

Tugas 04:

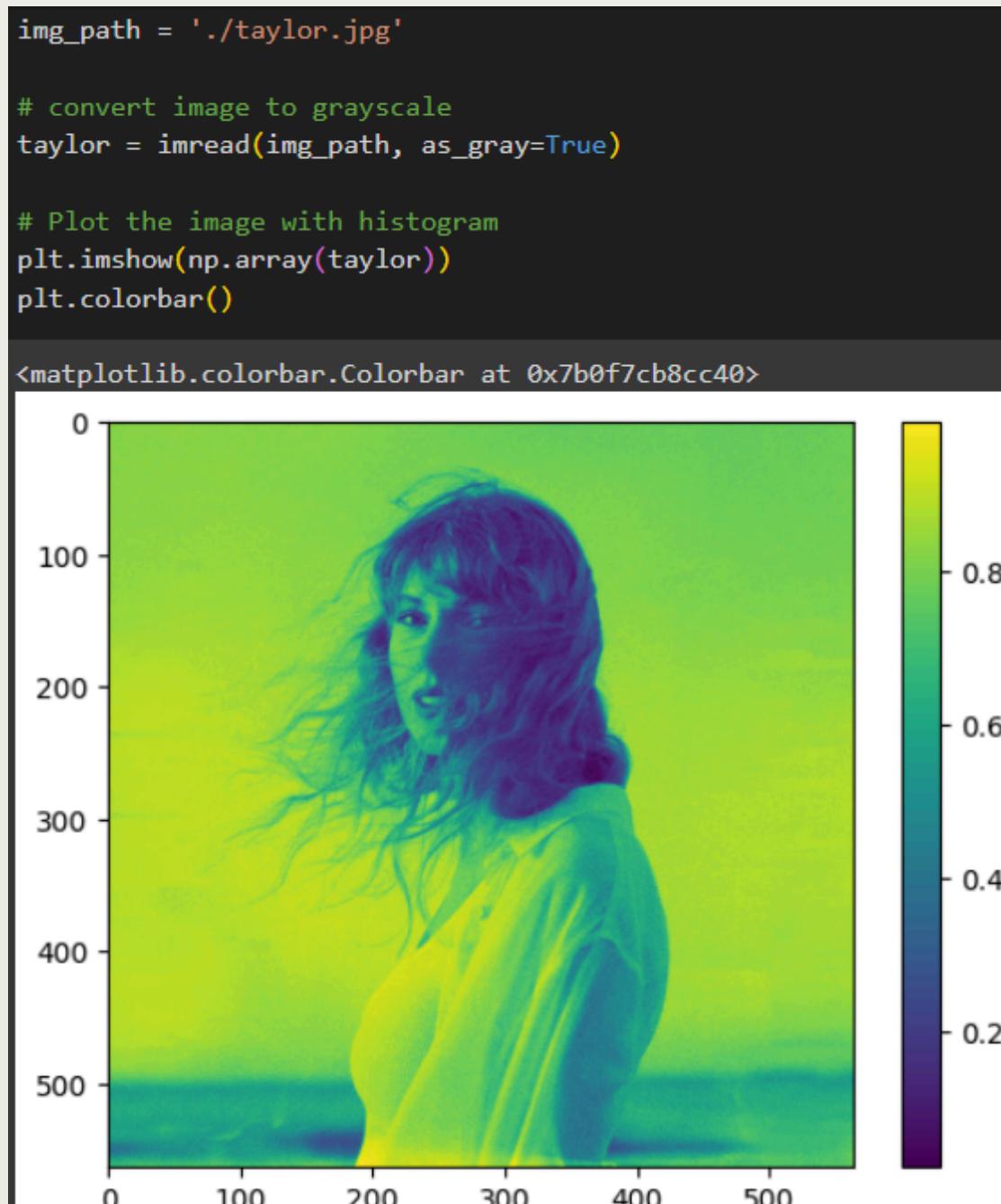
# *Image Filtering*

Image Processing

Faiqa Hadya Hanifa

4521210058

# Filtering



Dalam pemrosesan gambar, filter merupakan operasi matematika yang diterapkan pada gambar untuk mengubah tampilannya atau mengekstrak fitur tertentu. Filter dapat digunakan untuk menghaluskan, mempertajam, menyempurnakan tepian, menghilangkan noise, dan banyak lagi. Banyak jenis filter yang dapat digunakan dalam pemrosesan citra, termasuk filter linier, filter non-linier, dan filter morfologi. Filter linier menerapkan kombinasi linier piksel gambar masukan untuk menghasilkan gambar keluaran, sedangkan filter nonlinier menerapkan fungsi nonlinier pada piksel gambar masukan. Filter morfologi adalah jenis filter non-linier yang didasarkan pada bentuk atau struktur piksel gambar.

Modul SciPy menyediakan beberapa fungsi filter untuk penghalusan, penajaman, dan deteksi tepi, yang mana akan digunakan dalam tugas ini. Pertama, impor gambar yang akan digunakan dan definisikan dalam sebuah variabel.

# Mean Filter

Setelah menyiapkan matriks untuk mengolah citra, gambar yang difilter dibandingkan dengan gambar berpiksel, dapat dilihat bahwa hasil *mean filter* ini lebih halus dengan mengurangi *noise*, namun mengaburkan gambar.

```
# initialize the filter of size 5 by 5
# the filter is divided by 25 for normalization
k = np.ones((5,5))/25
print(k)

[[0.04 0.04 0.04 0.04 0.04]
 [0.04 0.04 0.04 0.04 0.04]
 [0.04 0.04 0.04 0.04 0.04]
 [0.04 0.04 0.04 0.04 0.04]
 [0.04 0.04 0.04 0.04 0.04]]

np.sum(k)

1.0
```

Tepi tajam yang hanya merupakan batas antara piksel gelap dan terang menjadi lebih halus karena piksel gelap mengurangi intensitas piksel di sekitarnya dan piksel terang melakukan hal sebaliknya.

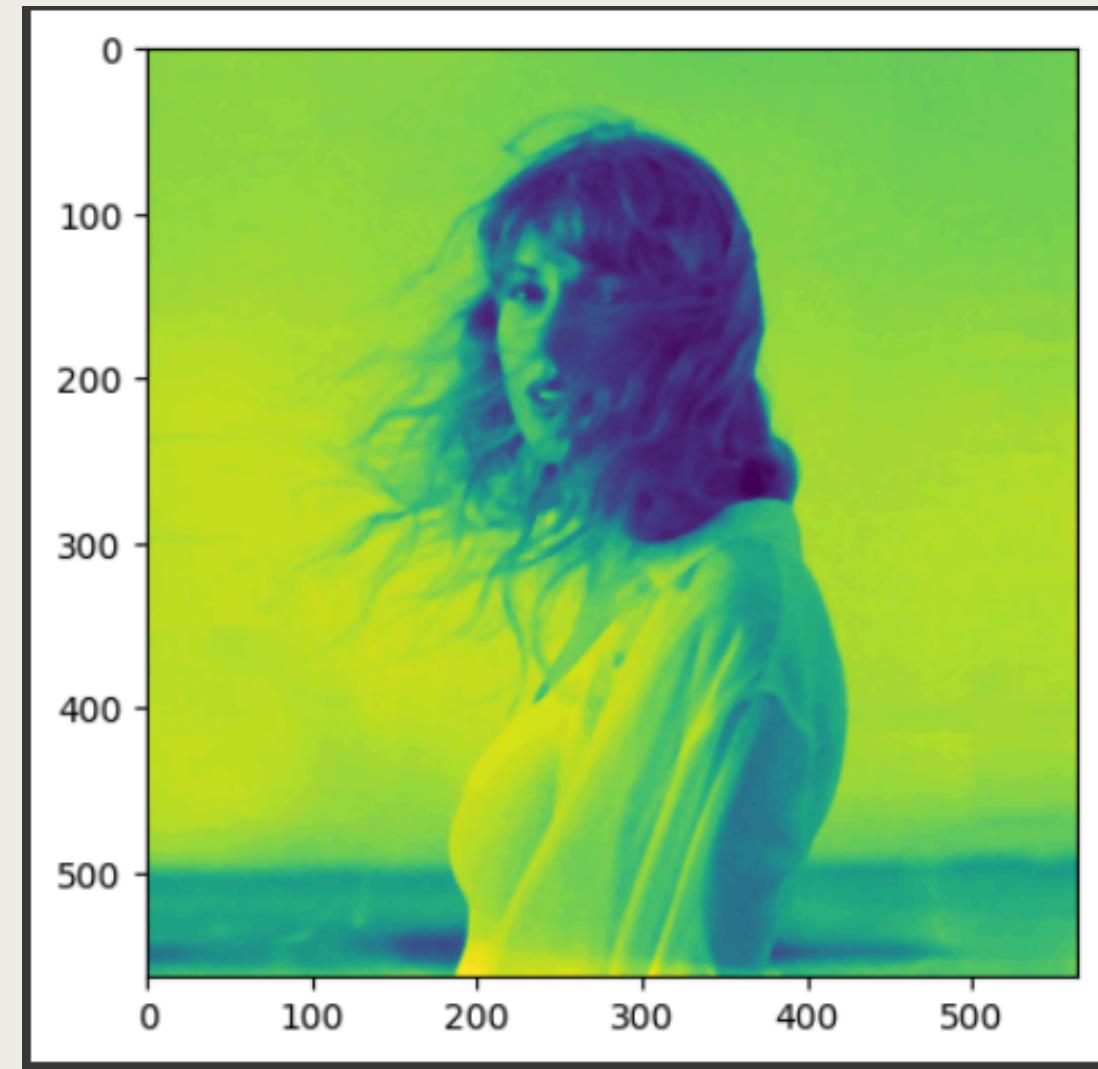


# *Median Filter*

```
# Median filter
med = scipy.ndimage.filters.median_filter(taylor, size=5, footprint=None, output=None, mode='reflect', cval=0.0, origin=0)

plt.imshow(med)
plt.show()

## Median filter is commonly used to remove noise and impulse or black and white spot noise
```



Filter median adalah filter klasik yang mempertahankan tepian. Filter ini mengambil sekumpulan piksel dan mengembalikan nilai median dalam kelompok yang tertutup oleh filter. Karena daerah di dekat tepi akan memiliki banyak nilai gelap dan banyak nilai terang (tetapi hanya sedikit nilai di antaranya), maka median pada suatu tepi kemungkinan besar akan berupa terang atau gelap, dan bukan nilai di antaranya. Dengan cara ini, bagian tepi yang halus tidak didapatkan, tapi mengurangi noise. Parameter size 5 akan mengaplikasikan ukuran 5x5 median filter. Mode dapat dispesifikasi sesuai keinginan bagaimana gambar akan diubah. Titik-titik hitam pada langit dan laut berkurang tanpa memburamkan gambar secara signifikan.

