Areal Team project proposal

Areal Team
areal@chalearn.org

David Biard
david.biard@u-psud.fr

Robin Duraz
robin.duraz@u-psud.fr

Samuel Berrien
samuel.berrien@u-psud.fr

Hao Liu
haoliuxdu@gmail.com

Trung Vu-Thanh
trung.vu-thanh@u-psud.fr

Théo Cornille
theocornille3@gmail.com

Guillaume Charpiat
Mentor,
guillaume.charpiat@inria.fr

Nicolas Girard
Mentor,
nicolas.girard@inria.fr

Yuliya Tarabalka
Mentor,
yuliya.tarabalka@inria.fr

The Inria Aerial Image Labeling addresses a core topic in remote sensing: the automatic pixelwise labeling of aerial imagery (link to paper).

1 Background

Since aerial imagery services and high resolution appeared, aerial imagery has become of the most important components of various industries. Energy, mining, military situation, disaster management, urban planning and more industries as well as other organizations in emergency situations can make use of aerial images to enhance their productivity and quality of work. One of the most expensive and time-consuming tasks required to use images is the labeling task, especially the detection of buildings.
Recent breakthroughs in image understanding techniques using the deep learning methods and improvements in hardware like GPUs have opened the way for people to experiment with different approaches and techniques.
Furthermore, as we can see with figures 1 & 2, binary classification (it is a building Vs it isn't) which does not seem too complicated can easily cause simple models, like a basic CNN, to fail in their classification. In this, almost the whole is classified as a building, which it obviously isn't.
Our project tackles the issue of the semantic segmentation of satellite images by trying to classify pixel by pixel, for each image, whether the pixel represents a building or not. We intend to develop a complex neural network suitable for this task, or try other available models to see if they can be appropriate for these kinds of tasks. We will train our models on three-dimensional images, with their corresponding label being a binary mask, two-dimensional matrices with ones for pixels representing buildings, and zero otherwise. With a satellite image as input, our network then will be able to output a binary mask representing its predictions for each pixel.

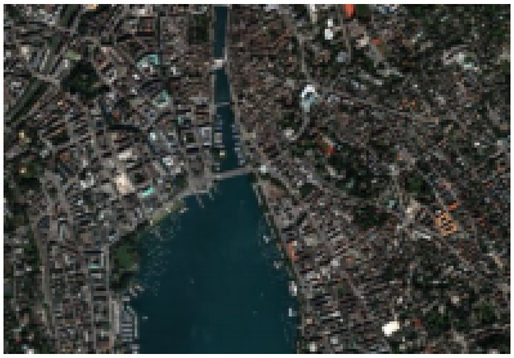


Fig. 1: Original image - Zurich Lake

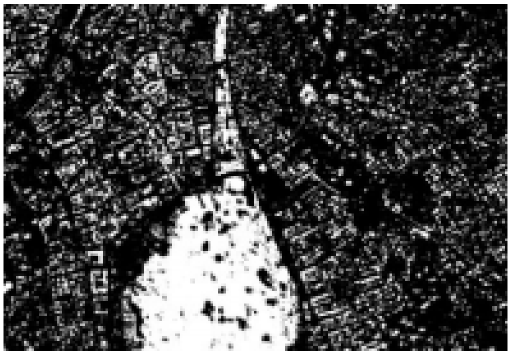


Fig. 2: Classification made by a classical CNN

2 Material & Method

In order to do this project, we have a train set of 180 5000x5000 TIF images. Those images are separated in 5 towns, each having 36 images. Each image is about 75 MB, totaling around 14 GB for all our data which is much too high to fulfill our requirements. Hence, we needed to reduce its size. First, we decided to compress images in JPEG format, gaining a factor of more than 10, to end up with around 1.4 GB. Then, we decided to divide each image in 25 smaller 1000x1000 images for an easier time training with smaller images. It doesn't change the whole size of the data, but skill makes it easier to use than bigger images. Then, we tried to use grayscale images instead of RGB, to gain a factor 3. The results were satisfying enough: given well chosen parameters of our neural networks, it could go up to more than 80% accuracy, which is good enough for our challenge. Thanks to that, we got further down to less than 500MB. We just have to take about 60% of our 1000x1000 images to go down beyond the required 300MB.

In terms of metrics of assessment, we will keep the metrics used with this database, i.e. IoU & Accuracy. IoU, in this case, represents intersection over union, which is how many pixels are classified as buildings, in both predictions and labels, over how many pixels are classified as buildings, in predictions *or* labels. Our main metric will be accuracy while IoU will be a complementary one.

To know how to divide our data between train set and test set, we did some small experiments trying different split sizes and, over a number of repetition, computed mean and standard deviation of accuracy with regard to this split size. The results are in Figures 3 & 4. With the different split ratios that we tested, we see even at the smallest values: 0.5, 0.5 that accuracy on validation set is already decreasing so it could and should be interesting to see with smaller values if validation accuracy and logloss are satisfying.

In the end, data will be of two kinds: one being the original processed images, and the other will be the representation of these images by a trained CNN.

3 Preliminary Results

From the paper [1] the authors first trained a base fully convolutional network (FCN) [2], from which they then derived other architectures. To provide a finer classification, they got a multi-layer perceptron (MLP) network on top of the base FCN, as explained in [3]. The MLP is simply a neural network with one hidden layer, applied to every pixel individually. They also include the performance of a skip network, which is an alternative way of combining features to refine the predictions of a coarse base FCN (see [3]). A comparison was studied to compare the performance of these three methods to this image labeling challenge.

Table 1: Numerical eval. on small validation set.

