More efficient Geospatial ML modelling techniques for identifying man-made features in Aerial Ortho-imagery

Samuele Buosi, Shubham Sonarghare, John McDonald and Tim McCarthy

Maynooth University

Abstract

Deep learning techniques are used to achieve state-of-art accuracy in semantic segmentation on aerial ortho-imagery datasets. These algorithms are known to be efficient in terms of accuracy but at the expense of computational power required for training and subsequent inference operations. In this paper we strive to achieve a comparable performance but with lower floating point operations per second (FLOPS) and less training time. With this in mind, we chose to evaluate the EfficientNet-B0 network configured with 5.3 millions parameters and 0.39 billion FLOPS as a feature extractor operating inside a U-net architecture, achieving accuracy levels (mean F1 score of 0.869) comparable to a state-of-the-art deep learning architecture (U-net with Resnet50 as backbone) configured with 25.6 million parameters and 4.1 billion FLOPS which achieved a mean F1 score of 0.87. These promising results demonstrate that employing EfficientNet as the feature extractor in semantic segmentation on aerial ortho-imagery can be an effective strategy, in achieving higher performance results in terms of computational power, especially when running these networks on the edge.

Keywords: Deep Learning, Supervised Image Segmentation, semantic segmentation, ortho-imagery, Deep convolutional neural network

1 Introduction

Over the past decade, advances in Machine Learning (ML) and in particular Deep Learning (DL) algorithms have resulted in significant advances in Computer Vision. One of the key applications is Semantic Segmentation which is used in a number of applications including; Robotic Localisation, Autonomous Driving, Scene Understanding and, building High-Definition Maps [Kemker et al., 2018].

In terms of geospatial applications, unmanned aerial vehicles (UAVs) are playing an increasing role in data gathering and mapping our real world environments. These robotic aerial data gathering platforms are now commonly found across the globe, collecting large volumes of data that require automated processing such as feature extraction to be carried out on the fly. Such requirement demands both computationally inexpensive and high accuracy feature extraction techniques [Ammour et al., 2017].

Most common and well-known traditional techniques in computer vision like Support Vector Machines, [Waske and Benediktsson, 2007], and Random Forests, [Pal, 2005], often result in less accurate outputs compared to the DL techniques that produce significantly improved accuracy but at the expense of resources required to train and carry out subsequent inference [O'Mahony et al., 2020]. In this paper we investigate the potential for EfficientNet family, [Tan and Le, 2019], to help reduce this expense in extracting man-made features in UAV aerial imagery. We investigate this hypothesis using an U-Net architecture, [Ronneberger et al., 2015], with an EfficientNet-B0, [Tan and Le, 2019] feature extractor. To assess the performance of the resulting architecture we utilise the International Society for Photogrammetry and Remote Sensing (ISPRS) benchmark dataset [ISPRS, 2016].

2 Prior Work

Recent developments in aerial robotic data gathering platforms, such as UAVs, now enable the rapid capture of aerial imagery at higher spatial-temporal resolutions as well as lower costs. In parallel, emerging developments in

contemporary DL algorithms in automating the data processing and feature extraction has resulted in new data products and information services for applications including; urban planning, land cover classification, Emergency Response, etc. [Ammour et al. 2017].

It is possible to generate an orthophoto from overlapping aerial imagery that is geometrically corrected (orthorectified) so it can be used to measure true distances and dimensions. The process of orthorectification enables various real-world phenomena and distortion such as topographic relief, lens distortion and camera orientation to be corrected [Habib et al., 2007].

Semantic Segmentation is an important algorithm that can assign a class to each pixel of a given image where the classes are defined *A Priori*. Semantic Segmentation applied to ortho-imagery is very useful and important because of its ability to detect and categorise one or more classes in the ortho-image [Liu et al., 2018]. Traditional image segmentation methods include; Watershed, Graph Cuts and Random Forests which have been used to classify high-resolution aerial images [Meyer and Beucher, 1990; Boykov and Jolly, 2001; and Pal, 2005]. However, DL techniques involving convolutional neural networks have proven to be more efficient and effective in extracting features from images compared to these more traditional approaches [Deng et al., 2009]. DL methods perform well even for semantic segmentation due to their ability to automatically extract features. For example, in 2015, there was a 20% relative improvement to 62.2% mean Intersection over Union (IoU) using a Fully Convolution Networks (FCNs based on the PASCAL VOC 2012 benchmark dataset compared to the state-of–the-art techniques of that time [Long et al., 2015].

There are many Neural Network architectures that utilise CNNs for semantic segmentation tasks e.g., U-net [Ronneberger et al., 2015], LinkNet [Chaurasia and Culurciello, 2017], Feature Pyramid Networks [Li et al., 2019]. As an example, [Wu et al., 2018] uses U-net [Ronneberger et al., 2015] for automatically segmenting building features from aerial imagery. Similarly, [Boonpook, et al., 2018] uses SegNet [Vijay et al., 2016] to extract building features from UAV images for riverbank monitoring. One of the novelties of these architectures is their compatibility and adaptability with a range of feature extractors. For example, one can use VGG [Simonyan and Zisserman, 2015] as the feature extractor in a U-net architecture [Ronneberger et al., 2015] or use ResNet [He et al., 2016] inside a LinkNet [Chaurasia and Culurciello, 2017]. The performance of these networks completely depends on the performance of the feature extractor in combination with how the architecture combines these features to segment the objects under observation. More recently, ScasNet [Liu et al., 2018] which utilized Resnet [He et al., 2016] as a feature extractor, achieved one of the best results with an overall accuracy of 91.1% on the ISPRS Potsdam benchmark dataset [Liu et al., 2018]. With a more complex feature extractor is it possible to achieve higher performance with respect to accuracy in resulting object segmentation, but this also increases the number of parameters to train. This gives rise to computationally more expensive requirements since high-performing DL techniques require relatively large volumes of training data to train models with a high number of parameters.

In this paper, we investigate the potential for a more efficient and scalable Semantic Segmentation Neural Network architecture that allows a comparable level of performance to be achieved similar to the actual state-of-art applied to ortho-imagery from the ISPRS Potsdam benchmark dataset. To this end, we employ a combination of a U-net architecture [Ronneberger et al., 2015], an EfficientNet [Tan and Le, 2019] feature extractor and focal/dice loss [Lin et al., 2020, Deng et al., 2018].

3 Technical Description

The main drawback of the majority of CNNs are their tendency to down-scale or reduce the spatial resolution of the features along the depth of the network which is not ideal in a segmentation context.

To overcome down-sampling of the spatial resolution, many Fully Convolutional Neural Networks have been suggested like Segnet [Vijay et al., 2016], U-net [Ronneberger et al., 2015]. We chose a U-net architecture with an EfficientNet-b0 as the feature extractor, after an initial assessment based on literature review, for this study.

The U-Net architecture is a CNN widely used for Semantic Segmentation. The original network consists of an encoder path and a decoder path that gives the U-shaped architecture. The Encoder part is composed by repeated convolution layers, each followed by a rectified linear unit (ReLU) layer and a maximum pooling layer. The decoder part is composed by sequence of up-convolutions and concatenations.