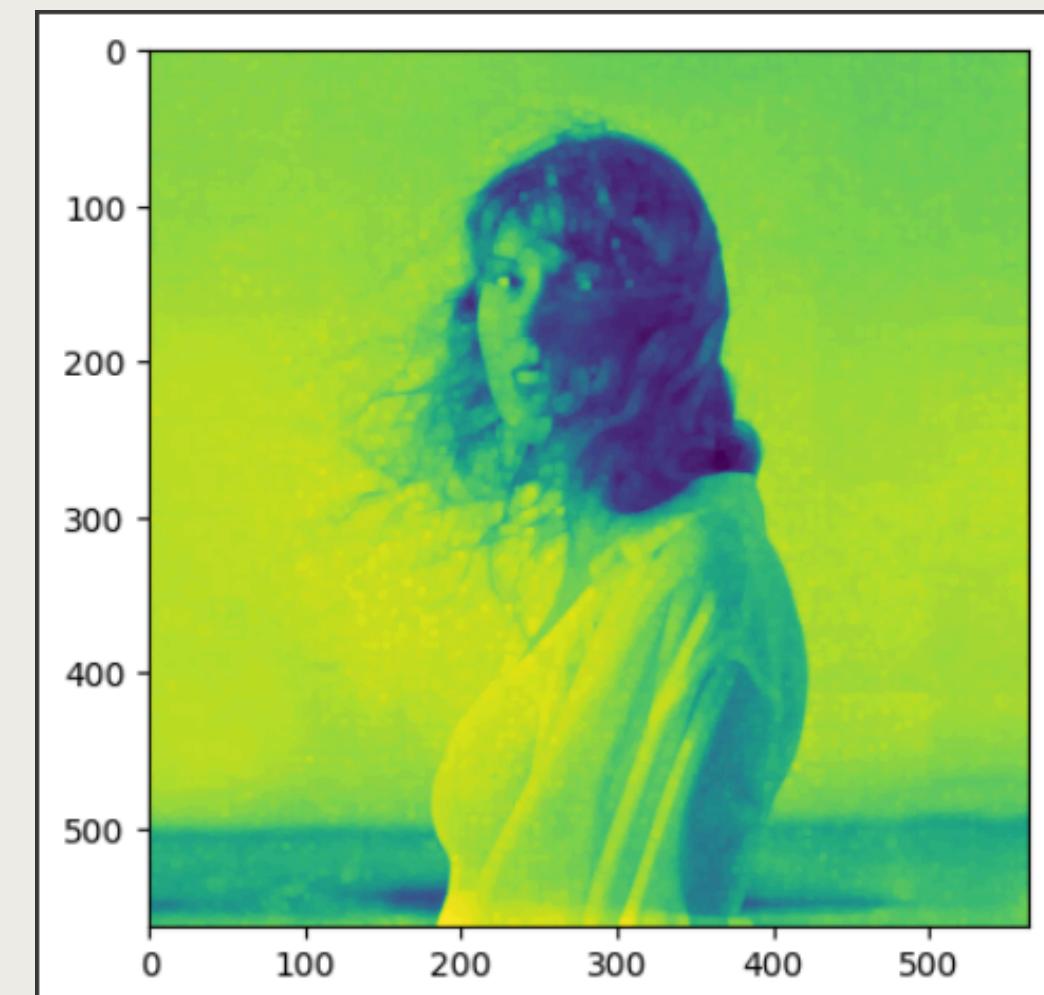
# Max Filter

```
# Max Filter
max = scipy.ndimage.filters.maximum_filter(taylor, size=5, footprint=None, output=None, mode='reflect', cval=0.0, origin=0)

