remote sensing

MDPI

Technical Note

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, 27100 Pavia, Italy; luca.ferrari11@universitadipavia.it

[2] Department of Geographic Information Science, University of Nanjing, Nanjing 210093, China; pzhangrs@smail.nju.edu.cn (P.Z.); peijun@nju.edu.cn (P.D.)

* Correspondence: fabio.dellacqua@unipv.it; Tel.: +39-0382-985664

† 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, and policy-making. Combining data from different sensors, such as High-Resolution multispectral Images (HRI) and light detection and ranging (LiDAR) data, has shown great potential in building extraction. Deep learning (DL) is increasingly used in multi-modal data fusion and urban object extraction. However, DL-based multi-modal fusion networks may under-perform due to insufficient learning of "joint features" from multiple sources and oversimplified approaches to fusing multi-modal features. Recently, a 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 over-parametrization, 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

Citation: Ferrari, L.; Dell'Acqua, F.; Zhang, P.; Du, P. Integrating EfficientNet into an HAFNet Structure for Building Mapping in High-Resolution Optical Earth Observation Data. *Remote Sens.* **2021**, *13*, 4361. https://doi.org/10.3390/rs13214361

Academic Editors: Antonio Robles-Kelly, Ajmal Mian, Jun Zhou, Naveed Akhtar, Pedram Ghamisi and Tat-Jun Chin

Received: 18 August 2021
Accepted: 25 October 2021
Published: 29 October 2021

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, and 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. Therefore, it is 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 Multi-modal 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.

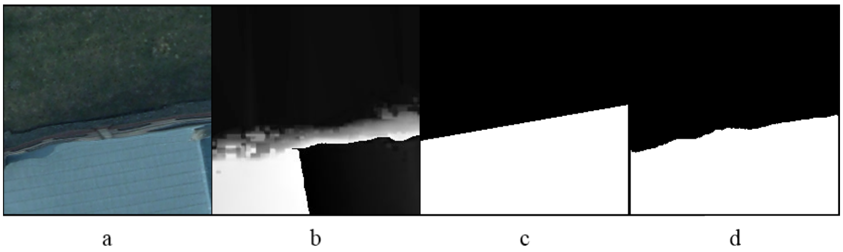

Figure 1. Harmful information in input data. (**a**) RGB patch containing discriminative information. (**b**) DSM patch containing incorrect information. (**c**) Ground truth map. (**d**) Segmentation result. The Att-MFBlock re-weights the RGB and the DSM input so that RGB information is highlighted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

As pointed out by researchers in Reference [9], the reason why AI models are still confined to offline data processing is that their weights and topology are often too large to fit into the available resources onboard Earth Observation satellites. At the same time, inference of DL systems is computationally intensive, and this can be a problem in a low-power-budget environment. New networks need to be engineered taking into account the different computation restrictions related to memory usage, training, and inference time cost. On-board data processing in spaceborne Earth Observation systems is gaining relevance, and methods for different Remote Sensing applications are being developed [9–13]. This trend is substantially accelerated by the recent joint effort of multiple Deep Learning research studies 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 fulfill this task.

A careful reorganization of existing architectures and introduction of efficient modules can solve the previously described problems and accelerate the transformation of AI-driven systems from offline processing tools to powerful dynamic edge applications. 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-based building extraction performances while, at the same time, affording a 92% reduction from the original number of network parameters. This substantial cut in requirements makes it possible to directly deploy the model as an on-board spaceborne urban mapping system.

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. EfficientNet has been used in applications from different domains. Although some of such domains were completely unrelated to Earth Observation (e.g., path prediction in autonomous driving, image classification in the mobile framework), a handful of researchers started also using this family of models for Remote Sensing. Because of their efficiency and capability of extracting highly discriminative features, EfficientNet models have been widely employed as Remote Sensing scene classifiers [17–19]. For example, in Bazi et al. [17], Lasloum et al. [19], EfficientNet-B3 networks are used for scene classification. Alhichri et al. [18] enriched the EfficientNet-B3-based model by adding an Attention module to further increase the classification performance. Salas et al. [20] used EfficientNet-B3 to map satellite images to census data in order to characterize vulnerable communities at the residential block level and, therefore, localize poor areas where poverty reduction policies can be implemented. According to the published papers, no instance of EfficientNet used as encoder for a segmentation model has yet been proposed.

