UMR IRISA

Institut de Recherche en Informatique et Systèmes Aléatoires

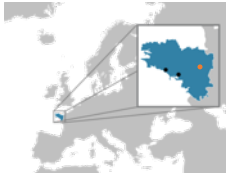
UYGU 2021 EARTH OBSERVATION APPLICATION SUMMER SCHOOL DEEP LEARNING FOR REMOTE SENSING

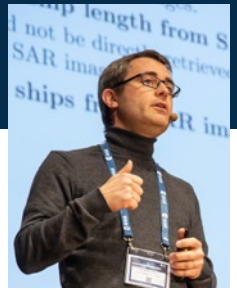
Sébastien Lefèvre

Deep Learning in Remote Sensing: **Good Practices** and **Solutions for Complex Data**

IRISA, University of South Brittany, Vannes, France – sebastien.lefevre@irisa.fr

SHORT SELF-INTRODUCTION





Who am I?

- Full Professor at University of South Brittany: www.univ-ubs.fr
- Founder and former head of the OBELIX group (25 researchers on AI4EO) at IRISA: www.irisa.fr/obelix
- Chair of the GeoData Science track
  EMJMD Copernicus Master in Digital Earth: www.master-cde.eu
- Co-chair of the MACLEAN workshop series at ECML-PKDD: sites.google.com/view/maclean21/

CONTENTS OF THE LECTURES

- Deep Learning for Remote Sensing… see previous lectures (they were great!)
  - Principles of Deep Learning
  - Applications to EO




- Lecture #1: Good practices
  - Training
  - Evaluation

- Lecture #2: Solutions for Complex Data
  - Focus on main EO tasks:
    semantic segmentation, object detection, change detection
  - Focus on some complex data:
    LiDAR, SAR, multi-view imagery, low-resolution data

CONTENTS OF THE LECTURES

- Deep Learning for Remote Sensing... see previous lectures (they were great!)
  - Principles of Deep Learning
  - Applications to EO

- Lecture #1: Good practices
  - Training
  - Evaluation

N. Audebert, B. Le Saux, S. Lefèvre.
Deep Learning for Classification of Hyperspectral Data: A Comparative Review.
IEEE Geoscience and Remote Sensing Magazine, 7(2):159-173, 2019

- Lecture #2: Solutions for Complex Data
  - Focus on main EO tasks:
    semantic segmentation, object detection, change detection
  - Focus on some complex data:
    LiDAR, SAR, multi-view imagery, low-resolution data

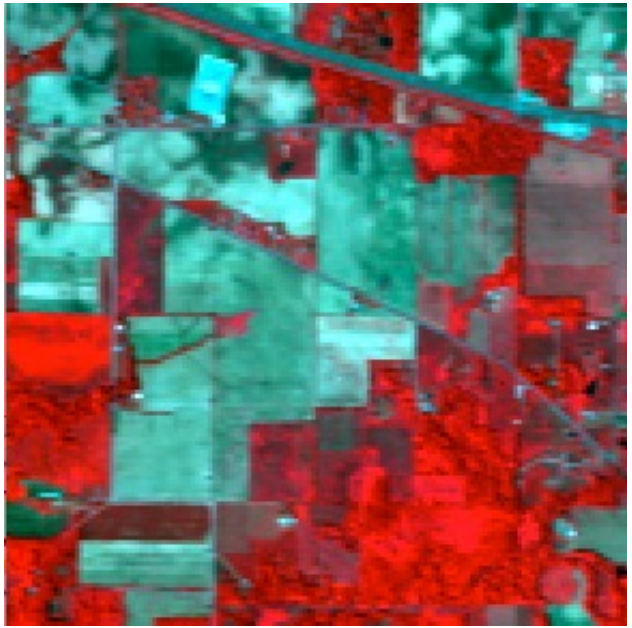
EVALUATING A DEEP NETWORK IS STRAIGHTFORWARD

Let us consider the most common land use / land cover classification problem:

1. Build a predictive model
   - design or choose a deep network
   - train it using a train set (+ a validation set to tune hyperparameters)
2. Assess the predictive model
   - using standard metrics (e.g. OA, AA, Kappa, etc.)
   - on some unseen data (a.k.a. the test set)

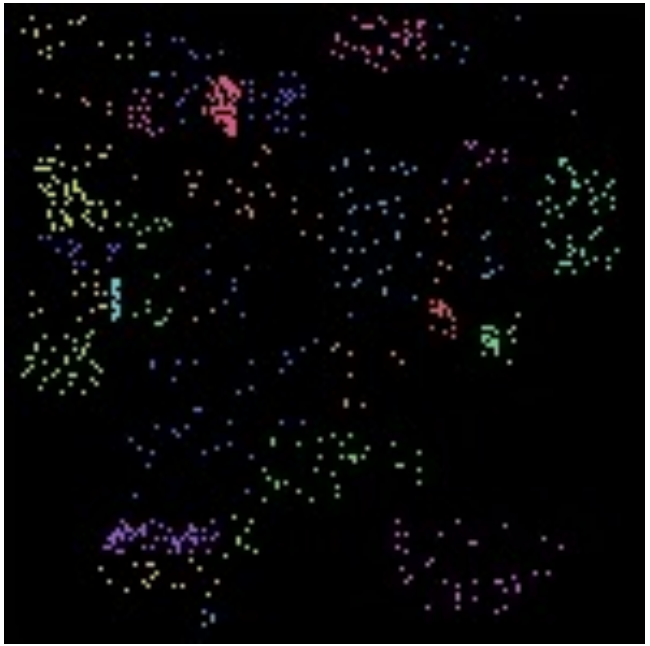
If comparison with other networks is sought, consider public datasets

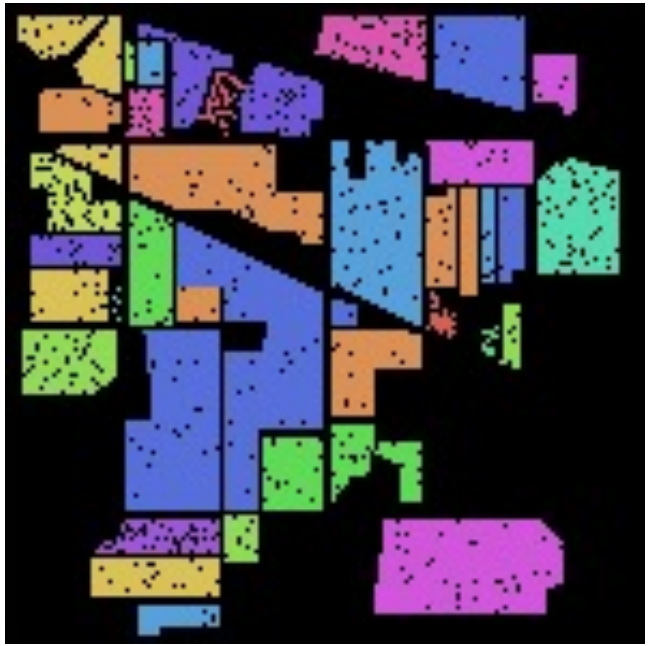
- RGB/Multispectral: ISPRS Vaihingen & Potsdam, …
- Hyperspectral: Indian Pines, Pavia U & C, Houston, …



Indian Pines, 145 x 145 pixels, 224 spectral bands, 16 classes, 10,249 labels

A bit old (1992), yet widely used (1320 publications since 2020)

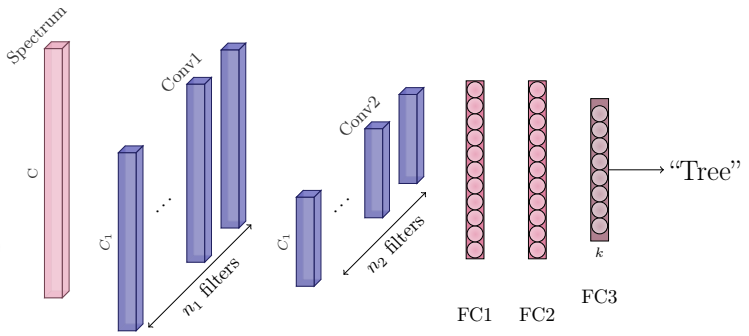


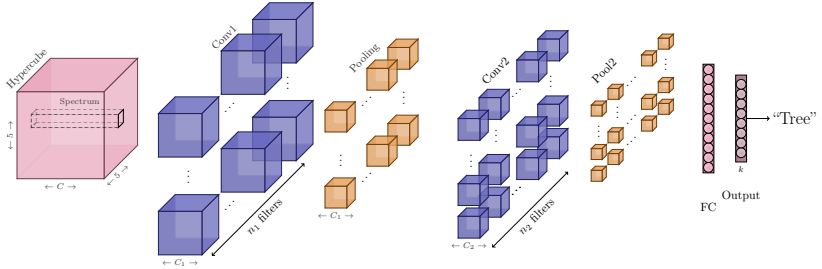


A random split of Indian Pines into train / test sets

Just an example, since no official train/test split is provided

Model	Accuracy
Nearest-Neighbour	75.63
1D CNN (Hu et al, 2015)	90.16
RNN (Mou et al, 2017)	85.7
3D CNN (Li et al, 2017)	99.07





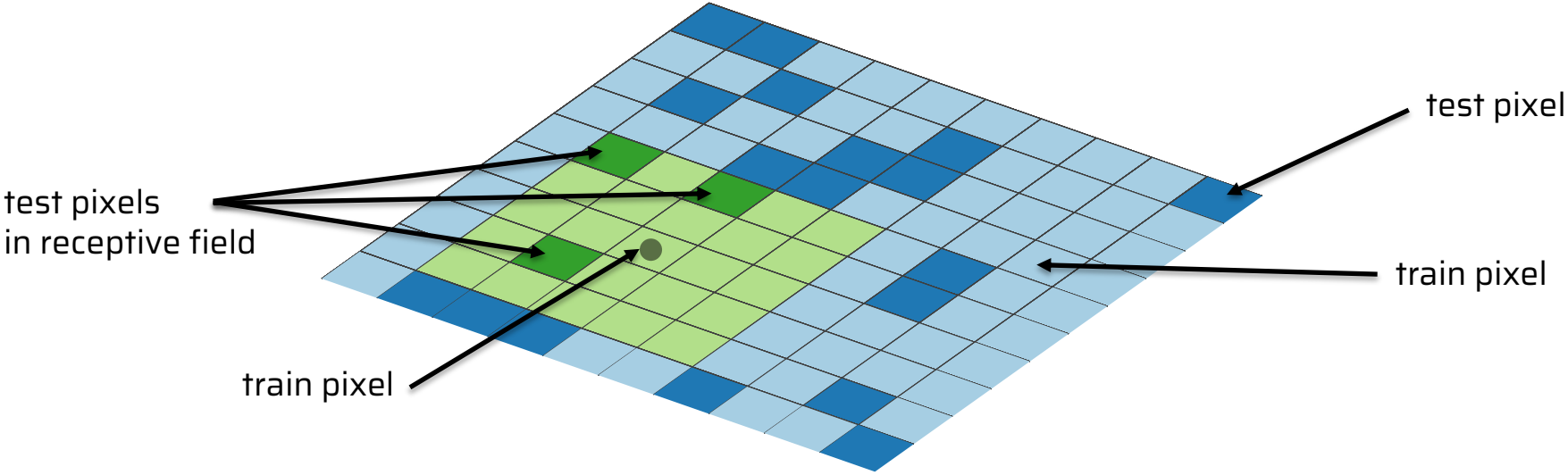
Evaluation of some networks... 3D CNN reaches almost perfect results, amazing?

Model	Accuracy reported in original paper	Accuracy reported with DeepHyperX
Nearest-Neighbour		75.63
1D CNN (Hu et al, 2015)	90.16	**89.34**
RNN (Mou et al, 2017)	85.7	**79.70**
3D CNN (Li et al, 2017)	99.07	**96.87**

still very high 🤔

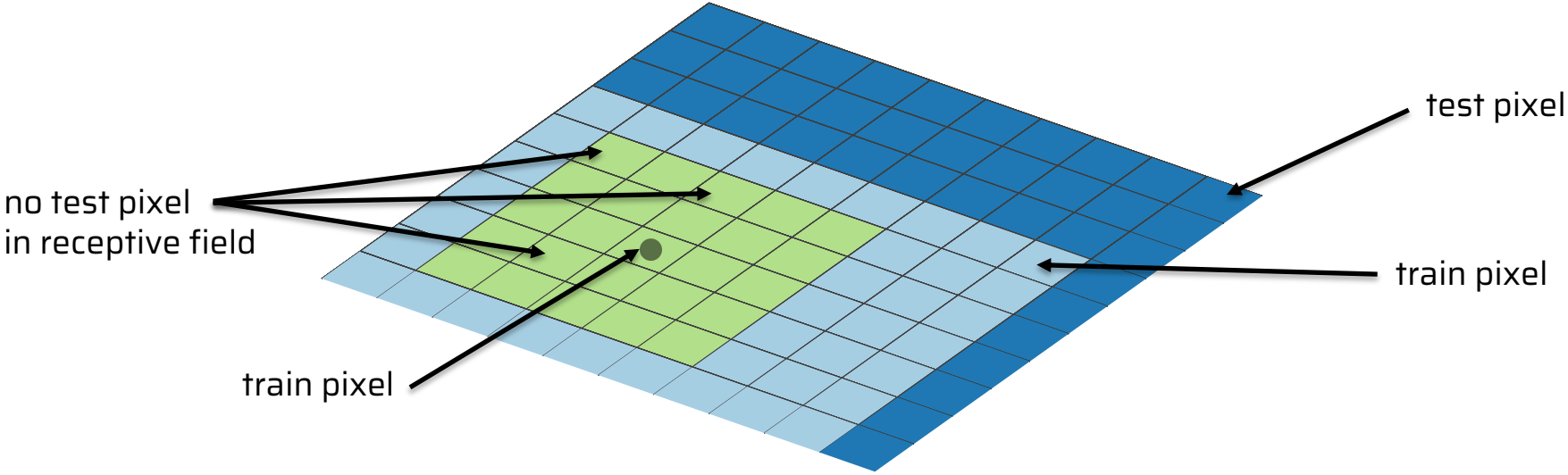
Public code is needed to reproduce the results…

DeepHyperX is a PyTorch toolbox for HSI classification
https://github.com/nshaud/DeepHyperX

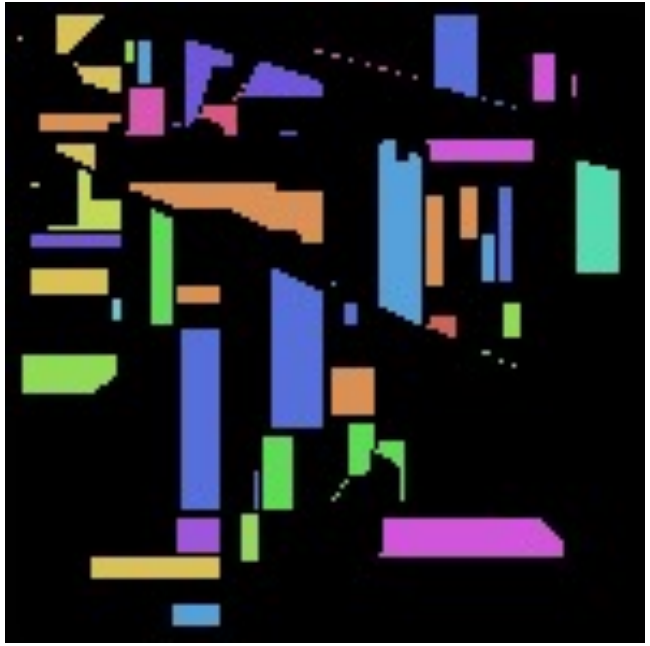


The devil is in the details

Possible overlap between train/test sets with a CNN due to its receptive field, making the network overfit and biasing its evaluation



A proper split is required to avoid overfitting and leading to fair assessment



A better split of Indian Pines into train / test sets
based on connected components

Model	Accuracy reported in original paper	Accuracy reported with DeepHyperX	Accuracy reported with DeepHyperX & disjoint train/test
Nearest-Neighbour		75.63	67.27
1D CNN (Hu et al, 2015)	90.16	89.34	82.99
RNN (Mou et al, 2017)	85.7	79.70	62.23
3D CNN (Li et al, 2017)	99.07	96.87	75.47

problem far from being solved! 😲

Be cautious with amazing results

Rely on public code, public data, public settings

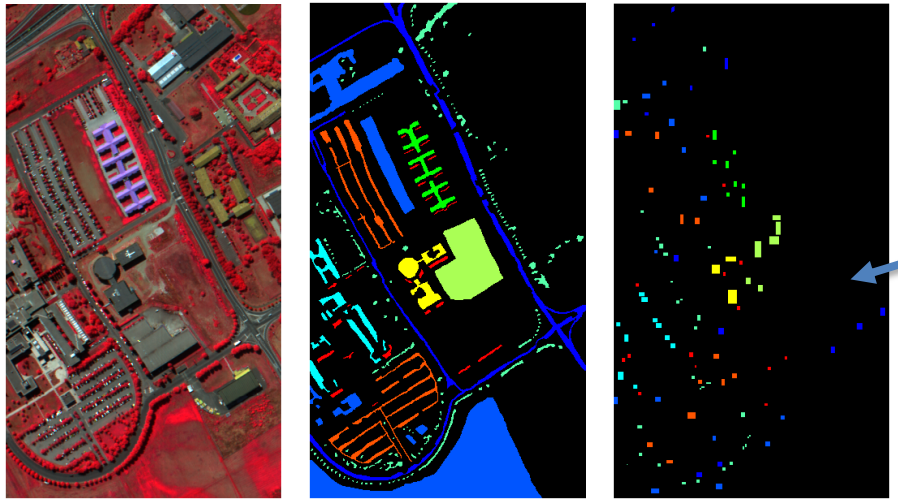
LESSONS LEARNT SO FAR

- Most papers divide the datasets in train/test splits by random sampling over the whole image, leading to a strong overlap 😠
- Some authors consider only a subset of the classes (e.g. ignoring the classes with few samples), preventing a fair comparison 😠
- Even with train/test splits done the same way, the number of samples in the train set might vary (e.g. 20%, 200 samples/class, etc), hardening the comparison 😠
- Some authors further divide the training set into training & validation, while others perform hyperparameters tuning directly on the test set 😠

MUST: use public train / test split, tune hyperparameters with a train subset

Some public initiatives & repositories:

- IEEE GRSS DFC and IEEE GRSS DASE, http://dase.grss-ieee.org (DASE 2.0 in progress)
- IEEE DataPort, http://ieee-dataport.org



The training set as provided by IEEE GRSS DASE: http://dase.grss-ieee.org

Thematic classes:

Trees	Asphalt	Bitumen
Gravel	Metal sheets	Shadows
Meadows	Bare soil	Self-blocking bricks

Pavia University, 610 x 340 pixels, 103 bands, 9 classes

Model	Accuracy reported in original paper	Accuracy reported with DeepHyperX (random)	Accuracy reported with DeepHyperX (DASE, disjoint)
Nearest-Neighbour		89.99	57.77
1D CNN (Hu et al, 2015)	92.56	90.59	**81.18**
2D+1D CNN (Ben Hamida, 2016)	94.6	92.39	**83.80**
3D CNN (Li et al, 2017)	99.39	96.71	**84.32**

-40%

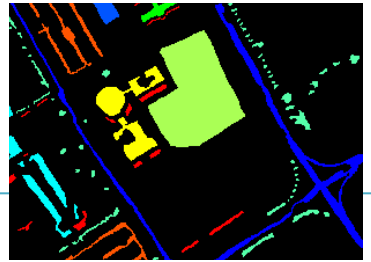
strong spatial correlation between neighbouring pixels

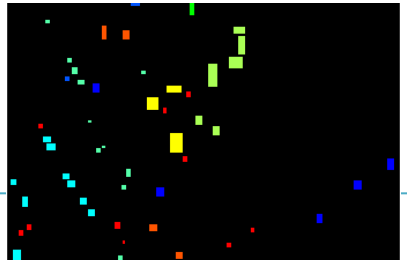
still (almost) perfect 🤩

still very high 🤔

Is the public train/test split reliable?