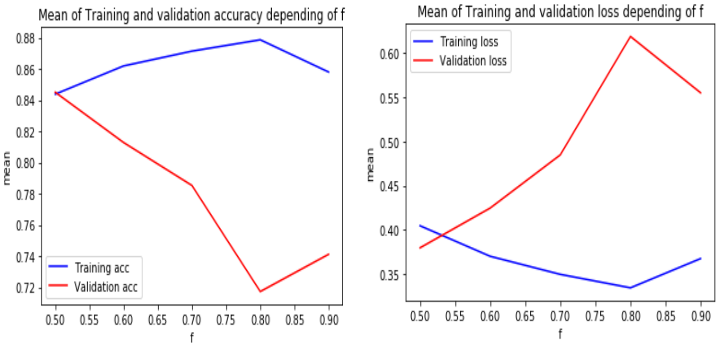


Fig. 3: Mean accuracy & log loss on train and validation

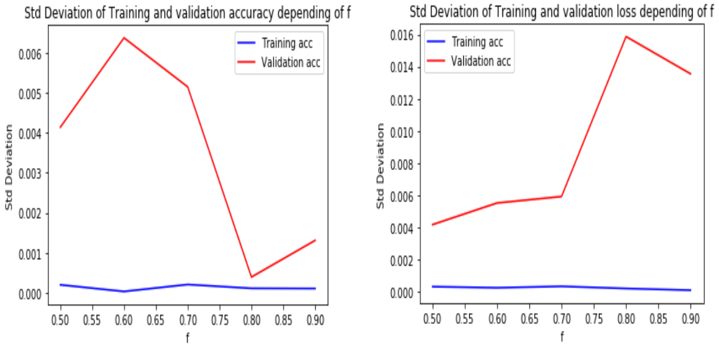


Fig. 4: Standard deviation for accuracy & logloss on train and validation

Comparison of performance of models							
		Austin	Chicago Kitsap	Co. West	Tyrol	Vienna	Overall
FCN	IoU	47.66	53.62	33.70	46.86	60.60	53.82
	Acc.	92.22	88.59	98.58	95.83	88.72	92.79
Skip	IoU	57.87	61.13	46.43	54.91	70.51	62.97
	Acc.	93.85	90.54	98.84	96.47	91.48	94.24
MLP	IoU	61.20	61.30	51.50	57.95	72.13	64.67
	Acc.	94.20	90.43	98.92	96.66	91.87	94.42

Table 2: Numerical evaluation on test set.

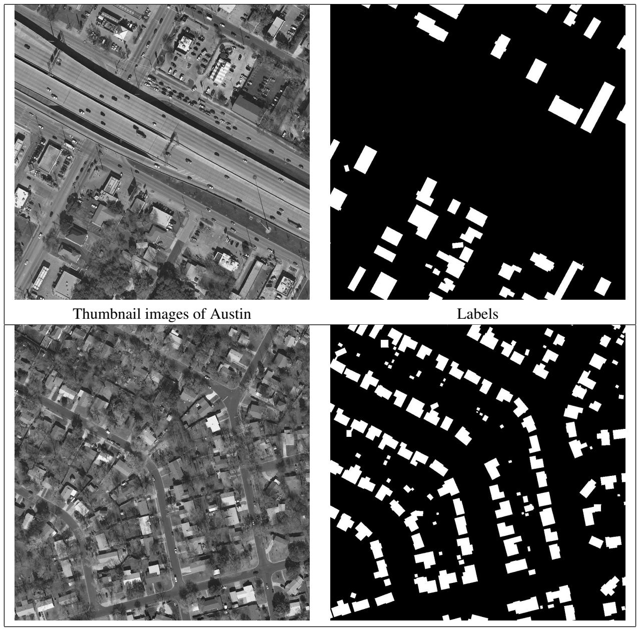
Comparison of performance of models							
		Belling.	Bloom.	Inns.	S. Fran-cisco	East Tyrol	Overall
FCN	IoU	44.83	35.38	36.50	44.92	43.69	42.19
	Acc.	94.48	94.07	92.97	82.60	95.14	91.85
Skip	IoU	52.91	46.08	58.12	57.84	59.03	55.82
	Acc.	95.14	94.95	95.16	86.05	96.40	93.54
MLP	IoU	56.11	50.40	61.03	61.38	62.51	59.31
	Acc.	95.37	95.27	95.37	87.00	96.61	93.93

The numerical results are summarized in Tables 2 and 3, for the validation and test sets, respectively. As mentioned in the paper [1], the overall system was trained for an extra 250,000 iterations, which took about 50 hours on a single GPU. Due to the fact that currently we don't have access to any GPU, we read and studied the paper and the leaderboard of this challenge. We will use those results as a goal we hope to attain, considering that we won't use the full database, and that we also modified the data we will be using.
Two other methods are proposed to deal with the problem. The first solution is logistic regression, because our goal is

to distinguish whether a pixel cooresponds to a building or not. It's a binary classification problem, so logistic regression should do the work.

The second is a bit more complicated, because it is with segmentation networks by autoencoders [4]. Nevertheless, for the sake of the simplicity of the problem, a Segnet might be too complicated for us. We will consider it again if we can try, or if other methods are not satisfactory. We tried to run our own tests to use as baselines, but we failed to achieve anything useful with the metrics we used, so we prefer keeping baselines from the literature as of now.

During our experiments, good results were often close to 85% accuracy for good parameter choices.



Thumbnail images of Austin

Labels

References

[1] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, Pierre Alliez. Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark. IEEE International Symposium on Geoscience and Remote Sensing (IGARSS), Jul 2017, Fort Worth, United States.

[2] Jonathan Long, Evan Shelhamer, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," in CVPR, 2015.

[3] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez, "High-resolution semantic labeling with convolutional neural networks," arXiv preprint arXiv:1611.01962, 2016.

[4] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 1 Dec. 2017.