

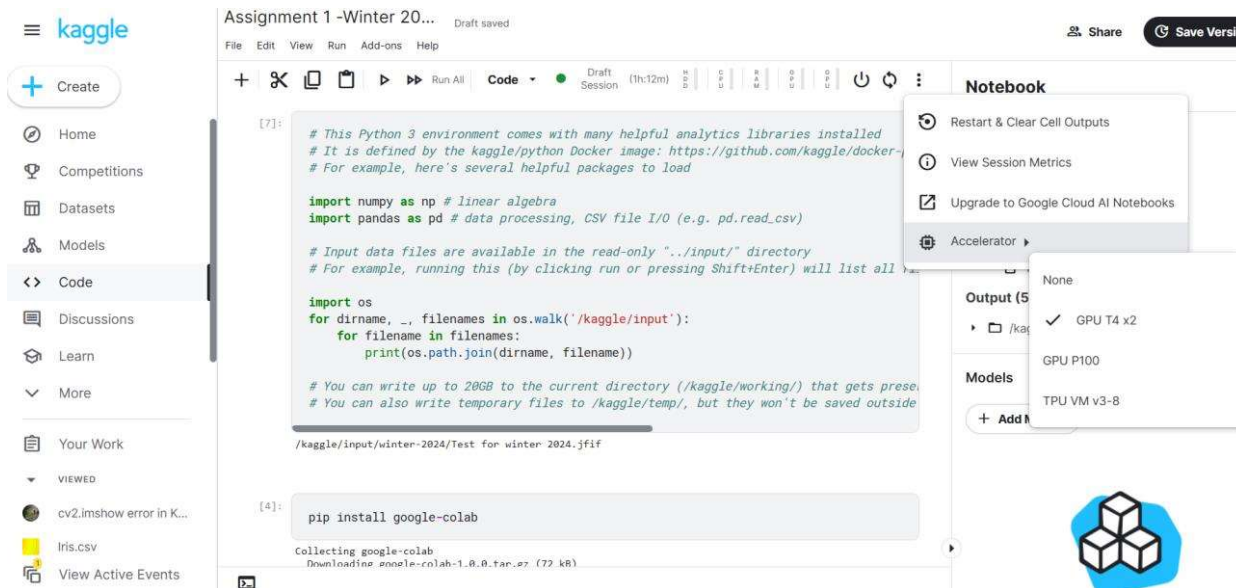
Assignment 1 –Key to completer by Kaggle

28-01-2024

Partial key for Module-1-Lab-processing-images

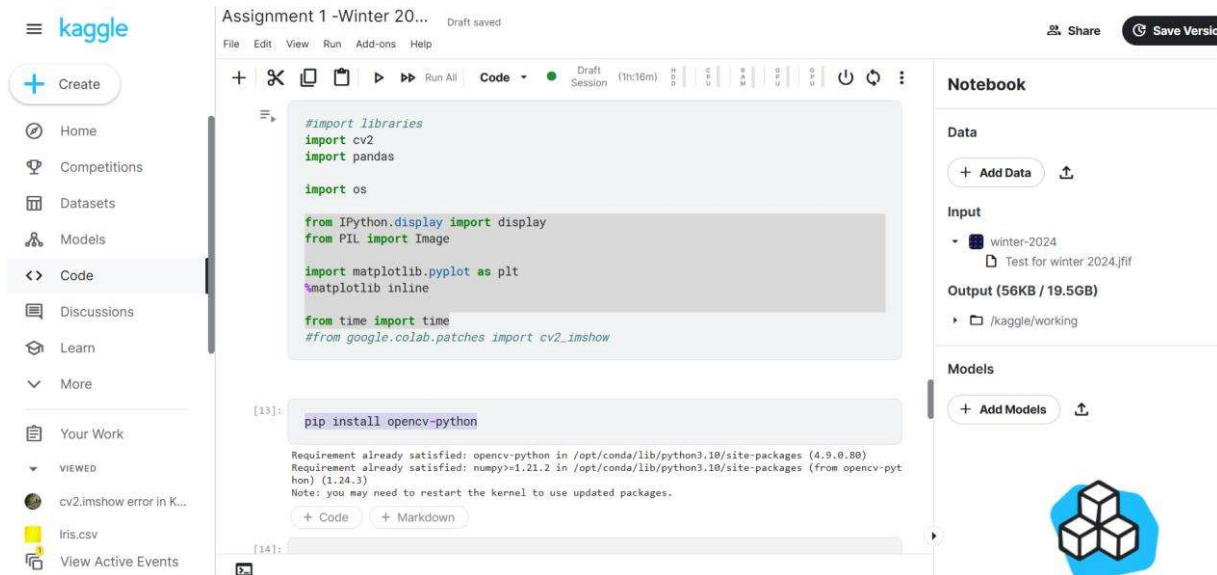
- Just to remind you that must experiencing many errors while using Kaggle/Colab
- Errors to me are not normal as I have already done with no issues in previous semesters.
- Images are on order as is in my notebook