Not really.





TRAIN/TEST SPLIT USING RANDOM OR DISJOINT SAMPLING ON A SINGLE IMAGE

YES when…

- the spatial extent of the NN is (far) lower than the spatial gap between train/test samples
- a single image is to be classified, generalization is not sought

NO when…

- there is some overlap between receptive fields of train/test samples
- multiple images will be classified (several areas, several dates), with a single model trained on a subpart of the dataset

BUT THEN

be cautious when reporting accuracy!

A NEW HOPE...

- Recent public datasets take care of proper splits between training and test sets, usually on a tile basis (some tiles for training, the remaining tiles for testing)

IEEE GRSS DFC 2018
https://doi.org/10.1109/JSTARS.2019.2911113

Red tiles = train, remaining tiles = test

Model	Accuracy	Kappa
SVM	42.51	0.39
1D NN	41.08	0.37
1D CNN (Hu et al, 2015)	47.01	0.44
RNN (Mou et al, 2017)	41.53	0.38
2D+1D CNN (Ben Hamida, 2016)	46.28	0.43
3D CNN (Li et al, 2017)	49.26	0.46

$$K = \frac{c \times s - \sum_k^K p_k \times t_k}{s^2 - \sum_k^K p_k \times t_k}$$

c = # correct predictions
s = # samples
p_k = # predicted k
t_k = # true k

problem far from being solved...

DFC winners (FCN) report OA = 41.09, Kappa = 0.37 with HSI only, but OA = 80.78, Kappa = 0.8 when fusing with LIDAR and post-processing (same scores with LIDAR-post only)

A FEW REMAINING TIPS & TRICKS

- If dataset = image with spatial correlation,
  2D approach will outperform pixelwise classifiers in most cases.
- Bigger models = more parameters to optimize = more training samples
- Large convolution kernels = slower, especially in 3D
- FCN efficient since they predict several pixels at a time
  + no fully connected layer = less parameters = easier to train
- Non-saturating activation functions (e.g. ReLU) alleviate vanishing gradients
  = help build deeper networks while being faster to compute than sigmoid or tanh
- Most important hyperparameter = learning rate α
  - if too high, loss will diverge or oscillate without reaching the local minimum
  - if too low, very slow to converge
  - so prefer the highest α that makes not the loss diverge at first then slowly decrease it during training, or use an optimizer with adaptive learning rate (e.g. ADAM)
- **And many other things**, among which: initialization, batch size/normalization, dropout, data shuffle, data augmentation, class weighting, etc

CONTENTS OF THE LECTURES

- Deep Learning for Remote Sensing… see previous lectures (they were great!)
  - Principles of Deep Learning
  - Applications to EO

- Lecture #1: Good practices
  - Training
  - Evaluation

J. Castillo-Navarro, B. Le Saux, A. Boulch, N. Audebert, S. Lefèvre.
Semi-Supervised Semantic Segmentation in Earth Observation: The MiniFrance suite, dataset analysis and multi-task network study.
Machine Learning, 2021

- Lecture #2: Solutio[illegible] Data
  - Focus on main EO tasks:
    semantic segmentation, object detection, change detection
  - Focus on some complex data:
    LiDAR, SAR, multi-view imagery, low-resolution data

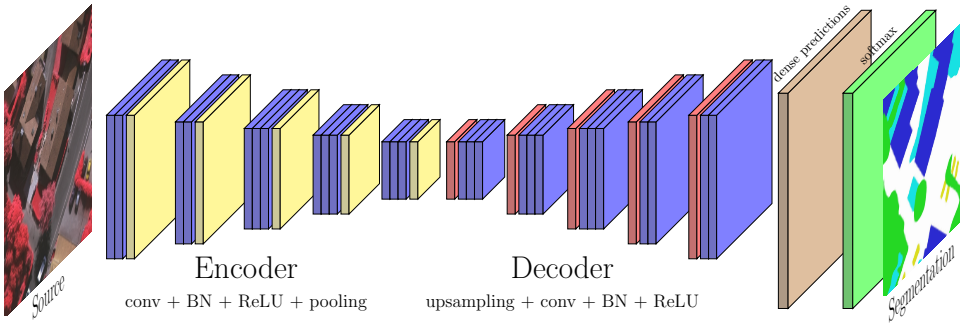
TRAINING A DEEP NETWORK IS STRAIGHTFORWARD

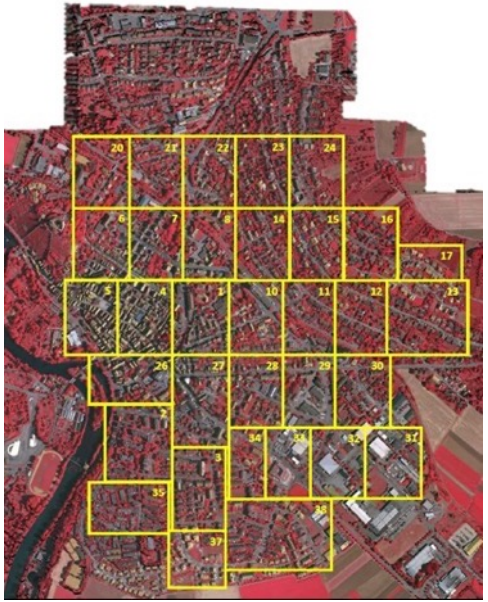
All you need is … numerous training samples ?

Let us consider a SegNet model successful on the ISPRS Vaihingen dataset.

N. Audebert, B. Le Saux, S. Lefèvre.
Beyond RGB: Very High Resolution Urban Remote Sensing with Multimodal Deep Networks.
ISPRS Journal of Photogrammetry and Remote Sensing, 140:20-32, 2018

OA ≃ 90%



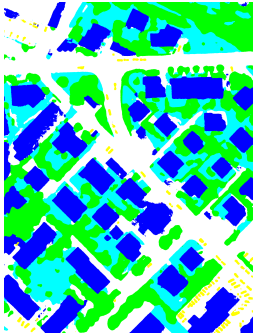


ISPRS Vaihingen, 33 IR-R-G tiles of ≃ 2000x1500 pixels (16 with GT), 6 classes

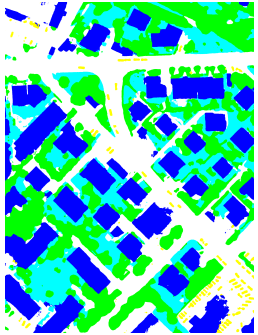
Benchmark is closed, so we split the 16 tiles into 12 for train, 4 for validation

(a) Tile 30

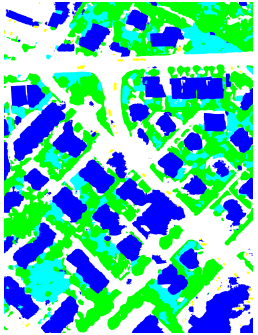
(b) Ground truth



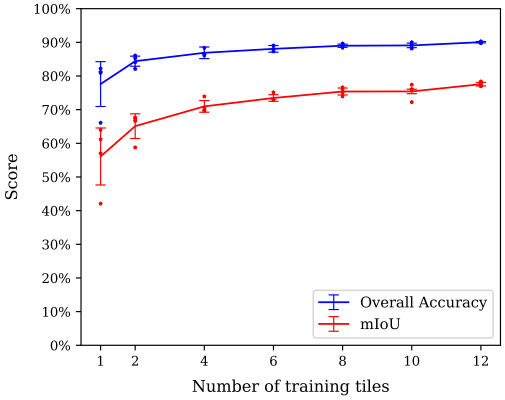
(c) 10 tiles training



(d) 6 tiles training

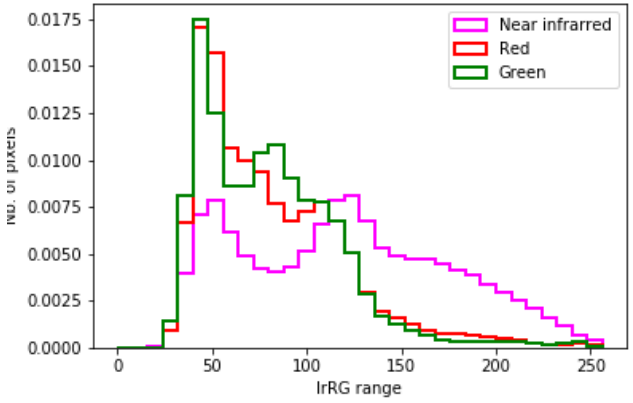


(e) 1 tile training

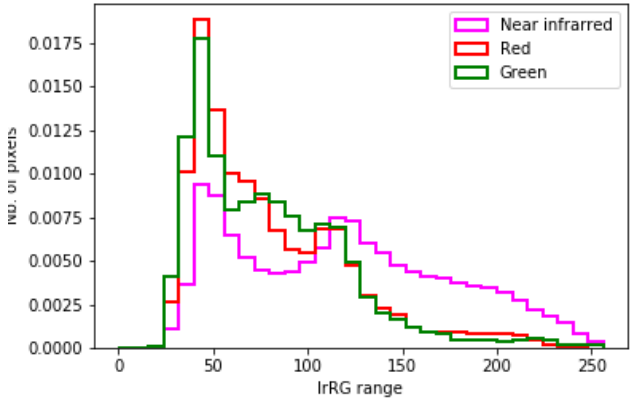


Prediction quality almost stable vs. number of training samples 😲

Why ?



(a) Vaihingen train set

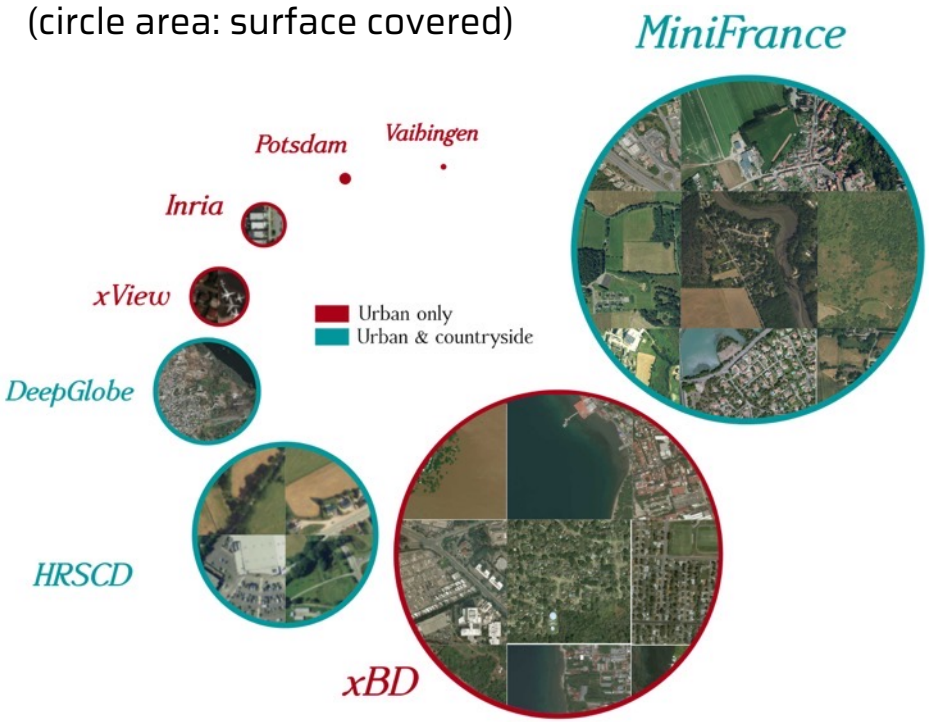


(b) Vaihingen val. set

Train and validation sets show very similar color distributions

Somehow the same problem as previously reported with random sampling on HSI

EO datasets at submeter resolution
(circle area: surface covered)



Very large dataset for semantic segmentation

Large-scale

- ≃ 53,000 km² (12x larger than DeepGlobe)
- ≃ 150 GB
- >2000 Aerial RGB images 10,000 x10,000 pixels, 50cm/pix, BD ORTHO (IGN)

Rich and varied

- 16 conurbations all over France, various climates & landscapes (Mediterranean, oceanic, montainous)

High semantic level of classes

- 14 land-use classes, Copernicus Urban Atlas

Underlying domain adaptation problem

- Train and test sets split by city

Designed for semi-supervised semantic segmentation

- Train split includes 2 labeled cities, 6 unlabeled ones

MiniFrance, a new, heterogeneous, challenging EO dataset

publicly available https://ieee-dataport.org/open-access/minifrance

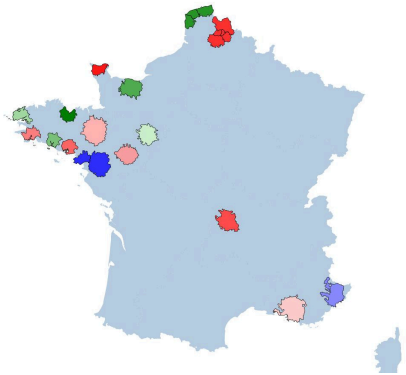


Fig. 1: Dataset overview.

Table 2: List of cities in MiniFrance and split details.

		Conurbation	*Tiles*	*% pixels*	*Color*
Training	*Labeled*	Nice	170	8.01 %	
		Nantes, Saint-Nazaire	226	10.65 %	
	Unlabeled	Le Mans	107	5.04 %	
		Brest	88	4.14 %	
		Lorient	68	3.20 %	
		Caen	126	5.94 %	
		Dunkerque, Calais, Boulogne-sur-Mer	150	7.07 %	
		Saint-Brieuc	71	3.34 %	
Test		Marseille, Martigues	162	7.63 %	
		Rennes	196	9.24 %	
		Angers	123	5.79 %	
		Quimper	79	3.72 %	
		Vannes	73	3.44 %	
		Clermont-Ferrand	150	7.07 %	
		Lille, Arras, Lens, Douai, Hénins	275	12.96 %	
		Cherbourg	57	2.68 %	

look similar 🤔

Table 3: Land Use classes available in MiniFrance.

Class	*% pixels*	*Color*
Urban fabric	9.6 %	
Industrial, commercial, public, military, private and transport units	6.4 %	
Mine, dump ans construction sites	0.7 %	
Artificial non-agricultural vegetated areas	1.1 %	
Arable land (annual crops)	29.5 %	
Permanent crops	1.0 %	
Pastures	29.0 %	
Complex and mixed cultivation patterns	0.0 %	
Orchards at the fringe of urban classes	0.0 %	
Forests	15.9 %	
Herbaceous vegetation associations	4.6 %	
Open spaces with little or no vegetation	0.4 %	
Wetlands	0.7 %	
Water	1.0 %	
Clouds, shadows or no data	0.1 %	

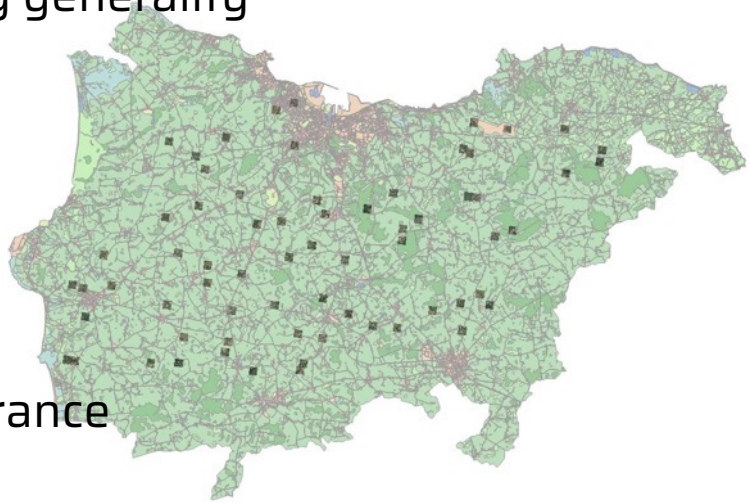
all urban fabric...
but look different
🤔

Fig. 2: Some samples of MiniFrance dataset on different localizations. Images (up) and their associated ground-truth (down). From left to right: Nice, Rennes and Vannes.

(IF MINI-FRANCE IS TOO LARGE FOR YOU)

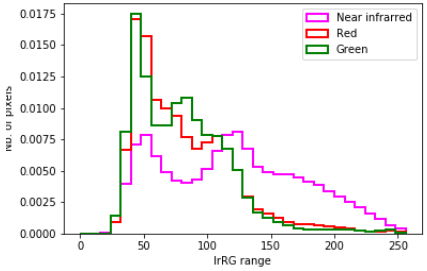
TinyMiniFrance

- 3,500 images of size 1,000 x 1,000 pixels (≃ 1,7% of MiniFrance)
- Uniform sampling over each region of miniFrance (including all tiles)
- Fast development and training, without losing generality

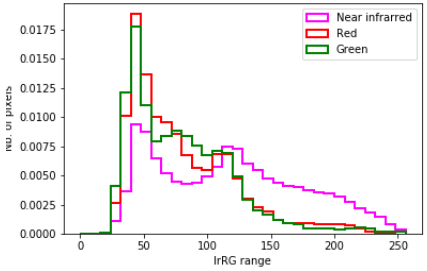


HOW TO

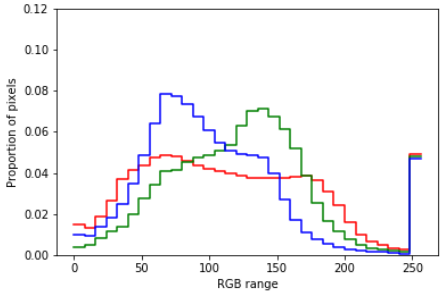
1. Design your model on TinyMiniFrance
2. Train, assess, optimize it on TinyMiniFrance
3. If needed, retrained the final model on MiniFrance



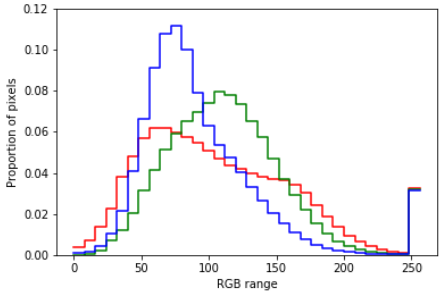
(a) Vaihingen train set



(b) Vaihingen val. set



(c) MiniFrance train set



(d) MiniFrance test set

Are colour histograms enough to assess variety?

