mathematics

MDPI

Article

Semantic Segmentation for Aerial Mapping

Gabriel Martinez-Soltero, Alma Y. Alanis, Nancy Arana-Daniel and Carlos Lopez-Franco *

Centro Universitario de Ciencias Exactas e Ingenierías, Universidad de Guadalajara, Blvd. Marcelino García Barragán 1421, Guadalajara C.P. 44430, Jalisco, Mexico; erasmo.on@gmail.com (G.M.-S.); almayalanis@gmail.com (A.Y.A.); nancyaranad@gmail.com (N.A.-D.)

* Correspondence: carlos.lopez@cucei.udg.mx

Received: 30 June 2020; Accepted: 27 August 2020; Published: 30 August 2020

Abstract: Mobile robots commonly have to traverse rough terrains. One way to find the easiest traversable path is by determining the types of terrains in the environment. The result of this process can be used by the path planning algorithms to find the best traversable path. In this work, we present an approach for terrain classification from aerial images while using a Convolutional Neural Networks at the pixel level. The segmented images can be used in robot mapping and navigation tasks. The performance of two different Convolutional Neural Networks is analyzed in order to choose the best architecture.

Keywords: mapping; semantic segmentation; convolutional neural networks; unet

1. Introduction

One of the most important tasks in mobile robotics is navigation. Robot navigation addresses the problem of moving a robot from its current position to a desired goal. Commonly, to achieve this task, the robot uses its onboard sensors to know the environment. In [1], the authors presented a method where Unmanned Aerial Vehicles (UAV) can work as a sensor for ground mobile robots, providing information of the environment from a perspective not available from the ground mobile. In Figure 1, we present an example of the information provided by the UAV.

Figure 1. Pictures taken by a Drone looking down to the ground.

An important problem that arises with the use of optical sensors is how to interpret the great amount of data that they provide. Usually, the data that are provided by an optical sensor are classified as traversable or no-traversable terrain. The maps can be three-dimensional (3D), as they are shown in [2–4] as well. In other applications, the difficulty of passing through different types of terrain is described with a cost assigned to some areas of the map like the ones that are described in [5,6].

In terms of path planning, this could give more possibilities to generate different paths, ones where the least possible distance is traveled, traversed in less time, or with less energy consumption.

Given the amount of data provided by a visual sensor mounted on a UAV, we require a preprocessing algorithm that extracts the most important information. To solve this problem, we propose an approach that is based on image segmentation. In image segmentation, the image is divided into different entities similar to Figure 2 where a dog is segmented from the image, and the rest is considered background. In our approach, a two-dimensional (2D) map is generated from aerial images using semantic segmentation with a Convolutional Neural Network (CNN). To perform this task, we assign a cost to each pixel in the image according to a classification over twelve classes. Our CNN has a U-shaped architecture that is based on U-net described in [7], which acts as an encoder-decoder, this architecture has shown great performance with limited data sets [8–10]. This semantic segmented image can be used to improve a map, since we have a pixel-level segmentation that can generate a more detailed map for a mobile robot navigation task.

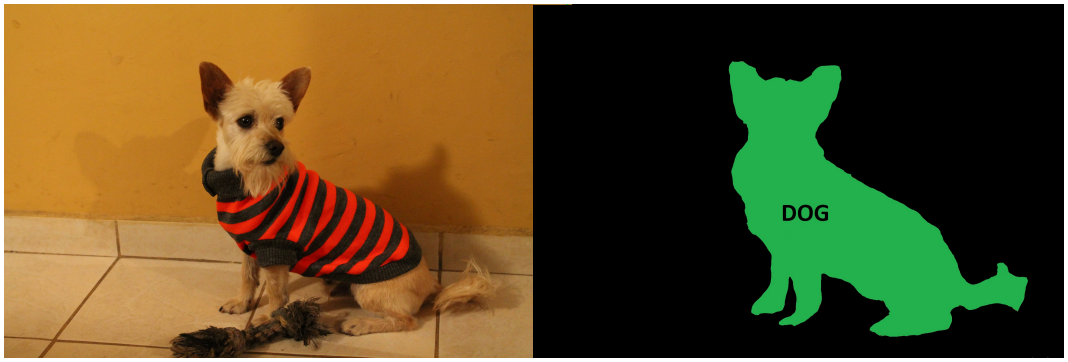

Figure 2. On the left side the input image and on the right the image where the dog is segmented.

The rest of the paper is organized, as follows: the related work is presented in Section 2. The architecture of the proposed approach is presented in Section 3. The experimental results of our approach are shown in Section 4. Finally, the conclusions are given in Section 5.

2. Related Work

The mapping task is commonly related to light detection and ranging (LiDAR) sensors. In some cases, the authors combine these sensors with algorithms to classify sections of the readings as in [11] where are combined with Support Vector Machines (SVM). In [12], the authors construct roadway maps with the KITTI dataset employing LiDar odometry. In [13], with a new LiDAR sensor, the possibility of map natural containers of water is opened. Recently, Independent Component Analysis (ICA) in conjunction with an SVM receives an input formed by LiDAR readings with images in high resolution to classify the land in seven types of plants and one more class for the unclassified, reaching an accuracy of 73.6% using only the information of the LiDAR plus 0.67% more by adding the RGB images of the terrain reaching 74.7% of accuracy. In [14], the authors made a semantic map using a 3D LiDAR sensor with a Multi-Layer Perceptron to classify the point cloud.

There are other approaches that only use images, for example, in [15], the authors used a sub-pixel mapping to deal with low-resolution images, this technique can be combined with other information such as [16,17]. Many of the mapping approaches use satellite images to map the roadways or terrain taken advantage of the files with multi-spectral bands like in [18,19]. Similarly, [20] maps a Chinese city classifying with a cascade topology of minimum distance, maximum likelihood and SVM. In [21], the authors map another city with an SVM.

Mapping the environment towards the robot could be performed with RGB-D images that include depth information [22]. Other works add thermal information to generate the map and later the path for the mobile robot. In [23], the authors created a point cloud labeled with eight classes using Conditional Random Fields (CRF) [24].

2.1. Convolutional Neural Networks

Certainly, the use of convolutional neural networks is one important part of the success of deep learning in image processing. Although neural networks operate primarily over a matrix, they also have a biological analogous, CNN captures representations of brain function, where every neuron is stimulated by a part of the visual field then all of the parts overlap to operate on the whole image, as shown in [25–28].

Convolution is the integral measuring of how much two functions overlap as one passes over the other. In image processing, the first function is the image and the second is the filter or kernel. Because the image is discrete, the convolution can be written as

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n) \tag{1}$$

where S is the map of features, I is the input image, K is the kernel or filter, and m, n are the indexes for rows and columns inside the kernel. Figure 3 shows a graphic description of how the output is generated.

$$I = \begin{bmatrix} I_{0,0} & I_{0,1} & I_{0,2} \\ I_{1,0} & I_{1,1} & I_{1,2} \\ I_{2,0} & I_{2,1} & I_{2,2} \end{bmatrix} \quad K = \begin{bmatrix} K_{0,0} & K_{0,1} \\ K_{1,0} & K_{1,1} \end{bmatrix} \quad S = \begin{bmatrix} I_{0,0}K_{0,0} + I_{0,1}K_{0,1} + I_{1,0}K_{1,0} + I_{1,1}K_{1,1} & I_{0,1}K_{0,0} + I_{0,2}K_{0,1} + I_{1,1}K_{1,0} + I_{1,2}K_{1,1} \\ I_{1,0}K_{0,0} + I_{1,1}K_{0,1} + I_{2,0}K_{1,0} + I_{2,1}K_{1,1} & I_{1,1}K_{0,0} + I_{1,2}K_{0,1} + I_{2,1}K_{1,0} + I_{2,2}K_{1,1} \end{bmatrix}$$

Figure 3. Graphic description of an input image of one channel of 3 × 3, A Kernel of 2 × 2 and the output.

One advantage of CNNs in image processing is that they share weights. Weight sharing reduces the parameters to learn; therefore, the required memory is lower. Subsequently, the convolutional layer applies the convolution of the kernel over the image or the output of another convolutional layer to produce a linear output that will be passed through a no lineal activation function, such as Sigmoid, Hyperbolic Tangent, Rectified Linear Unit (ReLU), Exponential Linear Unit (ELU), or Leaky ReLU. Other types of layers are the pooling layers that can be Max or Average, where the purpose is to reduce the number of parameters and controlling the overfitting, reducing the size of the network by taking the maximum or the average value of the area where the kernel is passing over the input. A graphic representation is shown in Figure 4, where max and average pooling layers of size 2 × 2 are passed to an array of 4 × 4.

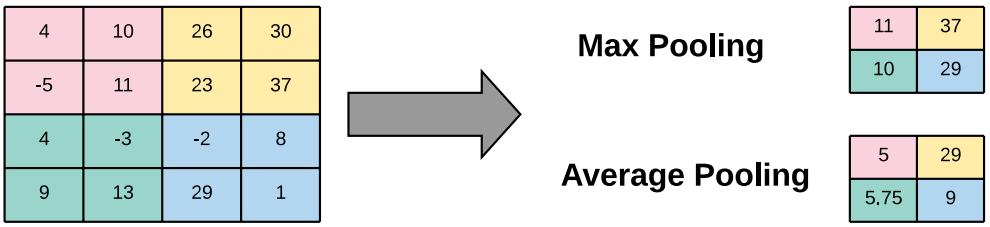

Figure 4. Graphic description of Max pooling and average pooling layer with a size of 2 × 2.

In addition to these two most used layers, there are other types, such as "network in network" [29], Flattened [30], Depthwise separable [31,32], spatial separable with asymmetric convolution [33], Deconvolution [34], etc. Some of them will be presented more in detail in Section 3. Given these elements, a CNN can learn features according to the application, in contrast with the hand-crafted features where the features should be designed manually. Finally, to enumerate some applications of CNN: there are image classification [35,36], object detection [37], image denoising [38], image super-resolution [39], and image segmentation, which will be detailed in the next section.

2.2. Semantic Segmentation

The main task of semantic segmentation is to understand the image at a pixel level labeling each pixel with a class, some classical approaches, such as [40], which use semisupervised learning of the class distributions and then does the segmentation by means of a multilevel logistic model, and [41] addresses the problem as one of signal decomposition, solving it with an alternating direction method of multipliers to perform separation of the text and moving objects from the background. With respect of CNNs normally is an end-to-end architecture, replacing the fully connected layers in common image classification networks for the layer that upsamples the features working as an encoder-decoder. Subsequently, the output usually has the same size as the input in which every pixel has a class assigned. Representing an improvement with respect to the common object detection or recognition where is bounding box enclosing the objects of interest. In terms of applications, semantic segmentation is used on autonomous vehicles commonly trained with the cityscapes dataset [42], human-computer interaction, or Biomedical Image analysis [7].

One of the most representative architectures for semantic segmentation is Fully Convolutional Network (FCN) [43], where the authors take famous classifiers ALexNet, VGG nets [44], and GoogLeNet [45] to connect them at the end upsampling layers for end-to-end learning by backpropagation from the pixel-wise loss. However, FCN has a problem with the pooling layer that is caused by the loss of information. For this reason, SegNet [46] adds to the upsampling layers a layer with the pooling indexes. The inconvenience is the lack of global context, to overpass this issue some changes have been implemented, such as atrous convolutions [47–49] to obtain different sub-region representations. Recent works, including Auto-deepLab [50], which alternates between optimizing the weights and the architecture of the network, Dual Attention Network (DANet) [51] that proposes modules to spatial contextual information and another to channel dimension to finally merge them, CCNet [52] introduces the criss-cross attention module to collect contextual information. Most of the approaches use Resnet101 [53] as the backbone encoder; this is summarized in [54]. Furthermore, approaches with the objective of performing in real-time or in hardware with limited resources namely ESNet [55] and FPENet [56].

Another popular architecture is U-net [7], which has an outstanding performance with datasets containing a short number of images. In the encoder part, it has the typical convolution layers followed by the pooling layers series, whereas, in the upsampling part, it concatenates a cropped copy of the features from the convolutions, providing local information to the global generated in the upsampling. The output layer has as many filters of 1x1 as classes to assign.

3. Semantic Segmentation Architecture

Our approach is based on the U-Net architecture, as it has shown great performance with limited datasets. The proposed architecture has fewer filters, in our case, the number of filters is reduced by four in each level when compared to the original. Using this architecture, we made another two different architectures, one with depthwise separable convolution (U-Net DS), see Figure 5, and another using spatial separable convolution (U-Net SS), with the objective to have fewer parameters to learn, since the intention is to run the network in the limited hardware of a mobile robot using less memory and speed up the segmentation. These two types of convolution are explained in the following two subsections. It is important to note that the activation functions were changed from ReLu to ELU, since it avoids the dying ReLU problem [57]. For the output layer is a 1×1 convolutional layer with twelve channels one channel for every class of interest with a softmax activation function, the classes are: Background, Tile, Grass, Person, Stairs, Wall, Roof, Tree, Car, Cement, Soil, and Injured person.

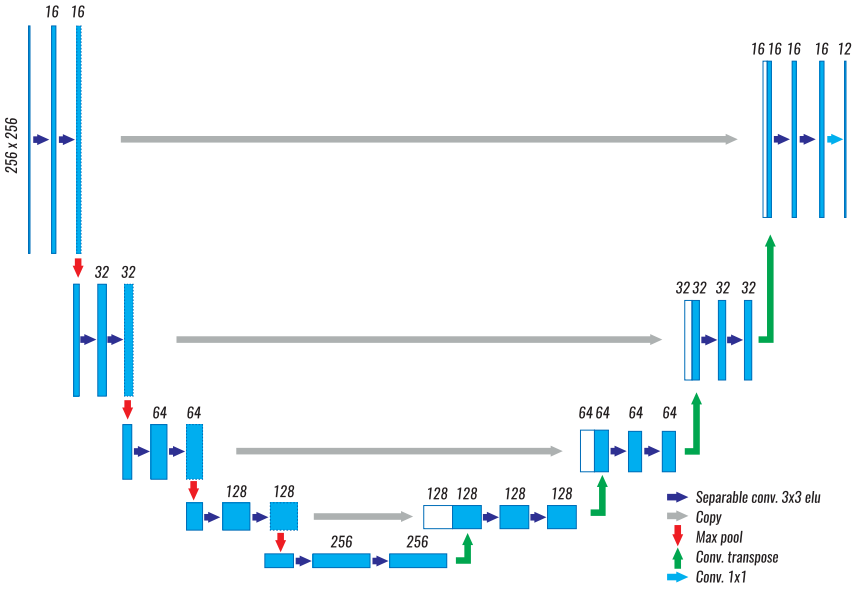

Figure 5. Graphic description the architecture selected.

3.1. Depthwise Separable Convolution

Depthwise Separable Convolutional divides the standard convolution into a depthwise convolution and an 1×1 convolution named pointwise. The first division takes an input of $h \times w \times m$ and applies m filters of size $f \times f \times 1$, where f is the height and width of the filter and m is equal to the number of channels in the input, as a result, an intermediate output of $(h - f + 1) \times (h - f + 1) \times m$ is generated. The pointwise part uses n filters of size $1 \times 1 \times m$ to generate an output of $(h - f + 1) \times (h - f + 1) \times n$, as shown in Figure 6.

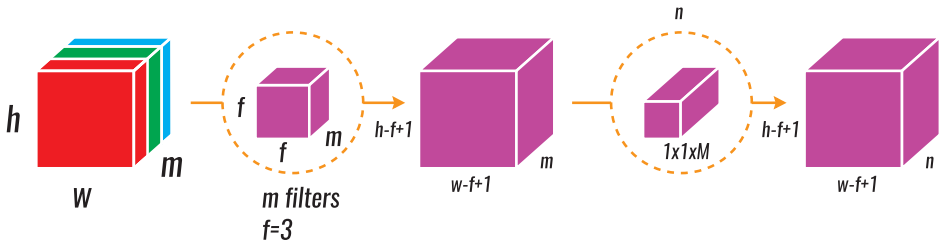

Figure 6. Graphic description of Depthwise Separable convolution.

The disadvantage of doing this separation is that the network has fewer parameters to learn, which means a loss in accuracy, however, the network size is smaller and, thus, it can run faster, suitable for limited hardware, such as mobile devices [58].

3.2. Spatial Separable Convolution

This convolution separates the kernel of $f \times f$ in two: one vertical and one horizontal, since they normally have dimensions of $f \times 1$ and $1 \times f$, respectively. This division of the filter reduces the number of multiplications. For example, if $f = 3$, instead of having nine multiplications in the conventional convolution, with the separation there are six multiplications, three for the vertical and

three for the horizontal, Figure 7 shows a graphic description of the separation and how the output has the same dimension as if a filter of $f \times f$ were applied to one channel image.

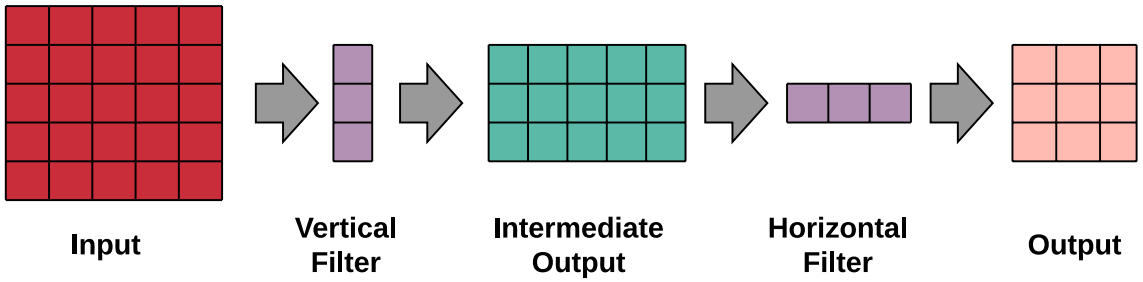

Figure 7. Graphic description of the Spatial separable convolution.

The advantage of this separation is that since the number of operations is reduced then the network runs faster. However, the number of parameters is less, and not all of the filters can be separated. In order to not limit the network, these separations were only added in the two in the second and fourth layers.

