Main

May 1, 2021

1 Kompresi gambar dengan SVD

Kita punya gambar kucing seperti dibawah, kita ingin menkrompesi gambar tersebut.

[4]:

```
import numpy as np
from matplotlib.image import imread
import matplotlib.pyplot as plt
import os

plt.rcParams['figure.figsize'] = [16,8]
plt.rcParams["savefig.format"] = 'png'

ogImageName = os.path.join(os.getcwd(),'images.jpg')
A = imread(ogImageName)
X = np.mean(A,-1) # Grayscale

img = plt.imshow(X);
img.set_cmap('gray')
plt.axis('off')
plt.show()
```

[5]:

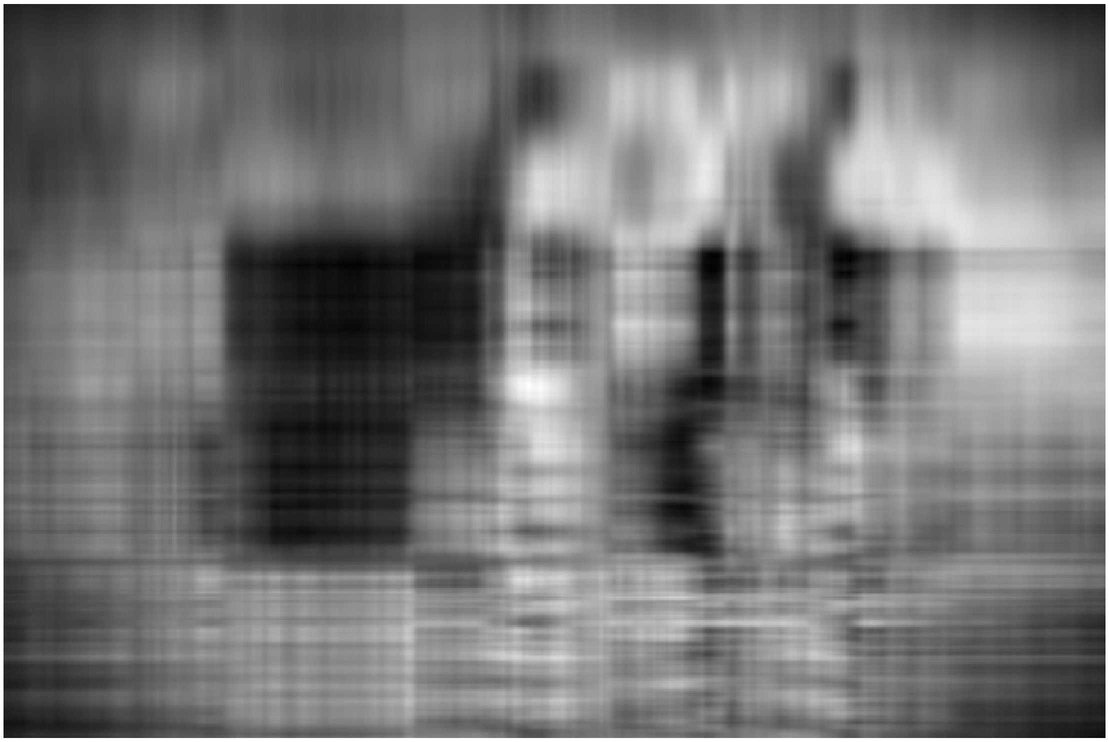
```
%%javascript
//Menjadikan output tidak perlu di scroll
IPython.OutputArea.prototype._should_scroll = function(lines) {
    return false;
}
```

```
<IPython.core.display.Javascript object>
```

