Deep Learning
Introduction to Deep Learning Models

Dr. – Ing. Gabriele Cavallaro
Postdoctoral Researcher
High Productivity Data Processing Group
Juelich Supercomputing Centre, Germany

Deep Learning in Remote Sensing: Challenges

June 6th, 2018
Juelich Supercomputing Centre, Germany, Germany
Outline of the Course

1. Introduction to Deep Learning
2. Fundamentals of Convolutional Neural Networks (CNNs)
5. Model Selection and Regularization
6. Fundamentals of Long Short-Term Memory (LSTM)
7. LSTM Applications and Challenges
8. Deep Reinforcement Learning
Outline

- Remote Sensing (RS) Overview
  - RS Background
  - Challenges for Deep Learning (DL) in RS
  - ISPRS 2D Semantic Labeling Contest
  - Data Augmentation
  - Transfer Learning
  - DL and Shallow Learning
  - Hierarchical Features of CNNs
Remote Sensing Background

[Diagram showing relationships between various components]
Remote Sensing

**Remote** (without physical contact) **Sensing** (measurement of information)

- The term **remote sensing** was first used in the United States in the 1950s by Ms. Evelyn Pruitt of the U.S. Office of Naval Research

- Measurement of radiation of different wavelengths reflected or emitted from distant objects or materials

- They may be categorized by class/type, substance, and spatial distribution
Platforms and Sensors

- **Platform**: selected according to the application

- **Active Sensor**: own source of illumination
  - Capture image in day and night
  - Any weather or cloud conditions

- **Passive Sensor**: natural light available
  - Great quality satellite imagery
  - Multispectral and Hyperspectral technology

Spatial Resolution

- The spatial minimum area discernable by a pixel (i.e., picture element)
- Influenced primarily by the sensor and the platform altitude

- Landscapes vary greatly in their spatial complexity
  - Some may be represented clearly at coarse levels of detail
  - Others are so complex that the finest level of detail is required
Spectral Resolution

- Remote sensing observes at varied wavelengths
- Ability to define fine wavelength intervals
- The finer the spectral resolution, the narrower the wavelength range of a particular band

Landsat (7-11 bands)  
AVIRIS (256 bands)

Spectrum of a mixture of three common minerals: kaolinite, Dolomite, Hematite

Radiometric Resolution

- Maximum number of brightness levels available
- Depends on the number of bits used in representing the energy recorded
- Higher radiometric resolution, sharper images
Temporal Resolution

- The time it takes for a satellite to complete one orbit cycle (revisit time)
- Depends on the satellite/sensor capabilities, swath overlap and latitude

Repeated imaging enables assessment of changes in the type or condition of surface features
Sentinel 2 Mission

- **Platform:** Twin polar-orbiting satellites, phased at 180° to each other
- **Temporal resolution** of 5 days at the equator in cloud-free conditions

<table>
<thead>
<tr>
<th>Spatial Resolution (m)</th>
<th>Band Number</th>
<th>S2A Central Wavelength (nm)</th>
<th>S2B Central Wavelength (nm)</th>
<th>S2A Bandwidth (nm)</th>
<th>S2B Bandwidth (nm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>2</td>
<td>496.6</td>
<td>492.1</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>560.0</td>
<td>559</td>
<td>45</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>664.5</td>
<td>665</td>
<td>38</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>835.1</td>
<td>833</td>
<td>145</td>
<td>133</td>
</tr>
<tr>
<td>20</td>
<td>5</td>
<td>703.9</td>
<td>703.8</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>740.2</td>
<td>739.1</td>
<td>18</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>782.5</td>
<td>779.7</td>
<td>28</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>8a</td>
<td>864.8</td>
<td>864</td>
<td>33</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1613.7</td>
<td>1610.4</td>
<td>143</td>
<td>141</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>2202.4</td>
<td>2185.7</td>
<td>242</td>
<td>238</td>
</tr>
<tr>
<td>60</td>
<td>1</td>
<td>443.9</td>
<td>442.3</td>
<td>27</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>945.0</td>
<td>943.2</td>
<td>26</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>1373.5</td>
<td>1378.9</td>
<td>75</td>
<td>78</td>
</tr>
</tbody>
</table>

~23 TB data stored per day

- Images for mapping land-use change, land-cover change biophysical variables
- Monitor the coastal and inland waters and help with risk and disaster mapping

