Convolutional Neural Network for Traffic Signal Inference based on GPS Traces

Y. Méneroux[1], V. Dizier[1], M. Margollé[1], M.D. Van Damme[1], H. Kanasugi[2], A. Le Guilcher[1], G. Saint Pierre[3], and Y. Kato[4]

1 **Univ. Paris-Est, LASTIG COGIT, IGN, ENSG, Saint-Mandé, France**
2 **CSIS, Institute of Industrial Sciences, The University of Tokyo, Japan**
3 **Cerema, Toulouse, France**
4 **Transport Consulting Division, NAVITIME JAPAN Co., Ltd**

Abstract

Map inference techniques aim at using GPS trajectories collected from a fleet of vehicles, to infer geographic information and enrich road map databases. In this paper, we investigate whether a Convolutional Neural Network can detect traffic signals on a raster map of features computed from a large dataset of GPS traces. Experimentation revealed that our model is able to capture traffic signal pattern signature on this very specific case of unnatural input images. Performance indices are encouraging but need to be improved through a more refined tuning of the workflow.

Keywords: Map Inference, Machine Learning, GPS Traces, Traffic Signal, Deep Learning

1 Introduction

Along with the proliferation of electronic devices equipped with a Global Positioning System (GPS) receiver large datasets of daily trajectories are collected by specialized fleets of vehicles (such as taxis, bus, delivery trucks...) or more recently, through collaborative driving mobile applications. When available, they constitute a valuable source of information to infer or enrich road map databases. With traditional cartography techniques, roads are manually digitized on orthorectified aerial images. Despite the fact that automatic or semi-automatic detection from images is getting increasingly efficient [8], road maps update speed is still limited by the (generally consequent) period separating two successive image acquisitions. Indeed aerial image campaigns are typically conducted every several years, which is to be put in perspective with the fast evolution of cities [2]. Furthermore, the process often needs to be completed with field surveys to acquire features that cannot be captured in the images.

Conversely, by the means of GPS traces coupled with efficient mining algorithms, road maps can be inexpensively generated from scratch and updated on a daily basis. This observation gave birth to a new field of research, known as *map inference* which aims at completing or replacing traditional survey techniques with floating car data [1]. Though it was initially restricted to the geometry of roads, refining topology, inferring attributes and now detecting road infrastructure (traffic signs, traffic calming devices...) are becoming achievable [5, 7]. Detailed road maps are of utmost importance for traffic flow simulation and subsequent urban planing, for autonomous vehicles or merely for computing accurate routing time estimations. On the other hand, supervised statistical learning techniques [4] have been successful in a wide range of applications, from signal and image classification and segmentation to medical diagnosis. In our application, they guarantee the genericity of the approach (provided that labeled instances are available, the process may be customized to infer other types of information, in different environments without having to design a new algorithm *ex nihilo*). Considering the large number of GPS traces that are typically recorded every day, this raises two questions: firstly how to design accurate algorithms to extract the relevant information among such massive amount of data, and secondly how to implement them efficiently.

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In this paper, we propose to train a convolutional neural network (CNN) to detect stop lines associated to traffic signals on a rasterized map of trajectories. The remaining of the paper is organized as follows: the workflow of our application is presented in the next section, then section 3 provides implementation details and preliminary results obtained on a real-world dataset. Eventually, section 4 concludes the paper.

2 Workflow

The key idea of our approach is to reduce the full dataset of GPS traces into a regular grid of aggregated features, which is much more tractable from the computational view point (the amount of information to process grows only with the spatial extent of the concerned area and no longer with the GPS dataset size). The overall workflow is depicted on figure 1 hereafter.

