Homework #1

Deep Learning for Computer Vision

NTU, Fall 2022

Outline

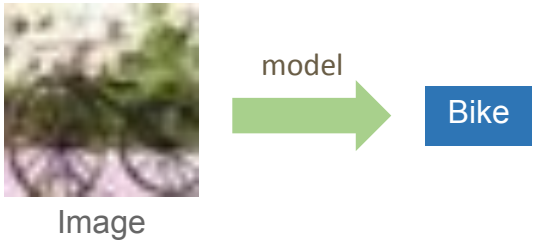
- Problem 1: Image classification
- Problem 2: Semantic segmentation
- Tools
- Submission
- Homework policy

Outline

- Problem 1: Image classification
- Problem 2: Semantic segmentation
- Tools
- Submission
- Homework policy

Problem 1: Image Classification

- Image classification - predict a label for each image
  - Input : RGB image
  - Output : classification label



- You need to perform image classification with the following methods:
  - **A.** Train a CNN model **from scratch**
  - **B.** Try alternative models/methods (e.g., fine-tune a pre-trained model)

Dataset

- The dataset consists of 25,000 colored images (32x32 pixels) with 50 classes.
- We split the dataset into
  - **p1_data/train_50/**
    - 22500 images
    - Images are named ‘{class label}_{***image_id***}.**png**’
  - **p1_data/val_50/**
    - 2500 images
    - Naming rules are same as p1_data/train_50/
    - You can **NOT** use validation data to train your model in a fully supervised manner,
      but feel free to consider semi-supervised, etc. approaches using both training & validation data

Model Evaluation

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

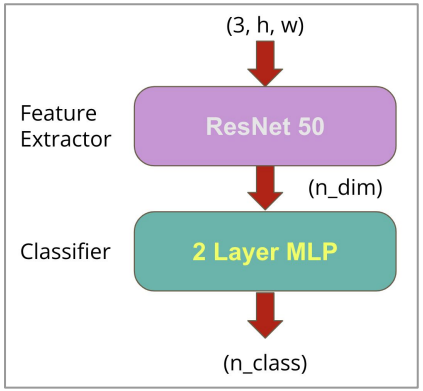
$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Grading (45%)

- Public Baseline (**10%**) - 2500 validation data
  - simple baseline (5%) - **0.75**
  - strong baseline (5%) - **0.86**
- Private Baseline (**10%**) - 5000 testing data
  - simple baseline (5%)
  - strong baseline (5%)
  - Will be announced after deadline
- You only need to submit one model (either A or B) for the above public/private evaluation.

Grading (45%)

Sample Graph for Q1



Report (**25%**)

1. **(2%)** Draw the network architecture of method A or B.
   - The graph should be brief and clear
   - It would be fine to straight copy the figure from the paper
2. **(1%)** Report accuracy of your models (both A, B) on the validation set.
3. **(4%)** Report your implementation details of model A.
   - Including but not limited to optimizer, loss function, cross validation method
4. **(4%)** Report your alternative model or method in B, and describe its difference from model A.

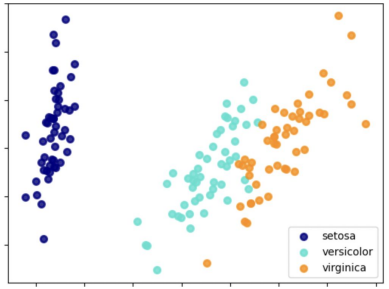
Grading (45%)

Report (**25%**)

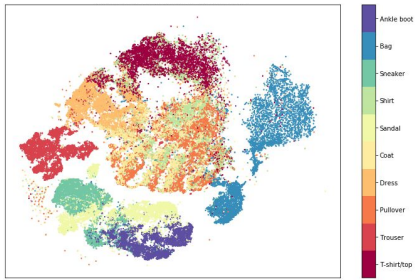
5. **(7%)** Visualize the learned visual representations of **model A** on the **validation set** by implementing **PCA** (Principal Component Analysis) on the output of **the second last layer**. Briefly explain your result of the PCA visualization.

6. **(7%)** Visualize the learned visual representation of **model A**, again on the output of the second last layer, but using **t-SNE** (t-distributed Stochastic Neighbor Embedding) instead. Depict your visualization from **three different epochs** including the first one and the last one. Briefly explain the above results.

PCA



t-SNE



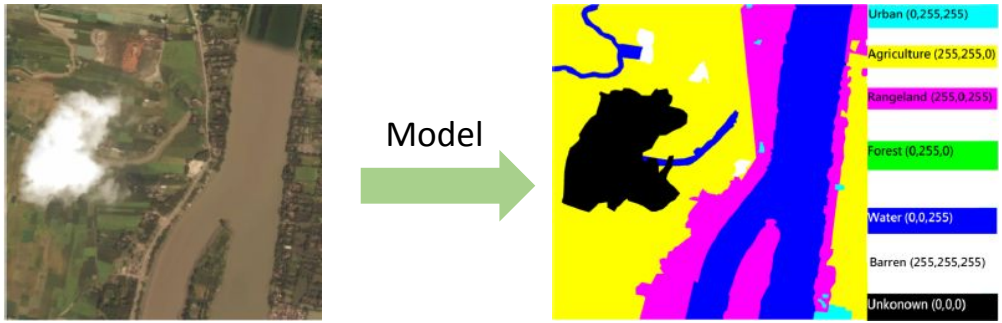
Outline

- Problem 1: Image classification
- Problem 2: Semantic segmentation
- Tools
- Submission
- Homework policy

Problem 2: Semantic Segmentation

Task Definition

- Semantic segmentation - predict the label for each pixel in an image
  - Input : RGB image
  - Output : Semantic segmentation mask



Image

Segmentation Prediction

Problem 2: Semantic Segmentation

Task Definition

- You need to implement two segmentation models and provide implementation details in the report.
  - A: **VGG16 + FCN32s (baseline model)**

    Implement VGG16-FCN32s model to perform segmentation.
  - B: **An improved model**

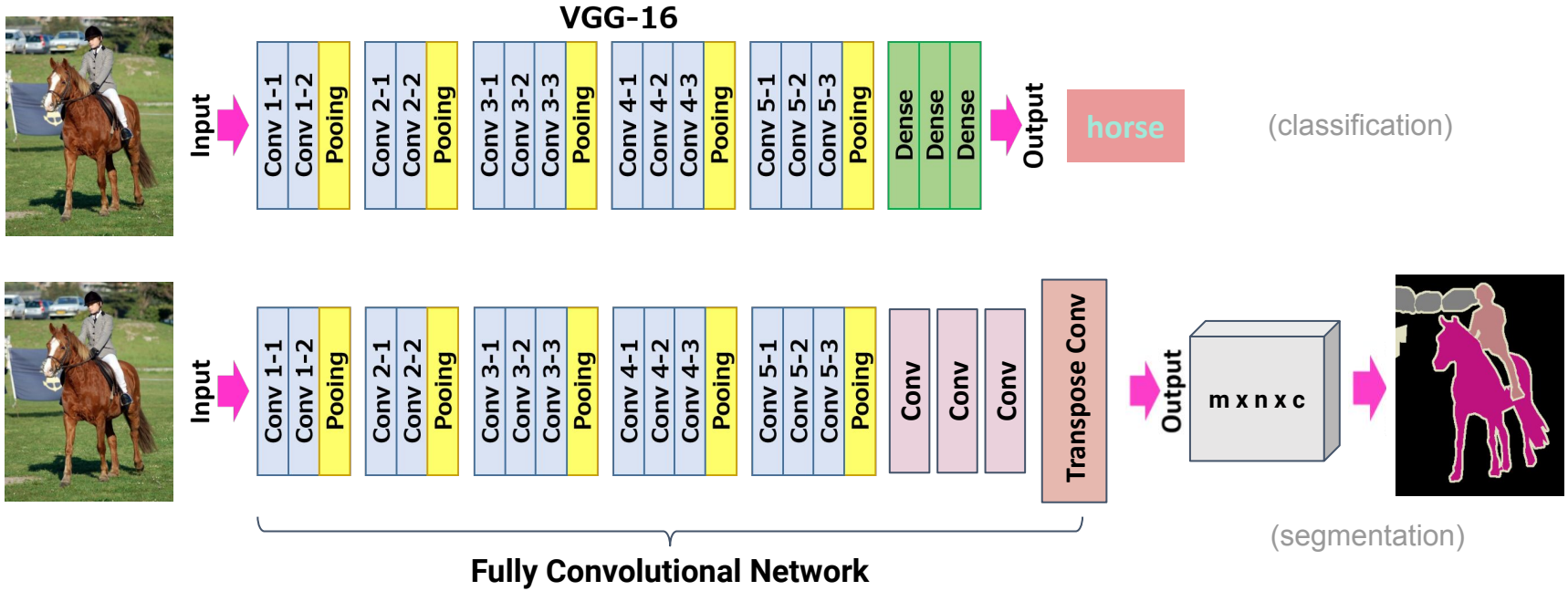
    Implement an improved model to perform segmentation.