IDEAL LEARNING CONDITIONS

Class representativeness

To properly learn a given class,
any learning algorithm needs to see at least some examples of this class during training.

Labelled training data must contain a good representation of all classes in the dataset, ideally with the same distribution than in the test data.

Appearance similarity

(In a standard supervised setting)
appearance features in the training set should have the same distribution as those on the test set to achieve good inference results.

Labelled training data must cover all range of appearances of different visual features in the dataset.

This constraint is hardly met in practice.

IDEAL LEARNING CONDITIONS

Class representativeness

To properly learn a given class,
any learning algorithm needs to see at least some examples of this class during training.

Labelled training data must contain a good representation of all classes in the dataset, ideally with the same distribution than in the test data.

Appearance similarity

(In a semi-supervised learning setting) unlabelled data provide more information on the possible visual features, help learning a wider appearance of each class, favouring generalization, bringing more robustness against distribution shift.

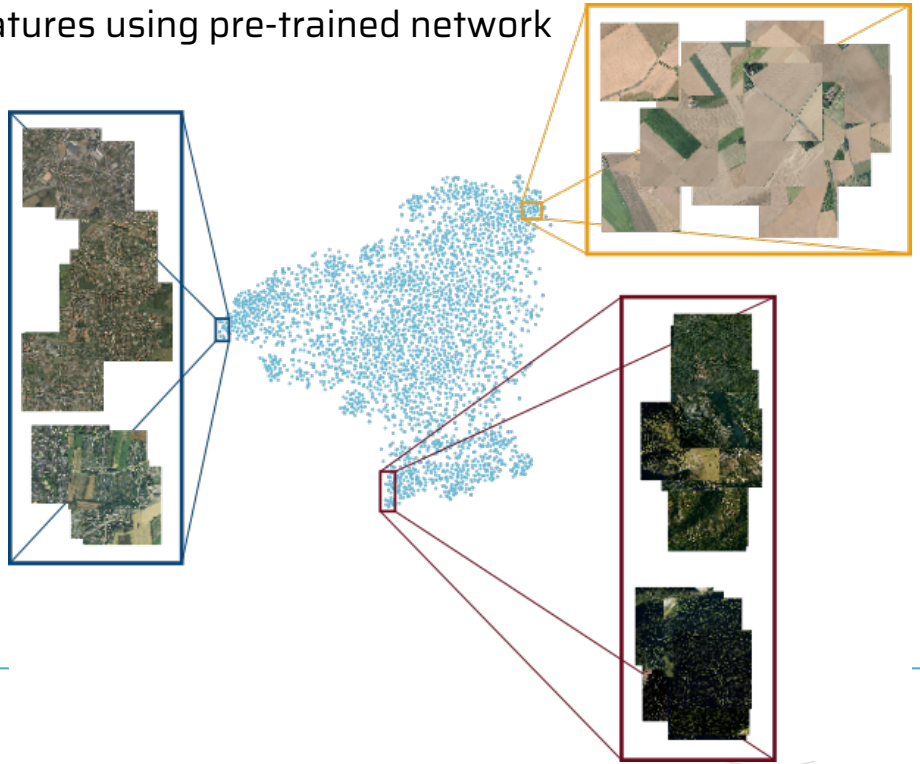
Training data (**labelled and unlabelled**) must cover all range of appearances of different visual features in the dataset.

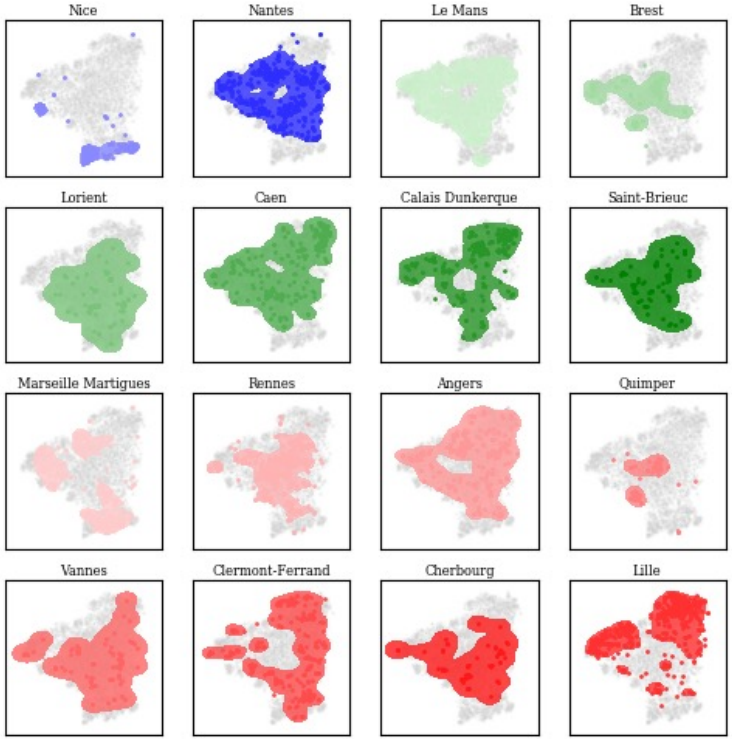
Unlabelled data are far less costly to collect.

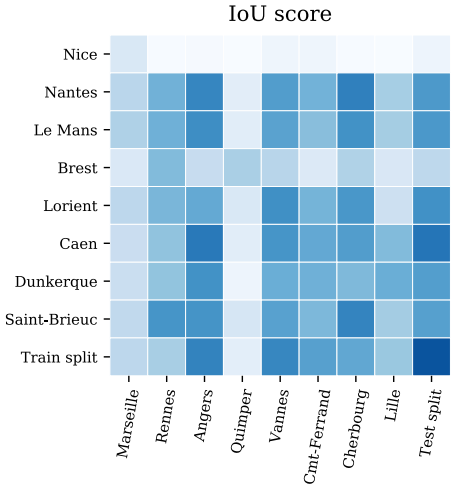
ASSESSING APPEAREANCE SIMILARITY

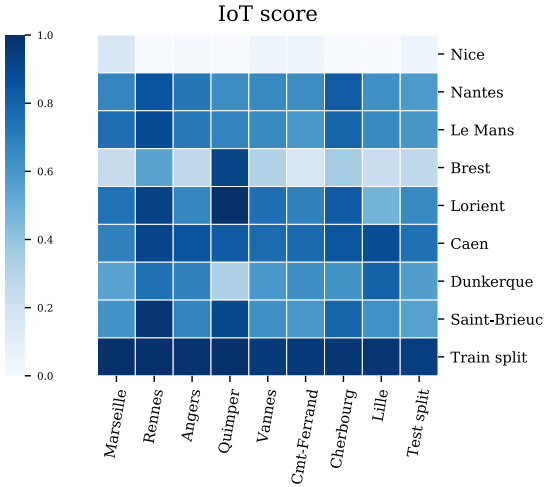
How to assess appeareance similarity between train and test sets?

1. For each image in the dataset, compute CNN features using pre-trained network (e.g. VGG16, ResNet34)
2. Apply t-SNE to the set of high-dimensional feature vectors to obtain a 2D representation
3. Each point in the 2D space can be traced back to the original tile and city it comes from. One-class SVM is used to estimate distribution of the city images in the 2D space.
4. Appearance similarity and coverage between cities is evaluated using
    - IoU between 2D surfaces
    - IoT (Intersection over Test area score)







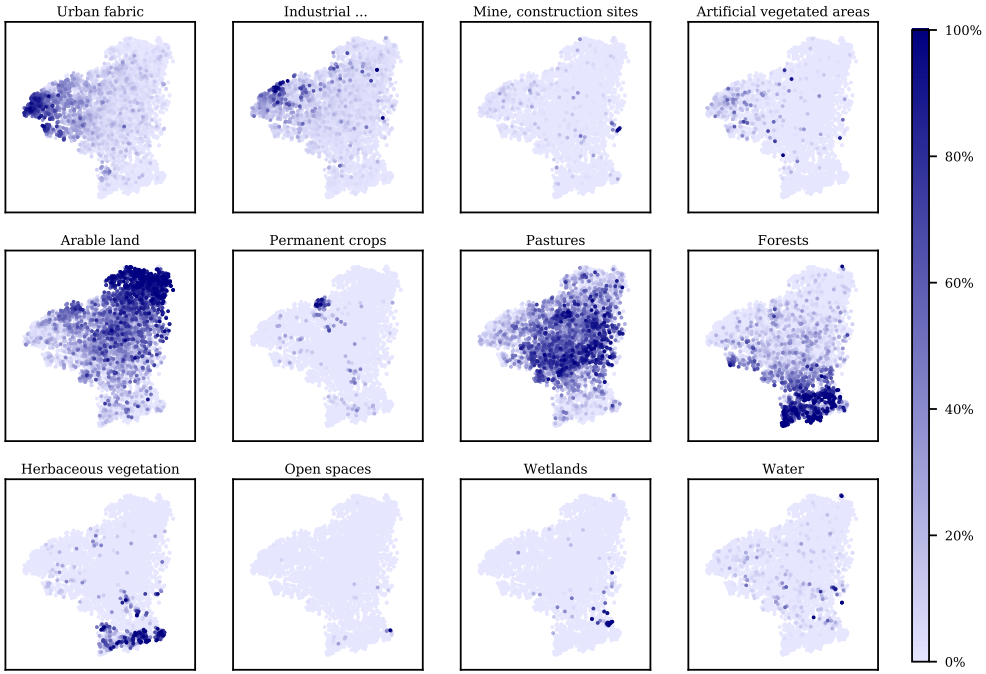


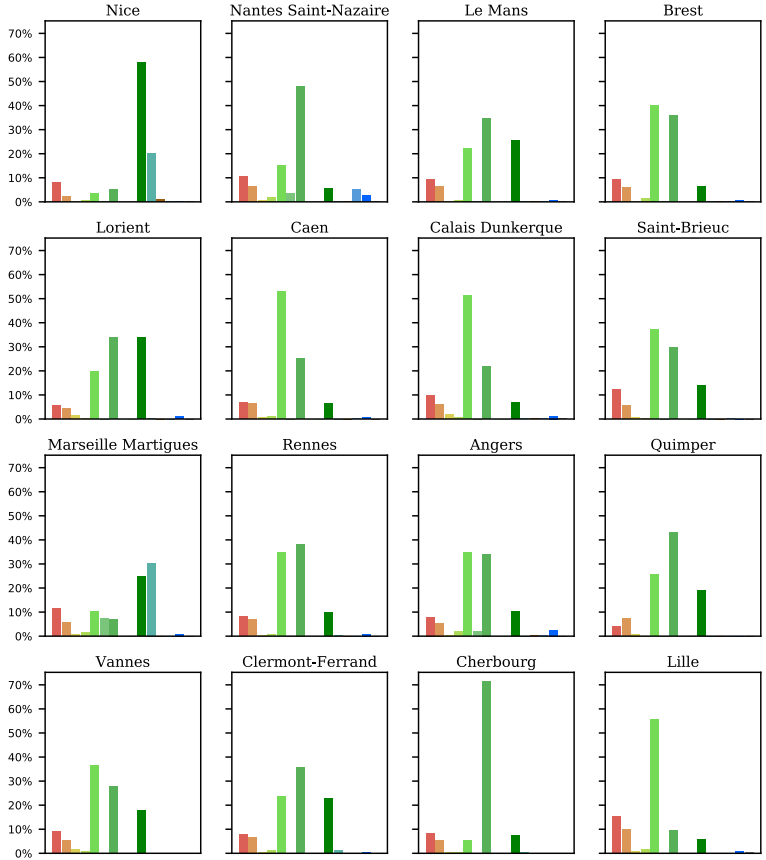
Without any training so far, we can see that

- IoU not very high: no identical appearance between cities in the train and test set
- IoT high: testing cities well covered by the ensemble of training cities

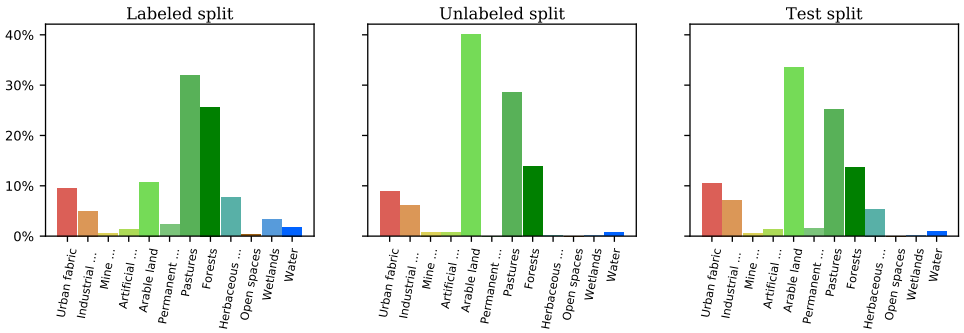
ASSESSING CLASS REPRESENTATIVENESS

1. Per-city class distribution using histograms
2. Visualization on the t-SNE

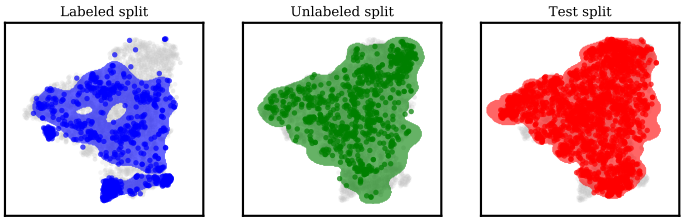




(TINY)MINIFRANCE

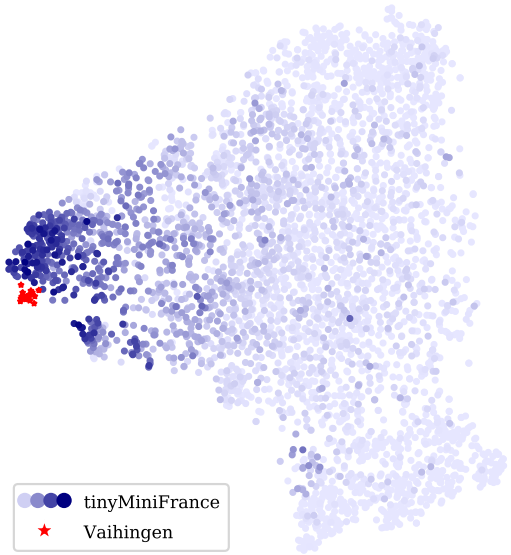


Class representativeness ✔



S_1 - S_2	$IoU(S_1, S_2)$	$IoT(S_1, S_2)$
Labeled - Test	0.63	0.64
Unlabeled - Test	0.87	0.93

Appearance similarity ✔ (thanks to unlabelled data)



Much larger variety of appearances in urban scenes, and in tiny(MiniFrance) dataset in general

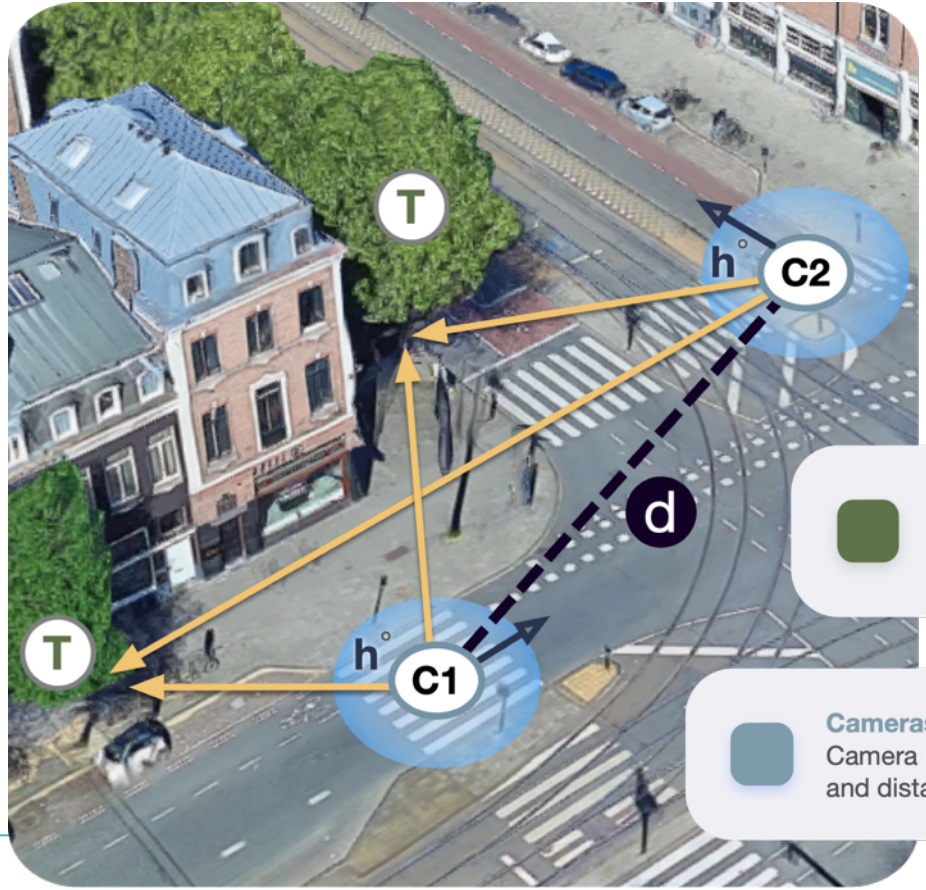
CONTENTS OF THE LECTURES

- Deep Learning for Remote Sensing… see previous le
  - Principles of Deep Learning
  - Applications to EO
- Lecture #1: Go
  - Training
  - Evaluati
- Lecture #2: Solu ex Data
  - Focus on main EO tasks:
    semantic segmentation, object detection, change detection
  - Focus on some complex data:
    LiDAR, SAR, multi-view imagery, low-resolution data

A.S. Nassar, S. Lefèvre, J.D. Wegner.
Multi-View Instance Matching with Learned Geometric Soft-Constraints.
ISPRS International Journal of Geo-Information, 9(11):687, 2020.