2.2. Attention-Aware Multi-modal Fusion Block

The Attention-Aware Multi-modal Fusion Block is a computational module introduced in Reference [8] to adaptively re-weight feature channels from different modalities, therefore highlighting discriminative features and suppressing irrelevant ones. The module is based on the Attention mechanism [21] 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. The Attention mechanism has been extensively used in Remote Sensing applications; however, there exist only a small number of scenarios where the Attention block has been used as a way to fuse the features extracted from models' encoders [22–25]. Zheng et al. [22] developed a multilevel attention mechanism through adversarial learning to detect oil palm trees. Cai and Wei [23] created a new method to fuse hyperspectral images with attention. Huang et al. [24] used a attention-based fusion block to better detect different remote sensing objects. Shi et al. [25] introduced a multilevel features fusion method with attention to improve the segmentation accuracy of pixels near object boundaries. As shown, something similar to the previously proposed Attention-Aware Multi-modal Fusion Block is presented; however, major differences exist between the proposed solutions, and only the core idea of fusing and enhancing features with attention is preserved.

3. HAFNet and HAFNetE

In this section, we introduce HAFNetE, an efficient hybrid attention-aware fusion network for building extraction, starting from its predecessor HAFNet or Hybrid Attention-aware Fusion Network. HAFNet is a multi-modal building extraction segmentation network that utilizes cross-modal and individual features to perform builiding footprint extraction, and it accepts HRI RGB images and LiDAR data as its inputs. The overall architecture is comprised of three streams: RGB, DSM, and cross-modal. All the streams are built as parallel SegNets [26], where the encoder part is characterized by a VGG-16 structure. The RGB and DSM streams are designed to learn individual modal features. These features are then fused together after each set of convolutional operations with an Attention-Aware Multi-modal Fusion Block (Att-MFBlock) in the cross-modal stream. The extracted features from each stream are decoded in their respective decoder stream and finally combined at the decision stage using again an Attention-Aware Multi-modal Fusion Block 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. Starting from this existing scheme, HAFNetE preserves the three-stream network concept but utilizes both a completely different single stream architecture and encoder structure. The model architecture is shown in Figure 2.

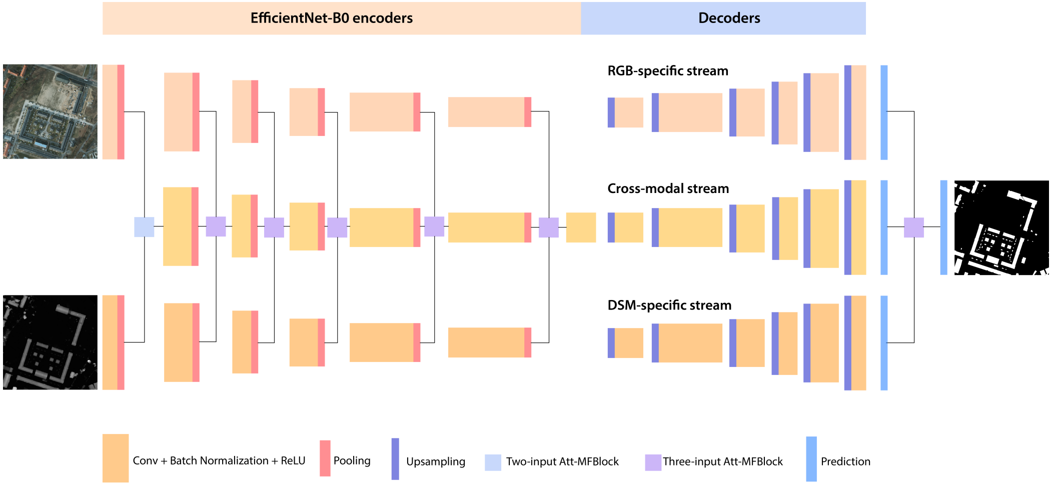

Figure 2. Scheme of the HAFNetE network.