Classification of Remote Sensing Images

- Perhaps the most common form of image interpretation

- Applications: environmental management, agricultural planning, health studies, climate and biodiversity monitoring, and land change detection
Update of Land-Cover Maps at Country Scale

- Important task for regularly monitoring the Earth’s surface
- Generation of reliable maps with spatial consistency is challenging
  - Data from several satellite orbit tracks, different dates and presence of clouds
- Cannot rely on field campaigns
  - Use existing databases to build the reference sets for supervised classification
Germany – 1 year of Sentinel 2 Data

- Sentinel-2 tile gridding is based on the NATO Military Grid Reference System
- Each tile covers an area of 100 km x 100 km (excluding overlapping edges of 9.8 km)
- Typical file size of all bands with a tile: ~700 MB
- Germany can be covered with 56 tiles

```
   Tile
```

- Time serie of tile images for 1 year (365 days / 5 days = 73 acquisitions)
- Data size to be processed: 73 acq. * 56 tiles * 700Mb = 2.72 TB

Other Applications

- Environmental assessment and monitoring: *Urban growth*


- Global change detection and monitoring: *Deforestation*

Challenges for Deep Learning in Remote Sensing
Deep Learning in Remote Sensing

- DL has risen to the top in numerous areas in the last years
  - Computer vision, speech recognition, etc.

- Deep learning is also taking off in RS

Exponential increase of the paper published


- RS possesses a number of unique challenges for deep learning
Main Challenges for DL in RS

- **Multimodal data:** geometries and content are completely different
  - From optical (multi- and hyperspectral), Lidar, and synthetic aperture radar (SAR) sensors

- **High temporal resolutions data**
  - Shift from individual image analysis to time-series processing

- **Big remote sensing data:** high spatial and more-so spectral dimensionality
  - Traditional DL systems operate on relatively small grayscale or RGB imagery

*SAR images: noisy data*  
*Optical data: From four to hundreds to of channels*
Limited Number of Remote Sensing Datasets

- The existing datasets have a number of limitations
  - Small scale of scene classes and the image numbers
  - Lack of image variations and diversity
  - Saturation of accuracy

- They severely limit the development of new deep learning-based methods

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Image per class</th>
<th>Scene classes</th>
<th>Total images</th>
<th>Spatial resolution (m)</th>
<th>Image sizes</th>
<th>Year</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC Merced Land-Use</td>
<td>100</td>
<td>21</td>
<td>2100</td>
<td>0.3</td>
<td>256x256</td>
<td>2010</td>
<td>[15]</td>
</tr>
<tr>
<td>WHU-RS19</td>
<td>~50</td>
<td>19</td>
<td>1005</td>
<td>up to 0.5</td>
<td>600x600</td>
<td>2012</td>
<td>[16]</td>
</tr>
<tr>
<td>RSSCN7</td>
<td>400</td>
<td>7</td>
<td>2800</td>
<td>0.3</td>
<td>400x400</td>
<td>2015</td>
<td>[17]</td>
</tr>
<tr>
<td>SAT-6</td>
<td>--</td>
<td>6</td>
<td>405000</td>
<td>1</td>
<td>28x28</td>
<td>2015</td>
<td>[18]</td>
</tr>
<tr>
<td>SIRI-WHU</td>
<td>200</td>
<td>12</td>
<td>2400</td>
<td>2</td>
<td>200x200</td>
<td>2016</td>
<td>[19]</td>
</tr>
<tr>
<td>RSC11</td>
<td>~100</td>
<td>11</td>
<td>1323</td>
<td>0.2</td>
<td>512x512</td>
<td>2016</td>
<td>[20]</td>
</tr>
<tr>
<td>Brazilian Coffee Scene</td>
<td>1438</td>
<td>2</td>
<td>2876</td>
<td>0.3</td>
<td>64x64</td>
<td>2016</td>
<td>[21]</td>
</tr>
<tr>
<td>NWPU-RESISC45</td>
<td>700</td>
<td>45</td>
<td>31500</td>
<td>~30 to 0.2</td>
<td>256x256</td>
<td>2016</td>
<td>[22]</td>
</tr>
<tr>
<td>AID</td>
<td>~300</td>
<td>30</td>
<td>10000</td>
<td>0.6</td>
<td>600x600</td>
<td>2016</td>
<td>[23]</td>
</tr>
<tr>
<td>EuroSAT</td>
<td>~2500</td>
<td>10</td>
<td>27000</td>
<td>10</td>
<td>64x64</td>
<td>2017</td>
<td>[24]</td>
</tr>
</tbody>
</table>