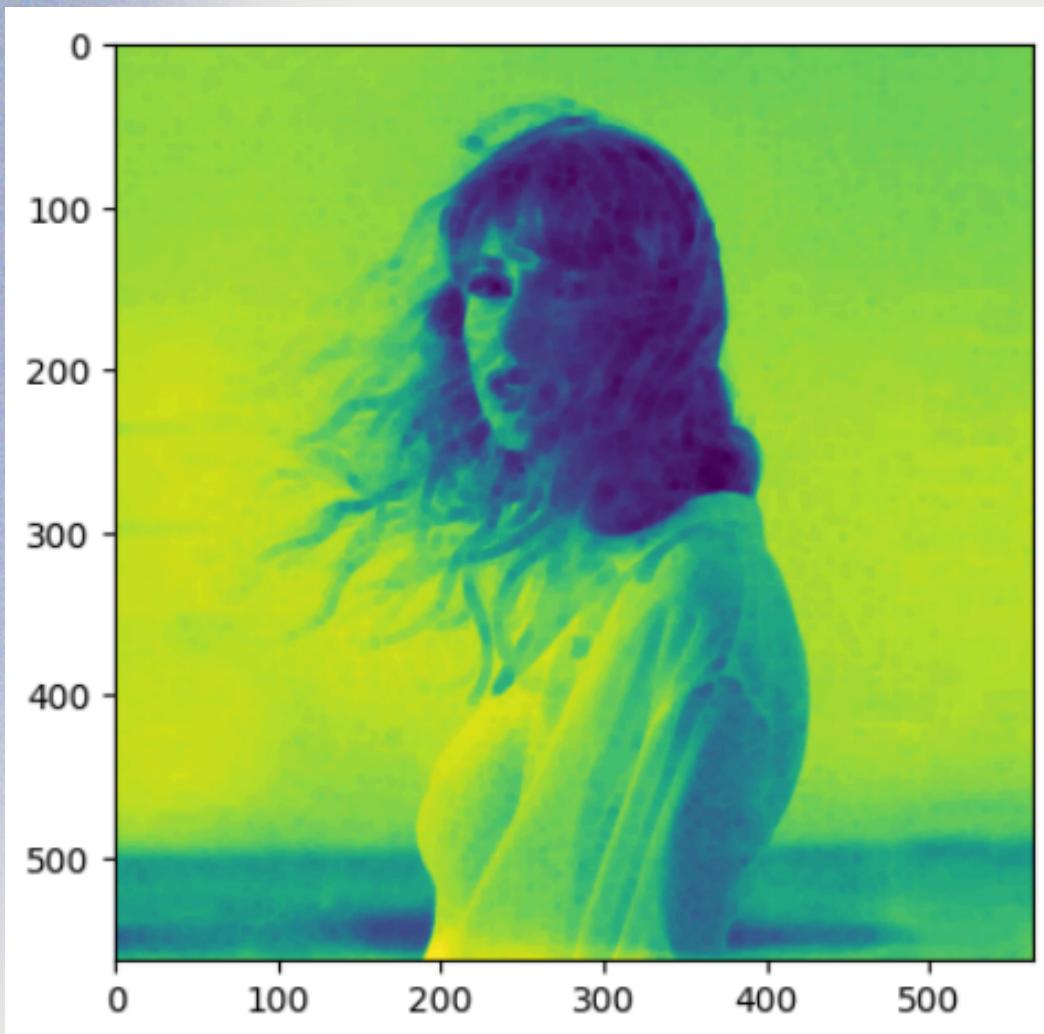
plt.imshow(max)
plt.show()

## Max filter enhances bright points, suitable for BW image
## the white pixels grew after applying the filter
```

