Bayesian Deep Learning with Monte Carlo Dropout for Qualification of Semantic Segmentation

Clément Dechesne, Pierre Lassalle, Sébastien Lefèvre

BAYESIAN DEEP LEARNING WITH MONTE CARLO DROPOUT FOR QUALIFICATION OF SEMANTIC SEGMENTATION

Clément Dechesne, Pierre Lassalle

CNES
18 avenue Edouard Belin
31401 Toulouse Cedex 9, France

Sébastien Lefèvre

Univ. Bretagne Sud / IRISA
Campus de Tohannic
56000 Vannes, France

ABSTRACT

Despite the intense development of deep neural networks for computer vision, and especially semantic segmentation, their application to Earth Observation data remains usually below accuracy requirements brought by real-life scenarios. Even if well-known deep learning methods produce excellent results, they tend to be over-confident and cannot assess how relevant their predictions are. In this work, a Bayesian deep learning method, based on Monte Carlo Dropout, is proposed to tackle semantic segmentation of aerial and satellite images. Bayesian deep learning can provide both a semantic segmentation and uncertainty maps. Based on the popular U-Net architecture, our model achieves semantic segmentation with high accuracy, e.g. F1-score and overall accuracy respectively reaching 90.84% and 93.22% on a public standard dataset. Uncertainty maps, also derived from our model, show a strong interest in qualitative evaluation of the segmentation and in the improvement of the database.

Index Terms— Deep learning, Semantic segmentation, Bayesian network, Optical imagery, Uncertainty estimation

1. INTRODUCTION

Deep learning methods have been widely used for semantic segmentation of optical images. Among the earliest works, [1] and [2] used several Fully Convolutional Neural Networks (FCN) for semantic segmentation on aerial orthophotos with three spectral bands (red, green, near-infrared), plus a digital surface model (DSM) of the same resolution. They both report excellent results for a 5-class classification task (*roads, buildings, low vegetation, tree, car*) with an overall accuracy greater than 88%, and also with an efficient detection of small objects (such as individual cars). In [3], in addition to the FCNN, a boundary detection CNN module is added, increasing the accuracy of the model. [4] used a refinement module in their FCNN trained on multispectral images for a 18-class classification task, achieving excellent results (overall accuracy greater than 93% and average accuracy of 59.8%). They also showed that data augmentation was meaningful for semantic segmentation.

Despite deep neural network architectures achieve state-of-the-art results in almost all classification tasks, they still make over-confident decisions. Indeed, on the one hand, it is easy to produce images (not recognizable to humans) that existing networks believe to be recognizable with high confidence [5]. On the other hand, a small change in the input image can lead to a very different prediction, still with a high confidence [6]. No measure of uncertainty of the prediction is provided from the current network architectures. Some works have been proposed for generating relevant probability estimates from a deep neural network [7] as a measure of model confidence. However, these metrics are based on softmax probabilities which cannot fully capture uncertainty.

Bayesian deep learning has been proposed for semantic segmentation to provide some measure of uncertainty in the prediction. It can be seen as an ensemble or forest of deep neural networks, each providing a single prediction. [8] showed that dropout (initially designed to avoid overfitting) can be used as a Bayesian approximation. [9] applied this method, called Monte Carlo Dropout (MCD), for the semantic segmentation of the Cityscape dataset. They designed a DeepLab model with MCD and achieved great results with an overall accuracy of 95.3% and Intersection over Union (IoU) of 78%. They also provided, along with the semantic segmentation output, several uncertainty maps (namely *predictive entropy* and *mutual information*), showing how the model was pretty uncertain of its prediction on pixels where the prediction was erroneous. [10] also applied MCD to a SegNet architecture. The model was trained on CamVid Road Scenes and SUN RGB-D Indoor Scene Understanding datasets. It achieved better results than state-of-the-art methods and also provided uncertainty maps (for all classes and per class, based on output variability). [11] compared MCD to another Bayesian deep model, where weights were sampled from a distribution. In this case, the model learns the parameters of the distribution instead of the weights. They showed that such models produce better results and more interpretable uncertainty maps. However, some specific training strategies were needed.

To our knowledge, Bayesian deep learning has never been

applied to remote sensing images yet. In this paper, we apply it using Monte Carlo Dropout on aerial images. In addition to semantic segmentation, we also provide confidence maps, indicating how confident the network is on its prediction. Qualification maps, that combine both segmentation accuracy and uncertainty are also derived.

Our paper is structured as follows: we first describe our method in Sec. 2. We then present the dataset and our results in Sec. 3. We finally draw some conclusions in Sec. 4.

2. METHOD

We briefly recall here the principles of Bayesian learning, highlight its relevance w.r.t traditional deep learning and explain how it is applied to neural networks. The proposed network architecture, inspired from U-NET [12] (see Figure 1a) but including Bayesian layers, will then be introduced.

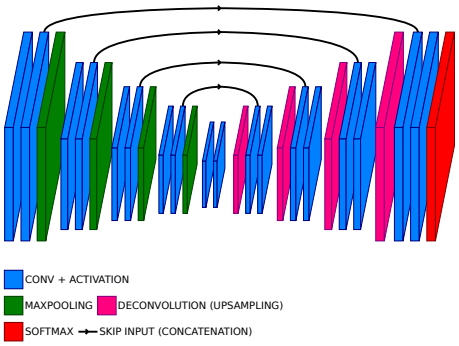
Bayesian learning for CNN has been recently proposed [13] and is based on *Bayes by Backprop* [14]. It produces results similar to traditional deep learning methods. However, the weights of the network are no longer simple points but are sampled according to a distribution whose parameters are learned. Therefore, each prediction is different from an other. With a large number of predictions, the average behavior produces relevant results, while the variability of the predictions allows us to assess the confidence of the model.

A simple way of implementing Bayesian Deep Learning is using Monte Carlo Dropout (MCD). [8] demonstrated that MCD is equivalent to traditional Bayesian Deep Learning. A layer with weights $\mathbf{M}_i$ followed by a dropout layer active in both training and prediction is equivalent to a Bayesian layer with weight $\mathbf{W}_i$ defined as:

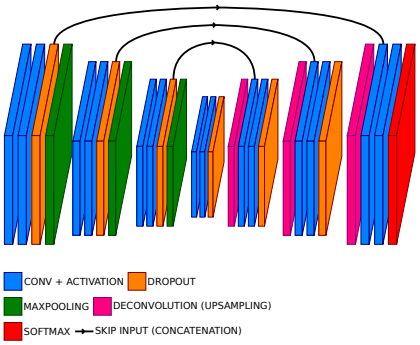
$$\begin{aligned} \mathbf{W}_i &= \mathbf{M}_i \cdot diag(z_i) \\ \text{with } z_i &\sim Bernoulli(p_i) \end{aligned} \tag{1}$$

with z_i the random (in)activation coefficients and $\mathbf{M}_i$ the weights matrix before dropout is applied. p_i is the activation probability for layer i and can be learned or set manually.

The model is composed of several convolution blocks (made of convolution layers with ELU activation), followed by a pooling layer (when downsampling) or a deconvolution layer (when upsampling). After a pooling, the number of filters of the convolution layers (resp. deconvolution) is multiplied (resp. divided) by 2. Each upsampled output is concatenated with the output of the convolution block of the same size. A model has therefore three parameters; the block size (i.e. the number of convolutions in a block), the number of poolings and the number of filters in the convolution layers of the first convolution block. In order to produce uncertainty maps, we exploit the MCD strategy. This is done by adding a dropout layer at the end of a convolution block. The dropout is active in both training and prediction. The last convolution layer of the last convolution block has a number of filters corresponding to the number of classes and a softmax activation. The architecture of the proposed model is presented in Figure 1b.



(a) Traditional U-NET.



(b) Architecture of a MCD model with a block size of 2 and 3 poolings.

Fig. 1: Traditional U-NET (a) and our derived MCD model (b).

3. EXPERIMENTS

We report here some preliminary experiments conducted on the ISPRS Vaihingen dataset. It is composed of 38 images at a spatial resolution of 9cm. The three bands of the images correspond to the near infrared, red and green bands delivered by the camera. 80% of the images are used for training/validation, the 20% remaining are kept for testing. Patches of size 128×128 were extracted in order to train the network. Data augmentation was applied by randomly flipping extracted patches.

The network was trained using the Adam optimizer [15] with a batch size of 64 and an initial learning rate of 0.001. The learning rate is reduced on plateau (learning rate divided by 10 if no decay in the validation loss is observed in the 10 last epochs) and we also perform early stopping (stop the training if no decay in the validation loss is observed in the 20 last epochs). These are standard parameters, allowing us to achieve the best results while avoiding over-fitting.

A trained Bayesian model produces different predictions for the same input data since its weights are sampled from a distribution. Therefore, several predictions need to be performed. For each iteration, the model will produce a pixel-wise probability. The final semantic segmentation is obtained through a majority vote from all these predictions. From this semantic segmentation, one can derive confusion matrices and several metrics, e.g. precision, recall, accuracy, F1-score and kappa coefficient (κ).