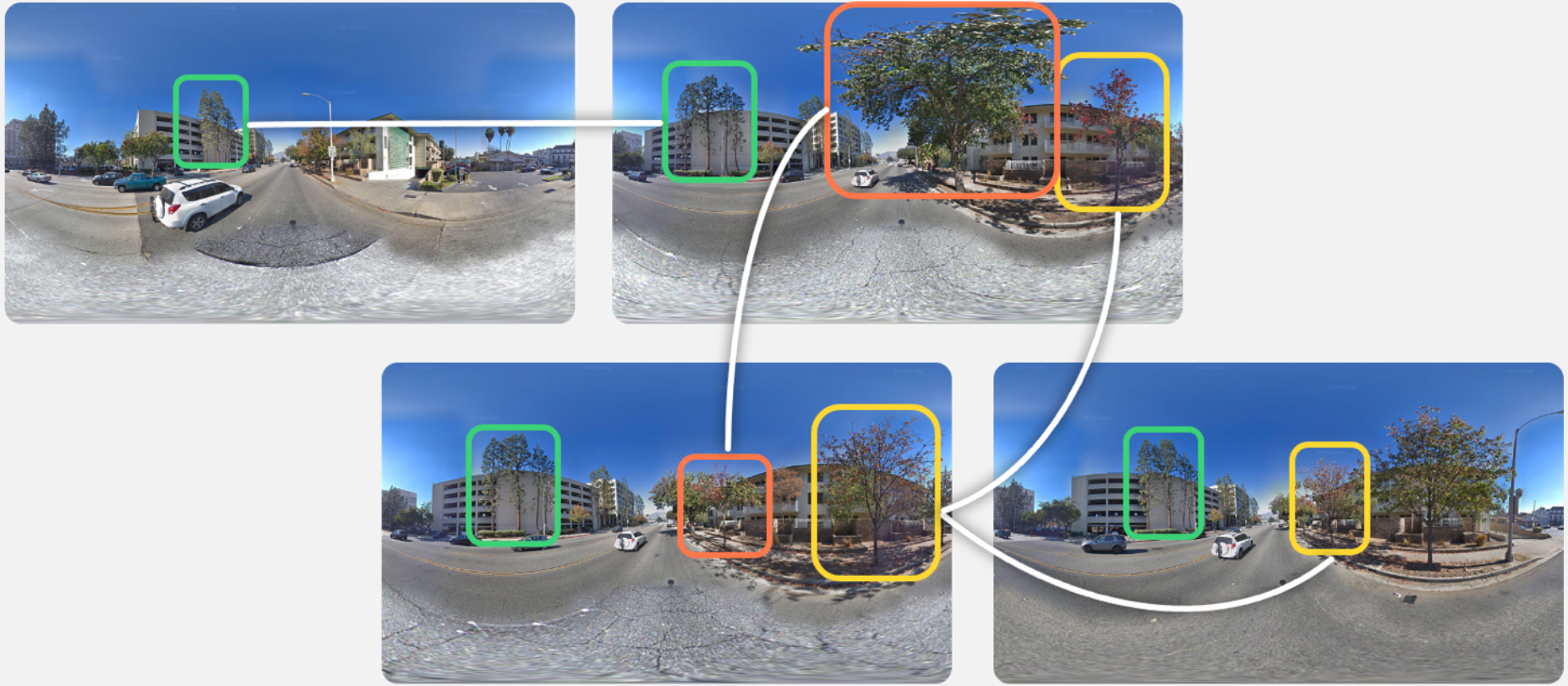
A.S. Nassar, S. D'aronco, S. Lefèvre, J.D. Wegner.
GeoGraph: Graph-Based Multi-view Object Detection with Geometric Cues End-to-End.
European Conference on Computer Vision, 488-504, 2020.

A.S. Nassar, S. Lefèvre, J.D. Wegner.
Simultaneous multi-view instance detection with learned geometric soft-constraints.
International Conference on Computer Vision, 6559-6568, 2019.

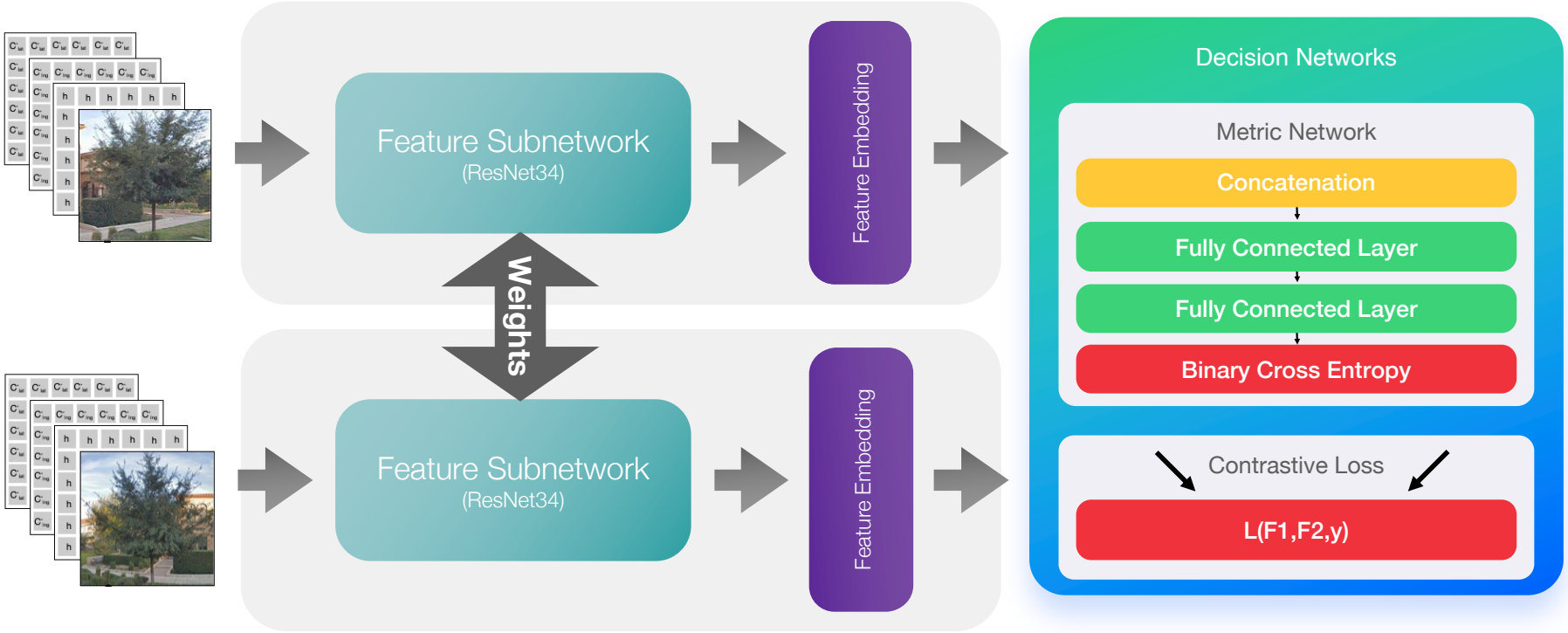


Object
Bounding Box, and Geo-coordinate.

Cameras
Camera Heading, geo-coordinate, and distance between cameras.



A FIRST SIAMESE NETWORK SOLUTION



Similar Objects



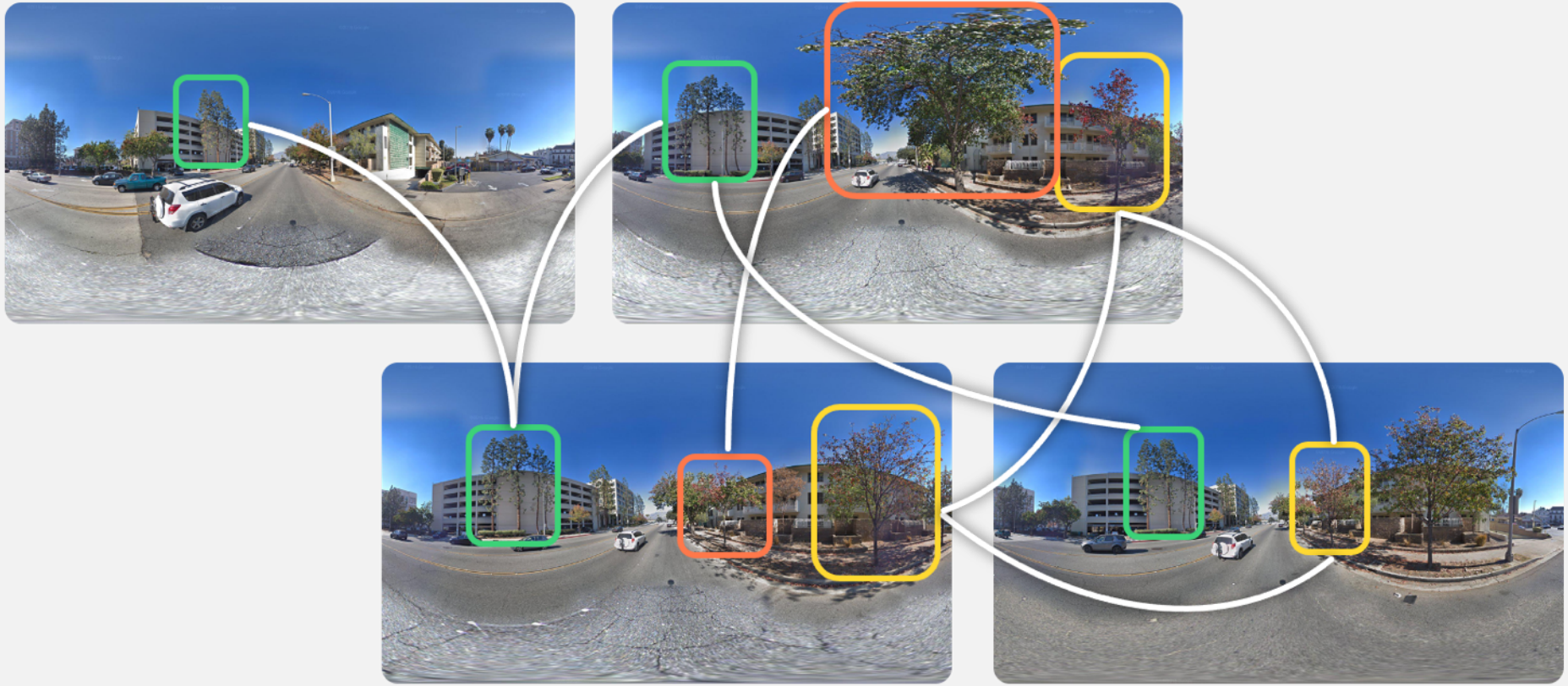


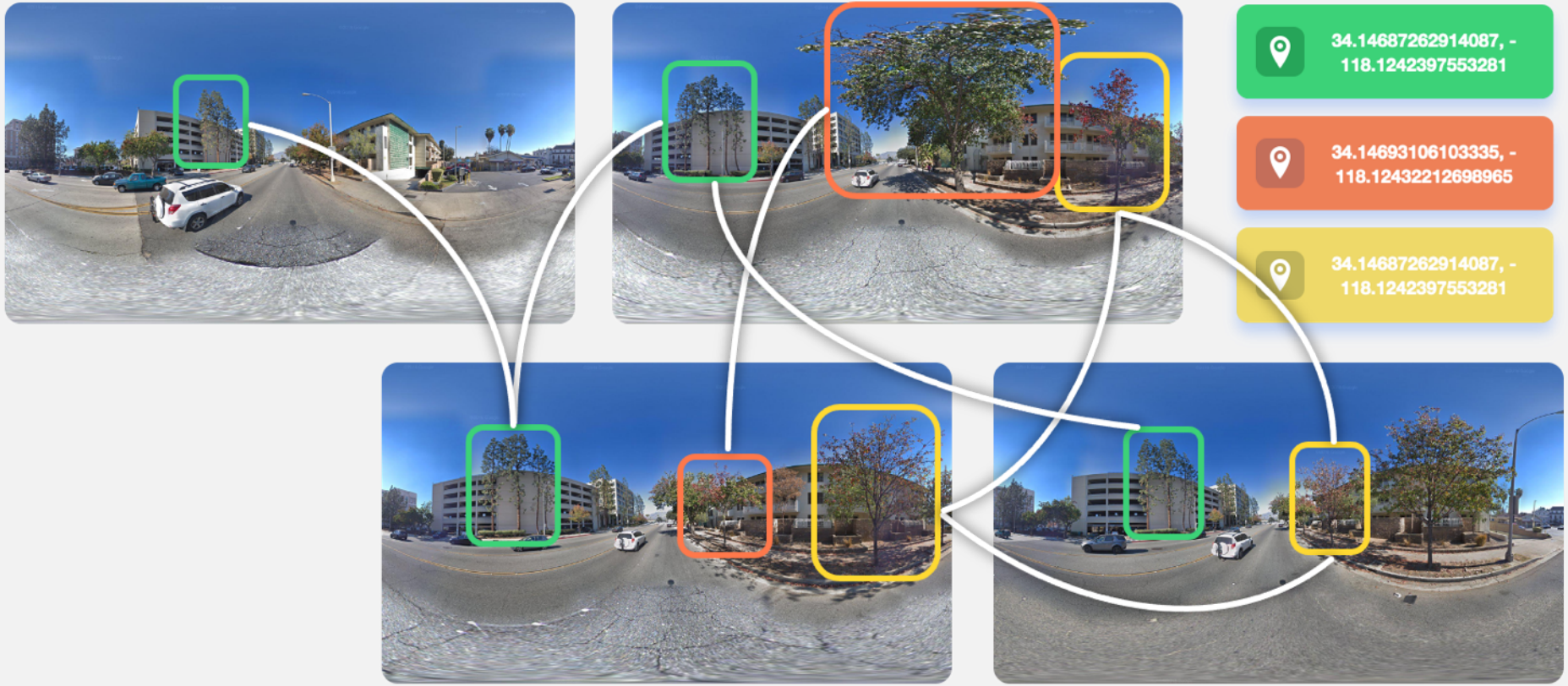

Different Objects



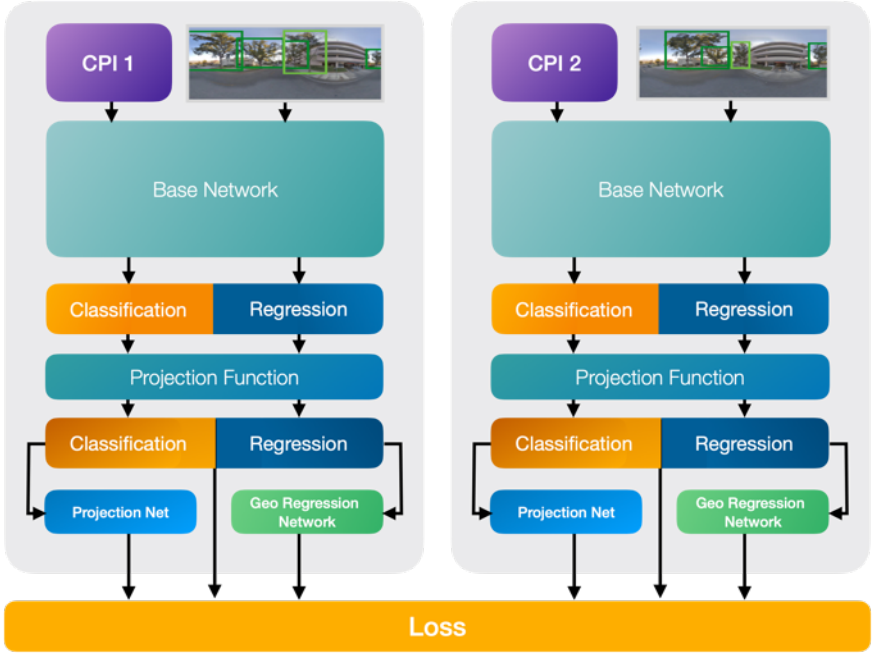


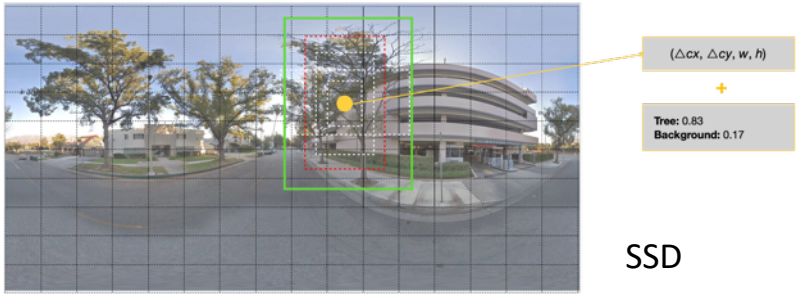

	Loss	Pasadena
Siamese Network Without Geometry	*Contrastive*	78.0 ± 0.611
	Metric	80.1 ± 0.5
Siamese Network With Geometry and concatenated features at input	*Contrastive*	79.6 ± 0.61
	Metric	81.1 ± 0.62
Ours	*Contrastive*	**81.75 ± 0.82**
	Metric	**82.3 ± 0.22**




34.14687262914087, -118.1242397553281
34.14693106103335, -118.12432212698965
34.14687262914087, -118.1242397553281

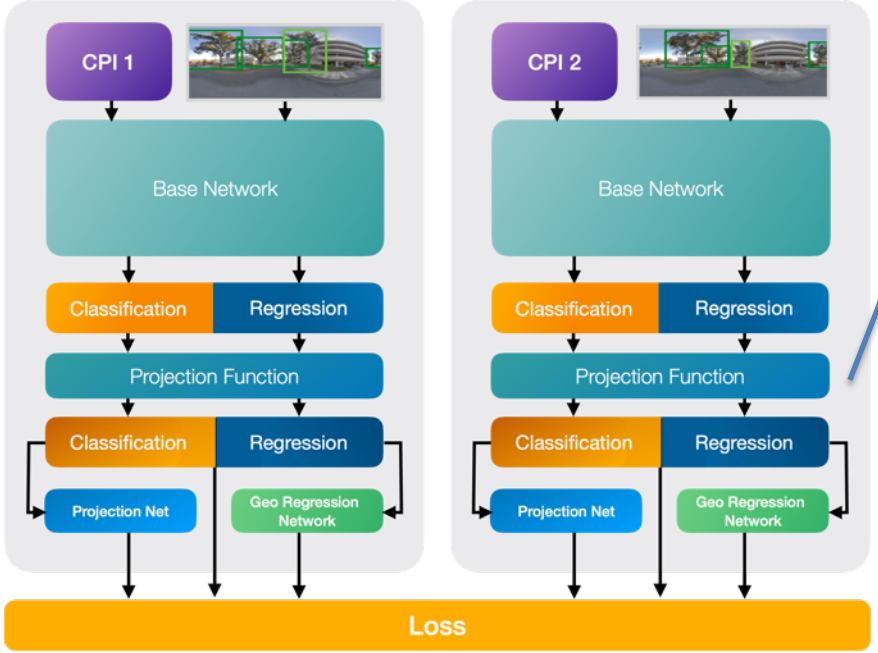
A SECOND SIAMESE NETWORK SOLUTION

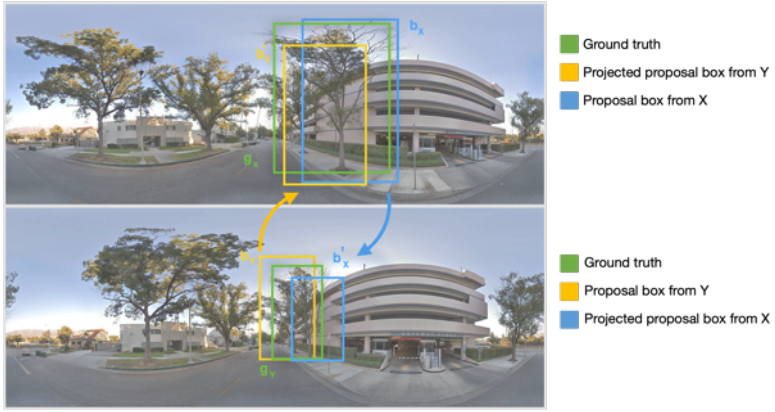




SSD

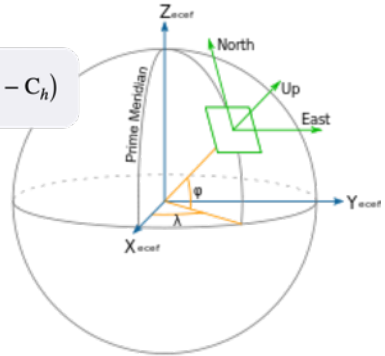
A SECOND SIAMESE NETWORK SOLUTION





- Object location in Earth, North, Up (ENU) coordinates:

$$(e_x, e_y, e_z) = \left(R\cos[C_{lat}]\sin[T_{lng} - C_{lat}], R\sin[T_{lat} - C_{lat}], -C_h\right)$$



