ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume IV-3/W2-2020, 2020
2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS 2020), 22–26 March 2020, Santiago, Chile

AUGMENTING DATA USING GAUSSIAN MIXTURE EMBEDDING FOR IMPROVING LAND COVER SEGMENTATION

Dario A. B. Oliveira

IBM Research Rua Tutoia, 1157, Sao Paulo, Brasil, 04007-900 - dariobo@br.ibm.com

KEY WORDS: Land Cover Segmentation, Image Synthesis, Latent Data Representation, Gaussian Mixture Models.

ABSTRACT:

The use of convolutional neural networks improved greatly data synthesis in the last years and have been widely used for data augmentation in scenarios where very imbalanced data is observed, such as land cover segmentation. Balancing the proportion of classes for training segmentation models can be very challenging considering that samples where all classes are reasonably represented might constitute a small portion of a training set, and techniques for augmenting this small amount of data such as rotation, scaling and translation might be not sufficient for efficient training. In this context, this paper proposes a methodology to perform data augmentation from few samples to improve the performance of CNN-based land cover semantic segmentation. First, we estimate the latent data representation of selected training samples by means of a mixture of Gaussians, using an encoder-decoder CNN. Then, we change the latent embedding used to generate the mixture parameters, at random and in training time, to generate new mixture models slightly different from the original. Finally, we compute the displacement maps between the original and the modified mixture models, and use them to elastically deform the original images, creating new realistic samples out of the original ones. Our disentangled approach allows the spatial modification of displacement maps to preserve objects where deformation is undesired, like buildings and cars, where geometry is highly discriminant. With this simple pipeline, we managed to augment samples in training time, and improve the overall performance of two basal semantic segmentation CNN architectures for land cover semantic segmentation.

1. INTRODUCTION

Land cover segmentation is a very common application of remote sensing and is of great interest in many fields, such as agriculture and urban planning (Bokusheva et al., 2016). Many different land cover segmentation methods have been proposed in the literature, mostly based on object-based image analysis (Blaschke et al., 2014), and more recently on convolutional neural networks (Zhu et al., 2017). In (Papadomanolaki et al., 2016), the authors compared different well-established deep architectures (AlexNet, AlexNet- small, VGG) for the classification of NAIP SAT-4/SAT-6 dataset using CNNs. In (Audebert et al., 2017), the fully convolutional neural network SegNet architecture was modified to achieve semantic segmentation of multimodal airborne imagery.

While semantic segmentation is a widely discussed topic in the community (Yu et al., 2018, Buda et al., 2017), methods that show impressive results on this task still struggle with very imbalanced databases (Garcia-Garcia et al., 2017). In remote sensing, this tough scenario has been also reported (Zhu et al., 2017), and recent works deal with it using tools such as loss function weighting (Zhu et al., 2017), and augmenting classes with fewer samples using translations, rotations and scaling (Zhu et al., 2017). Even if this procedures are able to improve the CNNs segmentation performance, their impact depends very much on the number of original samples available to balance the database, since these transformations derive a limited number of new samples.

Different models, such as Generative Adversarial Networks, have been applied for balancing imbalanced training sets (Creswell et al., 2018, Karras et al., 2018, Park et al., 2019, Guo et al., 2019), but since they model the training samples distribution, the data created using them do not often add discriminating power to classification models. Other methods, such as Variational Auto-Enconders (Kingma, Welling, 2013), were also used to create latent data representations of a given set of samples, but the modification of such encoded data to derive new discriminant data is not straightforward.

This paper presents a method for data augmentation in the context of semantic segmentation of imbalanced data. First, we use an encoder-decoder CNN that takes as input an original image, encode it into a latent embedding and decode the embedding into a Gaussian Mixture Model (GMM) that best represents the input image. Second, we modify the latent embedding of a given sample, at random and in training time, to generate a new, slightly different GMM for the given input. Finally, we compute the displacement map of original and modified GMMs using an elastic registration method, and use it to warp the original image, generating new samples. We also modify the displacement map to strategically preserve the geometry of some highly geometry dependent classes, such as buildings and cars. With this pipeline, we propose a method to generate new realistic samples out of a small set of training samples, in training time. Our results report an increase in the overall accuracy of land cover segmentation for two widely used semantic segmentation CNN architectures.

