ColorMapGAN: Unsupervised Domain Adaptation for Semantic Segmentation Using Color Mapping Generative Adversarial Networks

Onur Tasar

http://www-sop.inria.fr/members/Onur.Tasar
https://www.linkedin.com/in/onur-tasar-814ba7133

Université Côte d'Azur, INRIA, Titane team

1. Introduction
   - Semantic Segmentation
   - Limitations of the Traditional Approach
   - Domain Adaptation
   - Existing Approaches
2. Proposed Method
   - ColorMapGAN
   - Overall Framework for Domain Adaptation
3. Experiments
   - Experimental Setup
   - Results
4. Conclusion

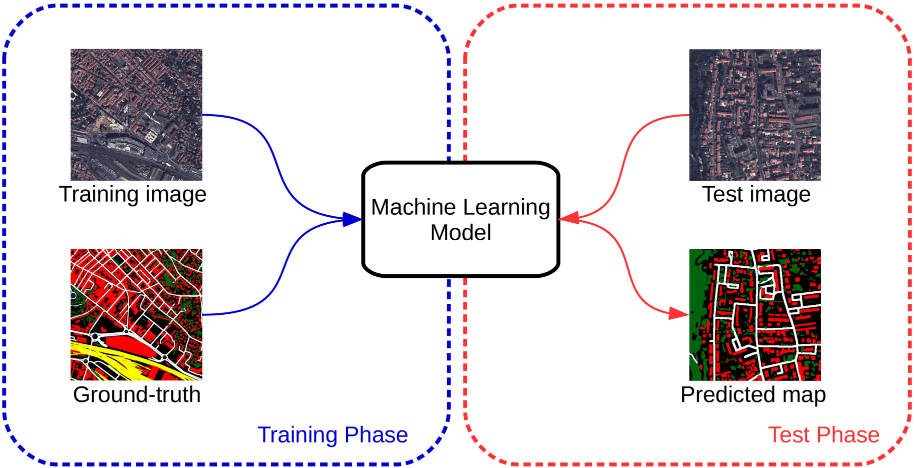
1 Introduction
- Semantic Segmentation
- Limitations of the Traditional Approach
- Domain Adaptation
- Existing Approaches

2 Proposed Method
- ColorMapGAN
- Overall Framework for Domain Adaptation

3 Experiments
- Experimental Setup
- Results

4 Conclusion

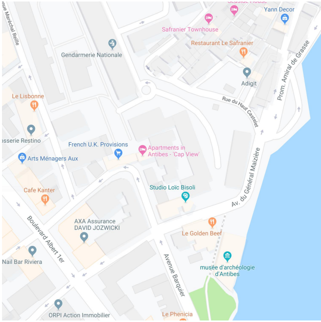
Semantic segmentation



Why is it important?

- Useful for a wide range of real-world applications:
  - Navigation, urban monitoring, etc.
- Manual annotation is too costly
  - The process needs to be automatized

Satellite view



Maps view

Problems of the traditional approach

- Using some part of a city as training, the other part as test data
- Training and test data have very similar statistics
- Almost never happens in real-world applications

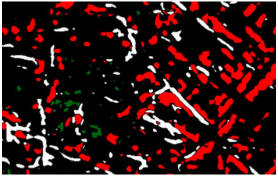
Vaihingen dataset

Domain adaptation definition

- New satellite images are collected from different locations of the world everyday
- Large domain shift between images because of atmospheric effects, differences in acquisition, seasonal changes, etc.
- The model generates unsatisfactory maps because of the domain shift
- Aim at developing a method that is robust to such shift


Béziers (tr. img.)


Roanne (test img.)


Roanne pred.