However, here is all I have done to complete the most part of this section:



1- Make sure you are using GPU 4 *2

2- Make sure you run “pip install google-colab”



3- Run “pip install opencv-python”

Running CPU

Assignment 1-Winter 20...

Draft saved

FileEditViewRunAdd-onsHelp

+K[]⌵⏮⏪⏩⏭Run⌘Code+

✔ Draft Session (Python)


123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100

123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100

#!time
#Display an image locally in a popup window using cv2.imshow
img = cv2.imread('/kaggle/input/winter-2024/Test for winter 2024 .jpg')
cv2.imshow('my_image', img)
cv2.waitKey(0)
cv2.destroyAllWindows()

plt.imshow(img)
plt.title('My Image')
plt.show()

My Image




CPU times: user 275 ms, sys: 148 ms, total: 423 ms
Wall time: 275 ms

+ Code+ Markdown

[?]#!time
#Display an image using ipython display
#Import libraries
import cv2

import numpy as np
import os

from IPython.display import display
from PIL import Image
img = cv2.imread('/kaggle/input/winter-2024/Test for winter 2024 .jpg')
display(Image.fromarray(img))



CPU times: user 53.0 ms, sys: 3.4 ms, total: 56.3 ms
Wall time: 51.1 ms

Notebook

Data

+ Add Data ⌵

Input


winter-2024

Output (50KB / 19.5GB)

/kaggle/working

Models

+ Add Models ⌵



No models added
Add a Kaggle model

Notebook options

Schedule a notebook to run

Code Help

Find code help

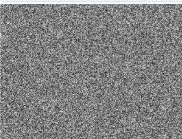
Assignment 1-Winter 20... Draft saved

File Edit View Run Add-ons Help

+
+
+
+
+

[7x]


```
Write  
import cv2  
  
import numpy as np  
import os  
# Make an array of 128,000 random bytes.  
randomByteArray = bytearray(os.urandom(128000))  
flatNumpyArray = np.array(randomByteArray)  
  
# Convert the array to make a 400x400 grayscale image.  
grayImage = flatNumpyArray.reshape((400, 400))  
# cv2.imwrite('images/RandOmGray.png', grayImage)  
# grayImage2 = cv2.imread('images/RandOmGray.png')  
  
display(Image.fromarray(grayImage))
```



CPU times: user 15.8 ms, sys: 2.15 ms, total: 15.7 ms
Wall time: 15.8 ms

[7x]

```
Write  
# Convert the array to make a 400x400 color image and write to file.  
bgImage = flatNumpyArray.reshape((400, 400, 3))  
img = cv2.cvtColor('images/RandOmGray.png', bgImage)  
#read local image  
img2 = cv2.imread('images/RandOmColor.png') # with the OpenCV function imread(), the order of colors is BGR (blue, green, red).  
display(Image.fromarray(imgImage))
```



CPU times: user 15.7 ms, sys: 2.15 ms, total: 15.8 ms
Wall time: 15.8 ms

Share Save Version 0

Notebook

Data

- + Add Data ↕

Input


- Winter-2020

Output (5KB / 10,50B)

- ⌵ Aggregate viewing

Models

- + Add Models ⬇



No models added
Add a Kaggle model

Notebook options

- Schedule a notebook to run

Code Help

- 🔍 Find code help

Assignment 1-Winter 20... Draft saved

File Edit View Run Add-ons Help

Code

```

[24]:
%%time
original = cv2.imread('/kaggle/input/winter-2024/test for winter 2024.jpg')
img = original.copy()
mask = np.zeros(img.shape[0], np.uint8)

bgModel = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
fgModel = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

rect = (100, 1, 421, 378)
cv2.grabCut(img, mask, rect, bgModel, fgModel, 5, cv2.GC_INIT_WITH_RECT)

mask2 = np.where((mask==2) | (mask==0), 0, 1).astype('uint8')
img = img[mask2==1]

plt.subplot(121)
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('grabcut')
plt.axis('off')
plt.show()

plt.subplot(122)
plt.imshow(cv2.cvtColor(original, cv2.COLOR_BGR2RGB))
plt.title('original')
plt.axis('off')
plt.show()