3.3. Training

The proposed architectures were trained during 100 epochs with a batch size of eight, the loss function was the categorical cross-entropy. The accuracy of the model was measured with the Dice function (F1 score), defined as

$$Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{2}$$

The dataset used contains photos that were taken from lower heights with drones or with cellphones from places, such as roofs or bridges similar to the Inria Aerial Dataset [59], but from a height where is easy to detect a mobile robot. It consists of 2647 images, where 2203 were taken for the training. The images do not have a fixed resolution or shape. They were labeled by an expert in the software labelme [60], see Figure 8. The quantity of images in which each class is present is shown in Table 1 except for the class background that is not labeled. The dataset was augmented with the transformations presented in Table 2. The dataset is available to the reader and the information can be consulted in the Supplementary Material section.

Table 1. Number of images in which each class is present.

Class	Number of Images
Cement	2117
Car	1746
Grass	1254
Person	864
Tree	728
Soil	672
Wall	575
Roof	390
Tile	354
Stairs	91
Injured person	10

Figure 8. Examples of the annotations that were made over the images.

Table 2. Transformations applied to the dataset.

Augmentations	Parameters
Zoom	[0.9, 1.2]
Flip	horizontal and vertical
Rotation	$[-45°, 45°]$
Brightness	[0.9, 1.2]
Translation	$[-0.1, 0.1]$
Fog	0.05 probability to add
Crop	0.05 probability

Our approach is compared against the typical U-Net, two networks that are focused on real-time semantic segmentation: the Light-weight Context Guided Network (CGNet) [61], DABNet [62], Additionally, a light version of the high-resolution network (HRNet) [63] with the number of kernels in each block reduced by a half. Finally, against Exfuse [64] with ResNet18 as the backbone encoder. Figures 9 and 10 present the comparative of the training.

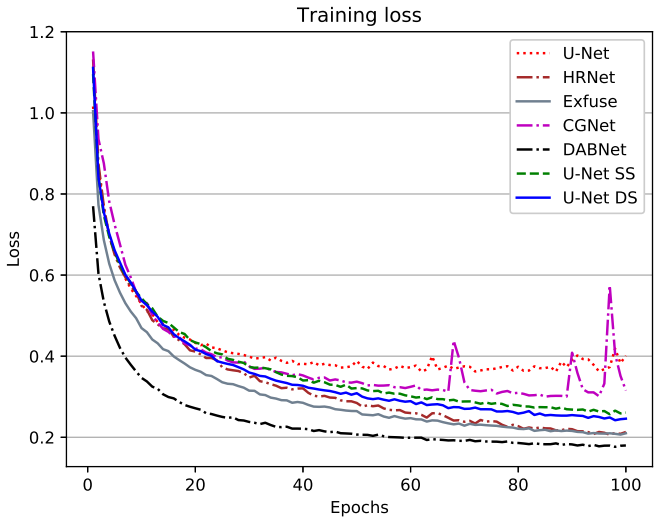

Figure 9. Comparison of the Loss function of the architectures during 100 epochs.

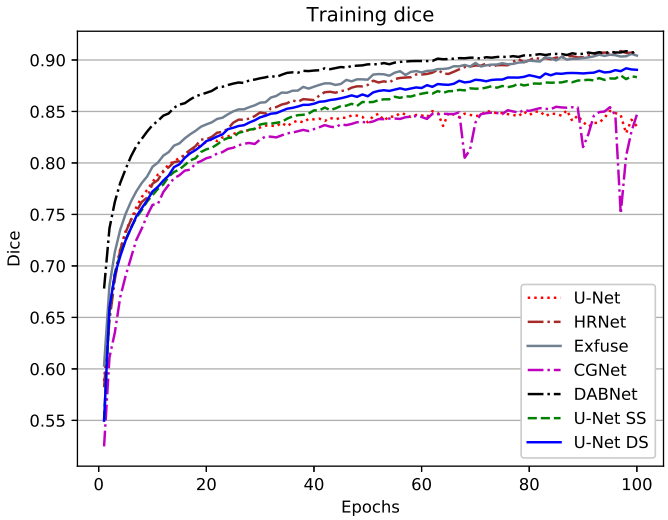

Figure 10. Comparison Dice function (F1 score) of the during the training.

4. Results

The networks were tested in a PC with windows as operative system, an AMD Ryzen 7 3750H processor, 16 GB of RAM, and a GPU RTX2060. The dataset for the test contains 444 images.

The results in the F1 score and the IoU, another popular metric, which is the Intersection over Union (IoU) or Jaccard Index defined by

$$Jaccard(A, B) = \text{IoU} = \frac{|A \cap B|}{|A \cup B|} \tag{3}$$

are presented in Table 3. It is shown that, as it occurred in the training, the ones that outperformed our approach were Exfuse and DABNet. Exfuse surpassed ours by 0.0363 in Dice 0.0915 in IOU, DABNet got a higher score with a difference of 0.0225 in Dice and 0.0282 in the IoU with respect to the U-Net with DS, but DABNet has an increase in the number of parameters of 96% and Exfuse 3997%. Concerning the frames per second (FPS) the one with the best results was U-Net DS with 17.2725 FPS against DABNet with 11.0042 FPS, which represents that our proposed architecture is 56.96% faster than DABNet and 64.48%. On the side of the other architectures, these were outperformed in all four aspects, called Dice, IOU, parameters, and FPS. These results are easily visible in Figures 11 and 12.

Table 3. Results on the test dataset in Dice and Intersection over Union (IoU) scores of each network.