- ImageNet dataset: 14,197,122 images [25] ImageNet project
ISPRS 2D Semantic Labeling Contest
ISPRS 2D Semantic Labeling Contest

- 2D semantic segmentation that assigns labels to multiple object categories

- Acquired by airborne sensors
  - Very high resolution true ortho photo tiles
  - Digital surface models (DSMs) derived from dense image matching techniques

- Very heterogeneous appearance of objects
  - High intra-class variance and low inter-class variance

Vaihingen
Town with many detached buildings and small multi story buildings

[26] 2D Semantic Labeling Contest

Potsdam
Historic city with large building blocks, narrow streets and dense settlement structure
Vaihingen Dataset (1)

- 33 patches of different sizes with 9 cm spatial resolution
- Manually classified into six land cover classes
  - Impervious surfaces, Building, Low vegetation, Tree, Clutter/background

True orthophoto
Near infrared
Red
Green

DSM
One band
Grey levels mapped to DSM heights

Groundtruth

Lecture 3 - Deep Learning in Remote Sensing: Challenges
The groundtruth is provided for only 16 patches

For the remaining scenes is unreleased and used for evaluation of submitted results

“Participants shall use all data with ground truth for training or internal evaluation of their method”
Practical 1: Access JURECA and Find the Dataset
Vaihingen Dataset in JURECA

- Access JURECA: `ssh -X train???@jureca.fz-juelich.de`
- Data location: `/homea/hpclab/train001/data/vaihingen/`

```
[train002@jrl05 ~]$ ls /homea/hpclab/train001/data/vaihingen/
vaihingen_11.hdf5 vaihingen_1.hdf5 vaihingen_28.hdf5 vaihingen_37.hdf5
vaihingen_13.hdf5 vaihingen_21.hdf5 vaihingen_30.hdf5 vaihingen_3.hdf5
vaihingen_15.hdf5 vaihingen_23.hdf5 vaihingen_32.hdf5 vaihingen_5.hdf5
vaihingen_17.hdf5 vaihingen_26.hdf5 vaihingen_34.hdf5 vaihingen_7.hdf5
[train002@jrl05 ~]$ 
```

- HDF5 files creation:

```r
str = 'Vaihingen_1.hdf5';
h5write(str , '/x_1', cat(3, Near Infrared, Red, Green, Normalized DSM,NDVI))
h5write(str , '/y_1', Groundtruth)
h5write(str , '/m_1', Boundaries)
```
- Uncertainty associated with the boundary of objects in the groundtruth
- These boundaries are ignored for the evaluation
- Groundtruth images with boundary pixels (in a 3 pixel radius) provided
Normalized Difference Vegetation Index (NDVI)

- Create additional relevant features from the existing raw features in the data
- Increase the predictive power of the classifier
If annotated samples are available, the classifier parameters are learned in a supervised way.

How to estimate the generalization error: split the groundtruth into three disjoint sets.

**Performances**: usually more influenced by the amount and quality of the training samples (i.e., sampling design) rather than the classifier/model complexity.
Assess Classifier Performance

- **Training set**: used to train the model
  - How do we ensure that the model is not overfitting to the data in the training set?

- **Validation set**: used to validate the model during training
  - Its classification is based only on the model that is learnt from the training set
  - The model weights are updated based on this set
  - Help to adjust the hyperparameters (e.g., number of hidden layers, learning rate, etc.)

- **Test set**: used to test the model after it has been trained
Practical 2:
Generate your Copy of the Training and Validation Sets
Approach for Training and Validation Set Generation

- Generate dataset of 256x256 sized image patches

```python
def main(arguments):
    data_path = arguments[1]
    output_path = arguments[2]

    # files used for training:
    training_nums = [1, 3, 5, 7, 11, 13, 17, 21, 26, 28, 34, 37]

    # files used for validation:
    validation_nums = [30, 32]

    # generate and save the training and validation set:
    overlap = 0.6
```
Get the Code and Test the Python Environment

1. Get a copy of the folder `/homea/hpclab/train001/tools/resnet50-fcn`
   - Create a new folder in your local path $ ~/semseg
   - Copy $ cp -R /homea/hpclab/train001/tools/resnet50-fcn ~/semseg/