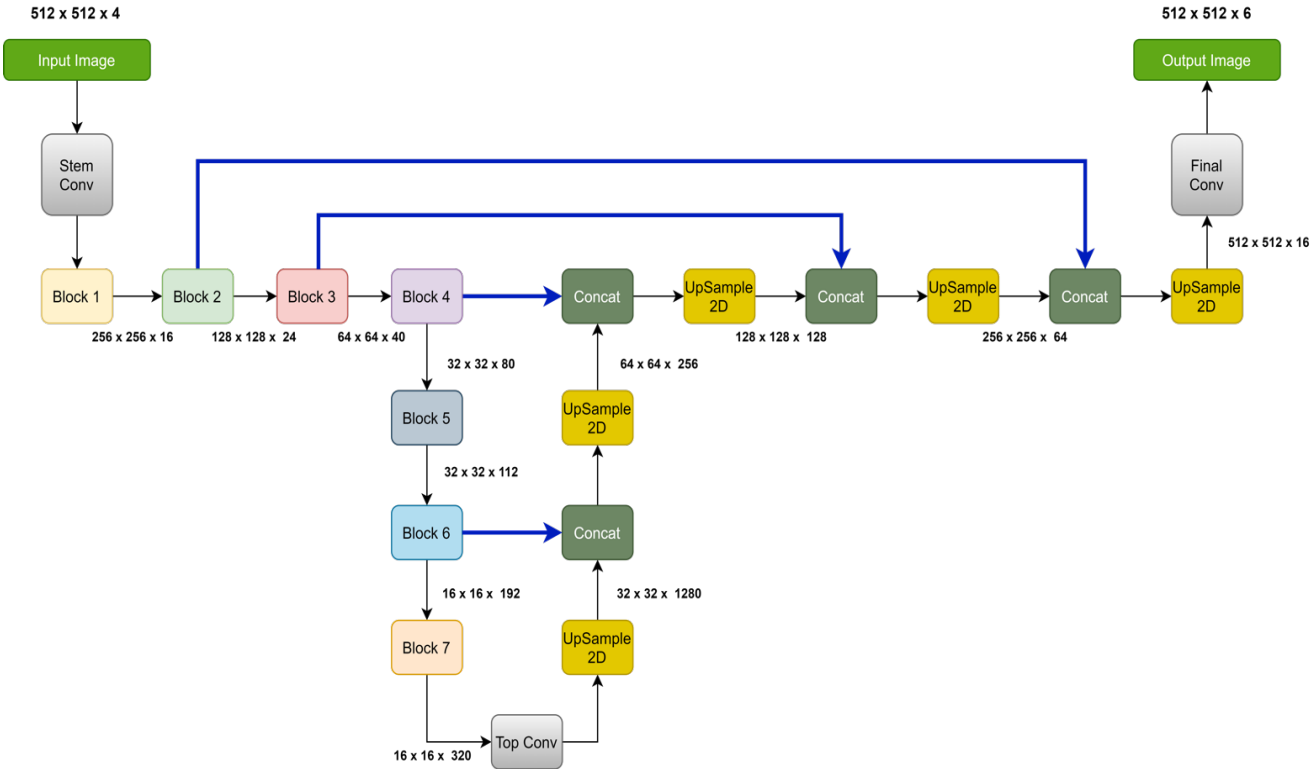

Figure 1: Overall U-net architecture using EfficientNet-b0

The U-net architecture readapted with the EfficientNet-B0 as the encoder is detailed in Figure 1. The EfficientNet is a family of Convolutional Neural Networks developed in the context of AutoML where the authors have investigated a possible solution for neural network (NN) scaling for efficiency [Tan and Le, 2019]. Tan and Le, [Tan and Le, 2019], created a first baseline EfficientNet-B0 inspired by a MnasNet and scaled up to the B7 network using their new compound scaling method, optimizing both accuracy and FLOPS at the same time. As a result, the network is faster and smaller compared to the other major networks used based on the ImageNet benchmark dataset [Tan and Le, 2019]. Specifically, EfficentNet-B0 uses 4.9 times less parameters and 11 times less FLOPS compared to ResNet-50 while providing 77.1 % as Top-1 accuracy on ImageNet compared to 76.1% of ResNet-50 [He et al., 2016]. Figure 2 shows the EfficientNet-B0 architecture.

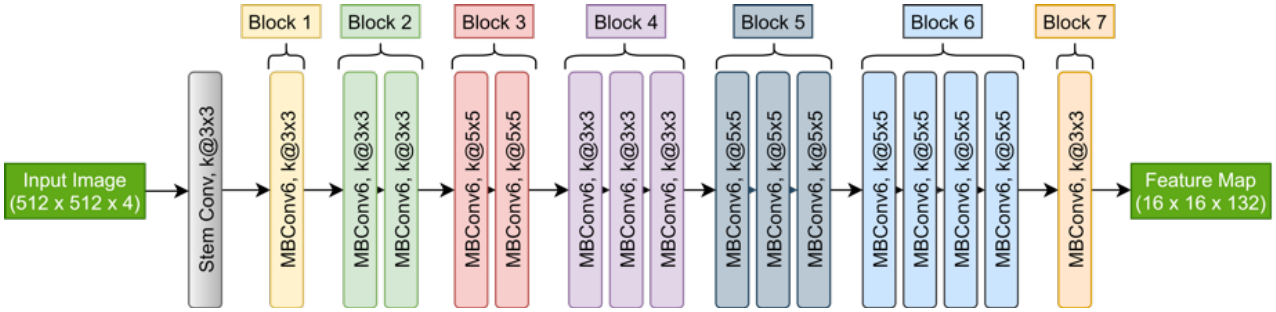

Figure 2: Architecture of EfficientNet-B0 as feature extractor

Along with the architecture, it also important to carefully select the loss function which will penalize the network for incorrect predictions and detections. Standard cross-entropy loss is calculated as the average of per-pixel loss. This poses a huge issue when the number of foreground pixels are far less than the number of background pixels. Although, weighted cross entropy loss helps alleviate this problem, it does not result in a significant improvement. To overcome this issue, we used a combination of a focal and dice loss. While the focal loss helps in learning hard negative examples and addresses the issue of class imbalance, dice loss helps to learn better class boundaries [Lin et al., 2020, Deng et al., 2018].