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 [27]. 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, such as ResNet-50 [28]. 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 [29]: EfficientNet models [16] versus classical models.

Model	Top-5 Acc	#Params	#FLOPs
EfficientNet-B0 [16]	93.3%	5.3 M	0.39 B
EfficientNet-B2 [16]	94.9%	9.2 M	1.0 B
EfficientNet-B4 [16]	96.4%	19 M	4.2 B
VGG-16 [30]	91.9%	138 M	19.6 B
ResNet-50 [28]	93.0%	26 M	4.1 B
SENet [21]	96.2%	146 M	42 B

At the individual stream level, the SegNet structure is substituted with a U-Net network [31]. U-Net 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 U-Net 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 U-Net. For these reasons, U-Net was selected as the single-stream subnetwork.

To summarize, HAFNetE is a complete overhaul of the original HAFNet model. VGG16 encoders are substituted with EfficientNet encoders, and the SegNet architecture at the individual stream level is replaced with a U-Net. The only aspects retained from the previous version are the idea of combining features extracted in the HRI-RGB and LiDAR-derived DSM streams into a new cross-modal stream and the method used to fuse the encoded information. The substituted encoders and the restructured network architecture provide a completely new and, most importantly, efficient way of extracting and processing information from data. As it will be discussed thoroughly in Section 5, even though the HAFNetE model provides an improvement at an application level in terms of segmentation capability, the most remarkable and actionable result with respect to the previously proposed HAFNet is the advanced and carefully designed, efficient architecture, that translates into a massive enhancement of computational efficiency.

A part from a few models, most of the newly proposed networks are designed to score highest in segmentation performances largely disregarding the associated computational cost. This latter can make the model impossible to use in most of real-world scenarios, where end users do not have enough computational resources, or, even if they do, the final application does not permit the use of related technologies (e.g., on-board spaceborne systems). Memory footprint, training time, and inference time are aspects that cannot be overlooked when deploying a system in production. HAFNetE is engineered taking all these details into account and with the explicit goal of making the network deployable in a on-board spaceborne system.

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 Labeling Challenge [32], in the German city of Potsdam, and it is composed of high-resolution true-color orthophoto images and the corresponding normalized DSM data. The dataset also includes a smaller dataset on the German city of Vaihingen, but this part has not been included in our experiments. As it will be explained later in this paper, in terms of ortophotos, the Vaihingen dataset contributes false-color IRRG images only, whose radiometric behavior does not match what was learnt on RGB images by the pre-trained networks used in the proposed method.

In the original dataset, each parcel of land was classified into six common land cover classes, and this classification is distributed as Ground Truth (GT) to support the supervised learning procedure. The problem addressed in this paper, i.e., basic building mapping, only uses two labels, namely "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.

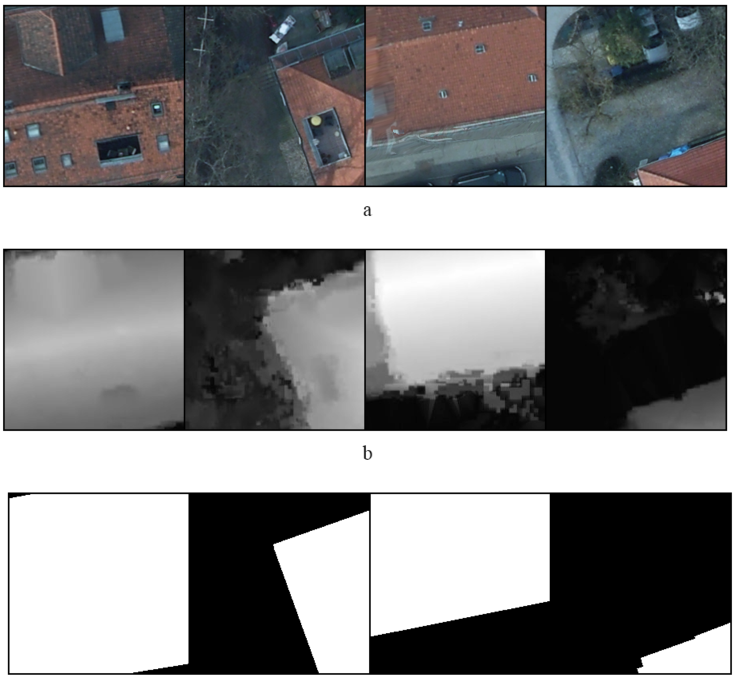

Figure 3. (**a**) RGB image patch. (**b**) DSM patch. (**c**) Corresponding binary thematic map. Building pixels are displayed in white, whereas non-building pixels are displayed in black.

