JÜLICH Forschungszentrum | JÜLICH SUPERCOMPUTING CENTRE

The Influence of Sampling Methods on Pixel-Wise Hyperspectral Image Classification with 3D Convolutional Neural Networks

Julius Lange[1], Gabriele Cavallaro[2], Markus Götzl[2,3], Ernir Erlingsson[3], Morris Riedel[2,3]

[1] *Humboldt University of Berlin*

[2] *Jülich Supercomputing Centre, Forschungszentrum Jülich*

[3] *School of Engineering and Natural Sciences, University of Iceland*

IGARSS 2018

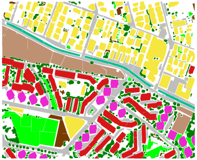
Outline

Outline

Remote Sensing Image Classification

Supervised Learning

- Annotated samples are available
- Supervised optimization of the classifier parameters
- How to obtain reliable semantic maps from unseen data?



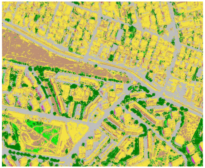
Spectral-Spatial Classifiers

The Value of Spatial Information

- Spectral classifiers: can’t deal with high spatial resolution images
- Exploit the spatial autocorrelation of data
- Improving the understanding of remote sensing images

Data set - true color image
VHR panchromatic (0.6*m*)
Multispectral (R,G,B and NIR)

Buildings	Blocks	Roads
Light Train	Vegetation	Trees
Bare Soil	Soil	Tower



spectral features

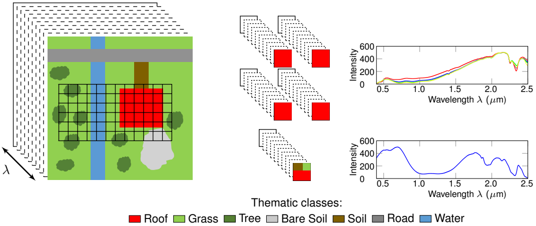
spectral + **spatial features**

Spatial Autocorrelation

Remote Sensing Data

- Spectral dependence degree between a pixel and its neighbors
- Measure of statistical separability between spatial objects

1. Intrinsic property → types of land cover classes
2. Spatial resolution → pixel's size
3. Pre-processing → feature engineering, CNNs, etc.



Accuracy Assessment

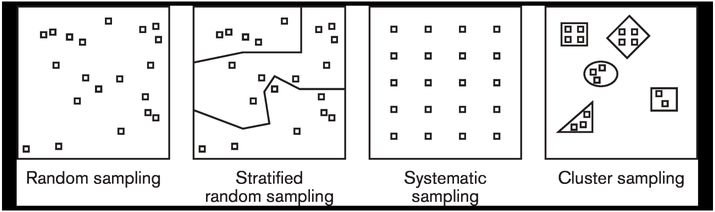
Sampling Step

- Influenced by:
  - The model of the classifier
  - The sampling scheme

- Groundtruth split into three disjoint sets: *training*, *validation* and *test*

- Sampler determines amount and distribution of samples across the scene

- It can significantly affect the test phase



Motivation

What this Work is About

- Practice of training and validating new classifiers within a single image
- Groundtruth split with random sampling
- It was a natural choice for spectral classifier
- Could already violate the independence assumption (bias train and test sets)
  - Inevitable spatial autocorrelation between adjacent pixels
  - Direct neighboring or nearby pixels present in both train and test sets
  - Spatial closeness: information from one set may leak into the respective other

Motivation

What this Work is not About

- Existing datasets have a number of limitations
  - Lack of image variations and diversity
  - Saturation of accuracy

Dataset and Reference	Number of uses
IEEE GRSS 2013 Data Fusion Contest[338]	4
IEEE GRSS 2015 Data Fusion Contest[339]	1
IEEE GRSS 2016 Data Fusion Contest[340]	2
Indian Pines[341]	27
Kennedy Space Center[342]	8
Pavia City Center[343]	13
Pavia University[343]	19
Salinas[344]	11
Washington DC Mall[345]	2

Open-source Hyperspectral datasets used for DL papers, *John Ball, 2017* [1]

Motivation

What this Work is not About

- Transfer learning, domain adaptation and active learning
- Recent advancement in EO benchmark data creation

