Report # DLCV Homework 2

Student ID: R07943158

Baseline Model

1. Describe how you pre-process the data. (5%) (Any data augmentation technique used? Do you normalize the data?)

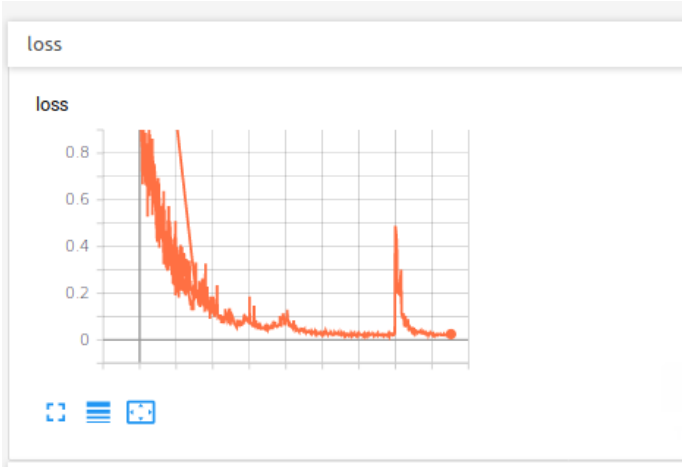
I initially perform Image Processing. I scale the image to scale [0,1]. Finally we normalize the image with MEAN = [0.485, 0.456, 0.406], STD = [0.229,0.224,0.225]

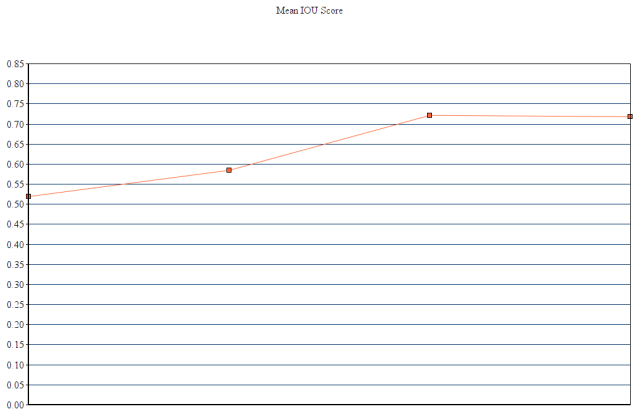
Then I also do a little data augmentation .

I import transform function from torch vision and do random horizontal flip.

2. Show the following two figures:

(i) Training loss versus number of training iterations (Y coordinate: training loss. X coordinate: number of iterations.) (5%)
(ii) IoU score on validation set versus number of training iterations (Y coordinate: IoU score on validation set. X coordinate: number of epochs.)





Here are my two graphs. First showing the training loss v/s number of iterations and second showing iou_score v/s number of epochs.

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

Class 0:

Class 1:

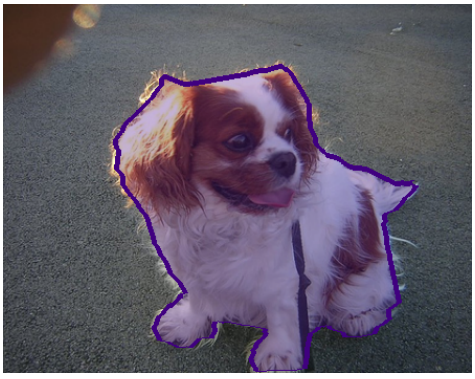
Class 2:

Class 3:

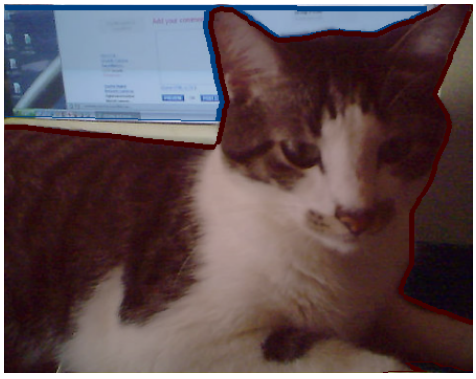
Class 4:

Class 5:

Class 6:



Class 7:



Class 8:

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.

The Validation Class mIou:

```
l/Hw2_data/val/seg/
# preds: 500; pred.shape: (500, 352, 448)
# labels: 500; labels.shape: (500, 352, 448)
class #0 : 0.93018
class #1 : 0.78158
class #2 : 0.71256
class #3 : 0.80681
class #4 : 0.58761
class #5 : 0.58549
class #6 : 0.59697
class #7 : 0.72298
class #8 : 0.73944

mean_iou: 0.718179
```