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 for 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 ≈80%, validation ≈10%, and test ≈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.

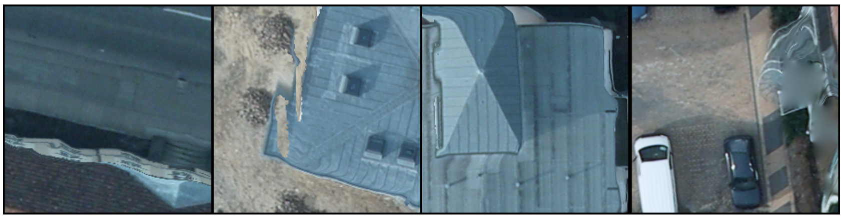

Figure 4. Example of visible distortion in RGB input images.

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. The model was trained using a subsection of the Potsdam dataset. The True OrtoPhoto (TOP) in such dataset come as TIFF files in different channel compositions, namely IRRG, RGB, and RGBIR. Since the model was initialized with pre-trained EfficientNet-B0 weights tuned on RGB-coded images, the RGB version of the TOP images offered in the Potsdam dataset was used. On the other hand, the Vaihingen dataset provides only IRRG TOP images; because of this mismatch, only the Potsdam section of the ISPRS 2D Semantic Labeling Challenge was used to train, validate, and test the model. It should be noted that, in any case, the Potsdam dataset contains most of the images of the entire ISPRS dataset, and, because of its dimensions in terms of number of images and single image size, the data covers a great range of variability and diverse edge cases that make the sole Potsdam section suitable for the standard training, validation, and testing Deep Learning model procedure.

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; thus, they were partitioned into multiple non-overlapping 224 × 224 tiles. This latter is the size of images in the ImageNet dataset [29] 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. Thus, a data-balancing strategy is 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 (e.g., 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 [33] 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. In the end, the number of effective training patches was 8800.

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) [34]. 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 the optimizers, 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.

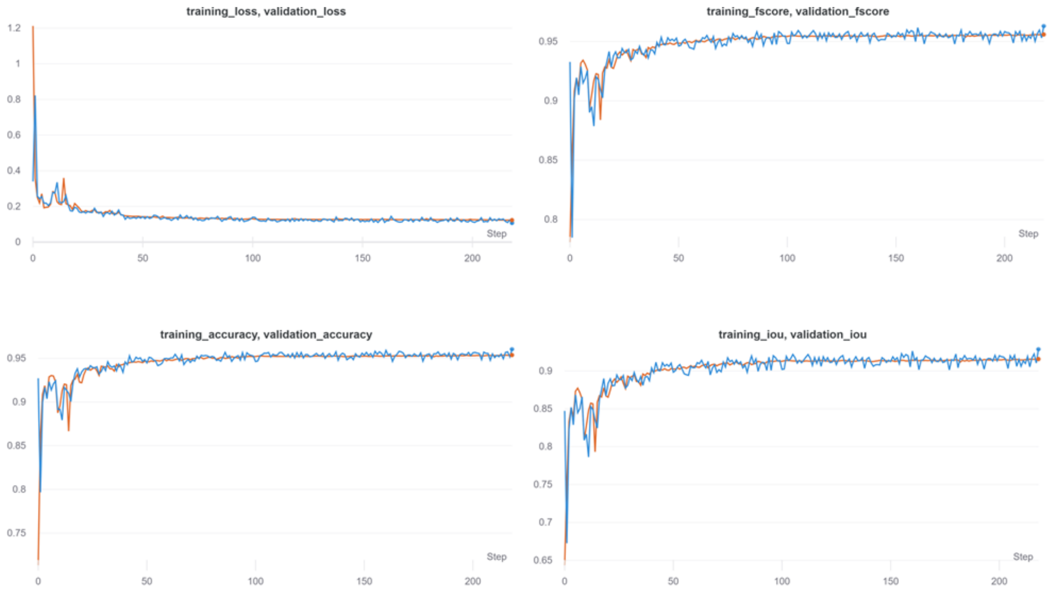

Figure 5. Training (blue) and validation (orange) curves obtained using the Adam optimizer. From **top-left**, clockwise, the four graphs represent the measures of loss, fscore, IoU, and accuracy, respectively.