Dataset	Image per class	Scene classes	Total images	Spatial resolution (m)	Image sizes	Year	Refernce
UC Merced Land-Use	100	21	2100	0.3	256x256	2010	[2]
WHU-RS19	50	19	1005	up to 0.5	600x600	2012	[3]
RSSCN7	400	7	2800	–	400x400	2015	[4]
SAT-6	–	6	405000	1	28x28	2015	[5]
Brazilian Coffee Scene	1438	2	2876	–	64x64	2016	[6]
SIRI-WHU	200	12	2400	2	200x200	2016	[7]
NWPU-RESISC45	700	45	31500	30 to 0.2	256x256	2016	[8]
AID	300	30	10000	0.6	600x600	2017	[9]
EuroSAT	2500	10	27000	10	64x64	2017	[10]
RSI-CB128	36000	–	45	3	128x128	2017	[11]
RSI-CB256	24000	–	35	0.3	256x256	2017	[11]

- Yet not enough (e.g., Imagenet with 14,197,122 images)
- Data variation between train and application phase remains in place

Outline

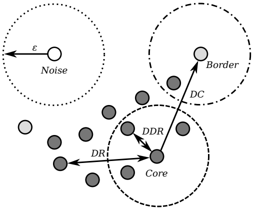
Aims of the Approach

- Capture the full spectral variation of the image
- Reduces overlap between train and test samples due to spatial processing
  1. Extract larger contiguous regions using the class labels
  2. Distribute them disjointly between the train and test set
- A bias, if present at all, would then only be relevant at the outer edges of such a region, but not for the inner pixels
- More objective and accurate evaluation

DBSCAN

Density-Based Spatial Clustering for Applications with Noise

- Can be used for computing n-connected components in an image
- They can have gaps (ε-sizes), filtered by (*minPoints*-area)
- Able to detect arbitrarily shaped clusters
- Don't need to know the number of clusters a priori



$minPoints = 4$

- Cluster core: point that contains within a spatial search radius ε at least a certain number of neighboring points *minPoints*

Extraction of Contiguous Regions

- Cluster the coordinates of pixels of each of the individual classes
- Each cluster corresponds to one of the identified region of that class
- DBSCAN performs n-connected-component-labeling
  - $\varepsilon = \{1, \sqrt{2}\}$ for 4- and 8-connectivity, respectively
- *minPoints* serves as threshold for potential gaps
  - i.e. If *minPoints* $<$ num. of connected components+1 $\rightarrow$ increasingly larger gaps

Regions Distribution Between the Train and Test Set

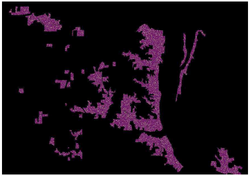
Not Efficient Approaches

- Regions could be randomly assigned to either one of the two sets
- However, num. of regions $<<$ num. of pixels within each region
- The likelihood of selecting an imbalanced train set rises
  - The set might not contain patterns present in the test set
- Each region could be subdivided into two disjoint parts
  - E.g., the “top” and “bottom”
- However, the number of biased pixels increases along the partition boundary

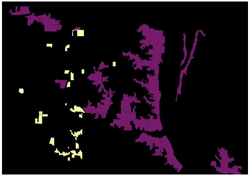
Regions Distribution Between the Train and Test Set

Proposed Approach

- *metric* function assigns to all regions of a class a partially ordered ranking
- It indicates their assignment priority to the train or test set
- Regions are added to a set until the pixels class split fraction is reached
- The region causing an over pixels assignment is split into two sub-regions



Random sampling



Cluster sampling (area)

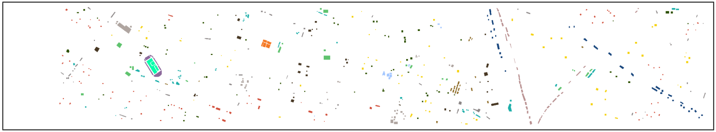
Cluster sampling (StdDev)

Outline

Dataset

University of Houston - 2013 GRSS data fusion contest

Hyperspectral image (144 spectral bands with 2.5m spatial resolution)

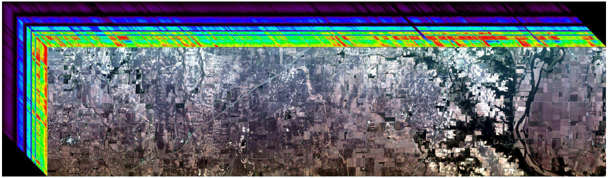


Thematic classes:

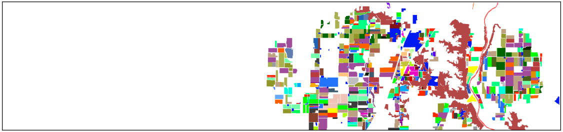
Healthy Grass (1251)	Trees (1244)	Residential (1268)	Highway (1227)	Parking Lot 2 (469)
Stressed Grass (1254)	Soil (1242)	Commercial (1244)	Railway (1235)	Tennis Court (428)
Synthetic grass (697)	Water (325)	Road (1252)	Parking Lot 1 (1233)	Running Track (660)

Dataset

Indian Pines - AVIRIS - NASA (1992)



Hyperspectral image (220 spectral bands with 20m spatial resolution)



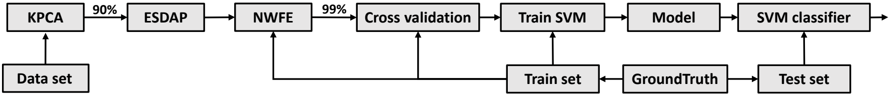
Thematic classes:

BareSoil (57)	Corn-MinTill (1049)	Hay (1128)	Soybeans-NS (1110)	Soybeans-NaTill (2157)	Buildings (17195)
Soybeans-CleanTill (5074)	Soybeans-NaTill-EW (2533)	Hay (pre.) (2185)	Corn-MinTill-EW (5629)	Concrete/Asphalt (69)	Corn-MinTill-NS (8862)
Hay-Alfalfa (2258)	Soybeans-CleanTill (pre.) (2726)	Soybeans-NaTill-NS (929)	Corn (17783)	Corn-Natill (4381)	Lake (224)
Soybeans-CleanTill-EW (11802)	Soybeans-NaTill-Drilled (8731)	Corn (presumed) (158)	Corn-Natill-EW (1206)	NotCropped (1940)	Soybeans-CleanTill-NS (10387)
Swampy Area (583)	Corn-EW (514)	Corn-Natill-NS (5685)	Oats (1742)	Soybeans-CleanTill-Drilled (2242)	River (3110)
Corn-NS (2356)	Fescue (114)	Oats (pre.) (335)	Soybeans-CleanTill-Weedy (543)	Trees (pre.) (580)	Corn-CleanTill (12404)
Grass (1147)	Orchard (39)	Soybeans-MinTill (15118)	Wheat (4979)	Corn-CleanTill-EW (26486)	Grass/Trees (2331)
Pasture (10386)	Soybeans-Drilled (2667)	Woods (63562)	Corn-CleanTill-NS (39678)	Grass/Pasture-mowed (19)	Pond (102)
Soybeans-MinTill (1832)	Unknown (144)	Corn-CleanTill-NS-Irrigated (800)	Grass/Pasture (73)	Soybeans (9391)	Soybeans-MinTill-Drilled (8098)
Corn-CleanTill-NS (pre.) (1728)	Grass-runway (37)	Soybeans (pre.) (894)	Soybeans-MinTill-NS (4953)		

Experimental Setup (1)

Feature Engineering Steps Combined with Support Vector Machine

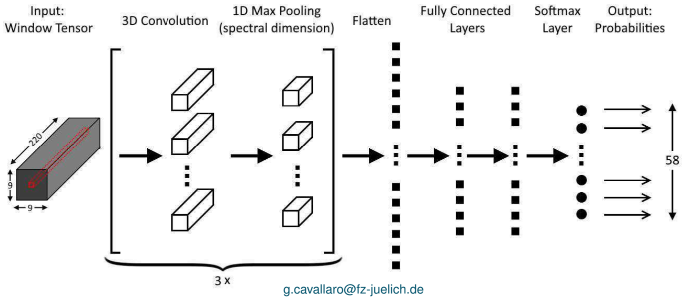
- Data dimensionality reduction
  - Kernel Principal Component Analysis (KPCA)
- Spatial information enhancement
  - Extended Self-Dual Attribute Profiles (ESDAPs)
- Feature extraction
  - Nonparametric Weighted Feature Extraction (NWFE)



Experimental Setup (2)

3D Convolutional Neural Network

Feature	Representation / Value
Conv. Layer Filters	48, 32, 32
Conv. Layer Filter size	$(3,3,5)$, $(3,3,5)$, $(3,3,5)$
Pooling size	$(1,1,3)$, $(1,1,3)$, $(1,1,2)$
Dense Layer Neurons	128, 128
Activation Functions	rectified linear unit (ReLU)
Loss Function	mean-squared error (MSE)
Optimization	stochastic gradient descent (SGD)
Training Epochs	600
Batch Size	50
Learning Rate	1.0
Learning Rate Decay	5×10^{-6}