A possible solution

- Unpaired image to image translation (I2I). Generate fake training image that is:
  - semantically exactly the same as original training image
  - spectrally as similar as possible to test image

Béziers
Training city

Fake Béziers

Roanne
Test city

A possible solution

- Unpaired image to image translation (I2I). Generate fake training image that is:
  - semantically exactly the same as original training image
  - spectrally as similar as possible to test image

Béziers
Training city

Fake Béziers

Roanne
Test city

A possible solution

- Unpaired image to image translation (I2I). Generate fake training image that is:
  - semantically exactly the same as original training image
  - spectrally as similar as possible to test image

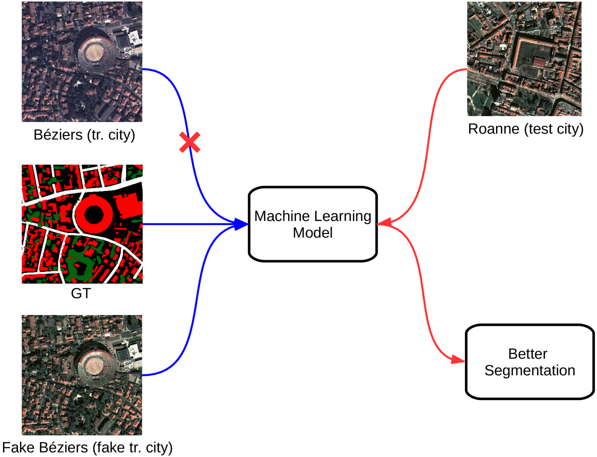
Béziers
Training city

Fake Béziers

Roanne
Test city

A possible solution contd

- Use fake image and GT for the original image to train a model



Existing unpaired I2I & domain adaptation approaches

Learning based approaches:
⇒ CycleGAN[1]
⇒ UNIT[3]
⇒ MUNIT[4]
⇒ DRIT[6]
⇒ AdaptSegNet[7]

Non-learning based approaches:
⇒ Histogram matching[2]
⇒ Color constancy algorithms (e.g., gray world[5])

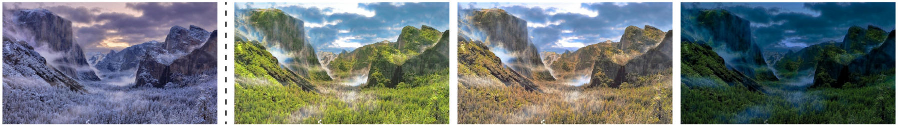
[1] Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks".
[2] Gonzalez and Woods, *Digital Image Processing (3rd Edition)*.
[3] Liu, Breuel, and Kautz, "Unsupervised image-to-image translation networks".
[4] Huang et al., "Multimodal unsupervised image-to-image translation".
[5] Buchsbaum, "A spatial processor model for object colour perception".
[6] Lee et al., "Diverse image-to-image translation via disentangled representations".
[7] Tsai et al., "Learning to adapt structured output space for semantic segmentation".

CycleGAN & UNIT

CycleGAN

UNIT

MUNIT & DRIT

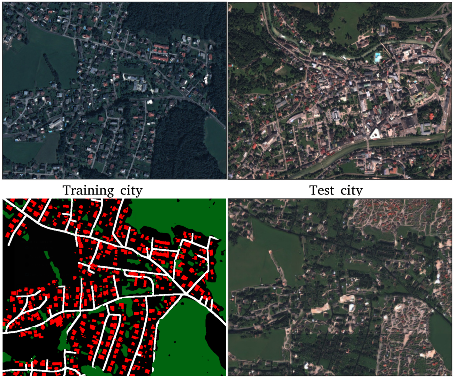


MUNIT

DRIT

Limitations of the existing I2I approaches

- Satellite images contain a lot of heterogeneous and complex structures
- When generating a fake city, I2I approaches:
  - can transfer the style of test image to training image successfully
  - are not able to keep original and fake training images semantically consistent



Training city | Test city

GT for training city | Fake training city

1. Introduction
   - Semantic Segmentation
   - Limitations of the Traditional Approach
   - Domain Adaptation
   - Existing Approaches
2. Proposed Method
   - ColorMapGAN
   - Overall Framework for Domain Adaptation
3. Experiments
   - Experimental Setup
   - Results
4. Conclusion

ColorMapGAN

- Do we need convolution, pooling, etc. operations?
- Can we map each color of training image to another one for style transfer?
- Generator of ColorMapGAN consists of only 1 matrix multiplication and 1 matrix addition


Training city


Test city


Fake training city

ColorMapGAN - Color mapping

- RGB is all the possible colors of training city
- R'G'B' the colors of fake training city
- W and K are learnable parameters. How to learn them?

