DLCV HW3

TAs
ntu.dlcvta@gmail.com

Outline

- Sementic Segmentation
  - Task Definition
  - Fully Convolutional Network (e.g., VGG-16)
- Assignment
  - TODO
  - Implementation Details (Data, Pretrained-model)
  - Model Evaluation & Grading Policy
- Deadline & Homework policy
- Tutorial & Documentation
- Q & A

Outline

- Sementic Segmentation
  - Task Definition
  - Fully Convolutional Network (e.g., VGG-16)
- Assignment
  - TODO
  - Implementation Details (Data, Pretrained-model)
  - Model Evaluation & Grading Policy
- Deadline & Homework policy
- Tutorial & Documentation
- Q & A

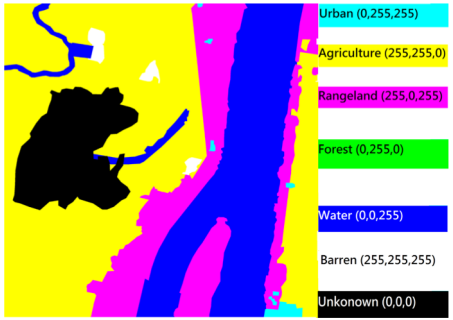
Semantic Segmentation

Task Definition

- In this assignment, you will need to perform semantic segmentation, which predicts a label to each pixel with CNN models.
  - Input : RGB image
  - Output : Semantic Segmentation/Prediction

Image

CNN



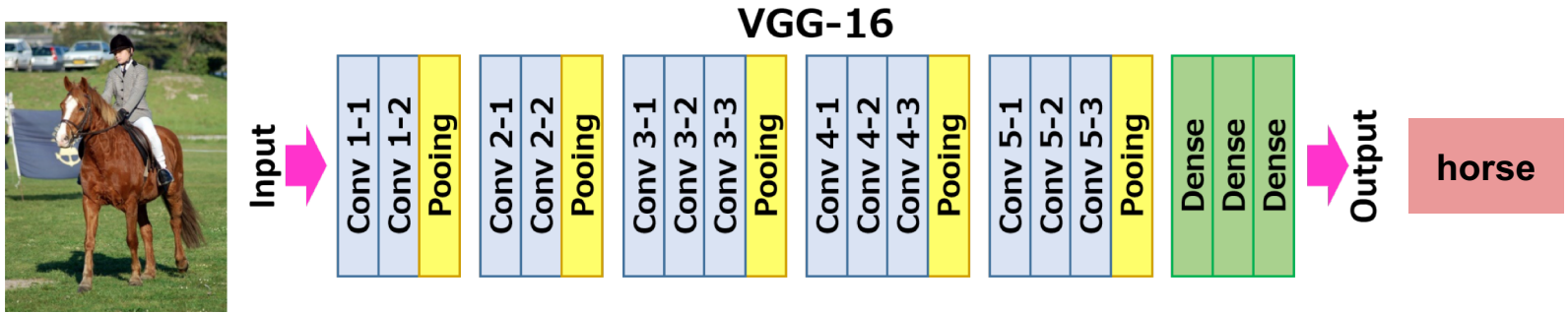
Semantic Segmentation Prediction

Outline

- Sementic Segmentation
  - Task Definition
  - Fully Convolutional Network (e.g., VGG-16)
- Assignment
  - TODO
  - Implementation Details (Data, Pretrained-model)
  - Model Evaluation & Grading Policy
- Deadline & Homework policy
- Tutorial & Documentation
- Q & A

Sementic Segmentation

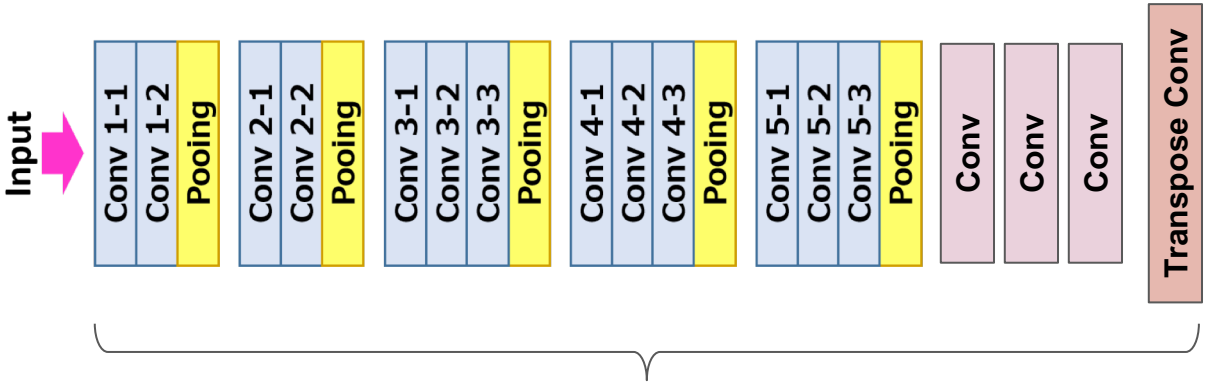
Convolutional Network - VGG16 for Image Classification



