DLCV HW2
Problem 1:
Semantic Segmentation

Outline

- Homework Introduction
- Homework Policy
- Others

Outline

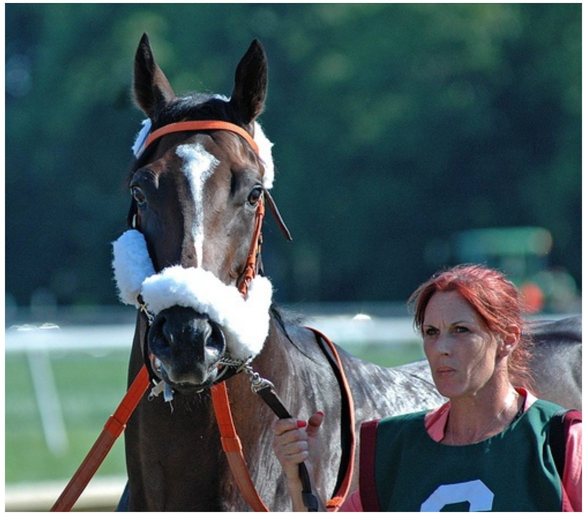
- **Homework Introduction**
- Homework Policy
- Others

Homework 2 Problem 1

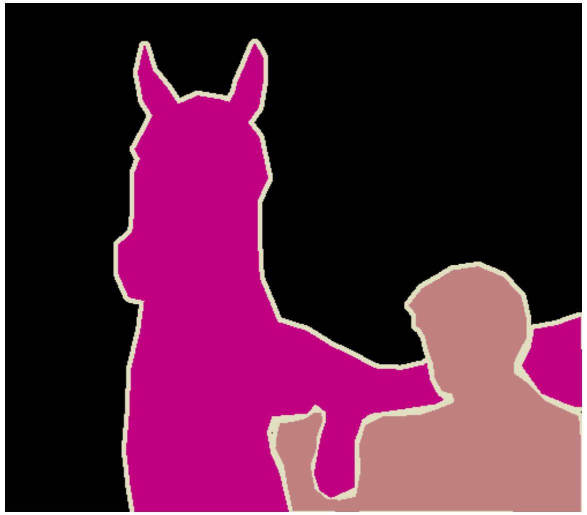
- In this problem, you will need to implement two semantic segmentation models and answer some questions in the report.
  - A baseline model
    - Implement a specific model to perform semantic segmentation.
    - The performance of the baseline model should pass the simple baseline.
  - An improved model
    - Base on the baseline model, you can design you own model or implement some existing models (FCN, SegNEt, Unet …).
    - The performance of the improved model must be better than the baseline model on the validation set.

Semantic Segmentation

- Semantic segmentation aims at **classifying** each pixel in an image to a pre-defined class.



Input: RGB Image

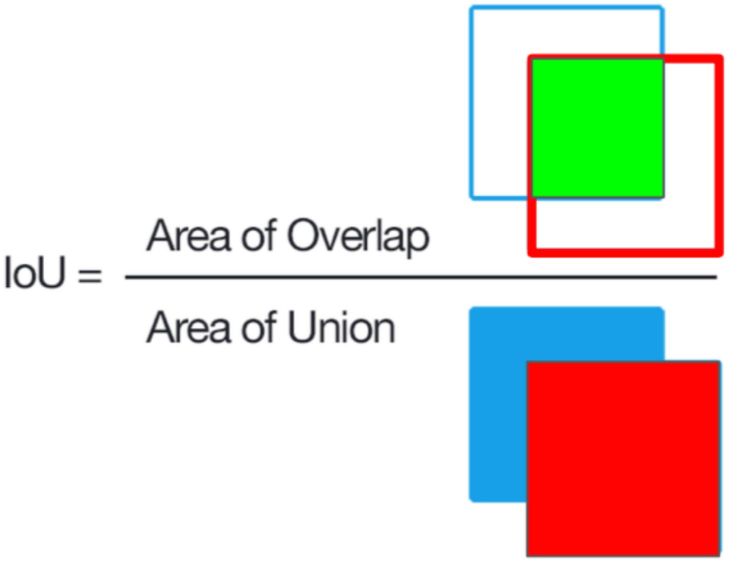


Output: Segmentation
(Grayscale image: each pixel value is
the class this pixel belong to.)

