OFF THE SHELF DEEP LEARNING PIPELINE FOR REMOTE SENSING APPLICATIONS

Rachit TRIPATHI[1], *Adrien CHAN-HON-TONG*[2] *and Alexandre BOULCH*[2]

1: QuantCube Technology 2: ONERA, the french aerospace lab

ABSTRACT

Designing specific index for a some remote sensing applications require a large research effort not scalable to the multitude of applications.

Inversely, using off the shelf deep learning pipeline could be good enough for some applications.

We describe off the shelf deep learning application on the 2017 data fusion contest (IEEE-IGARSS) for local climate zone estimation. While being completely non expert to local climate zone estimation, and while having only few meta parameters, these pipelines reach honorable scores on this dataset compared to hard to tune winner pipeline of the challenge.

***Index Terms*—** deep learning, remote sensing

1. INTRODUCTION

The popularization of remote sensing images (e.g. the free availability of sentinel images) could allow a rupture for large remote sensing applications including climate observation, biomass estimation, drought monitoring... However, seeing the large spectrum of possible applications of remote sensing images, we can wonder about the research effort to correctly extract information from these new data. Designing specific index to handle specific problem may not be an scalable way to take full advantages of all these newly available data.

Inversely, we claim that using off the shelf deep learning pipeline could be good enough for some applications.

To argue our statement, we present here experiments done on the data fusion contest 2017 (IEEE-IGARSS). Data fusion contest (DFC) are a set of challenges of the remote sensing community. The 2017 challenge is about predicting Local Climate Zones (LCZ) from training cities to unknown cities [1]. LCZ aims to offer a typology of both landscape and urban locations designed to study heat island and heat propagation in cities. The LCZ of a location is completely defined by it landscape/urban configuration e.g. high dense metallic building will lead to LCZ named *dense high rise*.

We show that off the shelf deep learning pipelines can reach honorable scores for LCZ estimation (compare to the state of the art) without requiring careful tunning.

In the context of the DFC2017, provided inputs for doing this LCZ prediction are remote sensing images and crow based map: 9 bands *landsat* at a resolution of 100m with multiple images per city, 9 bands sentinel2 images at 100m, *openstreetmap* information available for the selected cities (rasterized at 5m) and not registered 9 bands *sentinel2* images at 5m.

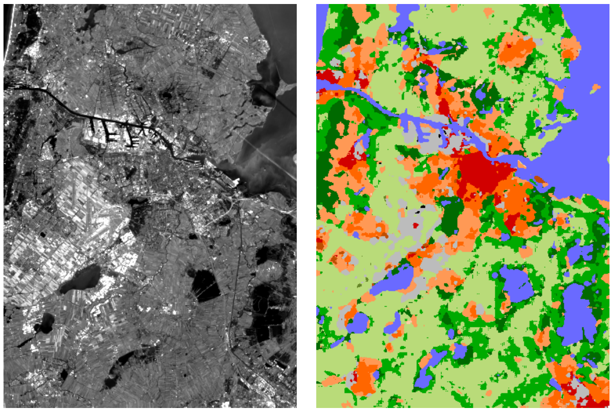

Fig. 1. Landsat 8 data (band 4) and predicted LCZ map on the test city of Amsterdam.

Currently, even before the data fusion contest, there were works on LCZ estimation from public data (e.g [2]). With the challenge, there were a large research effort on this problem [3, 4, 5, 6]. In [3], only landsat and osm are handled. Both basic and specific features are extracted from images. Basic features are mean-variance. Specific ones are mostly built on infrared measure. Thus, a first step of atmospheric correction is performed on landsat images to improve infrared images. Then well known normalized difference index (we will call it *ndi*) like NDVI, NDWI, MNDWI, NDBI, BSI and WRI (see [7] for a brief review) are extracted from images. In addition, morphological profiles are extracted by combining osm, NDVI and morphologic operator. Then, two kind of ensemble classifier are trained on these features.

Seeing [2, 3] ([4, 5, 6] are currently not available), we can argue that most of these LCZ papers either needs careful tuning or use very specific index designed especially for LCZ. Here, we offer instead very generic to off the shelf deep learning pipelines to infer LCZ from multi modal data.

Table 1. Results of the leave one city out

method	berlin	HK	paris	rome	sao paulo	average
images (1vsall)	42%	25%	74%	22%	43%	38%
osm (1vsall)	47%	43%	59%	33%	21%	36%
images + osm (1vsall)	53%	51%	72%	30%	54%	48%
cnn + svm (raw + osm)	50%	52%	73%	33%	68%	51%
images + ndi + osm	51%	53%	73%	34%	68%	52%
cnn + svm (ndi + osm)	57%	53%	67%	48%	52%	54%
CNN Pyramid Pooling	71%	67%	69%	51%	80 %	67.6%

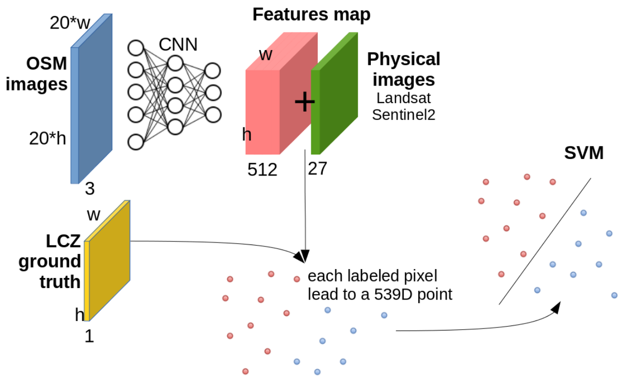

Fig. 2. scheme of the cnn+svm pipeline.

2. DEEP LEARNING FOR LCZ

2.1. CNN + SVM

The first pipeline is inspired from [8]. It is simply built by extracting a set of feature maps directly or with convolutional deep neural network (CNN) and using support vector machine (SVM) to perform pixelwise classification (precisely libsvm with 1vs1, linear kernel and default parameter [9]). Notice that this pipeline has 0 parameter, and thus, is a good example of off the shelf pipeline (see figure 2). Code is available here https://github.com/achanhon/CNN_SVM_for_DFC2017.

Raw images are used directly as features. For landsat data, we compute the mean and variance maps for each channel, in such a way that we exploit the multiple acquisitions. Then, we concatenate all provided bands for both landsat and sentinel leading to a 27 image based features per 100m pixel.

Additional features are generated using VGG16 [10] initialized imagenet weights on OSM data. We compute an ad hoc mask per city from osm data: it is formed using building, green area from landuse, and road (pixel value is either 0 or 255 depending on the presence of the item in osm). The features are extracted at several layers and rescaled to 100m resolution. In our experiment, osm typically provides 512 features per 100m pixel. Notice that, we do not train the cnn from an imagenet initialization, instead we just use the pretrained version without adjustment in the convolution weight (only small change have been done on the pooling structure to take into account the difference in resolution between images and osm).

2.2. Late and cascaded fusion with CNN

We used OSM and sentinel2. We design a specific networks inspired from U-Net [11] for each data sources, each with more pooling than upsampling as data are more resolved than label maps. Then, we concatenate resulting maps and forward it in a second network.

In late fusion, this second network is just a series of convolutions (since convolutions does not reduce a size of feature maps, we can use them directly to predict the labels). It is a *late* fusion because each modality is processed independently and only high level representation are combined at the top of the network.

We also evaluated cascaded fusion in which the second network is large. Two architecture have been tested for this second network.

1. U-Net (again): we concatenate upsampled lower layer with top ones to predict the labels.

2. Pyramid Pooling like: Inspired from [12], we implement a network with different levels of pooling along with upscaling and concatenation plus a final convolution to get the prediction.

2.3. Early fusion with CNN

The final approach uses all the data available in the challenge with additional Sentinel 1 data. The Sentinel 1 composite (S1 composite) is a three channel composite: VV polarization for ascending acquisition, VH for ascending and descending and VV for descending. The final product is the mean over the year 2016.

The CNN architecture used in this section is based on SegNet [13]. We set up an encoder for each input type (Sentinel 1 and 2, Landsat 8 and OSM). The decoder input is the concatenation of all coded signals. The reference signal (the one use for unpooling operation) is Landsat 8. Compared to the late fusion (previous section), we merge features instead of activations that why we can speak of early fusion. See figure 3 for visual representation of the 3 different CNN pipelines.

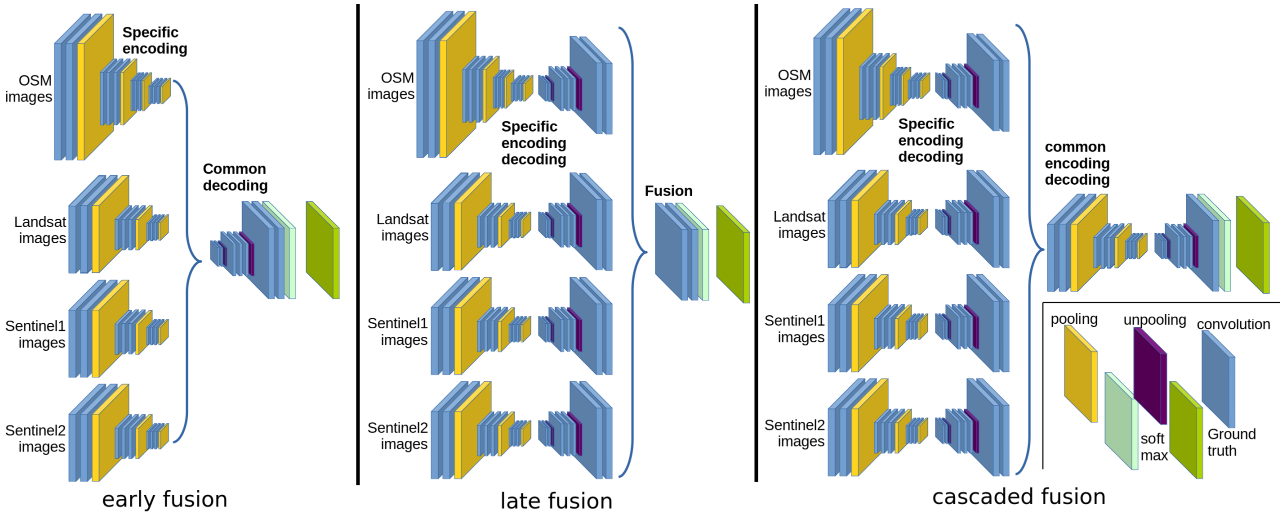

Fig. 3. Visualization of the 3 different cnn pipelines.
This is a schematic visualization and not the real architectures of the 3 pipelines. In all these 3 pipelines, weight are optimized by stochastic gradient descent (and not just restored from pretrained model). Training is done in classic fashion: forwarding the input, we get an output which has the same shape as expected ground truth ; a loss measures the distance between the produced output and the ground truth resulting in a gradient which is computed by backpropagated across the network and used to update all weights.

Table 2. Results on the test database

method	test
cnn + svm (raw + osm)	58%
cnn + svm (ndi + osm)	57%
fusion Pyramid Pool.	52%
Early fusion	56.6 %
Early fusion additional cities	**64.3** %

We trained two models: one on the cities of the DFC2017 training set and, in order to estimate the influence of the training data set size, one with additional cities: Dublin, Houston, Sydney, Vancouver and Warsaw. The ground truth associated with these cities is denser but with poor quality: coarse areas and noised labels.

3. EXPERIMENT

Following the rules of the 2017 data fusion contest, we evaluate the quality of the LCZ estimation by measuring the pixel wise accuracy (this is so a semantic segmentation problem).

The server evaluation results are presented in table 2 (notice that inspired from [3], we both use raw image and ndi in cnn+svm).

In addition, to the server evaluation, we also provide a leave one city out protocol: all training cities except one are used to train the model which is then applied to the excluded city, this operation being done for all cities. Leave one city out results (when available) are detailed in table 1 with an *average* result. In order to penalize, very unstable results across cities we weight the worse accuracy by a factor 2 in the *average* result.

4. DISCUSSION

The main result is that these off the shelf pipelines compare honorably, in our opinion, to hard to tune state of the art of LCZ estimation which is between 69 to 74% (see the output of [1]).