Network	Dice	IoU
U-net	0.7139	0.5873
HRNet	0.6965	0.5728
Exfuse	0.8015	0.7460
CGNet	0.7250	0.6024
DABNet	0.7867	0.6827
U-net SS	0.7609	0.6486
U-net DS	0.7652	0.6545

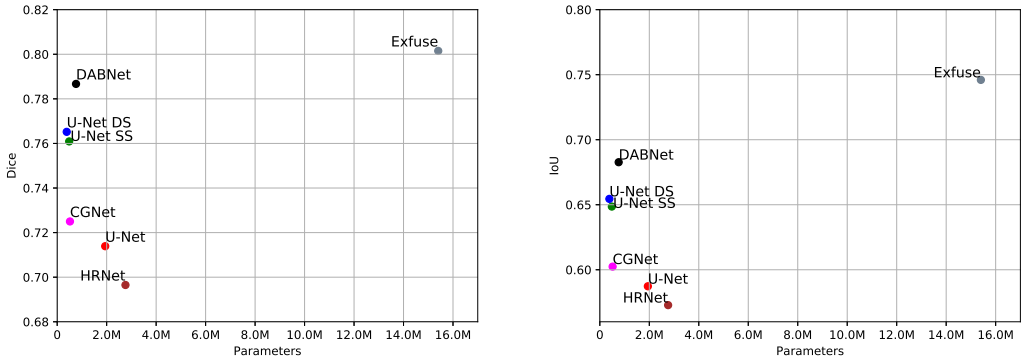

Figure 11. Comparison of the Dice and IoU scores vs. number of parameters.

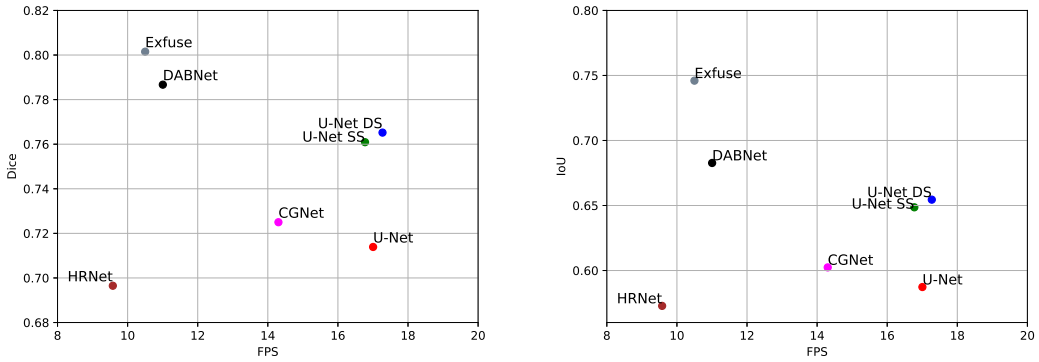

Figure 12. Comparison of the Dice and IoU scores vs. FPS.

Furthermore, a comparison of the U-Net DS varying the number of kernels in each level was done. Let us say that the number of kernels base is C. Subsequently, in the architecture, this value in each level is as $[C, 2C, 4C, 8C, 16C]$, in our approach $C = 16$ as Figure 5. The comparison was performed against U-Net DS with $C = 32$ and $C = 64$. The one with the best scores in Dice and IOU was the one with $C = 32$ getting 0.8292 and 0.7457, respectively, being even better than the one using $C = 64$, this could be due to overfitting. Taking in consideration that memory usage and velocity is the priority, once again U-Net DS using $C = 16$ had the best trade-off, having just 25.33% of parameters of U-Net DS with $C = 32$ and 8.93% faster at the cost of losing 6.4% in Dice and 9.1% in IoU scores, see Figures 13 and 14. Moreover U-Net DS with $C = 32$ and $C = 64$ are faster than the DABNet, but they have at least the double of parameters.

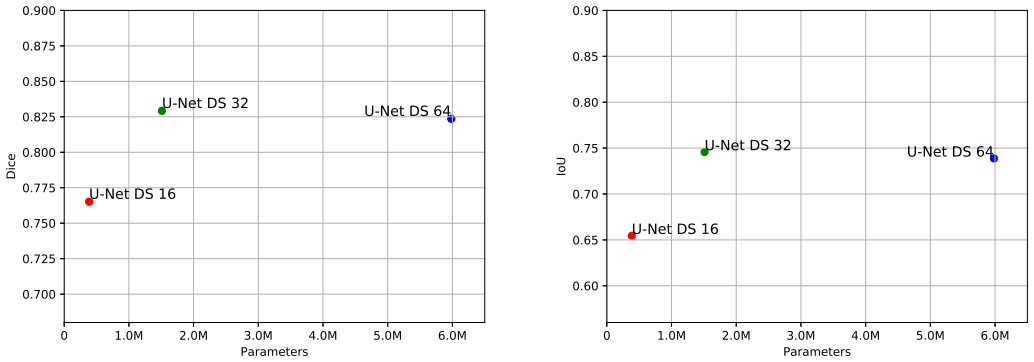

Figure 13. Comparison of the U-Net DS changing values of C to 16, 32 and 64 in Dice and IoU scores vs. number of parameters.

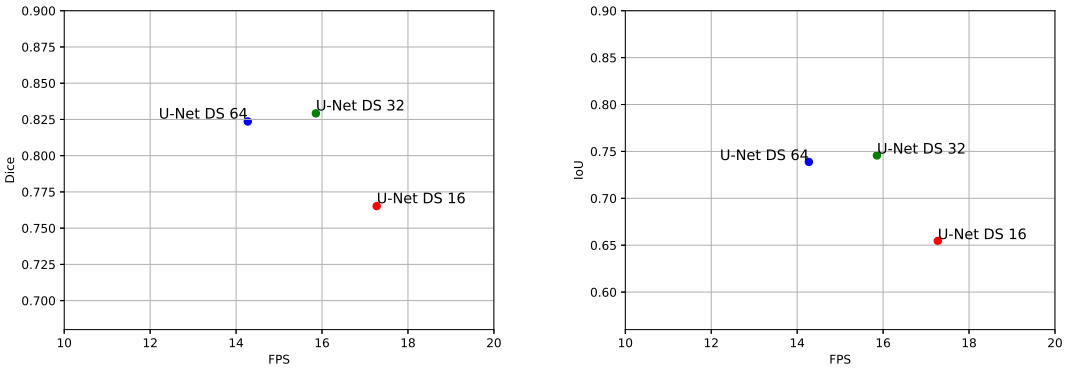

Figure 14. Comparison of the U-Net DS changing values of *C* to 16, 32 and 64 in Dice and IoU scores vs. FPS.