```

grabcut original

CPU times: user 1.41 s, sys: 114 ms, total: 1.52 s
Wall time: 1.11 s

```

[25]:
%%time
plt.subplot(121)
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('grabcut')
plt.axis('off')

plt.subplot(122)
plt.imshow(cv2.cvtColor(original, cv2.COLOR_BGR2RGB))
plt.title('original')
plt.axis('off')
plt.show()

```

grabcut original

CPU times: user 1.41 s, sys: 114 ms, total: 1.52 s
Wall time: 1.11 s

Notebook

Data

+ Add Data

Input

winter-2024

Output (56KB / 19.5GB)

/kaggle/working

Models

+ Add Models

No models added
Add a Kaggle model

Notebook options

Schedule a notebook to run

Code Help

Find code help

Assignment 1-Winter 20... Draft saved

File Edit View Run Add-ons Help

Code

```

[21]:
%%time
plt.subplot(121)
plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
plt.title('grabcut')

plt.subplot(122)
plt.imshow(cv2.cvtColor(original, cv2.COLOR_BGR2RGB))
plt.title('original')

plt.show()

```

grabcut original

CPU times: user 513 ms, sys: 169 ms, total: 681 ms
Wall time: 484 ms

+ Code + Markdown

```

[22]:
pip install ipywidgets

```

Collecting ipywidgets
Downloading ipywidgets-8.1.1-py3-none-any.whl.metadata (2.4 kB)
Requirement already satisfied: comm>=0.1.3 in /usr/local/lib/python3.10/site-packages (from ipywidgets) (0.2.1)
Requirement already satisfied: ipython>=6.1.0 in /usr/local/lib/python3.10/site-packages (from ipywidgets) (8.20.0)
Requirement already satisfied: traitlets>=4.3.1 in /usr/local/lib/python3.10/site-packages (from ipywidgets) (5.14.1)
Collecting widgetsnbextension<4.0.0, >=3.0.0 in /usr/local/lib/python3.10/site-packages (from ipywidgets) (3.6.4)
Collecting jupyterlab-widgets<3.0.0, >=2.0.0 in /usr/local/lib/python3.10/site-packages (from ipywidgets) (2.0.0)
Requirement already satisfied: decorator in /usr/local/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (5.1.1)
Requirement already satisfied: jedi<=0.16 in /usr/local/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (0.19.1)
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (0.1.6)
Requirement already satisfied: prompt-toolkit<3.1.0, >=3.0.42 in /usr/local/lib/python3.10/site-packages (from ipython>=6.1.0->ipywidgets) (3.0.43)
Requirement already satisfied: parso<0.8.0, >=0.7.0 in /usr/local/lib/python3.10/site-packages (from jedi<=0.16->ipywidgets) (0.8.3)
Requirement already satisfied: pexpect in /usr/local/lib/python3.10/site-packages (from prompt-toolkit<3.1.0, >=3.0.42->ipywidgets) (4.9.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/site-packages (from prompt-toolkit<3.1.0, >=3.0.42->ipywidgets) (0.2.13)
Requirement already satisfied: typing-extensions in /usr/local/lib/python3.10/site-packages (from traitlets>=4.3.1->ipywidgets) (4.11.0)
Installing collected packages: ipywidgets, widgetsnbextension, jupyterlab-widgets
Successfully installed ipywidgets-8.1.1 widgetsnbextension-3.6.4 jupyterlab-widgets-2.0.0

Notebook

Data

+ Add Data

Input

winter-2024

Output (56KB / 19.5GB)

/kaggle/working

Models

+ Add Models

No models added
Add a Kaggle model

Notebook options

Schedule a notebook to run

4- You may need to run “ipywidgets”

[illegible]

5- Run "pip install opencv-contrib-python"

6- You may need to run “pip install --upgrade pip”

Running GPU

```
!pip install pycuda # install cuda
import pycuda.driver as cuda
import pycuda.autoinit
from pycuda.compiler import SourceModule
```