[ref1] Image from Pascal VOC dataset. http://host.robots.ox.ac.uk/pascal/VOC/

Evaluation

- Evaluation metric: mean Intersection over Union (**mIoU**) score
  - For each class, IoU is defined as:
    - True positive/ (True positive + False Positive + False Negative)
  - mIoU is calculated by averaging the IoU over all classes



Dataset

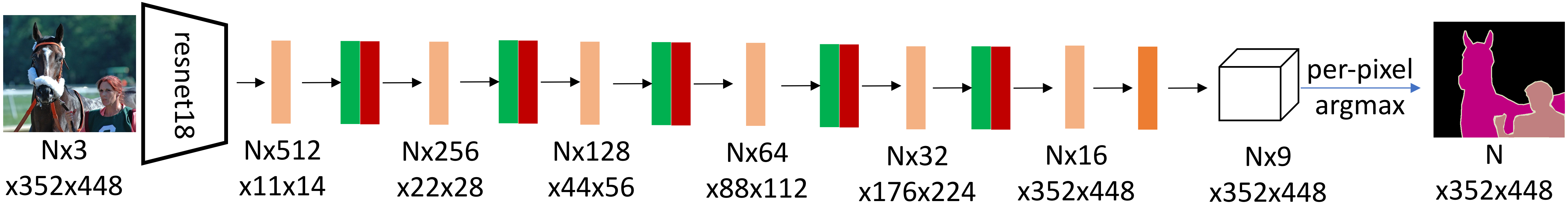
- Class (9 classes in total):
  - Class 0: background
  - Class 1: person
  - Class 2: aeroplane
  - Class 3: bus
  - Class 4: tv/monitor
  - Class 5: horse
  - Class 6: dog
  - Class 7: cat
  - Class 8: car

- Data Split:
  - Training set: 5460 image-segmentation pairs
  - Validation: 500 image-segmentation pairs
  - Test: 500 (Only TAs have the test data)
- Image size
  - Input RGB image: 352x448x3
  - Ground truth segmentation: 352x448
- Note: You should **not** use the validation set to train your model. Anyone using validation set to train their model will get zero points in this homework.

Download dataset: https://drive. google. com/file/d/1wFDqKvaV7kmdlJMuhM2VOq4kQzVV5AIy/view? usp = sharing

Baseline Model - Overview

- Use resnet18 as backbone (or called feature extractor) and use some transpose convolutional layers and convolution layers to build a model.
- This model must pass the simple baseline.



🟧 feature maps with shape =
(batch_size, #channel, height, width)

🟩 transpose convolution layer
(kernel_size=4, stride=2, padding=1, bias=False)

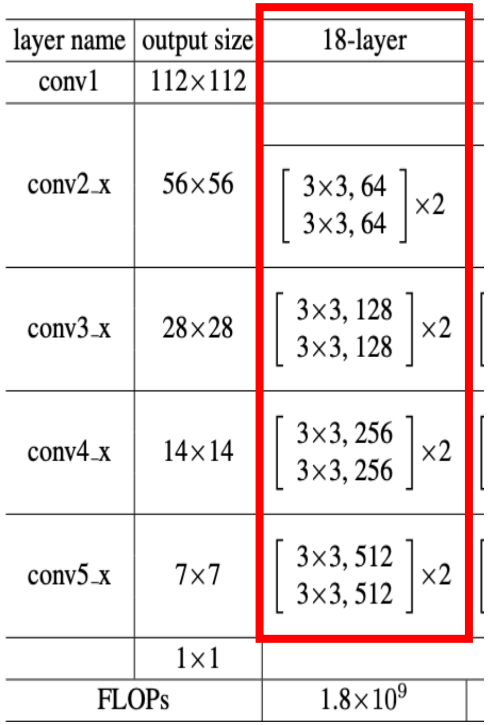
🟧 convolution layer
(kernel_size=1, stride=1, padding=0, bias=True)