Additionally, Table 4 presents the results for each class using the U-net with depthwise separable convolutions, which gives a major understanding of the general results. The low score in the injured person class is due to the dataset lacks a considerable amount of examples with injured persons whereas classes, like tile, cement, and grass, have a similar texture or color, or the car class where the majority of the cars have a similar shape just changing the color are easy to learn for the net. Additionally, most of the higher scores match with the number of examples per class, as an example, the class cement is present in 2117 images from 2203 from the training dataset, while classes with low scores, such as Injured Person and Stairs are present in 10 and 91 images, respectively. Qualitative results are shown in Figure 15 with the input, the output of four of the twelve classes, and finally, the map with all of the classes colored. The simplicity of the implementation, the low inference time, and the low hardware requirements such as memory, processor or graphic card are some of the many advantages of the proposal achieved thanks to the architecture and the choice of a reduced number of filters combined with the use of depthwise convolution.

Table 4. Results on the test dataset for each class.

Class	Dice	IoU
Background	0.4830	0.3184
Tile	0.8091	0.6794
Grass	0.7907	0.6538
Person	0.4327	0.2760
Stairs	0.6067	0.4354
Wall	0.6351	0.4653
Roof	0.6511	0.4827
Tree	0.7757	0.6336
Car	0.8211	0.6965
Cement	0.8529	0.7436
Soil	0.6524	0.4841
Injured Person	0.0352	0.0179

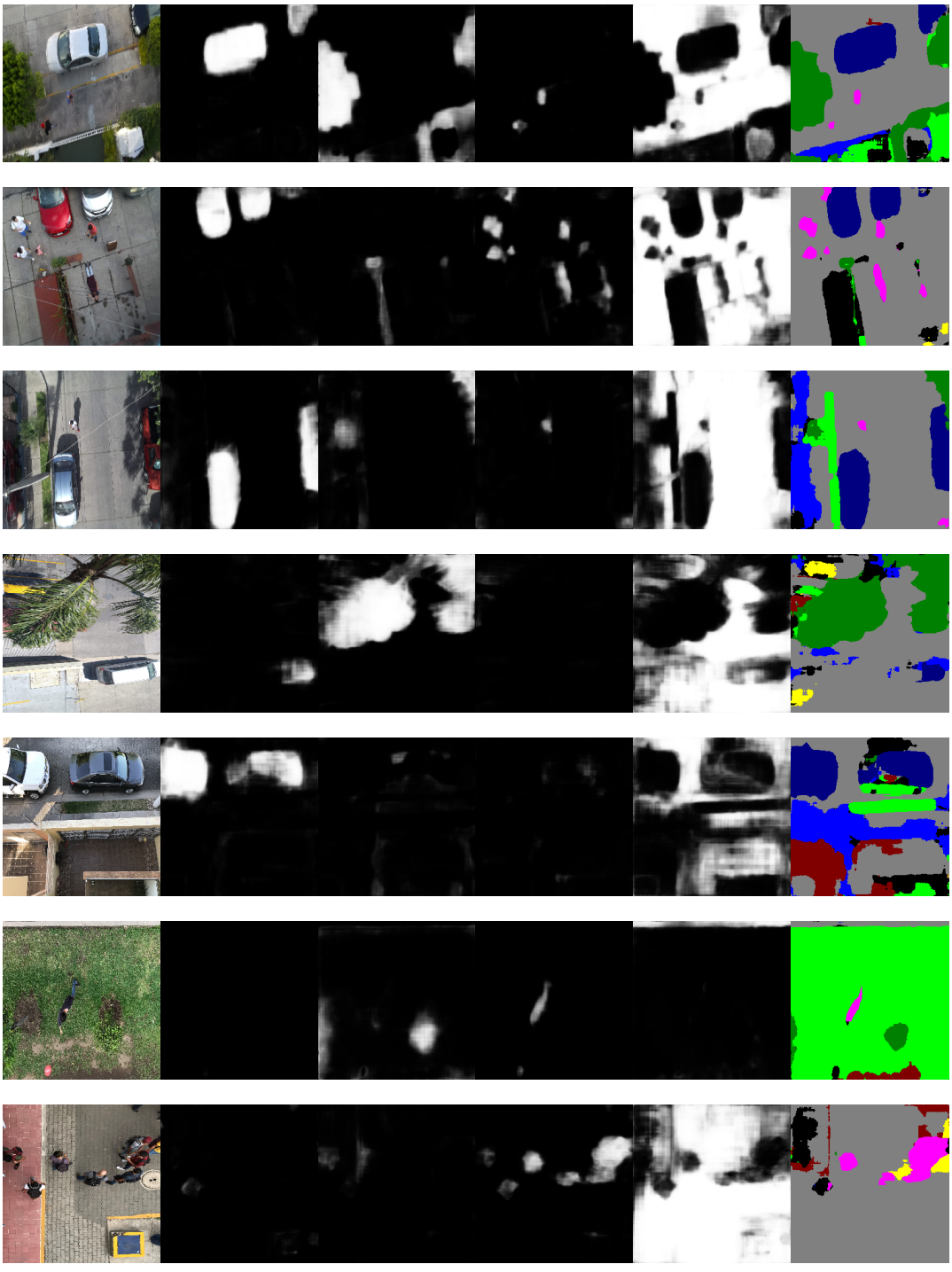

Figure 15. *Cont.*

Figure 15. Results on the test, first column the input image. The second column shows the output layer corresponding to the car class. In the third column the output layer for the tree class. The fourth column contains the pixels classified with the person label. Fifth column shows the output for the cement class and, finally, the output with all of the classes presented in the image.

5. Conclusions

In this paper, the authors proposed a method to extract the most important information from aerial images using a CNN for image segmentation. To solve the problem two different architectures of CNN have been proposed. The experimental results show that the U-net with depthwise separable convolutions is the best architecture for this problem due to it had the best trade-off having fewer parameters that correspond to less memory usage and an increase of 50% in the FPS despite the loss of 2% and 4% in the Dice and IoU scores. This architecture is able to segment the dataset correctly. With these results, an UAV can send aerial images to a mobile robot, which can apply the proposed algorithm to perform the task of mapping the terrain. The results of the algorithm can be used by the path planning algorithm of the mobile robot to perform navigation tasks.