This paper is organized as follows. In next section we detail our methodology, data used, pre-processing, methods for augmenting data and segmentation networks used for evaluation. Then, we present our experimental design and discuss the results achieved. Finally, we draw conclusions from this work and report future research possibilities.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-3-W2-2020-71-2020 | © Authors 2020. CC BY 4.0 License.
Primary publication at IEEE Xplore: https://doi.org/10.1109/LAGIRS48042.2020.9165670

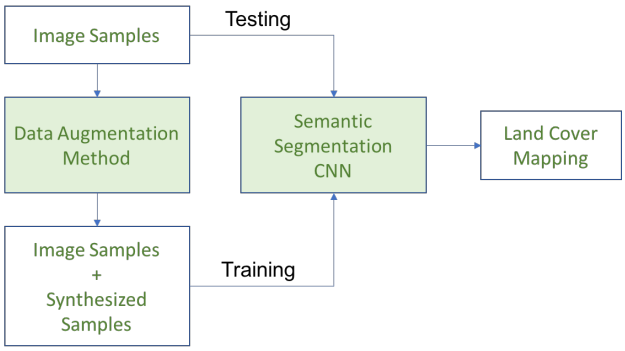

Figure 1. *Semantic segmentation schema: with a given samples database, we propose a method to augment the training set, train the semantic segmentation CNN and evaluate its performance using the testing set.*

2. METHODOLOGY

The data augmentation methodology herein proposed is tested in the context of land cover semantic segmentation, as depicted in Figure 1. Our experimental pipeline consists of organizing a training, validation and testing database; augmenting training data towards a more balanced corpus using our proposed method; and training and evaluating two different semantic segmentation networks using the original imbalanced test set. Each of these steps are presented in the following.

2.1 Data and Pre-Processing

The data used in our experiments was made publicly available by the International Society for Photogrammetry and Remote Sensing in the context of 2D semantic labelling contest proposed by the Commission II / Working Group II/4. It consists of very high resolution true ortho-photo (TOP) tiles and the corresponding digital surface models (DSM) derived from dense image matching techniques in the region of Vaihingen, Germany.

Each image has been classified manually into six land cover classes: 1) impervious surfaces, 2) buildings, 3) low vegetation, 4) trees, 5) cars, and 6) clutter/background. Annotated data was provided for approximately half of the images in the original database, which we used for training, validation and testing.

The data was pre-processed to highlight characteristics we believe to be important for discriminating between the suggested classes. We derived four channels from the given CIR (Near infra-red, R and G) bands and DSM images provided:

1. The normalized digital surface model (nDSM) was generated using the digital terrain model (DTM) filter from the SAGA GIS software. This filter basically smooths the DSM to find the DTM and subtract it from the DSM, deriving the off-terrain height of structures.

2. The NDVI (Gandhi et al., 2015) vegetation index imaging using the near infrared (NIR) and red band.

3. The Red band.

4. The Green band.

Each channel was normalized by a simple procedure: we computed the histogram of pixel intensities for a given channel considering all images, determined its 2% and 98% percentiles and transformed linearly the intensities values targeting the interval [0,1]. Values below 2% percentile were set to zero and above 98% percentile were set to 1.

Our experimental database, divided in tiles as detailed in the experimental design section, was composed by 10 four-channel normalized images for training, 3 for validation, and 3 for testing, with varied sizes.

2.2 Data Augmentation Method

The proposed data augmentation method is represented in Figure 2. First we train an encoder-decoder CNN that encodes input images into a latent representation, and then decodes the latent data into a mixture of Gaussians that reconstructs the input. Then, we change the latent embedding used to generate the mixture parameters and decode it into new mixture models slightly different from the original. Finally, we compute the displacement maps between the original and the modified mixture models, and use them to elastically deform the original images, creating new realistic samples out of the original ones. Each of these steps are detailed in the following.

