LGMVIP*DS* October_23_Task_Number_1-4

Image to Pencil Sketch

By Mouli Nahal

In [30]:
```
import numpy as np
import pandas as pd

import os
```

Importing Libraries

In [3]:
```
import numpy as np
import pandas as pd
import PIL
from PIL import Image, ImageChops, ImageFilter
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
import cv2
from IPython.display import display, Image as ipimage
```

Using PIL library

In [5]:
```
# Opening color image using Image.open
img=Image.open('D:\LetsGrowMore\puppy.jpg')
img_resize=img.resize((400,500))
img_resize
```

Out[5]:

In [6]:
```
# Converting Color image to Gray Image using 'convert('L')'
gray_img=img_resize.convert('L')
gray_img
```

Out[6]:

In [7]:
```
# Inverting Image means substracting individual pixel values from 255,
# Here it is done using ImageChops.invert() function from PIL library
inverted_img=ImageChops.invert(gray_img)
inverted_img
```

Out[7]:

In [8]:
```
# Using lambda and eval function, every pixel value is substracted from 255 to get
inverted_img1=Image.eval(gray_img,lambda x:255-x)
inverted_img1
```

Out[8]:

In [9]:
```
np.array(inverted_img1)
```

Out[9]:
```
array([[ 81,  81,  81, ...,  86,  86,  86],
       [ 81,  81,  81, ...,  86,  86,  86],
       [ 81,  81,  81, ...,  85,  85,  85],
       ...,
       [146, 148, 147, ..., 137, 139, 139],
       [147, 148, 148, ..., 137, 138, 137],
       [148, 149, 149, ..., 137, 136, 137]], dtype=uint8)
```

In [10]:
```
# Applying Blurring to image using 'GaussianBlur' ImageFilter
inverted_img_blurred=inverted_img.filter(ImageFilter.GaussianBlur(radius=5))
inverted_img_blurred
```

Out[10]:

In [11]:
```
np.array(inverted_img_blurred)
```

Out[11]:
```
array([[ 81,  81,  81, ...,  86,  86,  86],
       [ 81,  81,  81, ...,  86,  86,  86],
       [ 81,  81,  81, ...,  85,  85,  85],
       ...,
       [142, 142, 141, ..., 139, 139, 140],
       [143, 142, 141, ..., 138, 138, 139],
       [143, 143, 142, ..., 138, 138, 139]], dtype=uint8)
```

In [12]:
```
#dividing Gray Image Pixel Values by Inverte Image Pixel Values
pencil_sketch=np.array(gray_img)/np.array(inverted_img_blurred)
pencil_sketch
```

Out[12]:
```
array([[2.14814815, 2.14814815, 2.14814815, ..., 1.96511628, 1.96511628,
        1.96511628],
       [2.14814815, 2.14814815, 2.14814815, ..., 1.96511628, 1.96511628,
        1.96511628],
       [2.14814815, 2.14814815, 2.14814815, ..., 2.        , 2.        ,
        2.        ],
       ...,
       [0.76760563, 0.75352113, 0.76595745, ..., 0.84892086, 0.83453237,
        0.82857143],
       [0.75524476, 0.75352113, 0.75886525, ..., 0.85507246, 0.84782609,
        0.84892086],
       [0.74825175, 0.74125874, 0.74647887, ..., 0.85507246, 0.86231884,
        0.84892086]])
```

In [13]:
```
#Converting floating pixel values to 'uint8' data type so as to be compatible for p
pencil_sketch_uint=pencil_sketch.astype('uint8')
pencil_sketch_uint
```

Out[13]:
```
array([[2, 2, 2, ..., 1, 1, 1],
       [2, 2, 2, ..., 1, 1, 1],
       [2, 2, 2, ..., 2, 2, 2],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=uint8)
```

In [14]:
```
# Scaling Image by multiplying individual pixel values with 255 to get pencil sketc
Image.fromarray(255*pencil_sketch_uint)
```

Out[14]:

Using CV2 Library

In [15]:
```
img_cv=cv2.imread('D:\LetsGrowMore\puppy.jpg')
img_cv
```

Out[15]:
```
array([[[123, 188, 166],
        [123, 188, 166],
        [123, 188, 166],
        ...,
        [115, 182, 161],
        [115, 182, 161],
        [115, 182, 161]],

       [[123, 188, 166],
        [123, 188, 166],
        [123, 188, 166],
        ...,
        [116, 183, 162],
        [116, 183, 162],
        [116, 183, 162]],

       [[123, 188, 166],
        [123, 188, 166],
        [123, 188, 166],
        ...,
        [116, 183, 162],
        [116, 183, 162],
        [116, 183, 162]],

       ...,

       [[ 54, 122,  99],
        [ 54, 122,  99],
        [ 54, 122,  99],
        ...,
        [ 72, 126, 121],
        [ 72, 126, 121],
        [ 72, 126, 121]],

       [[ 54, 122,  99],
        [ 54, 122,  99],
        [ 54, 122,  99],
        ...,
        [ 74, 126, 119],
        [ 74, 125, 121],
        [ 74, 125, 121]],

       [[ 52, 122,  99],
        [ 52, 122,  99],
        [ 51, 121,  98],
        ...,
        [ 75, 127, 120],
        [ 75, 127, 120],
        [ 74, 126, 119]]], dtype=uint8)
```

