



---

# SPATIAL FILTER

## MODULE 7

---



Arranged By:

Rajendra Rakha Arya Prabaswara

(1941720080/21)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG



## Practicum

[https://github.com/Rjndrkha/PCVK\\_Genap\\_2022](https://github.com/Rjndrkha/PCVK_Genap_2022)

### 1. Mount Image From Drive & Importing libraries

```
+ Kode + Teks Salin ke Drive

from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive

[ ] import numpy as np
import matplotlib.pyplot as plt
import cv2 as cv
import math
from google.colab.patches import cv2_imshow
from PIL import Image as im
```

### 2. Konvolusi Function & Process

```
{x} [ ] def convolution2d(image, kernel, stride, padding):
# add padding
image = np.pad(image, padding, mode='constant', constant_values=0)

# pick img input width and height values including insertion pad
kernel_height, kernel_width = kernel.shape
padded_height, padded_width = image.shape

"""
Formula to find the output size using convolution:
[W-K+2SP/S]+1
W = input img size (width/height)
K = kernel size
P = padding size
S = stride value

"""

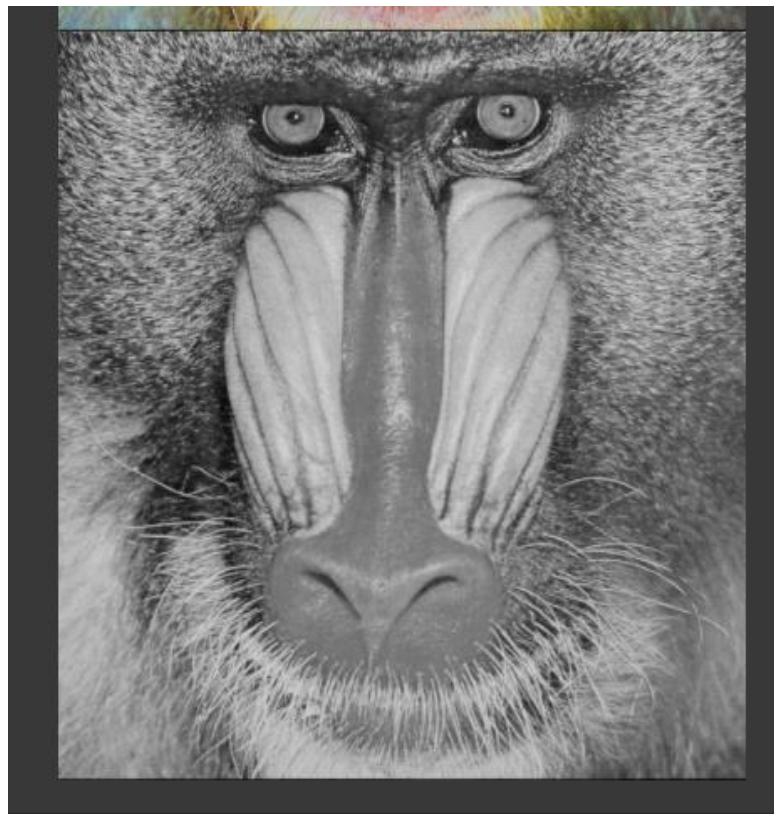
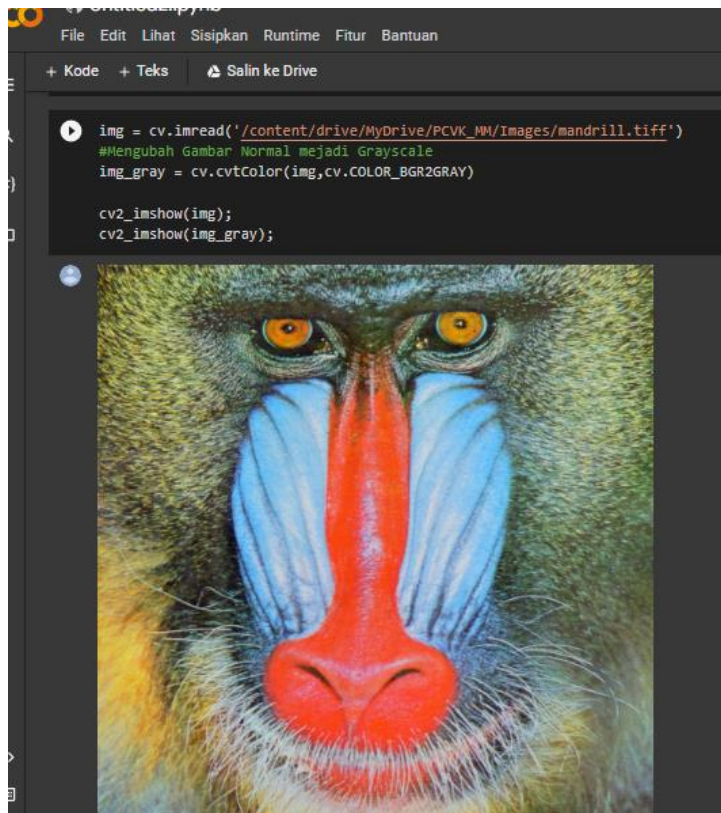
output_height = (padded_height - kernel_height) // stride + 1
output_width = (padded_width - kernel_width) // stride + 1

new_image = np.zeros((output_height, output_width)).astype(np.float32)

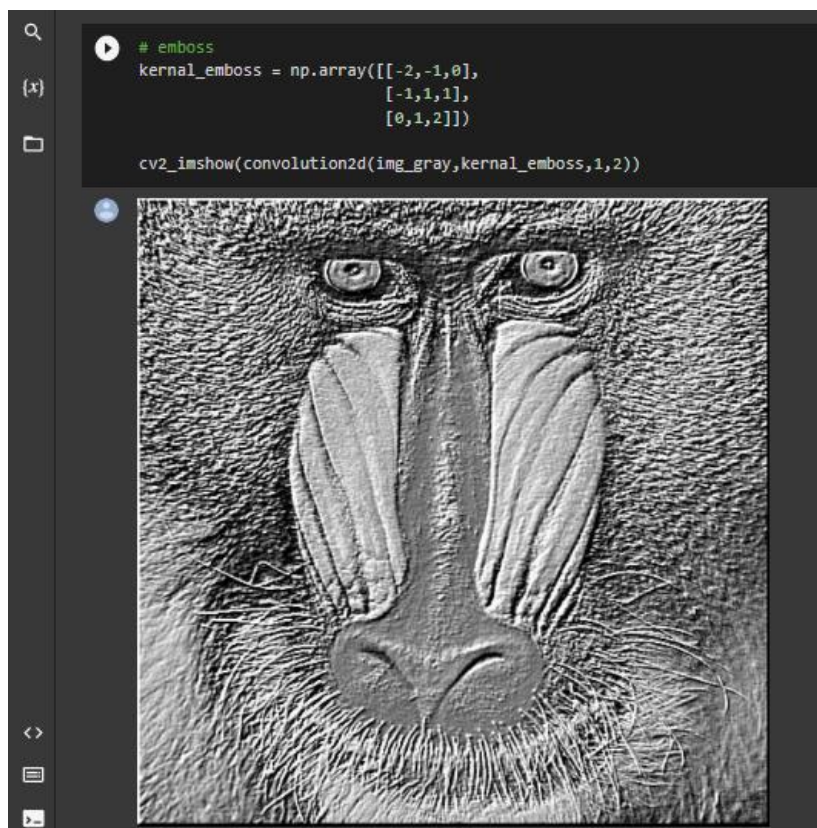
# convolution process
for y in range(0, output_height):
    for x in range(0, output_width):
        new_image[y][x] = np.sum(image[y * stride:y * stride + kernel_height, x * stride:x * stride + kernel_width] * kernel).astype(np.float32)
return new_image
<>
```



