Deep Learning Models to Count Buildings in High-Resolution Overhead Images

Sylvain Lobry
Wageningen University
Wageningen, The Netherlands
sylvain.lobry@wur.nl

Devis Tuia
Wageningen University
Wageningen, The Netherlands
devis.tuia@wur.nl

***Abstract*—This paper addresses the problem of counting buildings in very high-resolution overhead true color imagery. We study and discuss the relevance of deep-learning based methods to this task. Two architectures and two loss functions are proposed and compared. We show that a model enforcing equivariance to rotations is beneficial for the task of counting in remotely sensed images. We also highlight the importance of robustness to outliers of the loss function when considering remote sensing applications.**

***Index Terms*—Deep learning, remote sensing, regression, counting, equivariance, loss functions**

I. Introduction

Thanks to their ability to learn multiscale, translation-invariant features, deep learning methods have proven useful to tackle remote sensing tasks such as single pixels segmentation, image classification and anomaly detection [1]. Instead, here we study the suitability of such methods for counting objects in remotely sensed images. Counting can be useful for several tasks, from urban planning to crowd estimation (by counting the number of pedestrians or cars [2]) or vegetation [3] and wildlife monitoring [4]. In remote sensing, efforts have been dedicated to counting cars in very high-resolution images [5]–[7] which can be used as an estimation of the crowdedness of a specific place. This work is focused on the specific task of counting buildings in remotely sensed images. This task has been tackled by [8] based on the assumption that the number of buildings is linearly correlated to the number of line segments contained in the image.

While solving the same problem, counting methods can be divided in three categories based on their formulation:

1) Object detection followed by counting: these methods first detect the objects of interest either using classical feature extraction methods (e.g. SIFT keypoints [5]) or in recent works using convolutional neural networks (CNNs) [9]. It allows to profit from the research dedicated to object detection and have the advantage of giving a visual explanation of the results (*via* the localized objects). However, they tend to fail when some objects are partially occluded or merged.
2) Integration of an estimated density: another family of approaches estimates a density map from the images which can then be integrated for the actual counting. Compared to the object detection, which is a discrete counting, these approaches allow to count partially occluded objects while retaining spatial information. Recent works use CNN-based models to estimate the density map. In [4], object/background separation is predicted along with the density map and an uncertainty estimation. In [10], a composition loss is used to penalize errors both at the density map and counting levels.
3) Regression: a third approach is to treat the problem as a regression task. In this setting, the model predicts directly the number of objects of interest. These models are trained with image/count pairs and must learn the objects of interest by themselves [3].

This work has been supported by the Swiss National Science Foundation (SL, DT, grant PZ00P2-136827, http://p3.snf.ch/project-136827)

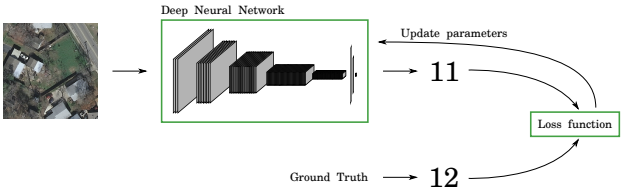


Fig. 1: Counting as a regression task.

Our proposed approach is of the last category and is summarized in Figure 1: we compare a classical architecture (Resnet-50) to the recently proposed RotEqNet [11], which by encoding rotation equivariance is particularly suited to train CNNs from scratch with smaller datasets. We also compare two loss functions, making different assumptions about the types of errors to be minimized. We applied the proposed architectures to the task of counting buidings from the INRIA dataset [12]

II. Method

As mentioned in the introduction, methods based on object detection fail when objects are merged. This is frequently the case when considering buildings. Therefore, we formulate our problem as a regression task. More precisely, we want to predict the number of buildings $\hat{y}$ in a remotely-sensed image patch x. We will solve this problem in a supervised manner: we have a collection of N image patches $\boldsymbol{x} = \{x_1, \ldots, x_N\}$, for which we know the true number of buildings $\boldsymbol{y}$. When using a deep learning approach to solve such a problem, we

Fig. 2: RotEqNet architecture considered in this study. The image is first rotated (we consider 17 different orientations) and convoluted to obtain a 6 channels vector field. After max-pooling, the same operation is repeated to obtain a 16 channel vector field. After the last layer (of depth 128), the vector fields are vectorized and passed through a multi-layer perceptron with 1 hidden layer of dimension 1024 to obtain the final prediction.

need to define both the network architecture and an appropriate loss function. In the following, we study both these elements.

A. Network architectures

Convolution operations are inherently equivariant to translations. In general this is a desirable property when dealing with images of all kinds and we argue that this partly explains the performances achieved by CNNs in computer vision. However, the task of counting in a image is also invariant to rotations, which is not enforced by CNN models by design. To this effect, we compare a rotation equivariant network [11] learned end-to-end with a more classical approach fine tuning a ResNet-50 [13] pre-trained on ImageNet.

Finetuned ResNet50 It has been observed that, when adding more layers to an already deep (and converging) model, performances first stagnate and then decrease. This is the so-called *degradation* problem. This should not happen, as the extra layers should be able to learn an identity mapping, and therefore at least maintain the original performances. [13] solves this issue by learning residual mappings instead of the usual direct mapping. This can be interpreted as a collection of many small paths instead of a single very deep network [14], which in practice makes the network efficient. In this work, we use the ResNet-50 architecture (50 layers residual network) from [13].

A common approach to solve a specific task with a general-purpose network is to fine-tune it: the first layers (which achieve generic, lower level features extraction) are kept intact while the last fully connected layer is replaced and trained to solve the problem. Recent remote sensing applications of such a procedure can be found in [15], [16].

RotEqNet. When dealing with remote sensing images, few assumptions can be made about objects' orientations. Intuitively, we would like our model to recognize buildings, which would most likely include the detection of straight angles (buildings are often rectangles). However, a non-invariant general-purpose model needs to learn to detect these straight angles in every direction, which adds complexity.

Marcos *et al.* [17] tackle this issue by applying the convolution operator to different orientations of the input image and passing further only the highest scoring orientation ('orientation pooling') as a vector field. A convolution operator working on this vector field is then defined and used as a filter for latter stages in the architecture. Therefore the model is equivariant to rotations and has been shown to match performances of much larger models in remote sensing semantic segmentation [11]. We use the architecture shown in Figure 2. Note that since we are performing regression of a single scalar, we do not seek equivariance, but rather invariance (for a rotation of the image, we want the response to remain unchanged). Nonetheless, in [17] authors demonstrate that rotation equivariant CNNs are also suited to tackle rotation invariant tasks.

B. Loss functions

When training a neural network, the choice of the loss function is essential as it will define the error which will be backpropagated and hence, how the network's parameters will be updated. A classical loss function for regression [3] is the Mean Square Error (MSE) :

$$L_{MSE}(y, \hat{y}) = \frac{1}{N} \sum_i (y_i - \hat{y}_i)^2 . \quad (1)$$

Since it is unbiased, MSE will learn mostly from the worst predictions. This can be a problem when only few examples are particularly difficult, or when the ground truth is incorrect. In our case, errors in the dataset can be attributed to the following reasons:

1) the source of information (e.g. cadastre) is not updated exactly as the image is acquired (see Figure 4a);
2) the delineation of the buildings is not clear (e.g some buildings can be merged, see Figure 4b).

A more robust alternative to MSE is the pseudo-Huber loss, which is a smoothed approximation of the Huber loss [18]:

$$L_{Huber}(y, \hat{y}) = \frac{1}{N} \sum_i \delta^2 \left(\sqrt{1 + \frac{y_i - \hat{y}_i}{\delta}^2} - 1 \right) ; \quad (2)$$

where δ is a factor determining the slope when $|y_i - \hat{y}_i| \gg \delta$. This function has a quadratic behavior when $|y_i - \hat{y}_i| \ll \delta$ and is linear for $|y_i - \hat{y}_i| \gg \delta$. This means that strong outliers will not have a larger derivative than wrongly estimated samples. Both losses are shown and compared in Figure 3.

III. Results and discussion

A. Dataset

For our experiments, we used the training dataset provided by [12]. The aerial orthorectified RGB imagery dataset covers 405 km^2 with a resolution of 30 cm and is spread across five

Fig. 3: Mean Square Error (MSE) and pseudo-Huber loss. Both loss functions shows a quadratic behavior around $y - \hat{y}$, but the pseudo-Huber loss approximates a straight line for extreme values, making it less sensible to outliers.

regions of various densities (from low density, e.g. Tyrol area in Austria, to high density, e.g. Chicago in the USA). For this dataset, a ground truth made from official cadastral records is provided in the form of a binary (building/non-building) segmentation map.