- 7- Make sure you are running following in one cell:
- !pip install pycuda # install cuda
 - import pycuda.driver as cuda
 - import pycuda.autoinit
 - from pycuda.compiler import SourceModule

```
%%time
# -*- coding: utf-8 -*-
Created on Mon Aug 13 12:15:39 2018

@author: bhaumik

#pass c code to perform work on GPU

mod = SourceModule \
(
    """
#include<stdio.h>
#define INDEX(a, b) a*256+b

__global__ void bgr2gray(float *d_result, float *b_img, float *g_img, float *r_img)
{
    unsigned int idx = threadIdx.x+(blockIdx.x*(blockDim.x*blockDim.y));

    unsigned int a = idx/256;
    unsigned int b = idx%256;
    d_result[INDEX(a, b)] = (0.299*r_img[INDEX(a, b)]+0.587*g_img[INDEX(a, b)]+0.114*b_img[INDEX(a, b)]);

}

    """
)
```

CPU times: user 9.14 ms, sys: 2.72 ms, total: 11.9 ms
Wall time: 137 ms

```

[68]: %time
import numpy as np
import ipywidgets
from IPython.display import display
from IPython.display import display
from PIL import Image

import matplotlib.pyplot as plt
%matplotlib inline

from time import time
#convert image to jpg for display
def bgr8_to_jpeg(value, quality=75):
    return bytes(cv2.imencode('.jpg', value)[1])

h_img = cv2.imread('/kaggle/input/winter-2024/Test for winter 2024.jfif',1)
print(h_img.shape)

h_result=r_img

h_gray=cv2.cvtColor(h_img,cv2.COLOR_BGR2GRAY)
#print a
b_img = h_img[:, :, 0].reshape(119448).astype(np.float32)
g_img = h_img[:, :, 1].reshape(119448).astype(np.float32)
r_img = h_img[:, :, 2].reshape(119448).astype(np.float32)
h_result=r_img
bgr2gray = mod.get_function("bgr2gray")

#create instance of cuda function
#bgr2gray = mod.get_function("bgr2gray")
print(h_result.shape)

#call pycuda function
bgr2gray(cuda.Out(h_result),cuda.In(b_img),cuda.In(g_img),cuda.In(r_img),block=(1024, 1, 1),grid=(64, 1, 1))
h_result=np.reshape(h_result,(252,474)).astype(np.uint8)
#using Cv2
#cv2.imshow('Grayscale Image',h_result)
#cv2.waitKey(0)
#cv2.destroyAllWindows()

#using plt
plt.imshow(h_result)
plt.title('My Image')
plt.show()
#display transformed image and using widget
image_widget = ipywidgets.Image(format='jpeg')
image_widget.value = bgr8_to_jpeg(h_result)
display(image_widget)
#original image
orgimage_widget = ipywidgets.Image(format='jpeg')
orgimage_widget.value = bgr8_to_jpeg(h_img)
display(orgimage_widget)

```

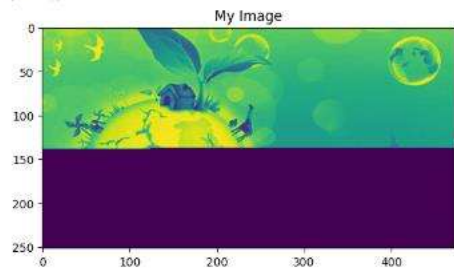
```

(252, 474, 3)
(119448,)