Since this segmentation is not sufficient to assess the reliability of the model, other metrics able to evaluate the uncertainty of the network were also computed. We investigate here two types of uncertainty measures among those reviewed in [13]. The Epistemic uncertainty (or model uncertainty) represents what the model does not know due to insufficient training data. The Aleatoric uncertainty is related to the measurement noise of the sensor. Combined, these two uncertainties form the predictive uncertainty of the network. In this work, two metrics were derived, namely the entropy of the predictive distribution (a.k.a. predictive entropy) and the mutual information between the predictive distribution and the posterior over network weights [9]. These metrics are very interesting since mutual information mostly captures epistemic (or model uncertainty) whereas predictive entropy captures predictive uncertainty which combines both epistemic and aleatoric uncertainties. The Predictive entropy is computed as follow:

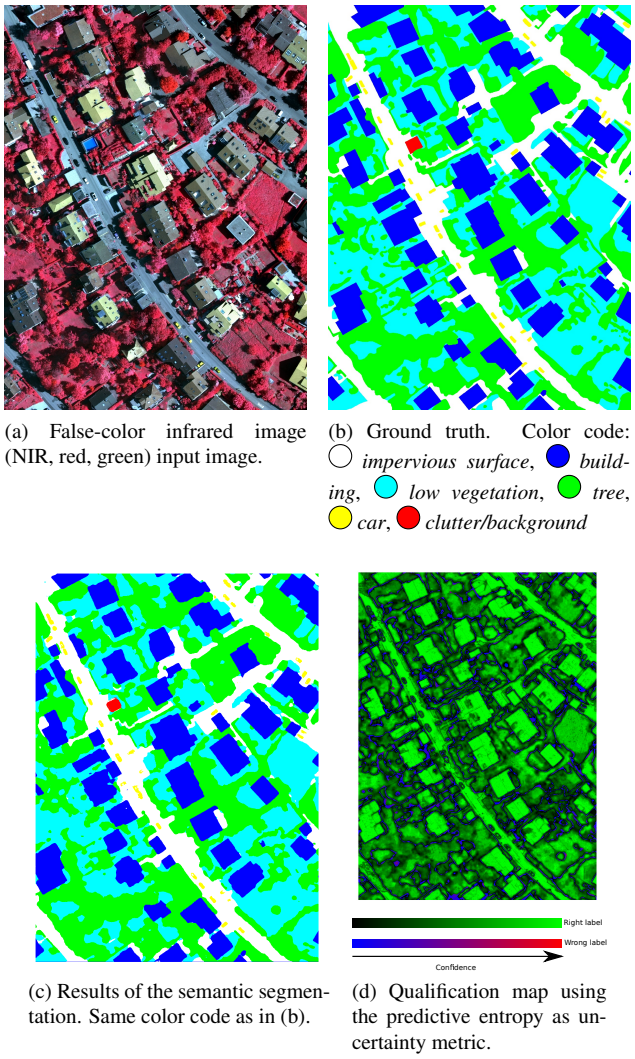
$$\hat{\mathbb{H}} = -\sum_{c}\left(\frac{1}{T}\sum_{t} p_{c,\hat{w}_t}(y|x)\right)\log\left(\frac{1}{T}\sum_{t} p_{c,\hat{w}_t}(y|x)\right) \tag{2}$$

where c ranges over all the classes, T is the number of Monte Carlo samples, $p_{c,\hat{w}_t}(y|x)$ is the softmax probability of input x being in class c , and $\hat{w}_t$ are the model parameters on the t^{th} Monte Carlo sample. The mutual information is computed as follow:

$$\hat{\mathbb{I}} = \hat{\mathbb{H}} + \frac{1}{T}\sum_{c,t} p_{c,\hat{w}_t}(y|x)\log(p_{c,\hat{w}_t}(y|x)) \tag{3}$$

In order to evaluate more precisely the impact of uncertainty metrics, qualification maps were computed. A qualification map combines the validity of the majority vote (if the network predicted the right or the wrong label) and the uncertainty of the majority vote (how confident is the network in its prediction). For the sake of visual understanding, we compute two different color gradients describing the uncertainty of the prediction, for the well-labelled and wrongly-labelled pixels respectively.

The results obtained on the ISPRS Vaihingen dataset are presented in Figure 2 and Table 1. We achieved very high scores, with an overall accuracy of 93.22% and a F1-score of 90.84%. It is slightly better than other results reported on this dataset (with overall accuracy usually ranging from 80% to 91%). Figure 2d shows that wrongly predicted pixel are indeed predicted with a low confidence (blue pixels). It is also interesting to note that similar classes (e.g. *low vegetation* and *tree*) are well predicted but again with a low confidence. This shows that our network, and more generally Bayesian deep learning, is relevant to provide both a high-quality semantic segmentation but also some associated uncertainty metrics.



