Homework #1

Deep Learning for Computer Vision

NTU, Fall 2021

110/10/12

2021/10/26 (Tue.) 11:59 PM (GMT+8) due

Outline

- Task description & Implementation details
  - Problem 1: Image classification
  - Problem 2: Semantic segmentation
- Grading
- Submission
- Homework policy

Outline

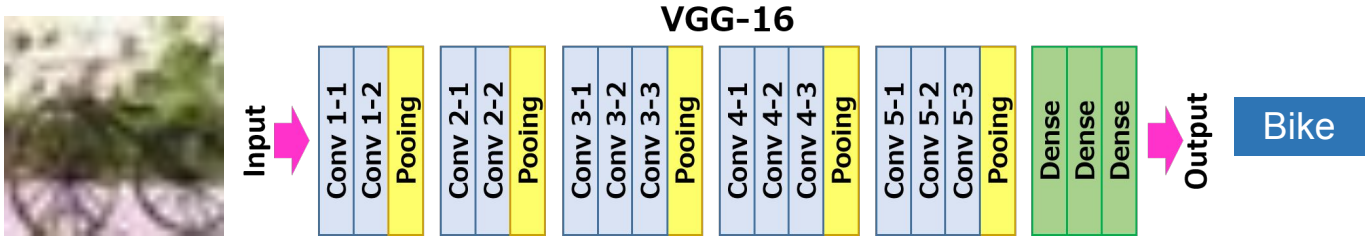
- Task description & Implementation details
  - Problem 1: Image classification
  - Problem 2: Semantic segmentation
- Grading
- Submission
- Homework policy

Problem 1: Image classification

Task Definition

- You will need to perform **image classification** by training a CNN model to predict a label for each image.
  - Input : RGB image
  - Output : classification label

Baseline model



- You can use pre-trained vgg16 (or called feature extractor) as the backbone to build the model. You're also allowed to use other networks to fulfill the task.

Dataset

- The dataset consists of 25000 32x32 colour images in 50 classes.
- We split the dataset into
  - train_50/
    - 22500 images
    - images are named ‘{class label}_{image id}.png’
  - val_50/
    - 2500 images
    - Naming rules are the same as train_50/
    - Note that you **CANNOT** use validation data to train your model

Sample Submission

- Predict the class labels for all the images
  - Output format: csv file
  - The first row must be: ‘image_id, label’

	A	B	C
1	image_id	label	
2	0000.png	0	
3	0001.png	0	
4	0002.png	0	
5	0003.png	0	
6	0004.png	0	
7	0005.png	0	
8	0006.png	0	
9	0007.png	0	
10	0008.png	0	

Model Evaluation

- Evaluation metric: **Accuracy**
  - accuracy is calculated over **all test images**.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

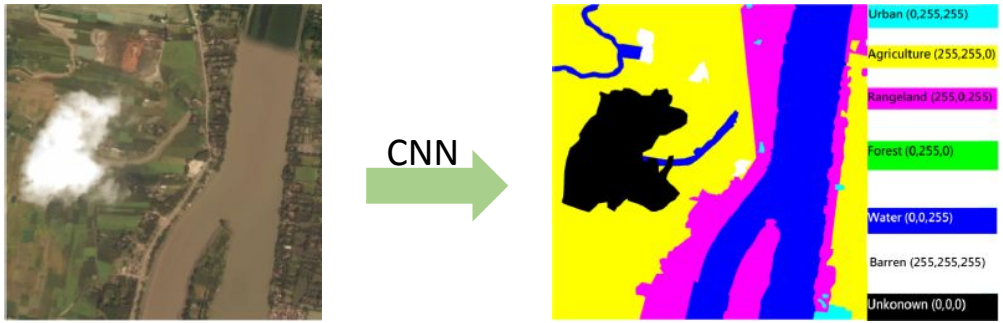
Outline

- Task description & Implementation details
  - Problem 1: Image classification
  - Problem 2: Semantic segmentation
- Grading
- Submission
- Homework policy

Problem 2: Semantic Segmentation

Task Definition

- You will need to perform semantic segmentation by training a CNN model to predict the label for each pixel in an image.
  - Input : RGB image
  - Output : Semantic segmentation/prediction

