In [1]:

```
!pip install tensorflow matplotlib numpy

import numpy as np
import tensorflow as tf
from tensorflow.keras import layers, models
import matplotlib.pyplot as plt
```

```
Requirement already satisfied: requests<3,>=2.21.0 in c:\users\kudk73\anaconda3\lib\site-packa
ges (from tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorflow) (2.31.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in c:\users\kudk73\anacon
da3\lib\site-packages (from tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorflow) (0.
7.2)
Requirement already satisfied: werkzeug>=1.0.1 in c:\users\kudk73\anaconda3\lib\site-packages
(from tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorflow) (2.2.3)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in c:\users\kudk73\anaconda3\lib\site-pa
ckages (from google-auth<3,>=1.6.3->tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorf
low) (5.3.2)
Requirement already satisfied: pyasn1-modules>=0.2.1 in c:\users\kudk73\anaconda3\lib\site-pac
kages (from google-auth<3,>=1.6.3->tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorfl
ow) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in c:\users\kudk73\anaconda3\lib\site-packages (f
rom google-auth<3,>=1.6.3->tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorflow) (4.
7.2)
Requirement already satisfied: requests-oauthlib>=0.7.0 in c:\users\kudk73\anaconda3\lib\site-
packages (from google-auth-oauthlib<2,>=0.5->tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0
->tensorflow) (1.3.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\kudk73\anaconda3\lib\site-
packages (from requests<3,>=2.21.0->tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorf
low) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\kudk73\anaconda3\lib\site-packages (fr
om requests<3,>=2.21.0->tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorflow) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\kudk73\anaconda3\lib\site-packag
es (from requests<3,>=2.21.0->tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorflow)
(1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\kudk73\anaconda3\lib\site-packag
es (from requests<3,>=2.21.0->tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorflow)
(2023.7.22)
Requirement already satisfied: MarkupSafe>=2.1.1 in c:\users\kudk73\anaconda3\lib\site-package
s (from werkzeug>=1.0.1->tensorboard<2.16,>=2.15->tensorflow-intel==2.15.0->tensorflow) (2.1.
1)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in c:\users\kudk73\anaconda3\lib\site-pack
ages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard<2.16,>=2.15->tensorflow-i
ntel==2.15.0->tensorflow) (0.4.8)
Requirement already satisfied: oauthlib>=3.0.0 in c:\users\kudk73\anaconda3\lib\site-packages
(from requests-oauthlib>=0.7.0->google-auth-oauthlib<2,>=0.5->tensorboard<2.16,>=2.15->tensorf
low-intel==2.15.0->tensorflow) (3.2.2)

[notice] A new release of pip is available: 24.0 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
WARNING:tensorflow:From C:\Users\kudk73\anaconda3\Lib\site-packages\keras\src\losses.py:2976:
The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.
sparse_softmax_cross_entropy instead.
```

```
C:\Users\kudk73\anaconda3\Lib\site-packages\pandas\core\arrays\masked.py:61: UserWarning: Pand
as requires version '1.3.6' or newer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

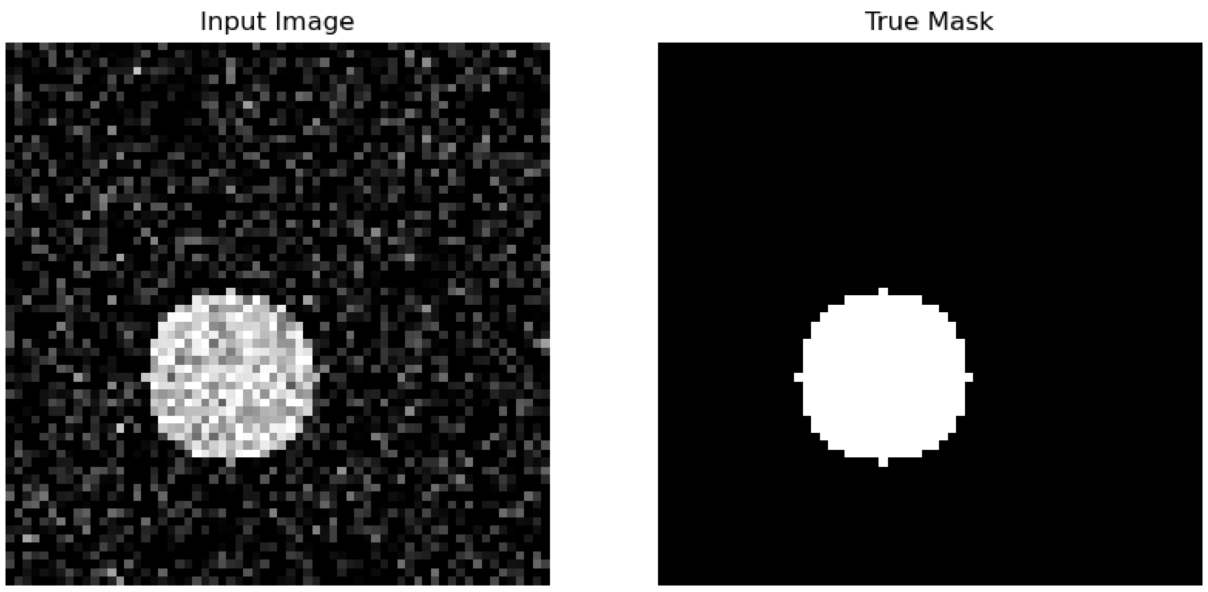
In [4]:

```python
def create_circle_mask(shape, center, radius):
    h, w = shape
    y, x = np.ogrid[:h, :w]
    mask = (x - center[0]) ** 2 + (y - center[1]) ** 2 <= radius ** 2
    return mask.astype(np.float32)
def create_square_mask(shape, center, size):
    h, w = shape
    y, x = np.ogrid[:h, :w]
    mask = (np.abs(x - center[0]) <= size//2) & (np.abs(y - center[1]) <= size//2)
    return mask.astype(np.float32)
np.random.seed(42)
n_samples = 100
img_size = 64
X = np.zeros((n_samples, img_size, img_size, 1), dtype=np.float32)  # Images
y = np.zeros((n_samples, img_size, img_size, 1), dtype=np.float32)  # Masks
for i in range(n_samples):
    img = np.zeros((img_size, img_size))
    mask = np.zeros((img_size, img_size))
    cx = np.random.randint(20, img_size - 20)
    cy = np.random.randint(20, img_size - 20)
    if np.random.rand() > 0.5:
        mask = create_circle_mask((img_size, img_size), (cx, cy), 10)
    else:
        mask = create_square_mask((img_size, img_size), (cx, cy), 15)
    img = mask * 0.8 + np.random.randn(*img.shape) * 0.2
    img = np.clip(img, 0, 1)

    X[i, :, :, 0] = img
    y[i, :, :, 0] = mask

print("X shape:", X.shape)  # (100, 64, 64, 1)
print("y shape:", y.shape)  # (100, 64, 64, 1)
```

```
X shape: (100, 64, 64, 1)
y shape: (100, 64, 64, 1)
```

In [5]:

```
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.title("Input Image")
plt.imshow(X[0, :, :, 0], cmap='gray')
plt.axis("off")

plt.subplot(1, 2, 2)
plt.title("True Mask")
plt.imshow(y[0, :, :, 0], cmap='gray')
plt.axis("off")
plt.show()
```



In [6]:

```
def simple_unet():
    inputs = layers.Input(shape=(64, 64, 1))
    c1 = layers.Conv2D(16, (3, 3), activation='relu', padding='same')(inputs)
    c1 = layers.Conv2D(16, (3, 3), activation='relu', padding='same')(c1)
    p1 = layers.MaxPooling2D((2, 2))(c1)
    c2 = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(p1)
    c2 = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(c2)
    p2 = layers.MaxPooling2D((2, 2))(c2)
    c3 = layers.Conv2D(64, (3, 3), activation='relu', padding='same')(p2)
    c3 = layers.Conv2D(64, (3, 3), activation='relu', padding='same')(c3)
    u1 = layers.UpSampling2D((2, 2))(c3)
    u1 = layers.concatenate([u1, c2])
    c4 = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(u1)
    c4 = layers.Conv2D(32, (3, 3), activation='relu', padding='same')(c4)
    u2 = layers.UpSampling2D((2, 2))(c4)
    u2 = layers.concatenate([u2, c1])
    c5 = layers.Conv2D(16, (3, 3), activation='relu', padding='same')(u2)
    c5 = layers.Conv2D(16, (3, 3), activation='relu', padding='same')(c5)
    outputs = layers.Conv2D(1, (1, 1), activation='sigmoid')(c5)
    model = models.Model(inputs=[inputs], outputs=[outputs])
    return model
model = simple_unet()
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
model.summary()
```

```
WARNING:tensorflow:From C:\Users\kudk73\anaconda3\Lib\site-packages\keras\src\backend.py:1398:
The name tf.executing_eagerly_outside_functions is deprecated. Please use tf.compat.v1.executi
ng_eagerly_outside_functions instead.

WARNING:tensorflow:From C:\Users\kudk73\anaconda3\Lib\site-packages\keras\src\layers\pooling\m
ax_pooling2d.py:161: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instea
d.

WARNING:tensorflow:From C:\Users\kudk73\anaconda3\Lib\site-packages\keras\src\optimizers\__ini
t__.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer
instead.

Model: "model"
______________________________________________________________________________________________
____
 Layer (type)                Output Shape                 Param #   Connected to
==============================================================================================
====
 input_1 (InputLayer)        [(None, 64, 64, 1)]          0         []

 conv2d (Conv2D)             (None, 64, 64, 16)           160       ['input_1[0][0]']

 conv2d_1 (Conv2D)           (None, 64, 64, 16)           2320      ['conv2d[0][0]']

 max_pooling2d (MaxPooling2  (None, 32, 32, 16)           0         ['conv2d_1[0][0]']
 D)

 conv2d_2 (Conv2D)           (None, 32, 32, 32)           4640      ['max_pooling2d[0][0]']

 conv2d_3 (Conv2D)           (None, 32, 32, 32)           9248      ['conv2d_2[0][0]']

 max_pooling2d_1 (MaxPoolin  (None, 16, 16, 32)           0         ['conv2d_3[0][0]']
 g2D)

 conv2d_4 (Conv2D)           (None, 16, 16, 64)           18496     ['max_pooling2d_1[0][0]']

 conv2d_5 (Conv2D)           (None, 16, 16, 64)           36928     ['conv2d_4[0][0]']

 up_sampling2d (UpSampling2  (None, 32, 32, 64)           0         ['conv2d_5[0][0]']
 D)

 concatenate (Concatenate)   (None, 32, 32, 96)           0         ['up_sampling2d[0][0]',
                                                                     'conv2d_3[0][0]']

 conv2d_6 (Conv2D)           (None, 32, 32, 32)           27680     ['concatenate[0][0]']

 conv2d_7 (Conv2D)           (None, 32, 32, 32)           9248      ['conv2d_6[0][0]']

 up_sampling2d_1 (UpSamplin  (None, 64, 64, 32)           0         ['conv2d_7[0][0]']
 g2D)

 concatenate_1 (Concatenate  (None, 64, 64, 48)           0         ['up_sampling2d_1[0][0]',
 )                                                                   'conv2d_1[0][0]']

 conv2d_8 (Conv2D)           (None, 64, 64, 16)           6928      ['concatenate_1[0][0]']

 conv2d_9 (Conv2D)           (None, 64, 64, 16)           2320      ['conv2d_8[0][0]']

 conv2d_10 (Conv2D)          (None, 64, 64, 1)            17        ['conv2d_9[0][0]']