- Object distance from camera

$$z = \sqrt{e_x^2 + e_y^2}$$

- Object pixel coordinates in image

$$x = \left(\pi + \arctan(e_x, e_y) - C_{yaw}\right) W/2\pi$$

$$y = \left(\pi/2 - \arctan(-h, z)\right) H/\pi$$

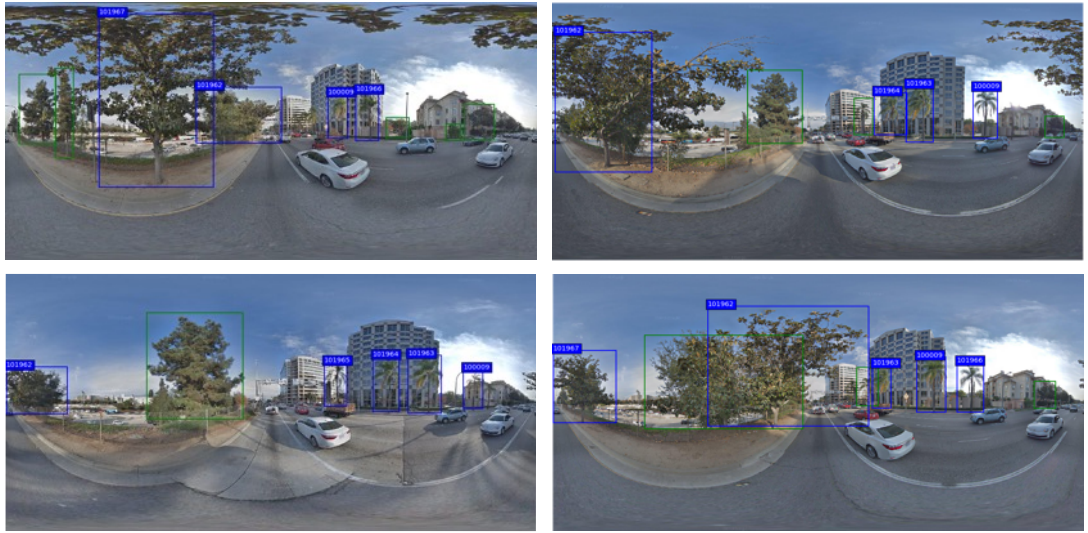
Illustrating reprojection

Pasadena multi-view ReID dataset

6,020 individual tree objects

6,141 panormas

Approx 4 trees
per panorama are annotated



Qualitative and quantitative assessment

Ground truth

Prediction

Instance labeled ground truth

Qualitative and quantitative assessment

Ground truth

Prediction

Qualitative and quantitative assessment

Method	Data set	Detection mAP	Re-ID mAP
Monocular	*Pasadena*	0,597	-
Ours	*Pasadena*	**0,682**	0,731

Method	Data set	Geo-localization Error (m)
Monocular	*Pasadena*	77,41
MRF	*Pasadena*	3,83
Ours	*Pasadena*	**3,13**

Qualitative and quantitative assessment

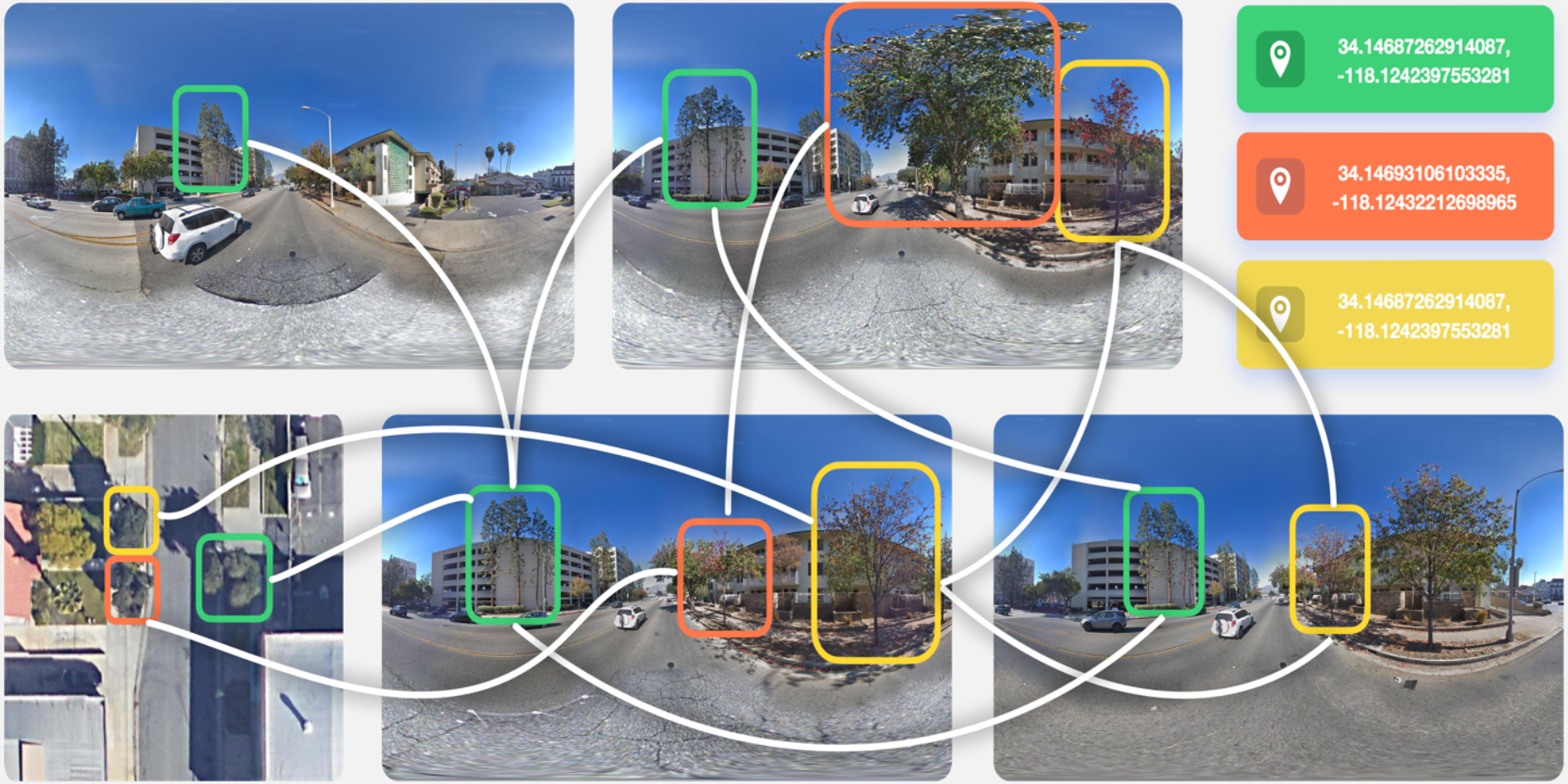
Method	Data set	Detection mAP	Re-ID mAP
Monocular	*Pasadena*	0,597	-
Ours	*Pasadena*	**0,682**	0,731

Method	Data set	Geo-localization Error (m)
Monocular		77,41
MRF	*Pasadena*	3,83
Ours	*Pasadena*	

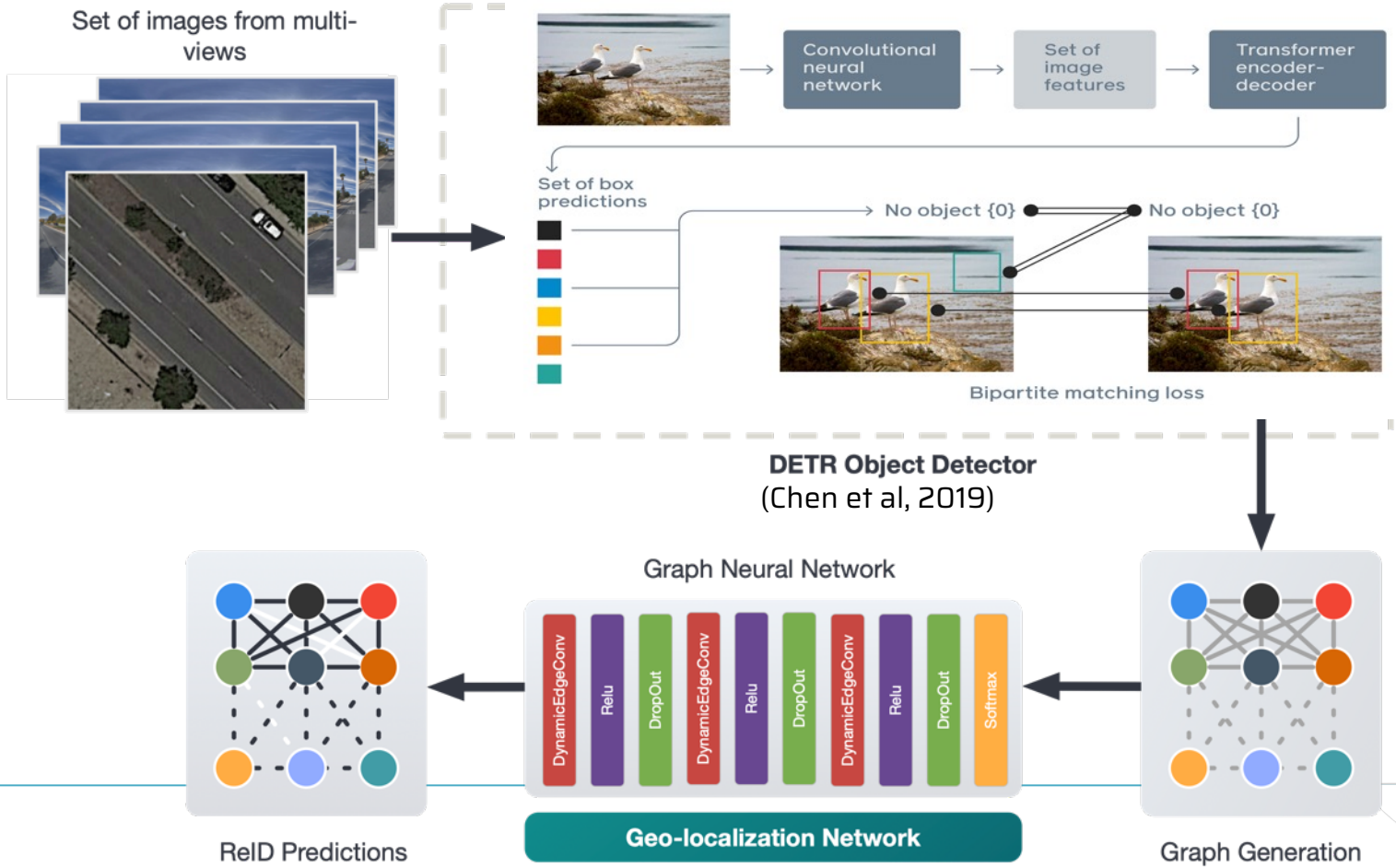
More than 2 views?

Aerial imagery?

Qualitative and quantitative assessment

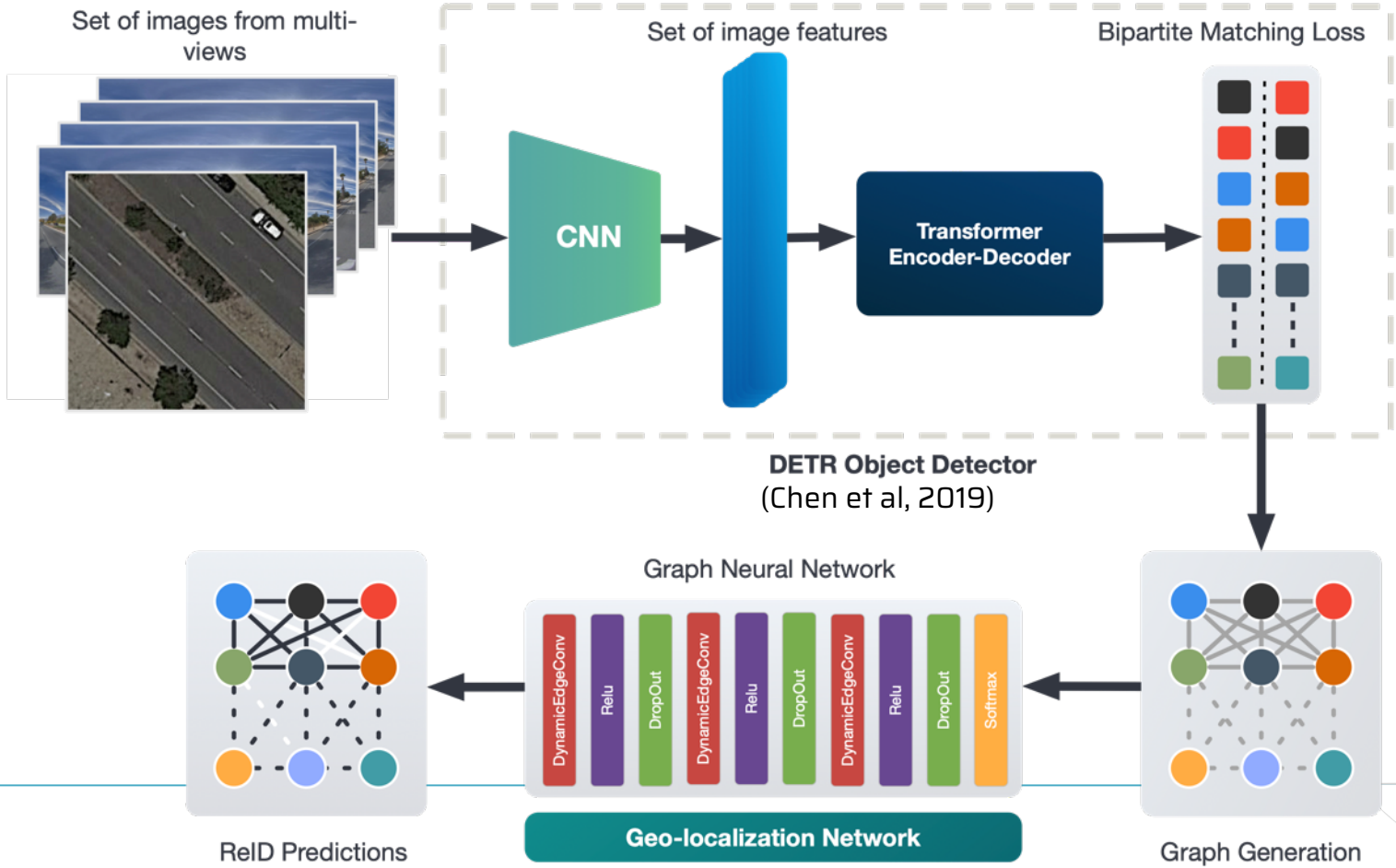

34.14687262914087,
-118.1242397553281
34.14693106103335,
-118.12432212698965
34.14687262914087,
-118.1242397553281

