ON THE USE OF DEEP NEURAL NETWORKS FOR THE DETECTION OF SMALL VEHICLES IN ORTHO-IMAGES

Jean Ogier du Terrail and Frederic Jurie

jean.ogier-duterrail@unicaen.fr
frederic.jurie@unicaen.fr

ABSTRACT

This paper addresses the question of the detection of small targets (vehicles) in ortho-images. This question differs from the general task of detecting objects in images by several aspects. Firstable, the vehicles to be detected are small, typically smaller than 20x20 pixels. Secondly, due to the multifariousness of the landscapes of the earth several pixel structures similar to that of a vehicle might emerge (roof tops, shadow patterns, rocks, buildings), whereas within the vehicle class the inter-class variability is limited as they all look alike from afar. Finally, the imbalance between the vehicles and the rest of the picture is enormous in most cases. Specifically, this paper is focused on the detection tasks introduced by the VEDAI dataset [1]. This work supports an extensive study of the problems one might face when applying deep neural networks with low resolution and scarce data and proposes some solutions. One of the contributions of this paper is a network severely outperforming the state-of-the-art while being much simpler to implement and a lot faster than competitive approaches. We also list the limitations of this approach and provide several new ideas to further improve our results.

Index Terms— detection,learning, aerial-imagery

1. INTRODUCTION AND RELATED WORK

Although automatic detection of objects in images is an old problem and benchmarks have been around for a while [2], the computer vision community only recently began to put the focus on detecting small objects. Modern benchmarks like [3] that favored small objects have proven to be surprisingly challenging and to this day no approach has clearly gained the upper-hand on the others. In spite of the progress of the technologies behind satellite optics it is likely that aerial imagery pictures will always be filled with small objects with relatively low resolution. Furthermore, as aforementioned, the vehicles to detect occupy a few hundreds pixels on often very large images (around a million pixels) so most pixels/windows on such images will have to be classified as belonging to a sink class that we will call background. Those very peculiar conditions explain why one cannot directly apply winning methods on [2] or [4] like [5] on such benchmarks.

Modern pipelines for the detection of objects in images can be roughly classified in 3 main groups. The first one contains *grid-based* regression methods. The user defines a grid on the image and regress bounding boxes on the images based on offsets w.r.t this grid [6, 7, 8]. The second one is *region proposals* based methods, with most of them consisting of a cascade starting with class agnostic region proposals [9, 10, 11] followed by a classifier [12, 13, 14]. This category also includes the recent development that began with [5] and that consists of learning the region proposal part together with the classification (see e.g. [15]). The last one, which is not so popular anymore, contains *sliding window* based methods [16, 17]. We argue that none of the methods could be applied without extensive modifications to the task of detecting small targets in large images. For the first class of methods the grid used would have to be excessively large with a close-knit network, which would require to work on pooling or upsampling the final output and that goes against the philosophy of grid-based methods, which is to be coarse to be fast. In the second class of methods, problem specific region proposal algorithms would have to be designed so that they could detect small low-resolution objects with high recall. For instance, we think that the work we propose could be used as a first step in such pipelines.

Vehicle detection on aerial imagery has been recently studied in [18, 19, 20, 21, 22]. The most recent articles on the subject are [23, 24], based on handcrafted descriptors, and [25, 26, 27] using convolutional neural networks.[25] also uses a convolutional neural network with a similar architecture, but does not provide any strategy to deal with class imbalance, it uses a different and outdated cost function and is much much slower. [26] introduces a new and large dataset and reports good performances but the context-based method it used is impractical in our case, furthermore, the metric used to measure performances is much less demanding in terms of precision of the localization.

This work concentrates on developing hard-mining strategies to deal with classes with few examples and is a lot faster than all former approaches due to the small sized network and the fully convolutional inference and is specifically tailored for the detection of small objects.