==============================================================================================
====
Total params: 117985 (460.88 KB)
Trainable params: 117985 (460.88 KB)
Non-trainable params: 0 (0.00 Byte)
______________________________________________________________________________________________
____
```

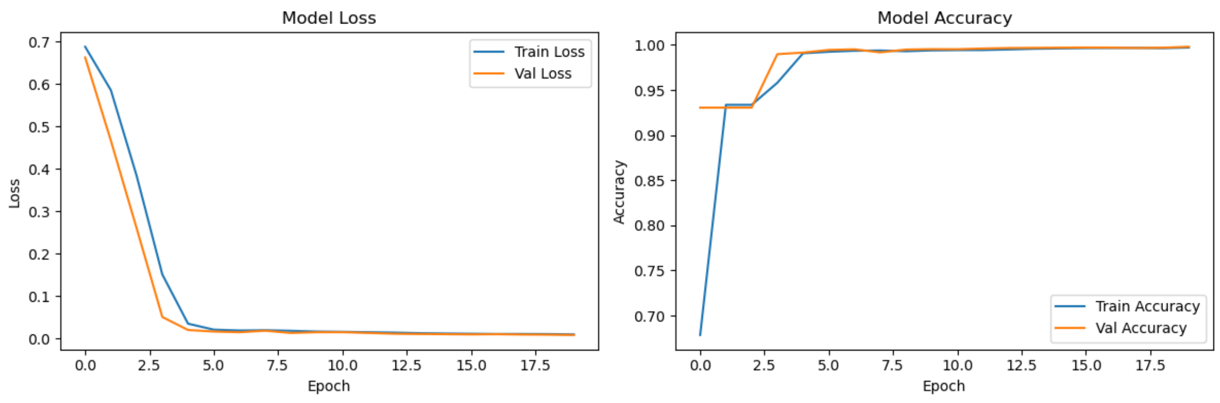
In [7]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
history = model.fit(
    X_train, y_train,
    epochs=20,
    batch_size=16,
    validation_data=(X_test, y_test),
    verbose=1
)
```

```
Epoch 1/20
WARNING:tensorflow:From C:\Users\kudk73\anaconda3\Lib\site-packages\keras\src\utils\tf_utils.p
y:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.Ragg
edTensorValue instead.

WARNING:tensorflow:From C:\Users\kudk73\anaconda3\Lib\site-packages\keras\src\engine\base_laye
r_utils.py:384: The name tf.executing_eagerly_outside_functions is deprecated. Please use tf.c
ompat.v1.executing_eagerly_outside_functions instead.

5/5 [==============================] - 4s 151ms/step - loss: 0.6878 - accuracy: 0.6786 - val_l
oss: 0.6626 - val_accuracy: 0.9305
Epoch 2/20
5/5 [==============================] - 0s 84ms/step - loss: 0.5853 - accuracy: 0.9336 - val_lo
ss: 0.4662 - val_accuracy: 0.9305
Epoch 3/20
5/5 [==============================] - 0s 80ms/step - loss: 0.3836 - accuracy: 0.9336 - val_lo
ss: 0.2599 - val_accuracy: 0.9305
Epoch 4/20
5/5 [==============================] - 0s 88ms/step - loss: 0.1505 - accuracy: 0.9580 - val_lo
ss: 0.0503 - val_accuracy: 0.9897
Epoch 5/20
5/5 [==============================] - 0s 72ms/step - loss: 0.0345 - accuracy: 0.9907 - val_lo
ss: 0.0196 - val_accuracy: 0.9915
Epoch 6/20
5/5 [==============================] - 0s 73ms/step - loss: 0.0204 - accuracy: 0.9924 - val_lo
ss: 0.0159 - val_accuracy: 0.9944
Epoch 7/20
5/5 [==============================] - 0s 73ms/step - loss: 0.0183 - accuracy: 0.9935 - val_lo
ss: 0.0144 - val_accuracy: 0.9951
Epoch 8/20
5/5 [==============================] - 0s 71ms/step - loss: 0.0190 - accuracy: 0.9938 - val_lo
ss: 0.0180 - val_accuracy: 0.9917
Epoch 9/20
5/5 [==============================] - 0s 75ms/step - loss: 0.0177 - accuracy: 0.9930 - val_lo
ss: 0.0126 - val_accuracy: 0.9948
Epoch 10/20
5/5 [==============================] - 0s 72ms/step - loss: 0.0159 - accuracy: 0.9939 - val_lo
ss: 0.0144 - val_accuracy: 0.9953
Epoch 11/20
5/5 [==============================] - 0s 72ms/step - loss: 0.0152 - accuracy: 0.9942 - val_lo
ss: 0.0144 - val_accuracy: 0.9952
Epoch 12/20
5/5 [==============================] - 0s 72ms/step - loss: 0.0144 - accuracy: 0.9943 - val_lo
ss: 0.0127 - val_accuracy: 0.9960
Epoch 13/20
5/5 [==============================] - 0s 71ms/step - loss: 0.0135 - accuracy: 0.9949 - val_lo
ss: 0.0108 - val_accuracy: 0.9966
Epoch 14/20
5/5 [==============================] - 0s 72ms/step - loss: 0.0120 - accuracy: 0.9957 - val_lo
ss: 0.0102 - val_accuracy: 0.9967
Epoch 15/20
5/5 [==============================] - 0s 73ms/step - loss: 0.0113 - accuracy: 0.9961 - val_lo
ss: 0.0097 - val_accuracy: 0.9969
Epoch 16/20
5/5 [==============================] - 0s 75ms/step - loss: 0.0107 - accuracy: 0.9964 - val_lo
ss: 0.0092 - val_accuracy: 0.9971
Epoch 17/20
5/5 [==============================] - 0s 73ms/step - loss: 0.0102 - accuracy: 0.9966 - val_lo
ss: 0.0097 - val_accuracy: 0.9970
Epoch 18/20
5/5 [==============================] - 0s 73ms/step - loss: 0.0102 - accuracy: 0.9967 - val_lo
ss: 0.0089 - val_accuracy: 0.9966
Epoch 19/20
5/5 [==============================] - 0s 74ms/step - loss: 0.0099 - accuracy: 0.9964 - val_lo
ss: 0.0086 - val_accuracy: 0.9969
Epoch 20/20
5/5 [==============================] - 0s 75ms/step - loss: 0.0090 - accuracy: 0.9971 - val_lo
ss: 0.0079 - val_accuracy: 0.9979
```