The Dice Loss is defined by

$$TP(c) = \sum_{i=1}^{N} p_i(c) g_i(c)$$

$$FN(c) = \sum_{i=1}^{N} (1 - p_i(c)) g_i(c)$$

$$FP(c) = \sum_{i=1}^{N} p_i(c)(1 - g_i(c))$$

$$\mathcal{L}_{Dice} = C - \sum_{c=0}^{C-1} \frac{2TP(c)}{2TP(c)+FP(c)+FN(c)} \quad (1)$$

where C is the total number of classes, N is the total number of pixels, $p_i(c)$ is the predicted class of the pixel, $g_i(c)$ is the ground truth class of the pixel. TP, FP and FN are respectively the true positives, false positives, false negatives of a particular class. The Focal Loss is defined by

$$\mathcal{L}_{Focal} = -\lambda \frac{1}{N} \sum_{c=0}^{C-1} \sum_{i=1}^{N} g_i(c)(1 - p_i(c))^{\gamma} log(p_i(c)) \quad (2)$$

The focusing parameter γ was set to 2 and the weighting factor λ was set to 0.25 in our experiment. Thus, the total loss is given by,

$$\mathcal{L}_{DF} = \mathcal{L}_{Dice} + \mathcal{L}_{Focal}$$

$$= C - \sum_{c=0}^{C-1} \frac{2TP(c)}{2TP(c)+FP(c)+FN(c)} - \lambda \frac{1}{N} \sum_{c=0}^{C-1} \sum_{i=1}^{N} g_i(c)(1 - p_i(c))^{\gamma} log(p_i(c)) \quad (3)$$

4 Experiments

4.1 Implementation

We implemented U-net architecture using Tensorflow 2.3.1 with CUDA 10.1 support. Training images are read on the fly and randomly augmented using Tensorflow data API. We did our performance tests using a graphics processing unit (GPU) NVIDIA GeForce GTX 1650 with 4 GB of GPU memory.

4.2 Benchmark Dataset

We applied and studied the performance of the architecture described in section 3 with the ISPRS Potsdam benchmark dataset [ISPRS, 2016]. This benchmark dataset contains 38 ortho-images of same size of 6000 x 6000 pixels generated from cropping a larger orthophoto at a ground sampling distance (GSD) of 5 cm. Each ortho-image in the dataset consist of 4 channels IRRGB (Infrared, Red, Green, Blue) and for each ortho-image, there is a corresponding Digital Surface Model (DSM), representing elevation and normalised DSM (nDSM) data. The ground truth labels are also provided for training purposes for 24 of these 38 ortho-images. An example of the dataset is detailed in Figure 3 where a ISPRS RGB patch is overlapped with the ground truth. The ground truth colour map used for ISPRS classes/objects is listed in Table 1.

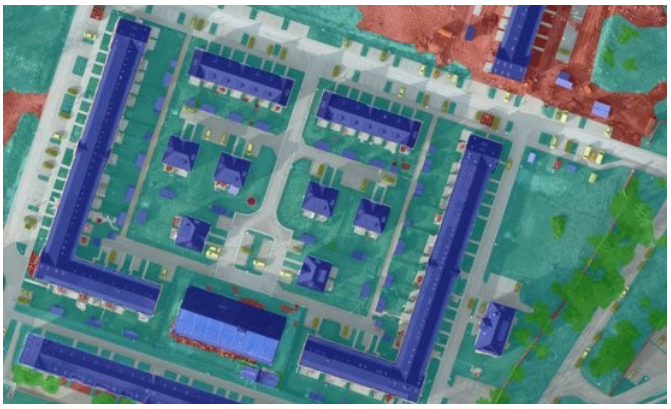

Figure 3: Labels overlapped on a RGB ortho-image crop from ISPRS Potsdam dataset

Colour	Class
White	Impervious Surfaces
Blue	Buildings
Cyan	Low Vegetation
Green	Trees
Yellow	Car
Red	Clutter

