Article

Integrating EfficientNet into an HAFNet structure for Building Mapping in High-Resolution Optical Earth Observation Data

Luca Ferrari[1], Fabio Dell'Acqua [1,†], Peng Zhang [2] and Peijun Du [2,*]

[1] CNIT, Pavia Unit - Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Pavia, Italy; fabio.dellacqua@unipv.it
[2] University of Nanjing, Nanjing, P.R.C.; e-mail@e-mail.com
[*] Correspondence: fabio.dellacqua@unipv.it dupjrs@126.com; Tel.: +39 0382 985664 (F.D.) +81-xxxx-xxx-xxxx (P.D.)
[†] F.D. is also with Ticinum Aerospace, a spin-off company from the University of Pavia, Italy

Abstract: Automated extraction of buildings from earth observation (EO) data is important for various applications, including updating of maps, risk assessment, urban planning, policy making. Combining data from different sensors such as high-resolution multispectral (HRI) and light detection and ranging (LiDAR) has shown great potential in building extraction. Deep learning (DL) is increasingly used in multimodal data fusion and urban object extraction. However, DL-based multimodal fusion networks may underperform due to insufficient learning of "joint features" from multiple sources and oversimplified approaches to fusing multimodal features. Recently, an hybrid attention-aware fusion network (HAFNet) has been proposed for building extraction from a dataset including co-located Very-High-Resolution (VHR) optical images and Light Detection And Ranging (LiDAR) joint data. The system reported good performances thanks to the adaptivity of the attention mechanism to the features of the information content of the three streams but suffered from model overparametrization, which inevitably leads to long training times and heavy computational load. In this paper the authors propose a restructuring of the scheme, which involved replacing VGG-16-like encoders with the recently proposed EfficientNet, whose advantages counteract exactly the issues found with the HAFNet scheme. The novel configuration was tested on multiple benchmark datasets, reporting great improvements in terms of processing times, and also in terms of accuracy. The new scheme, called HAFNetE (HAFNet with EfficientNet integration), appears indeed capable of achieving good results with less parameters, translating into better computational efficiency. Based on these findings, we can conclude that, given the current advancements in single-thread schemes, the classical multi-thread HAFNet scheme could be effectively transformed by the HAFNetE scheme by replacing VGG-16 with EfficientNet blocks on each single thread. The remarkable reduction achieved in computational requirements moves the system one step closer to on-board implementation in a possible, future "urban mapping" satellite constellation.

Keywords: attention mechanism; building mapping; data fusion; EfficientNet; HAFNet; high-resolution imagery (HRI); light detection and ranging (LiDAR); mapping; urban areas

1. Introduction

Building information extraction from Earth observation data is key to a wide range of applications including map generation, urban sprawl monitoring, risk mapping, urban planning. In this framework, the joint use of high resolution imagery and LiDAR data has been proposed, to produce comprehensive results by exploiting the complementary information given by the two data types. Several fusion techniques have been proposed that combine data both at the feature level [1–5] and at the decision level [6,7]; despite the range of solutions available, however, a few unresolved issues remain. In feature-level fusion, some methods use only cross-modal features, which provide good discriminative power most of the times but fail in specific edge cases. On the other hand, individual features combined only at the decision level are often not discriminative enough to produce proper building extraction. However, they can still be useful in cases where a single data source would mislead the classifier because it contains noisy or corrupted information. It is therefore necessary to build a system that utilizes both individual and cross-modal features. Moreover, the fusion strategy should be such that useful discriminative features are highlighted, whereas irrelevant or noisy ones are suppressed. The Hybrid Attention-aware Fusion Network (HAFNet) [8] offers a solution to these problems by introducing the Attention-Aware Multimodal Fusion Block (Att-MFBlock), a computational module used to adaptively re-weight individual and cross-modal features. The proposed model achieves state-of-the-art-segmentation accuracy and provides great performance even in specific edge cases where either data type introduces noise and potentially harmful information. Consider the example in Figure 1, where the DSM dataset suggests that the right half of the building visible on the bottom of the RGB image is not there. The information fed by the DSM dataset is clearly wrong and can negatively impact local results, but the HAFNet structure and specifically the attention mechanism can detect it and filter it out.

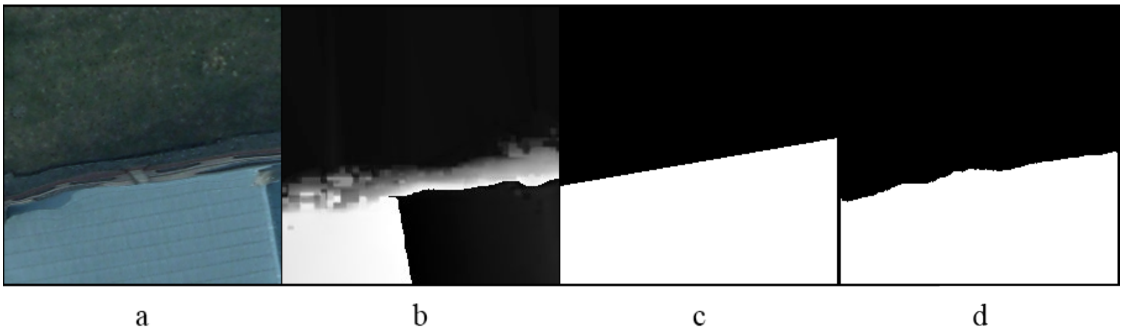

Figure 1. Harmful information in input data. (**a**) RGB patch containing discriminative information. (**b**) DSM patch containing incorrect information. (**c**) Ground truth map. (**d**) Segmentation result. The Att-MFBlock reweights the RGB and the DSM input so that RGB information is highlighted and the damaged DSM information is suppressed.

The high performance of the HAFNet model, however, comes at the cost of an enormous number of parameters. Such overparametrization of the model conveys disadvantages both at the development level and at the deployment level, including slow training, long inference time and massive memory footprint. All the mentioned consequences can pose problems in a time when AI applications are moving on the edge and models are expected to work with very limited computing and memory resources. On-board data processing in spaceborne Earth Observation systems, for instance, is gaining relevance, and methods for different Remote Sensing applications are being developed [9–13]. Moreover, this trend is substantially accelerated by the recent joint effort of multiple Deep Learning researches of providing new implementations of efficient network architectures that limit the overall number of parameters while achieving state-of-the-art performances. These networks [14–16] are built out of custom-designed operation modules that fulfil this task. Motivated by these considerations, in this paper we propose an efficient implementation of the HAFNet model called HAFNetE that exceeds state of the art fusion building extraction performances while affording at the same time a 92% reduction from the original number of network parameters.

2. Building blocks

In this chapter the core elements of the proposed method are presented and described.

2.1. EfficientNet

EfficientNet [16] is a convolutional neural network (CNN) architecture and scaling method that scales the network dimensions (depth, width, resolution) using a compound coefficient. The basic building block of the network is the inverted bottleneck residual block (previously introduced with MobileNetV2 [15]), a custom convolutional module that provides a good compromise between performance and memory footprint. The EfficientNet family of models is specifically designed for cases where computational resources are limited. However, even with a limited number of parameters, the network can still provide great performance. EfficientNet reaches state-of-art transfer accuracy on multiple benchmark datasets with one order of magnitude fewer parameters.

2.2. *Attention-Aware Multimodal Fusion Block*

The Attention-Aware Multimodal Fusion Block is a computational module introduced in [8] to adaptively reweight feature channels from different modalities and therefore highlighting discriminative features and suppressing irrelevant ones. The module is based on the Attention mechanism [17] that produces significant performance improvements. The module is comprised of multiple stages. In the first stage, a global average pooling operation is performed to abstract global spatial information of each channel. Pooled features are then processed in a bottleneck where linear and non-linear operations are applied in order to learn the interactions between channels. The concatenated channel-wise statistics are then multiplied by the corresponding input features. The final fused features are obtained by an element-wise summation of the re-weighted features.

2.3. *HAFNet*

HAFNet (Hybrid Attention-aware Fusion Network), is a multimodal building extraction segmentation network that utilizes HRI RGB images and LiDAR data as its inputs. The overall architecture is comprised of three streams: RGB, DSM and cross-modal. Each stream is a SegNet [18] where the encoder part is characterized by a VGG-16 structure. RGB and DSM features extracted from the input data are fused after each set of convolutional operations with an Attention-Aware Multimodal Fusion Block (Att-MFBlock) in the cross-modal stream. At the decision stage, predictions coming from the three streams are combined using an Att-MFBlock to produce the final segmented output. By using both individual and cross-modal streams it is possible to learn more discriminative features and therefore achieve a comprehensive building extraction result.

3. HAFNetE

In this section we introduce HAFNetE, an efficient hybrid attention-aware fusion network for building extraction. The HAFNetE architecture is based on the previously proposed HAFNet network described in subsection2.3, that utilizes cross-modal and individual features to operate builiding extraction. The model architecture is shown in Figure 2.