As stated earlier, the model encoders were initialized with the pre-trained EfficientNet-B0 weights, so a small learning rate $lr = 1 \times 10^{-3}$ was used to optimize loss. The learning rate was modulated using different learning rate schedulation strategies, including Cosine Annealing Warm Restart and Multi-step 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 min/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. Discussion of Results

In this section, we show the results of the HAFNetE model presented in Section 3 trained according to the procedure illustrated in Section 4.3, discuss its features, and highlight the advancements it permits.

5.1. Segmentation Performance 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 U-Net 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. To give the reader a visual sense of typical results from the proposed method, Figure 6 shows the final classification results on a set of test patches. Figure 7 shows, instead, the classification results on a larger scale, providing examples on two entire sample tiles.

Although the biggest advancement from the previous model can be measured in terms of computational efficiency, a segmentation performance improvement can be noticed thanks to the fine-tuning procedure that further enhanced the model's segmentation capabilities, raising the F1-score to 96.68% and IoU to 93.64%. Refer to Table 3 for further details. For the reader's convenience, F1-scores for other state-of-the-art methods on the Potsdam dataset (building) are presented in Table 4.

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.

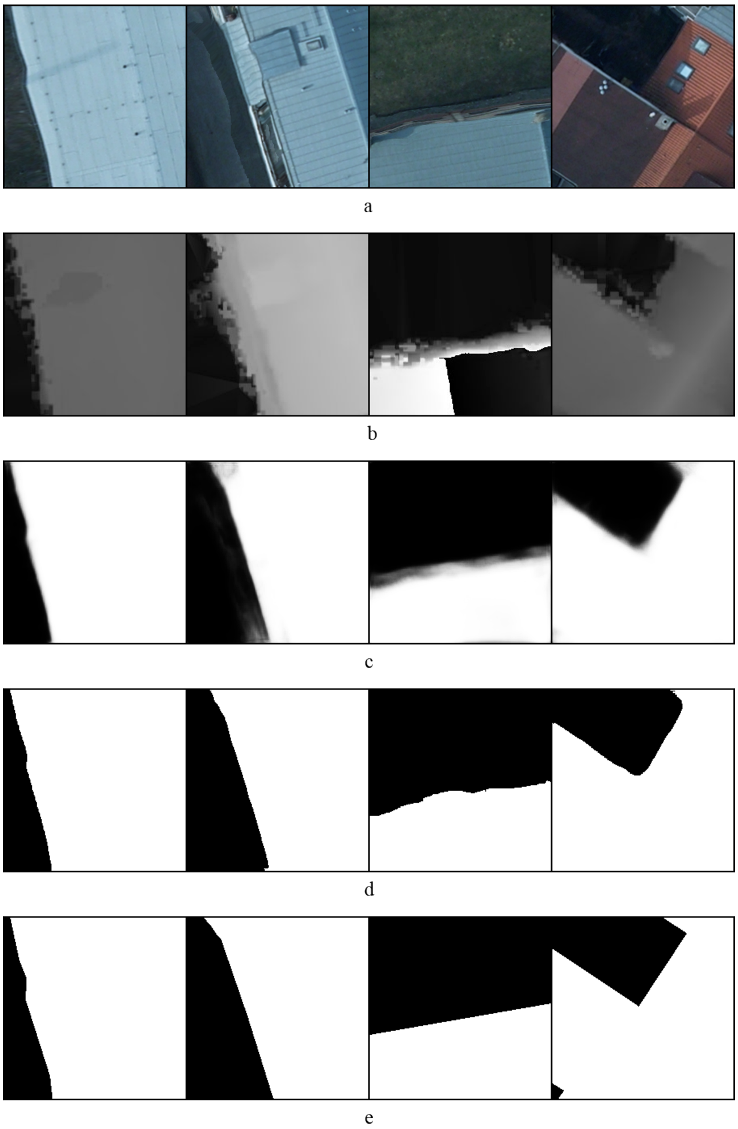