2.2.1 GMM Enconder-Decoder CNN Mixture models are conveniently used to describe systems composed by subpopulations within an overall population. Gaussian mixture models (GMM) in particular, are widely applied in different areas, ranging from speaker recognition to image retrieval, finance, electron and atomic position, spectroscopy, cellular components. GMM has also been shown to be useful for modeling for colour features in order to classify coloured textures in images (Permuter et al., 2003). Herein, we propose to model images as GMM models to extract their latent data representation. To simplify our model and ease convergence, we used the magnitude image from the four-channel images, such that the GMM was composed by simple bi-dimensional Gaussians.

A n-dimensional Gaussian distribution is written as

$$\mathcal{N}(\vec{x}, \vec{\mu}, \mathbf{\Sigma}) = \frac{1}{\sqrt{2\pi\mathbf{\Sigma}}} \exp\left[-\frac{1}{2}\left(\vec{x}-\vec{\mu}\right)^T \mathbf{\Sigma}^{-1}\left(\vec{x}-\vec{\mu}\right)\right], \tag{1}$$

where $\vec{\mu}$ is the mean and $\mathbf{\Sigma}$ is the covariance matrix

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho_{1,2}\sigma_1\sigma_2 & \dots \\ \vdots & \ddots & \\ \rho_{n,1}\sigma_n\sigma_1 & & \sigma_n^2 \end{bmatrix}. \tag{2}$$

The mixture of K Gaussian distributions is expressed as

$$\mathcal{M}(\vec{x}) = \sum_{i=1}^{K} \alpha_i \, \mathcal{N}(\vec{x}, \vec{\mu}_i, \mathbf{\Sigma}_i), \sum_{i=1}^{K} \alpha_i = 1. \tag{3}$$

Here, we introduce an architecture, represented in Figure 3, for end-to-end unsupervised learning of mixture of multivariate Gaussian distributions. The parameters of the mixture are learned from a latent representation of the input data at the same time they are used to reconstruct the input using Eq. 1. A convolutional encoder is used to create a latent representation $\vec{y}$ for

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-3-W2-2020-71-2020 | © Authors 2020. CC BY 4.0 License.
Primary publication at IEEE Xplore: https://doi.org/10.1109/LAGIRS48042.2020.9165670

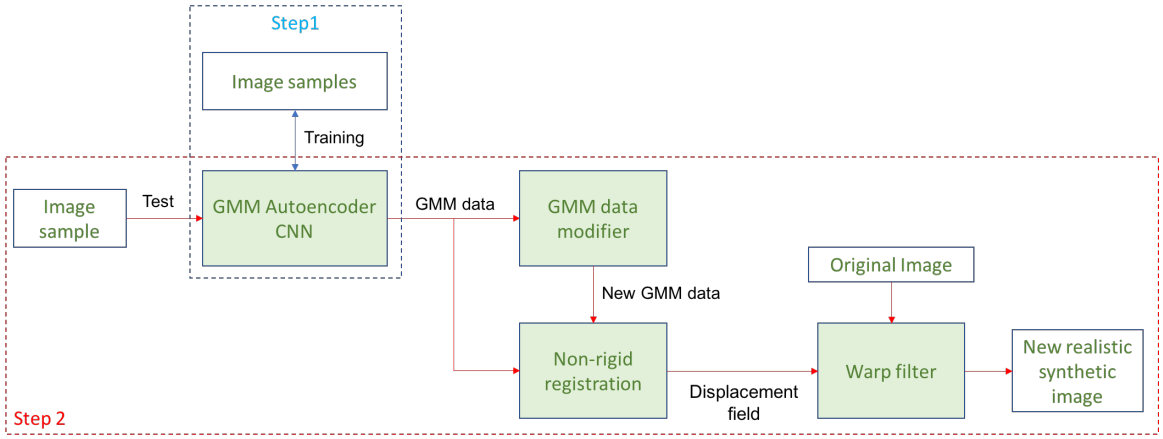

Figure 2. *Our two-step approach: first we extract the images latent data representation using a GMM Autoencoder CNN; then we use the CNN output to generate new realistic images using a non-rigid registration schema..*

