EX NO: 2	IMAGE SEGMENTATION

Aim:

To build and evaluate a simple Convolutional Neural Network (CNN) model using TensorFlow and Keras for segmenting objects in grayscale or RGB images, such as identifying foreground objects in a binary segmentation task.

Algorithm:

1. **Import Libraries:** Import TensorFlow, Keras, NumPy, and Matplotlib.
2. **Prepare Dataset:** Load or simulate image data and corresponding binary masks for segmentation.
3. **Normalize Data:** Scale pixel values of images and masks between 0 and 1.
4. **Build CNN Model:** Construct a simple encoder-decoder style CNN with Conv2D, MaxPooling2D, UpSampling2D layers.
5. **Compile the Model:** Use 'binary_crossentropy' loss and 'adam' optimizer.
6. **Train the Model:** Fit the model on training data.

Code:

```
import numpy as np

import tensorflow as tf

from tensorflow.keras import layers, models

import matplotlib.pyplot as plt
```

```
# Simulate random grayscale images and masks
x_train = np.random.rand(100, 64, 64, 1)  # 100 training images
y_train = (x_train > 0.5).astype(np.float32)  # simple thresholded mask
model = models.Sequential([
    layers.Conv2D(16, (3, 3), activation='relu', padding='same', input_shape=(64, 64, 1)),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(32, (3, 3), activation='relu', padding='same'),
    layers.UpSampling2D((2, 2)),
    layers.Conv2D(1, (3, 3), activation='sigmoid', padding='same')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.fit(x_train, y_train, epochs=5, batch_size=10, validation_split=0.1)
sample_img = x_train[0:1]
predicted_mask = model.predict(sample_img)[0]
plt.figure(figsize=(8, 4))
plt.subplot(1, 2, 1)
plt.imshow(sample_img[0, ..., 0], cmap='gray')
plt.title('Original Image')
plt.axis('off')
plt.subplot(1, 2, 2)
plt.imshow(predicted_mask[..., 0], cmap='gray')
```

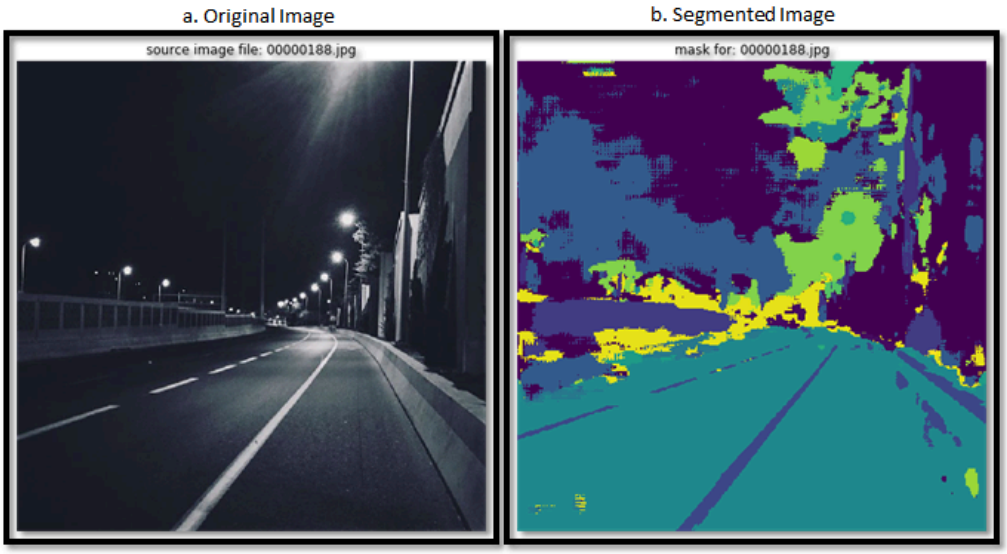
```
plt.title('Predicted Mask')

plt.axis('off')

plt.tight_layout()

plt.show()
```

Output:



Result:

A simple CNN model was successfully built and trained to perform binary image segmentation. The predicted segmentation mask shows the model's ability to distinguish regions based on pixel intensity patterns. The sample output displays both the input image and its corresponding predicted mask.