### 3. Normal Image To GrayScale



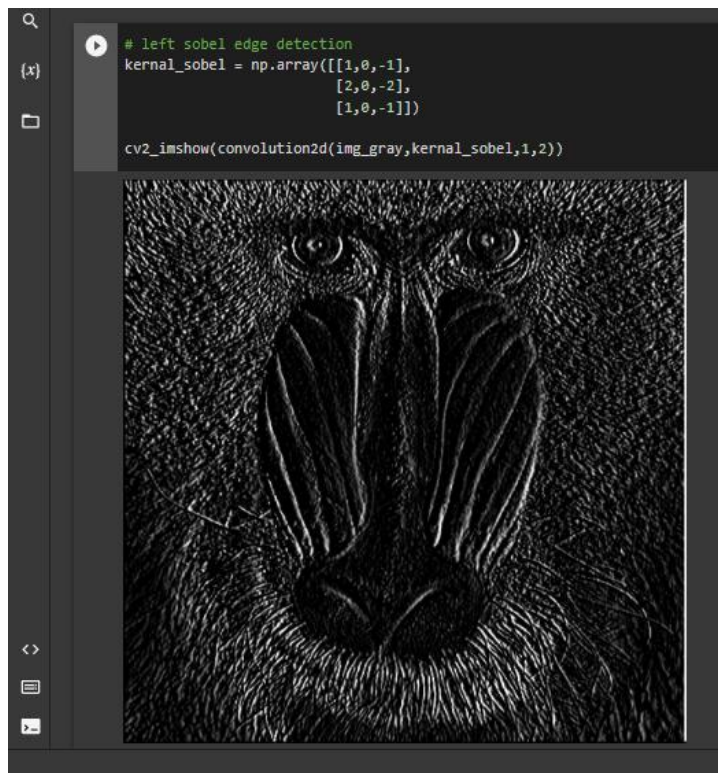
### 4. Image emboss



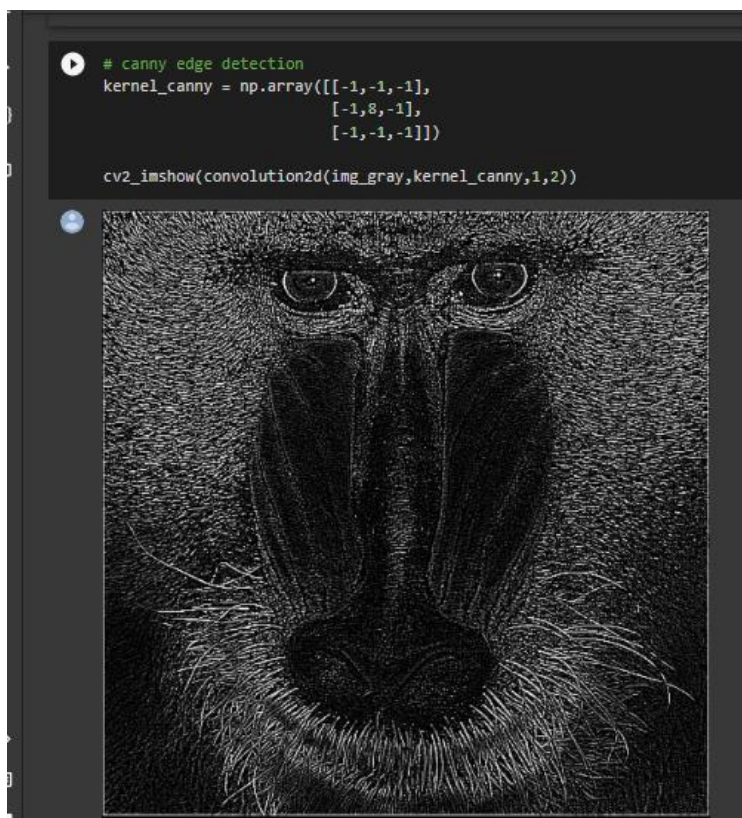




## 5. Left sobel Edge Image Detection

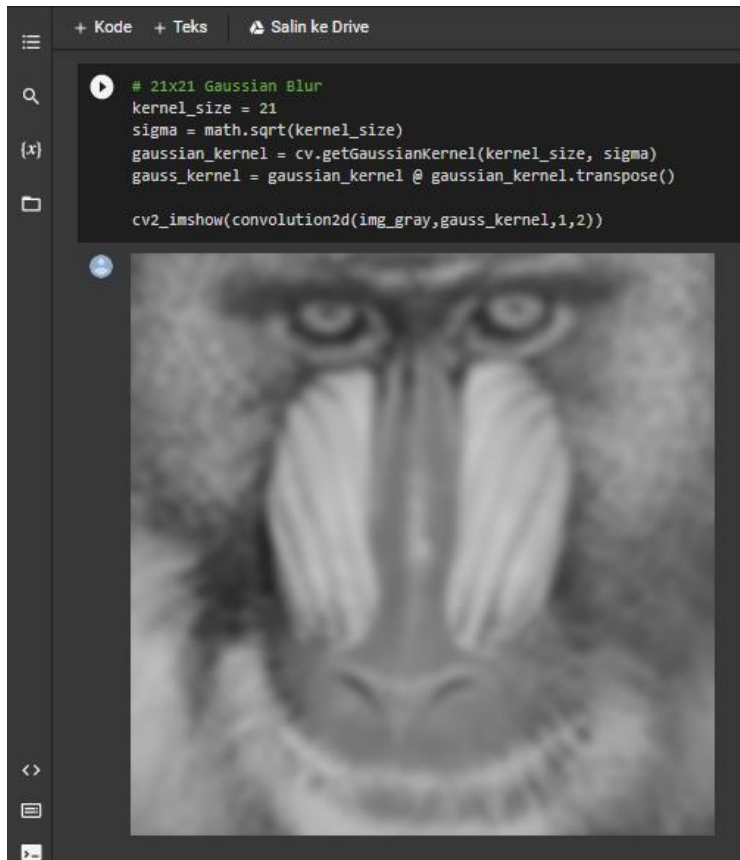


## 6. Canny Edge Image Detection





## 7. 21x21 Image Gaussian Blur





**Jurusan Teknologi Informasi Politeknik Negeri  
Malang.**

Rajendra Rakha Arya Prabaswara

1941720080-3H/20

---