Max filter merupakan jenis filter nonlinier yang menggantikan setiap nilai piksel dari gambar dengan nilai maksimum, yaitu nilai piksel paling terang di sekitar piksel yang sesuai pada gambar masukan. Dapat dilihat warna putih mendominasi pada area pupil, ujung helai rambut, dan baju.



# *Min Filter*



```
# Min Filter
min = scipy.ndimage.filters.minimum_filter(taylor, size=5, footprint=None, output=None, mode='reflect', cval=0.0, origin=0)

plt.imshow(min)
plt.show()

## This filter is used to enhance the darkest points in the image
```

Filter ini merupakan kebalikan dari Max Filter, dimana pixel akan digantikan dengan nilai minimum dari sebuah deret matriks yang berukuran sesuai dengan jendela filter.

Warna gelap pada gambar menegas, terutama pada area wajah dan rambut belakang.

# Edge Detection

# Sobel Filter

```
from skimage import filters, feature  
  
# Sobel Filter  
sob = filters.sobel(taylor)
```

Fungsi ini menerapkan filter Sobel pada gambar, yang merupakan jenis filter deteksi tepi untuk menyempurnakan tepi pada gambar dengan cara mengkonvolusi citra.

# Prewitt Filter

```
# Prewitt Filter  
prew = filters.prewitt(taylor)
```

Filter Prewitt mendeteksi tepi yang mirip dengan filter Sobel. Ini menghitung gradien intensitas gambar dan menyempurnakan tepi gambar dengan mengidentifikasi area dengan perubahan intensitas yang cepat.

# Canny Filter

```
# Canny Filter  
can = np.asarray(taylor).astype(float)  
  
np.min(can), np.max(can)  
  
can = can/float(np.max(can))  
can = feature.canny(can, sigma=1, low_threshold=0.05, high_threshold=0.24)
```

Canny memiliki kemampuan untuk menyaring noise sebelum melakukan perhitungan tepi sehingga tepi yang dihasilkan lebih banyak.