Supplementary Materials: The data set can be accessed at https://github.com/GabrielMtzSoltero/SSEGfor_aerial_mapping/.

Author Contributions: All of the experiments in this paper have been designed and performed by G.M.-S. The reported results on this work were analyzed and validated by A.Y.A., C.L.-F., and N.A.-D. All authors are credited for their contribution on the writing and edition of the presented manuscript.

Funding: The authors thank the support of CONACYT Mexico, through Projects CB258068 ("Project supported by *Fondo Sectorial de Investigación para la Educación*"), and PN4107 ("Problemas Nacionales").

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

References

1. Qin, H.; Meng, Z.; Meng, W.; Chen, X.; Sun, H.; Lin, F.; Ang, M.H. Autonomous Exploration and Mapping System Using Heterogeneous UAVs and UGVs in GPS-Denied Environments. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1339–1350. [CrossRef]
2. Ye, E.; Shaker, G.; Melek, W. Lightweight Low-Cost UAV Radar Terrain Mapping. In Proceedings of the 2019 13th European Conference on Antennas and Propagation (EuCAP), Krakow, Poland, 31 March–5 April 2019; pp. 1–5.
3. Kim, J.H.; Kwon, J.W.; Seo, J. Multi-UAV-based stereo vision system without GPS for ground obstacle mapping to assist path planning of UGV. *Electron. Lett.* **2014**, *50*, 1431–1432. [CrossRef]
4. Jiang, Z.; Wang, J.; Song, Q.; Zhou, Z. A simplified approach for a downward-looking GB-InSAR to terrain mapping. In Proceedings of the 2016 IEEE International Conference on Digital Signal Processing (DSP), Beijing, China, 16–18 October 2016; pp. 194–198.

5. Arana-Daniel, N.; Valencia-Murillo, R.; Alanís, A.Y.; Villaseñor, C.; López-Franco, C. Path Planning in Rough Terrain Using Neural Network Memory. In *Advanced Path Planning for Mobile Entities*; IntechOpen: London, UK : 2017. Available online: https://www.intechopen.com/books/advanced-path-planning-for-mobile-entities/path-planning-in-rough-terrain-using-neural-network-memory (accessed on 1 June 2020)
6. Hata, A.Y.; Wolf, D.F. Terrain mapping and classification using Support Vector Machines. In Proceedings of the 2009 6th Latin American Robotics Symposium (LARS 2009), Valparaiso, Chile, 29–30 October 2009; pp. 1–6.
7. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015*; Springer: Cham, Switzerland, 2015; pp. 234–241.
8. Abraham, N.; Khan, N.M. A Novel Focal Tversky Loss Function With Improved Attention U-Net for Lesion Segmentation. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019; pp. 683–687.
9. Iglovikov, V.; Shvets, A. TernausNet: U-Net with VGG11 Encoder Pre-Trained on ImageNet for Image Segmentation. *arXiv* **2018**, arXiv:1801.05746.
10. Jaeger, P.F.; Kohl, S.A.A.; Bickelhaupt, S.; Isensee, F.; Kuder, T.A.; Schlemmer, H.P.; Maier-Hein, K.H. Retina U-Net: Embarrassingly Simple Exploitation of Segmentation Supervision for Medical Object Detection. In *Proceedings of the Machine Learning for Health NeurIPS Workshop*; Dalca, A.V., McDermott, M.B., Alsentzer, E., Finlayson, S.G., Oberst, M., Falck, F., Beaulieu-Jones, B., Eds.; Proceedings of Machine Learning Research (PMLR): Vancouver, BC, Canada 2020; Volume 116, pp. 171–183.
11. David, L.C.G.; Ballado, A.H. Mapping mangrove forest from LiDAR data using object-based image analysis and Support Vector Machine: The case of Calatagan, Batangas. In Proceedings of the 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Cebu City, Philippines, 9–12 December 2015; pp. 1–5.
12. Hamieh, I.; Myers, R.; Rahman, T. Construction of Autonomous Driving Maps employing LiDAR Odometry. In Proceedings of the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada, 5–8 May 2019; pp. 1–4.
13. Fernandez-Diaz, J.C.; Glennie, C.L.; Carter, W.E.; Shrestha, R.L.; Sartori, M.P.; Singhania, A.; Legleiter, C.J.; Overstreet, B.T. Early Results of Simultaneous Terrain and Shallow Water Bathymetry Mapping Using a Single-Wavelength Airborne LiDAR Sensor. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 623–635. [CrossRef]
14. Sun, L.; Yan, Z.; Zaganidis, A.; Zhao, C.; Duckett, T. Recurrent-OctoMap: Learning State-Based Map Refinement for Long-Term Semantic Mapping With 3-D-Lidar Data. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3749–3756. [CrossRef]
15. He, D.; Zhong, Y.; Ma, A.; Zhang, L. Sub-pixel intelligence mapping considering spatial-temoporal attraction for remote sensing imagery. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 602–605.
16. Xu, X.; Zhong, Y.; Zhang, L.; Zhang, H. Sub-pixel mapping based on a MAP model with multiple shifted hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *6*, 580–593. [CrossRef]
17. He, D.; Zhong, Y.; Zhang, L. Spectral–Spatial–Temporal MAP-Based Sub-Pixel Mapping for Land-Cover Change Detection. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 1696–1717. [CrossRef]
18. Senturk, S.; Sertel, E.; Kaya, S. Vineyards mapping using object based analysis. In Proceedings of the 2013 Second International Conference on Agro-Geoinformatics (Agro-Geoinformatics), Fairfax, VA, USA, 12–16 August 2013; pp. 66–70.
19. Zhang, Y.; Liu, Q.; Liu, G.; Tang, S. Mapping of circular or elliptical vegetation community patches: A comparative use of SPOT-5, ALOS And ZY-3 imagery. In Proceedings of the 2015 8th International Congress on Image and Signal Processing (CISP), Shenyang, China, 14–16 October 2015; pp. 776–781.
20. Cao, S.; Xu, W.; Sanchez-Azofeif, A.; Tarawally, M. Mapping Urban Land Cover Using Multiple Criteria Spectral Mixture Analysis: A Case Study in Chengdu, China. In Proceedings of the IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 2701–2704.