Each region is divided in 36 tiles of size 5000 × 5000 pixels. For each region, we select 4 tiles for testing our model, and use the remaining 32 tiles for training. We regularly sample patches of size 224 × 224 pixels (without overlap), leading to a total of 77440 patches in the training set and 9680 patches for testing. We obtain the ground truth for our specific task by counting the number of connected components in the provided binary segmentation map for each patch. We note that an important proportion of the tiles (32.9%) has no buildings, and that the maximum number of buildings per tile is 38.

B. Experiments

In this section, we report the results achieved by the 4 investigated models (combinations of the two architectures and loss functions). The ResNet-50 models are fine-tuned in their last fully convolutional layer, while the RotEqNet models are trained from scratch. Both models have been optimized with the Adam optimizer introduced in [19] and a learning rate of 10^{-3} until convergence. We fixed the parameter δ of the Huber loss (see Equation 2) to 0.5 experimentally.

As a comparison, we report the results obtained by an approach based on segmentation followed by counting. We used the winning model of the 2017 INRIA challenge (segmentation task on the same dataset) presented in [20] and based on the U-Net architecture of [21] for the segmentation. This model has been trained on the same train/test sets described in subsection III-A.

For each model, we report the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) defined as follow:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_i (y_i - \hat{y}_i)^2}; \quad (3)$$

TABLE I: Results of the studied models on our test dataset. The bold numbers indicate the best performing model for each column.

Architecture	Loss Function	All images		Less than 6 buildings	
		RMSE	MAE	RMSE	MAE
RotEqNet	MSE	2.7655	1.8047	1.8911	1.2576
	Huber	2.7466	1.6608	1.611	0.9831
ResNet-50	MSE	2.5677	1.7844	1.8296	1.3383
	Huber	2.6433	1.6754	1.5452	1.0695
Baseline		5.7666	2.2266	6.1904	1.9102

$$\text{MAE} = \frac{1}{N}\sum_i |y_i - \hat{y}_i|. \quad (4)$$

These metrics are computed on the whole testing dataset ('All images'), or only on the tiles containing strictly less than 6 buildings ('Less than 6 buildings'). Note that the latter subset of tiles represents 69.8% of the testing dataset.

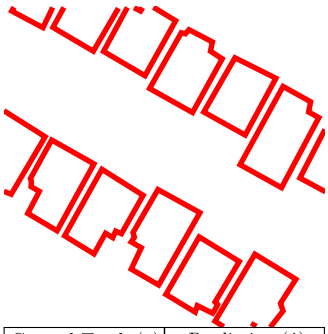
C. Discussion

Results are reported in Table Table I and discussed below.

Architectures: Our models outperform the baseline, showing that the formulation of the problem as a regression task is relevant. When comparing the architectures ResNet-50 generally obtains slightly better performances than RotEqNet. While the MAE is sometimes lower for RotEqNet, the RMSE is always lower for ResNet-50, indicating less variance in its error distribution. However, it is important to note that RotEqNet is much shallower and has not been pre-trained on a multi-million images dataset like ImageNet. While it would be interesting to compare both architectures on similar settings, it was not possible to do so in this study mainly due to the computational cost of pre-training on ImageNet. The fact that the errors are similar confirms that fine-tunning a widely used model trained on a much larger dataset is still a valid methodology, even when the considered problem (regression) is different from the original one (classification). It also confirms that fully training a model enforcing equivariance to rotation is also a good option when considering remote sensing problems, as these problems typically exhibit equivariance (or invariance in our specific case) properties by the overhead perspective.

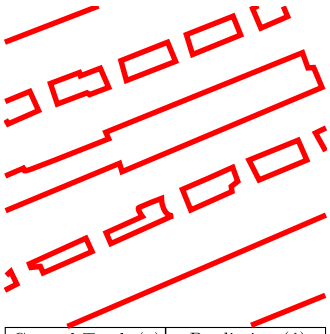
Loss functions: when we consider all images from the dataset both MSE and Huber obtain equivalent results. However, the Huber loss performs better when only considering images with a low number of buildings. This comes from the fact that the MSE is highly influenced by outliers, leading the loss value to be too dependent on outliers and very difficult cases. On the contrary, the Huber loss is more robust to outliers, as its gradient is constant for large errors. In practice, it allows the network to learn from a greater selection of samples during the training procedure.

Qualitative visual analysis: We show visual examples of predictions made by RotEqNet trained with the Huber loss in Figure 4. These examples show two outliers which are



Ground Truth (y)	Prediction ($\hat{y}$)
13	0

(a) Ground truth not up-to-date with the image.