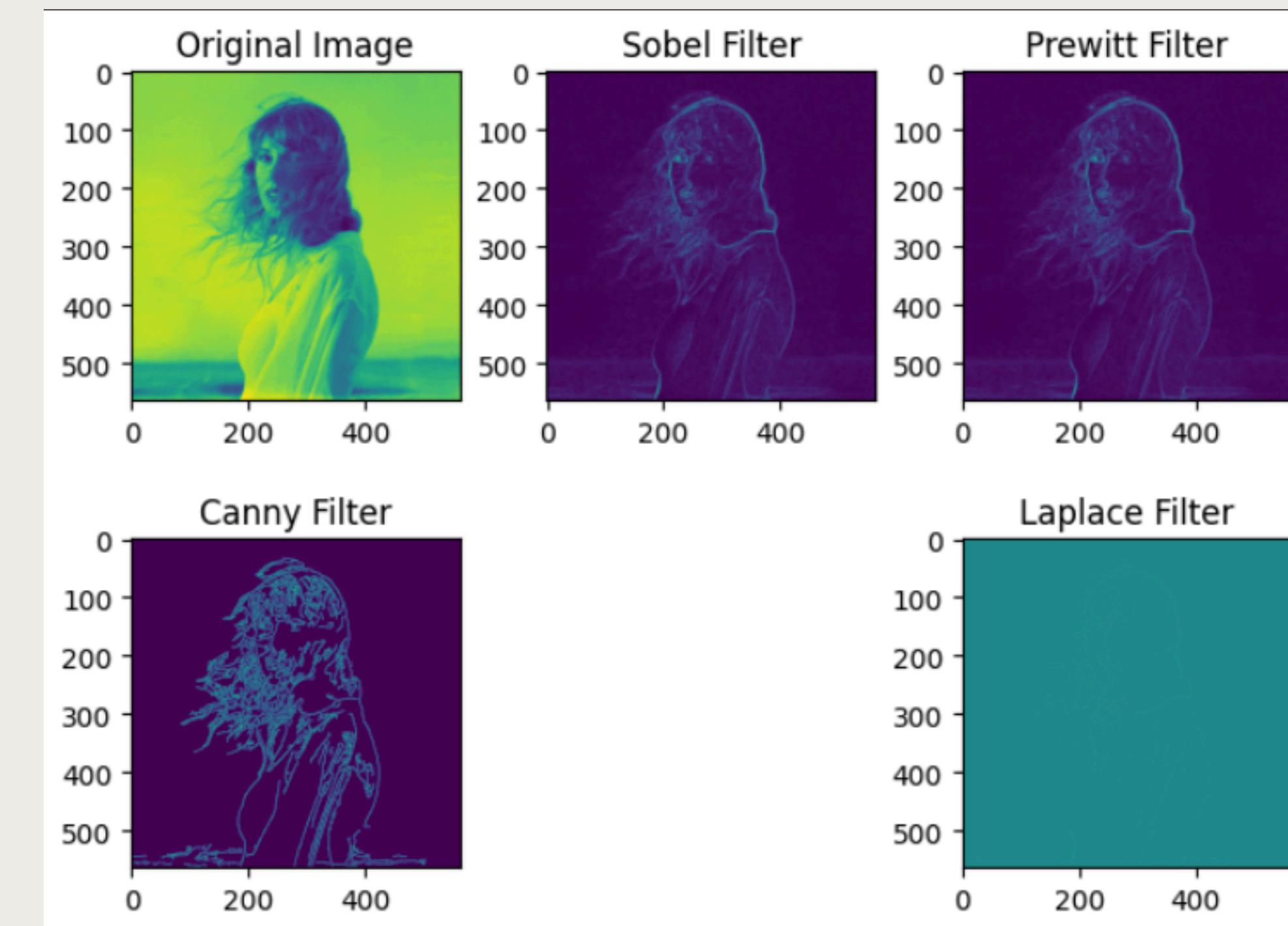
# Laplace Filter

```
# Laplace Filter  
lap = scipy.ndimage.filters.laplace(taylor)
```

Filter laplace berfungsi untuk menerapkan filter Laplacian, yaitu filter turunan orde kedua yang dapat digunakan untuk mendeteksi tepi dan sudut pada gambar.

Dengan menggunakan grafik dapat dilihat perbandingan bagaimana fungsi filter yang mengubah gambar satu sama lain.

Dibandingkan filter lainnya, Canny filter mendekksi tepian lebih banyak, jelas, dan tajam. Sedangkan metode laplacian tak dapat menangkap tepian dari objek-objek pada citra digital.



*Thank You!*

*Access the Full Notebook:*

[https://colab.research.google.com/drive/1PUPKQbquTzNrHpW-nztR4BDsEeVYIk\\_F?usp=sharing](https://colab.research.google.com/drive/1PUPKQbquTzNrHpW-nztR4BDsEeVYIk_F?usp=sharing)