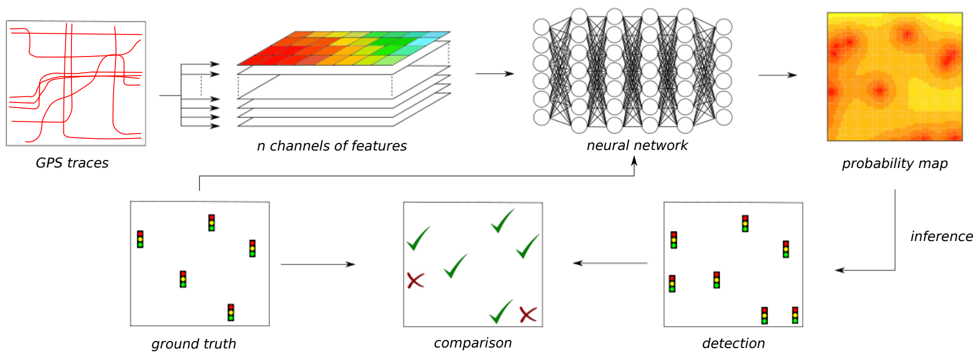

Figure 1 Overall workflow

The first step allocates segments of GPS trajectories on a regular grid. In each cell, n aggregated features (such as mean and standard deviation of speeds, number of traces...) are computed to produce a set of raster maps, which may be seen as a single image with n channels. This operation may be performed efficiently through parallelization techniques and thus is scalable with the daily number of GPS traces recorded. The resulting image is input into a CNN, trained with labeled instances to detect traffic signals. CNN is an extension of artificial neural networks, where each layer of neurons performs a series of convolutions on the output of the previous layer, decreasing the number of parameters in the model while keeping the ability to extract very expressive features from spatially structured data [3, 4]. This makes CNN perfectly adapted to image-based learning. In this work, we investigate CNN capabilities in the context of recognition on images rasterized from GPS traces. The output of the network is a probability map, *i.e.* pixel values are proportional to the traffic signal presence probability. Note that here we compute a per-pixel estimation often referred to as *dense learning* in the literature. Local maxima of the output map are eventually extracted, and compared to the ground truth for validation.

3 Implementation and preliminary results

The experimental data represent a set of 11862 GPS traces, located in Mitaka city (16 km^2), suburbs of Tokyo (Japan). They were provided by NAVITIME JAPAN, a private company developing navigation technologies and providing various kinds of web application services such as route navigation, travel guidance, and other useful information services for moving people. The municipality has 669 traffic signals, reported by photo interpretation.

Traces containing fewer than 10 points (approximately 1% of all traces), were rejected. Subsequently, for each trace, a set of indices were computed from individual points: speed, acceleration, bearing, whether the point is a stop (*i.e.* if its acceleration is positive and the preceding GPS record has a zero speed or is decelerating). The bearing is used to distinguish the flow direction on two-way streets, and takes discrete values in the 8 directional sectors. A 5 meter-resolution base grid is made. The size of 5 m can reliably locate traffic signals. In each grid cell, computed indices on points are aggregated: mean and standard deviation of speeds, means of accelerations, number of stop points and most frequent orientation. These 5 features will provide 5 raster images. Feature values are converted on a 8-bit grayscale. To differentiate the value 0 from the absence of data, the minimum threshold taken by the values is 16 (*e.g.* the mean of speeds is distributed between 16 and 255). The ground truth is a binary image: cells containing a traffic signal take the value 255 (white) and remaining cells are set to 0 (black).

Feature maps and ground truth images are cut by a sliding-window into small 60×60-pixel output images, (corresponding to 300 × 300 m on the field). The overlap between images is equal to 20 pixels, hence the traffic signals have different positions in the image, so the spatial context changes. This artificial increase in the number of training instances may help the model to deal with a large variety of traffic signal configurations.

We validated the model with cross-validation on 8 geographical zones, each containing approximately 90 traffic signals and 200 windows. Areas have distinguishing morphologies like downtown, residential district, motorway environment, etc. (e.g. the zone 5 contains many unsignalized intersections).

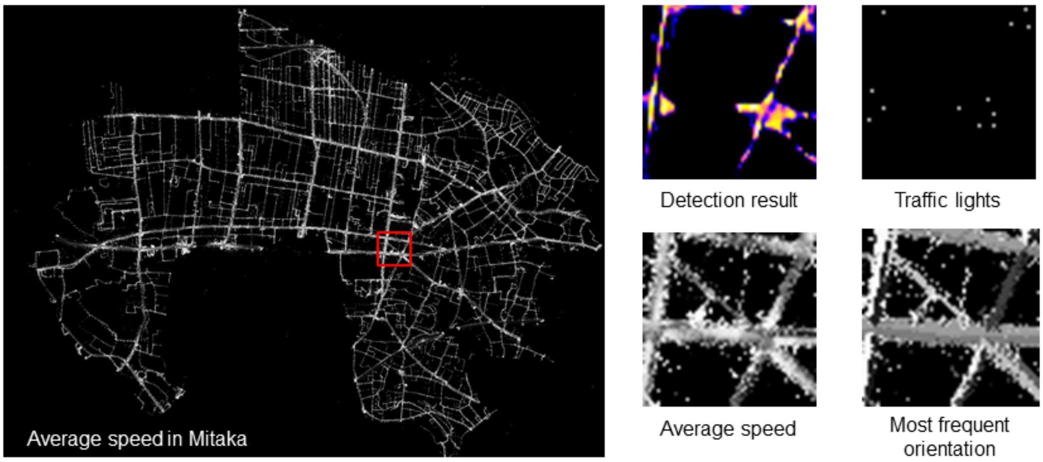

Figure 2 Example of 2 features and the result for one image

Our neural network architecture is inspired from U-Net [6]. Since traffic signals are overwhelmingly under-represented in comparison with background, we used, as proposed in the U-NET framework, a weighted loss function, in order to penalize false negatives. The weight, corresponding to a rough average estimate of 3 traffic signals per image, was set to 1:1200 (3/60×60). It is important to note that, to produce a probability map with the same resolution as the input image, in the architecture pooling operators were replaced by upsampling operators. The network model has been implemented using Keras[1], a Python library that overloads TensorFlow[1].

Since a numerical validation of the model would require to have an efficient method to extract

[1] Keras: `https://keras.io/` TensorFlow: `https://www.tensorflow.org/`

traffic signals from probability map (which has to be addressed in future research), numerical validation is not possible yet. However a surrogate manual validation provided the following results: the recall rate stands at 75% with a precision of 60% (65% F-measure). The experiments were performed using a personal computer with 3.30 GHz CPU, 4-core processor and 8 GB of memory. The training step took 70 minutes for 54900 images for a batch size of 100 instances and 130 epochs. The validation step took 1.136 seconds for testing 60 images. This benchmark suggests that the processing time required to cover the entire Tokyo area (2190 km^2) is below 7 minutes, which highlights the scalability of our approach.

Though not satisfying yet for a fully automatic detection, these preliminary results are encouraging. We believe that the GPS dataset is not sufficiently large to ensure the convergence of the results. Particularly, a significant number of intersections are only covered by a few traces. Besides, with a more extensive dataset, we may calibrate the model more thoroughly without falling into the trap of overfitting. Results may be improved also by adding new features (congestion, road type, etc.). It would be interesting as well to input the aerial image in addition into the network to investigate whether this can further help the detection and localization process.

4 Conclusion

We presented a method based on CNN to detect traffic signals on a map of aggregated features computed from a dataset of GPS traces. Preliminary results demonstrated the potential and the scalability of the approach. The extraction of most likely locations from the probability map is an important aspect that needs to be addressed in future works. Let us note that the validation is as well problematic since it requires to measure the tradeoff between detection performance (recall and precision) and geometric accuracy. An important perspective of improvement lies in extending this approach to detect other types of infrastructure elements (such as yield signs, pedestrian crossings, speed bumps...) including more ephemeral events (roadworks zones...).

Acknowledgements. We would like to thank Professor Xuan Song, for his precious advice.

References

1 James Biagioni and Jakob Eriksson. Inferring road maps from global positioning system traces: Survey and comparative evaluation. *Transportation Research Record: Journal of the Transportation Research Board*, (2291):61–71, 2012.
2 Yihua Chen and John Krumm. Probabilistic modeling of traffic lanes from gps traces. November 2010.
3 Zhiling Guo, Qi Chen, Guangming Wu, Yongwei Xu, Ryosuke Shibasaki, and Xiaowei Shao. Village building identification based on ensemble convolutional neural networks. *Sensors*, 17(11):2487, 2017.
4 Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
5 Seth Rogers, Pat Langley, and Christopher Wilson. Mining gps data to augment road models. In *KDD*, 1999.
6 O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015.
7 KBA Van Winden. Automatically deriving and updating attribute road data from movement trajectories. 2014.
8 Weixing Wang, Nan Yang, Yi Zhang, Fengping Wang, Ting Cao, and Patrik Eklund. A review of road extraction from remote sensing images. *Journal of Traffic and Transportation Engineering (English Edition)*, 3(3):271 – 282, 2016.