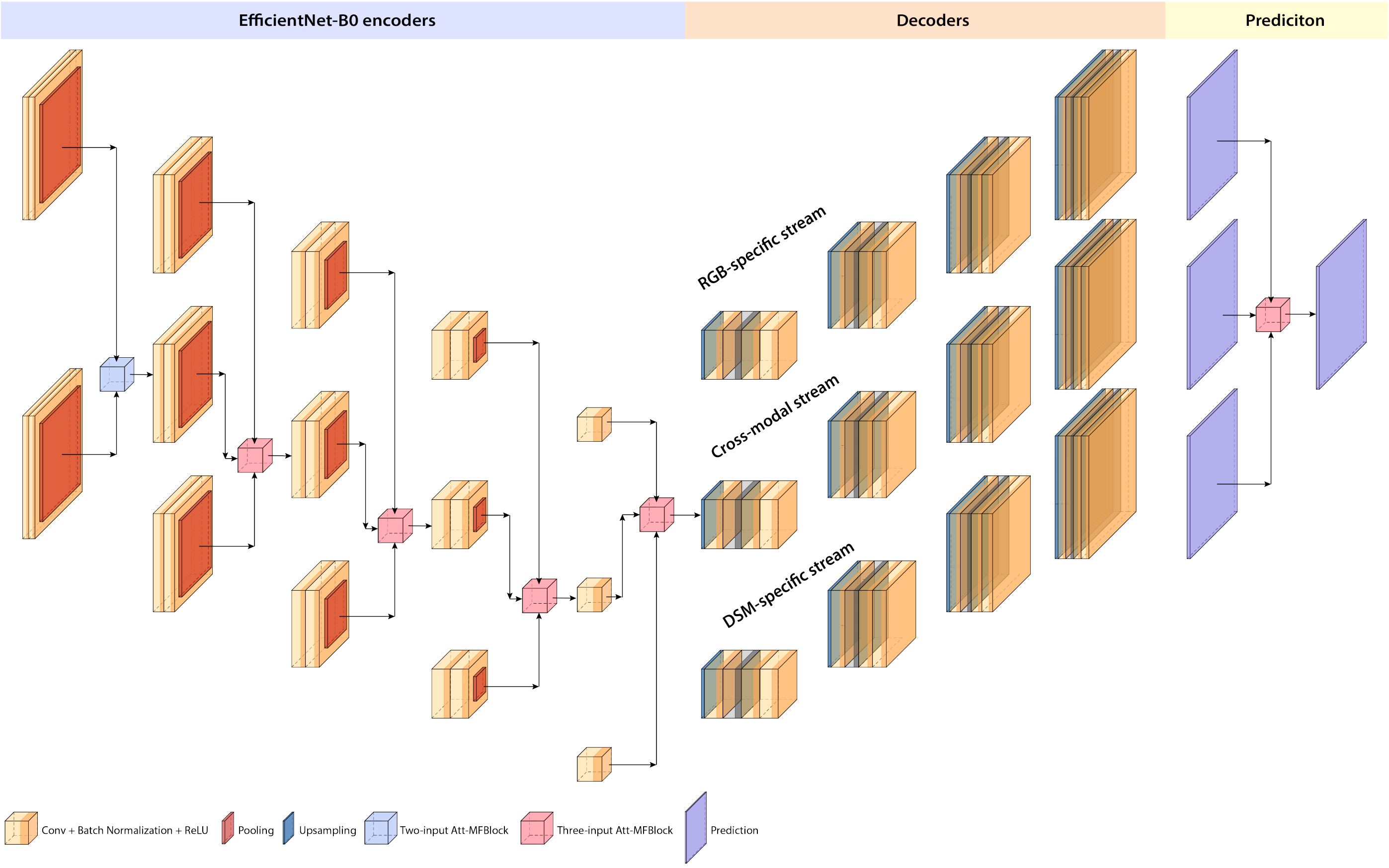

Figure 2. Scheme of the HAFNetE network. The U-shaped trend of the three streams represents the U-net-based structure of each stream

The network is comprised of three subnetworks (streams): the RGB stream, the DSM stream and the cross-modal stream. RGB HRI images and LiDAR-derived DSM data are fed as input to the model where features are extracted respectively by the RGB stream encoder and the DSM stream encoder. The extracted features are then combined in the cross-modal stream encoder by using the previously discussed Attention-aware multi-fusion block. The cross-modal specific stream is added to combine different modalities at an early stage and therefore to learn more discriminative cross-modal features [19]. After the decoding phase, predictions coming from the three streams are fused using the Att-MFBlock [8] to provide a comprehensive building extraction result. Unlike the previous HAFNet model, whose architecture was based on three parallel SegNet-like streams using VGG16-style encoders in each of them, HAFNetE introduces modifications both at the encoder level and at the single stream level. VGG-16 encoders are substituted with EfficientNet encoders. This family of models is specifically designed for good encoding performance even with limited available resources. This translates to simple networks with fewer parameters. Small models yield multiple advantages: faster training, shorter inference times and bearable memory footprint on the system where the model is deployed. Multiple networks characterized by these features exist (MobileNet, MobileNetV2 etc.), however an EfficientNet-B0-type encoder was selected across the candidates because it offers a good compromise in the performance/computational cost trade-off. As a matter of fact, by reducing the number of parameters in the model, performance is likely to decrease. However, EfficientNet, by scaling the number of parameters according to the Compounding Scaling method [16], attains high performances with approximately 11× fewer parameters than classical models like ResNet-50 [20]. An efficiency comparison between EfficientNet models and classical models is reported in Table 1.

Table 1. Comparison of image classification efficiency based on the Imagenet dataset [21]: EfficientNet models [16] vs classical models

Model	Top-5 Acc	Params	FLOPs
EfficientNet-B0[16]	93.3%	5.3M	0.39B
EfficientNet-B2[16]	94.9%	9.2M	1.0B
EfficientNet-B4[16]	96.4%	19M	4.2B
VGG-16[22]	91.9%	138M	19.6B
ResNet-50[20]	93.0%	26M	4.1B
SENet[17]	96.2%	146M	42B

At the individual stream level, the SegNet structure is substituted with a Unet network [23]. Unet has a similar architecture to the previously utilized SegNet and offers a suitable alternative to it, thanks to its effective feature re-localization capability. The conceptually simple architecture of Unet makes it easy and elegant to implement. Moreover, one objective of the research is to assess whether the previously proposed HAFNet three-streams network can be generalized and effectively being employed using different base models such as Unet. For these reasons Unet was selected as the single-stream subnetwork.

