In [9]:
```
import warnings
warnings.filterwarnings("ignore")
import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from IPython.display import Image,display
import tensorflow as tf
from tensorflow.keras.models import load_model
```

In [10]:
```
model=load_model(filepath='Fruits_DenseNet201_model.keras',compile=False)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

In [11]:
```
class_labels=['Lime_Bad','Lime_Good']
```

In [16]:
```
image_path='Newli.jpeg'
```

In [17]:
```
import cv2
image=cv2.imread(image_path)
image=cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
image=cv2.resize(image,(128,128))
plt.imshow(image)
plt.title(label='Input Image')
plt.axis('off')
plt.show()
```

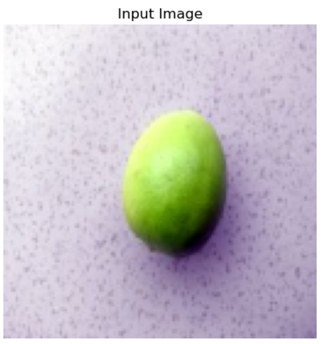
In [18]:
```
image=np.array(image)
image=image/255.0
image=np.expand_dims(image,axis=0)
print(image.shape)
```

```
(1, 128, 128, 3)
```

In [19]:
```
model_prediction=model.predict(image)
model_class=np.argmax(model_prediction[0])
model_label=class_labels[model_class]
fruit_name=model_label.split("_")[0]
fruit_quality=model_label.split("_")[1]
print(f"Fruit classified as -> {fruit_name}")
print(f"Quality assessed as -> {fruit_quality}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 16s 16s/step
Fruit classified as -> Lime
Quality assessed as -> Bad
```

In [21]:
```
image_path='Fruits/Lime_Good/fincheck.jpg'
import cv2
image=cv2.imread(image_path)
image=cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
image=cv2.resize(image,(128,128))
plt.imshow(image)
plt.title(label='Input Image')
plt.axis('off')
plt.show()
```



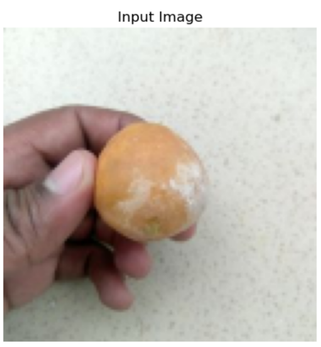
In [22]:
```
image=np.array(image)
image=image/255.0
image=np.expand_dims(image,axis=0)
print(image.shape)
```

```
(1, 128, 128, 3)
```

In [23]:
```
model_prediction=model.predict(image)
model_class=np.argmax(model_prediction[0])
model_label=class_labels[model_class]
fruit_name=model_label.split("_")[0]
fruit_quality=model_label.split("_")[1]
print(f"Fruit classified as -> {fruit_name}")
print(f"Quality assessed as -> {fruit_quality}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 590ms/step
Fruit classified as -> Lime
Quality assessed as -> Good
```

In [24]:
```
image_path='Fruits/Lime_Bad/blime.jpg'
import cv2
image=cv2.imread(image_path)
image=cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
image=cv2.resize(image,(128,128))
plt.imshow(image)
plt.title(label='Input Image')
plt.axis('off')
plt.show()
```



In [25]:
```
image=np.array(image)
image=image/255.0
image=np.expand_dims(image,axis=0)
print(image.shape)
```

```
(1, 128, 128, 3)
```

In [26]:
```
model_prediction=model.predict(image)
model_class=np.argmax(model_prediction[0])
model_label=class_labels[model_class]
fruit_name=model_label.split("_")[0]
fruit_quality=model_label.split("_")[1]
print(f"Fruit classified as -> {fruit_name}")
print(f"Quality assessed as -> {fruit_quality}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 255ms/step
Fruit classified as -> Lime
Quality assessed as -> Bad
```

In [ ]:
```
```