from \_\_future\_\_ import print\_function  
from \_\_future\_\_ import division  
import cv2 as cv  
import numpy as np  
import argparse  
import os  
def loadExposureSeq(path):  
 images = []  
 times = []  
 with open(os.path.join(path, 'resources/list.txt')) as f:  
 content = f.readlines()  
 for line in content:  
 tokens = line.split()  
 images.append(cv.imread(os.path.join(path, tokens[0])))  
 times.append(1 / float(tokens[1]))  
 return images, np.asarray(times, dtype=np.float32)  
parser = argparse.ArgumentParser(description='Code for High Dynamic Range Imaging tutorial.')  
parser.add\_argument('--input', type=str, help='Path to the directory that contains images and exposure times.')  
args = parser.parse\_args()  
if not args.input:  
 parser.print\_help()  
 exit(0)  
images, times = loadExposureSeq(args.input)  
calibrate = cv.createCalibrateDebevec()  
response = calibrate.process(images, times)  
merge\_debevec = cv.createMergeDebevec()  
hdr = merge\_debevec.process(images, times, response)  
tonemap = cv.createTonemap(2.2)  
ldr = tonemap.process(hdr)  
merge\_mertens = cv.createMergeMertens()  
fusion = merge\_mertens.process(images)  
cv.imwrite('resources/fusion.png', fusion \* 255)  
cv.imwrite('resources/ldr.png', ldr \* 255)  
cv.imwrite('resources/hdr.hdr', hdr)

usage: ipykernel\_launcher.py [-h] [--input INPUT]  
ipykernel\_launcher.py: error: unrecognized arguments: --f=/home/dread/.local/share/jupyter/runtime/kernel-v2-6998D69J03V212Ui.json

An exception has occurred, use %tb to see the full traceback.  
  
SystemExit: 2

/home/dread/miniconda3/envs/tf/lib/python3.9/site-packages/IPython/core/interactiveshell.py:3534: UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.  
 warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)

images, times = loadExposureSeq(args.input)

calibrate = cv.createCalibrateDebevec()  
response = calibrate.process(images, times)

merge\_debevec = cv.createMergeDebevec()  
hdr = merge\_debevec.process(images, times, response)

tonemap = cv.createTonemap(2.2)  
ldr = tonemap.process(hdr)

merge\_mertens = cv.createMergeMertens()  
fusion = merge\_mertens.process(images)

cv.imwrite('resources/fusion.png', fusion \* 255)  
cv.imwrite('resources/ldr.png', ldr \* 255)  
cv.imwrite('resources/hdr.hdr', hdr)