Figure 6. (**a**) Input RGB patches. (**b**) Input DSM patches. (**c**) Model soft predictions. (**d**) Thresholded predictions. (**e**) Label patches. Please note that the corrupted DSM input (**b**) is adaptively re-weighted by the Att-MFBlock, thus suppressing misleading information. Thanks to this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

5.2. Novelties Introduced

As discussed in Sections 1 and 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 5 shows a performance comparison between the HAFNetE and the HAFNet model.

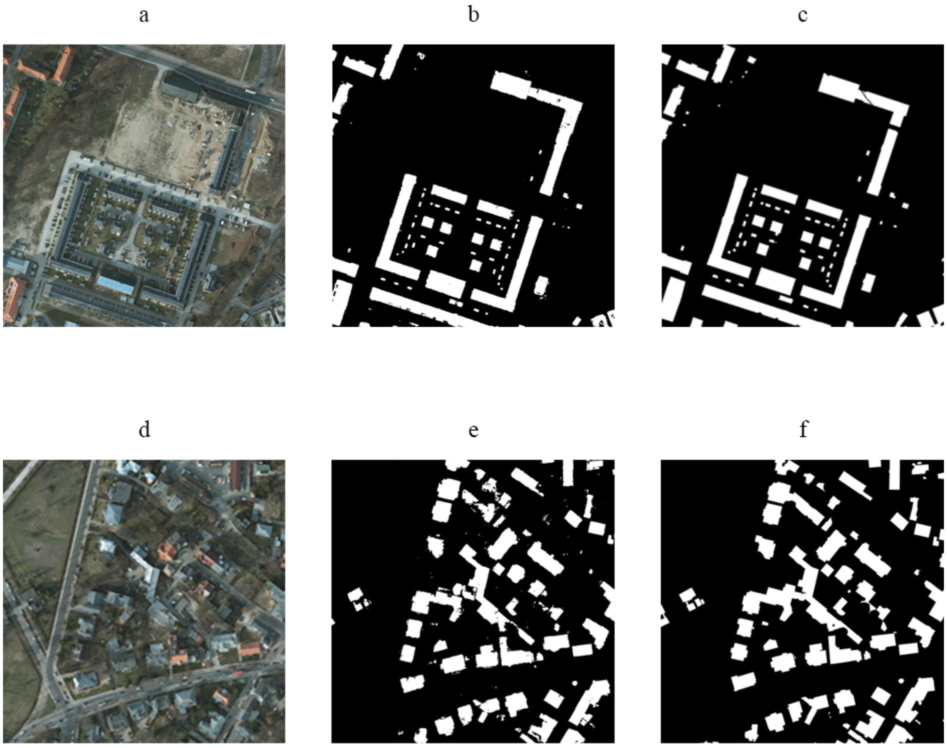

Figure 7. Two examples of results obtained on two entire 6000 × 6000 pixel tiles from the Potsdam dataset. (**a**) RGB TOP image of Tile 1. (**b**) HAFNetE classification result on Tile 1. (**c**) Ground Truth map for Tile 1. (**d**) RGB TOP image of Tile 2. (**e**) HAFNetE classification result on Tile 2. (**f**) Ground Truth map for Tile 2.

Table 4. F1-scores for state-of-the-art methods on the Potsdam dataset (building).

Method	F1 Score
DeepLab v3 + [35]	95.8%
MANet [36]	95.91%
DSMFNet [37]	96.0%
DP-DCN [38]	95.36%
REMSNet [39]	96.17%
MMAFNet [40]	96.26%
HAFNetE	96.68%

Table 5. 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 88.978 M to 6.982 M. 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:

  $$\text{HAFNet: } 88.978\,(Millionsofparameters) \approx 360\,\text{MB}$$

  vs.

  $$\text{HAFNetE: } 6.982 \times 16\text{bits}/8 \approx 14\,\text{MB}.$$

  The used memory can be further compressed to a 8-bit fixed point systolic array in order to make the model directly deployable to dedicated AI platforms, such as Intel's Myriad 2 or Google's Google Coral 28-nm Tensor Processing Unit (TPU) that features 8 MB of on board memory. 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 for on-board processing in spaceborne Earth observation platforms.