2. METHODOLOGY

2.1. Introduction

In contrast with all the previous detection results on VEDAI, we do not use any sliding a window over the image to do inference but get rid of this expensive step using fully convolutional networks, as proposed by [28]. In terms of architectures, we have experimented several variants around the simple LeNet-5 architecture ([29]) . This architecture seems indeed a good candidate for our detection task as the LeNet5 network has only 2 max-pooling layers (non-overlapping). Consequently, when used as fully convolutional network the resolution of the output heat maps is 4 times smaller than the original image. In practice there is no need to upsample the heat map by unpooling with max-pooling switches or using transposed convolutions like in ([30] or [28]), nor to use dilated convolutions as in [31] nor even to use the shift-and-stitch trick of [16]. One key difference of detection on ortho-images and detection on commonly used datasets like Ms COCO, is that, because the distance between the sensor/camera and the ground is known in advance, all targets of a same class share the same size in pixels, approximately, and this size can be estimated accurately. Therefore there is no need to adopt a multi-scale approach nor to regress the width and height of the bounding boxes of the vehicles. For this reason, we can set once and for all the size of the extracted patches each class having a different sized associated patch. By doing that we also set the size of the first fully connected layer. The imbalance between the sink class and the vehicle class being so overwhelming we had to use a multi-step approach to get rid of most of the backgrounds and simplify the classification as noticed by [27] once the vehicles and the background are identified classification between the different classes is more straightforward.

2.2. Training

We first extract positive patches around the targets. The negative patches used are sampled uniformly from the images until we reach a ratio of 5 to 1 (which has been determined by cross-validation). Too much negative samples and the imbalance is too strong, too little and the heterogeneity of the background class is not fully captured. A critical point that is also investigated in section 5 is the imbalance factor applied to the cross-entropy cost. Then the network is trained using stochastic gradient descent (SGD), with a dropout of 0.5[32] in the fully connected layers. When it reaches convergence, after around 100 epochs, then we do inference on the images on the training set in a fully convolutional fashion. From 1064x1064 pixels image (images have been up-sampled to 1024x1024 and padded) we get 256x256 heat maps. To select the maximums we do traditional Non Maximum Suppression (NMS). We then consider the remaining maximums to be the result of our detection and we evaluate them on the ground truth in this process we do not estimate orientations for bounding boxes working with squared bounding boxes is sufficient to get good results even with an orientation dependent metric.

In the process we sample negative patches that have been misclassified and weak true positives. In fact we experimented with 4 different strategies of hard-mining results are presented in section 5. We repeat the process multiple times.

2.3. Hard-Mining strategies

Bootstrapping offers a lot of liberties on how the hard examples are chosen. One could for instance pick a limited number of false positives per image or one could fix a threshold and only pick a false positive if its score is superior than a fixed threshold (0.5 for instance). There is also the question of weak true positives whether to include them at all and if yes from which threshold should we pick them. The previous experiments used the thresholds (0.5,0.5) while limiting the number of examples in each image to be 25 at max. We tested the 4 following different hard-mining variants on our network :

- Strategy 1: the number of hard examples per image is set to be exactly N (with N=25 in our experiments) whatever their scores. No weak true positive are added.

- Strategy 2: Same strategy that 1 but with weak true positive examples whose scores are less than 0.5 (scores are normalized probabilities).

- Strategy 3: We chose only hard negative with scores superior to 0.5 and weak true positives with scores less than 0.5

- Strategy 4: Same as strategy 3 without weak true positives.

These 4 strategies are evaluated in Section 3.

2.4. Addressing class imbalance issues

One of the issues with detection is the imbalance between the few present vehicles (targets) on images and the large variety of backgrounds. This issue is even more present in aerial imagery because of the relative size of full images and small vehicles like cars. In this context, VeDAI is especially challenging as each image only contains a few vehicles. There are many techniques to effectively fight this. Controlling vehicles/background proportion in each batch, minority oversampling like [33], or modifying the cost itself to give more weights to the misclassification of classes that are less present.

We chose the latter because of its simplicity. The cost function implemented is the weighted cross-entropy:

$$L(w) = \sum_{i=1}^{N} C * t_i * log(f_w(x_i)) + (1 - t_i) * log(f_w(x_i)) \quad (1)$$

where x_i is one of the N image-patches in the training set, $f_w(x)$ is the score given by the convolutional neural network to a patch x, t_i is 1 when x_i belongs to the class under test 0 otherwise, C is a scalar defined as the ratio of Negative (background) vs Positive (vehicles) examples in the batch.

