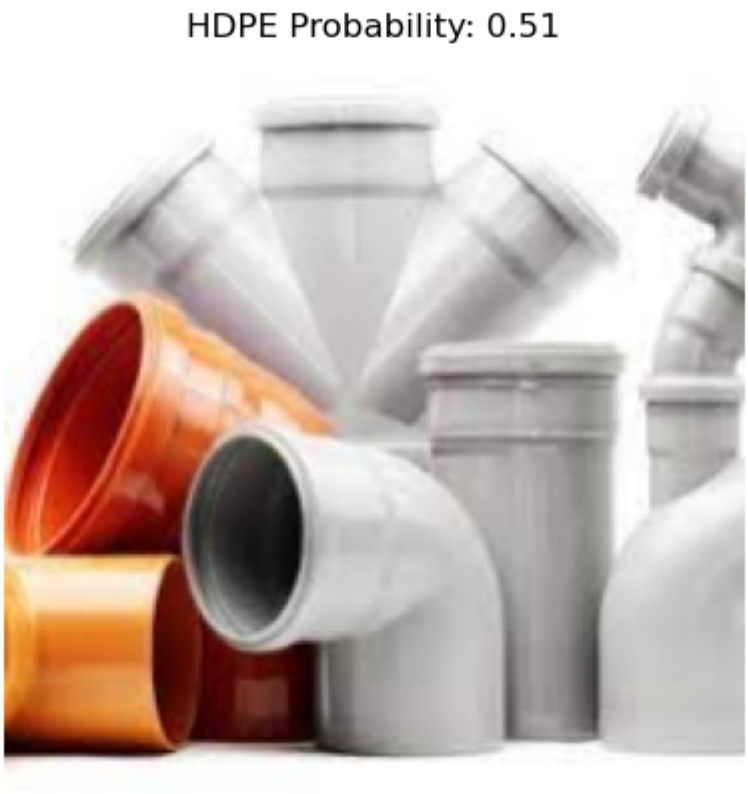
```
WARNING:tensorflow:6 out of the last 7 calls to <function Model.make_p
redict_function.<locals>.predict_function at 0x17e5d3c70> triggered t
f.function retracing. Tracing is expensive and the excessive number of
tracings could be due to (1) creating @tf.function repeatedly in a loo
p, (2) passing tensors with different shapes, (3) passing Python objec
ts instead of tensors. For (1), please define your @tf.function outsid
e of the loop. For (2), @tf.function has reduce_retracing=True option
that can avoid unnecessary retracing. For (3), please refer to http
s://www.tensorflow.org/guide/function#controlling_retracing (https://w
ww.tensorflow.org/guide/function#controlling_retracing) and https://ww
w.tensorflow.org/api_docs/python/tf/function (https://www.tensorflow.o
rg/api_docs/python/tf/function) for  more details.

WARNING:tensorflow:6 out of the last 7 calls to <function Model.make_p
redict_function.<locals>.predict_function at 0x17e5d3c70> triggered t
f.function retracing. Tracing is expensive and the excessive number of
tracings could be due to (1) creating @tf.function repeatedly in a loo
p, (2) passing tensors with different shapes, (3) passing Python objec
ts instead of tensors. For (1), please define your @tf.function outsid
e of the loop. For (2), @tf.function has reduce_retracing=True option
that can avoid unnecessary retracing. For (3), please refer to http
s://www.tensorflow.org/guide/function#controlling_retracing (https://w
ww.tensorflow.org/guide/function#controlling_retracing) and https://ww
w.tensorflow.org/api_docs/python/tf/function (https://www.tensorflow.o
rg/api_docs/python/tf/function) for  more details.

1/1 [==============================] - 0s 45ms/step
```

In [62]:

```
plt.imshow(img[0])
plt.title(f'HDPE Probability: {probs[0][0]:.2f}')
plt.axis('off')
plt.show()
```



TESTING WITH PET IMAGE

In [63]:

```
img_path = "/Users/3gendatasystemsprivatelimited/Desktop/PlasticScan Final/PET/PET1.
img = cv2.imread(img_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (224, 224))
```

In [64]:

```
img = img / 255.0
img = np.expand_dims(img, axis=0)
```

In [65]:

```
probs = model.predict(img)
```

```
1/1 [==============================] - 0s 175ms/step
```

In [66]:

```
plt.imshow(img[0])
plt.title(f'HDPE Probability: {probs[0][0]:.2f}')
plt.axis('off')
plt.show()
```

TESTING WITH LDPE IMAGE

In [67]:

```
img_path = "/Users/3gendatasystemsprivatelimited/Desktop/PlasticScan Final/LDPE/LDPE
img = cv2.imread(img_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
img = cv2.resize(img, (224, 224))
```

In [68]:

```
img = img / 255.0
img = np.expand_dims(img, axis=0)
```

In [69]:

```
probs = model.predict(img)
```

```
1/1 [==============================] - 0s 21ms/step
```

In [70]:

```
plt.imshow(img[0])
plt.title(f'HDPE Probability: {probs[0][0]:.2f}')
plt.axis('off')
plt.show()
```