Julia Maricalva A08922107

Homework #2

Deep Learning for Computer Vision

Due: 108/11/5 (Tue.) 03:00

Total Score: 110 points

Contact TAs by ntudlcvta2019@gmail.com

Julia Maricalva A08922107

Index

Problem 1: Semantic Segmentation

Baseline model

1. Describe how you pre-process the data. (Any data augmentation technique used? Do you normalize the data?)

In the pre-process of data I have used as a base the demo that was given in class.

No type of data augmentation has been used for the baseline model.

I have normalized the data with the following values:

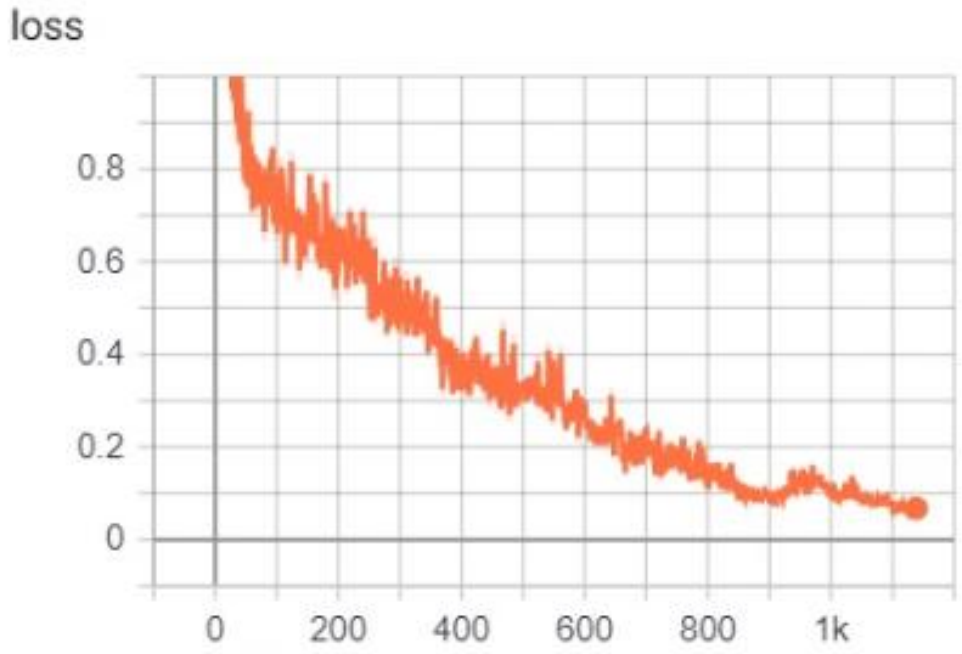
MEAN = [0.485, 0.456, 0.406]

STD = [0.229, 0.224, 0.225]

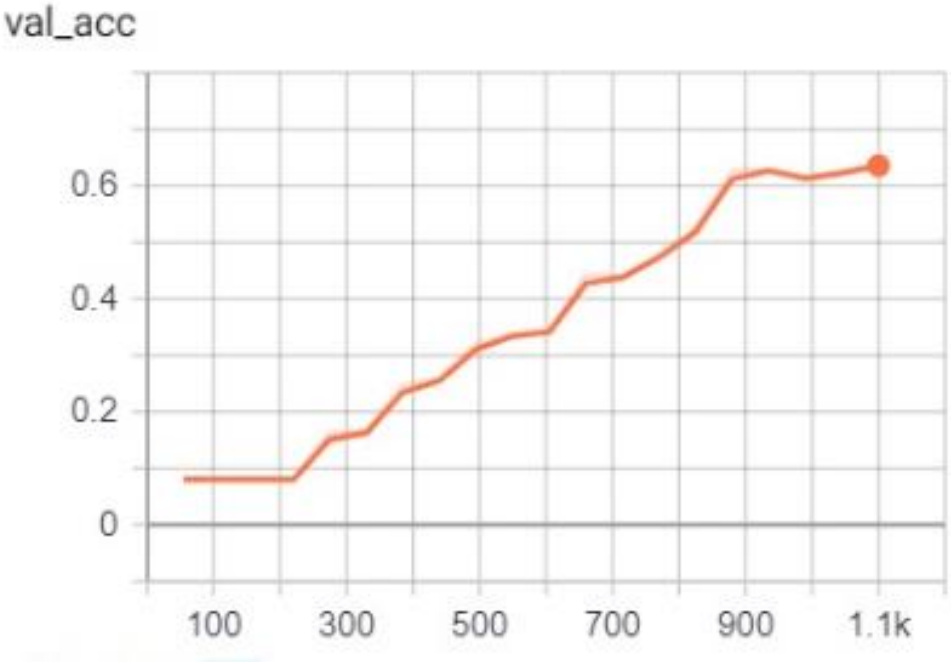
This normalization is taken from Imagenet.

