```
# STEP 1: Install required packages
!pip install opencv-python-headless

# STEP 2: Import libraries
import os
import zipfile
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from google.colab import files

# STEP 3: Upload ZIP file manually via sidebar or use below to upload programmatically
# files.upload()  # Uncomment if you want a file dialog

# STEP 4: Extract uploaded ZIP file
zip_file_name = "seg_pred.zip"  # Change this if your uploaded file has a different name
with zipfile.ZipFile(zip_file_name, 'r') as zip_ref:
    zip_ref.extractall("dataset")

# STEP 5: Set paths and parameters
dataset_dir = "dataset"
img_size = 150
batch_size = 32

# STEP 6: Image preprocessing with ImageDataGenerator
datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.2
)

train_data = datagen.flow_from_directory(
    dataset_dir,
    target_size=(img_size, img_size),
    batch_size=batch_size,
    class_mode='categorical',
    subset='training'
)

val_data = datagen.flow_from_directory(
    dataset_dir,
    target_size=(img_size, img_size),
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation'
)

# STEP 7: Define CNN model
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(img_size, img_size, 3)),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(train_data.num_classes, activation='softmax')
])

# STEP 8: Compile and train
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

print("🔁 Training CNN Model...")
model.fit(train_data, epochs=10, validation_data=val_data)

# STEP 9: Evaluate
loss, accuracy = model.evaluate(val_data)
print(f"\n✅ Final Accuracy: {accuracy*100:.2f}%")

# STEP 10: Save model (optional)
model.save("image_classifier_model.h5")
```

```
Requirement already satisfied: opencv-python-headless in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.11/dist-packages (from opencv-python-headless) (2.0.2)
Found 5658 images belonging to 1 classes.
Found 1414 images belonging to 1 classes.
/usr/local/lib/python3.11/dist-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
Training CNN Model...
Epoch 1/10
/usr/local/lib/python3.11/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` cl
  self._warn_if_super_not_called()
/usr/local/lib/python3.11/dist-packages/keras/src/ops/nn.py:907: UserWarning: You are using a softmax over axis -1 of a tensor of sh
  warnings.warn(
/usr/local/lib/python3.11/dist-packages/keras/src/losses/losses.py:33: SyntaxWarning: In loss categorical_crossentropy, expected y_p
  return self.fn(y_true, y_pred, **self._fn_kwargs)
177/177 ━━━━━━━━━━━━━━━━━━━━ 214s 1s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 2/10
177/177 ━━━━━━━━━━━━━━━━━━━━ 224s 1s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 3/10
177/177 ━━━━━━━━━━━━━━━━━━━━ 216s 1s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 4/10
177/177 ━━━━━━━━━━━━━━━━━━━━ 225s 1s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 5/10
177/177 ━━━━━━━━━━━━━━━━━━━━ 221s 1s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 6/10
177/177 ━━━━━━━━━━━━━━━━━━━━ 221s 1s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 7/10
177/177 ━━━━━━━━━━━━━━━━━━━━ 260s 1s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 8/10
177/177 ━━━━━━━━━━━━━━━━━━━━ 220s 1s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 9/10
177/177 ━━━━━━━━━━━━━━━━━━━━ 221s 1s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
Epoch 10/10
177/177 ━━━━━━━━━━━━━━━━━━━━ 224s 1s/step - accuracy: 1.0000 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.0000e+00
45/45 ━━━━━━━━━━━━━━━━━━━━ 14s 310ms/step - accuracy: 1.0000 - loss: 0.0000e+00
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is c

Final Accuracy: 100.00%
```