In [8]:

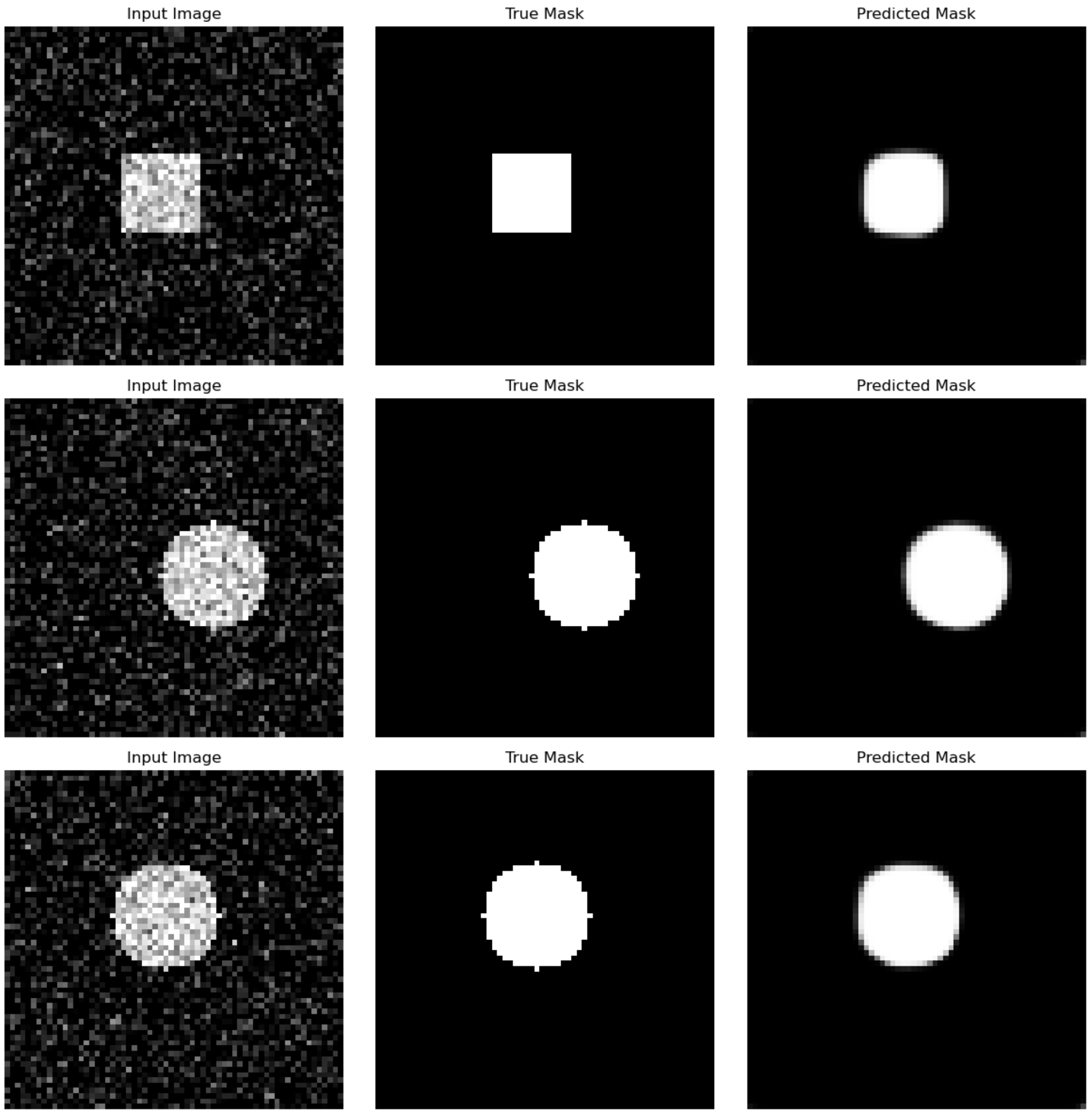
```
plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Val Loss')
plt.title('Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.subplot(1, 2, 2)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Val Accuracy')
plt.title('Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.tight_layout()
plt.show()
```



In [9]:

```python
y_pred = model.predict(X_test)
n_show = 3
plt.figure(figsize=(12, 4 * n_show))
for i in range(n_show):
    plt.subplot(n_show, 3, 3*i + 1)
    plt.imshow(X_test[i, :, :, 0], cmap='gray')
    plt.title("Input Image")
    plt.axis("off")
    plt.subplot(n_show, 3, 3*i + 2)
    plt.imshow(y_test[i, :, :, 0], cmap='gray')
    plt.title("True Mask")
    plt.axis("off")
    plt.subplot(n_show, 3, 3*i + 3)
    plt.imshow(y_pred[i, :, :, 0], cmap='gray')
    plt.title("Predicted Mask")
    plt.axis("off")
plt.tight_layout()
plt.show()
```

```
1/1 [==============================] - 0s 200ms/step
```


Input Image
True Mask
Predicted Mask
Input Image
True Mask
Predicted Mask
Input Image
True Mask
Predicted Mask