2. Show the following two figures:

1. Training loss versus number of training iterations (Y coordinate: training loss.X coordinate: number of iterations.)



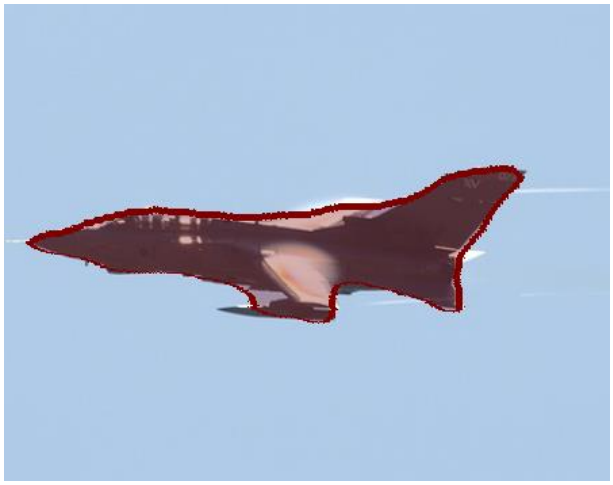
2. IoU score on validation set versus number of training iterations (Y coordinate: IoU score on validation set. X coordinate: number of epochs.)



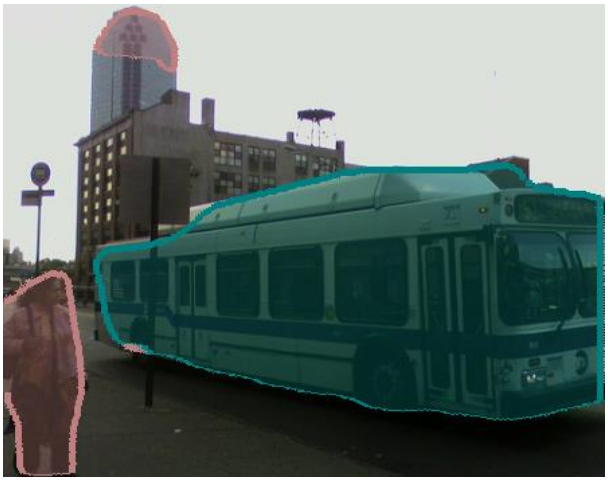
3. Visualize at least one semantic segmentation result for each class.

Class 1: person

Class 2: aeroplane



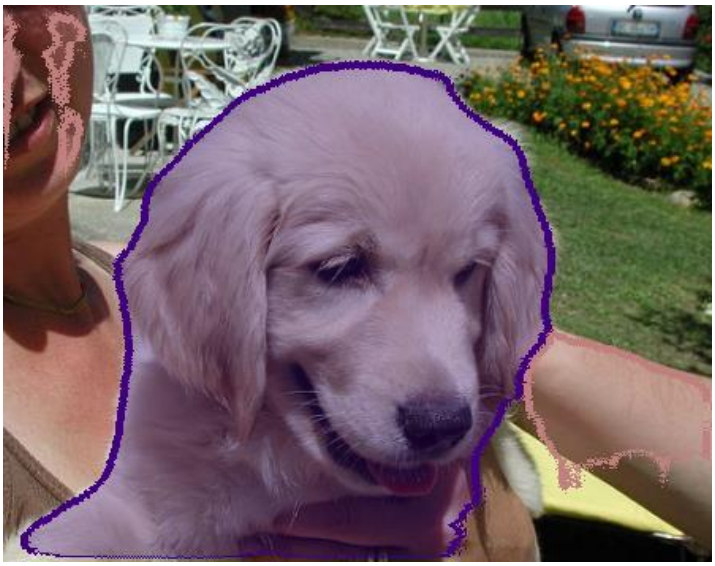
Class 3: bus



Class 4: tv/monitor

Class 5: horse

Class 6: dog



Class 7: cat

Class 8: car

4. Report mIoU score and per-class IoU score of the baseline model. Which class has the highest IoU score? Which class has the lowest IoU score? Please also hypothesize the reason why.