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

Sementic Segmentation

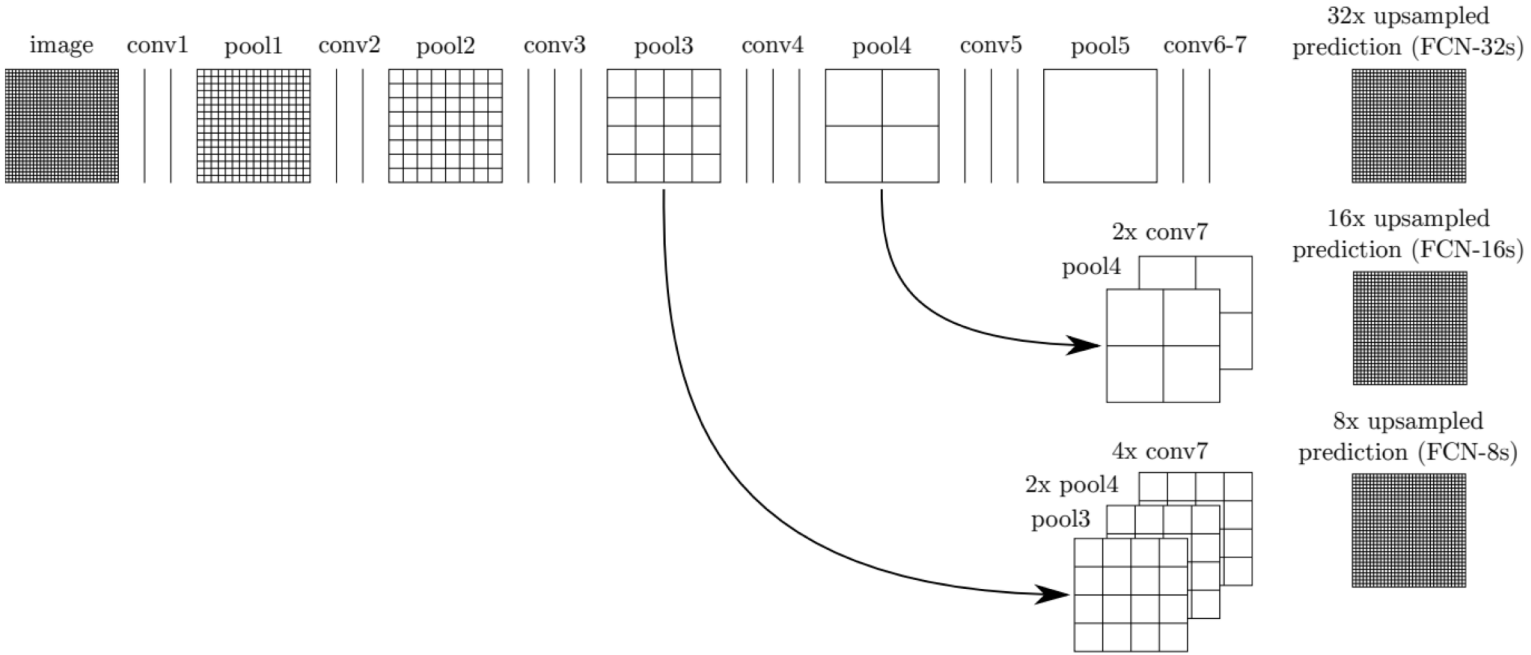
Fully Convolutional Network - FCN 32s



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

Sementic Segmentation

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



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

Outline

- Sementic Segmentation
  - Task Definition
  - Fully Convolutional Network (e.g., VGG-16)
- Assignment
  - To-Do
  - Implementation Details (Data, Pretrained-model)
  - Model Evaluation & Grading Policy
- Deadline & Homework policy
- Tutorial & Documentation
- Q & A

To-Do’s

- In this assignment, you will need to implement **two** segmentation models and provide discussions.
    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 a improved model to perform segmentation. The performance of this model should be better than that of the baseline model. You may choose any model different from VGG16-FCN32s, such as FCN16s, FCN8s, U-Net, SegNet, etc.