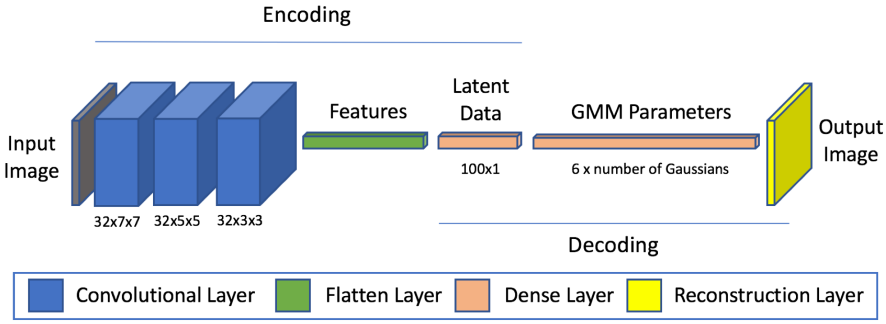

Figure 3. The GMM enconder-decoder architecture is presented in details. Our proposed architecture has a encoder composed by three convolutional layers, followed by a Flatten layer that creates a feature array, and a dense layer that creates the latent data information; and a decoder that gets the latent data information and maps it into GMM parameters through another dense layer, ultimately used to reconstruct the density output image. A trained network is capable of mapping from connvolutional features to latent data, and from latent data to a GMM that resembles the input image.

$n-$dimensional input data $\vec{x}$, with $n = 2$ (considering our one-channel bi-dimensional input images), according to

$$\vec{y} = \Phi\left(\mathbf{W}_0\vec{x} + \vec{b}_0\right). \tag{4}$$

The latent representation $\vec{y}$, which is of size $\ell \times 1$ (in our experiments $\ell = 100$), is then used as input for dense layers that in parallel estimate the parameters α_i, $\vec{\mu}_i$ and $\mathbf{\Sigma}_i$, $0 \leq i \leq K$, the "Gaussian Parameters" layer in Figure 3. Because of the constraint expressed in Eq. 3, we use `softmax` activation for estimating α. We used `tanh` for estimating the mean vector of each component, which lie in the range $[-1, 1]$ representing the domain of the input data. Standard deviation and correlation parameters are estimated through `sigmoid` activation and they are composed to build the covariance matrices given by Eq. 2. Therefore, the parameters of our mixture model can be expressed as

$$\vec{\alpha} = softmax\left(\mathbf{W}_1\vec{y} + \vec{b}_1\right), \tag{5}$$

$$\vec{\mu}_1, \ldots, \vec{\mu}_K = tanh\left(\mathbf{W}_2\vec{y} + \vec{b}_2\right), \tag{6}$$

$$\vec{\sigma}_1, \ldots, \vec{\sigma}_K = sigmoid\left(\mathbf{W}_3\vec{y} + \vec{b}_3\right), \tag{7}$$

$$\rho^{(1)}_{1,2}, \ldots, \rho^{(K)}_{n-1,n} = sigmoid\left(\mathbf{W}_4\vec{y} + \vec{b}_4\right). \tag{8}$$

In total, our architecture estimates $1/2(n^2 + 3n + 2) \times K$ parameters (for bi-dimensional images, $6 \times K$ parameters, as shown in Figure 3) from the latent representation of the input data. The last layer in our architecture, the "Reconstruction Layer" in Figure 3, uses the estimated parameters to create the n-dimensional density map in Eq. 3. The resulting density map is compared to the input data according to a root mean square logarithmic error loss function, and visual results of the training process are shown in Figure 4-I.

We defined experimentally a total of $K = 50$ Gaussians as enough to represent the latent data from training samples. The tuning of the number of Gaussians can be easily performed by defining a target average squared root error between reconstructed images and the original ones when the network converges. As intuitively expected, this error decreases when the number of GMM Gaussians increases, creating a richer representation of the input images.

2.2.2 Latent Data Deformable Decoding Having trained the GMM enconder-decoder CNN, we compute the samples latent data representation, modify them and generate new samples using a deformation schema based on decoded GMMs, as shown in Figure 2.

First we submit a given sample to our GMM CNN, compute the original latent data representation, and decode it into the

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-3-W2-2020-71-2020 | © Authors 2020. CC BY 4.0 License.
Primary publication at IEEE Xplore: https://doi.org/10.1109/LAGIRS48042.2020.9165670

corresponding GMM model, say GMM_a.