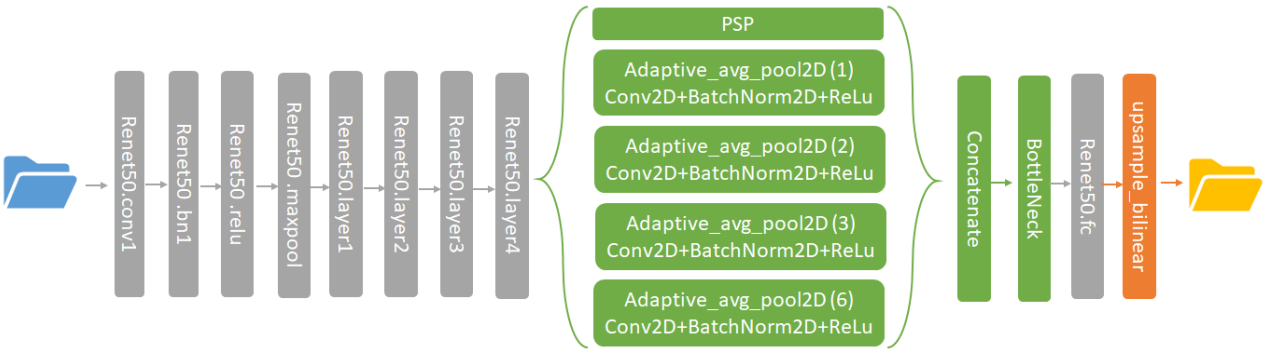
```
class #0 : 0.90072
class #1 : 0.74584
class #2 : 0.63381
class #3 : 0.67263
class #4 : 0.38855
class #5 : 0.50046
class #6 : 0.58476
class #7 : 0.71354
class #8 : 0.64099

mean_iou: 0.642368

0.6423675470875233
```

The class which class has the highest IoU score is the first class, which correspond to the class person. This may be because a person has a very specific shape. In a network, the more specific, the better the model will be trained, because it can recognize more easily that shape in future images.

The class which class has the lowest IoU score is the fourth class, which correspond to the class tv/monitor. The monitor is the class that is worst recognized with much difference, this may be because it is the simplest shape, and therefore the least specific. Therefore, being a simple way, it will be difficult for the network to recognize it and that is why it has the lowest rate.

Improved model

1. Draw the model architecture of your improved model.



2. Discuss the reason why the improved model performs better than the baseline one. You may conduct some experiments and show some evidences to support your discussion.

The architecture uses Resnet50, implemented together with Pyramid Scene Parsing,

The pyramid grouping module merges features under four different pyramid scales. The thickest level highlighted in red is the global grouping to generate a single bin output. The next pyramid level separates the entity map into different subregions and forms a grouped representation for different locations. The output of different levels in the pyramidal grouping module contains the characteristics map with varied sizes.

Basically, this model allows us to make a better recognition of the images, as we see in the next section.

3. To prove that your improved model is better than the baseline one, report the mIoU score of your improved model. Please also show some semantic segmentation results of your improved model and the baseline model.

```
class #0 : 0.90197
class #1 : 0.74383
class #2 : 0.68133
class #3 : 0.76542
class #4 : 0.44481
class #5 : 0.72281
class #6 : 0.76715
class #7 : 0.80569
class #8 : 0.75924

mean_iou: 0.732473

Epoch: [4] ACC:0.7324732158313919
```

Problem 2: Image Filtering

1. Given a variance σ^2, the convolution of a 2D Gaussian kernel can be reduced to two sequential convolutions of a 1D Gaussian kernel. Show that convolving with a 2D Gaussian filter is equivalent to sequentially convolving with a 1D Gaussian filter in both vertical and horizontal directions.