Table 1: ISPRS colour and class definition

4.3 Training and Evaluation

For the experiments, we pre-processed the raw ISPRS Potsdam dataset and generated 4681 patches of 512x512 pixels each with the infrared (IR), red (R), green (G), and normalized digital surface model (nDSM) band. Every patch has the correlated mask in a different folder with the same patch name in .tif format. For training, we used an 80/20 split so, 80% of all the 4681 patches was used for train the model and the remaining 20% patches was used for validation purposes. Data was also normalized, and data augmentation was applied, which consisted of random rotation of 90°, vertical and horizontal flips with a probability of 0.5. We choose a batch size of 4 due to our memory constraints. We did not use the Transfer Learning technique because most of the common pre-trained weights are based on RGB images, but in this case, we have 4 channels corresponding to IR, RG and nDSM data. Hence, we initialized the network with Xavier initialization [Glorot and Bengio, 2010]. The initial learning rate (LR) was set to 0.001 with a learning rate scheduler that monitored the validation loss. The LR was set to decrease by a factor of 0.1 every 5 epochs if the validation loss doesn't reduce. The minimum LR was set to 1e-15. The optimizer chosen was Adam [Kingma and Ba, 2015].

We trained two models using the ISPRS Potsdam dataset and created a comparison table (Table 2) with the F1 score metric (2) per class and reporting the number of parameters and FLOPS required. All the models are based on the same U-net architecture but with a different feature extractor. We chose to compare EfficientNet B0 with ResNet50 because these two architectures have comparable performances [Tan and Le, 2019]

We assessed quantitative performance of the two models based on the F1 score applied to all the six classes as,

$$F1 = 2\frac{Precision*Recall}{Precision+Recall} \tag{4}$$

where, Precision and Recall are defined by:

$$Precision = \frac{TruePositives}{TruePositives+FalsePositives} \tag{5}$$

$$Recall = \frac{TruePositives}{TruePositives+FalseNegatives} \tag{6}$$

4.4 Results and Analysis

We generated predictions for each of the fourteen ortho-images contained in the ISPRS Potsdam test dataset and compared to the ground truth calculating the metrics for both architectures. We also produced qualitative results as shown in Figure 4 where we show the IRRG image, the ground truth, the prediction with Resnet50 and the prediction with EfficientNet-B0 based on an ortho-image from the ISPRS Potsdam test dataset.

As seen from Table 2, EffientNet-b0 resulted in almost the same weighted F1 score as ResNet-50 but with 4.9x less parameters and 11x less FLOPS. This resulted in comparable performance when comparing EfficientNet-b0 to ResNet-50 but with significantly less computational overhead.

Architecture	Num. of parameters	FLOPS	Weighted Mean F1	F1-Scores					
				Impervious Surfaces	Buildings	Low Vegetation	Trees	Car	Clutter
U-net + EfficientNet -B0	5.3M	0.39B	0.869	0.89	0.95	0.82	0.83	0.89	0.41
U-net + ResNet50	25.6M	4.1B	0.87	0.89	0.95	0.82	0.82	0.88	0.45