RGB (256 x 256 x 256, 3)

0	0	0
0	0	1
0	0	2
⋮	⋮	⋮
255	255	255

W (256 x 256 x 256, 3)

$W_{0,0}$	$W_{0,1}$	$W_{0,2}$
$W_{1,0}$	$W_{1,1}$	$W_{1,2}$
$W_{2,0}$	$W_{2,1}$	$W_{2,2}$
⋮	⋮	⋮
$W_{16777215,0}$	$W_{16777215,1}$	$W_{16777215,2}$

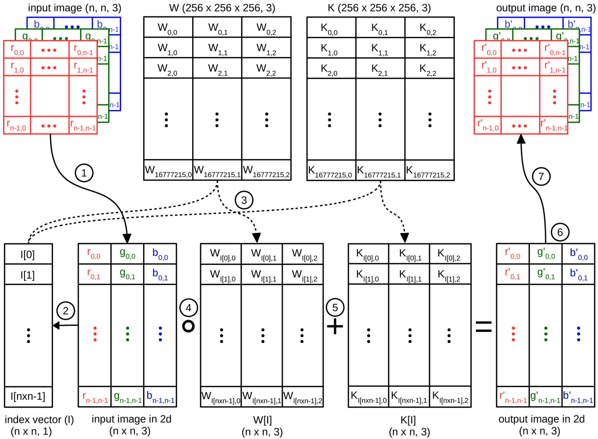
K (256 x 256 x 256, 3)

$K_{0,0}$	$K_{0,1}$	$K_{0,2}$
$K_{1,0}$	$K_{1,1}$	$K_{1,2}$
$K_{2,0}$	$K_{2,1}$	$K_{2,2}$
⋮	⋮	⋮
$K_{16777215,0}$	$K_{16777215,1}$	$K_{16777215,2}$

R'G'B' (256 x 256 x 256, 3)

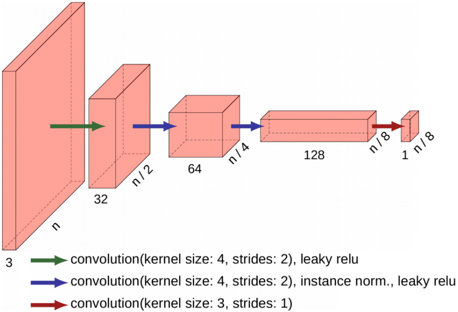
81	10	7
61	171	1
0	0	92
⋮	⋮	⋮
25	155	51

ColorMapGAN - Generator



1. convert the input image into a 2d matrix
2. find index of each color
3. retrieve the elements from W and K matrices using the index matrix
4. perform element-wise matrix multiplication
5. perform matrix addition operation
6. replace all the values larger than 1 by 1, and smaller than -1 by -1
7. reconstruct the output image

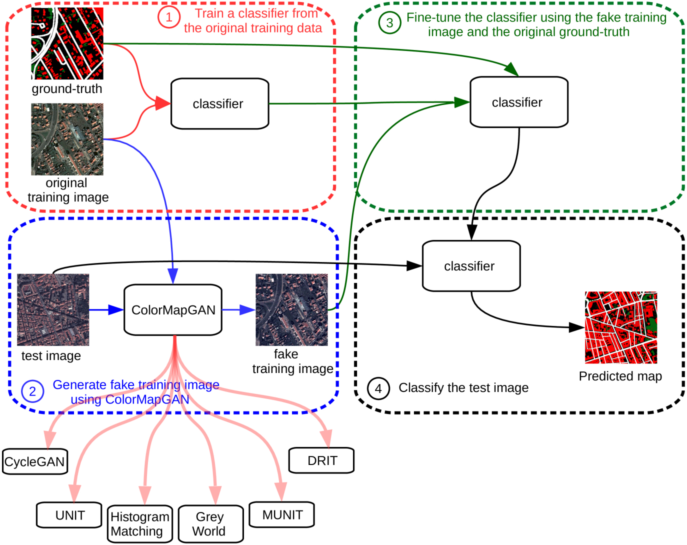
ColorMapGAN - Discriminator & Training details



- Use LSGAN[8] as the loss functions
- Use Adam optimizer to train generator and discriminator jointly

[8] Mao et al., "Least squares generative adversarial networks".

Overall Framework



1 Introduction
- Semantic Segmentation
- Limitations of the Traditional Approach
- Domain Adaptation
- Existing Approaches

2 Proposed Method
- ColorMapGAN
- Overall Framework for Domain Adaptation

3 Experiments
- Experimental Setup
- Results

4 Conclusion

Data set

- Luxcarta data set containing RGB, 8 bit, Pléiades images
- Spatial resolution is 1m
- GT for road, building, tree classes is provided

Bad Ischl (Austria)

Villach (Austria)

Béziers (France)

Roanne (France)

Data set contd.

The statistics for the Luxcarta data set.

City	# of patches	Area (km^2)	Class frequency (%)		
			building	road	tree
Bad Ischl	457	27.71	5.51	6.03	35.38
Villach	749	43.59	9.26	10.63	19.91
Béziers	407	25.75	19.09	17.62	10.91
Roanne	384	25.84	18.44	8.33	14.78

- Use 256×256 training patches
- Perform city to city domain adaptation between the following city pairs:
  - Bad Ischl ⇔ Villach
  - Béziers ⇔ Roanne

Original and Fake Bad Ischl (test city in pair1: Villach)

Villach(test)

Bad Ischl(tr.)

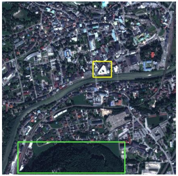
CycleGAN

UNIT

MUNIT

DRIT

Gray world

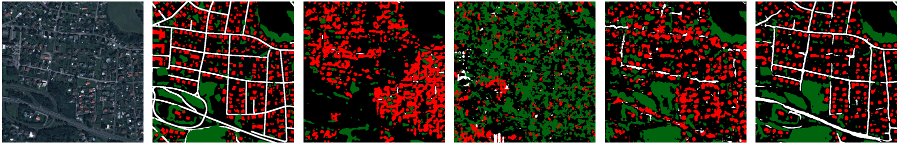


Hist. match.

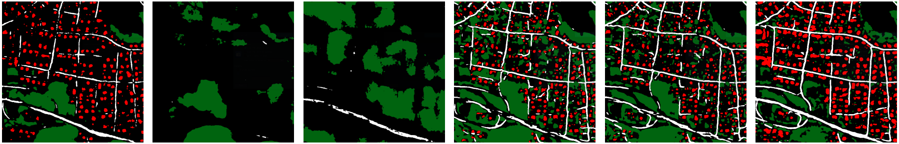
ColorMapGAN

Predictions for Villach (training city in pair1: Bad Ischl)

- We did not add "our framework with" statement for the sake of simplicity



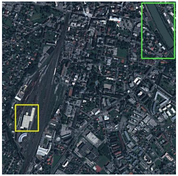
Villach | GT | U-net | AdaptSN S | AdaptSN M | CycleGAN



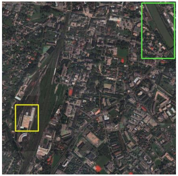
UNIT | MUNIT | DRIT | Gray world | Hist. match. | ColorM.GAN

Original and Fake Villach (test city in pair1: Bad Ischl)

Bad Ischl(test)



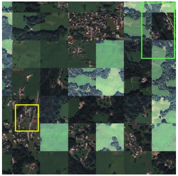
Villach(tr.)



