IMAGE SEGMENTATION TECHNIQUES

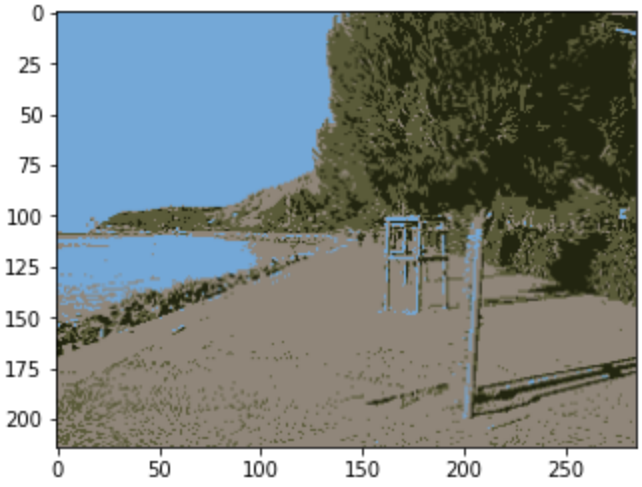
GUIDE: DR. RAGHU KRISHNAPURAM, MR. SADGUN SRINIVAS
AUTHOR: YASH RAJKUMAR NEHRA

Traditional Techniques (Not deep learning based)

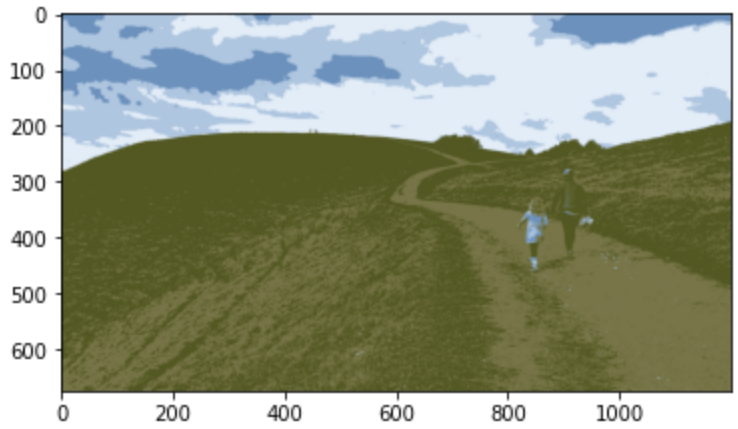
Most of the techniques were tried on some of the following images: *(Numbered Left-Right→Top-Bottom)*

K-means clustering based

- This method is almost automatic and requires only ‘k’ (the number of segments) by the user.
- This works well when there is a fairly stark contrast in the color composition (RGB) of different segments. (It may work better in some other color space like HSV)
- Due to the above constraint, it doesn’t give good results on
    - Images 1 and due to texture on the floor in 2
    - 3 and 4 due to shadow effects
- Gave satisfactory results on images 5 and 6 as follows
    - With k = 4

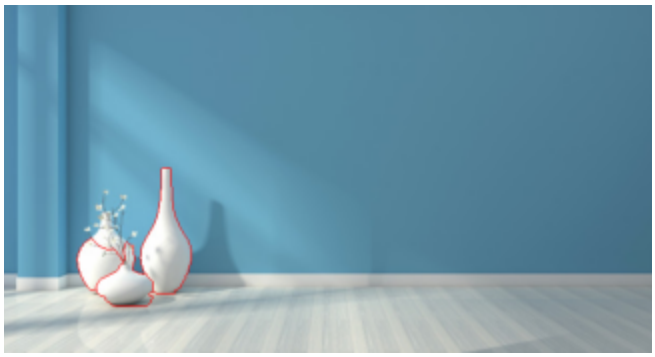


- With k=5 (Barring the 2 persons, the segmentation is satisfactory)



Watershed

- This requires substantial human intervention initially in the form of supplying markers (initial seeds) for the segments to grow into
- Essentially does instance segmentation and especially useful when there are touching or overlapping objects of interest.
- Worked well on the following images
  - Image 4



  - Image 3

- Image 2

Active Contours

- This algorithm requires the user to tweak and set the following parameters
  - alpha, beta: alpha controls the extensibility (stretchability) and beta controls the smoothness (curvature) of the spline (snake)
  - w_line, w_edge: these control the attraction of the spline towards higher/lower intensity regions and edges respectively

Grabcut

- This is the best algorithm so far as it requires minimum human intervention only in the form of providing the rectangular bounding box which entirely contains the foreground object
- This is used for foreground background separation
- The results for favorable cases are quite satisfactory: (These may be improved further using interactive corrections)
  - Image 2

- Image 1:

- Image 3

- Image 4

- Image 5

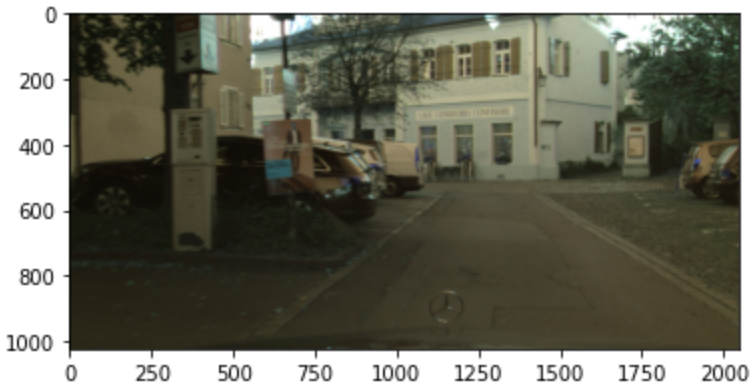
Deep Learning based Techniques

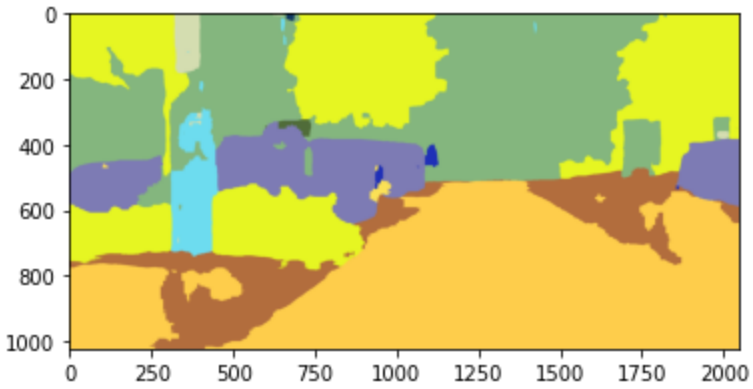
Due to lack of CUDA support, I was able to just run inference on the image below using just **BiSeNet**. Nevertheless, I have studied the working and code of others too and there exist *expt.ipynb* notebooks at suitable places in the directories of each implementation, and one having CUDA support may be able to successfully try them.

Shelfnet:

BiSeNet:

This is the image that was used for running inference on and the fine segmented image:





FROM HERE, IT'S CLEARLY EVIDENT THAT DEEP LEARNING BASED MODELS OUTPERFORM THE TRADITIONAL MODELS BY A HUGE MARGIN.

HardNet: