Detection, Classification and Boundary Regularization of Buildings in Satellite Imagery Using Faster Edge Region Convolutional Neural Networks

Kinga Reda and Michal Kedzierski *

Institute of Geospatial Engineering and Geodesy, Faculty of Civil Engineering and Geodesy, Military University of Technology, 00-908 Warsaw, Poland; kinga.reda@student.wat.edu.pl
* Correspondence: michal.kedzierski@wat.edu.pl; Tel.: +48-261-83-90-76

Received: 9 June 2020; Accepted: 9 July 2020; Published: 13 July 2020

Abstract: With the development of effective deep learning algorithms, it became possible to achieve high accuracy when conducting remote sensing analyses on very high-resolution images (VHRS), especially in the context of building detection and classification. In this article, in order to improve the accuracy of building detection and classification, we propose a Faster Edge Region Convolutional Neural Networks (FER-CNN) algorithm. This proposed algorithm is trained and evaluated on different datasets. In addition, we propose a new method to improve the detection of the boundaries of detected buildings. The results of our algorithm are compared with those of other methods, such as classical Faster Region Convolution Neural Network (Faster R-CNN) with the original VGG16 and the Single-Shot Multibox Detector (SSD). The experimental results show that our methods make it possible to obtain an average detection accuracy of 97.5% with a false positive classification rate of 8.4%. An additional advantage of our method is better resistance to shadows, which is a very common issue for satellite images of urban areas. Future research will include designing and training the neural network to detect small buildings, as well as irregularly shaped buildings that are partially obscured by shadows or other occlusions.

Keywords: convolutional neural networks; scene classification; faster R-CNN; urban land cover classification; high-resolution satellite imagery; image analysis

1. Introduction

High-resolution remote sensing satellite imagery can provide the geometric features, spatial features and textures of many objects, including various types of buildings. For many years, automatic methods of object detection and classification using very high-resolution images (VHRS) have been an important research problem. The dynamic technological development of satellite systems has made it possible to acquire images with better spatial resolution, which has led to the possibility of extracting more details of objects contained in the images, i.e., easier and more effective detection of objects in the image. Furthermore, because of their range and temporal resolution, they provide large amounts of information in a short time, and thus, they are playing an increasingly important role in updating, controlling and analyzing the spatial development of many studied areas [1–3]. This opens up new possibilities for obtaining information on the land cover of urban areas at a very detailed level [4,5]. Owing to the large amount of data transferred by satellite images, it is necessary to extract only those data that are necessary in the analysis of, for example, buildings. This ability can be the basis of a system that aims to detect buildings that are illegally erected or used contrary to the local land-use plan. Knowing the functional purpose of the structures, it is also possible to perform a statistical
analysis of the area. However, performing these analyses manually or by using frequently inaccurate classification methods is time-consuming and laborious. Therefore, research topics related to the development of new and intelligent methods of classification of land cover, especially in urban areas, are still valid. In addition, given the variability in the information content of satellite imagery, it is difficult to implement one effective classical classification method. Because of the benefits of detecting buildings in high-resolution satellite images, this topic has attracted many researchers. Methods of extracting object features can be distinguished according to whether they are based on data or based on a model [6]. The former relies on mathematical operations applied to a given image, without prior knowledge of what it may contain. Examples of these types of methods are various types of filtration or edge detection algorithms. Another way to detect objects is segmentation, a process in which the image is divided into specific regions that are homogeneous in terms of selected values. Neither of these methods provides an unambiguous answer about the location of buildings that are in the image, let alone their purpose.

Work on the detection of objects has significantly accelerated since the introduction of deep neural networks, which are very powerful models for more efficient classification of objects. Research efforts have developed even more dynamically since the second decade of the twenty-first century, when deep convolutional networks were introduced. These models greatly facilitate all work with images, which results in the detection of objects by means of semantic segmentation (Convolutional Network (FCN) [7], U-Net [8]), classification (AlexNet [9], Visual Geometry Group Network (VGG) [10], GoogleNet [11], Residual Network (ResNet) [12]), classification and location, and object detection using bounding boxes (Region Convolutional Neural Network (R-CNN) [13], Fast R-CNN [14], Faster R-CNN [15], Region Fully Convolutional Network (R-FCN) [16], You Only Look Once (YOLO) [16], Single-Shot Multibox Detector (SSD) [17]) or using masks (Mask R-CNN [18]). These possibilities have inspired researchers to use them to solve contemporary problems. In recent years, a lot of research has been carried out on the subject of building extraction from different types of imagery using algorithms based on segmentation [19–22] or the detection of structures [23–29]. Convolutional Neural Networks allow for the extraction of image features at the semantic level, which makes the detection of objects with various shapes and colors possible. The work of M. Vakalopoulou’s team is an example of building detection based on an ImageNet framework [22], whereas Ghandour et al. presented a building detection method with shadow verification. Algorithms have been used to detect roof tile buildings, flat building detection has been used to detect non-tile flat buildings according to shape features, and results fusion has been used to fuse and aggregate results from previous blocks [30]. The Tong Bai team uses the improved algorithm Faster R-CNN (region-based Convolutional Neural Network), which adopts DRNet (Dense Residual Network) and RoI (Region of Interest) Align to utilize texture information and to solve the region mismatch problems [31]. Another example of a solution to this problem is the work of the Evangelos Maltezos team, which used LiDAR [Light Detection and Ranging] data to detect buildings [32]. Algorithms using semantic segmentation classify each pixel in one of two cases, namely, if a pixel belongs to a building or to its surroundings. Boonpook et al. [23] proposed another approach, and they reached over 90% accuracy of building detection in UAV images using semantic segmentation. Ji, S. et al. presented a CNN-based two-part structure used for a change detection framework for locating changed building instances as well as changed building pixels from very high-resolution (VHR) aerial images [33]. Li W. et al. propose a U-Net-based semantic segmentation method for the extraction of building footprints from high-resolution multispectral satellite images using the SpaceNet building dataset [34]. However, these methods do not solve the problem of detection and classification of buildings due to their purpose.

This article (1) presents how to use convolutional neural networks for the automatic detection and classification of buildings, (2) proposes a modification of the convolutional network known as Faster R-CNN, i.e., Faster Edge Region Convolution Neural Network (FER-CNN), (3) verifies the impact of the selected optimization method on the detection and classification of buildings, (4) verifies the detection of a building’s shape based on Mask R-CNN, (5) determines a new method of
correcting the shape of detected buildings while maintaining good results of building categorization by using the Ramer–Douglas–Peucker (RDP) algorithm and (6) proposes a new method of building boundary regularization.

This paper is structured as follows. In Section 2, the research method is explained. In Section 3, the test data and results are presented. Section 4 presents the discussion. Finally, Section 5 provides a brief summary of this work.

2. Methodology

The proposed research methodology took into account the use of convolutional neural networks for the detection and classification of buildings in satellite imagery of urban and suburban areas. A new method of boundary correction of the detected buildings is also presented. First, the performances of Faster Region Convolution Neural Network (Faster R-CNN), modified by the authors and called Faster Edge Region Convolution Neural Network (FER-CNN), and Single-Shot Multibox Detector (SSD) were compared in the detection and classification of structures; in this analysis, the focus was on the choice of network training parameters because they play a significant role in the successful and effective training of the network and, therefore, in the attainment of results with the best accuracy. For this purpose, a brief analysis of the impact of the optimization algorithm on the results was carried out. The aim of optimization is to find the extremum of the set objective function, i.e., the optimal solutions to the problems posed. These algorithms differ from classic optimization methods. They work on the principle of indirect optimization of the performance of the trained models by reducing the cost function, which will minimize the expected error, referred to as risk. The reason for using indirect optimization is that only a training database is available, which results in a lack of knowledge about the distribution of generated data.

Because there is a need to know the exact location of buildings within the imagery, the operation capabilities of Mask R-CNN were tested, and as a result, buildings were detected by means of polygons with shapes similar to those of the buildings. By means of additional processing of the obtained results, a correction of the building’s shape was performed using the RDP algorithm, and a new method of performing building boundary regularization was developed.