4. Experiment Design

4.1. Dataset

The datasets used to train and evaluate the model come from the publicly available data repository of the ISPRS 2D Semantic Labelling Challenge [24]. The dataset covers the two German cities of Potsdam and Vaihingen and it is comprised of high-resolution true-color orthophoto images and the corresponding normalized DSM data. Each dataset has been classified into six common land cover classes and this classification is provided as Ground Truth (GT) to support the supervised learning procedure. The problem we are addressing, i.e. basic building mapping, only uses two labels, i.e. “building” and “non-building”, therefore binary thematic maps containing only the desired classes were created by merging previous classes into the two relevant ones using simple image processing techniques. In Figure 3 an example of an image patch with the corresponding binary thematic map is presented.

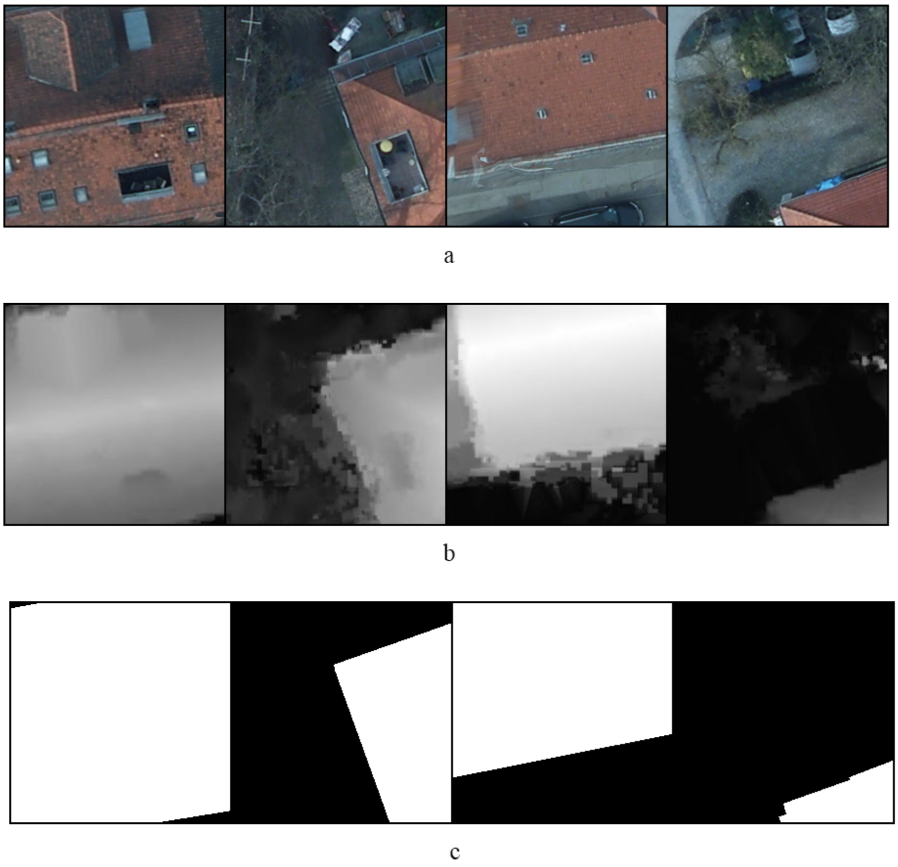

Figure 3. (**a**) RGB image patch. (**b**) DSM patch. (**c**) Corresponding binary thematic map . Building pixels are displayed in white whereas non-building pixels are displayed in black.

The organizers of the Challenge also defined a partition of the dataset into training and testing images. Since our research involved a Deep Learning method and consequently the need of hyperparameter tuning, the dataset was split into three subsets: one for training, one for validation and one for testing. The Potsdam dataset contains 38 images that were randomly assigned to one of the three subsets so that the training subset contained $\approx$ 80%, validation $\approx$ 10% and test $\approx$ 10% of the original images. It is to be noted that visual inspection of orthophoto images revealed noticeable geometrical distortions in some places as in the example of Figure 4.

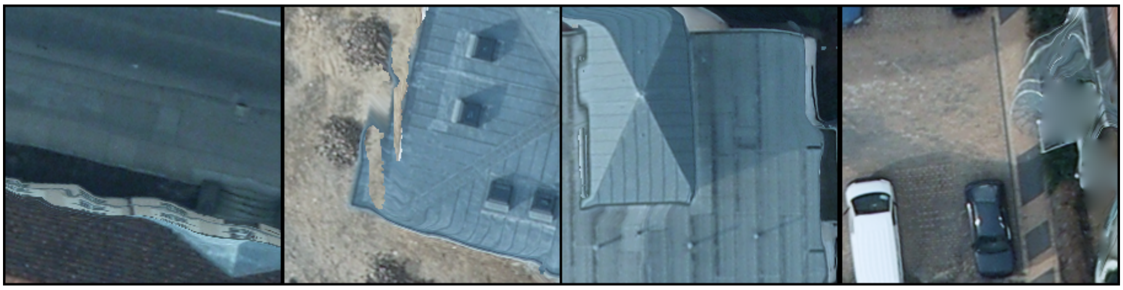

Figure 4. Example of visible distortion in RGB input images.

These are probably due to stitching of multiple images in the production phase, and such distortions are not reflected in the ground truth, thus creating a mismatch between optical data and reference. Although the phenomenon is not very frequent across the dataset, this must be taken into account in evaluating results as it can lead to a underestimation of the actual capability of the model in segmenting the input.

4.2. Model Performance Metrics

For sake of completeness, various standard metrics were used to evaluate the model performance, namely the overall accuracy (OA), the F1 score and the intersection over union (IoU). For the readers' convenience, the definition of the first three metrics are reported below.

$$precision = \frac{tp}{tp + fp}; \quad recall = \frac{tp}{tp + fn}; \quad F_{score} = 2 \cdot \frac{p \cdot r}{p + r} \tag{1}$$

In the expressions above, *tp*, *fp*, *fn* refer to the number of true positive, false positive and false negative cases respectively. The IoU metric is defined as:

$$IoU = \frac{target \cap detected}{target \cup detected} \tag{2}$$

Here, *target* represents the set of building pixels from the ground truth, and *detected* represents the set of pixels assigned to class "building" by the classifier. It is important to note that the number of building pixels is about one order of magnitude smaller than non-building pixels in the average considered image patch. In a segmentation setting with strong class imbalance, IoU is probably slightly more representative than the other measures, since it gauges the overlap rate of the detected target pixels and the labeled target pixels.

4.3. Training procedure

4.3.1. Data processing

The Potsdam dataset contains images the size of 6000 × 6000 pixels, too big to fit entirely into the GPU memory; they were thus partitioned into multiple non-overlapping 224 × 224 tiles. This latter is the size of images in the Imagenet dataset [21] and was indeed selected to maximize the encoding capabilities of the RGB and DSM encoders that were pre-trained on such standard dataset. However, this setting is not binding and the model is flexible on the size of the input images. As previously noted, the dataset is extremely unbalanced and most of the patches extracted from the images do not contain any building pixel. By training the model on this dataset, the net will be biased towards the non-building class, and in the evaluation phase the performance metrics may stay high simply because the model is most of the time correctly predicting that the examined patch does not contain buildings. A data-balancing strategy is thus required to avoid the network to settle on a fairly high accuracy by simply ignoring the comparatively few building pixels altogether, which results into a useless trained network. Two different approaches can be used to tackle the problem. The first method implies using a weighted loss function during training (eg. Weighted Binary Cross Entropy) that assigns a larger weight to samples containing buildings and therefore induces stronger changes in the net parameters when a building is being processed. The second method [25] suggests training the model only on positive examples, i.e patches containing more than a pre-set number or percentage of building pixels in our case. This second approach was selected because it is expected not to affect the generalization capabilities of the network. The method was implemented by filtering the extracted patches so that only patches containing at least 5% of positive pixels (building pixels) survived.

4.3.2. Model training

The proposed HAFNetE was implemented using the PyTorch framework and following the design patterns of the PyTorch library Segmentation Models PyTorch (SMP) [26]. Training and evaluation phases were conducted using a NVIDIA GeForce RTX 1080Ti GPU (11 GB memory). Since data had been previously balanced during the preprocessing phase, a simple non-weighted version of Binary Cross Entropy loss was used. Multiple experiments were carried out to choose the best optimizer for minimizing the loss function (Stochastic Gradient Descent (SGD), Adagrad, Adam). Table 2 shows validation metrics using the different optimization strategies.

Table 2. Validation metrics using different optimization strategies

Optimizer	Validation IoU	Validation F1-score	Validation Accuracy
SGD	85.56%	92.15%	92.07%
Adagrad	89.76%	90.32%	91.98%
Adam	91.58%	95.59%	96.41%

Of all, Adam converged to the highest performance metrics as visible from the percentages reported in Table 2. The observed training curves are shown in Figure 5

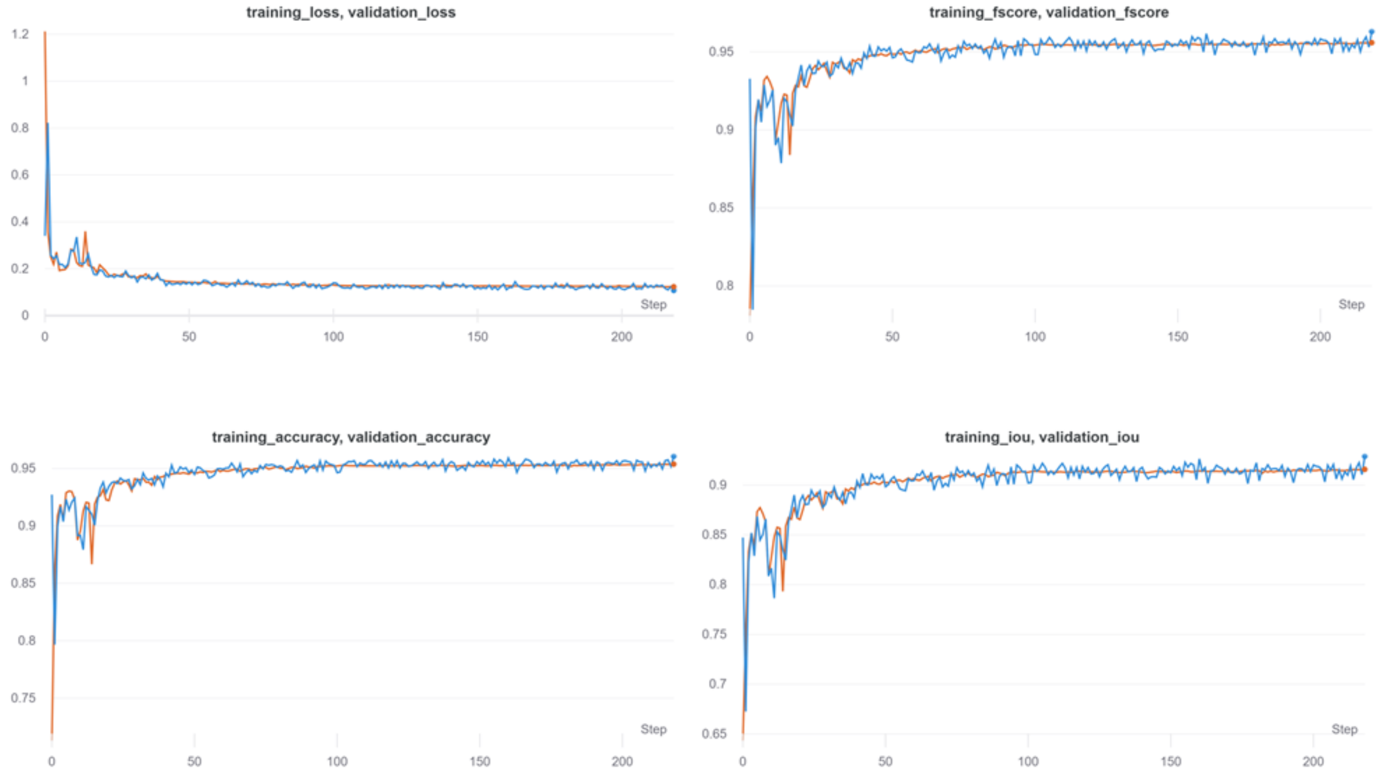

Figure 5. Training (*blue*) and validation (*orange*) curves obtained using the Adam optimizer

As stated earlier, the model encoders were initialized with the pre-trained EfficientNet-B0 weights, so a small learning rate $lr = 1 \cdot 10^{-3}$ was used to optimize loss. The learning rate was modulated using different learning rate schedulation strategies including Cosine Annealing Warm Restart and Multistep LR. In the end, the simplest one (Multi Step LR) was selected, with learning rate reduced by a factor of $\gamma = 0.1$ at epochs 2 and 5. The selected γ factor is a standard setting in learning schedulation while the milestones selected to perform the schedulation steps were found by experiments. The model was trained for 10 epochs for a total time of 50 minutes/run. A batch size of 20 was selected by a trial-and-error procedure in order to saturate the GPU and therefore achieve the maximum training speed given the available hardware acceleration. In order to further increase the overall model performance, the net was fine-tuned for 10 more epochs on a small, augmented subset of the original training set starting from the saved weights of the previous run and continuing the optimization process with a very small learning rate. Results are reported in Table 3.

Table 3. Quantitative validation results after the main training phase and after fine-tuning

Training mode	Validation IoU	Validation F1-score	Validation Accuracy
Main training	91.58%	95.59%	96.41%
Fine-tuning	93.64%	96.68%	97.55%

5. Results and discussion

In this section we show the results of the HAFNetE model presented in Section 3 trained according to the procedure illustrated in subsection 4.3.

5.1. Preliminary assessment

The first aspect to be evaluated is the overall capability of the model of completing the segmentation task. In particular, it is important to assess whether the newly introduced architecture provides at least the same model performance offered by the original HAFNet. The following results are presented after running the model both in the validation phase and in the test phase. After 1.5 training epochs the model reached the same performance of the original HAFNet, probably thanks to a combination of:

- the pre-trained encoders already providing good basic encoding power, plus
- the reduced overall model size speeding up training.

These first training steps set a solid starting point, however we needed to assert that specific characteristics of the previous model were preserved, as confirmed through several experiments: SegNet-like re-localization capability and re-weighting of decision-level features. As stated in Zhang *et al.* [8] regarding adaptability of the scheme to different networks, we can confirm this applies to the HAFNetE model where a Unet network in each thread replaces the previously proposed SegNet. Moreover, the highly discriminative power granted by the attention fusion block at the decision level remains intact. Figure 6 shows the final segmentation results on a set of test patches.