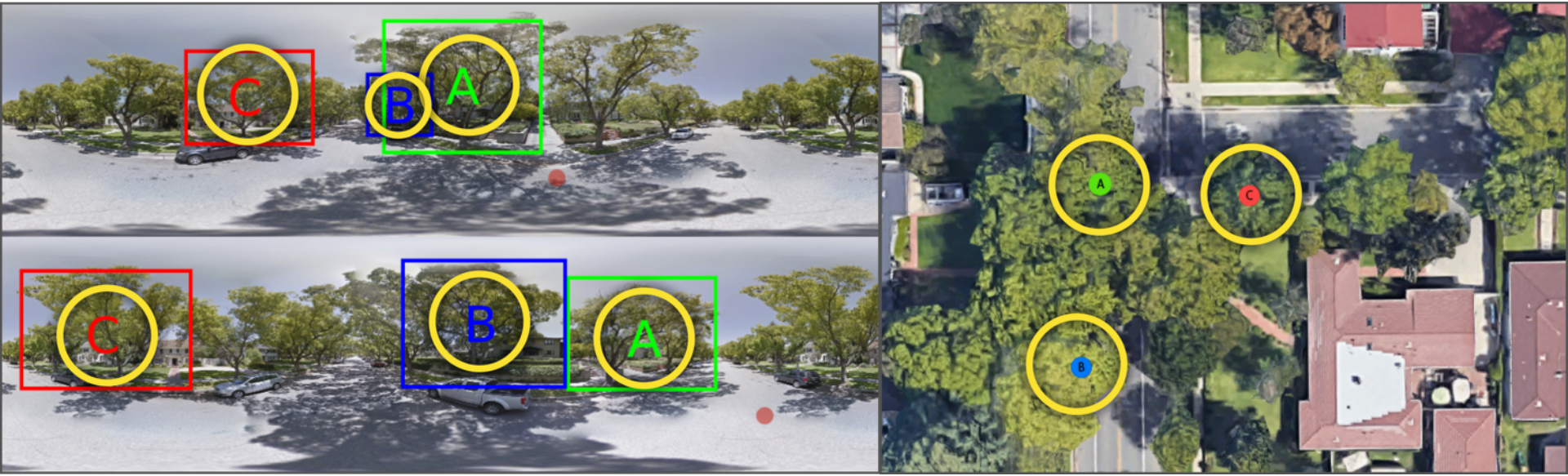
A GRAPH CNN SOLUTION



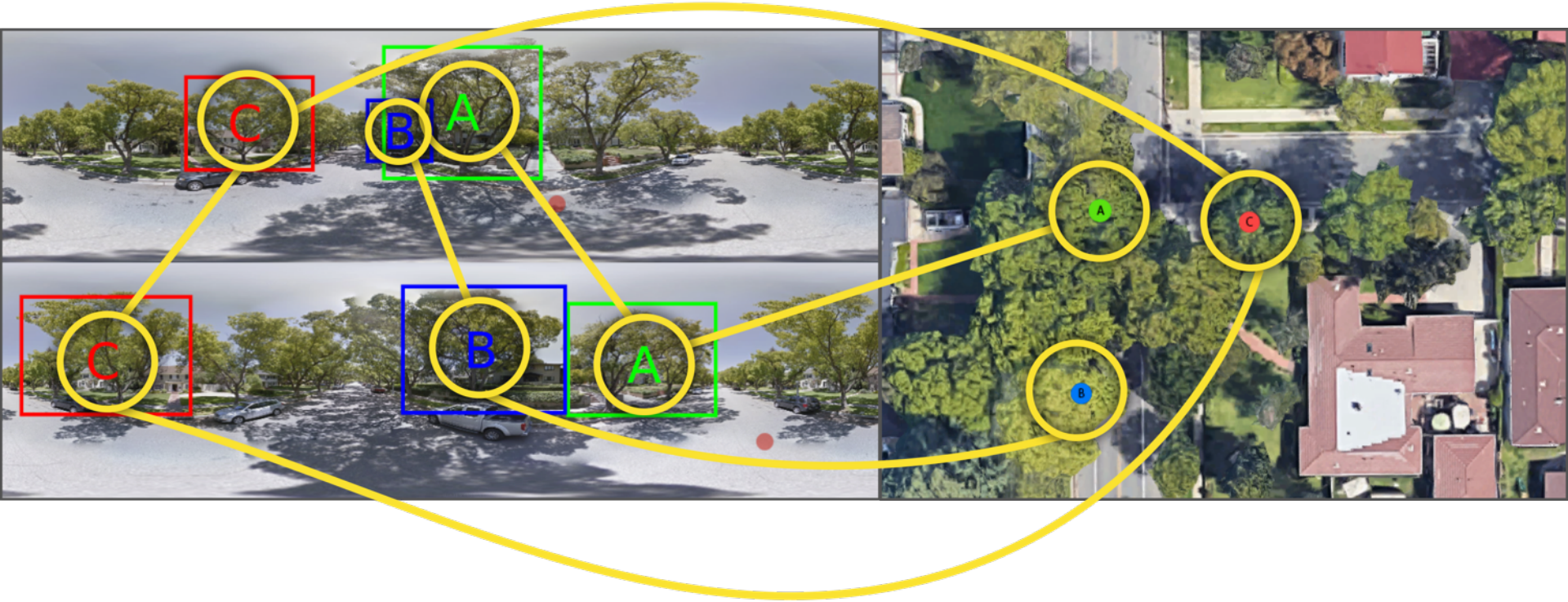
IRISA

A GRAPH CNN SOLUTION

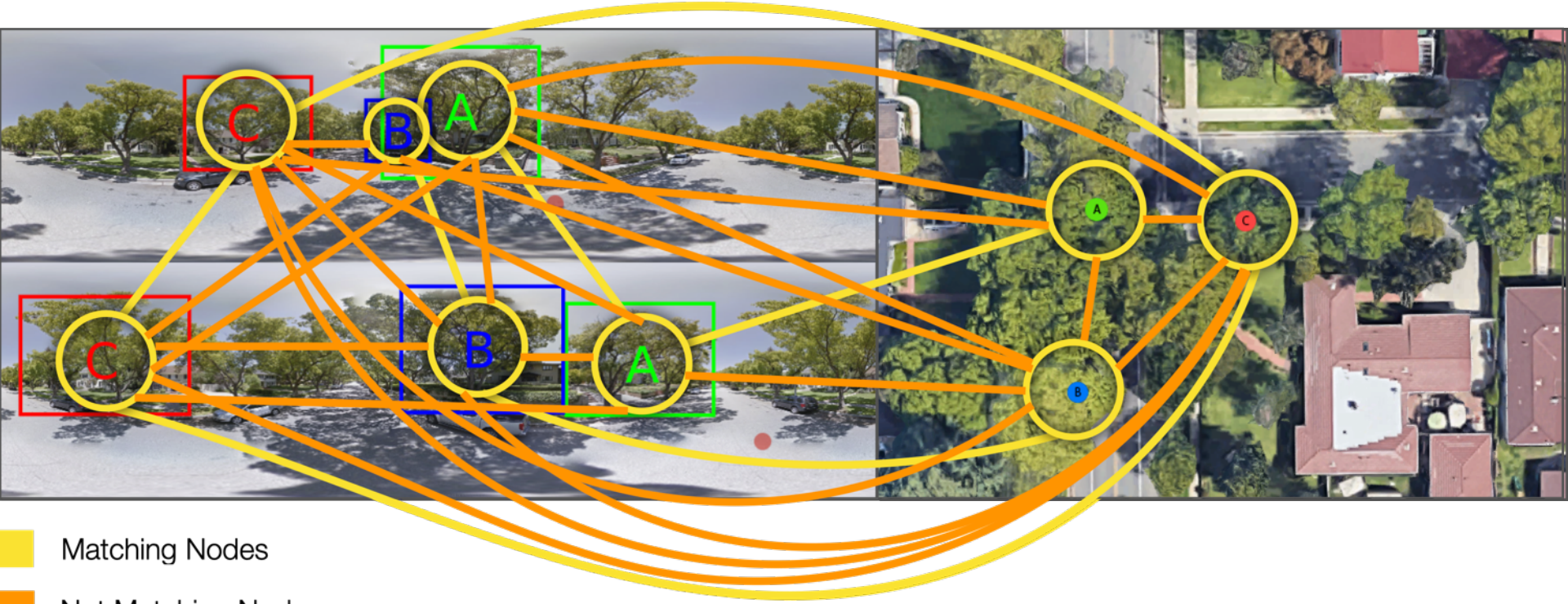




Graph of object occurrences as nodes



Graph of object occurrences as nodes

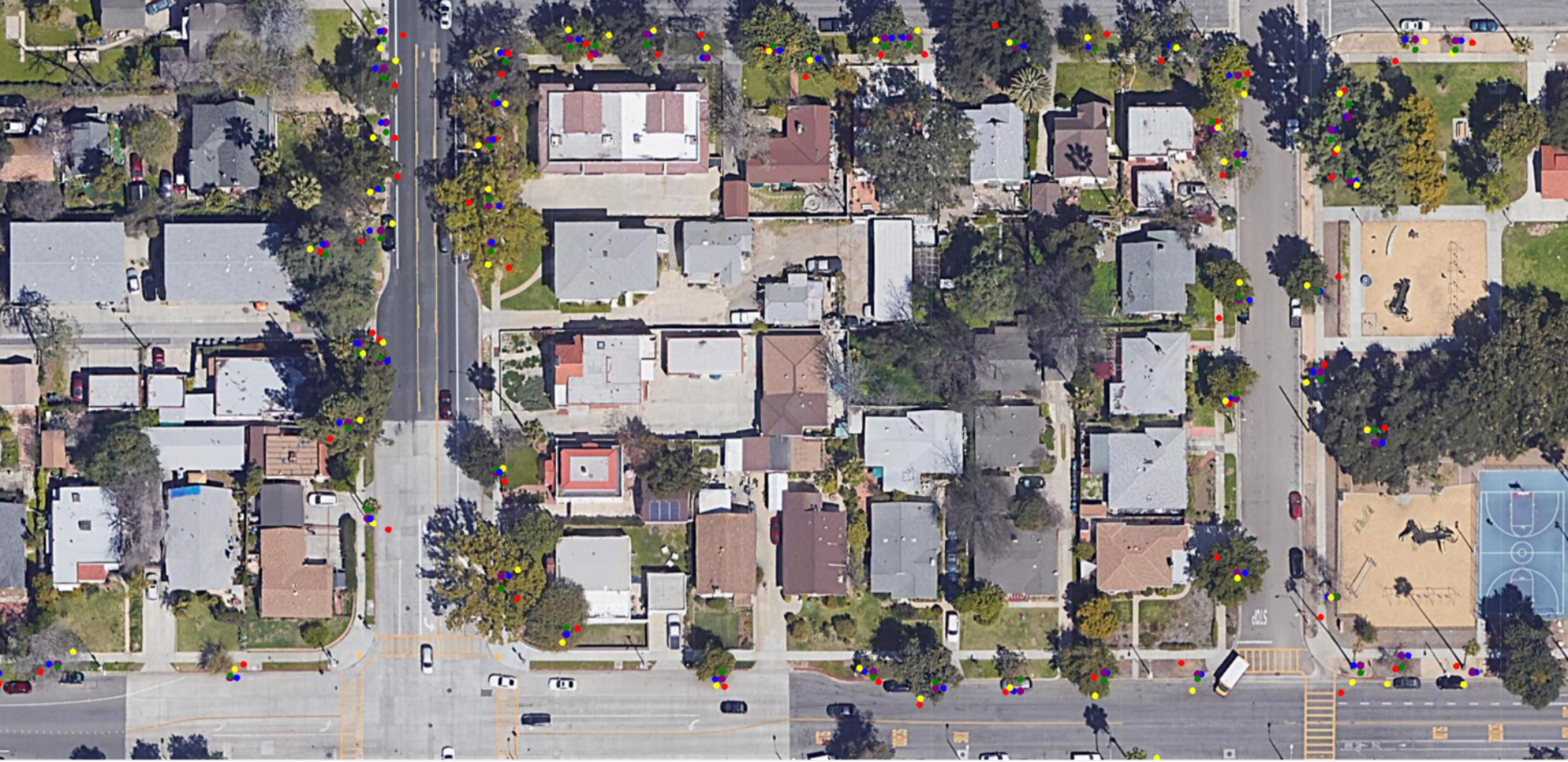


Matching Nodes

Not Matching Nodes

Graph of object occurrences as nodes

Pasadena Dataset (Aerial + Street View)



Green, purple, blue, yellow and red circles represent the ground-truth, GeoGraphV2, GeoGraph, SSD-ReID-Geo and MRF respectively.

UMR IRISA

Green, purple, blue, yellow and red circles represent the ground-truth, GeoGraphV2, GeoGraph, SSD-ReID-Geo and MRF respectively.

Method	# Views	Type	Data set	Detection mAP	Re-ID mAP	Geo-localization error (m)
MRF	*4*	*SV*	*Pasadena*	0,742	-	3,83
SSD-ReID-Geo	*2*	*SV*		0,682	0,731	3,13
GeoGraph	*2*	*SV*		**0.742**	0,754	2,94
GeoGraph	*4*	*SV*			0,763	2,75
GeoGraphV2	*4*	*SV*		0,721	**0,866**	2,32
GeoGraphV2	*5*	*SV*		0,71	0,815	**1,862**
GeoGraphV2		*AR*		0,41		

Mappilary Dataset

CONTENTS OF THE LECTURES

- Deep Learning for Remote Sensing… see previous lectures (they were great!)
  - Principles of Deep Learning
  - Applications to EO

- Lecture #1: Good practices
  - Training
  - Evaluation

- Lecture #2: Solutions for Complex Data
  - Focus on main EO tasks:
    semantic segmentation, object detection, change detection
  - Focus on some complex data:
    LiDAR, SAR, multi-view imagery, low-resolution data

CONTENTS OF THE LECTURES

- Deep Learning for Remote Sensing… see previous lectures (th
  - Principles of Deep Learning
  - Applications to EO

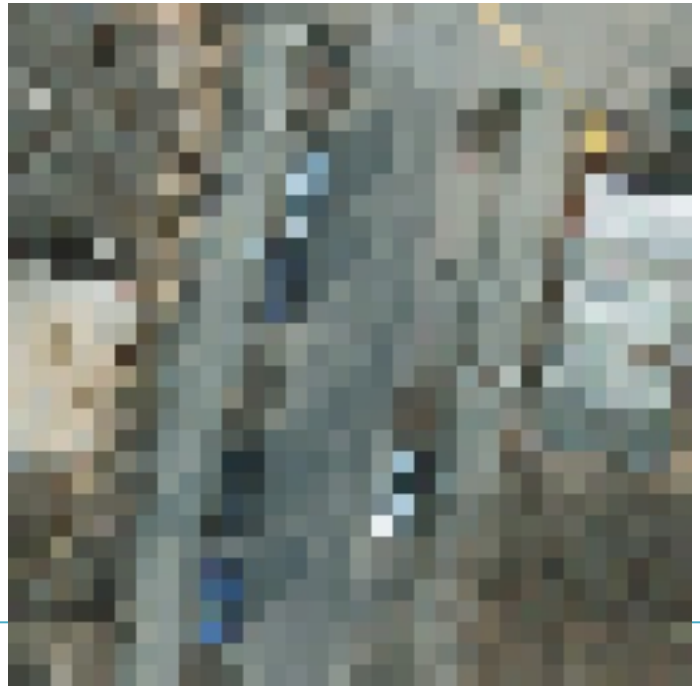
L. Courtrai, M.T. Pham, S. Lefèvre.
Small Object Detection in Remote Sensing Images Based on Super-Resolution with Auxiliary Generative Adversarial Networks.
Remote Sensing, 12(19):3152, 2020

M.T. Pham, L. Courtrai, C. Friguet, S. Lefèvre, A. Baussard.
YOLO-Fine: One-Stage Detector of Small Objects Under Various Backgrounds in Remote Sensing Images.
Remote Sensing, 12(15):2501, 2020

- # 2: Solutions for Complex Data
  - Focus on main EO tasks:
    semantic segmentation, object detection, change detection
  - Focus on some complex data:
    LiDAR, SAR, multi-view imagery, low-resolution data

SMALL OBJECT DETECTION, AN EO PROBLEM

When the spatial resolution is too low w.r.t. the size of objects of interest

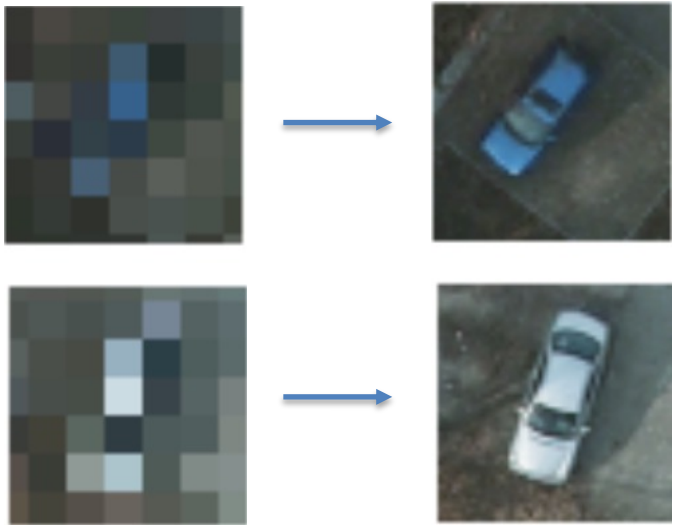


SMALL OBJECT DETECTION, AN EO PROBLEM

When the spatial resolution is too low w.r.t. the size of objects of interest

SMALL OBJECT DETECTION, AN EO PROBLEM

When the spatial resolution is too low w.r.t. the size of objects of interest

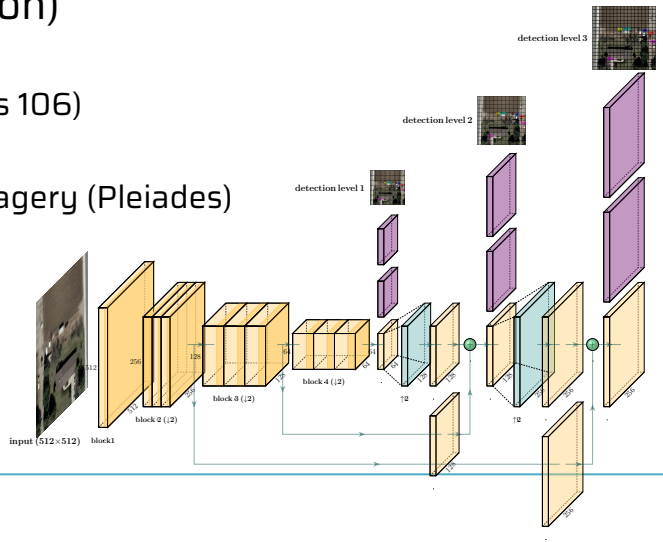


SMALL OBJECT DETECTION, AN EO PROBLEM

Approach #1:
adapt a standard object detector
to small objects

YOLO-fine (1-stage, fast detection)

- inspired from YOLOv3
- tailored for small objects (68 layers vs 106)
- lightweight network (18 MB vs 237)
- Able to cope with aerial & satellite imagery (Pleiades)



SMALL OBJECT DETECTION, AN EO PROBLEM

Approach #1:
adapt a standard object detector
to small objects

YOLO-fine (1-stage, fast detection)

- inspired from YOLOv3
- tailored for small objects (68 layers vs 106)
- lightweight network (18 MB vs 237)
- Able to cope with aerial & satellite imagery (Pleiades)

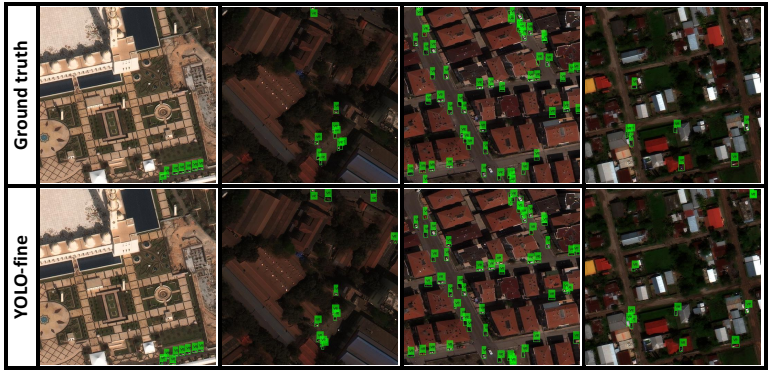
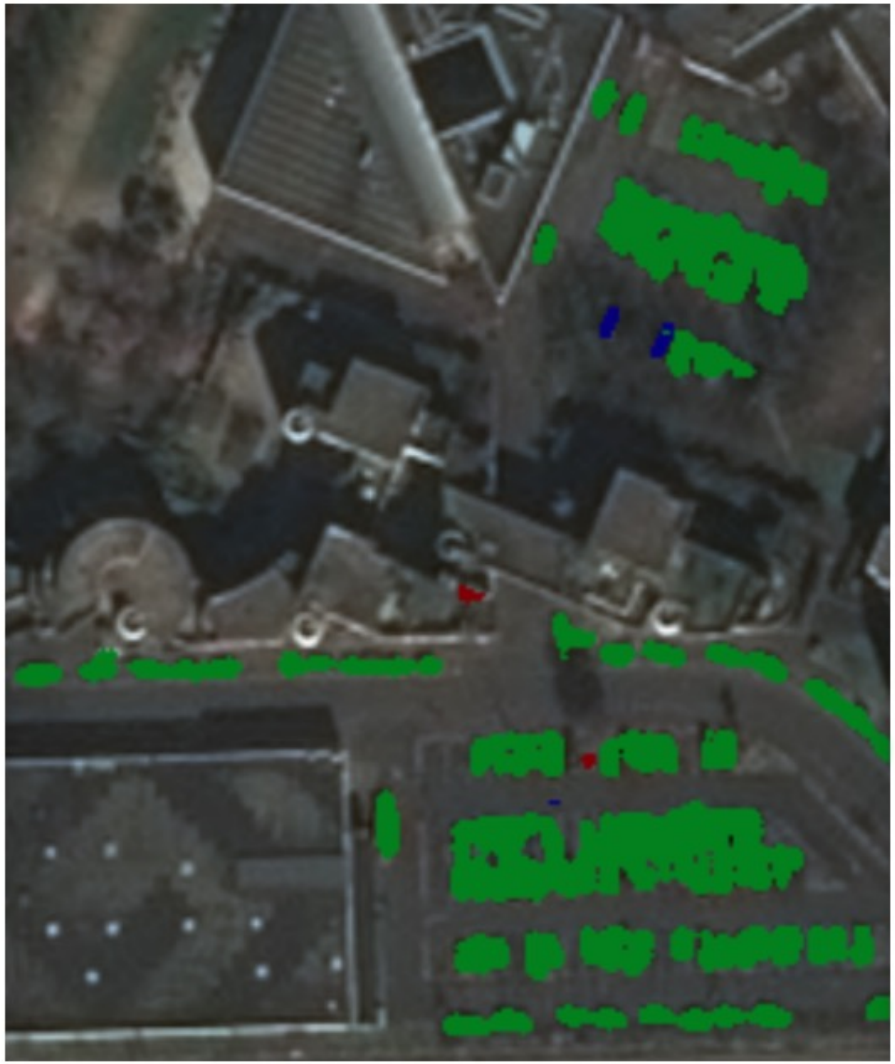


Figure 10. Illustration of detection results on XVIEW.