2.1. Faster R-CNN

The work of Faster R-CNN (Faster Region Convolution Neural Network) can be divided into two stages [15]. In the first stage, the input data are processed using the feature extractor and the VGG16 model [10]. The result of this process is a map of features that is used in the next stage. In this part, Faster R-CNN consists of two networks. The first is the Region Proposal Network (RPN), which is responsible for generating regions (called a region proposal), on the basis of which the second network performs structure detection. This is directed at those regions that most likely contain objects. As a result, the time needed to generate the proposed regions is reduced (Figure 1).
2.2. Our Method: Faster Edge Region CNN

FER-CNN (Figure 2) was used to train the network, in which our own convolutional network was used, generating a feature map. This network consists of six modules, followed by a Maxpooling and Dropout layer. In addition, the parametric PReLU [Parametric Rectified Linear Unit] activation function was used.

2.3. Single-Shot MultiBox Detector Network

The second stage of research on the impact of the optimization method on the detection and classification of structures was to train the model based on the Single-Shot MultiBox Detector (SSD) network [17]. This model was proposed by Christian Szegedy’s team in 2016 [11] to increase the efficiency and precision of structure detection. It is based on the VGG-16 architecture; however, the fully
connected layers were replaced by additional convolution layers, enabling the detection of structures of different sizes and the gradual reduction of the output data size for subsequent layers (Figure 3).

![SSD Detector Architecture](image)

**Figure 3.** Single-Shot Multibox Detector (SSD) detector architecture.

### 2.4. Mask R-CNN

Mask R-CNN is a deep neural network designed for instance segmentation of a structure, which, unlike Faster R-CNN and SSD, not only provides information about the bounding box but also inlays a mask on the image similar in shape to the outline of the object [35]. This architecture differs significantly from the previously described methods, and it works in two stages: first, it generates suggestions for regions in which the structure may be located; next, it predicts the class of the object, corrects the coordinates of the bounding box and then creates a mask at the pixel level.

By using Mask R-CNN, it is possible to detect a structure in the image and obtain its approximate shape. The disadvantage of this action is that the building is represented with a mask, so the exact location of its edges and their coordinates are unknown. In order to solve this problem, we propose a method that makes it possible to detect buildings in a satellite image while maintaining its membership to one of the defined categories.

This algorithm displays structures belonging to one class in the image and then uses the Ramer–Douglas–Peucker algorithm [36] to minimize the number of points that create the building contour. This operation is performed for each of the seven classes, and the result of the algorithm is an image that shows the boundaries of buildings (while maintaining their classification) and a set of image coordinates of the structure. As a result, we obtain a building contour that consists of a smaller number of points (Figure 4).
Figure 4. Scheme of the building edge detection algorithm.

2.5. Building Boundary Regularization Method

Given the nature of those structures, which are buildings, it can be seen that, in most cases, the adjacent walls of the building form a right angle with each other. On the basis of the above, we propose using the following algorithm to correct the detected edges while remembering that wall projections can also take other shapes (e.g., arches) (Figure 5).
The proposed method takes the first pair of points as the “base” edge $P_nP_{n+1}$ and then checks the angle that it creates with the next edge $P_{n+1}P_{n+2}$. If the sine value for this angle is in the range from 0.9925 to 1.0000, it is assumed that the angle error is less than 7° (condition I); it corrects this edge by leading a straight line perpendicular to the base edge that passes through the $P_{n+1}$ vertex.

Figure 5. Scheme of the proposed building boundary regularization method.
leading a straight line perpendicular to the base edge that passes through the \( P_{n+1} \) vertex, and then it projects the point \( P_{n+2} \) onto this straight line to create point \( P'_{n+2} \) (substitutes the coordinates of \( P_{n+2} \) with \( P'_{n+2} \)). As a result of this action, a new, corrected building edge is created with a beginning at \( P_{n+1} \) and an end at \( P_{n+2} \) (Figure 6).

![Building boundary regularization example.](image)

Figure 6. Building boundary regularization example.

Note that not all building edges intersect at right angles. Therefore, the condition in which the angle sine value is in the range from 0.1219 to 0.9925 (condition II) is considered. For this case, the length of the \( P_{n+1}P_{n+2} \) segment is checked first, and if it is less than 5 pixels (in the case of a pixel size equal to 0.5 m, it is a length of less than 2.5 m), this indicates the “truncation” of a corner of the building. This phenomenon is very often found as a result of mask rounding on the corners. In such an instance, the program determines the corner of the building at the point of intersection of the lines that pass through the points \( P_n, P_{n+1} \) and \( P_{n+2}, P_{n+3} \) (it is first checked to determine whether these lines form an angle of 90° ± 7° with each other). If the length of the edge is longer than 5 pixels, the algorithm checks the next pair of edges; if it is less than five, it does not make corrections, but if it is at least 5, it approximates on the basis of these points (Equation (1)). This program also checks to determine whether there is a case in which all the angles of the figure will meet this condition. If so, then the program inscribes them in an ellipse (Equation (2)).

\[
\varphi(x) = a_0 + a_1x + a_2x^2 + \ldots + a_mx^m = \sum_{k=0}^{m} a_kx^k \tag{1}
\]

\[
\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{2}
\]

Another condition that the program checks is the case in which the angle value is between 0.0000 and 0.1219 (condition III). If this condition is met, it checks the distance of the point \( P_{n+1} \) from the line passing through the points \( P_n, P_{n+2} \). If this distance is less than 5 pixels, the program removes the point \( P_{n+1} \).

On the basis of the above-mentioned conditions, the algorithm checks all the edges of the figure, which is the first iteration (this algorithm performs three iterations because the differences in the shape of the building outline for a larger number is insignificant) (see Figure 7). In the case of a different spatial resolution, a correction of the distance between points and the straight line should be made.
This work explores the potential of convolution networks in the detection and classification of buildings in satellite images of urban and suburban areas.

All calculations included in this work were carried out using a PC with an Intel Core i5 CPU, NVIDIA 2070 RTX processor, 16 GB RAM memory and Ubuntu 16.04 operating system. In order to prepare images for network training, the Rasterio (v. 1.1.3), OpenCV (v. 4.1.0) and Numpy (v. 1.16.1) libraries were mainly used, while the Keras (v. 2.2.4) and TensorFlow (v. 1.13.1) libraries were used to implement neural network models.

First, a comparison was performed between Faster R-CNN, FER-CNN, and SSD in terms of their abilities to detect and classify structures, with a focus on the choice of network training hyperparameters because they play a very important role in successful and effective training and, therefore, also in the attainment of results with the best accuracy. The hyperparameters include the number of filters in the convolution layer and the activation function. The ReLU [Rectified Linear Unit] activation function was used in order to train Faster RCNN and SSD networks, while the parametric version of this function (PReLU) was used for Faster Edge Region CNN. Each model was trained for 200 epochs with a batch size of 2. For this purpose, a brief analysis of the impact of the optimization algorithm on the results was carried out. The aim of the optimization is to find the extremum of the set objective function, i.e., the optimal solutions to the problems posed. These algorithms differ from classic optimization methods. They work on the principle of indirect optimization of the performance of trained models by reducing the cost function, which will minimize the expected error, referred to as risk. The reason for using indirect optimization is that only a training database is available, which results in a lack of knowledge about the distribution of generated data. Overall, the optimizer’s task is to adjust the network with data from the loss function. The optimizer allows the best possible efficiency to be obtained for training data. Shortening the process of learning could be achieved by the appropriate choice of hyperparameter. Examples of such optimizers are Momentum, RMSProp and Adam.

Because there is a need to determine the exact location of buildings in the images, the Mask R-CNN performance was tested. With this mask, the buildings were detected by means of polygons with shapes similar to the buildings’ structures. By means of additional processing of the obtained results, the building’s shape was corrected using the RDP algorithm and the algorithm proposed in Section 2.4.