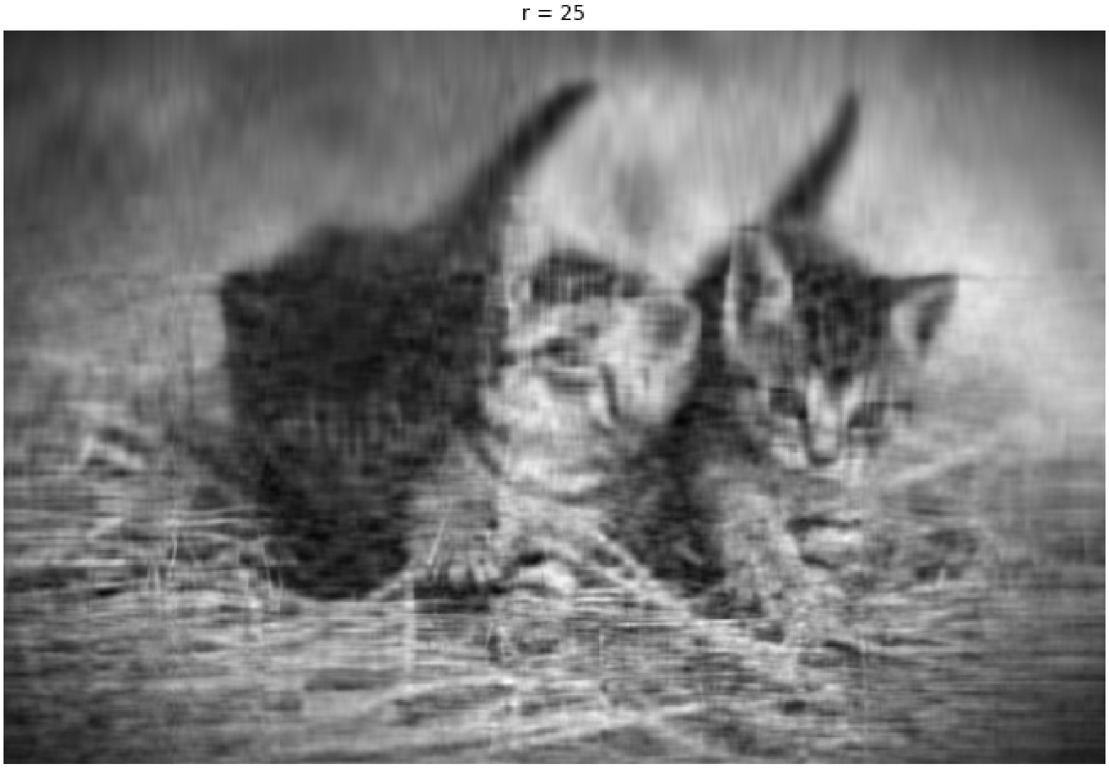
[6]:

```
from scipy.linalg import svd
from math import log,sqrt
# Menghitung SVD
U,L,A = svd(X, full_matrices=False)
L = np.diag(L)

m = X.shape[0]
n = X.shape[1]

j = 0
imageName = []
for r in (5,25,50,150,200):
    compressed = U[:,:r] @ L[0:r,:r] @ A[:r,:]
    plt.figure(j+1)
    j+=1
    img = plt.imshow(compressed)
    img.set_cmap('gray')
    ratio = (m*n)/(r*(m+n+1))
    mse = ((np.sum(X) - np.sum(compressed))**2)/(m*n)
    psnr = 10*log(255/sqrt(mse),10)
    plt.axis('off')
    plt.title('r = ' + str(r))
    plt.show()
    name = os.path.join(os.getcwd(),'compressed','images_compressed_r' + str(r)␣
 ↪+ ".jpg")
    plt.imsave(name,compressed, cmap='gray')
    imageName.append(name)
    print(f'''DATA FOR r = {r}
RATIO    = {ratio}
MSE      = {mse}
PSNR     = {psnr}
    ''')
```