Then change GMM_a to create a new sample. Modifying GMMs might be executed in many different ways, and in this work we propose to do that indirectly by modifying the latent data to be decoded into GMMs. This way we avoid dealing with any numerical constraints of modifying GMMs, and let the decoder do it by simple considering a slightly different latent data as input. We change the latent data array in a stochastic fashion by multiplying each value by a random number taken from a normal distribution controlled by the standard deviation σ, and create new latent data in training time. Then we submit this new data to our decoder and create a new GMM model, say GMM_b.

With the original GMM_a and the created GMM_b, we propose to create new samples by deforming the orignal sample, modelled by GMM_a, towards the created GMM_b. The Level Sets motion registration filter (provided by SimpleITK (Yaniv et al., 2018)) was used to deform the input image. We take GMM_a as the moving image and GMM_b as the fixed image, and run the registration procedure. Then we use the computed displacement field and apply it to warp the original input image, deforming the sample and creating a new sample that resambles GMM_b. We used 50 iterations for the registration method, as default in SimpleITK.

It is noteworthy that deformation is likely to modify the geometry of objects found in the scene, and this is highly undesirable for some land cover classes, such as buildings and cars. Our disentangled methodology allows us to cope with this problem by simply erasing the deformation field of objects from these classes. This way we deform classes that can be deformed, and preserve the geometry of objects highly dependent on geometry.

This simple schema allow us to create realistic new samples, and control to some extent, the deformation magnitude observed in the data augmentation process, as shown in Figure 4. Our method preserves the geometry of buildings, and allows for stronger deformations of the original and annotated images, by increasing the standard deviation of latent data modifier, as observed in the dashed yellow, blue and red lines in Figure 4. The lines illustrate the deformation observed in the vegetation, impervious surface and trees classes for different deviations.

2.3 CNN semantic segmentation architectures

To evaluate the impact of our data augmentation method we tested two basal fully convolutional network architectures for semantic segmentation: SegNet (Badrinarayanan et al., 2017) and U-Net (Ronneberger et al., 2015). Both architectures are composed by a sequence of convolutional layers organized in blocks with down-sampling and up-sampling layers, and with skip connections in the case of U-Net. This encoder-decoder architectures segment images by mapping them into the respective labelled data. Details are fully documented in the respective papers (Badrinarayanan et al., 2017, Ronneberger et al., 2015). While many more powerful architectures are available for segmenting images, we consider that the evaluation of state-of-art semantic segmentation networks is beyond the scope of this paper, since our contribution is solely on improving training sets. We further expect that semantic segmentation networks in general would benefit from using richer and more balanced training sets.

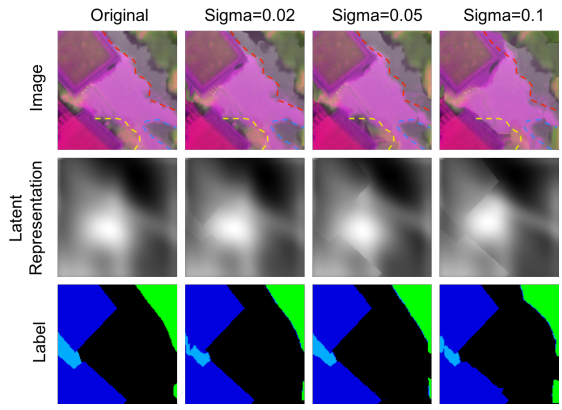

Figure 4. *Generation of new samples given an input image and a GMM modifier standard deviation. One can observe buildings in dark blue, impervious surfaces in black and vegetation in green. While buildings geometry is preserved, one can notice stronger deformations with higher GMM modifier standard deviations (sigma), highlighted by the yellow, blue and red dashed lines. .*

3. EXPERIMENTAL DESIGN

Our image database consisted of 10 full images for training, 3 for validation and 3 for testing. We used a stochastic sampling method that delivers a batch of 128×128 tiles upon request, each of them taken at random from the available training images and at randomly selected x and y positions. The sampling method also filtered tiles where less than 3 classes were present, to cope with the imbalanced nature of land cover data. Each tile was also rotated at random considering 0, 90, 180 and 270 degrees. Additionally, each sorted tile was deformed twice using our proposed method, so that each sample selection generated 3 training samples: 1 original and 2 deformed.