(a) False-color infrared image (NIR, red, green) input image.

(b) Ground truth. Color code: *impervious surface*, *building*, *low vegetation*, *tree*, *car*, *clutter/background*

(c) Results of the semantic segmentation. Same color code as in (b).

(d) Qualification map using the predictive entropy as uncertainty metric.

Fig. 2: Results on the ISPRS dataset using the MCD model.

4. CONCLUSION

In this work, we consider semantic segmentation within a Bayesian deep learning context in order to both improve segmentation accuracy and provide some measures of prediction uncertainty. Our model includes the Monte Carlo Dropout method into the popular U-Net architecture, leading to an original Bayesian model. The proposed network performs well on the well-established ISPRS Vaihingen dataset, with results comparable or better to existing methods (overall ac-

Method	Measure	Overall	Class					
			impervious surface	building	low vegetation	tree	car	clutter/background
Proposed (Bayesian U-Net)	F1-Score	90.84	97.15	92.85	90.53	87.49	83.12	93.90
	Accuracy	93.22	98.78	95.00	94.84	99.96	99.76	98.10
	Precision	88.79	97.36	94.04	89.24	78.07	80.45	93.58
	Recall	93.27	96.94	91.70	91.86	99.49	85.98	94.22
	$\kappa \times 100$	93.06	96.38	89.01	86.99	87.47	83.00	92.72
Baseline (U-Net)	F1-Score	89.85	96.79	91.94	89.02	86.38	82.24	92.75
	Accuracy	92.23	98.63	94.37	94.02	99.95	99.75	97.74

Table 1: Comparative evaluation of our Bayesian U-Net with MCD and a standard U-Net architecture considered as baseline.

curacy of 93.22% and F1-score of 90.84%). More importantly, our Bayesian deep network is able to extract uncertainty maps that are very useful for assessing the output segmentation. Qualification of land cover maps is a strong requirement for delivering AI-driven EO products. Furthermore, one can analyse such maps, together with initial ground truth, to spot areas where the ground truth might be erroneous, before conducting some automatic or manual correction. In this context, the uncertainty maps can be exploited to generate reference data with higher accuracy.

We now plan to evaluate how Bayesian deep learning can help to improve the quality of reference data. First, the areas where the network predicted the wrong label with high confidence need to be re-inspected and corrected if needed. It will also require us to assess whether the network had appropriate reasons to be confident or not. Then we can re-trained the network using the updated ground truth, and experimentally assess the possible gain in prediction quality. We would like also to investigate Bayesian neural network considering another variational inference. This would allow us to use different distributions for the network weights (such as a normal distribution) since it tends to produce more significant uncertainty maps [11]. The main challenge here is the increase in number of parameters, leading to training issues that need to be addressed. Finally, as every ensemble method, one notable issue is also the inference time, since multiple predictions need to be performed.

5. REFERENCES

[1] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla, "Semantic segmentation of aerial images with an ensemble of cnss," *ISPRS Annals*, vol. 3, pp. 473–480, 2016.

[2] N. Audebert, B. Le Saux, and S. Lefèvre, "Semantic segmentation of earth observation data using multimodal and multi-scale deep networks," in *ACCV*. Springer, 2016, pp. 180–196.

[3] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: Improving semantic image segmentation with boundary detection," *P&RS*, vol. 135, pp. 158–172, 2018.

[4] R. Kemker, C. Salvaggio, and C. Kanan, "Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning," *P&RS*, vol. 145, pp. 60–77, 2018.

[5] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *CVPR*, 2015, pp. 427–436.

[6] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[7] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *arXiv preprint arXiv:1706.04599*, 2017.

[8] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*, 2016, pp. 1050–1059.

[9] J. Mukhoti and Y. Gal, "Evaluating bayesian deep learning methods for semantic segmentation," *arXiv preprint arXiv:1811.12709*, 2018.

[10] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," *arXiv preprint arXiv:1511.02680*, 2015.

[11] T. M. LaBonte, C. Martinez, and S. A. Roberts, "We know where we don't know: 3d bayesian cnns for credible geometric uncertainty," Tech. Rep., Sandia National Lab, Albuquerque, NM, USA, 2020.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*. Springer, 2015, pp. 234–241.

[13] K. Shridhar, F. Laumann, and M. Liwicki, "A comprehensive guide to bayesian convolutional neural network with variational inference," *arXiv preprint arXiv:1901.02731*, 2019.

[14] A. Graves, "Practical variational inference for neural networks," in *NIPS*, 2011, pp. 2348–2356.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.