As we could assess from the recorded metrics, the HAFNetE model can reach state-of-the-art classification performance. However, the most noticeable and relevant advancement from the previously proposed HAFNet model is the efficiency of the overall network. As described in Reference [9], EO Deep Learning applications are currently relegated to offline processing because models are not properly designed for operating at the edge. In most of the cases, model topology and effective number of parameters are too large to comply with satellites memory and power consumption requirements and that strongly limits the impact that Deep Learning can give to Earth Observation systems. HAFNetE has been engineered taking into account all these requirements and with a deployment-oriented approach. Classical models often disregard memory and computing limitations and, therefore, generally end up not being suitable for deployment as on-board spaceborne systems. HAFNetE represents an example of what DL can provide as an effective tool in real-world EO applications that can work directly on satellites and, consequently, empower new industrial possibilities.

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 Labeling Challenge [32]. 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, 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, such as, for example, EfficientNetV2 [41], which has just been released.

Author Contributions: Conceptualization, L.F., F.D. and P.D.; methodology, L.F.; software, L.F.; validation, L.F.; formal analysis, L.F. and F.D.; investigation, L.F. and F.D.; resources, P.D., L.F. and

P.Z.; data curation, L.F.; writing—original draft preparation, L.F.; writing—review and editing, L.F. and F.D; supervision, F.D. and P.Z. All authors have read and agreed to the published version of the manuscript.

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/ (accessed on 15 May 2021).

Acknowledgments: The authors wish to thank the University of Pavia for the "Internationalization@home" initiative for remote lecturing, which triggered this cooperation.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
Att-MFBlock	Attention-Aware Multi-modal Fusion Block
CNN	convolutional neural network
DSM	Digital Surface Model
EO	Earth Observation
GT	Ground Truth
HAFNet	Hybrid Attention-aware Fusion Network
HAFNetE	HAFNet with EfficientNet
HRI	High-resolution Remote sensing Imagery
IoU	Intersection over Union
SGD	Stochastic Gradient Descent
LiDAR	Light Detection and Ranging
OA	overall accuracy
SMP	Segmentation Models PyTorch,