Outline

- Sementic Segmentation
  - Task Definition
  - Fully Convolutional Network (e.g., VGG-16)
- Assignment
  - To-Do
  - Implementation Details (Data, Pretrained-model)
  - Model Evaluation & Grading Policy
- Deadline & Homework policy
- Tutorial & Documentation
- Q & A

Implementation Details

Provided Files

- hw3-train-validation.zip [link]
- vgg16_weights_tf_dim_ordering_tf_kernels.h5 [link]
- mean_iou_evaluate.py [link]

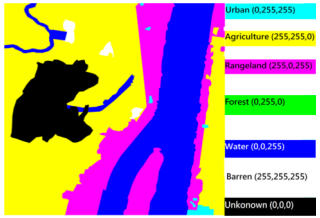
Implementation Details

Dataset Description

hw3-train-validation.zip
- train/
- validation/

RGB Image

FCN



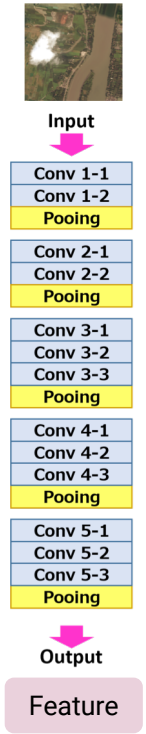
Semantic Segmentation Prediction

- train/
  - Contains 2313 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 pair
  - Naming rules are the same as train/
  - You **CANNOT** use validation data for training purposes.

Implementation Details

VGG16 Architecture

- You will 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)

Implementation Details

VGG16 Pretrained Model

- **Pretrained Weight :** vgg16_weights_tf_dim_ordering_tf_kernels.h5
- **Example** : loading pretrained weight (Keras)

```
from keras.layers import Input, Conv2D, MaxPooling2D
from keras.models import Model

# Build conv layers and pooling layers according to VGG16 model
img_input = Input(shape=(512, 512, 3))
x = Conv2D(64, (3, 3), activation='relu', padding='same', name='block1_conv1')(img_input)
x = Conv2D(64, (3, 3), activation='relu', padding='same', name='block1_conv2')(x)
x = MaxPooling2D((2, 2), strides=(2, 2), name='block1_pool')(x)
model = Model(img_input, x)

# Load pretrained weights accroding to layer names
weights_path = 'vgg16_weights_tf_dim_ordering_tf_kernels.h5'
model.load_weights(weights_path, by_name=True)
```

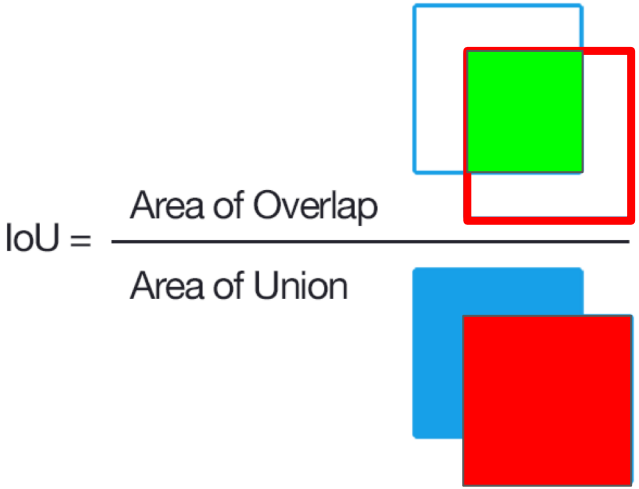
Outline

- Sementic Segmentation
  - Task Definition
  - Fully Convolutional Network (e.g., VGG-16)
- Assignment
  - TODO
  - Implementation Details (Data, Pretrained-model)
  - Model Evaluation & Grading Policy
- Deadline & Homework policy
- Tutorial & Documentation
- Q & A

Model Evaluation

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

    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**.



Model Evaluation

- Evaluation: **mean_iou_evaluate.py**
  - We provide the evaluation script for you
  - Usage:

    python3 mean_iou_evaluate.py < -g *ground_truth_directory* > < -p *prediction_directory* >
  - *ground_truth_directory* should contain ground truth mask images.
  - *prediction_directory* should contain your predicted mask images.
  - The names of your predicted mask images should follow that in the dataset.

    Ex. ‘0012_mask.png’

Grading Policy

- Baseline (50%)
  - validation set (20%)
  - testing set (30%)
- Report (55%)

- Bonus (5%)

Grading Policy

Baseline (50%)

- Implement baseline model **VGG16-FCN32s**
- **mIoU** score should be above the baseline score (valid: 0.635 / test: 0.625)
  - validation set (20%) / test set (30%, only TAs have the test set)
  - Note: baseline credit depends **only on VGG16-FCN32s** model.