2. All modules and python packages have been already prepared
   - Just run -> $ module restore dl_tutorial
   - How was it setup?
     - $ module use /usr/local/software/jureca/OtherStages
     - $ ml Stages/Devel-2017a
     - $ ml GCC/5.4.0 MVAPICH2
     - $ ml TensorFlow/1.4.0-Python-2.7.13
     - $ pip install --user virtualenv
     - $ pip install --user h5py
     - $ pip install --user keras
     - $ pip install --user sklearn
     - $ module store dl_tutorial

3. Check if Keras is available
   - $ python
   - >>> import keras

   ![Using TensorFlow backend.]

Lecture 3 - Deep Learning in Remote Sensing: Challenges
Generate the Training and Validation Set

4. Use the function \`~/semseg/resnet50-fcn/data_io.py\`
   - If you run \$ python data_io.py

```
[train002@jrl12 resnet50-fcn]$ python data_io.py
/homea/hpclab/train002/.local/lib/python2.7/site-packages/h5py/__init__.py:36: FutureWarning: Converting int16 to int, future, it will be treated as `np.float64 == np.dtype(float).type`
    from _conv import registerConverters as _register_converters

***************************************************************************
Two paremeters need to be specified:
1. Location of the input image patches (e.g., /homea/hpclab/train001/data/vaihingen/ )
2. Location to write training and validation sets (e.g., /homea/hpclab/train002/semseg/vaihingen/ )
***************************************************************************
```

- Create a new folder where to save the sets \$ mkdir ~/semseg/vaihingen
- Run the function:
  - \$ python data_io.py /homea/hpclab/train001/data/vaihingen/ ~/semseg/vaihingen/
The Outcome

- The patches are assigned to the training and validation sets

```
$ python data_io.py /homea/hpclab/train001/data/vaihingen/ ~/semseg/vaihingen/
/homea/hpclab/train002/.local/lib/python2.7/site-packages/h5py/_init_.py:36: FutureWarning: Conversion of the second argument of issubdtype from 'float' to 'np.float64' is deprecated. In future, it will be treated as 'np.float64 == np.dtype(float).type'.
    from _conv import register_converters as _register_converters
/homea/hpclab/train001/data/vaihingen/vaihingen_1.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_3.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_5.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_7.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_11.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_13.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_17.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_21.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_26.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_28.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_34.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_37.hdf5
Generated 2083 samples!
/homea/hpclab/train001/data/vaihingen/vaihingen_30.hdf5
/homea/hpclab/train001/data/vaihingen/vaihingen_32.hdf5
Generated 368 samples!
```

- These sets will be used for training the network

```
$ pwd
/homea/hpclab/train002/semseg/vaihingen
$ ls
vaihingen_train.hdf5  vaihingen_val.hdf5  vaihingen_test.hdf5
```

Lecture 3 - Deep Learning in Remote Sensing: Challenges
Transfer Learning and Data Augmentation
Limited Remote Sensing Training Data

- RS applications have massive amounts of temporal and spatial data (e.g., Sentinel 2)
- But not enough labeled training samples, which usually don’t fully represent:
  - Seasonal variations
  - Object variation (e.g., plants, crops, etc.)
- Most online hyperspectral data sets have little-to-no variety

[30] Indian Pines dataset

- DL systems with many parameters require large amounts of training data
  - Else they can easily overtrain and not generalize well
- DL systems in CV use very large training sets
  - e.g., millions or billions of faces in different illuminations, poses, inner class variations, etc.
DL systems with limited training data

- Possible approaches to mitigate small training samples:

  1. Data augmentation
     - Affine transformations, rotations, small patch removal, etc.

  2. Transfer learning
     - Train on other imagery to obtain low-level to mid-level features

  3. Use ancillary data
     - Other sensor modalities (e.g., LiDAR, SAR, etc.)

  4. Unsupervised training
     - training labels not required
Need for a Large Amount of Training Data

- State of the art DL networks have parameters in the order of millions
  - The learning model needs a proportional amount of examples
  - The number of parameters should be proportional to the complexity of the task