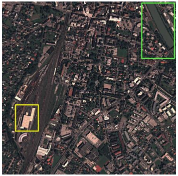
CycleGAN

UNIT

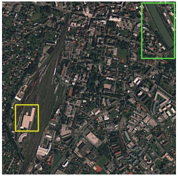
MUNIT



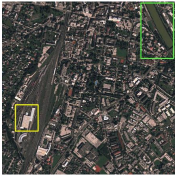
DRIT



Gray world



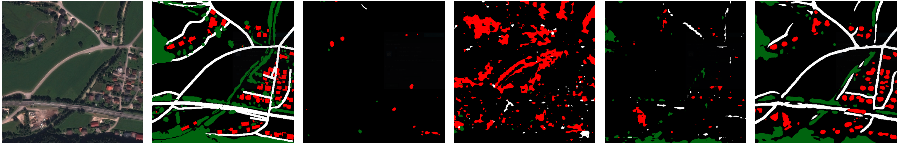
Hist. match.



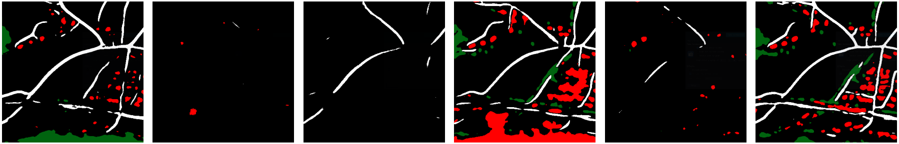
ColorMapGAN

Predictions for Bad Ischl (training city in pair1: Villach)

- We did not add "our framework with" statement for the sake of simplicity



Bad Ischl | GT | U-net | AdaptSN S | AdaptSN M | CycleGAN



UNIT | MUNIT | DRIT | Gray world | Hist. match. | ColorM.GAN

Original and Fake Béziers (test city in pair2: Roanne)

Roanne(test)

Béziers(tr.)

CycleGAN

UNIT

MUNIT

DRIT

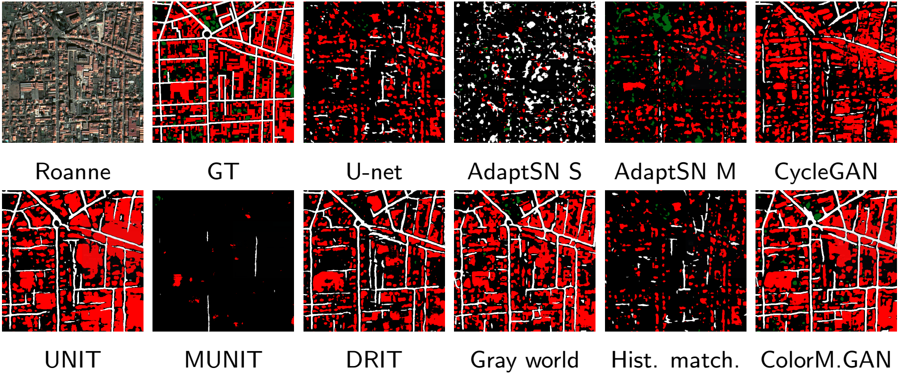
Gray world

Hist. match.

ColorMapGAN

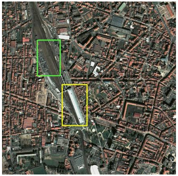
Predictions for Roanne (training city in pair2: Béziers)

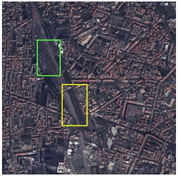
- We did not add "our framework with" statement for the sake of simplicity

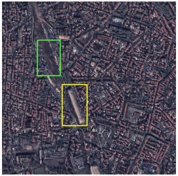


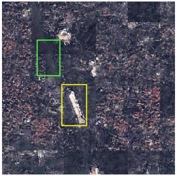
Original and Fake Roanne (test city in pair2: Béziers)

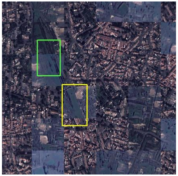

Béziers(test)

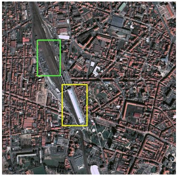

Roanne(tr.)