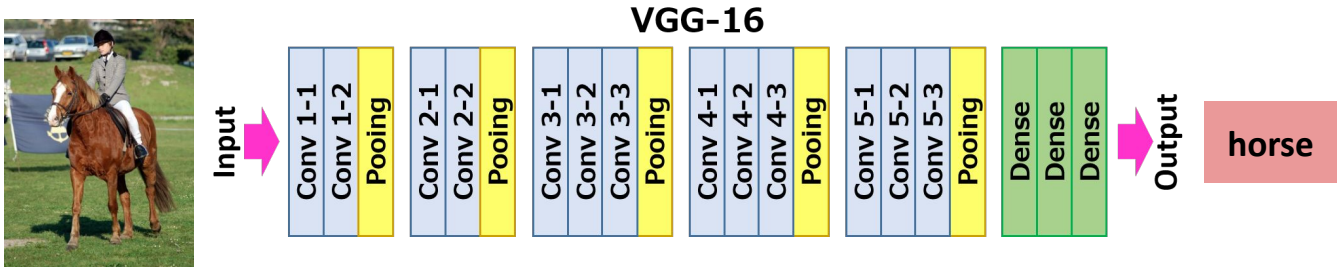


Image

Semantic Segmentation Prediction

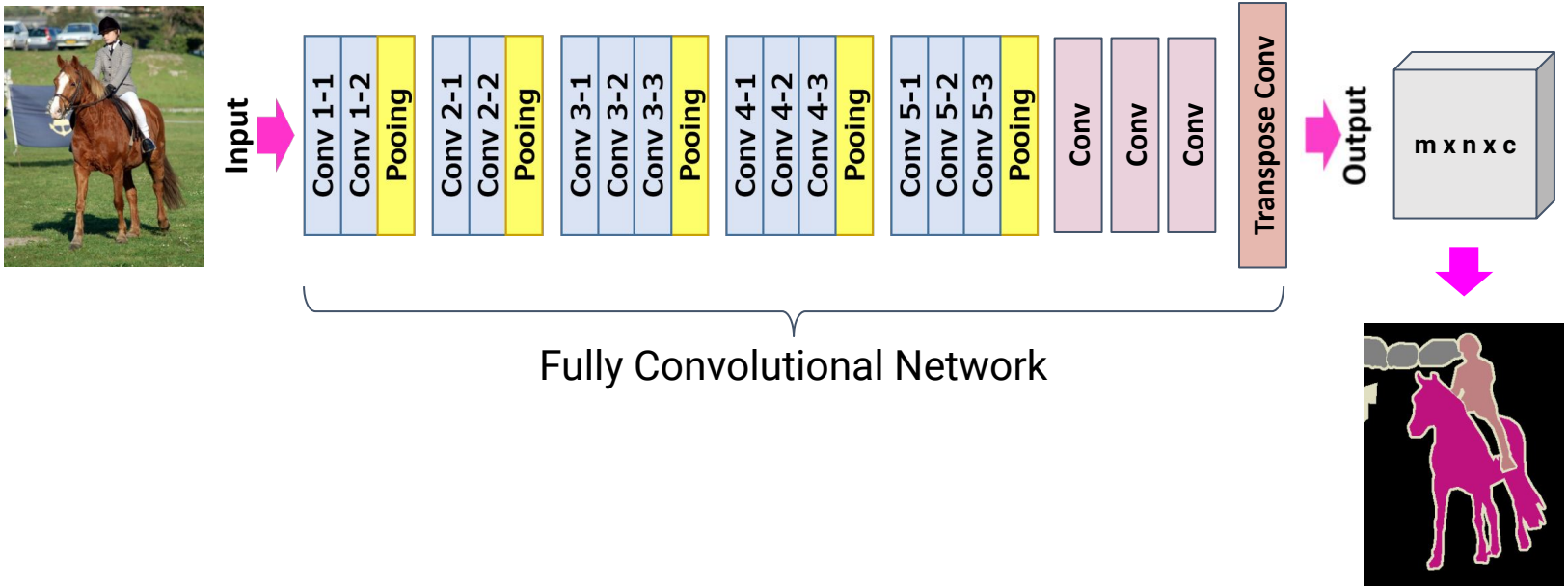
Semantic Segmentation

Convolutional Network - VGG16 for Image Classification



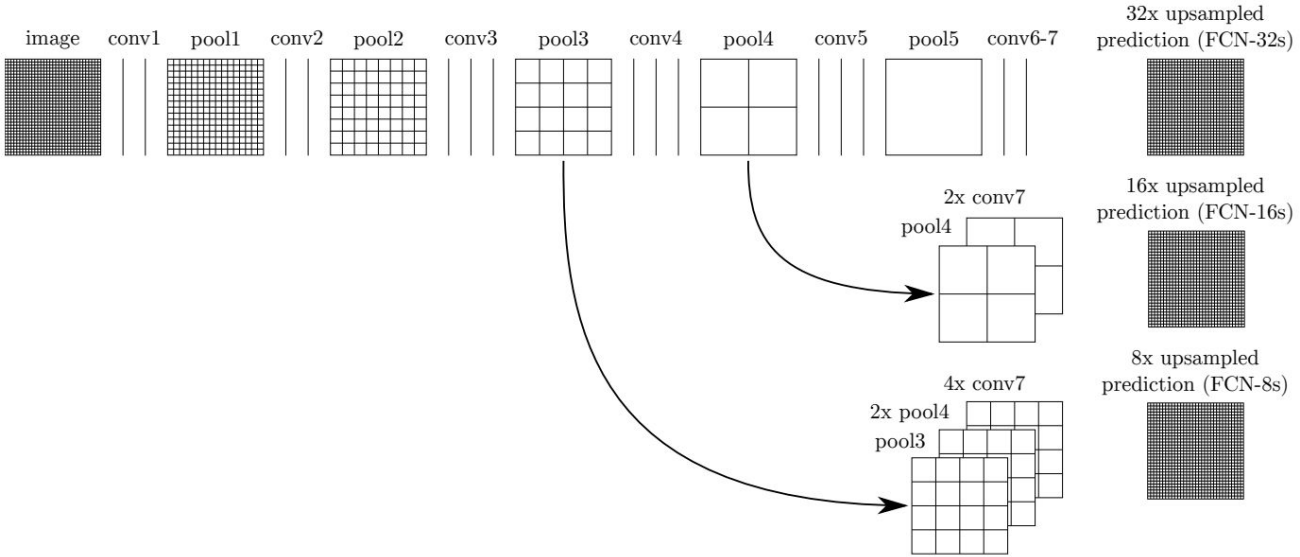
Reference : Long et al., “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015 (Best Paper) [link]

Fully Convolutional Network - FCN 32s



Reference : Long et al., “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015 (Best Paper) [link]

Semantic Segmentation
Fully Convolutional Network - FCN 32s / 16s / 8s



Reference : Long et al., “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015 (Best Paper) [link]

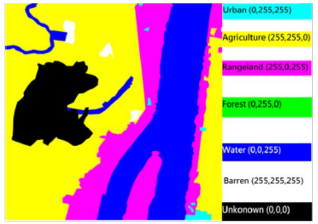
Problem 2: Semantic Segmentation
Baseline Model

- In this part, you will need to implement two segmentation models and provide some implementation details in the report.
    1. **VGG16 + FCN32s (baseline model)**
       Implement VGG16-FCN32s model to perform segmentation.
       The results of this model should pass the baseline performance.
    2. **An improved model**
       Implement an improved model to perform segmentation
       (i.e., with improved segmentation performance than that of the baseline).
       You may choose any model different from VGG16-FCN32s, such as FCN16s, FCN8s, U-Net, SegNet, etc.

Dataset

Image

CNN



Semantic Segmentation Prediction

- Image size: 512x512
- Mask size: 512x512
- There are **7** possible class labels for each pixel.