<table>
<thead>
<tr>
<th></th>
<th>VGGNet</th>
<th>DeepVideo</th>
<th>GNMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Used For</td>
<td>Identifying Image Category</td>
<td>Identifying Video Category</td>
<td>Translation</td>
</tr>
<tr>
<td>Input</td>
<td>Image</td>
<td>Video</td>
<td>English Text</td>
</tr>
<tr>
<td>Output</td>
<td>1000 Categories</td>
<td>47 Categories</td>
<td>French Text</td>
</tr>
<tr>
<td>Parameters</td>
<td>140M</td>
<td>~100M</td>
<td>380M</td>
</tr>
<tr>
<td>Data Size</td>
<td>1.2M Images with assigned Category</td>
<td>1.1M Videos with assigned Category</td>
<td>6M Sentence Pairs, 340M Words</td>
</tr>
<tr>
<td>Dataset</td>
<td>ILSVRC-2012</td>
<td>Sports-1M</td>
<td>WMT’14</td>
</tr>
</tbody>
</table>

- The available dataset is taken in a limited set of conditions
  - Different orientation, location, scale, brightness etc.
Data Augmentation

- Train with additional synthetically modified data
- Techniques to artificially increase the size of the training set
- Make minor changes such as flips, translations and rotations to the existing dataset
- Employed to counteract overfitting

“A poorly trained neural network would think that these three tennis balls, are distinct, unique images”
Essential Assumption: Invariance

- Ability to recognize an object as an object, even when its appearance varies in some way
- It allows to abstract an object's identity from the specifics of the visual input
  - E.g., relative positions of the viewer/camera and the object.
- Well-trained CNNs can be invariant to translation, viewpoint, size or illumination

[Image: Translation Invariance, Rotation/Viewpoint Invariance, Size Invariance, Illumination Invariance]
Popular Augmentation Techniques

- Flip horizontally and vertically
- Rotate
- Scaled outward or inward
- Crop: random sample a section
- Translate: moving the image along the X or Y direction
- Add noise

Data augmentation is more challenging for remote sensing

- Images exist in a variety of conditions (e.g., different seasons)
- They cannot be accounted for by the above simple methods
Classification Results Assessment

Traditional classifiers are based on the assumption that training and test samples are generated from the same feature space and distribution.

Remote sensing data usually present heterogeneous feature spaces and distributions due to differences in acquisition or changes in the nature of the object observed.

Most of the statistical models are likely to fail the prediction of new samples.
Transfer of Knowledge

- Direct solution: rebuild from scratch the predictive model using new training samples

- However it is preferable to reduce the need for and effort in recollecting new samples

- Other solutions: transfer learning, domain adaptation and active learning approaches

- Exploit the knowledge acquired by the available reference samples for classifying new images acquired over different geographical locations at diverse times with different sensors
Practical 3: Submit two Training Jobs
Create a new folder where the trained model will be saved

1. Get a copy of the script `/homea/hpclab/train001/script/submit_train_resnet50_fcn.sh`

2. Modify the highlighted parts

```bash
#!/bin/bash -x
#SBATCH--nodes=1
#SBATCH--ntasks=1
#SBATCH--output=train_resnet50_fcn_out.%j
#SBATCH--error=train_resnet50_fcn_err.%j
#SBATCH--time=01:00:00
#SBATCH--mail-user=g.cavallaro@fz-juelich.de
#SBATCH--mail-type=ALL
#SBATCH--job-name=train_resnet50_fcn
#SBATCH--partition=gpus
#SBATCH --gres=gpu:1
#SBATCH--reservation=deep_learning

### location executable
RESNET50_FCN=/homea/hpclab/train002/semseg/resnet50-fcn/train_resnet50_fcn.py

module restore dl_tutorial

### submit
python $RESNET50_FCN /homea/hpclab/train002/semseg/data/
/homea/hpclab/train002/semseg/models/resnet50_fcn_weights.hdf5 True False

3. Submit: $ sbatch submit_train_resnet50_fcn.sh
```

Location of training and validation sets

Create a new folder where the trained model will be saved
Edit the Script and Submit the Second Training Job