1. Audebert, N.; Le Saux, B.; Lefèvre, S. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS J. Photogramm. Remote Sens.* **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 J. Photogramm. Remote Sens.* **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. In Proceedings of the IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 3852–3855.
4. Hazirbas, C.; Ma, L.; Domokos, C.; Cremers, D. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 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 Sens.* **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 J. Photogramm. Remote Sens.* **2018**, *135*, 158–172.
7. Marcos, D.; Hamid, R.; Tuia, D. Geospatial correspondences for multimodal registration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 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 Sens.* **2020**, *12*, 3764, 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 Aerosp. Electron. Syst. Mag.* **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. In Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications, Austin, TX, USA, 3–4 March 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. *Sci. Rep.* **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 Sens.* **2020**, *12*, 2205, 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. *Eng. Appl. Artif. Intell.* **2020**, *96*, 103952, doi: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* **2017**, arXiv:1704.04861.
15. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
16. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
17. Bazi, Y.; Al Rahhal, M.M.; Alhichri, H.; Alajlan, N. Simple Yet Effective Fine-Tuning of Deep CNNs Using an Auxiliary Classification Loss for Remote Sensing Scene Classification. *Remote Sens.* **2019**, *11*, 2908, doi:10.3390/rs11242908.
18. Alhichri, H.; Alswayed, A.S.; Bazi, Y.; Ammour, N.; Alajlan, N.A. Classification of Remote Sensing Images Using EfficientNet-B3 CNN Model With Attention. *IEEE Access* **2021**, *9*, 14078–14094, doi:10.1109/ACCESS.2021.3051085.
19. Lasloum, T.; Alhichri, H.; Bazi, Y.; Alajlan, N. SSDAN: Multi-Source Semi-Supervised Domain Adaptation Network for Remote Sensing Scene Classification. *Remote Sens.* **2021**, *13*, 3861, doi:10.3390/rs13193861.
20. Salas, J.; Vera, P.; Zea-Ortiz, M.; Villaseñor, E.A.; Pulido, D.; Figueroa, A. Fine-Grained Large-Scale Vulnerable Communities Mapping via Satellite Imagery and Population Census Using Deep Learning. *Remote Sens.* **2021**, *13*, 3603, doi:10.3390/rs13183603.
21. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
22. Zheng, J.; Fu, H.; Li, W.; Wu, W.; Zhao, Y.; Dong, R.; Yu, L. Cross-regional oil palm tree counting and detection via a multi-level attention domain adaptation network. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 154–177, doi:10.1016/j.isprsjprs.2020.07.002.
23. Cai, W.; Wei, Z. Remote Sensing Image Classification Based on a Cross-Attention Mechanism and Graph Convolution. *IEEE Geosci. Remote Sens. Lett.* **2020**, 1–5, doi:10.1109/LGRS.2020.3026587.
24. Huang, X.; He, B.; Tong, M.; Wang, D.; He, C. Few-Shot Object Detection on Remote Sensing Images via Shared Attention Module and Balanced Fine-Tuning Strategy. *Remote Sens.* **2021**, *13*, 3816, doi:10.3390/rs13193816.
25. Shi, H.; Fan, J.; Wang, Y.; Chen, L. Dual Attention Feature Fusion and Adaptive Context for Accurate Segmentation of Very High-Resolution Remote Sensing Images. *Remote Sens.* **2021**, *13*, 3715, doi:10.3390/rs13183715.
26. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495.
27. Chen, H.; Li, Y. Three-stream attention-aware network for RGB-D salient object detection. *IEEE Trans. Image Process.* **2019**, *28*, 2825–2835.
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
29. ImageNet. Available online: https://image-net.org/index.php (accessed on 10 May 2021).
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
31. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
32. ISPRS 2D Semantic Labeling Contest. Available online: https://www2.isprs.org/commissions/comm2/wg4/benchmark/semantic-labeling/ (accessed on 10 May 2021).
33. Xia, X.; Lu, Q.; Gu, X. Exploring An Easy Way for Imbalanced Data Sets in Semantic Image Segmentation. *J. Phys. Conf. Ser.* **2019**, *1213*, 022003, doi:10.1088/1742-6596/1213/2/022003.
34. Yakubovskiy, P. Segmentation Models Pytorch. 2020. Available online: https://github.com/qubvel/segmentation_models.pytorch (accessed on 10 May 2021).
35. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
36. Shang, R.; Zhang, J.; Jiao, L.; Li, Y.; Marturi, N.; Stolkin, R. Multi-scale adaptive feature fusion network for semantic segmentation in remote sensing images. *Remote Sens.* **2020**, *12*, 872.
37. Cao, Z.; Fu, K.; Lu, X.; Diao, W.; Sun, H.; Yan, M.; Yu, H.; Sun, X. End-to-end DSM fusion networks for semantic segmentation in high-resolution aerial images. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1766–1770.
38. Peng, C.; Li, Y.; Jiao, L.; Chen, Y.; Shang, R. Densely based multi-scale and multi-modal fully convolutional networks for high-resolution remote-sensing image semantic segmentation. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 2612–2626.
39. Liu, C.; Zeng, D.; Wu, H.; Wang, Y.; Jia, S.; Xin, L. Urban land cover classification of high-resolution aerial imagery using a relation-enhanced multiscale convolutional network. *Remote Sens.* **2020**, *12*, 311.
40. Lei, T.; Li, L.; Lv, Z.; Zhu, M.; Du, X.; Nandi, A.K. Multi-Modality and Multi-Scale Attention Fusion Network for Land Cover Classification from VHR Remote Sensing Images. *Remote Sens.* **2021**, *13*, 3771.
41. Tan, M.; Le, Q.V. Efficientnetv2: Smaller models and faster training. *arXiv* **2021**, arXiv:2104.00298.