- train/
  - Contains 2000 image-mask (ground truth) pairs
  - Satellite images are named 'xxxx_sat.jpg’
  - Mask images (ground truth) are named 'xxxx_mask.png'
- validation/
  - Contains 257 image-mask pairs
  - Naming rules are the same as train/
  - Note that you **CANNOT** use validation data to train your model

Implementation Details

VGG16 Architecture

- You need to load the pretrained weights of following layers:

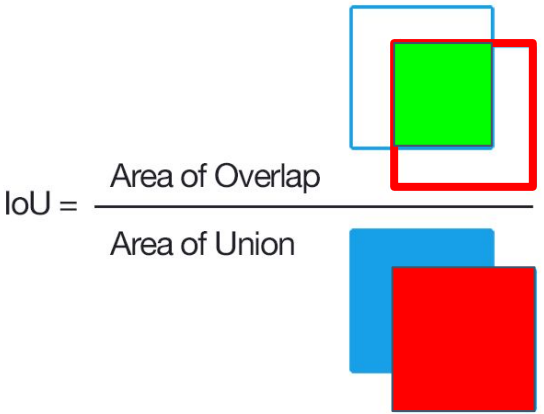
Name	Layer(Size, filter)
block1_conv1	Conv2D(64, 3x3)
block1_conv2	Conv2D(64, 3x3)
block1_pool	MaxPool2D(, 2x2)
block2_conv1	Conv2D(128, 3x3)
block2_conv2	Conv2D(128, 3x3)
block2_pool	MaxPool2D(, 2x2)
block3_conv1	Conv2D(256, 3x3)
block3_conv2	Conv2D(256, 3x3)
block3_conv3	Conv2D(256, 3x3)

Name	Layer(Size, filter)
block3_pool	MaxPool2D(, 2x2)
block4_conv1	Conv2D(512, 3x3)
block4_conv2	Conv2D(512, 3x3)
block4_conv3	Conv2D(512, 3x3)
block4_pool	MaxPool2D(, 2x2)
block5_conv1	Conv2D(512, 3x3)
block5_conv2	Conv2D(512, 3x3)
block5_conv3	Conv2D(512, 3x3)
block5_pool	MaxPool2D(, 2x2)

Model Evaluation

- Evaluation metric: **mean Intersection over Union (mIoU)**
  - For each class, IoU is defined as following:

    True Positive / (True Positive + False Positive + False Negative)
  - mean IoU is calculated by averaging over all classes **except Unknown(0,0,0)**.
  - mIoU is calculated over **all test images**.



Outline

- HW1 Intro - Image classification
  - Task definition
  - Dataset
  - Sample submission
- Grading
- Submission
- Homework policy

Grading - Problem 1 (30%)

- Public Baseline (**10%**) - 2500 validation data (val_50/)
  - simple baseline (5%) - accuracy of 70.0%
  - strong baseline (5%) - accuracy of 80.0%
  - Make sure **not to overfit** the public baseline
- Private Baseline (**10%**) - 5000 testing data (TA will evaluate your scores after deadline)
  - simple baseline (5%)
  - strong baseline (5%)

Grading - Problem 1

- Report (**10%**)
  1. (2%) Print the network architecture of your model.
  2. (2%) Report accuracy of model on the validation set. (TA will reproduce your results, error **±0.5%**)
  3. (6%) Visualize the classification result on validation set by implementing t-SNE on **output features of the second last layer**. Briefly explain your result of the tSNE visualization.

     *You can use scikit-learn to fulfill t-SNE. [link]

Grading - Problem 2 (70%)

- Public Baseline (**25%**) - 257 validation data (validation/)
  - simple baseline (15%): mIoU of 63.0%
  - strong baseline (10%): mIoU of 67.5%
  - Make sure **not to overfit** the public baseline
- Private Baseline (**25%**) - 313 testing data (TA will evaluate your scores after deadline)
  - simple baseline (15%)
  - strong baseline (10%)

Grading - Problem 2

- Report (20%)

1. (5%) Print the network architecture of your VGG16-FCN32s model.
2. (5%) Implement an improved model which performs better than your baseline model. Print the network architecture of this model.
3. (5%) Report mIoU of the improved model on the validation set.
   (TA will reproduce your results, error **±0.5%**)
4. (5%) Show the predicted segmentation mask of “validation/0010_sat.jpg”, “validation/0097_sat.jpg”, “validation/0107_sat.jpg” during the early, middle, and the final stage during the training process of this improved model.