1. Get a copy of the script
   `/homea/hpclab/train001/script/submit_train_resnet50_fcn_pretrained.sh`

2. Modify the highlighted parts

```bash
#!/bin/bash -x
#SBATCH--nodes=1
#SBATCH--ntasks=1
#SBATCH--output=train_resnet50_fcn_out.%j
#SBATCH--error=train_resnet50_fcn_err.%j
#SBATCH--time=01:00:00
#SBATCH--mail-user=g.cavallaro@fz-juelich.de
#SBATCH--mail-type=ALL
#SBATCH--job-name=train_resnet50_fcn
#SBATCH--partition=gpus
#SBATCH --gres=gpu:1
#SBATCH--reservation=deep_learning

### location executable
RESNET50_FCN=/homea/hpclab/train002/semseg/resnet50-fcn/train_resnet50_fcn.py
module restore dl_tutorial

### submit
python $RESNET50_FCN /homea/hpclab/train002/semseg/data/
/homea/hpclab/train002/semseg/models/resnet50_fcn_weights.hdf5 True True
```

Location of training and validation sets

Create a new folder where the trained model will be saved
Deep Learning and Shallow Learning
How to Extract Spatial Features?

- Very High Spatial Resolution images: huge amount of details

- Sub-metric resolution
- Allows for accurate analysis
- Objects with different scales and shapes
Traditional RS Pipeline

- Traditional feature extraction methods involve approaches to extract information
  - Based on spatial, spectral, textural, morphological content, etc.
- The features are designed for a specific task
  - e.g. Enhancing the spatial information (Self-Dual Attribute Profiles - SDAPs)

**Representative**: they contain salient structures of the input image

**Non-redundant**: same objects are present only in one or few levels of the SDAP

Lecture 3 - Deep Learning in Remote Sensing: Challenges
Deep Learning and Shallow Learning

- Shallow learning: learning networks that usually have at most one to two layers
  - E.G. Support Vector Machines (SVMs)
  - They compute linear or nonlinear functions of the data (often hand-designed features)

- DL means a deeper network with many layers of non-linear transformations
  - No universally accepted definition of how many layers constitute a “deep” learner
  - Typical networks are typically at least four or five layers deep.
Hierarchical Features of CNNs
Pyramid and CNNs Analogy

- Multi-scale signal representation
- An image is subject to repeated smoothing and subsampling
- Used for doing tasks at multiple scales

**Gaussian pyramid**
- Subsequent images are weighted and scaled down
- Each pixel is local average of the neighborhood on a lower level

**Laplacian pyramid**
- It saves the difference image of the blurred versions between each levels.

![Gaussian Pyramid](image1)

![Laplacian Pyramid](image2)
Deep Networks Learn Hierarchical Feature Representations

[38] H. Lee et al.
Online: https://www.oneonta.edu/faculty/baumanpr/geosat2/RS%20History%20II/RS-History-Part-2.html


Online: http://grindgis.com/remote-sensing/active-and-passive-remote-sensing

Online: http://uregina.ca/piwowari/Satellites/Hyperspectral.html

Online: https://earthobservatory.nasa.gov/IOTD/view.php?id=5145

Online: https://sentinel.esa.int/web/sentinel/missions/sentinel-2

[7] CORINE land cover
Online: https://land.copernicus.eu/pan-european/corine-land-cover

Online: http://www.cesbio.ups-tlse.fr/multitemp/?p=11778

[9] S2 prototype LC map at 20m of Africa 2016
Online: http://2016africalandcover20m.esrin.esa.int/

Online: https://en.wikipedia.org/wiki/Military_Grid_Reference_System

Online: https://manilabydaniellaandisabel.weebly.com/location-and-characteristics.html

Online: https://www.satimagingcorp.com/gallery/more-imagery/aster/aster-deforestation-bolivia/
Lecture Bibliography (2)

- [25] ImageNet project
Online: http://image-net.org/about-overview
Lecture Bibliography (3)

- [26] 2D Semantic Labeling Contest
  Online: http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html
- [27] Normalized Difference Vegetation Index (NDVI)
  Online: http://www.agasyst.com/portals/NDVI.html
- [28] NDVI & Classification
- [30] Indian Pines dataset: 220 Band AVIRIS Hyperspectral Image
  Online: https://purr.purdue.edu/publications/1947/1
- [31] Data Augmentation - How to use Deep Learning when you have Limited Data
  Online: https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced
- [32] Invariance property
  Online: https://i.stack.imgur.com/iY5n5.png
- [35] Pyramid: image processing
  Online: https://en.wikipedia.org/wiki/Pyramid_(image_processing)#Gaussian_pyramid
- [36] The Laplacian Pyramid
  Online: https://hypjudy.github.io/2017/05/10/panorama-image-stitching/
- [37] Typical CNN
  Online: https://commons.wikimedia.org/wiki/File:Typical_cnn.png