It is important to notice that this procedure is stochastic and does not necessarily derive the same training tiles every time (we used seeds to the random number generators to allow reproducibility of experiments). Also, since the deformation module has a random component in the latent data modification, we can generate different samples during training.

Validation and testing procedures consisted of tiling the validation and test images with a stride of 30 pixels, submitting them to the trained networks and aggregating the results into probability images per class. The final segmentation was obtained by finding the class with the highest probability per pixel in the resulting probability images.

The networks were trained using the $128x128$ samples and the respective annotated data. All training scheme used the same configuration: 150 epochs (enough for convergence) composed each by 150 steps. Each step comprised a batch of 36 samples taken at random (12 original, 24 deformed), so each epoch consisted of 5400 samples. We used Adam (Kingma, Ba, 2014) to optimize the gradients with initial learning rate of 0.0002 and momentum of 0.5. The loss function used was the categorical cross-entropy, commonly used for multi classes using CNNs. Validation ans test was performed considering the fixed set of validation images, not augmented whatsoever.

4. RESULTS AND DISCUSSION

For evaluating our method, we performed a total of four experiments using different training configurations:

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-3-W2-2020-71-2020 | © Authors 2020. CC BY 4.0 License.
Primary publication at IEEE Xplore: https://doi.org/10.1109/LAGIRS48042.2020.9165670

1. SegNet architecture and disabling our data augmentation method – samples were taken at random, rotated, but not deformed;
2. SegNet architecture and enabling our data augmentation method – samples were taken at random, rotated, and deformed using $sigma = 0.02, 0.05, 0.1$;
3. U-Net architecture and disabling our data augmentation method – samples were taken at random, rotated, but not deformed;
4. U-Net architecture and enabling our data augmentation method – samples were taken at random, rotated, and deformed using $sigma = 0.02, 0.05, 0.1$;

As observed in Figure 5 and in Table 1, U-Net performed considerably better that SegNet. It is also noteworthy that our data augmentation method was able to improve semantic segmentation performance in both architectures. Increasing the GMM modifier standard deviation ($sigma$), lead to better results, even though very high $sigma$ values would probably generate unrealistic patterns due to the non-rigid deformation applied.

Analyzing the F1-score per class, some important aspects are observed. We notice, that our gains are much related to deformed classes, and even if we increased score also in some non-deformed classes, in others no relevant improvement was observed. This finding was expected, since deformation added heterogeneity to these classes - and therefore robustness to segmentation models. At the same time the spatial modification of deformation field allowed our model to keep performance in classes where the geometry is highly discriminant.

It is interesting to observe the visual outcome highlighted by the red slashed squares in Figure 5: our method allowed a finer segmentation of thin structures in the SegNet architecture (a common issue in this architecture, which seemed to be improved by data augmentation), or a better definition of buildings in the U-Net architecture.

Numerically, considering the SegNet architecture, we observed a gain of 2.3% in the overall F1-score in our best configuration, but all configurations using our data augmentation method achieved better performances in comparison with the model trained only with scaling and rotation. In the U-Net architecture results were less impressive, but still all the best configurations for all classes were consistently linked to our data augmentation method. The small gains in performance observed in U-Net might be related to the fact that skip connections allow this architecture to cope with more detailed information for segmenting the images, which visually seems to be related to our gains in the SegNet architecture.

Considering the overall F1-score, our methodology delivered competitive results when compared with the benchmark provided by ISPRS Vaihingen 2D Labelling challenge, considering the semantic segmentation architectures used. While the state-of-art delivers around 90% of overall F1-score, our best configuration delivered 87.7%. This result, however, must be taken in perspective: our goal in this paper was to evaluate the increase in performance that our data augmentation approach brings, and not really the final segmentation score achieved. State-of-art methods in this benchmark use multi-scale and negative reinforcement techniques that were not explored in this work. We believe these methods would also benefit from our data augmentation approach, but we consider that such evaluation is beyond the scope of this paper.

5. CONCLUSIONS

Data augmentation is an usual solution for dealing with very imbalanced databases, such the ones observed in land cover segmentation. When it comes to CNN-based semantic segmentation, the overall performance is tightly related to the distribution of objects in the training scenes. Scenarios with scarce objects tends to be problematic, since modifying images to balance a semantic segmentation training database is not a trivial task. Therefore, innovative methods for data augmentation are usually required, specially because rotation, translation and scaling might be not sufficient for delivering satisfactory results.

In this sense, we proposed a new method, that allows for realistic data augmentation of selected samples, reducing the imbalance of semantic segmentation databases. Our method consisted of sequence of steps. First, we created a GMM encoder-decoder CNN to encode input images into a compact latent data representation to be decoded into mixture of Gaussians that best represents the input. Then, we used a non-rigid deformation schema for generating new samples by deforming an original GMM into a new GMM, decoded from a stochastic modification of the original latent data representation. This simple pipeline allowed us to perform data augmentation in training time, and delivered encouraging results in two different semantic segmentation architectures.

For further research, we intend to explore the class awareness of our method and test the use of GANs to reconstruct new samples from the GMMs decoded by the modified latent data representations.

REFERENCES

Audebert, N., Le Saux, B., Lefèvre, S., 2017. Semantic segmentation of earth observation data using multimodal and multi-scale deep networks. S.-H. Lai, V. Lepetit, K. Nishino, Y. Sato (eds), *Computer Vision – ACCV 2016*, Springer International Publishing, Cham, 180–196.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12), 2481-2495.

Blaschke, T., Hay, G. J., Kelly, M., Lang, S., Hofmann, P., Addink, E., Queiroz Feitosa, R., van der Meer, F., van der Werff, H., Vancoillie, F., Tiede, D., 2014. Geographic object-based image analysis: towards a new paradigm. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, 180–191. http://dx.doi.org/10.1016/j.isprsjprs.2013.09.014.

Bokusheva, R., Kogan, F., Vitkovskaya, I., Conradt, S., Batyrbayeva, M., 2016. Satellite-based vegetation health indices as a criteria for insuring against drought-related yield losses. *Agricultural and Forest Meteorology*, 220, 200 - 206.

Buda, M., Maki, A., Mazurowski, M. A., 2017. A systematic study of the class imbalance problem in convolutional neural networks. *CoRR*, abs/1710.05381. http://arxiv.org/abs/1710.05381.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A. A., 2018. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, 35(1), 53-65.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-3-W2-2020-71-2020 | © Authors 2020. CC BY 4.0 License.
Primary publication at IEEE Xplore: https://doi.org/10.1109/LAGIRS48042.2020.9165670

Table 1. F1-scores for the different tested models: average by class and overall.

FCN model	*sigma*	Imp. Surfaces	Buildings	Low Veg.	Trees	Cars	Average	Overall
SegNet	NA	0.8713	0.8816	0.7737	0.7522	0.5954	0.7748	0.8317
SegNet	0.02	0.8741	0.8788	**0.7876**	**0.8273**	0.5859	0.7907	0.8459
SegNet	0.05	**0.8862**	0.8906	0.7866	0.7848	**0.6893**	**0.8075**	0.8503
SegNet	0.1	0.8805	**0.9084**	0.7829	0.7980	0.4303	0.7600	**0.8545**
U-Net	NA	0.9029	0.9131	0.8022	0.8153	0.6469	0.8161	0.8711
U-Net	0.02	0.8908	0.9084	0.8064	0.8146	0.6125	0.8065	0.8633
U-Net	0.05	0.9024	0.9209	**0.8114**	0.8095	0.6619	0.8212	0.8728
U-Net	0.1	**0.9035**	**0.9221**	0.8106	**0.8201**	**0.6989**	**0.8311**	**0.8769**