2.5. Classification study: rotation invariance, *etc.*

This part aims at finding the weaknesses of our approach and further motivates a follow-up work on using this baseline network as the first part of a cascade. We created numerous classification sets using collected hard-negative samples. We looked at every possible combination of: contrast normalizing the patches (or not), angle normalizing the targets (meaning all targets have the same orientation)(or not) and shifting the patches from the targets (4 pixels apart in checkerboard distance)(or not). For simplicity we adopted the following code: S and $\not S$ mean respectively with shifted target normalization or without, A and $\not A$ means angle-normalized patches or all rotations included and C and $\not C$ contrast-normalized patches or no contrast normalization used. These different options are experimented in Section 3

3. EXPERIMENTS

The VeDAI dataset The VeDAI dataset [1] consists of 1200 images that come in two different resolutions 512x512 and 1024x1024. Every image is available in two versions either colored (RGB) or infrared. All experiments in the paper were conducted on the infrared version of the 512x512 images. The dataset is provided with 10 folds and 1340 cars in total, with an average of around 140 cars to detect per fold. The definition of what a positive detection is pretty different from the standard Intersection Over Union (IOU) criterion adopted by the Pascal VOC or MS COCO. A detection is considered correct if it lies within the ellipse centered on the ground truth and lying inside the edges of the target car (multiple hits on the same targets are counted as False Positives). We are interested in mainly two metrics namely the mAP which is measured across all folds and the recall at low FPPI (recall for a given rate of False Positive Per Image).

Results on VeDAI We first report general results on VeDAI using an architecture inspired by the LeNet-5 network. On overall, the network contains 2 convolutional layers followed by three fully connected like layers, as detailed Table 1.

Name	Type	Filter Size	Input	Output
		Stride	Size	Output
Conv1	convolution	5x5/1	45x45x3	41x41x96
Pool1	max-pooling	2x2/2	41x41x96	20x20x96
Conv2	convolution	5x5/1	20x20x96	16x16x192
Pool2	max-pooling	2x2/2	16x16x192	8x8x192
fc1	convolution	8x8/1	8x8x192	1x1x384
fc2	convolution	1x1/1	1x1x384	1x1x84
fc3	convolution	1x1/1	1x1x84	1x1x2
Softmax	Softmax	None	1x1x2	1x1x2

Table 1. Architecture of our network, inspired by LeNet-5

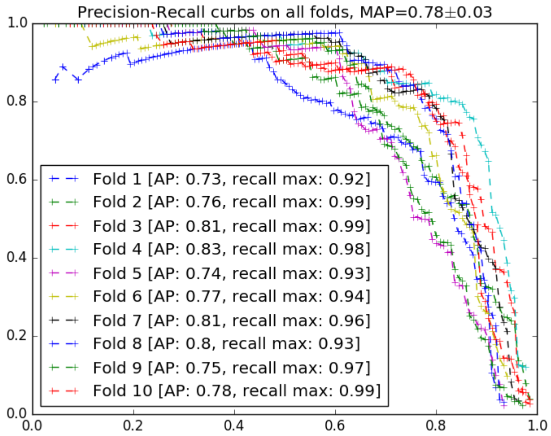

Fig. 1. Precision-Recall curves given by our RPN network on the 10 folds of the VEDAI dataset. The mAP is of 0.78± 0.03.

All convolutions are VALID type convolutions (no padding) and all fully connected layers are implemented as 1x1 convolutions. In order to make the results as good as possible, we experimented with the following parameters : (i) size of the receptive field, (ii) amount of regularization (iii) depth of each layer. We selected optimal parameters on a validation set. As there is none validation set by default in VeDAI, we split the 10 folds provided with the dataset into ten subsets of equal length and chose our parameters by training the network on 9 of them and validating on the tenth.

The selected network reaches a mAP of **77.8**±3.3 (see 1) which is a very large improvement (12%) from the previous state of the art, on the car category. As a comparison, the recent work of [23] reports a mAP of 69.6 ±3.4 for the same class.

We used an L2 regularization of 0.0001 and a wide networks with respectively for each layer 96, 192, 384 and 84 neurones. The network was trained using classic SGD with a fixed learning rate of 0.001.