    (You may choose any model different from VGG16-FCN32s,

    e.g., FCN16s, FCN8s, U-Net, SegNet, etc.)

Semantic Segmentation

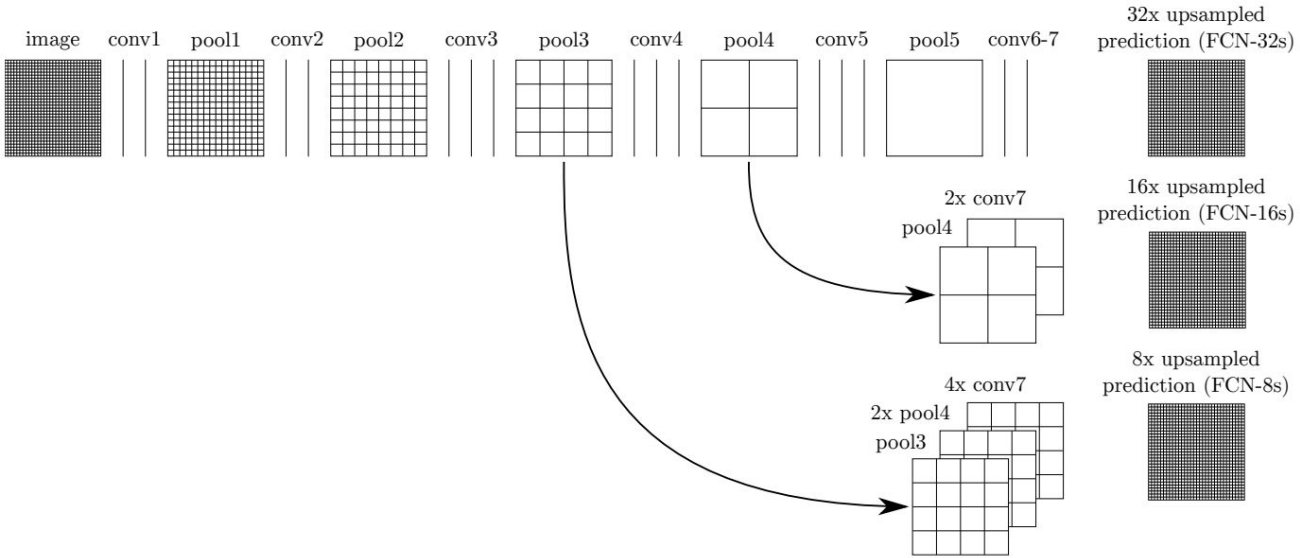
VGG16 + FCN32s



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

Semantic Segmentation

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

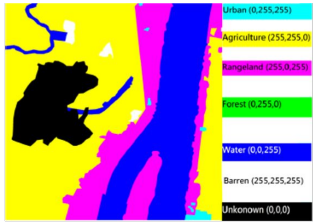


Reference : Long et al., “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015 (Best Paper) [link]

Dataset

Image

CNN



Semantic Segmentation Prediction

- Image size: 512x512
- Mask size: 512x512
- **7** possible class labels

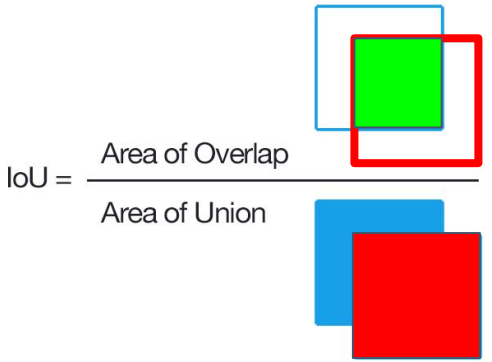
- **p2_data/train/**
  - Contains 2000 image-mask (ground truth) pairs
  - Satellite images are named 'xxxx_sat.**jpg**'
  - Mask images (ground truth) are named 'xxxx_mask.**png**'
- **p2_data/validation/**
  - Contains 257 image-mask pairs
  - Naming rules are the same as p2_data/train/
  - You can **NOT** use validation data to train your model in a fully supervised manner,
    but feel free to consider semi-supervised, etc. approaches using both training & validation data

Metric

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

    $$\textit{True Positive} / (\textit{True Positive} + \textit{False Positive} + \textit{False Negative})$$

  - mean IoU is calculated by averaging over the IoU of all classes **except Unknown(0,0,0)**.
  - mIoU is calculated over **all test images**.