r = 5

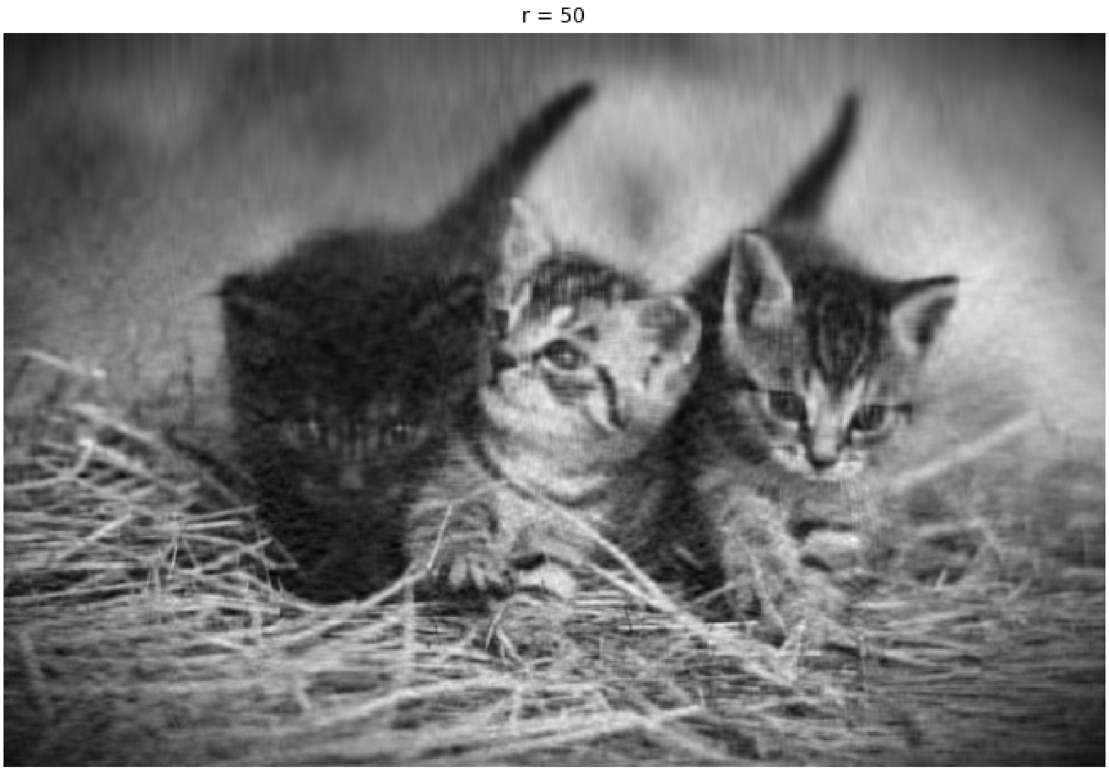


```
DATA FOR r = 5
RATIO    = 153.55201499531395
MSE      = 874.0448582450862
PSNR     = 9.357733192721454
```

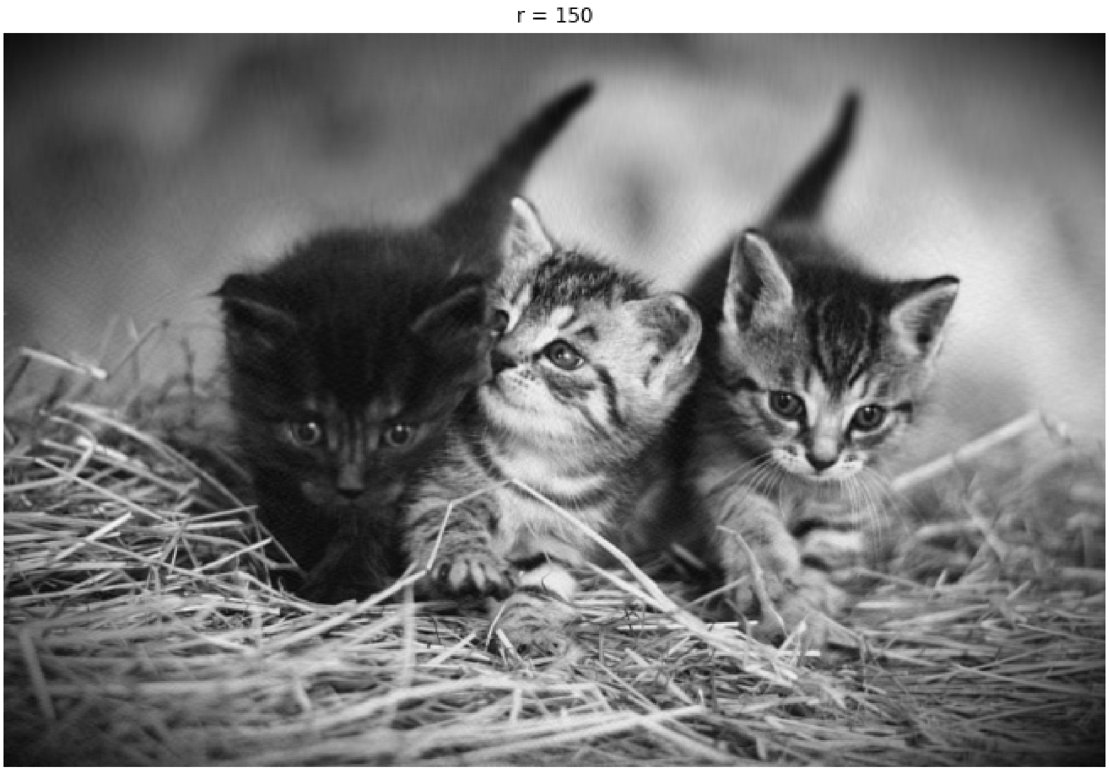


r = 25

```
DATA FOR r = 25
RATIO   = 30.710402999062794
MSE     = 32.765247088694096
PSNR    = 16.4883345673482
```



```
DATA FOR r = 50
RATIO   = 15.355201499531397
MSE     = 1.973766758301882
PSNR    = 22.588922652254286
```



