Department of Statistics & Data Science Institute
Columbia University

GR5293 Applied Machine Learning for Computer Vision

Homework #1

Instructor: Xiaofu He

Fall 2023

Requirement

1. For this assignment, design a K-means algorithm using cv2.kmeans() to segment the “Stop” sign in 24 distinct road images.

2. Begin with the provided starter code in “start.py”. Before test your code, create a folder named “results” under the same directory with your script and “images” folder. Complete the “get_box” function to ensure the script operates correctly. While you may introduce new helper functions as needed, avoid from altering the original starter code or adding new code directly under main. You may import the following additional packages beyond what's already included in the starter code: NumPy, Pandas, Matplotlib, and Math.

3. Code clarity. Aim to comment on every variable, loop, and function you introduce. Code that lacks proper comments or is difficult to understand may result in a deduction of points.

Submission & Grading

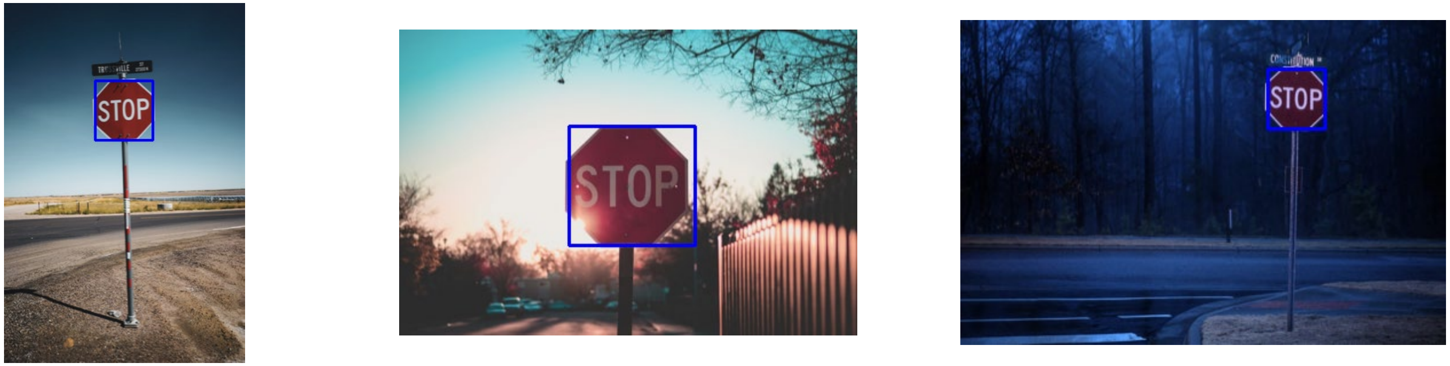
You need to submit 2 files before 11:59 PM, Oct 9th:

1. Your python script named “UNI_kmeans.py”. Ensure that your Python script can run. We will not provide debugging assistance.
2. “README.txt” that contains descriptions for each parameter used in your final cv2.kmeans() function. An explanation of the significance of each parameter and how its value impacts the algorithm. Situations where you believe the performance of your K-means algorithm may falter significantly.

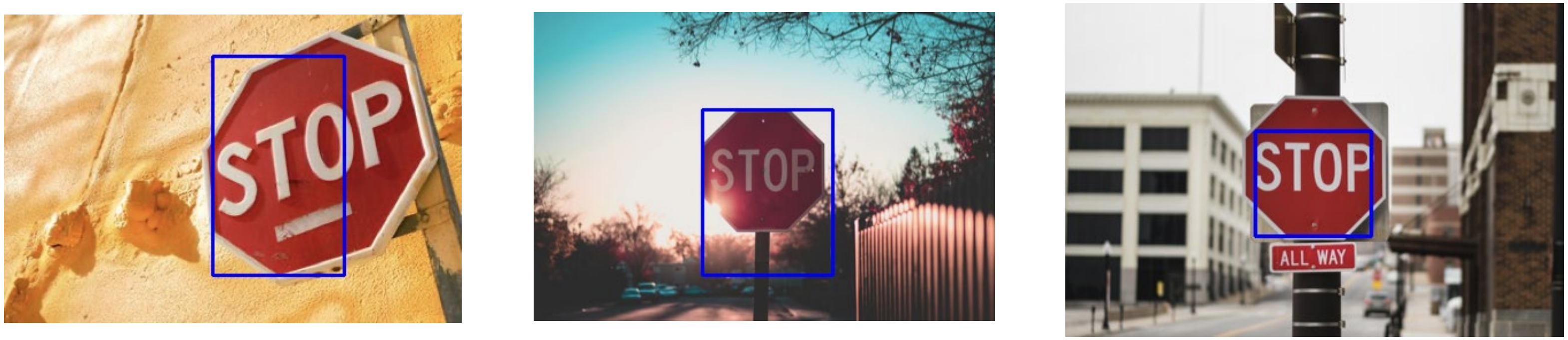
The Grading Rubrics:

Accuracy 65%	3% for completely correct, 1% or 2% for partially correct, and 0% for incorrect on each image. The maximum possible score is 72%, but only 65% will be counted towards the final score.
Python Script 20%	Coding requirement mentioned in the previous slide
README 15%	README requirement mentioned above

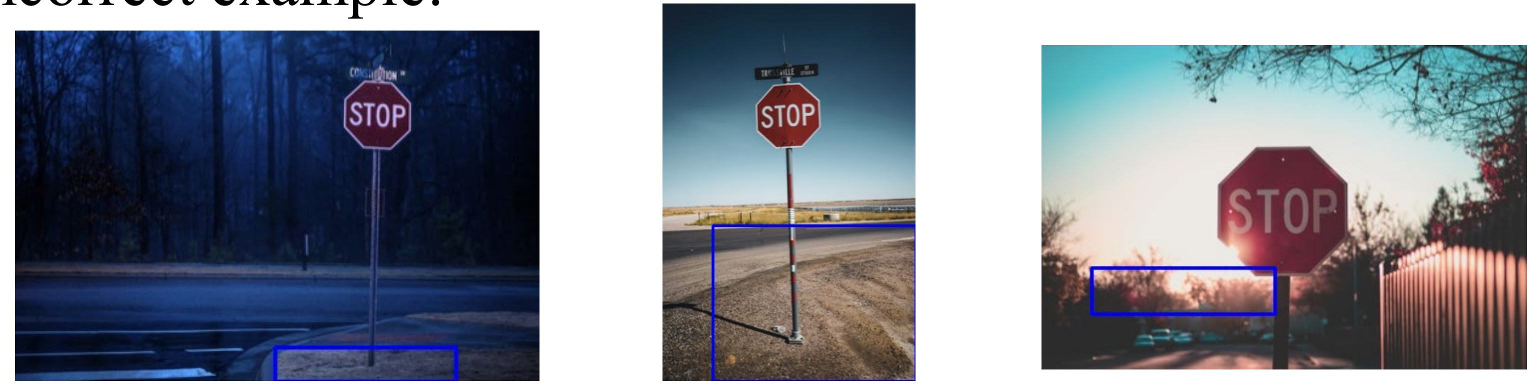
Completely correct example:



Partially correct example:



Incorrect example:



Useful Information

There's a K-means implementation tutorial in OpenCV:

https://docs.opencv.org/4.x/d1/d5c/tutorial_py_kmeans_opencv.html

A function in OpenCV that you might save your time ☺:

https://docs.opencv.org/4.x/d3/dc0/group__imgproc__shape.html#gadf1ad6a0b82947fa1fe3c3d497f260e0

The dataset credits to:

https://www.kaggle.com/datasets/andrewmvd/road-sign-detection