EXERCISE NO. 2
CNN FOR IMAGE SEGMENTATION

AIM:
To write a program to build a CNN model for image segmentation.

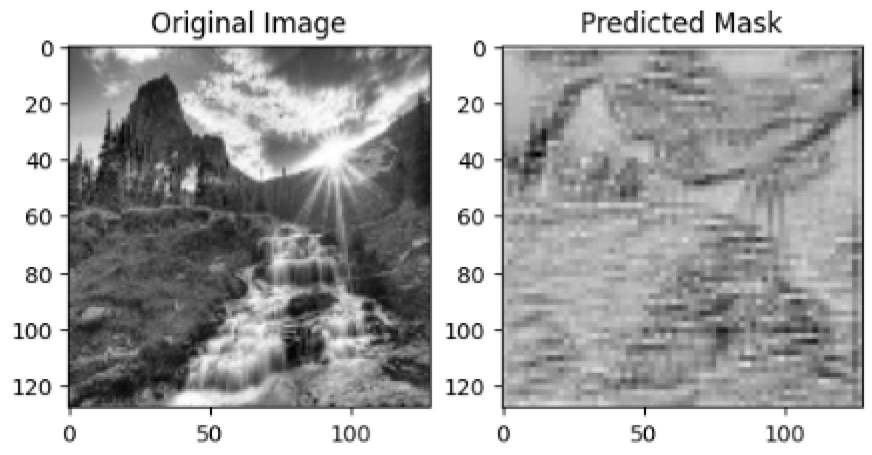
ALGORITHM:

1. Import the necessary libraries.
2. Load the image.
3. Apply thresholding.
4. Perform noise removal.
5. Build the CNN model.
6. Evaluate the model for the unknown image.

PROGRAM:

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

image = cv2.imread("/img-1.jpg", cv2.IMREAD_GRAYSCALE)
image = cv2.resize(image, (128, 128))

image_8u = image.astype(np.uint8)

_, mask = cv2.threshold(image, 0.5, 1, cv2.THRESH_BINARY + cv2.THRESH_OTSU)

image = image / 255.0

image = np.expand_dims(image, axis=(0, -1))  # Add batch & channel dims
mask = np.expand_dims(mask, axis=(0, -1))

model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation='relu', padding='same', input_shape=(128, 128, 1)),
    tf.keras.layers.MaxPooling2D((2, 2)),
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', padding='same'),
    tf.keras.layers.UpSampling2D((2, 2)),
    tf.keras.layers.Conv2D(1, (1, 1), activation='sigmoid')
])

model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

model.fit(image, mask, epochs=5, batch_size=1, verbose=1)

pred_mask = model.predict(image)[0]

plt.subplot(1, 2, 1)
plt.title("Original Image")
plt.imshow(image[0, :, :, 0], cmap='gray')

plt.subplot(1, 2, 2)
plt.title("Predicted Mask")
plt.imshow(pred_mask[:, :, 0], cmap='gray')

plt.show()
```

OUTPUT:



RESULT:
Thus the program has been successfully implemented and verified.