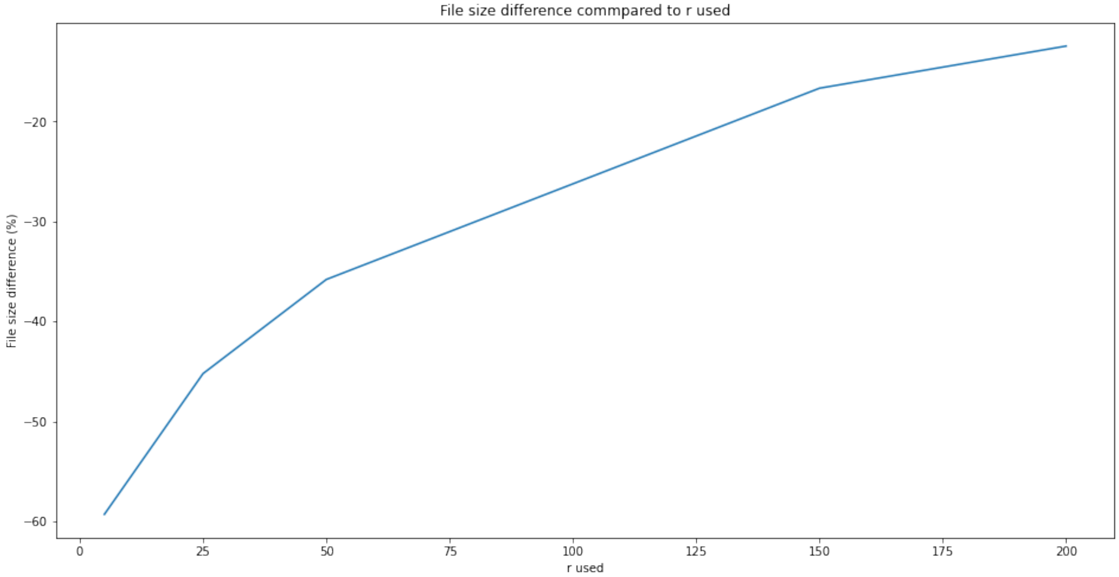
```
DATA FOR r = 150
RATIO   = 5.118400499843799
MSE     = 0.0010953666441862863
PSNR    = 38.867604245469195
```

```
DATA FOR r = 200
RATIO    = 3.838800374882849
MSE      = 1.3739513939304768e-05
PSNR     = 48.37554495921231
```

[7]:

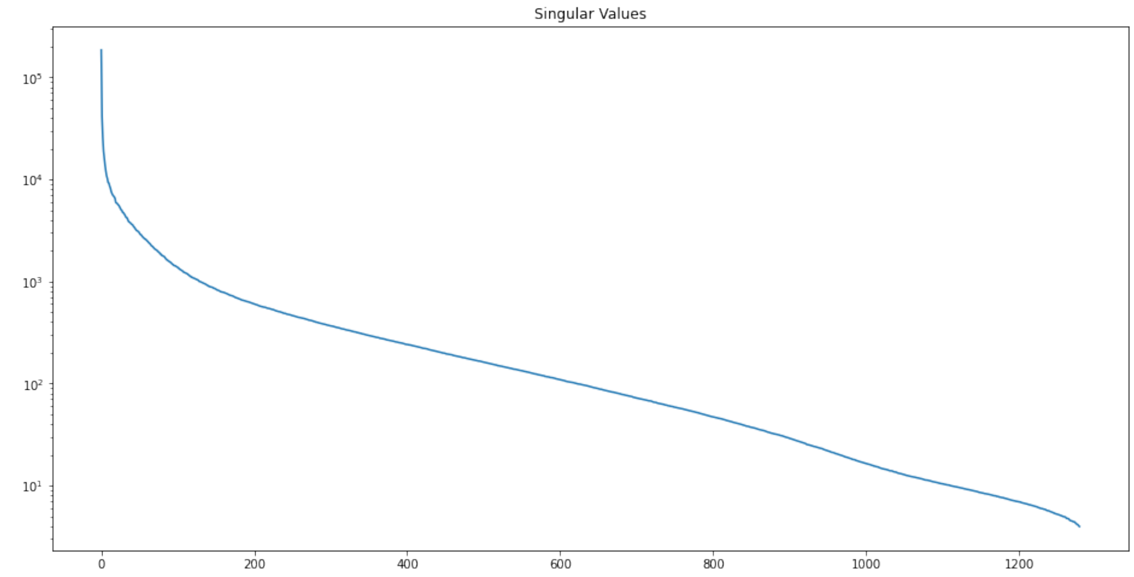
```python
#File Byte Analysis

ogSize = os.stat(ogImageName).st_size

cmpSizes = []
cmpSizesDiff = []
print(f"original = {ogSize/1000:.2f} kb")
for name,r in zip(imageName,[5,25,50,150,200]):
    cmpSize = os.stat(name).st_size
    cmpSizes.append(cmpSize/1000)
    cmpSizesDiff.append((cmpSize-ogSize)/ogSize *100)
    print(f'compressed(r={r}) = {cmpSize/1000:.2f} kb , % diff = {(cmpSize-ogSize)/ogSize *100:.2f}%')

plt.plot([5,25,50,150,200],cmpSizesDiff)
plt.title('File size difference commpared to r used')
plt.ylabel('File size difference (%)')
plt.xlabel('r used')
plt.show()
```