Table 2: Model comparison

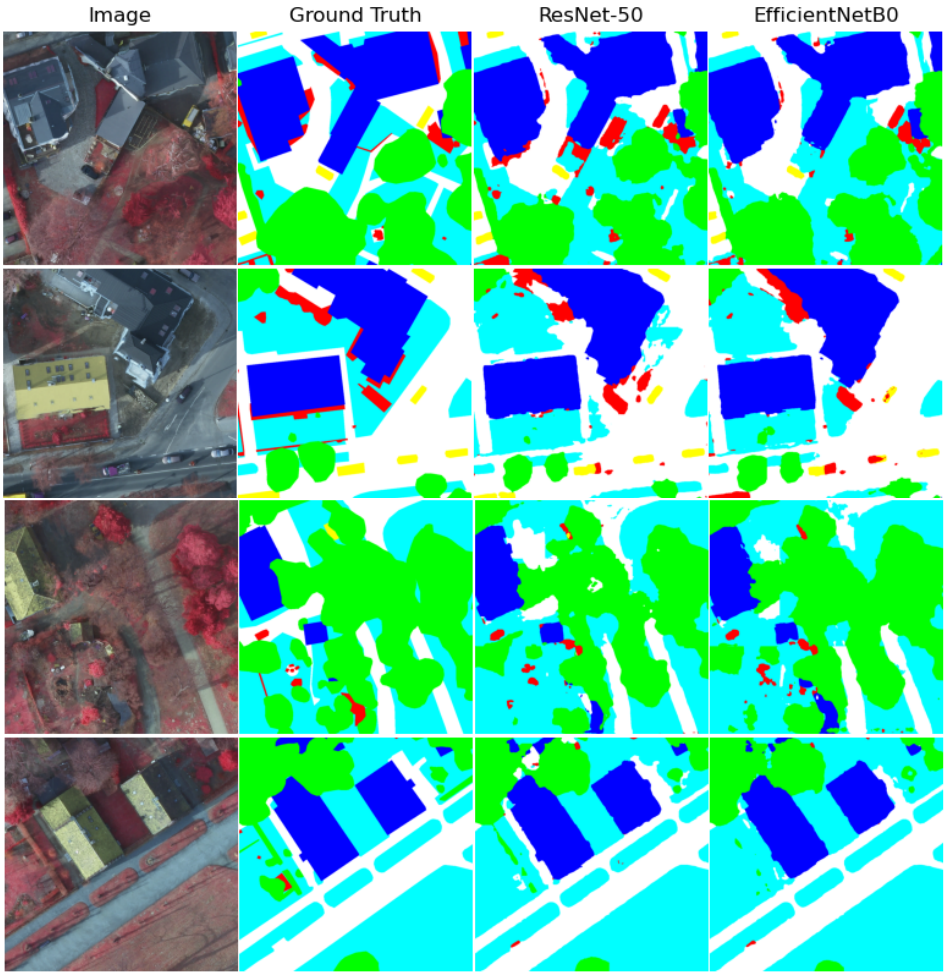

Figure 4: Qualitative comparison side by side of the inference from the models on the ISPRS Potsdam test set.

5 Conclusions

In this paper, we investigated a more efficient Neural Network architecture that can achieve state-of-art performance on Semantic Segmentation applied to ortho-imagery, captured using UAVs. We reviewed a Neural Network based on a U-net architecture but modifying the features extractor with the new EfficientNet-B0. We were not interested in accuracy alone, but also examining the possibility of reducing the computational power required by the common architecture ResNet50. Initial results are promising and scalable. Further experimentation could be conducted on testing and evaluating the robustness and versatility of these architectures using different datasets and comparing the results also with other well-known Semantic Segmentation architectures.

Acknowledgements

This material is based upon works supported by U-Flyte (Unmanned Aircraft Systems Flight Research) 17/SPP/3460 which is funded under the Science Foundation Ireland Strategic Partnership Programme

References

[Ammour et al. 2017] Ammour, N., Alhichri, H., Bazi, Y., Benjdira, B., Alajlan, N., & Zuair, M. (2017). Deep learning approach for car detection in UAV imagery. *Remote Sensing*, *9*(4). https://doi.org/10.3390/rs9040312

[Boykov and Jolly, 2001] Boykov, Y.Y.; Jolly, M.P. Interactive graph cuts for optimal boundary and region segmentation of objects in ND images. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Vancouver, BC, Canada, 7–14 July 2001; pp. 105–112.

[Boonpook, et al., 2018] Boonpook, W., Tan, Y., Ye, Y., Torteeka, P., Torsri, K., & Dong, S. (2018). A deep learning approach on building detection from unmanned aerial vehicle-based images in riverbank monitoring. Sensors (Switzerland), 18(11). https://doi.org/10.3390/s18113921

[Chaurasia and Culurciello, 2017] Chaurasia, A., & Culurciello, E. (2017). LinkNet: Exploiting encoder representations for efficient semantic segmentation. 2017 IEEE Visual Communications and Image Processing, VCIP 2017, 2017-Janua, 1–4. https://doi.org/10.1109/VCIP.2017.8305148

[Deng et al., 2018] Deng R.; Shen C.; Liu S.; Wang H.; Liu X., "Learning to predict crisp boundaries," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 11210 LNCS, pp. 570–586, 2018.