Grading Policy

Report (55%) (report template link)

1. (5%) Print the network architecture of your VGG16-FCN32s model.
2. (10%) Show the predicted segmentation mask of “validation/0008_sat.jpg”, “validation/0097_sat.jpg”, “validation/0107_sat.jpg” during the early, middle, and the final stage during the training stage. (For example, results of 1st, 10th, 20th epoch)
3. (15%) Implement an improved model which performs better than your baseline model. Print the network architecture of this model.
4. (10%) Show the predicted segmentation mask of “validation/0008_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.
5. (15%) Report mIoU score of both models on the validation set. 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 reasoning.

Note: please follow the template when writing your report

Grading Policy

Bonus (5%)

- (5%) [bonus] Calculate the result of d/dw G(w):

objective function:

$$G(\boldsymbol{w}) = -\sum_n \left[t^{(n)} \log \mathrm{x}(\boldsymbol{z}^{(n)}; \boldsymbol{w}) + (1 - t^n) \log \left(1 - \mathrm{x}(\boldsymbol{z}^{(n)}; \boldsymbol{w})\right) \right] \geq 0$$

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} G(\boldsymbol{w})$$

choose the weights that minimise the network's surprise about the training data

$$\frac{\mathrm{d}}{\mathrm{d}\boldsymbol{w}} G(\boldsymbol{w}) = \sum_n \frac{\mathrm{d}G(\boldsymbol{w})}{\mathrm{d}x^{(n)}} \frac{\mathrm{d}x^{(n)}}{\mathrm{d}\boldsymbol{w}} = -\sum_n (t^{(n)} - x^{(n)}) \boldsymbol{z}^{(n)}$$

= prediction error x feature

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{w}} G(\boldsymbol{w})$$

iteratively step down the objective (gradient points up hill)

Outline

- Sementic Segmentation
  - Task Definition
  - Fully Convolutional Network (e.g., VGG-16)
- Assignment
  - TODO
  - Implementation Details (Data, Pretrained-model)
  - Model Evaluation & Grading Policy
- Deadline & Homework policy
- Tutorial & Documentation
- Q & A

Deadline & Homework Policy

- Report and source code deadline : **5/2 01:00 (GMT+8)**
- Late policy : Up to 3 free late days in a semester. 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.

Homework Policy - Submission

- **DLCV2018SPRING/hw3** on your GitHub repository should include the following files:
  - hw3_YourStudentID.pdf
  - hw3.sh (for VGG16-FCN32s model)
  - hw3_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.

Homework Policy - Submission

- 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.
- Dropbox tutorial: link

Homework Policy - Bash Script

- TA will run your code as shown below:
  - bash hw3.sh $1 $2
  - bash hw3_best.sh $1 $2
    - $1: testing images directory (images are named 'xxxx_sat.jpg')
    - $2: output images directory
- You **should** name your output images as 'xxxx_mask.png'
- Note that you should **NOT** hard code any path in your file or script
- Your testing code have to be finished in **10 mins.**

Homework Policy - Packages

- Python : 3.6
- Tensorflow : 1.6
- Keras : 2.0.7
- Pytorch : 0.3.1
- h5py : 2.7.1
- Numpy : 1.14.2
- Pandas : 0.22.0
- Matplotlib, Scikit-image, Pillow, Scipy, Python standard Lib.
- **E-mail or ask TA first if you want to import other packages.**

Outline

- Sementic Segmentation
  - Task Definition
  - Fully Convolutional Network (e.g., VGG-16)
- Assignment
  - TODO
  - Implementation Details (Data, Pretrained-model)
  - Model Evaluation & Grading Policy
- Deadline & Homework policy
- Tutorial & Documentation
- Q&A

Tutorials

- Keras
  - Keras Demo 1 - 李宏毅教授 [Link]
  - Keras Demo 2 - 李宏毅教授 [Link]
  - Keras Demo　- 莫煩 [Link]
- Tensorflow
  - Getting Start with Tensorflow - Official Website [Link]
  - CNN Tutorial - Official Website [Link]

Documentations

- Keras : link
- Tensorflow : link
- Pytorch : link

Reminder

The training process may take about **a day**, please start this homework as early as possible.

Outline

- Sementic Segmentation
  - Task Definition
  - Fully Convolutional Network (e.g., VGG-16)
- Assignment
  - TODO
  - Implementation Details (Data, Pretrained-model)
  - Model Evaluation & Grading Policy
- Deadline & Homework policy
- Tutorial & Documentation
- Q & A

Q & A

- If you have any question, you can:
  - Use TA hours (please check course website for time/location)
  - Contact TAs by e-mail (ntu.dlcvta@gmail.com)
  - Post your question under hw3 FAQ section in FB group (DLCV Spring 2018)
  - Useful website: link
  - **DO NOT** directly message TAs (we will ignore any direct message)