In [16]:
```
gray_img_cv=cv2.cvtColor(img_cv,cv2.COLOR_BGR2GRAY)
gray_img_cv
```

Out[16]:
```
array([[174, 174, 174, ..., 168, 168, 168],
       [174, 174, 174, ..., 169, 169, 169],
       [174, 174, 174, ..., 169, 169, 169],
       ...,
       [107, 107, 107, ..., 118, 118, 118],
       [107, 107, 107, ..., 118, 118, 118],
       [107, 107, 106, ..., 119, 119, 118]], dtype=uint8)
```

In [17]:
```
inverted_imgcv=255-gray_img_cv
inverted_imgcv
```

Out[17]:
```
array([[ 81,  81,  81, ...,  87,  87,  87],
       [ 81,  81,  81, ...,  86,  86,  86],
       [ 81,  81,  81, ...,  86,  86,  86],
       ...,
       [148, 148, 148, ..., 137, 137, 137],
       [148, 148, 148, ..., 137, 137, 137],
       [148, 148, 149, ..., 136, 136, 137]], dtype=uint8)
```

In [18]:
```
pencil_sketchcv=gray_img_cv/inverted_imgcv
pencil_sketchcv
```

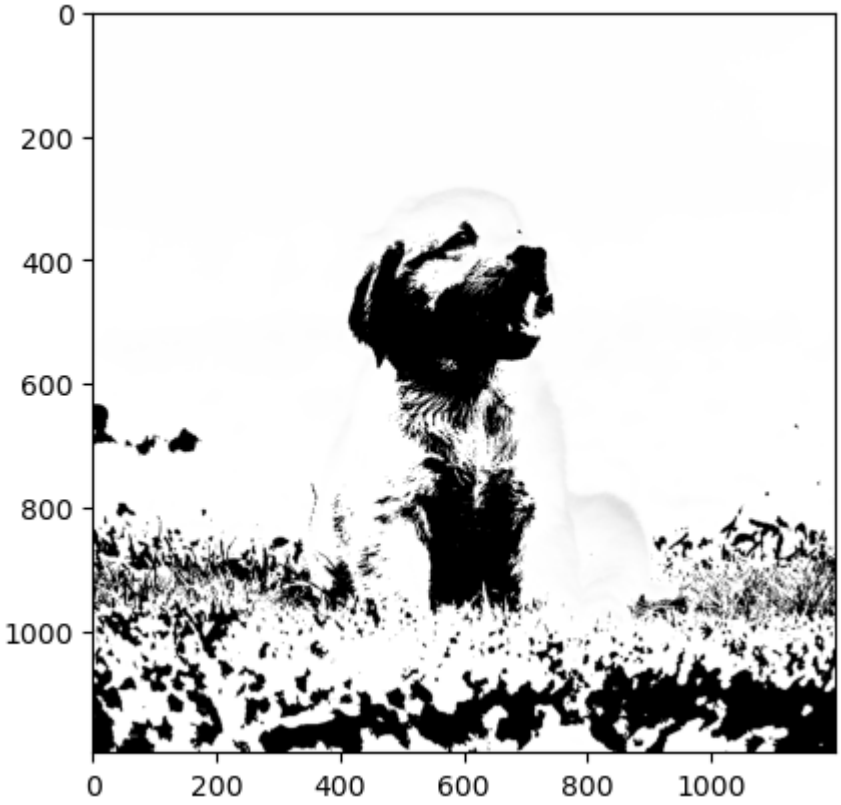
Out[18]:
```
array([[2.14814815, 2.14814815, 2.14814815, ..., 1.93103448, 1.93103448,
        1.93103448],
       [2.14814815, 2.14814815, 2.14814815, ..., 1.96511628, 1.96511628,
        1.96511628],
       [2.14814815, 2.14814815, 2.14814815, ..., 1.96511628, 1.96511628,
        1.96511628],
       ...,
       [0.72297297, 0.72297297, 0.72297297, ..., 0.86131387, 0.86131387,
        0.86131387],
       [0.72297297, 0.72297297, 0.72297297, ..., 0.86131387, 0.86131387,
        0.86131387],
       [0.72297297, 0.72297297, 0.7114094 , ..., 0.875     , 0.875     ,
        0.86131387]])
```

In [19]:
```
pencil_sketchcv_uint=pencil_sketchcv.astype('uint8')
pencil_sketchcv_uint
```

Out[19]:
```
array([[2, 2, 2, ..., 1, 1, 1],
       [2, 2, 2, ..., 1, 1, 1],
       [2, 2, 2, ..., 1, 1, 1],
       ...,
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0],
       [0, 0, 0, ..., 0, 0, 0]], dtype=uint8)
```

In [20]:
```
plt.imshow(pencil_sketchcv_uint*255, cmap='gray')
```

Out[20]:
```
<matplotlib.image.AxesImage at 0x1d583b5f670>
```



In [ ]:

In [ ]: