Segmentation of Photovoltaic Panels in Aerial Photography using Group Equivariant FCNs

Lars Bokkers[1], Luca Ambrogioni, and Umut Güçlü

Radboud University, Donders Institute for brain Cognition and Behaviour,
Nijmegen, Netherlands
l.bokkers@student.ru.nl[1]

Abstract. Previous research has shown the benefits of group equivariant convolutions for image recognition tasks. With this work we apply group equivariance to the segmentation of photovoltaic (PV) panel installations in aerial photography to determine whether the benefits translate to aerial photography segmentation. We create a custom annotation of PV panel installations in two Dutch cities using open access aerial photography. We show that group equivariant versions of traditional and residual convolutional neural networks indeed perform at least as well as the traditional versions and provide better generalization.

1 Introduction

In the last decade, the number of photovoltaic (PV) installations on rooftops has increased fast in The Netherlands [13]. For differing reasons, various public and private parties are interested in knowing where PV installations are located. With this work we aim to fulfill this data requirement by using a fully convolutional network to segment aerial photography into regions that represent solar panels and those that do not. This results in both the location and size of PV installations.

We are specifically interested in the performance benefits of group equivariant convolutions as proposed by Cohen and Welling [3]. These type of convolutions constrain the network to converge with filters invariant to pre-defined symmetries.

This work makes the following novel contribution; we show that group equivariant convolutions improve performance and generalization of fully convolutional networks applied to aerial photography segmentation.

The remainder of this work is structured as follows. In Section 3 we first reiterate dilated and group equivariant convolutions and follow with a description of our models and the data that was used. In Section 4 we list our results and interpretation thereof and end with a discussion and our conclusions in Section 5.

2 Related Work

2.1 Aerial Photography Segmentation

Image segmentation has already been widely been applied to aerial photography. For example, [11] applied *Fully Convolutional Networks (FCNs)* segmentation of the ISPRS Vaihingen and Potsdam datasets. The aerial photography was segmented into various urban and suburban regions such as buildings, vegetation, streets and vehicles. Additionally, Li et al. combined a larger version of U-Net with residual learning to get very high accuracy on coastline segmentation to separate land and sea [5,6].

2.2 PV Installation Segmentation

In [7], the authors used an architecture based on work by the Oxford Visual Geometry Group (VGG) [12]. Using a sliding-window approach, they classified 41x41 pixel segments on the presence of solar panels true/false. To remain computationally feasible, the sliding-window had a 5 pixel-stride, effectively creating a mask of 1/25th the size of the original image. They then upscaled this image to get a mask of the original size.

Next, the same research group proposed an architecture inspired by *U-Net*: *SegNet* [10,2]. This architecture contains both a contracting segment and an expansive segment. The contracting segment is again inspired by VGG. Where the contracting segment contains the traditional pooling and convolution layers, the expansive segment contains upscaling and transposed convolution layers. This way, the expansive path essentially mirrors the contracting path. Additionally, skip-connections are used to transfer low-level local features through the network. At the end of the network, 1x1 convolutions are used to make per-pixel classifications.

3 Methods

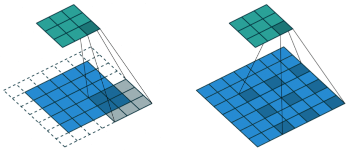
3.1 Convolutions

Dilated Convolutions The networks used in this network are based on the *Context Network* defined in [14]. This network is a feed-forward convolutional neural network that uses *dilated convolutions*. While for normal convolutions, all elements in the filter are placed on the image sampling grid with translations of 1 along each axis, for dilated convolutions the translation is larger.

Concretely, given an n by m input with c channels $\boldsymbol{X} \in \mathbb{R}^{c \times n \times m}$, and a convolution weight tensor for an x by y filter $\boldsymbol{w} \in \mathbb{R}^{c \times x \times y}$, the output of the dilated convolution operation $\boldsymbol{Y} = \boldsymbol{X} * \boldsymbol{w}$ given dilation factor $d \in Z, d \leq 1$ is defined in Equation 1. Note that a dilated convolution with $d = 1$ is equal to the regular (non-dilated) convolution operation. See Figure 1 for a visual representation of the dilated convolution operation.

$$\boldsymbol{Y}_{c,i,j} = \sum_a \sum_b \boldsymbol{w}_{c,a,b} \boldsymbol{X}_{c,i+d \cdot a',j+d \cdot b'} \tag{1}$$

Group Equivariant Convolutions Regular convolutional layers learn multiple instances of the same filter in different poses leading to less general and thus overfitted filterbanks. Cohen and Welling describe group equivariant convolutions to combat this shortcoming [3]. Given a symmetry group, a single filter is expanded into a group of filters containing each possible pose in the symmetry group. The expanded filterbank is applied in the forward pass with the regular convolution operation. In the backward pass, the backpropogated errors are collapsed using the inverse symmetry operation. This combines the error from each filter pose and results in training all filter poses as a single filter. For a detailed description of symmetry groups and corresponding proofs we refer the reader to the work of Cohen and Welling. In this work, we use the symmetry group of 90° point-wise rotations.



(a) `dilation=1` (b) `dilation=2`

Fig. 1: Visualization of dilated convolution operation. The blue segment represents image sampling grid, the green segment represents output of the convolution weights. Note the larger receptive area for the dilated convolution with the same number of filters. Visualizations from [4].

3.2 Models

Our networks operate as illustrated in Figure 2. Given an RGB image that is normalized to a mean of 0.5 and standard deviation on all channels. The output is an grayscale image with the per-pixel likelihood that a PV is present. A per-model optimized threshold is applied to convert the likelihoods to classifications which are used to compute model performance.

Our baselines are the context network as described by Yu and Kolton (see Table 1) with 20 channels and 64 channels, which we call `FCN20` and `FCN64`. The larger version is trained to rule out the possibility of an information bottleneck. Additionally, we train versions of both networks with residual blocks which we call `ResFCN20` and `ResFCN64`. Each network has an additional 1x1 2D-convolution

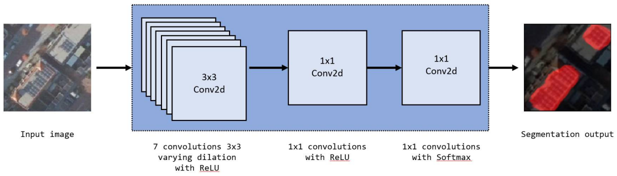


Fig. 2: **Our PV panel installation segmentation model structure.** The input of the network is an RGB image. The image is fed through 7 convolution layers with 3x3 kernels, followed by a convolution layer with 1x1 kernels. Finally, an additional convolution layer with 1x1 kernels is applied with Softmax activation. All other convolution layers use ReLU activation. Table 1 lists the details of the non-classfication layers.

layer with softmax activation to make the final pixel-wise classification. We then compare each of the described networks to their group equivariant counterpart: `GFCN5`, `GFCN16`, `ResGFCN5` and `ResGFCN16`. See Table 2 for an overview of the networks.

Each group equivariant network has $1/4^{th}$ of the filters of the non-equivariant one as our focus is on the generalization of convolution filters and symmetry group $P4$ contains 4 symmetry instances. That way, the amount of output features created by each layer and used by the next layer stays identical, allowing us to truly consider the benefit of standardization of filters. It should be noted that this results in a lower amount of trainable parameters for the group equivariant versions of the networks.

Layer	1	2	3	4	5	6	7	8
Kernel Size	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation	1	1	2	4	8	16	1	1
Receptive Field	3×3	5×5	9×9	17×17	33×33	65×65	67×67	67×67
Output Channels	C	C	C	C	C	C	C	C

Table 1: The Context Network as defined by Yu and Koltun. C is either 20 or 64.

3.3 Data

We created a custom dataset based on open access aerial photography of The Netherlands[9]. The photography, collected in 2017, has a ground-level resolution of 25 × 25cm per pixel. A non-exhaustive annotation with 3,192 polygons indicating photovoltaic panel installation locations was created by a single annotator for two cities; Nijmegen and Tilburg.

Network	Filters	Channels	Residuals	Group Equivariance	Trainable Parameters
FCN20	20	20	✗	✗	23,862
ResFCN20	20	20	✓	✗	24,024
FCN64	64	64	✗	✗	231,172
ResFCN64	64	64	✓	✗	231,686
GFCN5	5	20	✗	✓	5,997
ResGFCN5	5	20	✓	✓	6,039
GFCN16	16	64	✗	✓	59,906
ResGFCN16	16	64	✓	✓	58,036

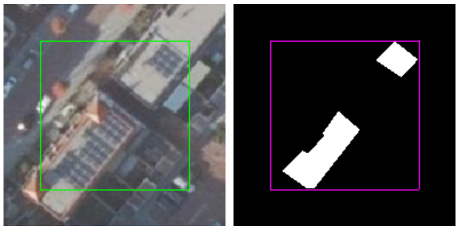
Table 2: Comparison of the FCN architectures.

From the polygons we created masks of 128 × 128 pixels (32 × 32 meters at ground level). The corresponding photography cutouts were pre-padded such that the output mask had had full context despite the representation shrinkage due to convolution applications. This resulted in input images of 194 × 194 pixels (48.5 × 48.5 meters at ground level) See Figure 3 for an input/output image-pair.

3.4 Training

Using a Stochastic Gradient Descent optimizer with momentum of 0.9, we trained the networks to minimize cross entropy loss. We applied class weighting to the loss in order to compensate for the imbalance between PV pixels and non-PV pixels. Each network was trained for 490 epochs with learning rate 10^{-2} and 10 epochs with learning rate 10^{-3} for fine-tuning.

Two types of data augmentation were used to artificially enlarge our dataset; (1) we randomly rotate between $-45°$ and $45°$ and (2) applied a horizontal flip with 50% probability. The rotation interval was chosen as anything additional rotation would fit in the domain of one of the pointwise 90° rotation symmetries, rendering any rotation outside the interval redundant.



(a) Input sample example.

(b) The corresponding mask.

Fig. 3: On the left is an input image for the network, on the right is the corresponding ground-truth mask. The purple rectangle in the mask image indicates the actual 128 × 128 size. A green rectangle has been added to the input image to indicate the same area. The imagery outside the green rectangle is the pre-padding.

3.5 Evaluation

We measure the accuracy of our models using the recall, precision, F1, intersection of union and DICE coefficient metrics. The network state corresponding to the lowest loss was used for performance evaluation.

4 Results

4.1 Performance

The performance of the networks is listed in Table 3. Comparing `FCN20` and `GFCN5`, we see `GFCN5` performs better than `FCN20` on both the train set and testset, albeit only a few percent. An increase in performance is in line with the findings of Cohen and Welling, though they saw a larger increase. We also see similar performance for `GFCN16` compared to `FCN64` on both sets. Although the former showed higher performance when including less-significant decimals, without multiple runs to test for significance we cannot conclude which performs better.

For the residual networks, the effect is less apparent. Comparing of `ResFCN20` and `ResGFCN5` shows an opposite effect for precision, F1, DICE and IoU on both sets; where we see better performance for the network not using group equivariance. Only the recall is scored higher by `ResGFCN5`. The comparison of `ResGFCN16` and `ResFCN64` shows a slightly better fit for `ResGFCN16` on the testset but on trainset. Like the comparison of `GFCN16` and `FCN64`; the difference between `ResGFCN16` and `ResFCN64` is only small.

Model	Set	Precision	Recall	F1	DICE	IoU
FCN20	train	0.502	0.967	0.658	0.955	0.494
	test	0.510	0.977	0.668	0.953	0.504
ResFCN20	train	0.501	0.973	0.658	0.955	0.494
	test	0.477	0.941	0.628	0.947	0.462
FCN64	train	0.542	0.980	0.696	0.961	0.537
	test	**0.550**	**0.987**	0.704	**0.960**	**0.550**
ResFCN64	train	0.523	0.987	0.681	0.959	0.520
	test	0.506	0.972	0.663	0.953	0.497
GFCN5	train	0.522	0.959	0.674	0.958	0.511
	test	0.524	0.965	0.676	0.955	0.514
ResGFCN5	train	0.442	0.976	0.602	0.944	0.437
	test	0.458	0.978	0.618	0.943	0.453
GFCN16	train	0.545	0.981	0.698	0.962	0.540
	test	**0.550**	**0.987**	**0.705**	**0.960**	0.546
ResGFCN16	train	0.518	0.983	0.676	0.958	0.513
	test	0.517	0.952	0.668	0.955	0.505

Table 3: Performance metrics on both train and test set for all trained models.

Model	Δ Precision (%)	Δ Recall (%)	Δ F1 (%)	ΔDICE (%)	ΔIoU (%)
FCN20	0.008 (+1.6)	0.009 (+1.0)	0.009 (+1.4)	-0.002 (-0.2)	0.010 (+2.1)
ResFCN20	0.024 (+5.13)	0.032 (+3.43)	0.030 (+4.71)	0.008 (+0.84)	0.032 (+6.88)
FCN64	0.008 (+1.4)	0.007 (+0.7)	0.008 (+1.2)	-0.002 (-0.2)	0.009 (+1.8)
ResFCN64	0.017 (+3.44)	0.015 (+1.59)	0.018 (+2.77)	0.006 (+0.64)	0.022 (+4.34)
GFCN5	0.002 (+0.3)	0.006 (+0.7)	0.003 (+0.4)	-0.003 (-0.3)	0.003 (+0.7)
GFCN16	0.005 (+0.9)	0.007 (+0.7)	0.006 (+0.9)	-0.002(-0.2)	0.007(+1.3)
ResGFCN5	-0.015 (-3.31)	-0.002 (-0.17)	-0.016 (-2.57)	0.001 (+0.08)	-0.015 (+3.38)
ResGFCN16	0.001 (+0.11)	0.031 (+4.24)	0.008 (+1.19)	0.003 (+0.33)	0.009 (+1.74)

Table 4: Difference between train and test scores as listed in Table 3. Positive numbers indicate higher score on trainset, negative numbers indicate higher score on testset.

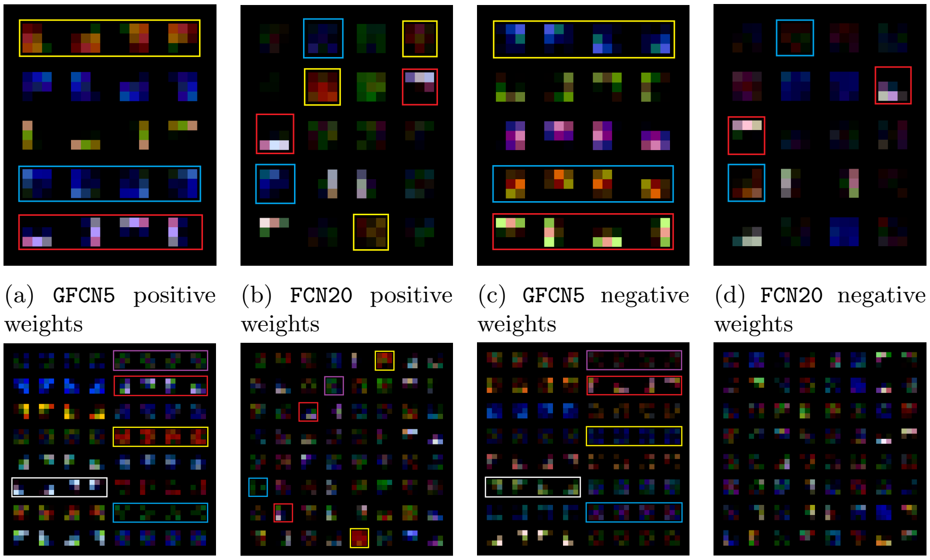
By using group equivariant convolutions, one would expect a more generalized model. We therefore also list the differences in performance between the train and test sets in Table 4.
We generally see lower relative differences between train and test scores for group equivariant networks than traditional networks. However, for the non-residual versions, the improvement in generalization (i.e. lower relative difference between train and test performance) is negligible. For the residual networks, the improvement in generalisation by using group equivariant networks is higher.

Additionally, residual networks appear to perform less well and generalise less well than non-residual networks. Both of these observations are surprising as we see no reason for group equivariance and residual learning to conflict. Indeed, due to the nature of residual learning, we would have expected at least equal performance. Additional experiments are required to determine whether residual learning and group equivariance conflict, whether residual learning is not suitable for this specific task or whether the sub-par performance of residual learning in our experiments is caused by properties of our dataset.

4.2 Filters

First Layer Weights By visualizing the filters in the first convolutional layer of the small non-residual networks, we can see some interesting effects. Figure 4 visualizes the weights of the first layer of `FCN20` and `GFCN5` in the first row. First of all, we observe in both the group equivariant network as well as in the traditional convolutional network, edge and corner contrast detecting filters. Additionally, we observe the existence of multiple similar filters in multiple poses in the traditional network, in some cases with counterparts in the group equivariant network. However, we do not always see all 4 poses of a filter in the non group-equivariant network.

Most interestingly however, the filters of the group equivariant network do not only respond to intensity contrasts, but also to color contrasts. For example, the



(a) `GFCN5` positive weights

(b) `FCN20` positive weights

(c) `GFCN5` negative weights

(d) `FCN20` negative weights

(e) `GFCN16` positive weights

(f) `FCN64` positive weights

(g) `GFCN16` negative weights

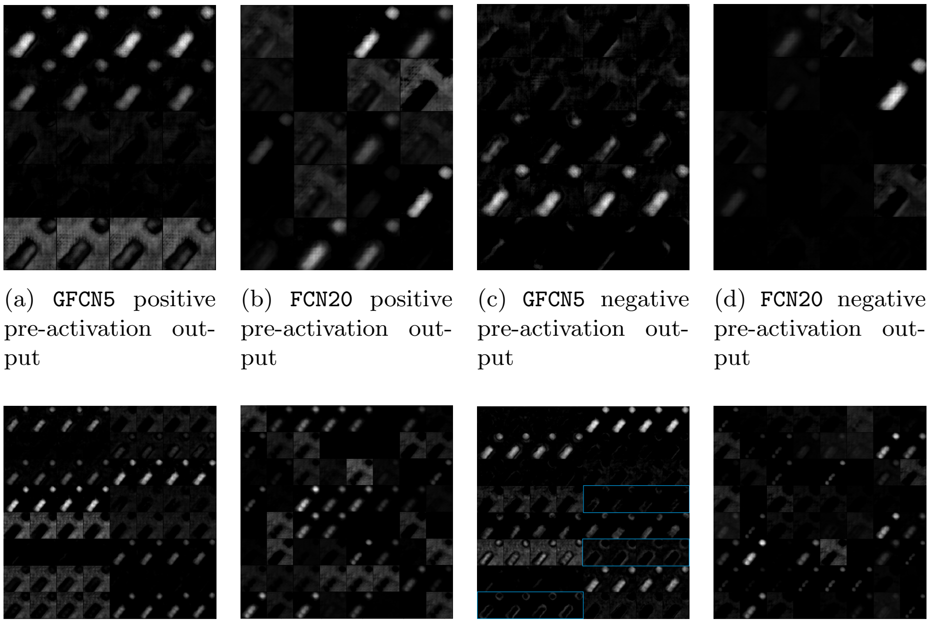
(h) `FCN64` negative weights

Fig. 4: Visualization the weights of the first convolutional layer. Top row: `GFCN5` and `FCN20`. Bottom row: `GFCN16` and `FCN64`. In (a), (b), (e) and (f) we see the positive weights. In (c), (d), (g) and (h) we see the negative weights.

group of filters marked with yellow respond strongly to a bright-red to dim-blue diagonal contrast and the group of filters marked with red respond strongly to a bright-purple to dim-green horizontal and vertical contrast.

Similarly, the filters of the first layer of `GFCN16` and `FCN64` are visualized in the same Figure in the bottom row. We see the phenomena observed in the small versions also in the large version. Additionally, variations of most of the filters in `GFCN5` are also visible in `GFCN16`. However, in contrast to our observations of the smaller networks, `FCN64` appear to have some filters with both intensity and color contrasts but also contains filters that respond to color yet without an apparent structure.

Finally, although we see clear, structural contrasts in both intensity and color in the group equivariant networks, a large number of filters in the traditional networks does not exhibit this clear structure. Consequently, this makes the interpretation of filters harder. Additionally, we see that filters in group equivariant networks are more distinct compared to filters in traditional networks. Given the these observations, it is surprising that the difference in performance between group equivariant networks and traditional networks is as small as it is.



(a) `GFCN5` positive pre-activation output

(b) `FCN20` positive pre-activation output

(c) `GFCN5` negative pre-activation output

(d) `FCN20` negative pre-activation output

(e) `GFCN16` positive pre-activation output

(f) `FCN64` positive pre-activation output

(g) `GFCN16` negative pre-activation output

(h) `FCN64` negative pre-activation output

Fig. 5: Visualization the pre-activation output of the 8^{th} convolutional layer of `GFCN5`, `FCN20`, `GFCN16` and `FCN64`. In (a), (b), (e) and (f) we see positive outputs. In (c), (d), (g) and (h) we see negative outputs. The blue regions contain features that seem to inhibit PV panel installations presence.

Eighth Layer Pre-Activation Output We visualize the last feature-layer before the activation function has been applied in Figure 5.

In Figure 4e, in the top row we see five rows of almost equivalent shapes for `GFCN5` whereas we see more variation in the filters of `FCN20` in Figure 4b, although the variation is mostly in intensity. However, if we look at the inhibiting parts of the layer output in Figures 4c and 4d, we see something different. Most of the filters in `FCN20` show little inhibition. Furthermore, we some differentiation in the first, second, third and fifth rows of `GFCN5` features in the misclassified regions around the PV installations.

Similarly, the bottom row of Figure 5 is a visualization of the output of layer 8 in `GFCN16` and `FCN64`. While we see similar patterns to Figure 5 for the smaller networks, we also see a few additional features in `GFCN16` which seem to be absent in `FCN64`. Specifically, the features with blue outlines appear to mostly inhibit near the boundaries between PV and non-PV segments. However, due to the use of the ReLU activation function, this inhibition is erased after activation.

Finally, in both both rows of Figure 5 we see features that appear to indicate the presence of PV panel installations.

5 Conclusion

The results show group equivariant networks slightly outperform traditional convolutional networks on both accuracy and generalization. In contrast to the work of Cohen and Welling, the performance increase is less pronounced. Additionally, this work keeps the amount of information after each layer constant instead of the number of trainable parameters. The group equivariant networks are therefore constrained in learning ability compared to the traditional convolutional networks. Had the amount of trainable parameters been kept constant instead, the group equivariant networks would likely have performed better, increasing the performance difference. Even with this bias towards the traditional networks, they are outperformed.

Furthermore, the addition of residual blocks decreased performance. It is not directly clear why this is the case. Additional research would be required to determine the cause.

As feature representations become more abstract, abstraction of symmetry might become less useful. An ablation test, where one or more layers at the end of the network are trained without group equivariance, is required to determine what the gain in performance is of group equivariance at layers representing abstract features.

Additionally, as the dataset was annotated by a single annotator the dataset is lacking in quality. By either switching to the dataset in [1] or by using multiple annotators to increase the quality, we expect to see increased performance.

We feel potential improvements and avenues for future research lie in improved usage of local information. Either through U-Net like skip-connections or through the addition of Conditional Random Fields such as proposed by [8].

We showed group equivariant convolutions improve segmentation of photovoltaic panels in aerial photography compared to regular convolution. Although our experiment favors regular convolutions, our models still outperform them.

Code Code and other used digital resources are available at: https://gitlab.socsci.ru.nl/l.bokkers/thesis

References

1. Bradbury, K., Saboo, R., Malof, J., Johnson, T., Devarajan, A., Zhang, W., Collins, L., Newell, R., Streltso, A.: Distributed solar photovoltaic array location and extent data set for remote sensing object identification. https://dx.doi.org/10.6084/m9.figshare.3385780 (2016), accessed: 2019-03-12

2. Camilo, J., Wang, R., Collins, L.M., Bradbury, K., Malof, J.M.: Application of a semantic segmentation convolutional neural network for accurate automatic detection and mapping of solar photovoltaic arrays in aerial imagery. arXiv preprint arXiv:1801.04018 (2018)
3. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: International conference on machine learning. pp. 2990–2999 (2016)
4. Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning. ArXiv e-prints (mar 2016)
5. 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. pp. 770–778 (2016)
6. Li, R., Liu, W., Yang, L., Sun, S., Hu, W., Zhang, F., Li, W.: Deepunet: a deep fully convolutional network for pixel-level sea-land segmentation. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **11**(11), 3954–3962 (2018)
7. Malof, J.M., Collins, L.M., Bradbury, K.: A deep convolutional neural network, with pre-training, for solar photovoltaic array detection in aerial imagery. In: 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). pp. 874–877. IEEE (2017)
8. Paisitkriangkrai, S., Sherrah, J., Janney, P., Van-Den Hengel, A.: Effective semantic pixel labelling with convolutional networks and conditional random fields. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2015)
9. PDOK: Open-access dutch aerial photography. https://www.pdok.nl/introductie/-/article/luchtfoto-pdok (page in Dutch) (2019), last Accessed: 2019-07-14
10. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
11. Sherrah, J.: Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. arXiv preprint arXiv:1606.02585 (2016)
12. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
13. Solar Solutions Int.: Nationaal solar trendrapport 2019. http://www.solarsolutions.nl/solar-trendrapport/ (2018), last accessed: 2019-03-12
14. Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015)