Grading - Problem 2 (55%)

- Public Baseline (15%) - 257 validation data
  - simple baseline (5%): **0.69**
  - strong baseline (10%): **0.73**
- Private Baseline (**20%**) - 313 testing data
  - simple baseline (10%)
  - strong baseline (10%)
  - Will be annouced after deadline
- You only need to submitt one model (either A or B) for the above public/private evaluation.

Grading - Problem 2

Report **(20%)**

1. **(5%)** Draw the network architecture of your VGG16-FCN32s model (model A).
2. **(5%)** Draw the network architecture of the improved model (model B)
   and explain it differs from your VGG16-FCN32s model.
3. **(3%)** Report mIoUs of two models on the validation set.
4. **(7%)** Show the predicted segmentation mask of “validation/0013_sat.jpg”, “validation/0062_sat.jpg”, “validation/0104_sat.jpg” during the early, middle, and the final stage during the training process of the improved model.
   - Tips: Given n epochs training, you could save the 1st, (n/2)-th, n-th epoch model, and draw the predicted mask by loading these saved models.

Outline

- Problem 1: Image classification
- Problem 2: Semantic segmentation
- Tools
- Submission
- Homework policy

Tool for Dataset

- **Download the dataset**
  - (Option 1) Manually download the dataset here

    https://drive.google.com/file/d/1LMIaOY8NSKWmGtbvTsjXcHVaYKnNN_9u

  - (Option 2) Run the bash script provided in the hw1 repository

    bash ./get_dataset.sh

Tools for Problem 2

- **mIoU**

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

- **Visualization**

  python3 viz_mask.py <--img_path xxxx_sat.jpg> <--seg_path xxxx_mask.png>

PyTorch Tutorial

- We provide some resources for someone who is not familiar with **PyTorch**
  - Introduction to PyTorch
  - Basic PyTorch classes
  - Sample colab notebook

Outline

- Problem 1: Image classification
- Problem 2: Semantic segmentation
- Tools
- Submission
- Homework policy

Submission

- Deadline: **2022/10/10 (Mon.) 23:59 (GMT+8)**
- Click the following link and sign in to your GitHub account to get your submission repository:

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

  - **You should connect your Github account to the classroom with your student ID**
  - If you cannot find your student ID in the list, please contact us (ntudlcv@gmail.com)
- By default, we will only grade your latest pushed commit **before the deadline** (**NOT** your last submission).
- Please e-mail the TAs if you'd like to submit another version of your repository specifying which commit to grade.

Submission

- Your GitHub repository **DLCV-Fall-2022/hw1-{GitHub_ID}** should include the following files:
  - hw1_1.sh
  - hw1_2.sh
  - hw1_download.sh or model checkpoints
  - hw1_<studentID>.pdf (e.g. hw1_r09942249.pdf)
  - your python files (Training & Inference code, both required)
- **DO NOT push the dataset to your repo**

Shell Script (Problem 1) - hw1_1.sh

- Provide a **script** to test images under the specified directory with your model, and save the classification results in the specified csv file.
- TAs will run your script as shown below:
  - bash hw1_1.sh $1 $2
    - $1: testing images directory with images named 'xxxx.**png**' (e.g. input/test_dir/)
    - $2: path of output csv file (e.g. output/pred.csv)
- The output csv file **must** have the same format as 'val_gt.csv'
- This section must be finished in **10 mins**, otherwise would be considered as a failed run.