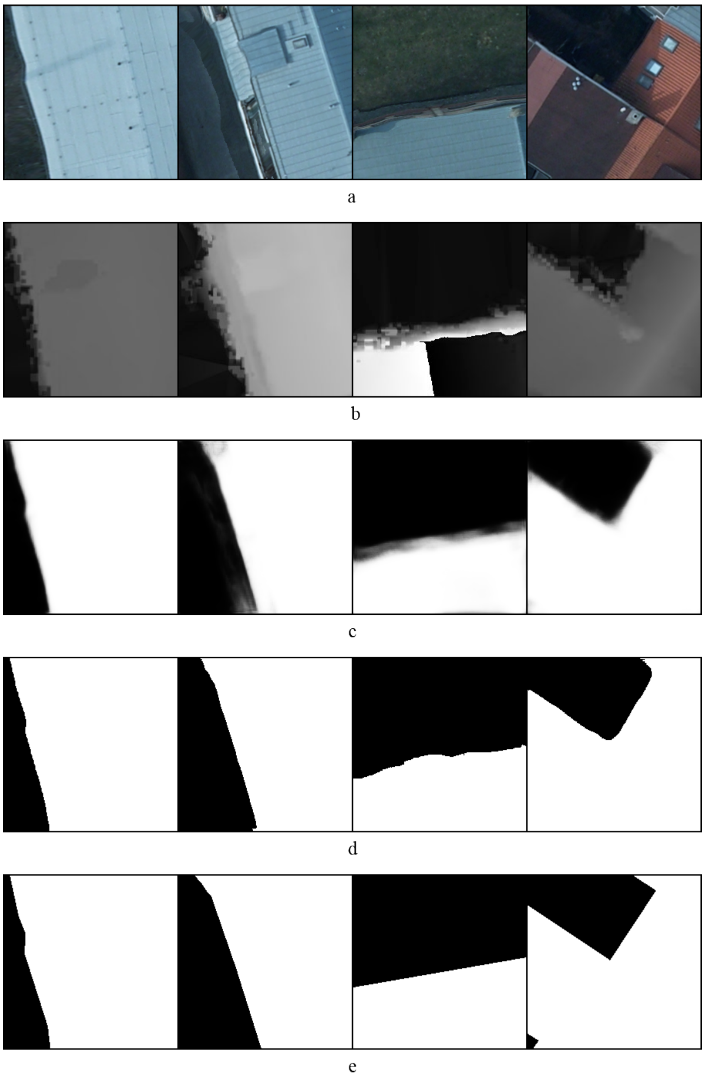

Figure 6. (**a**) Input RGB patches. (**b**) Input DSM patches. (**c**) Model soft predictions. (**d**) Thresholded predictions. (**e**) Label patches. Please note that the corrupted DSM input (**b**) is adaptively reweighted by the Att-MFBlock, thus suppressing misleading information. Thanks to this mechanism a final correct segmentation result is produced.

Performance metrics show that transfer learning is a suitable technique for achieving great segmentation results also in the Earth Observation domain and that the EfficientNet-B0 encoder is highly capable of extracting discriminative features even from the very beginning of the training process. In the next paragraph the benefits of the EfficientNet structure will be presented.

5.2. Novelties introduced

As discussed in Section 1 and in 2.3, HAFNet provides a very powerful tool to solve the building extraction problem, yet it involves a huge number of parameters translating into long training and inference times and a bigger memory footprint. The introduction of the Efficientnet-B0 structure in the model architecture conveys two simultaneous benefits, one at the application level and the other at the computational level, as discussed in the following.

5.2.1. Application level

Features extracted with EfficientNet-B0 encoders are highly discriminative and increase the model segmentation performance from the previously proposed HAFNet. Evaluation metrics show a significant increase in the net capability in detecting and relocating buildings as measured with IoU. Table 4 shows a performance comparison between the HAFNetE and the HAFNet model.

Table 4. HAFNetE and HAFNet performance comparison

Model	IoU	F1-score	Accuracy
HAFNet[8]	90.10%	**98.78%**	97.96%
HAFNetE	**93.64%**	96.68%	97.55%

5.2.2. Resource level

EfficientNet-B0-based streams architecture led to remarkable achievements not only at the application level, but also at a purely computational level. By substituting the VGG16-like encoders in the HAFNet model, the number of parameters shrunk dramatically from 88.978M to 6.982M. This size reduction brought multiple benefits that make the HAFNetE model production-ready:

- *Reduction of training time*: the number of weights in a network is directly correlated with the number of gradients updates that the GPU needs to operate to optimize the loss function. A 92% parameters reduction coupled with an extra pre-trained stream translates to a 80% reduction in training time to reach the same model performance.
- *Reduction of inference time*
- *Reduction of memory footprint*: the model weights are encoded as 32-bit floating point variables. To further speed up the inference procedure and limit the overall model size, weights are usually converted to 16-bit floating point. This conversion can sometimes affect the model performance, but in most cases the impact is negligible. Under these assumptions we can estimate the final model size:

  HAFNet: 88.978 (*Millions of parameters*) $\times$ 16 bits / 8 (*bit-to-byte conversion factor*) $\approx 360MB$
  vs.
  HAFNetE : 6.982 $\times$ 16 bits / 8 $\approx 14MB$

The memory footprint of the proposed model is much smaller than that of the reference one. Moreover, its computational and power demand are small; all these factors make it suitable in principle for on-board processing in spaceborne Earth observation platforms.

6. Conclusions

In this paper, we considered the problem of mapping buildings in urban areas using an AI-based fusion approach on two different and coordinated data sources, namely high-resolution visible optical data and LiDAR data. In this context, we introduced HAFNetE, a modified version of the previously proposed HAFNet model, which is among the most effective models for the considered tasks, albeit at the expense of computational requirements. The proposed network preserves all the powerful features that characterized the HAFNet model and takes a step forward by achieving better segmentation performance while drastically reducing the number of parameters. HAFNetE achieved a IoU figure of 93.64% on the popular benchmark dataset of ISPRS 2D Semantic Labelling Challenge [24]. These features pave the way to new possibilities for real-world exploitation of the devised Attention-aware block scheme. Faster training, shorter inference time, limited computational demand and limited memory footprint open up possibilities for an on-board AI-powered urban mapping application. The model segmentation performance can probably be pushed to the limit by changing the EfficientNet-B0 encoders with a bigger-sized encoder from the same family and therefore paying a price in terms of training/inference time and memory footprint. Future research plans include incorporation of new state of the art efficient networks in the HAFNetE model like for example EfficientNetV2 [27], which has just been released.