The Table 2 lists all the published results on VeDAI and

method	AP	Recall@0.01FPPI
DPM [34]	60.5 ± 4.2	13.4 ± 6.8
SVM+HOG31 [34]	55.4 ± 2.6	7.8 ± 5.5
SVM+LBP [34]	51.7 ± 5.2	5.5 ± 2.2
SVM+LTP [34]	60.4 ± 4.0	9.3 ± 3.7
SVM+HOG31+LBP [34]	61.3 ± 3.9	8.3 ± 5.2
SVM Fusion AED (HOG) [35]	69.6 ± 3.4	20.4 ± 6.2
Ours	$\mathbf{77.80 \pm 3.3}$	$\mathbf{31.04 \pm 11}$

Table 2. Comparisons with related works

Strategy	R1	R2	R3	R4	R5	R6
S1	0.00	0.21	0.39	0.36	0.28	0.25
S2	0.00	0.10	0.17	0.43	0.27	0.32
S3	0.00	0.35	0.34	0.31	0.25	0.29
S4	0.00	0.23	0.32	0.23	0.32	0.32

Table 3. Recall@0.01FPPI *w.r.t.* the number of passes for the 4 strategies, on a validation set

Config.			Acc.	Acc.	Acc.	Recall
				pos.	backgds	at 0.001 FP
$\not{A}$	$\not{S}$	$\not{C}$	98.35	53.91	98.94	28.02
		C	98.94	43.21	99.68	25.59
	S	$\not{C}$	98.78	66.67	99.21	41.86
		C	99.01	66.67	99.44	50.33
A	$\not{S}$	$\not{C}$	99.01	82.72	99.23	63.45
		C	99.10	86.42	99.27	71.68
	S	$\not{C}$	99.33	91.77	99.43	82.79
		C	**99.41**	**92.59**	**99.50**	**86.58**

Table 4. Learning variant/invariant representations. See Section 3 for details.

ours. We can see that in terms of Average Precision or recall the advantages of our method. [27] do not report any detection results only classification so we cannot compare to it directly.

Evaluation of the 4 hard-negative-mining strategies As said before, bootstrapping offers a lot of liberties on how the hard examples are chosen. We proposed in Section 2 four different hard-mining strategies we are now going to evaluate. Table 3 gives the Recall at 0.01 False Positive per Image (FPPI) for different rounds of hard mining. From this table we can make three observations: first, the best overall performance is obtained with S2 and should be preferred. Second, if one wants to limit the number of rounds of hard example mining, he should prefer S3 which gives 0.35 after only 2 rounds. Third, without hard mining, the performance of the network is very low.

Compensating the translations, rotations, *etc.*. We experimented the 3 normalization strategies given Section 2.5. As said before, to make the evaluation of the performance faster, we experimented these alternatives on a classification task. For building this classification set, we used all the targets from fold 1 and added hard negative examples.

There is 50000 examples in the training set with a ratio of negative over positive of 70 it is 5 times more than in the course of hard-mining passes as we had to sample only one positive example per target present instead of 5 as we have when we start detection training. We measure the accuracy by which percentage of images is classified correctly (with a fixed threshold of 0.5) but as the number of backgrounds is 70 times the number of targets, the accuracy is actually not very informative. It is more helpful to have also the accuracy on the target class patches, which verifies that the classifier would not get away with classifying everything as background and still get a good accuracy. We added the recall for 0.001 FP meaning the recall of the target classes patch obtained for a given proportion of 0.001 false positive in the validation set for the same reason. The results are given in Table 4. It is also interesting to see what would the network that was trained on perfectly centered patches do on shifted targets (it often happens when the equivalent sliding window step is not 1). We get 16.5 accuracy on the positive examples (instead of 53.91) and a recall at 0.001 FPPI of 18.20 (instead of 28.02). There is a severe degradation of the performances. The same phenomenon is observed when we train on all orientations and test on angle normalized patches. Accuracy on positive examples drops to 52.67 (instead of 82.71). The recall at 0.001 FPPI is also impacted 55.8 (instead of 63.45). Therefore this network performs much better when the targets are centered and normalized in rotations. We ran additional experiments to see if the angle normalization step of the cascade was really necessary so we added a data-augmentation module which rotated the patches in the current batch by a random angle uniformly chosen between 0 and 360. The network that performed poorly (53.91 accuracy on positive examples and 28.02 recall at 0.001 FPPI) gave reasonable results with 88.77 accuracy on the positive examples. So if we can center and normalize detected results we could use this trained classifier on top and it would better the results.

4. CONCLUSIONS

We have presented state-of-the-art results on a challenging benchmark and insights on how to tackle detection in the difficult context of few examples learning and extreme imbalance. This study cuts across the many articles using transfer learning to avoid having to deal with training a network from scratch. We are also confident that having found in 3 that angle normalization has such a strong effect on performances we could improve a lot on these results by using a cascade.

5. REFERENCES

[1] Sebastien Razakarivony and Frederic Jurie, "Vehicle Detection in Aerial Imagery: A small target detection benchmark," *Journal of Visual Communication and Image Representation*, 2015.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[3] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," *ArXiv e-prints*, May 2014.

[4] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[5] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[6] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015.

[7] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, and S. Ioffe, "Scalable, High-Quality Object Detection," *ArXiv e-prints*, Dec. 2014.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott E. Reed, "SSD: single shot multibox detector," *CoRR*, vol. abs/1512.02325, 2015.

[9] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip H. S. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *IEEE CVPR*, 2014.

[10] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[11] Larry Zitnick and Piotr Dollar, "Edge boxes: Locating object proposals from edges," in *ECCV*. September 2014, European Conference on Computer Vision.

[12] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *CoRR*, vol. abs/1311.2524, 2013.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *CoRR*, vol. abs/1406.4729, 2014.

[14] Ross B. Girshick, "Fast R-CNN," *CoRR*, vol. abs/1504.08083, 2015.

[15] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun, "R-FCN: object detection via region-based fully convolutional networks," *CoRR*, vol. abs/1605.06409, 2016.

[16] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *CoRR*, vol. abs/1312.6229, 2013.

[17] Christian Szegedy, Alexander Toshev, and Dumitru Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems 26*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 2553–2561. Curran Associates, Inc., 2013.

[18] Tao Zhao and Ram Nevatia, "Car detection in low resolution aerial images," *Image and Vision Computing*, vol. 21, no. 8, pp. 693–703, 2003.

[19] Franz Leberl, Horst Bischof, Helmut Grabner, and Stefan Kluckner, "Recognizing cars in aerial imagery to improve orthophotos," in *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*. ACM, 2007, p. 2.

[20] Line Eikvil, Lars Aurdal, and Hans Koren, "Classification-based vehicle detection in high-resolution satellite images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 1, pp. 65–72, 2009.

[21] Aniruddha Kembhavi, David Harwood, and Larry S Davis, "Vehicle detection using partial least squares," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 6, pp. 1250–1265, 2011.

[22] Karim Ali, Francois Fleuret, David Hasler, and Pascal Fua, "A real-time deformable detector," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 225–239, 2012.

[23] Sebastien Razakarivony, *Apprentissage de variétés pour la Détection et Reconnaissance de véhicules faiblement résolus en imagerie aérienne*, Ph.D. thesis, Université de Caen Basse-Normandie, 2014.

[24] Joshua Gleason, Ara V Nefian, Xavier Bouyssounousse, Terry Fong, and George Bebis, "Vehicle detection from aerial imagery," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2065–2070.

[25] Xueyun Chen, Shiming Xiang, Cheng-Lin Liu, and Chun-Hong Pan, "Vehicle detection in satellite images by parallel deep convolutional neural networks," in *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*. IEEE, 2013, pp. 181–185.

[26] T. Nathan Mundhenk, Goran Konjevod, Wesam A. Sakla, and Kofi Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," *CoRR*, vol. abs/1609.04453, 2016.

[27] Nicolas Audebert, Bertrand Le Saux, and Sébastien Lefèvre, "On the usability of deep networks for object-based image analysis," *CoRR*, vol. abs/1609.06845, 2016.

[28] Evan Shelhamer, Jonathan Long, and Trevor Darrell, "Fully convolutional networks for semantic segmentation," *CoRR*, vol. abs/1605.06211, 2016.

[29] Y LeCun, L Bottou, Y Bengio, and P Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[30] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, "Segnetme: A deep convolutional encoder-decoder architecture for image segmentation," *CoRR*, vol. abs/1511.00561, 2015.

[31] Fisher Yu and Vladlen Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015.

[32] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *CoRR*, vol. abs/1207.0580, 2012.

[33] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer, "Smote: synthetic minority oversampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[34] Sébastien Razakarivony and Frédéric Jurie, "Vehicle Detection in Aerial Imagery: A small target detection benchmark," *Journal of Visual Communication and Image Representation*, Dec. 2015.

[35] Sebastien Razakarivony, *Apprentissage de varietes pour la Detection et Reconnaissance de vehicules faiblement resolus en imagerie aerienne*, Theses, Université de Caen Basse-Normandie, Dec. 2014.