3.1. Experiment Data

The database was created on the basis of satellite images that were obtained using WorldView-2 and Pléiades (see Table 1) satellites and show a fragment of the city of Warsaw and its periphery (Poland).
Table 1. Technical characteristics of WorldView-2 and Pléiades satellites.

| Satellite       | Launch Date   | Orbital Altitude [km] | Pan Resolution at Nadir [m] | MS Resolution at Nadir [m] | Swath Width [km] |
|-----------------|---------------|-----------------------|-----------------------------|-----------------------------|------------------|
| WorldView-2     | 8 October 2009| 770                   | 0.46                        | 1.8                         | 16.5             |
| Pléiades        | 2 December 2012| 695                  | 0.50                        | 2.0                         | 20               |

The studied area covers the western part of Warsaw and is located in a rectangle limited by the 20°52’19” E, 21°00’07” E meridians and the 52°14’05” N, 52°17’58” N latitudes (Figure 8) in the WGS-84 reference system in the UTM system. From this image, six areas with different urban characteristics are distinguished: (1) apartment blocks with garages, (2) a block of flats with small shops, (3) diversified cases with occlusion, (4) dense block buildings, (5) structures shaded by trees or images with low contrast between buildings and the surroundings and (6) dense single-family houses and terraced buildings.

![Figure 8. Localization of the study area.](image)

Before performing operations on the image, in order to increase their resolution, pan sharpening was performed, which increased the resolution of the multispectral image to 0.5 m. These images were divided into smaller ones that measure 512 × 512. We used our own script for this, which divides the image into smaller parts with a vertical step of 250 and horizontal step of 350 pixels. As a result, a database was created consisting of 500 images with red, green, and blue channels and a spatial resolution of 0.5 m, which was divided into three sets of data: training (350 images), validation (75 images) and test (75 images).

In the resulting database, about 12,500 buildings of various types were marked with LabelImg [37]. They were divided into six categories: shopping center, block of flats, church, terraced houses, single-family house and garage (see Figure 9).
Figure 9. Database components.

Selected types of objects differ from each other by many parameters, including size, height, shape and roof (along with the installations located on it). A crucial feature that allows buildings to be distinguished is their size, which is why Figure 10 shows the difference between the horizontal dimensions of sample buildings in each category. The largest building of the featured classes is the shopping center, while the smallest is the garage (see Figure 10).

![Diagram comparing the lengths of buildings.](image)

Figure 10. Comparing the lengths of buildings.

Architectural variability within specific classes is an additional difficulty in classification. This phenomenon is very clearly visible among single-family homes. These structures can have different dimensions, shape, number of floors, or even a different type of structure. In the case of terraced buildings (or garages), another difficulty is the occurrence of different values for the ratio of the length of the sides of the structure because it can consist of either three or ten segments (see Figure 11).

![Image showing architectural variability.](image)

Figure 11. Cont.
The number of image pixels also affects the classification of buildings. In the WorldView-2 images, a small single-family house with dimensions of $8 \times 10$ m is made up of about 320 pixels, which makes it possible to identify many elements on the surface of the roof (installations, chimneys) and to determine the type of roof. In the case of images with lower resolution, e.g., acquired using the Ikonos satellite (where the resolution of the images is 1 m), the same building will consist of about 80 pixels, which makes it impossible to identify the details of the roof, and so models trained on the basis of these data cannot cope with classification.
The similarity among classes is another problem. This phenomenon is visible in the case of terraced houses and garages. These buildings can be distinguished from each other by the details on their roofs. In the case of images with lower resolution (or quality), the differences between these categories are small, which is why their proper identification on the basis of their surroundings is possible (see Figure 12). Classical methods of building detection in satellite images are very time-consuming and do not allow for division by their destination. Although edge detection or segmentation techniques provide the possibility to detect buildings, they require particular parameter choices for each image. In our method, data preparation for the algorithm and training are the most time-consuming, and when using the network, achieving results depends on image import.

Figure 12. Example of building similarity.

In addition, it should be noted that because of their diverse architecture, these types of structures are characterized highly variable shapes, and besides this, they are often accompanied by shadows, occlusion, noise, deformations, variable lighting or different resolutions (see Table 2). These factors have a significant impact on the algorithms’ performance, especially in urban areas, where building density is very high. In this case, an often-occurring phenomenon is the overlapping of building shadows on neighboring objects, or the “laying of buildings”, which can lead to the merging of many buildings into one.

Table 2. Factors influencing the accuracy of building detection and classification.

| Deformation Type | Occlusion | Lighting |
|------------------|-----------|----------|
|                   | ![Occlusion](image1.png) | ![Lighting](image2.png) |

|                    | Background | Scale |
|--------------------|------------|-------|
| ![Background](image3.png) | ![Scale](image4.png) |
When working with satellite imagery, these structures are identified only on the basis of roofs and the surroundings. In this case, one must pay attention to the heterogeneity of the roofs and the possibility of additional installations being present on them (which is possible, but not necessary).

### 3.2. Accuracy Assessment

To assess the accuracy of the detection and classification of structures, three parameters can be used: accuracy detection (AD) (Equation (3)), missing ratio (MR) (Equation (4)), and false-positive classification rate (FPCR) (Equation (5)).

\[
AD = \frac{\text{number of structures detected}}{\text{number of all structures}} \tag{3}
\]

\[
MR = \frac{\text{number of undetected structures}}{\text{number of all structures}} \tag{4}
\]

\[
FPCR = \frac{\text{number of incorrectly classified structures}}{\text{number of all classified structures detected}} \tag{5}
\]

### 3.3. Experimental Results

#### 3.3.1. Results of Faster R-CNN

First, three Faster R-CNN networks were trained using three optimization methods: Adam, Momentum and RMSProp. The training process was carried out for a constant number of iterations. The results of the optimization are presented in Table 3. The table presents a record of structure detectability in images. This was used to check the correct operation of the models. The assessment was made using parameters that determine the quality of building detection and classification. Detailed assessment results are provided in the Appendix A. Tables A1–A3 (see Appendix A) compare five parameters that determine the correctness of the network operation: detected objects (W), true objects (T), false objects (F), undetected objects (B) and the number of buildings in the image (S). These parameters allow for an assessment of an algorithm’s results. The main parameters are detected objects, true objects and the number of buildings in the image. Their application shows that the Adam and RMSProp algorithms yield good results (see Table 3). The second of these methods achieves slightly
better results for images 1 and 3, but the Adam algorithm is unrivaled compared with the others: in the case of visual analysis, it correctly determines the location of structures in the image.

The classification process of detected structures is also performed most accurately when using the Adam method. In addition, it can be seen that the most recognizable structures are the block of flats, while garages are the least recognizable.

Table 3. Detection of structures using the Faster R-CNN architecture.
FPCR: False-Positive Classification Rate

From the three parameters that define the accuracy of detection (Table 4), it can be seen that the Adam algorithm provides the best results in the second, fourth, fifth and sixth set, while RMSProp does better in other cases. In addition, the correctness of detecting individual building types was examined. The fewest errors in the classification of buildings occur when using the Adam algorithm. The exception is set 3, for which better results are obtained with the RMSProp algorithm.

3.3.2. Result of Our Method, FER-CNN

To be able to compare the results of these two networks, the same assumptions were made: the same image database and the same set of images for visual assessment were used, and the duration of both networks’ training was equal to 200,000 steps.

The results of the program operation, depending on the optimization used, are presented in the tables below (see Tables 5 and 6). Significant improvement is visible when fitting the structures into a bounding box, so they contain less shadowing of buildings (Table 5). In addition, as in the first case, the best results in the detection and classification of structures occur for the Adam and RMSProp optimizations.
Table 4. Accuracy assessment of the model’s operation: Faster R-CNN.