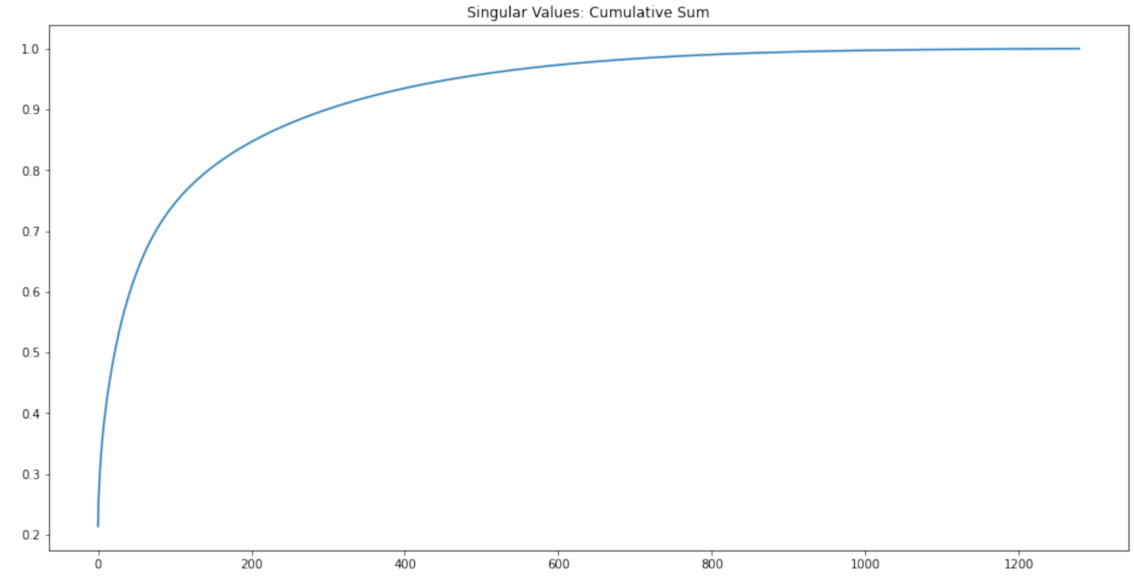
```
original = 294.33 kb
compressed(r=5) = 119.86 kb , % diff = -59.28%
compressed(r=25) = 161.25 kb , % diff = -45.21%
compressed(r=50) = 188.95 kb , % diff = -35.80%
compressed(r=150) = 245.22 kb , % diff = -16.69%
compressed(r=200) = 257.62 kb , % diff = -12.47%
```



[8]:

```
plt.figure(1)
plt.semilogy(np.diag(L))
plt.title('Singular Values')
plt.show()

plt.figure(2)
plt.plot(np.cumsum(np.diag(L))/np.sum(np.diag(L)))
plt.title('Singular Values: Cumulative Sum')
plt.show()
```


Singular Values
10^5
10^4
10^3
10^2
10^1
0
200
400
600
800
1000
1200


Singular Values: Cumulative Sum
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
200
400
600
800
1000
1200