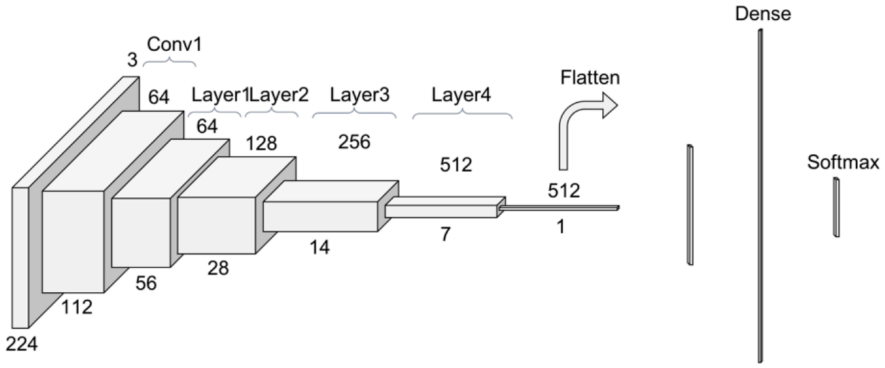
Class 0 (background) has the highest IoU score. While Class 5 has the least IoU score.

The Reasons may be as follows:
That class performs better in which feature extraction data is significant and large. As background is integral part of many images so may be that is the reason that model learns this class very efficiently.
On the other hand the class5 has least score . This may be due to insufficient data available. Thus model is not trained on good data and hence it cannot recognize this class.

Improved Model

1. Draw the model architecture of your improved model.

I simply use ResNet 34 instead of ResNet18 in my improve model.



Layer by Layer Look at ResNet 34

2.2 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.

Well I use Resnet34 in my improve model. I think feature extraction is more efficient in this model architecture than compared to the previous one. To prove this I conduct the training on same dataset and note their validation mIou:

Class	Baseline	Improved
0	0.90265	0.91850
1	0.73640	0.78964
2	0.65717	0.72818
3	0.69838	0.77526
4	0.39785	0.33272
5	0.52336	0.50673
6	0.57198	0.73388
7	0.71563	0.81305
8	0.63950	0.70346
Validation mIou	0.649212	0.701059

2.3 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.

```
r07943158@alcom-H310M-A:~/hw2-divyajainsurana$ python3 mean_iou_evaluate.py -p
l/Hw2_data/val/seg/
# preds: 500; pred.shape: (500, 352, 448)
# labels: 500; labels.shape: (500, 352, 448)
class #0 : 0.93220
class #1 : 0.78592
class #2 : 0.74852
class #3 : 0.81329
class #4 : 0.62099
class #5 : 0.61436
class #6 : 0.65129
class #7 : 0.77701
class #8 : 0.70765

mean_iou: 0.739027
```