|        | AD  | MR  | FPCR | Correctness of Class Detection |
|--------|-----|-----|------|-------------------------------|
|        | G   | T   | F    | B    | CH  | S    |
| Set 1  |     |     |      |      |     |     |
| A      | 0.72| 0.34| 0.12 | 0.36 | 0.60| 0.63 | 0.88 | -  | -  |
| M      | 0.85| 0.40| 0.41 | 0.36 | 0.20| 0.50 | 0.79 | -  | -  |
| R      | 0.91| 0.28| 0.33 | 0.55 | 0.50| 0.50 | 0.88 | -  | -  |
| Set 2  |     |     |      |      |     |     |
| A      | 0.91| 0.13| 0.11 | -    | -  | -    | 0.90 | -  | 0.50|
| M      | 0.87| 0.22| 0.11 | -    | -  | -    | 0.90 | -  | 0.50|
| R      | 0.96| 0.13| 0.10 | -    | -  | -    | 1.00 | -  | 0.50|
| Set 3  |     |     |      |      |     |     |
| A      | 0.88| 0.19| 0.27 | -    | 0.67| 1.00 | 0.86 | 0.50| 1.00|
| M      | 0.69| 0.38| 0.38 | -    | 0.67| 1.00 | 0.43 | 0.50| 1.00|
| R      | 0.88| 0.13| 0.17 | -    | 0.67| 1.00 | 1.00 | 0.50| 1.00|
| Set 4  |     |     |      |      |     |     |
| A      | 0.86| 0.14| 0.00 | -    | -  | -    | 0.88 | -  | 0.50|
| M      | 0.77| 0.23| 0.00 | -    | -  | -    | 0.80 | -  | 1.00|
| R      | 0.81| 0.23| 0.06 | -    | -  | -    | 0.80 | -  | 0.00|
| Set 5  |     |     |      |      |     |     |
| A      | 1.00| 0.15| 0.18 | -    | -  | -    | 0.89 | -  | 0.50|
| M      | 0.95| 0.20| 0.13 | -    | -  | -    | 0.83 | -  | 1.00|
| R      | 1.00| 0.15| 0.18 | -    | -  | -    | 0.89 | -  | 0.50|
| Set 6  |     |     |      |      |     |     |
| A      | 0.99| 0.15| 0.16 | 0.69 | 0.94| -    | -  | -  | -  |
| M      | 0.89| 0.40| 0.48 | 0.03 | 0.92| -    | -  | -  | -  |
| R      | 0.94| 0.19| 0.16 | 0.61 | 0.92| -    | -  | -  | -  |

A: Adam optimizer; M: Momentum; R: RMSProp optimizer; AD: accuracy detection; MR: missing ratio; FPCR: false-positive classification rate; G: garage; T: terrace house; F: single-family house; B: block of flats; CH: church; S: shopping center.

Table 5. Detection of structures using the Faster Edge Region Convolution Neural Network (FER-CNN) architecture.
Table 5. Cont.

| FER-CNN       | Adam   | Momentum | RMSProp |
|---------------|--------|----------|---------|
| shop          | ![Image](image1.png) | ![Image](image2.png) | ![Image](image3.png) |
| terrace house | ![Image](image4.png) | ![Image](image5.png) | ![Image](image6.png) |
| block of flats| ![Image](image7.png) | ![Image](image8.png) | ![Image](image9.png) |
| family house  | ![Image](image10.png) | ![Image](image11.png) | ![Image](image12.png) |
| church        | ![Image](image13.png) | ![Image](image14.png) | ![Image](image15.png) |
| garage        | ![Image](image16.png) | ![Image](image17.png) | ![Image](image18.png) |
Table 6. Accuracy assessment of the model’s operation: FER-CNN.

|       | AD  | MR  | FPCR | Correctness of Class Detection |       |
|-------|-----|-----|------|--------------------------------|-------|
|       |     |     |      | G    | T   | F   | B   | CH  | S    |       |
| Set 1 | A   | 0.87| 0.15 | 0.13 | 0.45 | 0.60 | 0.63 | 0.96 | -    | -    |
| M     | 0.89| 0.15| 0.23 | 0.64 | 0.20 | 0.75 | 0.83 | -    | -    |
| R     | 0.94| 0.09| 0.06 | 0.64 | 0.30 | 0.88 | 0.92 | -    | -    |
| Set 2 | A   | 1.04| 0.04 | 0.09 | -   | -   | -   | 1.00 | -    | 0.50 |
| M     | 0.83| 0.22| 0.04 | -   | -   | -   | 0.81 | -    | 0.50 |
| R     | 0.96| 0.09| 0.04 | -   | -   | -   | 0.95 | -    | 0.50 |
| Set 3 | A   | 1.00| 0.21 | 0.21 | -   | 0.67 | 1.00| 0.86 | 0.50 | 1.00 |
| M     | 0.79| 0.43| 0.21 | -   | 0.67 | 1.00| 0.43 | 0.50 | 1.00 |
| R     | 1.00| 0.14| 0.14 | -   | 0.67 | 1.00| 1.00 | 0.50 | 1.00 |
| Set 4 | A   | 0.91| 0.09 | 0.00 | -   | -   | -   | 0.93 | -    | 0.50 |
| M     | 0.84| 0.16| 0.00 | -   | -   | -   | 0.88 | -    | 0.00 |
| R     | 0.84| 0.21| 0.05 | -   | -   | -   | 0.83 | -    | 0.00 |
| Set 5 | A   | 1.05| 0.15 | 0.10 | -   | -   | -   | 0.89 | -    | 0.50 |
| M     | 0.95| 0.20| 0.15 | -   | -   | -   | 0.83 | -    | 0.50 |
| R     | 1.00| 0.15| 0.15 | -   | -   | -   | 0.89 | -    | 0.50 |
| Set 6 | A   | 1.00| 0.11 | 0.11 | -   | 0.75 | 0.97| -    | -    | -    |
| M     | 0.91| 0.37| 0.28 | -   | 0.00 | 0.98| -    | -    | -    |
| R     | 0.98| 0.14| 0.12 | -   | 0.64 | 0.98| -    | -    | -    |

A: Adam optimizer; M: Momentum; R: RMSProp optimizer; AD: accuracy detection; MR: missing ratio; FPCR: false-positive classification rate; G: garage; T: terrace house; F: single-family house; B: block of flats; CH: church; S: shopping center.

FPCR: False-Positive Classification Rate

Owing to the modification of the VGG-16 network architecture, significant improvements in the results are noticeable. There is an increase in the accuracy detection parameter (more buildings were detected), and a significant reduction in classification error can be observed. The program operating in this network model classifies structures significantly better, especially those that are particularly vulnerable to errors due to the similarity between classes, e.g., the side-by-side character of the construction of garages (Set 1).

3.3.3. Result of Single-Shot MultiBox Detector (SSD)

In the case of SSD, much better results are achieved for the RMSProp and Momentum algorithms. After applying the accuracy rating parameters, it can be seen that the first one achieves better results. Image 1 is an exception, for which the Momentum model is characterized by a slightly higher detection of buildings (Table 7).
Table 7. Detection of structures using the SSD architecture.

| Single-Shot MultiBox Detector | Adam | Momentum | RMSProp |
|-------------------------------|------|----------|---------|
| shop                          | ![Adam shop](image1) | ![Momentum shop](image2) | ![RMSProp shop](image3) |
| terrace house                 | ![Adam terrace house](image4) | ![Momentum terrace house](image5) | ![RMSProp terrace house](image6) |
| family house                  | ![Adam family house](image7) | ![Momentum family house](image8) | ![RMSProp family house](image9) |
| church                        | ![Adam church](image10) | ![Momentum church](image11) | ![RMSProp church](image12) |
| block of flats                | ![Adam block of flats](image13) | ![Momentum block of flats](image14) | ![RMSProp block of flats](image15) |
| garage                        | ![Adam garage](image16) | ![Momentum garage](image17) | ![RMSProp garage](image18) |
| shopping center               | ![Adam shopping center](image19) | ![Momentum shopping center](image20) | ![RMSProp shopping center](image21) |

In the case of SSD, much better results are achieved for the RMSProp and Momentum algorithms. After applying the accuracy rating parameters, it can be seen that the first one achieves accurate in terms of classification, but less accurate for structure detection. In the case of Faster R-CNN, while the number of incorrectly detected in the image than that in the case of Faster R-CNN, while the number of incorrectly...
After using the SSD model, one can notice a much larger number of buildings that are not detected in the image than that in the case of Faster R-CNN, while the number of incorrectly classified structures decreases (Table 8). Thus, it can be concluded that the SSD model is more accurate in terms of classification, but less accurate for structure detection.

Table 8. Accuracy assessment of the model's operation: SSD architecture.

| Set     | AD (Accuracy) | MR (Missing Ratio) | FPCR (False-Positive Classification Rate) | Correctness of Class Detection |
|---------|---------------|--------------------|------------------------------------------|--------------------------------|
|         | G  | T  | F  | B  | CH | S  |
| Set 1   | A  | 0.09 | 1.00 | 0.11 | 0.00 | 0.00 | 0.00 | - | - |
|         | M  | 0.72 | 0.40 | 0.11 | 0.00 | 0.60 | 0.63 | 0.88 | - | - |
|         | R  | 0.68 | 0.43 | 0.11 | 0.00 | 0.60 | 0.50 | 0.83 | - | - |
| Set 2   | A  | 0.17 | 1.00 | 0.17 | - | - | - | 0.00 | - | 0.00 |
|         | M  | 0.91 | 0.17 | 0.09 | - | - | - | 0.86 | - | 0.50 |
|         | R  | 0.91 | 0.13 | 0.04 | - | - | - | 0.95 | - | 0.00 |
| Set 3   | A  | 0.21 | 0.93 | 0.00 | - | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 |
|         | M  | 0.79 | 0.36 | 0.14 | - | 0.33 | 0.00 | 0.86 | 0.00 | 1.00 |
|         | R  | 0.86 | 0.29 | 0.14 | - | 0.67 | 0.00 | 0.86 | 0.50 | 1.00 |
| Set 4   | A  | 0.05 | 0.95 | 0.05 | - | - | - | 0.00 | - | 0.00 |
|         | M  | 0.70 | 0.28 | 0.02 | - | - | - | 0.71 | - | 0.00 |
|         | R  | 0.74 | 0.26 | 0.00 | - | - | - | 0.78 | - | 0.00 |
| Set 5   | A  | 0.10 | 0.95 | 0.00 | - | - | - | 0.00 | - | 0.50 |
|         | M  | 0.60 | 0.40 | 0.00 | - | - | - | 0.61 | - | 0.50 |
|         | R  | 0.65 | 0.25 | 0.00 | - | - | - | 0.72 | - | 0.00 |
| Set 6   | A  | 0.04 | 1.00 | 0.00 | - | 0.00 | 0.00 | - | - | - |
|         | M  | 0.83 | 0.37 | 0.20 | - | 0.28 | 0.83 | - | - | - |
|         | R  | 0.70 | 0.38 | 0.08 | - | 0.36 | 0.77 | - | - | - |

A: Adam optimizer; M: Momentum; R: RMSProp optimizer; AD: accuracy detection; MR: missing ratio; FPCR: false-positive classification rate; G: garage; T: terrace house; F: single-family house; B: block of flats; CH: church; S: shopping center.
In the case of SSD architecture, the accuracy detection parameter significantly worsens (depending on the image; for the RMSProp algorithm, it ranges from 0.65 to 0.91), which is also true for the missing ratio (for the same optimization method, it ranges from 0.25 to 0.40). Because fewer buildings are detected, there is a slight improvement in the classification error parameter, whose maximum value is 0.20.

3.3.4. Google Earth Image Database

In order to check the correctness of our method, the additional database was prepared, which consist of photos fragments from Google Earth. These photos show cities of different size, density and type of buildings—among the selected cities are Koszalin, Żyrardów, Opole and Suwałki. A database consists of 500 images, of different resolutions (the smallest image has a resolution of $241 \times 306$ pixels, and the largest- $856 \times 1144$ pixels), which were divided into two sets of data: training (400 images) and test (100 images). These images mark over 2200 buildings, which, as in the previous database, were divided into six categories: shopping center, block of flats, church, terraced houses, single-family house and garage. As in previous part, the three networks were trained for each architecture (Faster R-CNN, FER-CNN, SSD), which are differ from each other by the optimization method. The results are presented in Table 9.

| Table 9. Detection of structures using the Faster R-CNN, FER-CNN, SSD architecture. |
|---|---|---|---|---|---|
| Faster R-CNN | FER-CNN | SSD |
| Adam | Momentum | RMSProp |
| shop | terrace house | block of flats | church | family house | garage |

![Image showing detection results for Faster R-CNN, FER-CNN, SSD using Adam, Momentum, RMSProp optimization methods.]
Remote Sens. 2020, 12, x FOR PEER REVIEW 22 of 31

Table 9. Cont.

| Faster R-CNN | FER-CNN | SSD |
|--------------|---------|-----|
| Adam         | ![Adam Example](image) | ![Adam Example](image) |
| Momentum     | ![Momentum Example](image) | ![Momentum Example](image) |
| RMSProp      | ![RMSProp Example](image) | ![RMSProp Example](image) |

Remote Sens. 2020, 12, x FOR PEER REVIEW 21 of 31

In the case of SSD architecture, the accuracy detection parameter was estimated for each of the three networks: Faster R-CNN, FER-CNN, and SSD. The results are presented in Table 9.

| Faster R-CNN | FER-CNN | SSD |
|--------------|---------|-----|
| Adam         | ![Adam Example](image) | ![Adam Example](image) |
| Momentum     | ![Momentum Example](image) | ![Momentum Example](image) |
| RMSProp      | ![RMSProp Example](image) | ![RMSProp Example](image) |

The accuracy detection parameter was estimated using the FER-CNN network-based algorithm with the Adam optimization method. This algorithm has a significantly higher performance compared to the other methods used in this study.
As in the previous case of the database, consisting of fragments of satellite photos from the WorldView-2 satellite, the most relevant results can be reached using the FER-CNN network-based algorithm with the Adam optimization method. This algorithm has a significantly higher object detection rate (AD = 92%), while maintaining over 95% of correctness of the buildings classification. Additionally, the FER-CNN network-based algorithm has much higher accuracy in the detection of small objects (e.g., garage), even when they are partly shaded. In case of the total shading of the object and a roof color only slightly contrasting with the surrounding area, this algorithm does not detect these objects, similar to SSD and Faster R-CNN network-based detectors.

3.3.5. Box Detection

When comparing the operation of the above methods for the detection and classification of buildings in satellite images, one can see a significant advantage of our proposed FER-CNN (Table 10). An additional advantage of this network is better resistance to shadows, which is very common in satellite images of urban areas.

Table 10. Comparing the results for Faster R-CNN, FER-CNN and SSD.

|          | Faster R-CNN | Our FER-CNN | SSD |
|----------|--------------|-------------|-----|
|          | AD | MR | FPCR | AD | MR | FPCR | AD | MR | FPCR |
| Set 1    | A  | 0.72 | 0.34 | 0.12 | 0.87 | 0.15 | 0.13 | 0.09 | 1.00 | 0.11 |
|          | M  | 0.85 | 0.40 | 0.41 | 0.89 | 0.15 | 0.23 | 0.72 | 0.40 | 0.11 |
|          | R  | 0.91 | 0.28 | 0.33 | 0.94 | 0.09 | 0.06 | 0.68 | 0.43 | 0.11 |
|          | A  | 0.91 | 0.13 | 0.11 | 1.00 | 0.04 | 0.09 | 0.17 | 1.00 | 0.17 |
|          | M  | 0.87 | 0.22 | 0.11 | 0.83 | 0.22 | 0.04 | 0.91 | 0.17 | 0.09 |
|          | R  | 0.96 | 0.13 | 0.10 | 0.96 | 0.09 | 0.04 | 0.91 | 0.13 | 0.04 |
| Set 2    | A  | 0.88 | 0.19 | 0.27 | 1.00 | 0.21 | 0.21 | 0.21 | 0.93 | 0.00 |
|          | M  | 0.69 | 0.38 | 0.38 | 0.79 | 0.43 | 0.21 | 0.79 | 0.36 | 0.14 |
|          | R  | 0.88 | 0.13 | 0.17 | 1.00 | 0.14 | 0.14 | 0.86 | 0.29 | 0.14 |
| Set 3    | A  | 0.86 | 0.14 | 0.00 | 0.91 | 0.09 | 0.00 | 0.05 | 0.95 | 0.05 |
|          | M  | 0.77 | 0.23 | 0.00 | 0.84 | 0.16 | 0.00 | 0.70 | 0.28 | 0.02 |
|          | R  | 0.81 | 0.23 | 0.06 | 0.84 | 0.21 | 0.05 | 0.74 | 0.26 | 0.00 |
| Set 4    | A  | 1.00 | 0.15 | 0.18 | 1.00 | 0.15 | 0.10 | 0.10 | 0.95 | 0.00 |
|          | M  | 0.95 | 0.20 | 0.13 | 0.95 | 0.20 | 0.15 | 0.60 | 0.40 | 0.00 |
|          | R  | 1.00 | 0.15 | 0.18 | 1.00 | 0.15 | 0.15 | 0.65 | 0.25 | 0.00 |
| Set 5    | A  | 0.99 | 0.15 | 0.16 | 1.00 | 0.11 | 0.11 | 0.04 | 1.00 | 0.00 |
|          | M  | 0.89 | 0.40 | 0.48 | 0.91 | 0.37 | 0.28 | 0.83 | 0.37 | 0.20 |
|          | R  | 0.94 | 0.19 | 0.16 | 0.98 | 0.14 | 0.12 | 0.09 | 1.00 | 0.11 |

Analyzing the comparison in Table 9, it can be seen that, in many cases, the smallest classification error occurs in the case of SSD networks. On this basis, it can be concluded that algorithms working on the basis of this network detect fewer objects, but with a smaller classification error.