🟥 ReLU

Baseline Model

Input size	Output size	Layer type	Layer parameters			
			kernel size	stride	padding	bias
(Nx3x352x448)	(Nx512x11x14)	resnet18	-	-	-	-
(Nx512x11x14)	(Nx256x22x28)	Transpose conv	4	2	1	False
(Nx256x22x28)	(Nx256x22x28)	ReLU				
(Nx256x22x28)	(Nx128x44x56)	Transpose conv	4	2	1	False
(Nx128x44x56)	(Nx128x44x56)	ReLU				
(Nx128x44x56)	(Nx64x88x112)	Transpose conv	4	2	1	False
(Nx64x88x112)	(Nx64x88x112)	ReLU				
(Nx64x88x114)	(Nx32x176x224)	Transpose conv	4	2	1	False
(Nx32x176x224)	(Nx32x176x224)	ReLU				
(Nx32x176x228)	(Nx16x352x448)	Transpose conv	4	2	1	False
(Nx16x352x448)	(Nx16x352x448)	ReLU				
(Nx16x352x448)	**(Nx9x352x448)**	conv	1	1	0	True

Baseline Model –Resnet18



layer name	output size	18-layer
conv1	112×112	
conv2_x	56×56	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$
	1×1	
FLOPs		1.8×10^9

resnet18

- You do not have to implement resnet18 by yourself. Pytorch has implemented it for you. [link]
- You can also load the Imagenet pre-trained weight and bias.

[ref2] He at al., Deep Residual Learning for Image Recognition. In CVPR, 2016.

Report

1. Baseline model
    1. Describe how you pre-process the data. (5%) (Any data augmentation technique used? Do you normalize the data?)
    2. Show the following two figures:
        1. Training loss versus number of training iterations (Y coordinate: training loss. X coordinate: number of iterations.) (5%)
        2. IoU score on validation set versus number of training iterations (Y coordinate: IoU score on validation set. X coordinate: number of epochs.) (5%)
    3. Visualize at least one semantic segmentation result for each class. (5%)
    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. (10%)
2. Improved model (If the mIoU of your improved model is worse than that of the baseline model, you will get at most 20 points in this part.)
    1. Draw the model architecture of your improved model. (5%)
    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. (15%)
    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. (10%)

Model Performance (40%)

- On the validation set
  - Simple baseline (15%): 0.627
  - Strong baseline (5%): 0.701
- On the test set:
  - Simple baseline (15%): 0.659
  - Strong baseline (5%): 0.718
- TAs will execute your code to check if you pass the baseline.

Tools

- mIoU:
  - We provide the code to calculate mIoU score.
  - Usage:

```
python3 mean_iou_evaluate.py <-g ground_truth_directory> <-p prediction_directory>
```

- Visualization:
  - We provide the code to draw semantic segmentation map on RGB image.
  - Usage

```
python3 viz_mask.py  < --img_path path_to_the_rgb_image> < --seg_path path_to
_the_segmentation>
```

Outline

- **Homework Introduction**
- Homework Policy
- Other

Deadline and Academic Honesty

- Report and source deadline: **11/5 03:00 (GMT+8)**
- Late policy : Up to 3 free late days in a semester. If you are running out of late day points, late hand-ins will incur **a 30% penalty per day**.
- Taking any unfair advantages over other class members (or letting anyone do so) is strictly prohibited. Violating university policy would result in F for this course.
- Don't use additional data in this homewrok. (Using Imagenet pre-trained model is allowed.)
- Any form of cheating or plagiarism will not be tolerated, which will also result in F for students with such misconduct.

Submission

- **DLCV2019FALL/hw2** on your **GitHub** repository should include the following files :
  - hw2_YourStudentID**.pdf**
  - hw2.sh (for baseline model)
  - hw2_best.sh (for improved model)
  - your python files (e.g., Training code & Testing code)
  - your model files (can be loaded by your python file)
- **Don’t upload your dataset.**
- If any of the file format is wrong, you will get zero point.

Submission cont’d

- TAs will execute hw2.sh and hw2_best.sh to reproduce the mIoU score in your report on the validation set.

Trained Model

- If your model is larger than GitHub's maximum capacity (100MB), you can upload your model to another cloud service (e.g., Dropbox). However, you script file should be able to download the model **automatically**. [link]
- Do not delete your trained model before the TAs disclose your homework score and before you make sure that your score is correct.
- Use the **wget** command in your script to download you model files. Do not use the curl command.
- Note that you **should NOT hard code any path** in your file or script except for the path of your trained model.