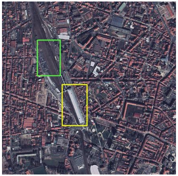

CycleGAN

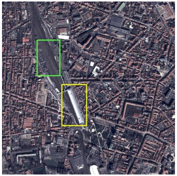

UNIT


MUNIT


DRIT

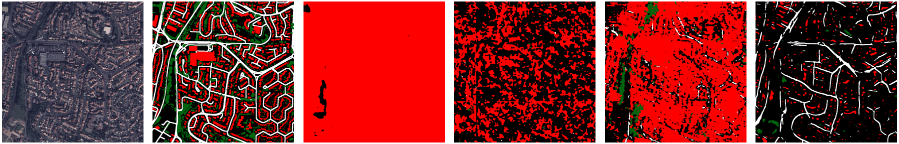

Gray world


Hist. match.

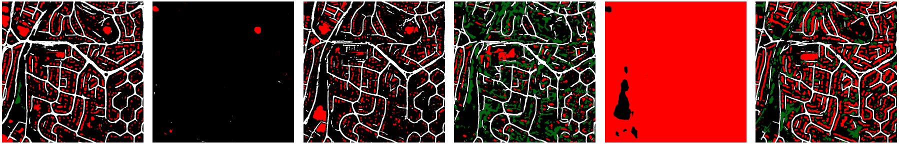

ColorMapGAN

Predictions for Béziers (training city in pair2: Roanne)

- We did not add "our framework with" statement for the sake of simplicity



Béziers GT U-net AdaptSN S AdaptSN M CycleGAN



UNIT MUNIT DRIT Gray world Hist. match. ColorM.GAN

Quantitative Results

IoU scores for the test cities in pair 1

	Method	Training: Bad Ischl, Test: Villach				Training: Villach, Test: Bad Ischl			
		building	road	tree	Overall	building	road	tree	Overall
	U-net	23.61	0.91	40.53	21.68	5.84	0.24	0.50	2.19
	AdaptSegNet Single	6.01	4.37	10.43	6.94	3.06	2.71	10.23	5.33
	AdaptSegNet Multi	24.59	9.02	56.08	29.86	14.26	4.46	24.66	14.46
Our proposed framework with	CycleGAN	43.03	28.96	**68.86**	46.95	43.62	38.69	**71.68**	**51.33**
	UNIT	30.86	15.84	63.00	36.57	19.29	36.83	35.57	30.56
	MUNIT	0.02	1.38	47.23	16.21	6.20	0.13	0.05	2.13
	DRIT	0.01	3.96	8.72	4.23	0.00	10.19	0.01	3.40
	Gray world	25.19	26.43	56.15	35.92	29.55	24.80	46.41	33.58
	Histogram matching	24.95	29.34	61.59	38.63	6.45	0.92	1.28	2.88
	ColorMapGAN (ours)	**48.47**	**37.82**	58.92	**48.40**	**49.16**	**41.75**	59.84	50.25

IoU scores for the test cities in pair 2

	Method	Training: Béziers, Test: Roanne				Training: Roanne, Test: Béziers			
		building	road	tree	Overall	building	road	tree	Overall
	U-net	26.13	11.16	7.79	15.03	19.85	0.00	0.00	6.62
	AdaptSegNet Single	6.61	11.05	3.37	7.01	11.07	4.19	3.71	6.32
	AdaptSegNet Multi	22.42	5.87	17.84	15.37	24.27	10.88	10.45	15.20
Our proposed framework with	CycleGAN	18.19	24.28	0.19	14.22	9.92	16.00	0.54	8.82
	UNIT	41.99	38.47	0.39	26.95	29.99	**39.19**	3.11	24.10
	MUNIT	10.17	1.66	0.31	4.05	7.49	0.84	0.13	2.82
	DRIT	42.16	41.77	1.18	28.37	25.73	36.54	0.68	20.98
	Gray world	51.47	40.42	18.25	36.71	14.61	31.32	**21.99**	22.64
	Histogram matching	18.64	9.87	4.81	11.11	20.63	0.02	0.00	6.88
	ColorMapGAN (ours)	**55.60**	**44.66**	**28.39**	**42.88**	**47.12**	35.18	21.91	**34.74**

