```
import tensorflow as tf
import cv2
import numpy as np


# Load the dataset
train_dataset = tf.keras.preprocessing.image_dataset_from_directory(
    'path/to/dataset',
    validation_split=0.2,
    subset='training',
    seed=123,
    image_size=(256, 256),
    batch_size=32)


# Build the model
model = tf.keras.Sequential([
    tf.keras.layers.experimental.preprocessing.Rescaling(1./255),
    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.MaxPooling2D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(3)
])


# Compile the model
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])

# Train the model
model.fit(train_dataset, epochs=10)


# Load the OpenCV module and start the webcam
cap = cv2.VideoCapture(0)


# Loop through each frame of the webcam input
while True:
    # Read a frame from the webcam
    ret, frame = cap.read()

    # Preprocess the image
    resized_image = cv2.resize(frame, (256, 256))
    normalized_image = resized_image / 255.0
    input_image = np.expand_dims(normalized_image, axis=0)


# Use the model to make a prediction on the preprocessed image
    prediction = model.predict(input_image)

    # Get the predicted class index
    predicted_class_index = np.argmax(prediction)

    # Display the result on the screen
    cv2.putText(frame, 'Weed Type: ' + str(predicted_class_index), (10, 30),
                cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
    cv2.imshow('Weed Detector', frame)

    # Press 'q' to quit
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break


# Clean up and release the resources
cap.release()
cv2.destroyAllWindows()
```

✓ 0s completed at 1:54 PM ● ×