3.3.6. Results of Edge Detection Using Mask R-CNN and Ramer–Douglas–Peucker (RDP)

The algorithm based on Mask R-CNN allows for the detection and classification of a structure using only 200 images in the training base. In addition, in order to check the algorithm’s ability to detect garages and small halls in the pictures, a seventh category of buildings was added—halls. It works well with objects of various sizes and those with low visibility due to low contrast with the surroundings, as well as with objects that are obscured by shadows. Moreover, this algorithm works well when it detects small structures such as garages or garden sheds (assumed to be in the garage category), as well as slightly larger warehouses (shopping centers).
However, attention should be paid to the alignment of the mask generated by the algorithm with the actual contour of the building. For structures with simple shapes, this error is much smaller than it is for structures with complex shapes, e.g., churches and shopping centers (Figure 13a,b).

![Figure 13.](image)

Figure 13. (a) Mask R-CNN (100 images); (b) Mask R-CNN (200 images); (c) algorithm based on Ramer–Douglas–Peucker (RDP); (d) our method (third iteration).
To obtain building outlines based on the generated masks, the RDP algorithm was used, followed by the proposed building boundary regularization method. Its application introduces a significant improvement in mapping the edges of buildings. In addition, as can be seen in Figure 13, distinguishing buildings with unusual architecture, such as churches or single-family terraced houses, is difficult, but our method can successfully recognize and locate them with faithful edge retention.

The algorithm we propose allows us to improve the edges generated by the RDP algorithm, but it depends on the quality of masks generated by Mask R-CNN. For Mask R-CNN, which was trained on the basis of a small database of 200 images, the method of building shape correction that we propose allows us to improve over 67% of angles on a right angle, as well as over 83% of edges. Due to these operations, the shape of the detected buildings is much closer to the real one. In the case of an increasing value of the parameter $d_{min}$ (e.g., $d_{min} = 10$) and angle $a_1$ (e.g., $a_1 = 12$), we can increase the number of correctly made corrections to 89% for angles and to 93% for edges.

4. Discussion

This article presents the results of a comparative analysis of state-of-the-art CNN-based object detection method for determining and classifying buildings, along with a new method of improving building boundary regularization. All networks used in the study were trained on the basis of a dataset consisting of 500 images with red, green, and blue channels and a spatial resolution of 0.5 m. The best results were obtained with FER-CNN. The obtained test results prove the universality of the presented approach for high-resolution satellite imagery for the detection of buildings.

Comparing the results obtained by using Faster R-CNN, FER-CNN and SSD, it can be concluded that (1) models based on the Adam algorithm achieved good results only for Faster R-CNN, while they generated errors when used in SSD networks. (2) The modification of the VGG network resulted in the better detection of structures in the image, and it was more resistant to shadows. (3) The time required to train an SSD network was approximately three times longer than that for Faster R-CNN. (4) The SSD-based model did not detect buildings that had low contrast with the surroundings, but it generated fewer errors when classifying objects. (5) The size of the files needed to run Faster R-CNN was about five times larger than that for an SSD network.

On the basis of the comparative analyses performed, the effectiveness of our method was 97.5%; however, there is a significant difference between classes. In most cases, buildings that represent categories differ significantly, including in size and appearance (e.g., garages and shopping centers). In addition, it should be kept in mind that these networks mainly make mistakes when classifying the above objects. The proposed network is most often wrong in the classification of garages (especially those in terraced houses), where the detection accuracy error is 37% and the error of classification is only 6%. Therefore, before using the presented networks for utility purposes, it is necessary to improve their algorithms in this area.

Comparing our method with classic techniques that allow for the detection of buildings, the time needed to detect the defined categories of buildings is significantly reduced. In addition, this approach does not require the selection of individual parameters for each image.

In relation to other similar studies using the Faster R-CNN model based on the ResNet101 network, researchers obtained a classification accuracy of 99% with 2000 epochs, whereas building extraction using support vector machine achieved 88.3% [38]. WorldView-2 images were also part of our test material. Tests performed on these data also confirmed our conclusions that convolutional neural networks better extract features and detect structures in high-resolution images. The training time and prediction time of our algorithm is not much longer than Faster R-CNN, which is due to the higher value of the input resolution parameter (Table 11). Other studies related to building detection have used the architecture of Res-U-N and Guided Filtering. An accuracy of 97% was achieved but without faithful reproduction of edges [21]. Other research results obtained by [39] showed that the use of an artificial neural network could produce an accuracy of 91.7%.
Table 11. Comparing the training/prediction time for Faster R-CNN, FER-CNN and SSD.

|                | Input Resolution | Training Time | Prediction Time |
|----------------|------------------|---------------|-----------------|
| Faster R-CNN   | 224 × 224        | 15 h          | 0.8 s           |
| SSD            | 300 × 300        | 35 h          | 0.9 s           |
| FER-CNN        | 512 × 512        | 19 h          | 1.0 s           |

From our research results and the analysis of the data contained in Table 10, it can be seen that our method, compared with the others investigated, makes it possible to increase the number of detected and classified buildings in fragments of satellite images. In addition, the correctness of detecting buildings using bounding boxes significantly increases. However, we believe that these results can be improved by:

- Increasing the training base by adding images of other types of areas and buildings, paying particular attention to increasing the number of images of garages, shopping centers and churches;
- Increasing the number of iterations during network training;
- Further modifying Faster R-CNN.

In the second part of the research, buildings were detected using Mask R-CNN, and then the building boundary regularization process was performed. Our proposed method significantly improved the shape of the detected boundaries. However, a certain limitation of our method (as can be seen in Figure 13) is that the identified edges deviate slightly from the actual shape of the building. The reason for this phenomenon is the fact that completely covering a structure with a mask is a significant problem. The result of this may be too small a database or too few iterations made during network training. Apart from this disadvantage, the network detects and classifies structures very well, even those with a small surface area, although it must be taken into consideration that it sometimes fails to detect small structures, which do not have sufficient contrast with their surroundings.

5. Conclusions

In this work, the capabilities of neural networks in the detection and classification of buildings located in satellite images were examined. In the first stage of research, three network models were examined: Faster R-CNN, FER-CNN and SSD; additionally, the optimization method was taken into account. For the first of these architectures, the model using the Adam algorithm achieved the best results (it gave slightly better results than the Momentum algorithm). After performing the modification of the VGG network, a significant increase in the correctness of detecting buildings in images was noted, especially for the Adam algorithm, although the RMSProp algorithm also performed very well. In the case of the SSD-based model, the best results were obtained by using the RMSProp algorithm, while the worst performance was found in the Adam algorithm.

In the second stage of this work, the capabilities of the algorithm based on Mask R-CNN for the detection and classification of objects were investigated. Then, because of the irregular shape of the training polygons, a method was proposed that allowed us to obtain an image with the edges of classified buildings marked on it.

This algorithm coped well with the detection and classification of structures from satellite imagery; however, the generated edges were slightly different from the actual shape of the building, especially in the case of buildings with a complex structure, e.g., churches. The application of the method, which performs a correction of the edges of buildings, significantly improved the rendition of their actual shapes. However, attention should be paid to the size of the database on the basis of which the Mask R-CNN model was trained: in the first case, it contained 100 images, and in the second, it contained 200 images. This shows how great the possibilities are of the application of neural networks, even in the case of such a small amount of data.
From the above, it can also be stated that building extraction is still an important and current research topic and requires further experiments. In future work, we will focus on constructing a network for more reliable building classification, improving model performance for accurate edge extraction and developing a new model for using relationships between specific groups of buildings. In addition, we plan to design and train a neural network to detect small buildings and buildings with irregular shapes that are partially obscured by shadows or other occlusions. In addition, we plan to increase the efficiency of the proposed method and increase its automation.

**Author Contributions:** Formal analysis M.K.; Methodology, K.R. and M.K.; Project administration, M.K.; Software, K.R.; Validation, M.K.; Visualization, K.R.; Writing—original draft, K.R. and M.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** This paper has been supported by the Military University of Technology, the Faculty of Civil Engineering and Geodesy.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

**Source Code and Database:** https://github.com/runnergirl13/GER-CNN.

### Appendix A

| Table A1. Summary of structure detection for Sets 1–6: Faster R-CNN architecture. |
| Set 1 | Set 2 | Set 3 | Set 4 | Set 5 | Set 6 |
|-------|-------|-------|-------|-------|-------|
| **A** | **M** | **R** | **A** | **M** | **R** | **A** | **M** | **R** | **A** | **M** | **R** | **A** | **M** | **R** | **A** | **M** | **R** |
| W     | 4     | 4     | 6     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| T     | 4     | 4     | 6     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| F     | 0     | 0     | 0     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 1     | 0     | 0     | 0     |
| B     | 7     | 7     | 5     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| S     | 11    | 11    | 11    | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| **W** | 6     | 5     | 5     | 0     | 0     | 0     | 3     | 2     | 3     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 26    | 1     |
| T     | 5     | 3     | 3     | 0     | 0     | 0     | 2     | 2     | 2     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 25    |
| F     | 1     | 2     | 2     | 0     | 0     | 0     | 1     | 0     | 1     | 0     | 0     | 1     | 0     | 0     | 1     | 0     | 0     | 1     |
| B     | 5     | 7     | 7     | 0     | 0     | 0     | 1     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 11    | 38    |
| S     | 10    | 10    | 10    | 0     | 0     | 0     | 3     | 3     | 3     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 36    | 36    |
| **W** | 7     | 12    | 14    | 0     | 0     | 0     | 2     | 3     | 2     | 0     | 0     | 0     | 2     | 2     | 3     | 73    | 88    | 71    |
| T     | 4     | 6     | 6     | 0     | 0     | 0     | 1     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 60    | 59    |
| F     | 3     | 6     | 8     | 0     | 0     | 0     | 1     | 2     | 1     | 0     | 0     | 0     | 2     | 2     | 3     | 13    | 29    | 12    |
| B     | 3     | 2     | 2     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 4     | 5     |
| S     | 8     | 8     | 8     | 0     | 0     | 0     | 1     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 64    | 64    |
| **W** | 21    | 24    | 23    | 19    | 19    | 21    | 7     | 4     | 7     | 38    | 33    | 34    | 36    | 33    | 33    | 36    | 33    | 33    | 36    |
| T     | 21    | 19    | 21    | 18    | 17    | 19    | 6     | 3     | 7     | 36    | 33    | 33    | 16    | 15    | 16    | 0     | 0     | 0     | 0     |
| F     | 0     | 5     | 2     | 1     | 2     | 2     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| B     | 3     | 5     | 1     | 2     | 4     | 2     | 1     | 4     | 0     | 5     | 8     | 8     | 2     | 3     | 2     | 0     | 0     | 0     | 0     |
| S     | 24    | 24    | 24    | 21    | 21    | 21    | 7     | 7     | 7     | 41    | 41    | 41    | 41    | 18    | 18    | 18    | 0     | 0     | 0     | 0     |
| Set 1 | Set 2 | Set 3 | Set 4 | Set 5 | Set 6 |
|-------|-------|-------|-------|-------|-------|
| A     | M     | R     | A     | M     | R     | A     | M     | R     | A     | M     | R     |
| W     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| T     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| F     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| B     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| S     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**Table A1. Cont.**

| Set 1 | Set 2 | Set 3 | Set 4 | Set 5 | Set 6 |
|-------|-------|-------|-------|-------|-------|
| A     | M     | R     | A     | M     | R     | A     | M     | R     | A     | M     | R     |
| W     | 0     | 0     | 1     | 1     | 1     | 0     | 0     | 0     | 0     | 0     | 0     |
| T     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| F     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| B     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| S     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**Table A2. Summary of structure detection for Sets 1–6: FER-CNN architecture.**

| Set 1 | Set 2 | Set 3 | Set 4 | Set 5 | Set 6 |
|-------|-------|-------|-------|-------|-------|
| A     | M     | R     | A     | M     | R     | A     | M     | R     | A     | M     | R     |
| W     | 5     | 7     | 8     | 1     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| T     | 5     | 7     | 7     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| F     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| B     | 6     | 4     | 3     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| S     | 11    | 11    | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**Church:**

| Set 1 | Set 2 | Set 3 | Set 4 | Set 5 | Set 6 |
|-------|-------|-------|-------|-------|-------|
| A     | M     | R     | A     | M     | R     | A     | M     | R     | A     | M     | R     |
| W     | 0     | 0     | 1     | 1     | 1     | 1     | 1     | 0     | 1     | 2     | 1     |
| T     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| F     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| B     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| S     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**W:** detected objects; **T:** true objects; **F:** false objects; **B:** undetected objects; **S:** number of buildings on the image; **A:** Adam optimizer; **M:** Momentum; **R:** RMSProp optimizer.
|                  | Set 1     | Set 2     | Set 3     | Set 4     | Set 5     | Set 6     |
|------------------|-----------|-----------|-----------|-----------|-----------|-----------|
|                  | A  M  R   | A  M  R   | A  M  R   | A  M  R   | A  M  R   | A  M  R   |
| church           | W  0 0 0  | 0 0 0 0   | 1 1 1 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
|                  | T  0 0 0  | 0 0 0 0   | 1 1 1 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
|                  | F  0 0 0  | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
|                  | B  0 0 0  | 0 0 0 0   | 1 1 1 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
|                  | S  0 0 0  | 0 0 0 0   | 0 0 0 0   | 2 2 2 0   | 0 0 0 0   | 0 0 0 0   |
| Shopping center  | W  0 0 0  | 1 1 1 1   | 1 1 1 1   | 0 0 1 2   | 1 0 0 0   | 0 0 0 0   |
|                  | T  0 0 0  | 1 1 1 1   | 1 1 1 1   | 0 0 1 1   | 1 1 1 0   | 0 0 0 0   |
|                  | F  0 0 0  | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
|                  | B  0 0 0  | 1 1 1 0   | 0 0 1 2   | 2 1 1 1   | 0 0 0 0   | 0 0 0 0   |
|                  | S  0 0 0  | 2 2 2 1   | 1 1 2 2   | 2 2 2 2   | 0 0 0 0   | 0 0 0 0   |

W: detected objects; T: true objects; F: false objects; B: undetected objects; S: number of buildings on the image; A: Adam optimizer; M: Momentum; R: RMSProp optimizer.

|                  | Set 1     | Set 2     | Set 3     | Set 4     | Set 5     | Set 6     |
|------------------|-----------|-----------|-----------|-----------|-----------|-----------|
|                  | A  M  R   | A  M  R   | A  M  R   | A  M  R   | A  M  R   | A  M  R   |
| garage           | W  1 0 0  | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
|                  | T  0 0 0  | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
|                  | F  1 0 0  | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
|                  | B  11 11  | 11 0 0    | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
|                  | S  11 11  | 11 0 0    | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
| terraced house   | W  0 7 8  | 0 0 0 0   | 0 2 3 0   | 1 0 0 0   | 0 14 13   | 0 0 0 0   |
|                  | T  0 6 6  | 0 0 0 0   | 0 1 2 0   | 0 0 0 0   | 0 10 13   | 0 0 0 0   |
|                  | F  1 1 2  | 0 0 0 0   | 0 1 1 0   | 1 0 0 0   | 0 4 0     | 0 0 0 0   |
|                  | B  10 4 4 | 0 0 0 3   | 2 1 0 0   | 0 0 0 0   | 36 26 23  | 0 0 0 0   |
|                  | S  10 10  | 10 0 0    | 3 3 3 0   | 0 0 0 0   | 0 36 36   | 0 0 0 0   |
| Single-family house | W  0 9 7  | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 4 68 57   |
|                  | T  0 5 4  | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 53 49   |
|                  | F  0 4 3  | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   | 0 4 15 8  |
|                  | B  8 3 4  | 0 0 1 1   | 1 1 0 0   | 0 0 0 0   | 64 11 15  | 0 0 0 0   |
|                  | S  8 8 8  | 0 0 1 1   | 1 1 0 0   | 0 0 0 0   | 0 64 64   | 0 0 0 0   |
| block of flats   | W  4 22 21| 4 18 21   | 2 7 7 2   | 29 32 1   | 11 13 0   | 0 0 0 0   |
|                  | T  0 21 20| 0 18 20   | 0 6 6 0   | 29 32 0   | 11 13 0   | 0 0 0 0   |
|                  | F  4 1 1  | 4 0 1 1   | 0 1 1 2   | 0 0 0 0   | 0 0 0 0   | 0 0 0 0   |
|                  | B  24 3 4 | 21 3 1    | 7 1 1 41  | 12 9 8 18| 7 5 0 0   | 0 0 0 0   |
|                  | S  24 24  | 24 21 21  | 7 7 7 41  | 41 41 41  | 18 18 18  | 0 0 0 0   |
Table A3. Cont.