Experimental Results

University of Houston

Sampling method	Feature Engineering with SVM				3DCNN				Metric
	Training set size (%)								
	10	30	60	90	10	30	60	90	
Random	97.68 (0.09)	99.62 (0.03)	99.85 (0.04)	99.90 (0.07)	96.03 (0.42)	98.97 (0.13)	99.51 (0.23)	99.75 (0.14)	OA
	97.73 (0.05)	99.63 (0.02)	99.87 (0.04)	99.96 (0.06)	95.26 (0.49)	98.86 (0.12)	99.40 (0.37)	99.73 (0.24)	AA
	97.54 (0.06)	99.59 (0.02)	99.84 (0.04)	99.89 (0.08)	95.70 (0.46)	98.89 (0.14)	99.47 (0.25)	99.73 (0.16)	Kappa
	97.76 (0.14)	99.61 (0.03)	99.83 (0.04)	99.91 (0.07)	95.60 (0.45)	98.93 (0.13)	98.20 (2.75)	99.74 (0.19)	F1
Size $\varepsilon = \sqrt{2}$ $minPoints = 9$	50.15	69.49	79.19	82.89	53.90 (2.86)	75.50 (0.66)	83.87 (1.33)	87.06 (0.51)	OA
	50.15	69.50	79.21	82.94	59.39 (3.88)	76.59 (0.59)	82.95 (1.13)	86.55 (1.57)	AA
	46.22	67.07	77.52	81.48	50.41 (3.07)	73.50 (0.71)	82.55 (1.44)	86.01 (0.56)	Kappa
	54.24	70.94	78.95	80.67	53.84 (4.05)	77.18 (1.03)	82.48 (1.96)	85.60 (1.39)	F1
StdDev $\varepsilon = \sqrt{2}$ $minPoints = 9$	63.47	66.62	74.01	80.36	58.50 (0.71)	58.37 (0.69)	70.59 (0.47)	79.36 (2.07)	OA
	63.48	66.62	74.02	80.41	60.15 (2.36)	62.38 (2.33)	72.24 (0.27)	80.95 (2.09)	AA
	60.59	63.97	71.90	78.77	55.19 (0.79)	55.19 (0.76)	68.26 (0.49)	77.68 (2.24)	Kappa
	64.50	67.29	77.38	77.11	55.71 (3.33)	58.74 (2.28)	72.38 (0.50)	79.39 (1.91)	F1

- Random sampling (SVM,3DCNN): average and stddev of five generated training sets
- Cluster sampling (SVM): single run (deterministic sampler)
- Cluster sampling (3DCNN): average and stddev of five random seeds used for the weights initialization

Experimental Results

Indian Pines

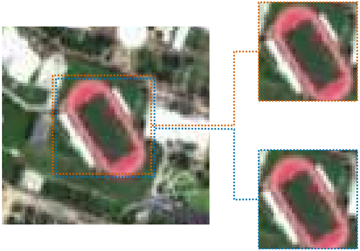
Sampling method	Feature Engineering with SVM				3DCNN				Metric
	Training set size (%)								
	10	30	60	90	10	30	60	90	
Random	77.83 (0.12)	84.84 (0.09)	87.78 (0.03)	89.10 (0.06)	90.32 (0.89)	96.59 (0.15)	97.89 (0.22)	98.34 (0.45)	OA
	77.83 (0.12)	84.84 (0.09)	87.78 (0.03)	89.10 (0.06)	nan	nan	nan	nan	AA
	75.98 (0.13)	83.61 (0.10)	86.79 (0.03)	88.22 (0.06)	89.54 (0.97)	96.32 (0.16)	97.73 (0.24)	98.21 (0.49)	Kappa
	66.75 (0.50)	76.14 (0.35)	79.91 (0.32)	81.63 (0.05)	70.60 (2.55)	82.39 (1.83)	81.08 (2.33)	81.98 (2.60)	F1
Size $\varepsilon = \sqrt{2}$ $minPoints = 9$	16.76	17.36	23.77	47.29	28.93 (1.47)	27.92 (1.19)	20.15 (1.65)	33.15 (1.24)	OA
	16.76	17.36	23.77	47.29	19.11 (0.00)	16.16 (0.00)	15.26 (0.00)	nan	AA
	8.33	8.40	14.71	42.75	24.64 (1.36)	23.01 (1.14)	15.59 (1.53)	28.37 (1.22)	Kappa
	nan	nan	nan	nan	14.55 (0.99)	14.47 (1.14)	13.14 (1.83)	18.22 (1.52)	F1
StdDev $\varepsilon = \sqrt{2}$ $minPoints = 9$	17.57	17.05	23.45	43.61	18.38 (0.78)	31.09 (0.47)	32.15 (1.23)	40.56 (1.65)	OA
	17.57	17.05	23.45	43.61	11.79 (0.00)	17.14 (0.00)	21.77 (1.49)	31.37 (0.33)	AA
	6.47	9.86	15.67	39.18	14.01 (0.74)	26.15 (0.41)	27.45 (1.19)	36.68 (1.79)	Kappa
	nan	nan	nan	nan	8.61 (0.60)	13.47 (0.94)	18.56 (0.95)	29.41 (2.81)	F1