21. Zhai, L.; Xie, W.; Sang, H.; Sun, J.; Yang, G.; Jia, Y. Land Cover Mapping with Landsat Data: The Tasmania Case Study. In Proceedings of the 2011 International Symposium on Image and Data Fusion, Tengchong, China, 9–11 August 2011; pp. 1–4.
22. Zhao, L.; Liu, Y.; Jiang, X.; Wang, K.; Zhou, Z. Indoor Environment RGB-DT Mapping for Security Mobile Robots. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Intelligent Robotics and Applications, Shenyang, China, 8–11 August 2019*; Springer: Cham, Switzerland, 2019; pp. 131–141.
23. Mitsou, N.; de Nijs, R.; Lenz, D.; Frimberger, J.; Wollherr, D.; Kühnlenz, K.; Tzafestas, C. Online semantic mapping of urban environments. In *Lecture Notes in Computer Science, Proceedings of the International Conference on Spatial Cognition, Kloster Seeon, Germany, 31 August–3 September 2012*; Springer: Berlin/Heidelberg, Germany, 2012, pp. 54–73.
24. Kumar, S.; Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In Proceedings of the Ninth IEEE International Conference on Computer Visio, Nice, France, 13–16 October 2003; pp. 1150–1157.
25. Eickenberg, M.; Gramfort, A.; Varoquaux, G.; Thirion, B. Seeing it all: Convolutional network layers map the function of the human visual system. *NeuroImage* **2017**, *152*, 184–194. [CrossRef]
26. Kuzovkin, I.; Vicente, R.; Petton, M.; Lachaux, J.P.; Baciu, M.; Kahane, P.; Rheims, S.; Vidal, J.R.; Aru, J. Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex. *Commun. Biol.* **2018**, *1*, 1–12. [CrossRef]
27. DiCarlo, J.J.; Zoccolan, D.; Rust, N.C. How does the brain solve visual object recognition? *Neuron* **2012**, *73*, 415–434. [CrossRef]
28. Cichy, R.M.; Khosla, A.; Pantazis, D.; Torralba, A.; Oliva, A. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Sci. Rep.* **2016**, *6*, 27755. [CrossRef] [PubMed]
29. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.
30. Jin, J.; Dundar, A.; Culurciello, E. Flattened convolutional neural networks for feedforward acceleration. *arXiv* **2014**, arXiv:1412.5474.
31. 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.
32. Chollet, F. Xception: Deep Learning With Depthwise Separable Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017 .
33. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
34. Noh, H.; Hong, S.; Han, B. Learning Deconvolution Network for Semantic Segmentation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015 .
35. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
36. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
37. Zhang, K.; Zhang, Z.; Li, Z.; Qiao, Y. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **2016**, *23*, 1499–1503. [CrossRef]
38. Ren, H.; El-Khamy, M.; Lee, J. Dn-resnet: Efficient deep residual network for image denoising. In *Lecture Notes in Computer Science, Proceedings of the Asian Conference on Computer Vision. Springer, Perth, Australia, 2–6 December 2018*; Springer: Cham, Switzerland, 2018; pp. 215–230.
39. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image super-resolution using deep convolutional networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef]
40. Li, J.; Bioucas-Dias, J.M.; Plaza, A. Semisupervised Hyperspectral Image Segmentation Using Multinomial Logistic Regression With Active Learning. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 4085–4098. [CrossRef]
41. Minaee, S.; Wang, Y. An ADMM Approach to Masked Signal Decomposition Using Subspace Representation. *IEEE Trans. Image Process.* **2019**, *28*, 3192–3204. [CrossRef]

42. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223.
43. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
44. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
45. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper With Convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
46. 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. [CrossRef]
47. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]
48. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
49. 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.
50. Liu, C.; Chen, L.C.; Schroff, F.; Adam, H.; Hua, W.; Yuille, A.L.; Fei-Fei, L. Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 82–92.
51. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3146–3154.
52. Huang, Z.; Wang, X.; Huang, L.; Huang, C.; Wei, Y.; Liu, W. Ccnet: Criss-cross attention for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 603–612.
53. 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.
54. Minaee, S.; Boykov, Y.; Porikli, F.; Plaza, A.; Kehtarnavaz, N.; Terzopoulos, D. Image segmentation using deep learning: A survey. *arXiv* **2020**, arXiv:2001.05566.
55. Wang, Y.; Zhou, Q.; Xiong, J.; Wu, X.; Jin, X. ESNet: An Efficient Symmetric Network for Real-Time Semantic Segmentation. In *Lecture Notes in Computer Science, Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Xi'an, China, 8–11 November 2019*; Springer: Cham, Switzerland, 2019; pp. 41–52.
56. Liu, M.; Yin, H. Feature Pyramid Encoding Network for Real-time Semantic Segmentation. *arXiv* **2019**, arXiv:1909.08599.
57. Pedamonti, D. Comparison of non-linear activation functions for deep neural networks on MNIST classification task. *arXiv* **2018**, arXiv:1804.02763.
58. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
59. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017.
60. Wada, K. labelme: Image Polygonal Annotation with Python. 2016. Available online: https://github.com/wkentaro/labelme (accessed on 28 August 2019).
61. Wu, T.; Tang, S.; Zhang, R.; Zhang, Y. Cgnet: A light-weight context guided network for semantic segmentation. *arXiv* **2018**, arXiv:1811.08201.
62. Li, G.; Yun, I.; Kim, J.; Kim, J. Dabnet: Depth-wise asymmetric bottleneck for real-time semantic segmentation. *arXiv* **2019**, arXiv:1907.11357.

63. Sun, K.; Zhao, Y.; Jiang, B.; Cheng, T.; Xiao, B.; Liu, D.; Mu, Y.; Wang, X.; Liu, W.; Wang, J. High-resolution representations for labeling pixels and regions. *arXiv* **2019**, arXiv:1904.04514.
64. Zhang, Z.; Zhang, X.; Peng, C.; Xue, X.; Sun, J. Exfuse: Enhancing feature fusion for semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 269–284.