```

```
#using plt
plt.imshow(h_result)
plt.title('My Image')
plt.show()
#display transformed image and using widget
image_widget = ipywidgets.Image(format='jpeg')
image_widget.value = bgr8_to_jpeg(h_result)
display(image_widget)
#original image
orgimage_widget = ipywidgets.Image(format='jpeg')
orgimage_widget.value = bgr8_to_jpeg(h_img)
display(orgimage_widget)
```

```
(252, 474, 3)
(119448,)
```



```
CPU times: user 429 ms, sys: 153 ms, total: 582 ms
Wall time: 315 ms
```

```
[60]: %%time
mod = SourceModule \
(
    ...

__global__ void add_num(float *d_result, float *d_a, float *d_b,int N)
{
    int tid = threadIdx.x + blockIdx.x * blockDim.x;
    while (tid < N)
    {
        d_result[tid] = d_a[tid] + d_b[tid];
        if(d_result[tid]>255)
        {
            d_result[tid]=255;
        }
        tid = tid + blockDim.x * gridDim.x;
    }
    }
    ...
}
```

```
CPU times: user 1.95 ms, sys: 0 ms, total: 1.95 ms
Wall time: 1.52 ms
```

```
[48]: img1 = cv2.imread('/kaggle/input/winter-2024/Test for winter 2024.jfif',0)
img2 = cv2.imread('/kaggle/input/winter-2024/Test for winter 2024.jfif',0)
print(img1.shape)
print(img2.shape)
```

```
(252, 474)
(252, 474)
```

```
[51]: h_img1 = img1.reshape(119448).astype(np.float32)
h_img2 = img2.reshape(119448).astype(np.float32)
print(h_img1.shape)
print(h_img2.shape)
```

```
(119448,)
(119448,)
```

```
[54]: N = h_img1.size
h_result=h_img1
print(N)
print(h_result)
```

```
119448
[229. 224. 217. ... 69. 72. 82.]
```

```
[64]: add_img = mod.get_function("add_num")
add_img(cuda.Out(h_result), cuda.In(h_img1), cuda.In(h_img2),np.uint32(N),block=(1024, 1, 1), grid=(64, 1, 1))
h_result=np.reshape(h_result,(252,474)).astype(np.uint8)
```



```
[48]: img1 = cv2.imread('/kaggle/input/winter-2024/Test for winter 2024.jfif',0)
img2 = cv2.imread('/kaggle/input/winter-2024/Test for winter 2024.jfif',0)
print(img1.shape)
print(img2.shape)
```

```
(252, 474)
(252, 474)
```

```
[51]: h_img1 = img1.reshape(119448).astype(np.float32)
h_img2 = img2.reshape(119448).astype(np.float32)
print(h_img1.shape)
print(h_img2.shape)
```

```
(119448,)
(119448,)
```

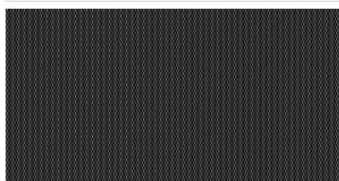
```
[54]: N = h_img1.size
h_result=h_img1
print(N)
print(h_result)
```

```
119448
[229. 224. 217. .... 69. 72. 82.]
```

```
[64]: add_img = mod.get_function("add_num")
add_img.cuda.Out(h_result), cuda.In(h_img1), cuda.In(h_img2), np.uint32(N), block=(1024, 1, 1), grid=(64, 1, 1))
h_result=np.reshape(h_result, (252, 474)).astype(np.uint8)
```

+ Code + Markdown

```
[65]: image_widget = ipywidgets.Image(format='jpeg')
image_widget.value = bgr8_to_jpeg(h_result)
display(image_widget)
```



```
%%time
#CPU version
img = cv2.imread('/kaggle/input/winter-2024/Test for winter 2024.jfif', cv2.IMREAD_GRAYSCALE)

def plotImages(imgr):
    plt.imshow(imgr)
    plt.title("CPU clahe")
    plt.xticks([])
    plt.yticks([])

#contrast Limited Adaptive Histogram Equalization
clahe = cv2.createCLAHE(clipLimit=5.0, tileGridSize=(8, 8))
imgr = clahe.apply(img)

plotImages(imgr)
```

```
CPU times: user 32.5 ms, sys: 2.37 ms, total: 34.9 ms
Wall time: 28.6 ms
```

CPU clahe



+ Code + Markdown

[7]:

```
%time
import pycuda.gpudarray as gpudarray
import pycuda.driver as cuda
import pycuda.autotinit
import numpy

a_gpu = gpudarray.to_gpu(numpy.random.randn(4,4).astype(numpy.float32))
a_doubled = (2*a_gpu).get()
print(a_doubled)
print(a_gpu)
```

```
[[-0.26158872  0.4961721  -0.83486335  1.9181659 ]
 [ 0.33141446  1.6056442   1.8213083  -2.345274 ]
 [ 1.0849017  -2.7049644   1.4376756  -0.14914745]
 [ 1.6721989   1.285488  -3.562288  -0.01868289]]
[[-0.13079436  0.24808605  -0.41783168  0.95908296]
 [ 0.16570723  0.8028221   0.91065013  -1.172637 ]
 [ 0.54245085  -1.3524822   0.7188378  -0.07457373]
 [ 0.83689845  0.642744  -1.781144  -0.00934144]]
CPU times: user 19 ms, sys: 3.91 ms, total: 22.9 ms
Wall time: 639 ms
```