Model	Precision	Recall	F1-score	mAP
SSD	0.80	0.62	0.70	68.09
EfficientDet(D0)	0.84	0.78	**0.81**	82.45
EfficientDet(D1)	0.80	0.75	0.78	82.51
Faster R-CNN	0.50	0.72	0.59	57.01
RetinaNet(50)	0.70	0.21	0.33	34.84
YOLOv2	0.75	0.41	0.53	47.68
YOLOv3	0.77	0.74	0.75	78.93
YOLOv3-tiny	0.66	0.61	0.64	62.03
YOLOv3-spp	0.81	0.73	0.77	77.34
YOLO-fine	0.87	0.72	0.79	**84.34**

Detection results on XVIEW. Best results of F1-score and mAP in bold.



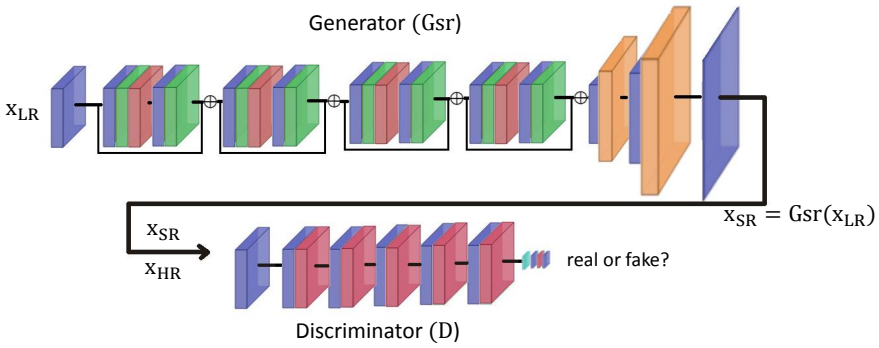
SMALL OBJECT DETECTION, AN EO PROBLEM

Approach #2:

use super-resolution to (artificially) increase image resolution

Object detection-driven super-resolution network

SMALL OBJECT DETECTION, AN EO PROBLEM



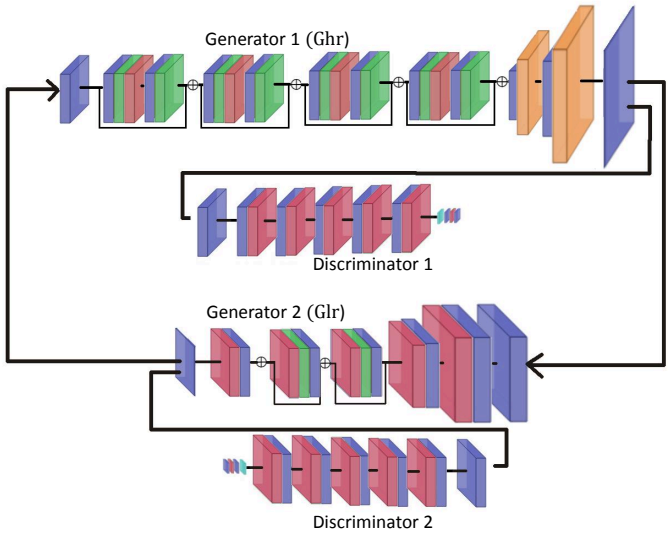
Approach # 2:

use super-resolution to (artificially) increase image resolution

Object detection-driven super-resolution network

- GAN for image super-resolution

SMALL OBJECT DETECTION, AN EO PROBLEM

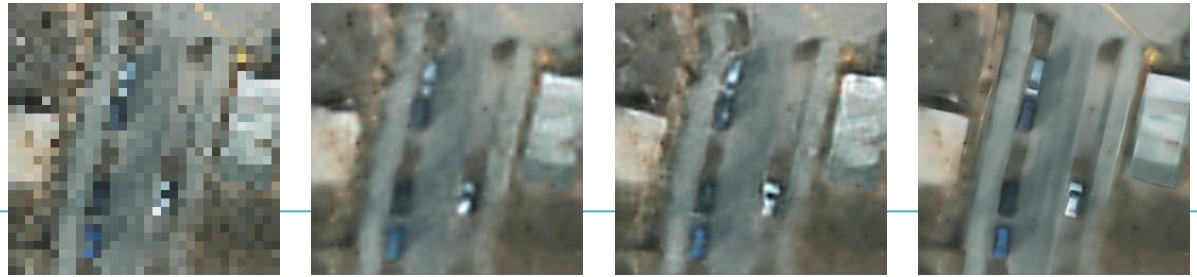


Approach # 2:

use super-resolution to (artificially) increase image resolution

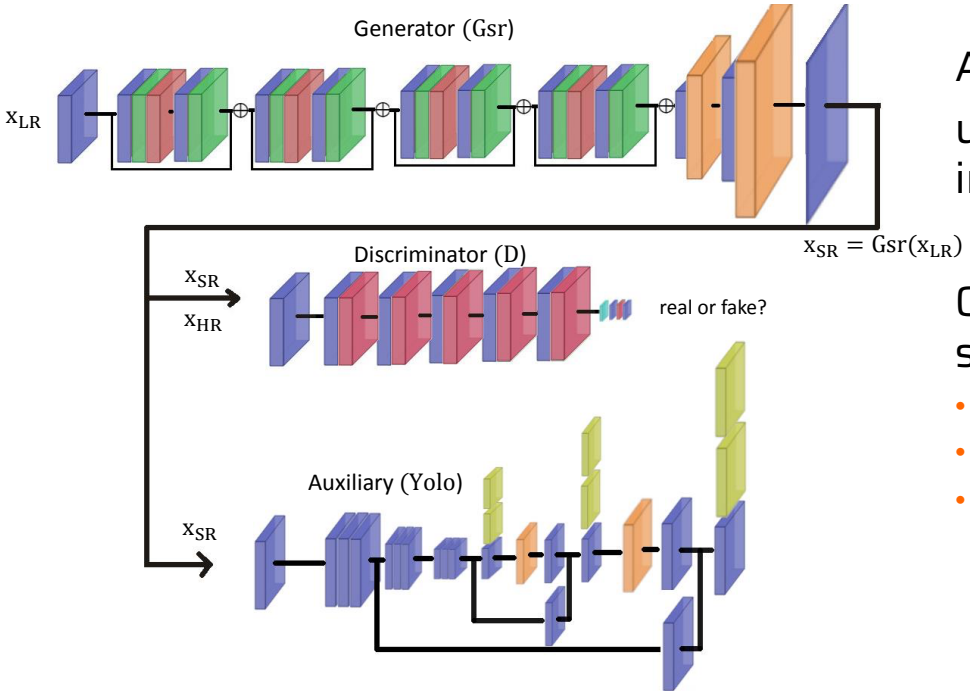
Object detection-driven super-resolution network

- GAN for image super-resolution
- Cycle-GAN performs better



LR EDSR SR-WGAN SR-CWGAN

SMALL OBJECT DETECTION, AN EO PROBLEM



Approach # 2:

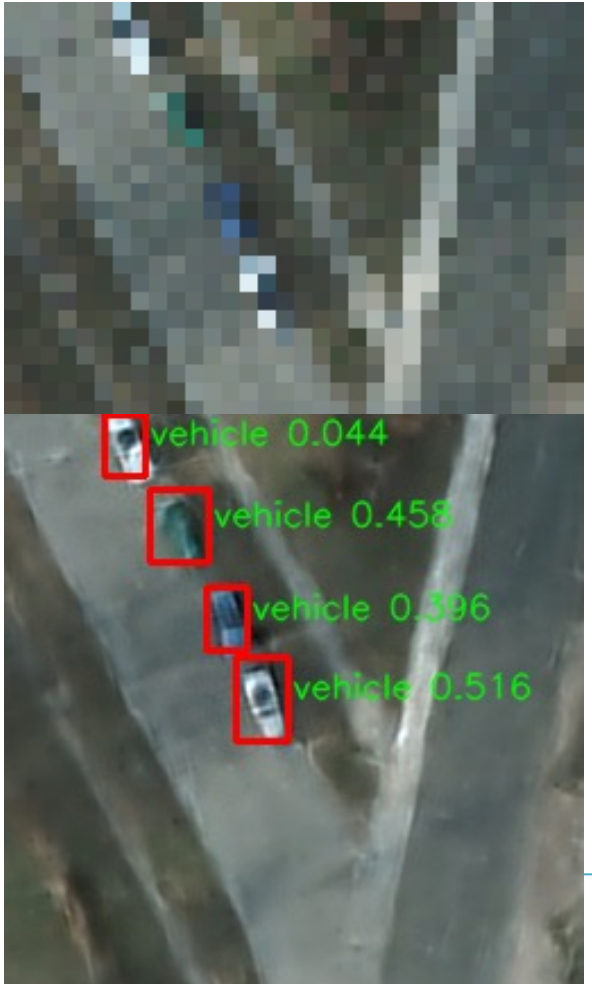
use super-resolution to (artificially) increase image resolution

Object detection-driven super-resolution network

- GAN for image super-resolution
- Cycle-GAN performs better
- with auxiliary tasks
  (object detection, semantic segmentation)

LR SR-CWGAN SR-CWGAN-Yolo HR

Method	IoU = 0.05	IoU = 0.25	IoU = 0.5
HR	96.36	93.57	82.14
Bicubic	22.80	17.40	09.53
EDSR	47.85	42.32	34.43
SR-WGAN	63.76	59.54	44.67
SR-CWGAN	66.72	62.82	47.18
SR-CWGAN-Yolo	**76.74**	**71.31**	**55.05**



CONTENTS OF THE LECTURES

- Deep Learning for Remote Sensing… see previous lectures (they were
  - Principles of Deep Learning
  - Applications to EO

- Lecture #
  - Tra
  - Eva

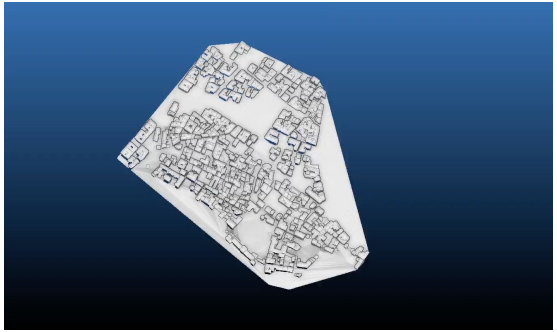
I. de Gelis, S. Lefèvre, T. Corpetti.
Change Detection in Urban Point Clouds: An Experimental Comparison with Simulated 3D Datasets.
Remote Sensing, 13(13):2629, 2021.

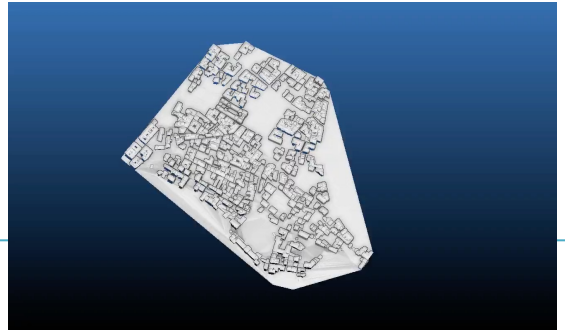
I. de Gelis, S. Lefèvre, T. Corpetti.
3D Urban Change Detection with Point Cloud Siamese Networks.
ISPRS Congress, 2021.

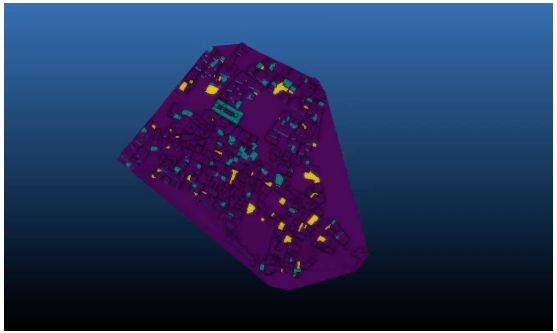
- Lecture # 2: Solutions for Complex Data
  - Focus on main EO tasks:
    semantic segmentation, object detection, change detection
  - Focus on some complex data:
    LiDAR, SAR, multi-view imagery, low-resolution data

URBAN CHANGE DETECTION BETWEEN 3D POINT CLOUDS

An important problem to assess urban evolution…
yet not enough data to train (and evaluate) deep networks







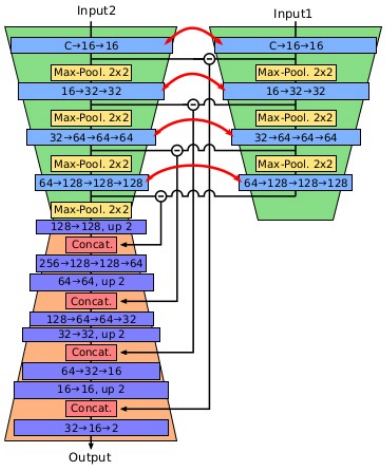
Urb3DCD
Public dataset available on DataPort
https://dx.doi.org/10.21227/2vsq-f173

Simulate ALS from 3D city model
Multiple subsets for bidat

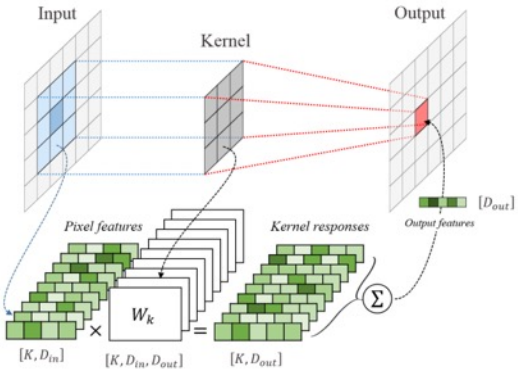
- Noise level
- Spatial resolution
- Scan angle
- Mimick multi-sensor data
- Small/medium/large training set

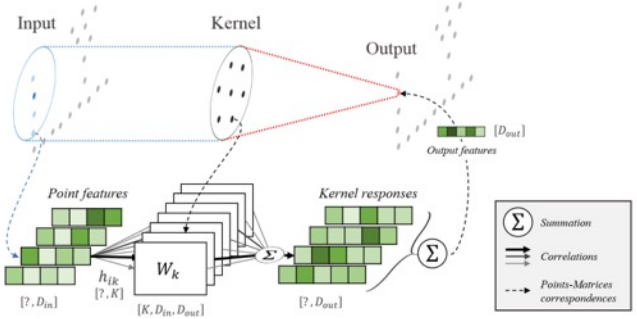
SIAMESE KPCONV DEEP NETWORK

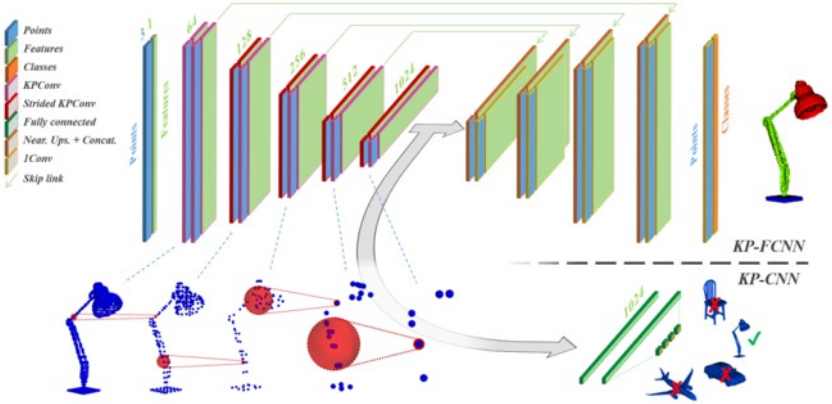
Coupling networks for change detection and for PC classification



Siamese for change detection
(Daudt et al, 2018)



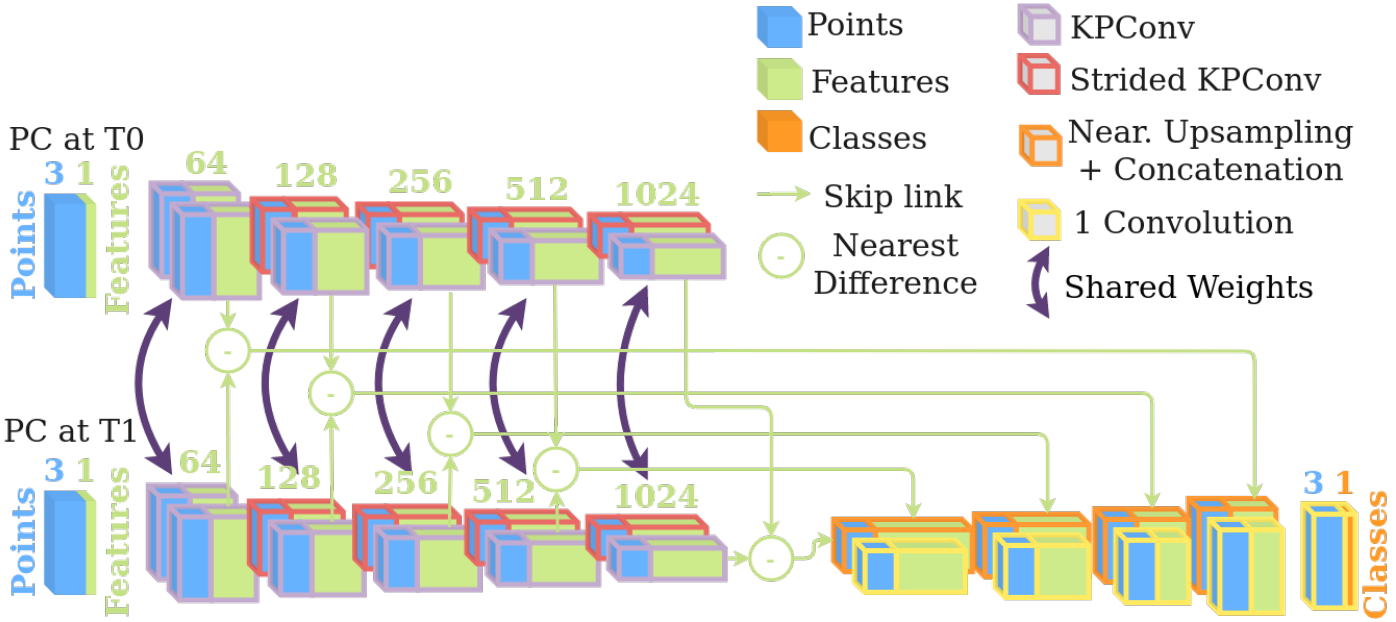




KPConv for point cloud analysis
(Thomas et al, 2019)