[Deng et al., 2009] Deng J., Dong W., Socher R., Li L., Li K. and L. F.-F. (2009). ImageNet: A Large-Scale Hierarchical Image Database. IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[Glorot and Bengio, 2010] Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks Xavier. AISTATS , volume 9 of JMLR Proceedings, page 249-256. JMLR.org.

[Habib et al., 2007] Habib, A. F., Kim, E. M., & Kim, C. J. (2007). New methodologies for true orthophoto generation. Photogrammetric Engineering and Remote Sensing, 73(1), 25–36. https://doi.org/10.14358/PERS.73.1.25

[He et al., 2016] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem, 770–778. https://doi.org/10.1109/CVPR.2016.90

[Huang et al., 2017] Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.

[ISPRS, 2016] International society for photogrammetry and remote sensing. 2D Semantic Labeling Challenge. Available at: https://www2.isprs.org/commissions/comm2/wg4/benchmark/2d-sem-label-potsdam/

[Kingma and Ba, 2015] Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 1–15.

[Kemker et al., 2018] Kemker, R., Salvaggio, C., & Kanan, C. (2018). Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, *145*, 60–77. https://doi.org/10.1016/j.isprsjprs.2018.04.014

[Li et al., 2019] Li, X., Lai, T., Wang, S., Chen, Q., Yang, C., & Chen, R. (2019). Weighted feature pyramid networks for object detection. Proceedings - 2019 IEEE Intl Conf on Parallel and Distributed Processing with Applications, Big Data and Cloud Computing, Sustainable Computing and Communications, Social Computing and Networking, ISPA/BDCloud/SustainCom/SocialCom 2019, 1500–1504. https://doi.org/10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00217

[Lin et al., 2020] Lin T. Y.; Goyal P.; Girshick R.; He K.; Dollar P., "Focal Loss for Dense Object Detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 318–327, 2020.

[Liu et al., 2018] Liu, Y., Fan, B., Wang, L., Bai, J., Xiang, S., & Pan, C. (2018). Semantic labeling in very high resolution images via a self-cascaded convolutional neural network. ISPRS Journal of Photogrammetry and Remote Sensing, 145, 78–95. https://doi.org/10.1016/j.isprsjprs.2017.12.007

[Long et al., 2015] Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.

[Marmanis et al., 2016] Marmanis D., Datcu M., Esch T., and Stilla U., "Deep Learning Earth Observation Classification using ImageNet Pretrained Networks," IEEE Trans. Geosci. Remote Sens. Lett., vol. 13, no. 1, pp. 105–109, 2016.

[Meyer and Beucher, 1990] Meyer, F.; Beucher, S. Morphological segmentation. J. Vis. Commun. Image R. 1990, 1, 21–46.

[O'Mahony et al., 2020] O'Mahony N.; Campbell S.; Carvalho A.; Harapanahalli S.; Hernandezo G.; Krpalkova L.; Riordan D.; Walsh J., "Deep Learning vs. Traditional Computer Vision," Adv. Intell. Syst. Comput., vol. 943, no. Cv, pp. 128–144, 2020.

[Pal, 2005] Pal, M. Random forest classifier for remote sensing classification. Int. J. Remote Sens. 2005, 26, 217–222.

[Peng et al., 2017] Peng, C.; Zhang, X.; Yu, G.; Luo, G.; Sun, J. Large Kernel Matters—Improve Semantic Segmentation by Global Convolutional Network. arXiv 2017, arXiv:1703.02719.

[Ronneberger et al., 2015] Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv 2015, arXiv:1505.04597.

[Simonyan and Zisserman, 2015] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, 1–14.

[Tan and Le, 2019] Tan M. and Le Q. V., "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." in Proceedings of the 36th International Conference on Machine Learning, 2019.

[Vijay et al., 2016] Vijay, B.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 2017, 39, 2481–2495.

[Waske and Benediktsson, 2007] Waske, B., & Benediktsson, J. A. (2007). Fusion of support vector machines for classification of multisensor data. IEEE Transactions on Geoscience and Remote Sensing, 45(12), 3858–3866. https://doi.org/10.1109/TGRS.2007.898446

[Wu et al., 2018] Wu, G., Shao, X., Guo, Z., Chen, Q., Yuan, W., Shi, X., Xu, Y., & Shibasaki, R. (2018). Automatic building segmentation of aerial imagery usingmulti-constraint fully convolutional networks. Remote Sensing, 10(3), 1–18. https://doi.org/10.3390/rs10030407