Sample CSV Format (Problem 1)

- Predict class labels for all images
  - Output format: csv file
  - The first row must be: ‘filename, label’
  - You should only output the filename instead of the whole file path (i.e. Given the image path ‘input/test_dir/0_450.png’, you should only output ‘0_450.png’)

```
filename,label
0_450.png,0
0_451.png,0
0_452.png,0
0_453.png,0
0_454.png,0
```

Shell Script (Problem 2) - hw1_2.sh

- Provide a **script** to test images under the specified directory with your model, and save the segmentation results as images in the specified output directory.
- TA will run your code as shown below:
  - bash hw1_2.sh $1 $2
    - $1: testing images directory with images named 'xxxx.**jpg**' (e.g. input/test_dir/)
    - $2: output images directory (e.g. output/pred_dir/)
- You **should** name your output **segmentation mask** as 'xxxx.**png**'
- This section must be finished in **10 mins**, otherwise would be considered as a failed run.

Model Checkpoint

- If your model checkpoints are larger than GitHub's maximum capacity (50 MB), you could download and preprocess (e.g. unzip, tar zxf, etc.) them in hw1_download.sh.
  - TAs will run `bash hw1_download.sh` prior to any inference if the download script exists, i.e. it is **NOT** necessary to create a blank `hw1_download.sh` file.
- Do **NOT** delete your model checkpoints before the TAs release your score and before you have ensured that your score is correct.

Download Tutorial

- Please use **wget** to download the model checkpoints from cloud drive (e.g. Dropbox) or your working station.
  - You should use **-O argument** to specify the filename of the downloaded checkpoint.
  - Please refer to this Dropbox Guide for a detailed tutorial.
- Google Drive is a widely used cloud drive, so it is allowed to use **gdown** to download your checkpoints from your drive.
  - It is also recommended to use -O argument to specify the filename.
  - Remember to set the permission visible to public, otherwise TAs are unable to grade your submission, resulting in zero point.
  - If you have set the permission correspondingly but failed to download with **gdown** because of Google's policy, TAs will manually download them, no worries!!

Environment

- Ubuntu 20.04.1 LTS
- NVIDIA GeForce RTX 2080 Ti (11 GB)
- GNU bash, version 5.0.17(1)-release
- Python 3.8

Packages

- imageio==2.21.2
- matplotlib==3.5.3
- numpy==1.23.1
- Pillow==9.2.0
- scipy==1.9.1
- torch==1.12.1
- torchvision==0.13.1

Optional:

- gdown, tqdm, glob, yaml
- transformers==4.21.3
- timm==0.6.7
- scikit learn==1.1.2
- pandas

If you want to use any other packages, please email TA first.

Final Check

- Ensure your code can be executed successfully on **Linux** system before your submission.
- Use only **Python3** and **Bash** script conforming to our environment, do not use other languages (e.g. CUDA) and other shell (e.g. zsh, fish) during inference.
  - Use the command **`python3`** to execute your testing python files.
- You must **NOT** use commands such as **sudo, CUDA_VISIBLE_DEVICES** or other commands to interfere with the environment; **any malicious attempt against the environment will lead to zero point in this assignment**.
- You shall **NOT** hardcode any path in your python files or scripts, while the dataset given would be the absolute path to the directory.

Outline

- Problem 1: Image classification
- Problem 2: Semantic segmentation
- Tools
- Submission
- Homework policy

Deadline and Academic Honesty

- Deadline: **2022/10/10 (Mon.) 23:59 (GMT+8)**
- Late policy : Late homework submission will be deducted by **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 and result in an F for students with such misconduct.**

Code Modification

- If your code cannot be executed, you have a chance to make minor modifications to your code. After modifying your code,
  - If we can execute your code, you will receive a **30% penalty** in your model performance score.
  - If we still cannot execute your code, no points will be given.
- TAs will release the log of execution after grading, please check.
  - Email the TAs if something goes wrong in your submission.

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 for discussion
- Post your question to NTU COOL
- Contact TAs by e-mail: ntudlcv@gmail.com
- Find the easter eggs in the slide!

¡edoctihsrepaP

Dos and Dont's for the TAs

- Do NOT send private messages to TAs via Facebook or spam TAs email.
  - TAs are happy to help, but they are not your tutors 24/7.
- TAs will NOT debug for you (e.g., coding, environment, dependencies, etc. issues).
- If you cannot make the TA hours, please email the TAs for an appointment.

Final Reminder

- Please start working on this homework a**s early as possible**.

  The training may take a few hours on a GPU or days on CPUs.
- As mentioned in the first class, we are NOT responsible for any computation resources.
- Failure to conform to the aforementioned regulations will incur penalty.
- TAs are here to help. Discussions/Q&As are welcomed :)