Ground Truth (y)	Prediction ($\hat{y}$)
2	6

(b) Connected buildings in the ground truth.

Ground Truth (y)	Prediction ($\hat{y}$)
22	22

(c) Right prediction.

Fig. 4: Visual examples from the test dataset and predictions made by RotEqNet trained with the Huber loss. The buildings from the ground truth are indicated in red.

particularly problematic if used for the training with the MSE loss:

in Figure 4a, we can see that the ground truth is not up-to-date with the image as it indicates buildings which are not in the image.

in Figure 4b, we can see an example where spatially connected buildings are merged during the counting (due to our approach of counting connected components in the labeling to establish the ground truth). Our model adapted to this behavior by under-predicting the number of buildings (compared to the reality).

Finally, we show in Figure 4c a case where our model predicts the right number of buildings.

IV. Conclusion

We compared two deep neural networks architectures (the widely used ResNet-50 and the rotation equivariant RotEqNet) and two loss functions (the Mean Squared Error and the Huber loss) to solve a 'counting through regression' problem on very high resolution remotely sensed color images. We showed that, besides the widely-used option of fine-tuning a model to a specific task, using a model equivariant to rotations is a good option when considering remotely sensed images, for which a strong prior is present due to the overhead perspective. Moreover, we recommend to use the Huber loss in regression tasks where the dataset shows an important variability, or is prone to outliers (which is frequently the case in remote sensing based counting). Future efforts will be focused on extending this reasoning to several categories of objects, also in the multi-category case, for which co-occurrence priors could also be exploited [22].

References

[1] X. X. Zhu, D. Tuia, L. Mou, G. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geoscience Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[2] V. A. Sindagi and V. M. Patel, "A survey of recent advances in CNN-based single image crowd counting and density estimation," *Pattern Recognition Letters*, vol. 107, pp. 3–16, 2018.

[3] M. V. Giuffrida, P. Doerner, and S. A. Tsaftaris, "Pheno-deep counter: a unified and versatile deep learning architecture for leaf counting," *The Plant Journal*, vol. 96, no. 4, pp. 880–890, 2018.

[4] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in *ECCV*, 2016.

[5] T. Moranduzzo and F. Melgani, "Automatic car counting method for unmanned aerial vehicle images," *IEEE TGRS*, vol. 52, no. 3, pp. 1635–1647, 2014.

[6] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, "A large contextual dataset for classification, detection and counting of cars with deep learning," in *ECCV*, 2016.

[7] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *ICCV*, 2017.

[8] J. Yuan and A. M. Cheriyadat, "Learning to count buildings in diverse aerial scenes," in *ACM SIGSPATIAL*, 2014.

[9] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *ECCV*, 2018.

[10] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *ECCV*, 2018.

[11] D. Marcos, M. Volpi, B. Kellenberger, and D. Tuia, "Land cover mapping at very high resolution with rotation equivariant CNNs: towards small yet accurate models," *ISPRS*, vol. 145A, pp. 96–107, 2018.

[12] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark," in *IEEE IGARSS*, 2017.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[14] A. Veit, M. J. Wilber, and S. Belongie, "Residual networks behave like ensembles of relatively shallow networks," in *NIPS*, 2016, pp. 550–558.

[15] R. Pilipović and V. Risojević, "Evaluation of convnets for large-scale scene classification from high-resolution remote sensing images," in *IEEE EUROCON*, 2017, pp. 932–937.

[16] B. Yu, L. Yang, and F. Chen, "Semantic segmentation for high spatial resolution remote sensing images based on convolution neural network and pyramid pooling module," *IEEE JSTARS*, no. 99, pp. 1–10, 2018.

[17] D. Marcos, M. Volpi, N. Komodakis, and D. Tuia, "Rotation equivariant vector field networks." in *ICCV*, 2017, pp. 5058–5067.

[18] P. J. Huber *et al.*, "Robust estimation of a location parameter," *The annals of mathematical statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[20] B. Huang, K. Lu, N. Audebert, A. Khalel, Y. Tarabalka, J. Malof, A. Boulch, B. Le Saux, L. Collins, K. Bradbury *et al.*, "Large-scale semantic classification: outcome of the first year of Inria aerial image labeling benchmark," in *IEEE IGARSS*, 2018.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[22] M. Volpi and D. Tuia, "Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images," *ISPRS*, vol. 144, pp. 48–60, 2018.