- Random sampling (SVM,3DCNN): average and stddev of five generated training sets
- Cluster sampling (SVM): single run (deterministic sampler)
- Cluster sampling (3DCNN): average and stddev of five random seeds used for the weights initialization

Random Sampling Evaluation

Independence Assumption Violated

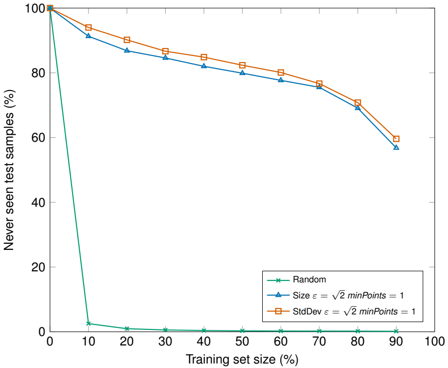
- The pattern recognition problem degrades to an almost memorization issue
- Classifying the same pixel class in the test set based on the previously seen similar instance in the training data is very likely
- Inherent to a variety of machine learning classifiers (SVMs, CNNs, etc.)



(3DCNN: insufficient offset between patches)

Overlap between train and test data

3DCNN - Indian Pines Dataset



Outline

Conclusions

- Random sampling introduces systematic bias
- Issue for spectral-spatial classifiers (e.g., processing pipelines, CNNs, etc.)
- More dependence between train and test samples leads to higher accuracies
- Proposed controlled sampling approach based on DBSCAN clustering algorithm
- Easy definition of contiguous regions and train-test-set assignment prioritization
- Accuracies on unseen test data closer to an actual out-of-sample performance

The End

Thank you for your attention.

Code available at:
https://github.com/Markus-Goetz/cluster-sampling

This research has been supported in part by the SIMDAS project and in part by the DEEP-EST project, which have received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreements No. 763558 and No. 754304, respectively.

Simulation & Data Science Center of Excellence

DEEP Projects

[1] J. Ball, D. Anderson, and C. S. Chan, “A Comprehensive Survey of Deep Learning in Remote Sensing: Theories, Tools and Challenges for the Community,” *SPIE Journal of Applied Remote Sensing (JARS), Special Section on Feature and Deep Learning in Remote Sensing Applications*, Sep 2017.

[2] “Dataset: UC Merced Land Use - Computer vision lab.” http://weegee.vision.ucmerced.edu/datasets/landuse.html, 2010.

[3] “DataSet: WHU-RS19.” http://www.xinhua-fluid.com/people/yangwen/WHU-RS19.html, 2012.

[4] “DataSet: RSSCN7.” https://www.dropbox.com/s/j80iv1a0mvhonsa/RSSCN7.zip?dl=0, 2015.

[5] “DataSet: SAT-6 airborne.” http://csc.lsu.edu/~saikat/deepsat/, 2015.

[6] “DataSet: Brazilian Coffee Scenes.” http://www.patreo.dcc.ufmg.br/2017/11/12/brazilian-coffee-scenes-dataset/, 2015.

[7] “DataSet: SIRI-WHU.” http://www.lmars.whu.edu.cn/prof_web/zhongyanfei/e-code.html, 2016.

[8] “DataSet: NWPU-RESISC45.” http://www.escience.cn/people/JunweiHan/NWPU-RESISC45.html, 2016.

[9] “DataSet: AID.” http://www.lmars.whu.edu.cn/xia/AID-project.html, 2017.

[10] “DataSet: EuroSAT.” http://madm.dfki.de/downloads, 2017.

[11] “DataSet: RSI-CB.” https://github.com/lehaifeng/RSI-CB, 2017.