$$f(x,y) * G(x,y) =$$

$$\sum_i \sum_i f(x-i, y-j) \frac{1}{2\pi\sigma^2} e^{\frac{-(i^2+j^2)}{2\sigma^2}}$$

$$= \sum_i [\sum_i f(x-i, y-j) \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-j^2}{2\sigma^2}}] \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-i^2}{2\sigma^2}}$$

$$= f(x,y) * G(y) * G^t(x)$$

2. Implement a discrete 2D Gaussian filter using a 3 × 3 kernel with $\sigma \approx \frac{1}{2*ln2}$. Use the provided lena.png as input, and plot the output image in your report. Briefly describe the effect of the filter.

We expand the image to see in more detail the effect of the filter. This effect is a Gaussian blur, the result of blurring an image by a Gaussian function. This effect is frequently used to reduce image noise and details. The visual effect of this blur technique is a soft blur that resembles when viewing the image through a translucent screen.

3. Consider the image I(x, y) as a function $I: \Re^2 \rightarrow \Re$. When detecting edges in an image, it is often important to extract information from the derivatives of pixel values. Denote the derivatives as follows:

$$I_x(x,y) = \frac{\partial I}{\partial x} \approx \frac{1}{2}(I(x+1,y) - I(x-1,y))$$
$$I_y(x,y) = \frac{\partial I}{\partial y} \approx \frac{1}{2}(I(x,y+1) - I(x,y-1)).$$

Implement the 1D convolution kernels $K_x \in \Re^{1x3}$ and $K_y \in \Re^{3x1}$ such that

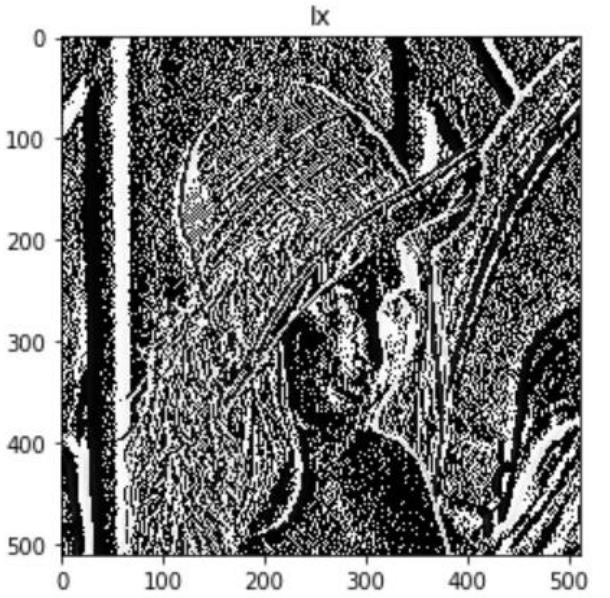
$$I_x = I * k_x$$
$$I_y = I * k_y.$$

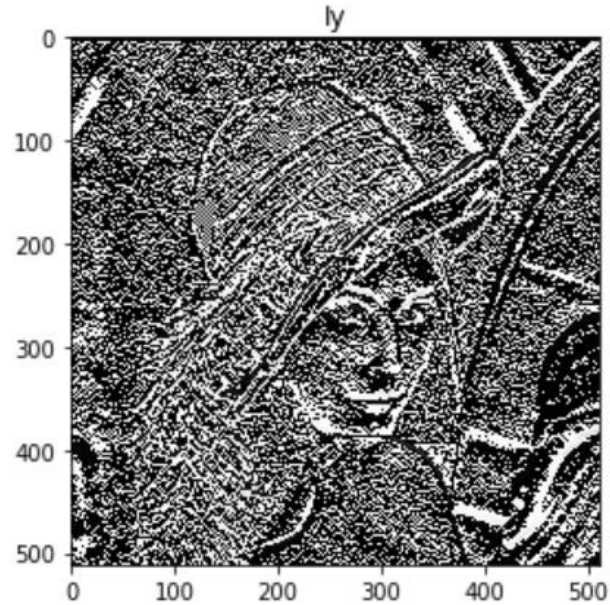
Write down your answers of K_x and K_y. Also, plot the resulting images Ix and Iy using the provided lena.png as input.