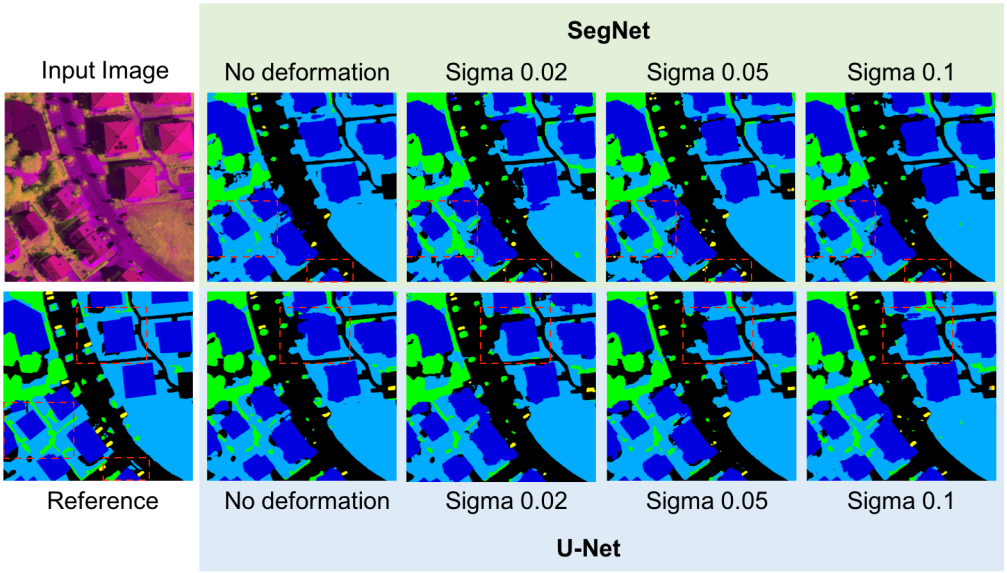

Figure 5. Visual outcome of our experiments. Buildings are observed in dark blue, impervious surfaces in black, low vegetation in light blue, cars in yellow and trees in dark blue. It is possible to visually attest the impact of our data augmentation method in both architectures, while U-Net performs better than SegNet. One can also notice that increasing $sigma$ usually leads to better results.

Gandhi, G. M., Parthiban, S., Thummalu, N., Christy, A., 2015. Ndvi: Vegetation Change Detection Using Remote Sensing and Gis, A Case Study of Vellore District. *Procedia Computer Science*, 57, 1199 - 1210. 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Rodríguez, J. G., 2017. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *CoRR*, abs/1704.06857. http://arxiv.org/abs/1704.06857.

Guo, Y., Chen, Q., Chen, J., Wu, Q., Shi, Q., Tan, M., 2019. Auto-Embedding Generative Adversarial Networks for High Resolution Image Synthesis. *arXiv:1903.11250 [cs]*. http://arxiv.org/abs/1903.11250. arXiv: 1903.11250.

Karras, T., Laine, S., Aila, T., 2018. A Style-Based Generator Architecture for Generative Adversarial Networks. *CoRR*, abs/1812.04948. http://arxiv.org/abs/1812.04948.

Kingma, D. P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980. http://arxiv.org/abs/1412.6980.

Kingma, D. P., Welling, M., 2013. Auto-Encoding Variational Bayes. *ArXiv e-prints*.

Papadomanolaki, M., Vakalopoulou, M., Zagoruyko, S., Karantzalos, K., 2016. Benchmarking Deep Learning Frameworks for the Classification of Very High Resolution Satellite Multispectral Data. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 83-88.

Park, T., Liu, M.-Y., Wang, T.-C., Zhu, J.-Y., 2019. Semantic Image Synthesis with Spatially-Adaptive Normalization. *arXiv:1903.07291 [cs]*. http://arxiv.org/abs/1903.07291. arXiv: 1903.07291.

Permuter, H., Francos, J., Jermyn, I. H., 2003. Gaussian mixture models of texture and colour for image database retrieval. *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, 3, III–569–72 vol.3.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *CoRR*, abs/1505.04597. http://arxiv.org/abs/1505.04597.

Yaniv, Z., Lowekamp, B. C., Johnson, H. J., Beare, R., 2018. SimpleITK Image-Analysis Notebooks: a Collaborative Environment for Education and Reproducible Research. *Journal of Digital Imaging*, 31(3), 290–303. https://doi.org/10.1007/s10278-017-0037-8.

Yu, H., Yang, Z., Tan, L., Wang, Y., Sun, W., Sun, M., Tang, Y., 2018. Methods and datasets on semantic segmentation: A review. *Neurocomputing*, 304, 82 - 103.

Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4), 8-36.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-3-W2-2020-71-2020 | © Authors 2020. CC BY 4.0 License.
Primary publication at IEEE Xplore: https://doi.org/10.1109/LAGIRS48042.2020.9165670