BLG453E
Homework-3

27.12.2021
Res. Asst. Yusuf Huseyin Sahin
sahinyu@itu.edu.tr

Oh baby!
That's why I'm easy
I'm easy like Sunday morning

Easy, Lionel Richie

- You should write all your code in Python language.
- Cheating is highly discouraged. If you are planning to use different libraries or functions, please ask me about it.
- Ninova only stores files under 20 MB. If you could not upload your results, you can share them with me via Dropbox, or send me private YouTube video links for each part's results.

1 - Part 1: Canny Edge Detection (50 pts)

BSDS500 [1] is a tiny dataset created to test hierarchical segmentation of each object/subject from an image according to their edge boundaries. An example image and its ground truth boundaries are given in Figure 1.

For this part of the homework, download the given dataset and run Canny edge detection algorithm on the test images. For the ground truth boundary maps of each image, combine the boundaries as in Figure 2 and use this as the edge ground truths. Calculate the average precision (for 200 images) between your findings using Canny and the edge ground truths. Upload the found edge images.

[1] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html

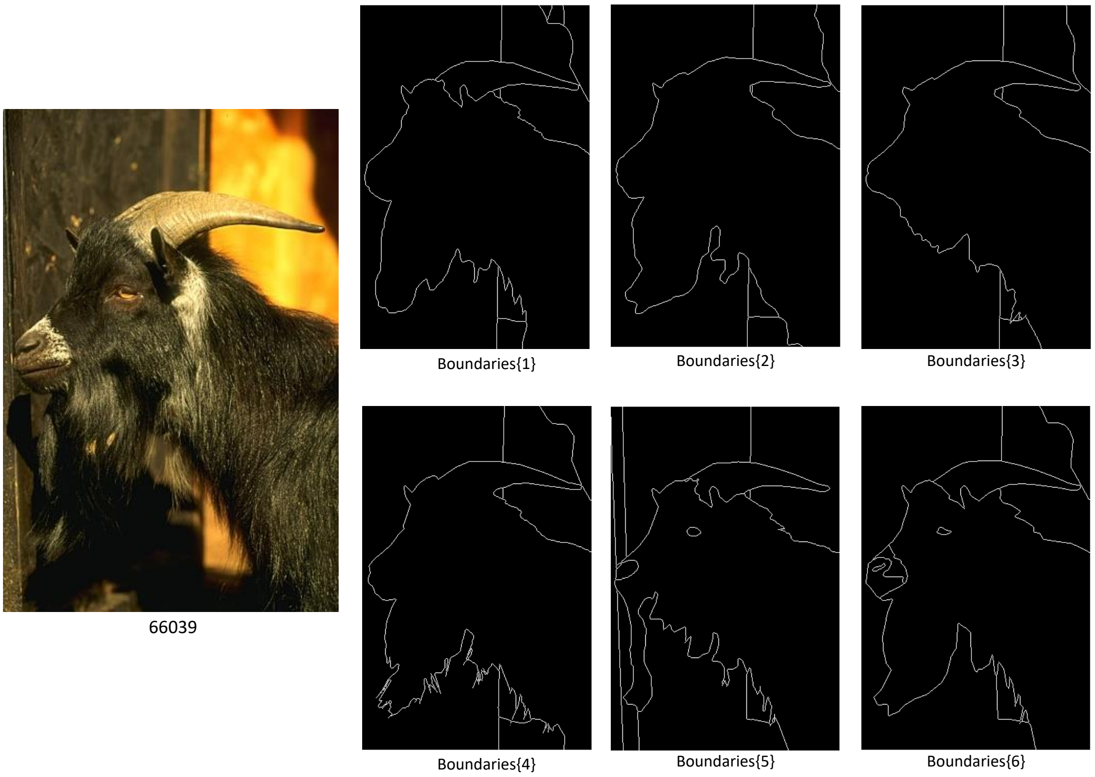

Figure 1: An image from BSDS500 dataset and its edge boundaries.

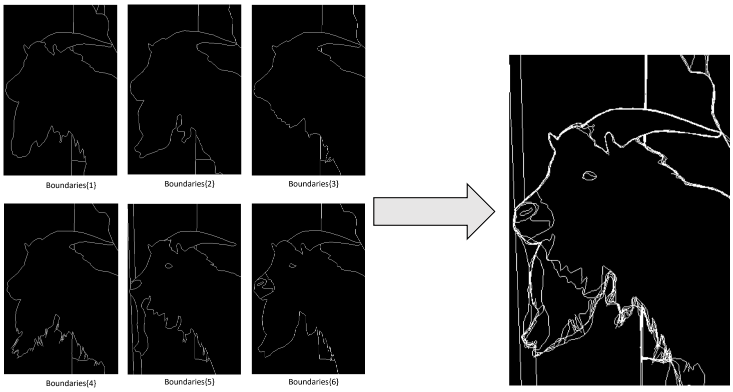

Figure 2: Merging the boundaries.

2 - Part 2: Convolutional Networks for Edge Detection (50 pts)

In this part of the homework, we will use the official code for the article ”Unmixing Convolutional Features for Crisp Edge Detection”[2] to examine the results for the same dataset. To do this, first install the code from its GitHub page[3]. Then, place the dataset and the trained network in the mentioned directories. After running the code in test mode, you will obtain results under the Output directory. Some examples from the outputs are given in Figure 3.

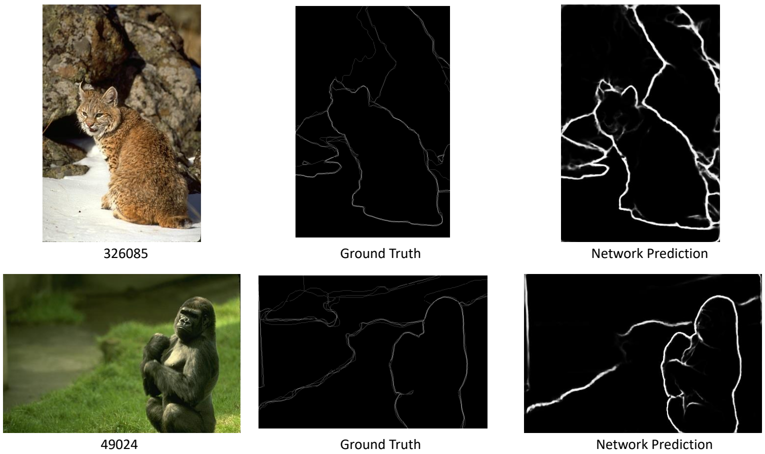

Figure 3: Selected images from BSD500 dataset, their ground truth edge boundaries and the network results.

Run the edge detection network for the test dataset again. Calculate the average precision. Upload your findings.

Running the code on CPU

If you want to run the code in CPU, you should do the following changes on the code.

- Delete every ”.cuda()” function from the code. This function is used to copy the selected weights to GPU.
- The official pretrained model is saved as a GPU model. Thus, instead of using it, download the model's CPU version from my website[4].
- Instead of the line ”model.load_checkpoint()”, use the following lines to load weights.

```
        model.cpu()
        model.load_state_dict(torch.load('pretrained/model_cpu.pth'))

```

[2] Huan, Linxi, et al. ”Unmixing Convolutional Features for Crisp Edge Detection.” IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[3] https://github.com/WHUHLX/CATS

[4] https://web.itu.edu.tr/sahinyu/model_cpu.pth