SIAMESE KPCONV DEEP NETWORK

Coupling networks for change detection and for PC classification



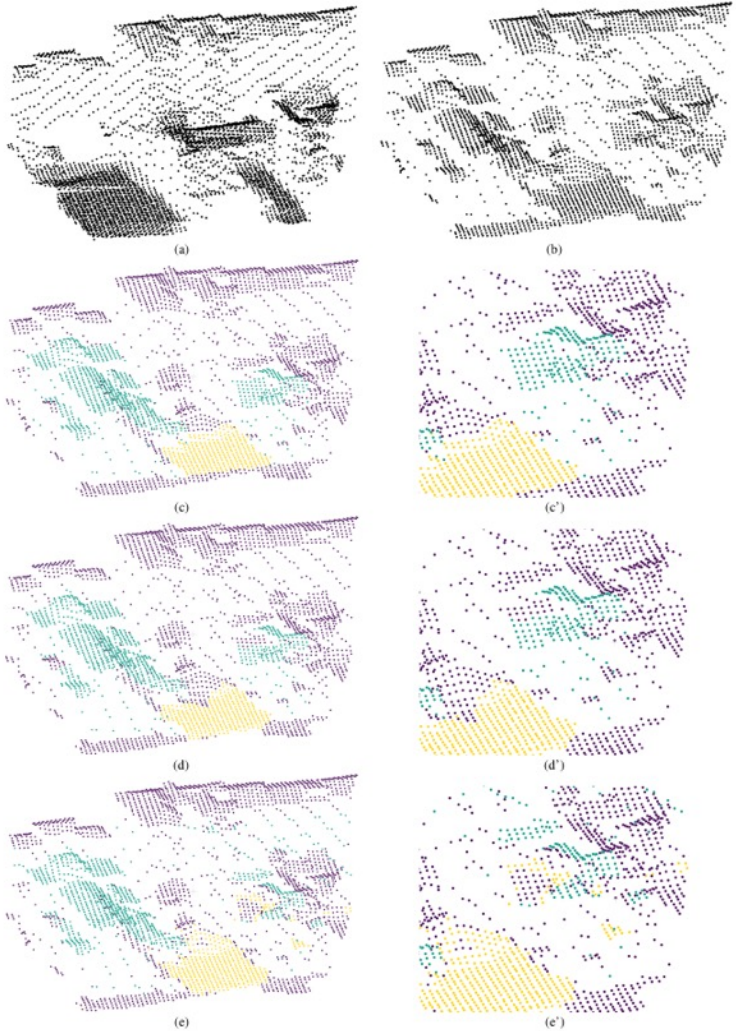


Figure 6. Visual change detection results. (a-b) the two input point clouds ; (c) simulated changes (purple: no change, blue: new construction, yellow: destruction) ; (d) our results; (e) results with (Tran et al., 2018) method; (') denote close-up views.

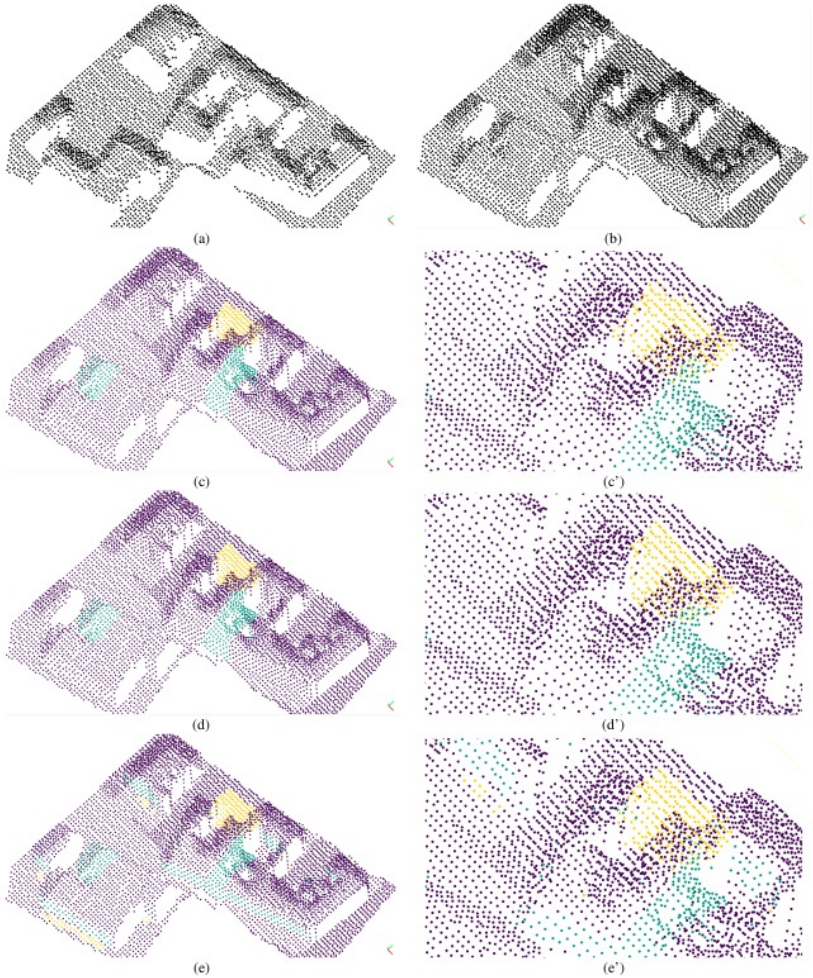
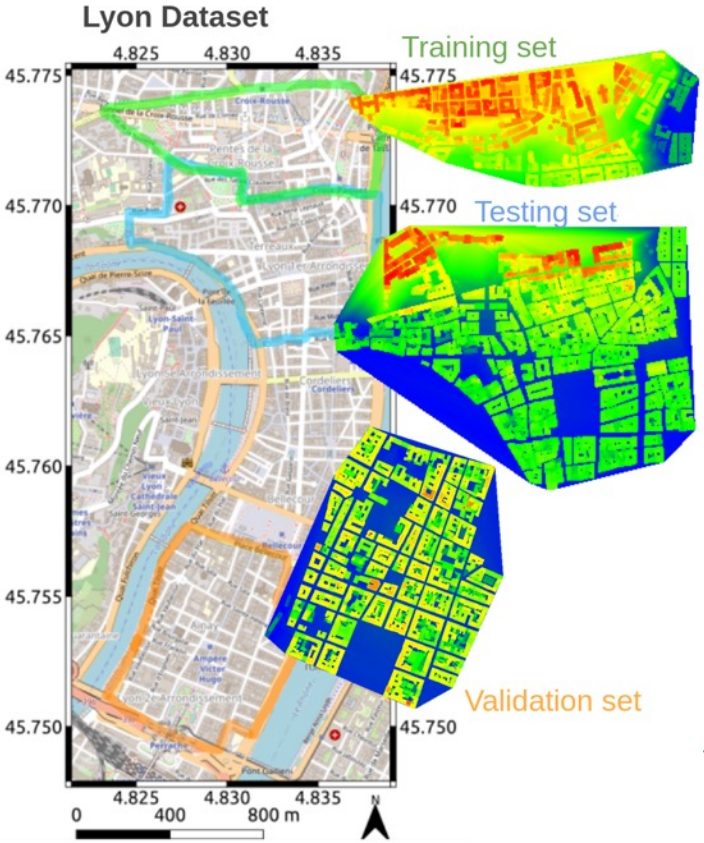


Figure 5. Visual change detection results. (a-b) the two input point clouds ; (c) simulated changes (purple: no change, blue: new construction, yellow: destruction) ; (d) our results; (e) results with (Tran et al., 2018) method; (') denote close-up views.

UMR IRISA

SIAMESE KPCONV DEEP NETWORK

Coupling networks for change detection and for PC classification



Metric (%)	Ours	(Tran et al., 2018)
mAcc	**96.24**	91.14
mIoU	**93.27**	74.53
mIoU Change	**90.22**	63.41

Table 3. Quantitative results (best in bold) of our Siamese KPConv network compared to RF + hand-crafted features.

Area	Method	Per class IoU (%)		
		Unchanged	New building	Destruction
1	Ours	**99.32**	**96.21**	**82.97**
	RF	98.29	80.08	70.81
2	Ours	**99.47**	**96.37**	**85.57**
	RF	95.54	52.99	61.35
3	Ours	**99.31**	**95.04**	**85.18**
	RF	96.46	64.57	58.37
Total	Ours	**99.37**	**95.87**	**84.57**
	RF	96.76	65.71	63.51

Table 4. Per area IoU scores (best in bold) of our Siamese KPConv network compared to RF + hand-crafted features.

CONTENTS OF THE LECTURES

- Deep Learning for Remote Sensing… see previous lectures (they were great!)
  - Principles of Deep Learning
  - Applications to EO

- Lecture #1: Good practices
  - Trainin
  - Eva

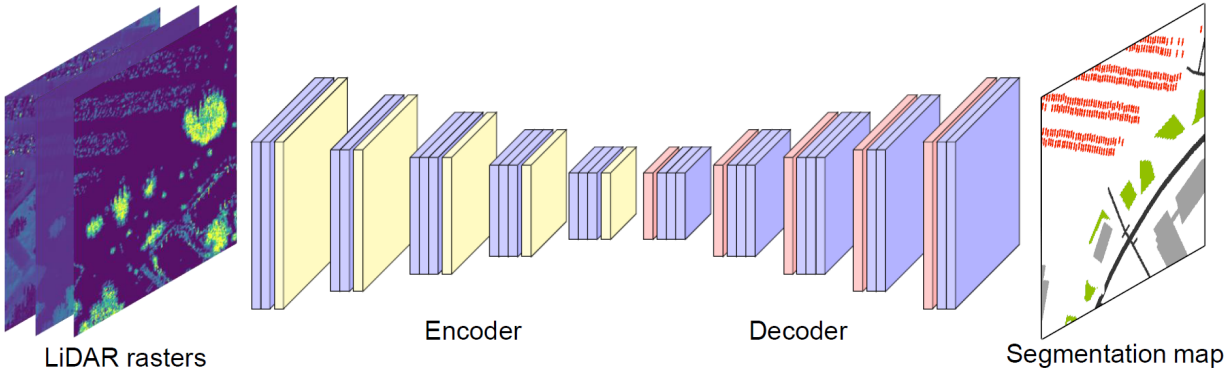
F. Guiotte, M.T. Pham, R. Dambreville, S. Lefèvre, T. Corpetti.
Semantic Segmentation of LiDAR Points Clouds: Rasterization Beyond Digital Elevation Models.
IEEE Geoscience and Remote Sensing Letters, 17(11):2016-2019, 2020.

- Lecture #2: Solutions for Complex Data
  - Focus on main EO tasks:
    semantic segmentation, object detection, change detection
  - Focus on some complex data:
    LiDAR, SAR, multi-view imagery, low-resolution data

DESIGNING/TRAINING 3D NETWORKS IS NOT STRAIGHTFORWARD

For most authors, LiDAR = DSM

LIDAR semantic segmentation is easy, even with pre-trained 2D networks



Rasterization pros

- Reduce complexity
- Regular sampling
- Prior known amount of data
- Reduce radiometric and altimetric artefacts thanks to aggregation of values

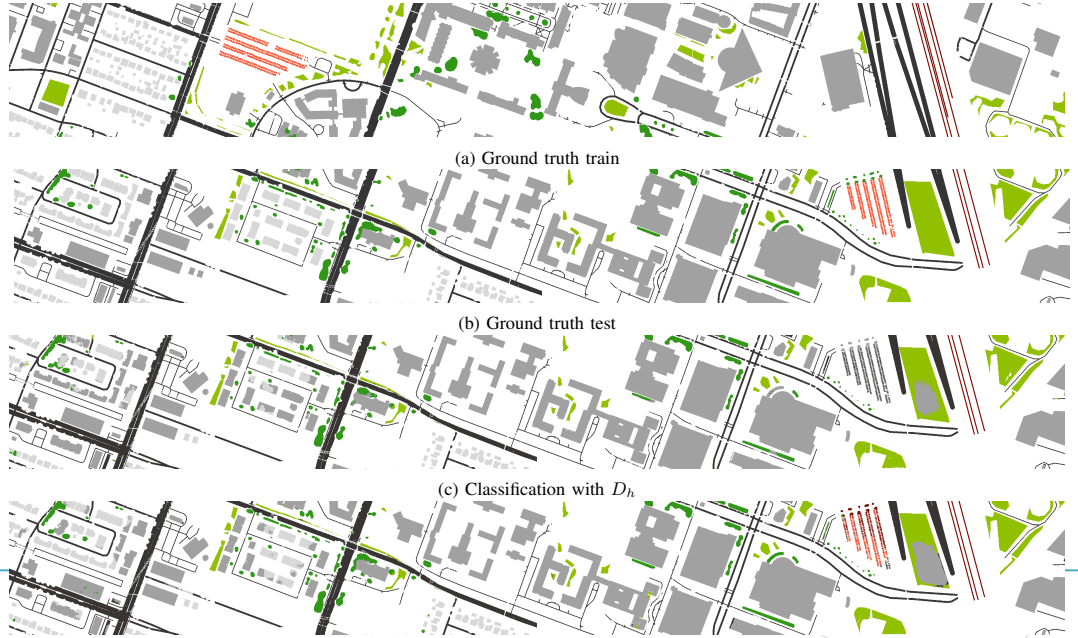
DESIGNING/TRAINING 3D NETWORKS IS NOT STRAIGHTFORWARD

Yet, DSM is only one raster among others:

- Max Z (D_h), Min Z (D_l)
- Intensity Max Z (I_h) , Min Z (I_l)
- # echoes (N)

Proposed approach

1. PC discretization
2. Feature extraction
3. Interpolation of empty cells



(a) Ground truth train

(b) Ground truth test

(c) Classification with D_h

(d) Classification with I_h, I_l, D_h

		Per-class accuracy (%)							Evaluation metrics		
		Grass	Trees	Residential	Non-res building	Roads	Cars	Trains	AA(%)	OA(%)	$\kappa(\times 100)$
1	D_h	**95.62**	77.66	44.18	98.72	**99.83**	0.09	100.00	73.72	90.13	85.24
2	N	16.08	92.48	46.50	97.63	95.39	6.50	100.00	64.94	87.27	78.37
3	I_l	56.94	89.35	41.96	**99.39**	97.16	98.81	99.94	83.36	86.98	80.10
4	D_l	90.93	89.67	22.09	99.12	99.57	0.00	99.94	71.57	87.67	81.31
5	I_h	59.82	95.15	47.78	98.27	96.43	**99.98**	99.96	85.34	87.54	81.14
6	$\{N, I_h, D_h\}$	93.15	**97.72**	39.47	98.99	99.66	13.42	100.00	77.49	90.66	86.01
7	$\{I_h, I_l, D_h\}$	81.64	94.12	76.95	98.36	99.38	57.10	100.00	**86.87**	**93.88**	**91.00**
8	$\{I_h, I_l, D_h, D_l\}$	87.63	92.84	60.20	93.32	96.98	68.16	100.00	86.44	92.31	88.57
9	$\{N, I_h, I_l, D_h, D_l\}$	79.13	95.35	**82.70**	96.28	99.19	10.05	100.00	80.38	92.42	88.97

TABLE I: Per-class accuracy, average accuracy (AA), overall accuracy (OA) and Cohen's Kappa coefficient (κ) for each feature and combination of them using Segnet.

Easy-to-use approach with informative feature maps

CONTENTS OF THE LECTURES

- Deep Learning for Remote Sensing… see previous lectures (they were great!)
  - Principles of Deep Learning
  - Applications to EO

- Lecture #1: Good practices
  - Training
  - Evaluation

More works!
But no remaining time…

- Lecture #2: Solutions for Complex Data
  - Focus on main EO tasks:
    semantic segmentation, object detection, change detection
  - Focus on some complex data:
    LiDAR, SAR, multi-view imagery, low-resolution data

TAKE-AWAY MESSAGE

Deep Learning = State-of-the-Art in Computer Vision, and in Remote Sensing as well!

But training a deep network might be tricky

- Quantitative results look astonishing?
  - ensure a valid evaluation protocol
- Label data are scarce?
  - train with unlabelled data as well
  - or explore ways to automatically generate label data (e.g. using ancillary sources)
- Numerous models are available
  - there is most probably already one fitting your needs
  - if not, consider fine-tune a generic model
  - If needed, you can build your own (simple) model
  - opt for open source code (EO Zoo still needed!)
  - stay tuned: DL is a very active field in CV, ML, & EO

TAKE-AWAY MESSAGE

Interested into (semi-supervised) semantic segmentation?

Nicolas Audebert 

- Public toolboxes: DeepHyperX and DeepNetsForEO
- Multimodal (RGB+DSM, RGB+OSM) early or late fusion
- Segment-to-detect
- Boundary-aware networks using distance transform
- DL4HSI survey, GAN for HSI

Javiera Castillo-Navarro 

- Public dataset: MiniFrance
- Semi-supervised networks with auxiliary self-trained tasks (e.g. BerundaNet)
- Energy-based Models (e.g. JEM)

ACKNOWLEDGEMENTS

PhD students & senior researchers from OBELIX

Nicolas Audebert, Ahmed Samy Nassar, Florent Guiotte (PhD completed)
Javiera Castillo-Navarro, Iris de Gelis (PhD in progress)

Minh-Tan Pham, Luc Courtrai

And remote colleagues

Bertrand Le Saux

Alexandre Boulch

Jan Dirk Wegner

Clément Deschesne

http://people.irisa.fr/Sebastien.Lefevre

www.irisa.fr @irisa_lab

Institut de Recherche en Informatique et Systèmes Aléatoires