Tools for Semantic Segmentation

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

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

Outline

- HW1 Intro - Image classification
  - Task definition
  - Dataset
  - Sample submission
- Grading
- Submission
- Homework policy

Submission

- Deadline: **110/10/26 (Tue.) 11:59 PM (GMT+8)**
- Click the following link and sign in to your GitHub account to get your submission repository:

  https://classroom.github.com/a/xIPPvRgS

- By default, we will only grade your last submission before the deadline (**NOT** your last submission). Please e-mail the TAs if you'd like to submit another version of your repository, and let us know which commit to grade.

Submission

- **DLCV-Fall-2021/hw1** on your GitHub repository should include the following files:
  - hw1_1.sh
  - hw1_2.sh
  - hw1_<studentID>.pdf
  - your python files (e.g., Training code & Testing code)
  - your model files (can be loaded by your python file)
- **DO NOT upload your dataset.**
- If any of the file format is wrong, you will get zero point.

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, your script file should be able to download the model **automatically**.
  - (Dropbox tutorial: https://drive.google.com/file/d/1XOz69Mgxo67IZNQWnRSjT2eZAZtpUAgZ/view)
- Do **NOT** delete your trained model before the TAs disclose your homework score. Do **NOT** delete your trained model 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 - Problem 1

- TA will run your code as shown below:
  - bash hw1_1.sh $1 $2
    - $1: testing images directory (images are named 'xxxx.png’)
    - $2: path of output csv file(predicted labels)
- The output csv file **must** have the same format as ‘sample_submission.csv’
- Note that you should **NOT** hard code any path in your file or script
- Your testing code have to be finished in **10 mins.**

Bash Script - Problem 2

- TA will run your code as shown below:
  - bash hw1_2.sh $1 $2
    - $1: testing images directory (images are named 'xxxx.png')
    - $2: output images directory (You must not create this directory in your code)
- You **should** name your output **semantic segmentation** as 'xxxx.png'
- Your testing code have to be finished in **10 mins.**

Bash Script (con’d)

- 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

- python==3.6
- certifi==2020.6.20
- cycler==0.10.0
- joblib==0.17.0
- kiwisolver==1.2.0
- matplotlib==3.3.2
- numpy==1.18.1
- pandas==1.1.3
- Pillow==8.0.0
- pyparsing==2.4.7
- python-dateutil==2.8.1
- pytz==2020.1
- scikit-learn==0.21.3
- scipy==1.2.1
- six==1.15.0
- torch==1.4.0
- torchvision==0.5.0
- and other standard python packages
- **E-mail or ask TA first if you want to import other packages.**

Packages

- 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.

Outline

- HW1 Intro - Image classification
  - Task definition
  - Dataset
  - Sample submission
- Kaggle
- Submission
- Homework policy

Deadline and Academic Honesty

- Deadline: **110/10/26 (Tue.) 11:59 PM (GMT+8)**
- Late policy : Up to 3 free late days in a semester (depends on your hw0 result). After that, late homework will be deducted 30% each 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.
- Students are encouraged to discuss the homework assignments, but you must complete the assignment by yourself. TA will compare the similarity of everyone's homework. Any form of cheating or plagiarism will not be tolerated, which will also result in F for students with such misconduct.

Penalty

- If we cannot execute your code, TAs will give you a chance to make minor modifications to your code. After you modify your code,
  - If we can execute your code, you will still receive a **30% penalty** in your model performance score.
  - If we still cannot execute your code, no point will be given.

Reminder

- Please start working on this homework as early as possible. The training may take a few hours on a GPU or days on CPUs.
- Please follows the HW policy.

How to find help

- Google!
- Use TA hours (please check course website for time/location)
- Post your question under hw1 FAQ section in FB group
- Contact TAs by e-mail: ntudlcv@gmail.com
- Some tutorials of Git and PyTorch: link

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.