Author Contributions: Conceptualization, L.F., F.D., P.D.; methodology, L.F.; software, L.F.; validation, L.F.; formal analysis, L.F., F.D.; investigation, L.F., F.D.; resources, P.D., L.F., P.Z.; data curation, L.F.; writing—original draft preparation, L.F.; writing—review and editing, L.F., F.D; supervision, F.D., P.Z.

Funding: This research received no external funding.

Data Availability Statement: Data used as a basis for our experiments were those used for the ISPRS 2D Semantic Labeling Contest. It can be accessed online at the following URL: https://www2.isprs.org/commissions/comm2/wg4/benchmark/semantic-labeling/.

References

1. Audebert, N.; Le Saux, B.; Lefèvre, S. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing* **2018**, *140*, 20–32.
2. Sun, Y.; Zhang, X.; Xin, Q.; Huang, J. Developing a multi-filter convolutional neural network for semantic segmentation using high-resolution aerial imagery and LiDAR data. *ISPRS journal of photogrammetry and remote sensing* **2018**, *143*, 3–14.
3. Xu, Y.; Du, B.; Zhang, L. Multi-source remote sensing data classification via fully convolutional networks and post-classification processing. IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2018, pp. 3852–3855.
4. Hazirbas, C.; Ma, L.; Domokos, C.; Cremers, D. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. Asian conference on computer vision. Springer, 2016, pp. 213–228.
5. Zhang, W.; Huang, H.; Schmitz, M.; Sun, X.; Wang, H.; Mayer, H. Effective fusion of multi-modal remote sensing data in a fully convolutional network for semantic labeling. *Remote Sensing* **2018**, *10*, 52.
6. Marmanis, D.; Schindler, K.; Wegner, J.D.; Galliani, S.; Datcu, M.; Stilla, U. Classification with an edge: Improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing* **2018**, *135*, 158–172.
7. Marcos, D.; Hamid, R.; Tuia, D. Geospatial correspondences for multimodal registration. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5091–5100.
8. Zhang, P.; Du, P.; Lin, C.; Wang, X.; Li, E.; Xue, Z.; Bai, X. A Hybrid Attention-Aware Fusion Network (HAFNet) for Building Extraction from High-Resolution Imagery and LiDAR Data. *Remote Sensing* **2020**, *12*. doi:10.3390/rs12223764.
9. Furano, G.; Meoni, G.; Dunne, A.; Moloney, D.; Ferlet-Cavrois, V.; Tavoularis, A.; Byrne, J.; Buckley, L.; Psarakis, M.; Voss, K.O.; Fanucci, L. Towards the Use of Artificial Intelligence on the Edge in Space Systems: Challenges and Opportunities. *IEEE Aerospace and Electronic Systems Magazine* **2020**, *35*, 44–56. doi:10.1109/MAES.2020.3008468.
10. Kothari, V.; Liberis, E.; Lane, N.D. The final frontier: Deep learning in space. Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications, 2020, pp. 45–49.
11. Mateo-Garcia, G.; Veitch-Michaelis, J.; Smith, L.; Oprea, S.V.; Schumann, G.; Gal, Y.; Baydin, A.G.; Backes, D. Towards global flood mapping onboard low cost satellites with machine learning. *Scientific Reports* **2021**, *11*, 7249. doi:10.1038/s41598-021-86650-z.
12. Giuffrida, G.; Diana, L.; de Gioia, F.; Benelli, G.; Meoni, G.; Donati, M.; Fanucci, L. CloudScout: A Deep Neural Network for On-Board Cloud Detection on Hyperspectral Images. *Remote Sensing* **2020**, *12*. doi:10.3390/rs12142205.
13. Maskey, A.; Cho, M. CubeSatNet: Ultralight Convolutional Neural Network designed for on-orbit binary image classification on a 1U CubeSat. *Engineering Applications of Artificial Intelligence* **2020**, *96*, 103952. doi:https://doi.org/10.1016/j.engappai.2020.103952.
14. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* **2017**.
15. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.
16. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. International Conference on Machine Learning. PMLR, 2019, pp. 6105–6114.
17. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.
18. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* **2017**, *39*, 2481–2495.
19. Chen, H.; Li, Y. Three-stream attention-aware network for RGB-D salient object detection. *IEEE Transactions on Image Processing* **2019**, *28*, 2825–2835.
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
21. ImageNet. https://image-net.org/index.php.
22. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
23. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.
24. ISPRS 2D Semantic Labeling Contest. https://www2.isprs.org/commissions/comm2/wg4/benchmark/semantic-labeling/.
25. Xia, X.; Lu, Q.; Gu, X. Exploring An Easy Way for Imbalanced Data Sets in Semantic Image Segmentation. *Journal of Physics: Conference Series* **2019**, *1213*, 022003. doi:10.1088/1742-6596/1213/2/022003.
26. Yakubovskiy, P. Segmentation Models Pytorch. https://github.com/qubvel/segmentation_models.pytorch, 2020.
27. Tan, M.; Le, Q.V. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298* **2021**.