The kernels K_x y K_y:

$$K_x = [\frac{1}{2}, 0, -\frac{1}{2}]$$

$$K_y = [\frac{1}{2}, 0, -\frac{1}{2}]^T$$

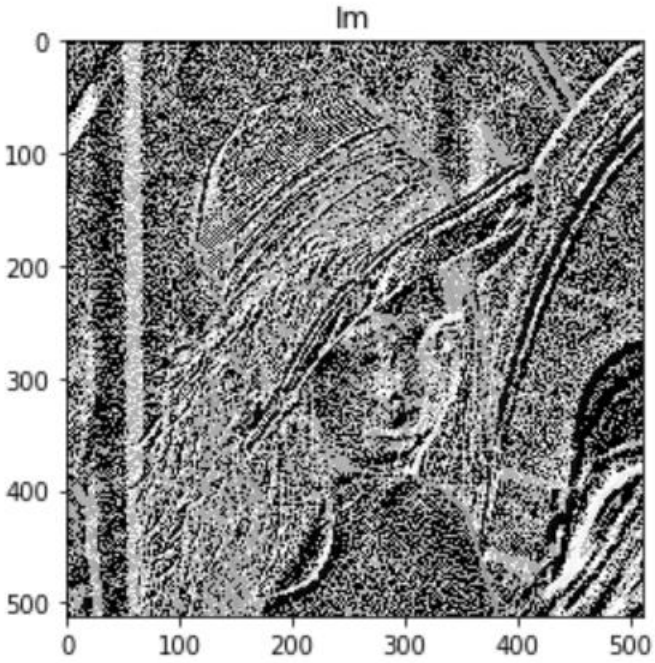


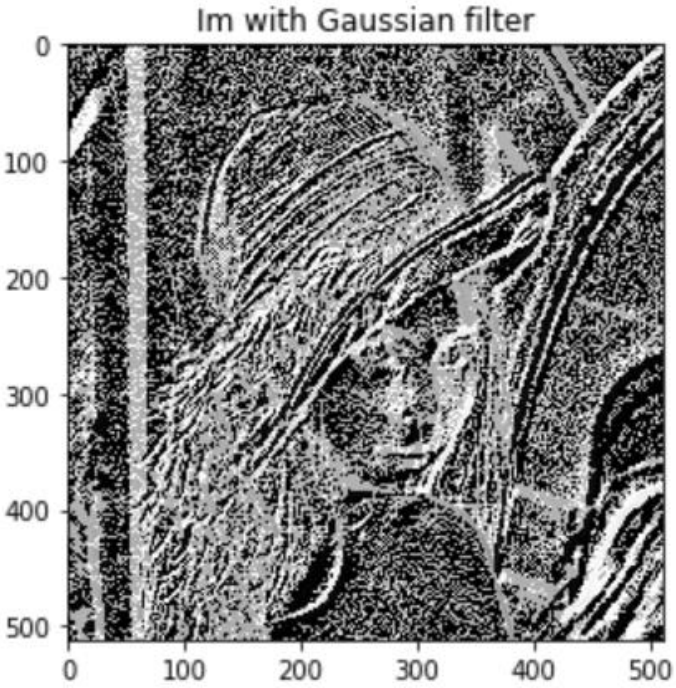


4. Define the gradient magnitude image Im as

$$I_m(x, y) = \sqrt{I_x(x, y)^2 + I_y(x, y)^2}.$$

Use both the provided lena.png and the Gaussian-filtered image you obtained in 2. as input images. Plot the two output gradient magnitude images in your report. Briefly explain the differences in the results.





We note that the image that has been treated with the gauss filter is more defined because by blurring the image we have removed noise, so we see the face more defined.

Bibliography

[1] Models https://pytorch.org/docs/stable/torchvision/models.html

[2] Layers http://jiaya.me/papers/PSPNet_cvpr17.pdf

[3] Pyramid Scene Parsing Network https://pytorch.org/docs/stable/nn.html

[4] Improved Model https://github.com/junfu1115/DANet/blob/master/encoding/models

[4] Collaborators:

Ricardo Manzanedo R08942139

Carlos Marzal A08922106

Javier Sanguino T08901105

Cèline Nauer A08922116