Bash Script

- TA will run your code as shown below:
  - CUDA_VISIBLE_DEVICES=#GPU bash hw2.sh $1 $2
  - CUDA_VISIBLE_DEVICES=#GPU bash hw2_best.sh $1 $2
    - $1: testing images directory (images are named ’xxxx.png')
    - $2: output images directory (You must **not** created this directory in your code.)
- If the input RGB image is xxxx.png, your output semantic segmentation map should be xxxx.png.
- You should save the predicted semantic segmentation maps to the output directory ($2).

Bash Script cont’d

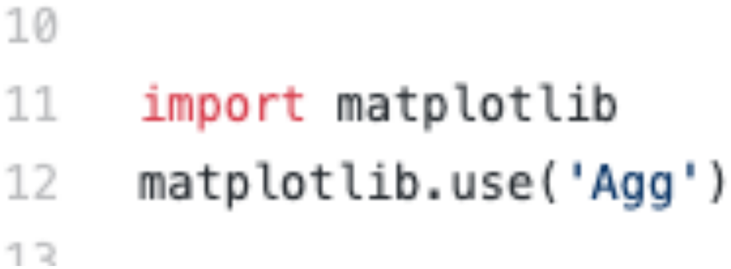
- Your testing code have to be finished in **10 mins.**
- You must **not** use commands such as rm, **sudo**, **CUDA_VISIBLE_DEVICES**, cp, mv, mkdir, cd, pip or other commands to change the Linux environment.
- In your **submitted** script, please use the command **python3** to execute your testing python files.
  - For example: **python3** test.py < -- img_dir $1> < -- save_dir $2>
- We will execute you code on **Linux** system, so try to make sure you code can be executed on Linux system before submitting your homework.

Packages

- python3.6
- pytorch==0.4.1
- scipy==1.2.0
- tensorboardX==1.8
- torchvision==0.2.1
- Other python3.6 standard library
- For more details, please refer to the requirements.txt in your homework package.

Packages cont’d

- If you use matplotlib in your code, please add matplotlib.use(“Agg”) in you code or we will not be able to execute your code.



```python
import matplotlib
matplotlib.use('Agg')
```

- Do not use imshow() or show() in your code or your code will crash.
- Use os.path.join to deal with path as often as possible.

Penalty

- If we can not reproduce your mIoU score on the validation set, you will get 0 points in model performance (40%) and you will receive a 30% penalty in your report score.
- If we can not execute your code, we will give you a chance to make minor modifications to your code. After you modify your code,
  - If we can execute your code and reproduce your results on the validation set, you will still receive a 30% penalty in your homework score.
  - If we can run your code but cannot reproduce your mIoU score on the validation set, you will get 0 points in model performance (40%) and you will receive a 30% penalty in your report score.
  - If we still cannot execute your code, you will get 0 in this problem.

Outline

- Homework Introduction
    - Semantic Segmentation
    - Dataset
    - Evaluation
    - Grading
        - Report (60%)
        - Model performance(40%)
    - Tools

- Homework Policy
    - Deadline and Academic Honesty
    - Submission
    - Trained Model
    - Packages
    - Tools
    - Penalty
- **Other**

Reminder

- Please start working on this homework earlier. The training may take a few hours on a GPU or days on CPUs.
- Please follow the rules.

DOs and DONTs for the TAs (& Instructor)

- Do NOT send private messages to TAs via Facebook.
  - TAs are happy to help, but they are not your tutors 24/7.
- TAs will NOT debug for you, including addressing coding, environmental, library dependency problems.
- TAs do NOT answer questions not related to the course.
- If you cannot make the TA hours, please email the TAs to schedule an appointment instead of stopping by the lab directly

How to find help

- Google !
- TA Hours @ BL-527
  - Monday 11:00 A.M. - 12:00 P.M.
  - Tuesday 11:00 A.M. - 12:00 P.M.
  - Thursday 10:00 A.M. - 11:00 A.M.
- DLCV Fall 2019 FB Club: https://www.facebook.com/groups/2698121610211527/
- TAs Emailbox: ntudlcvta2019@gmail.com