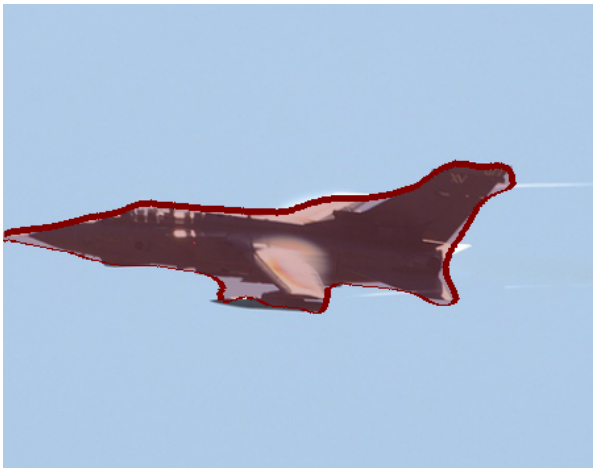
5

Improved Model

BaseLine Model

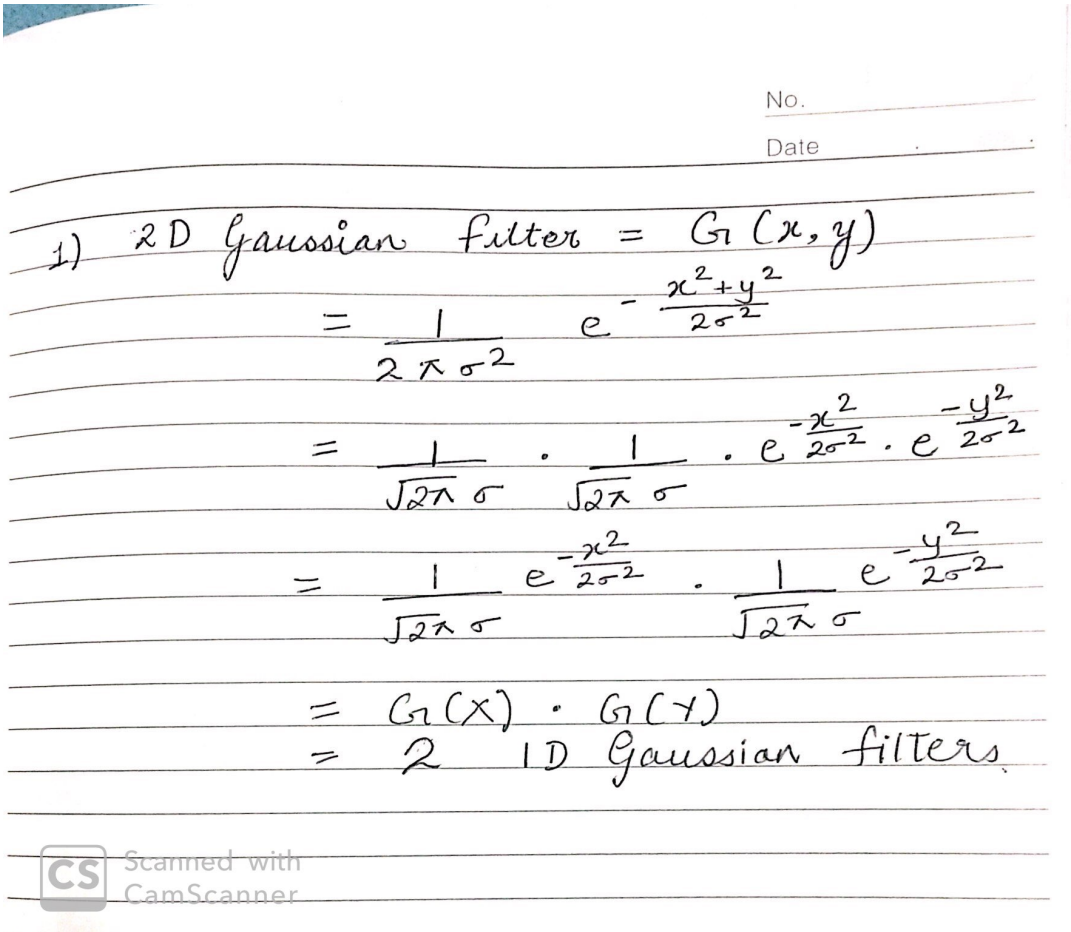
Improved Model

BaseLine Model



Problem 2

2.1 ***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.***



No.

Date

1) 2D Gaussian filter $= G_1(x, y)$

$$= \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$= \frac{1}{\sqrt{2\pi}\sigma} \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{\frac{-x^2}{2\sigma^2}} \cdot e^{\frac{-y^2}{2\sigma^2}}$$

$$= \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-y^2}{2\sigma^2}}$$

$= G_1(X) \cdot G_1(Y)$

$= 2$ 1D Gaussian filters.

2.2 ***Implement a discrete 2D Gaussian filter using a 3x3kernelwith 1/2ln2. Usetheprovided lena.png as input, and plot the output image in your report. Briefly describe the effect of the filter.***

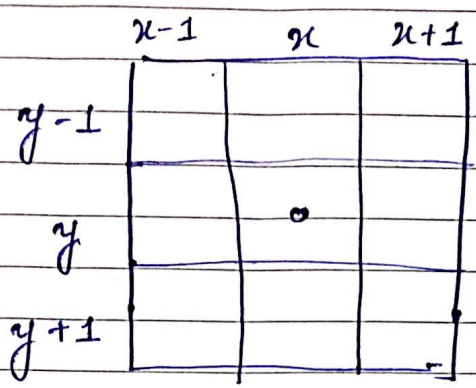
Filtering makes the image smoother than before. More filtering causes the image become a little blur.

Orignal Image

Image obtained after Filtering

2.3 Consider the image I(x, y) as a function I : R^2– > R. When detecting edges in an image, it is often important to extract information from the derivatives of pixel values. Write down your answers of k_x and k_y. Also, plot the resulting images I_x and I_y using the provided lena.png as input.

5.)



$$k_x = \left[-\tfrac{1}{2}, 0, \tfrac{1}{2}\right]$$

$$\# \; I * k_x = \frac{-1}{2}\left[I((x-1), y)\right] + \frac{1}{2}\left[I((x+1), y)\right]$$

$$= I_x$$

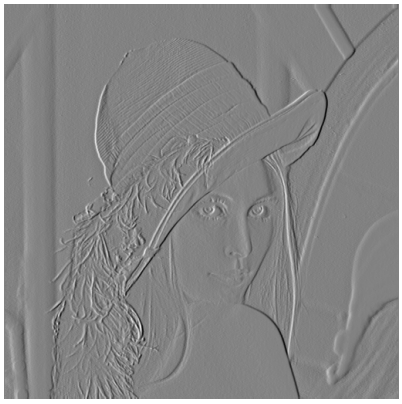
$$k_y = \begin{bmatrix} -1/2 \\ 0 \\ 1/2 \end{bmatrix}$$

$$\# \; I * k_y = \frac{-1}{2} \times I(x, y-1) + \frac{1}{2} \times I(x, y+1)$$

$$= I_y$$

$\therefore k_x = \left[-\tfrac{1}{2}, 0, \tfrac{1}{2}\right]$ $\quad k_y = \begin{bmatrix} -1/2 \\ 0 \\ 1/2 \end{bmatrix}$

Ix Iy



2.4 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 re- port. Briefly explain the differences in the results.

We notice that second image is more clearer in terms of texture. It may be due to processing of some noise.

Original Image

Image obtained after Filtering

Collabarators:

1. A08945201
2. T08902301