ColorMapGAN vs CycleGAN

- CycleGAN is unstable; original and fake training cities
  - in pair 2 are semantically inconsistent
  - in pair 1 are semantically consistent
- Even when it generates consistent output,
  - resolution of the fake image is lower
  - the output has artifacts

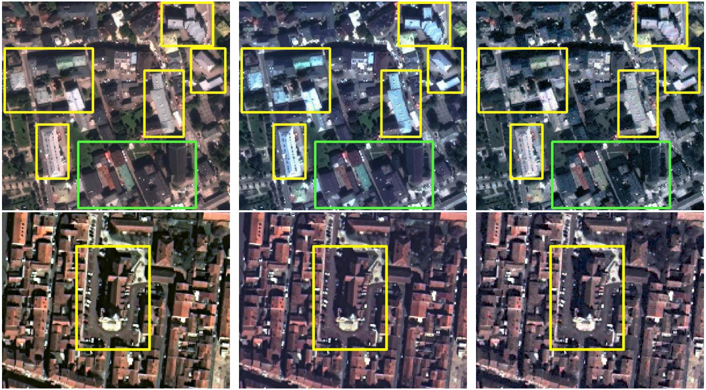
Villach

CycleGAN

ColorMapGAN

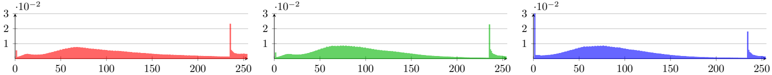
ColorMapGAN vs Histogram Matching

- Hist. match. does not take into account the contextual information

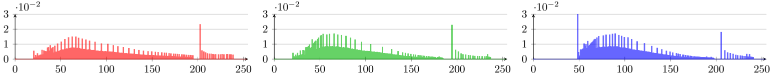


Roanne Hist. match. ColorMapGAN

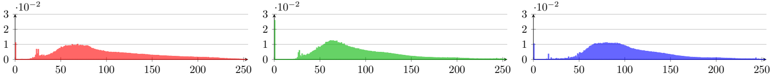
ColorMapGAN vs Histogram Matching contd.



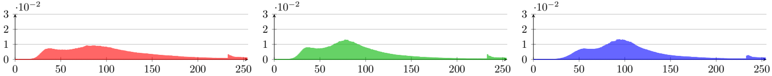
Roanne



Fake Roanne generated by histogram matching



Fake Roanne generated by ColorMapGAN



Béziers

Color histograms of building pixels. Red, green, and blue bins represent the histograms for red, green, and blue channels, respectively.

Running Times

- Note that we optimized ColorMapGAN for only 8,000 iterations
- It took around 6.5 minutes to train it

Training times for generating fake cities.

Method	Training time for 1 Iter. (secs.)
CycleGAN	0.11
UNIT	0.47
MUNIT	0.45
DRIT	0.29
ColorMapGAN	**0.05**

Execution times for generating fake cities.

City	Execution time (seconds)	
	Gray world	Histogram matching
Bad Ischl	**1.46**	19.77
Villach	**1.89**	26.78
Béziers	**1.37**	18.05
Roanne	**1.24**	20.32

Conclusion

- We show that spectral shift between images can easily be corrected by ColorMapGAN
- We proved its effectiveness in terms of qualitative, quantitative, and running times on two city to city pairs
- We believe the proposed approach can be used other real-world problems such as change detection