Indeed, due to small size of the ground, heavy CNN models like late or cascaded fusion strongly overfits and thus are overcome by models consisting in training only the SVM on the test set. This is consistent with the idea that deep learning is mainly relevant when large amount of data/ground truth is available. However, even in context with small amount of data, using pretrained cnn can provide interesting result (cnn+svm still achieves 58%). And, the required size of the dataset may not be that high: in our experiment, additional training data consisting in only 4 new cities gives a real boost in performance while being corrupted training data. Thus, we can be not so worry about preventing the model from overfitting.

Finally, we notice that the osm and image data are very complementary: both images and osm alone reach low performance probably because some classes are not distinguishable using only one modality (water is not present in osm resulting in a complete impossibility to predict at least this classe) and image may not be sufficiently resolved to infer density and thus elevation of building without osm. By the way, it is still not clear is image+osm are sufficient to distinguish between some classes like *middle rise* and *low rise*.

5. CONCLUSION

In our opinion, the main result of this work is that our off the shelf pipeline reach honorable results seeing the state of the art without requiring large tuning.

Off course (may be hopefully) designed index and algorithm performs still better than off the shelf deep learning (at least in our experiment ndi largely increases performance stability over different cities). But, this example highlights the interest of off the shelf deep learning pipeline to take advantage of newly available remote sensing image.

Thus, we argue that off the shelf deep learning pipelines may be more and more present for remote sensing applications.

Acknowledgment

As user of the DFC2017 data, the authors would like to thank the WUDAPT (http://www.wudapt.org/) and GeoWIKI (http://geo-wiki.org/) initiatives for providing the data packages used in this study, the DASE benchmarking platform (http://dase.ticinumaerospace.com/), and the IEEE GRSS Image Analysis and Data Fusion Technical Committee. Landsat 8 data available from the U.S. Geological Survey (https://www.usgs.gov/). OpenStreetMap Data OpenStreetMap contributors, available under the Open Database Licence http://www.openstreetmap.org/copyright. Original Copernicus Sentinel Data 2016 available from the European Space Agency (https://sentinel.esa.int).

6. REFERENCES

[1] D. Tuia, G. Moser, B. Le Saux, B. Bechtel, and L. See, “2017 ieee grss data fusion contest: Open data for global multimodal land use classification [technical committees],” *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 1, pp. 70–73, March 2017.

[2] P. Lopes, C. Fonte, L. See, and B. Bechtel, “Using openstreetmap data to assist in the creation of lcz maps,” in *2017 Joint Urban Remote Sensing Event (JURSE)*, March 2017, pp. 1–4.

[3] Naoto Yokoya, Pedram Ghamisi, and Junshi Xia, “The data fusion contest 2017: Open data for global multimodal land use classification,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[4] S. Sukhanov, I. Tankoyeu, J. Louradour, R. Heremans, D. Trofimova, and C. Debes, “Multilevel ensembling for local climate zones classification,” 2017, IEEE International Geoscience and Remote Sensing Symposium (IGARSS).

[5] Camila Souza dos Anjos, Marielcio Goncalves Lacerda, Leidiane do Livramento Andrade, and Roberto Neves Salles, “Classification of urban environments using feature extraction and random forest,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[6] Yong Xu, Fan Ma, Deyu Meng, Chao Ren, and Yee Leung, “A co-training approach to the classification of local climate zones with multi-source data,” in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[7] Komeil Rokni, Anuar Ahmad, Ali Selamat, and Sharifeh Hazini, “Water feature extraction and change detection using multitemporal landsat imagery,” in *Remote Sensing*, 2014.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Computer Vision and Pattern Recognition*, 2014.

[9] T. Joachims, T. Finley, and Chun-Nam J. Yu, “Cutting-plane training of structural svms,” in *Machine Learning*, 2009.

[10] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *CoRR*, 2014.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, Springer International Publishing, 2015.

[12] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia, “Pyramid scene parsing network,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[13] Vijay Badrinarayanan, Ankur Handa, and Roberto Cipolla, “Segnet a deep convolutional encoder decoder architecture for robust semantic pixelwise labelling,” in *arXiv preprint*, 2015.