|       | Set 1 | Set 2 | Set 3 | Set 4 | Set 5 | Set 6 |
|-------|-------|-------|-------|-------|-------|-------|
|       | A     | M     | R     | A     | M     | R     |
| W     | 0     | 0     | 0     | 0     | 0     | 0     |
| T     | 0     | 0     | 0     | 0     | 0     | 0     |
| F     | 0     | 0     | 0     | 0     | 0     | 0     |
| B     | 0     | 0     | 0     | 2     | 1     | 1     |
| S     | 0     | 0     | 0     | 0     | 0     | 0     |
|       |       |       |       |       |       |       |
| church|       |       |       |       |       |       |
| W     | 0     | 0     | 0     | 3     | 0     | 1     |
| T     | 0     | 0     | 0     | 1     | 0     | 1     |
| F     | 0     | 0     | 0     | 2     | 0     | 0     |
| B     | 0     | 0     | 0     | 2     | 1     | 2     |
| S     | 0     | 0     | 0     | 2     | 2     | 2     |

W: detected objects; T: true objects; F: false objects; B: undetected objects; S: number of buildings on the image; A: Adam optimizer; M: Momentum; R: RMSProp optimizer.

References

1. Li, L.; Wang, C.; Zhang, H.; Zhang, B.; Wu, F. Urban Building Change Detection in SAR Images Using Combined Differential Image and Residual U-Net Network. *Remote Sens.* 2019, 11, 1091. [CrossRef]
2. Cui, B.; Zhang, Y.; Yan, L.; Wei, J.; Wu, H. An Unsupervised SAR Change Detection Method Based on Stochastic Subspace Ensemble Learning. *Remote Sens.* 2019, 11, 1314. [CrossRef]
3. Neuville, R.; Pouliot, J.; Pou, F.; Billen, R. 3D Viewpoint Management and Navigation in Urban Planning: Application to the Exploratory Phase. *Remote Sens.* 2019, 11, 236. [CrossRef]
4. Luo, N.; Wan, T.; Hao, H.; Lu, Q. Fusing High-Spatial-Resolution Remotely Sensed Imagery and OpenStreetMap Data for Land Cover Classification Over Urban Areas. *Remote Sens.* 2019, 11, 88. [CrossRef]
5. Qin, Y.; Wu, Y.; Li, B.; Gao, S.; Liu, M.; Zhan, Y. Semantic Segmentation of Building Roof in Dense Urban Environment with Deep Convolutional Neural Network: A Case Study Using GF2 VHR Imagery in China. *Sensors* 2019, 19, 1164. [CrossRef] [PubMed]
6. Wang, J.; Qin, Q.; Ye, X.; Wang, J.; Yang, X.; Qin, X. A Survey of Building Extraction Methods from Optical High Resolution Remote Sensing Imagery. *Remote Sens. Technol. Appl.* 2016, 31, 653–662.
7. 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.
8. 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, Boston, MA, USA, 14–18 September 2015; pp. 234–241.
9. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 2012, 25, 1097–1105. [CrossRef]
10. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* 2014, arXiv:1409.1556.
11. 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, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
12. 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.
13. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 580–587.
14. Girshick, R. Fast R-CNN. In Proceedings of the International Conference on Computer Vision, Santiago, Chile, 13–16 December 2015; pp. 1440–1448.

15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 2016, 39, 1137–1149. [CrossRef]

16. Dai, J.; Li, Y.; He, K.; Sun, J. R-FCN: Object Detection via Region-based Fully Convolutional Networks. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.

17. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.

18. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.

19. 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]

20. Wu, G.; Shao, X.; Guo, Z.; Chen, Q.; Yuan, W.; Shi, X.; Xu, Y.; Shibasaki, R. Automatic Building Segmentation of Aerial Imagery Using Multi-Constraint Fully Convolutional Networks. *Remote Sens.* 2018, 10, 407. [CrossRef]

21. Xu, Y.; Wu, L.; Xie, Z.; Chen, Z. Building Extraction in Very High Resolution Remote Sensing Imagery Using Deep Learning and Guided Filters. *Remote Sens.* 2018, 10, 144. [CrossRef]

22. Vakalopoulou, M.; Karantzalos, K.; Komodakis, N.; Paragios, N. Building detection in very high resolution multispectral data with deep learning features. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Milan, Italy, 26–31 July 2015; pp. 1873–1876.

23. Boonpook, W.; Tan, Y.; Ye, Y.; Torteeka, P.; Torsri, K.; Dong, S. A Deep Learning Approach on Building Detection from Unmanned Aerial Vehicle-Based Images in Riverbank Monitoring. *Sensors* 2018, 18, 3921. [CrossRef]

24. Zuo, T.; Feng, J.; Chen, X. HF-FCN: Hierarchically Fused Fully Convolutional Network for Robust Building Extraction. In Proceedings of the Asian Conference on Computer Vision, Taipei, Taiwan, 20–24 November 2016; Springer: Cham, Switzerland, 2016; pp. 291–302.

25. Chen, K.; Fu, K.; Gao, X.; Yan, M.; Sun, X.; Zhang, H. Building extraction from remote sensing images with deep learning in a supervised manner. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, Fort Worth, TX, USA, 23–28 July 2017; pp. 1672–1675.

26. Alshehhi, R.; Marpu, P.R.; Woon, W.L.; Mura, M.D. Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. *ISPRS J. Photogramm. Remote Sens.* 2017, 130, 139–149. [CrossRef]

27. Zou, Z.; Shi, T.; Li, W.; Zhang, Z.; Shi, Z. Do Game Data Generalize Well for Remote Sensing Image Segmentation? *Remote Sens.* 2020, 12, 275. [CrossRef]

28. Perez, H.; Tah, J.; Mosavi, A. Deep learning for detecting building defects using convolutional neural networks. *Sensors* 2019, 19, 3556. [CrossRef]

29. Zhang, L.; Wu, J.; Fan, Y.; Gao, H.; Shao, Y. An Efficient Building Extraction Method from High Spatial Resolution Remote Sensing Images Based on Improved Mask R-CNN. *Sensors* 2020, 20, 1465. [CrossRef]

30. Ghandour, A.J.; Jeziyni, A.A. Autonomous Building Detection Using Edge Properties and Image Color Invariants. *Buildings* 2018, 8, 65. [CrossRef]

31. Bai, T.; Pang, Y.; Wang, J.; Han, K.; Luo, J.; Wang, H.; Lin, J.; Wu, J.; Zhang, H. An Optimized Faster R-CNN Method Based on DRNet and RoI Align for Building Detection in Remote Sensing Images. *Remote Sens.* 2020, 12, 762. [CrossRef]

32. Maltezos, E.; Doulamis, A.; Doulamis, N.; Ioannidis, C. Building extraction from LiDAR data applying deep convolutional neural networks. *IEEE Geosci. Remote Sens. Lett.* 2018, 16, 155–159. [CrossRef]

33. Ji, S.; Shen, Y.; Lu, M.; Zhang, Y. Building Instance Change Detection from Large-Scale Aerial Images using Convolutional Neural Networks and Simulated Samples. *Remote Sens.* 2019, 11, 1343. [CrossRef]

34. Li, W.; He, C.; Fang, J.; Zheng, J.; Fu, H.; Yu, L. Semantic Segmentation-Based Building Footprint Extraction Using Very High-Resolution Satellite Images and Multi-Source GIS Data. *Remote Sens.* 2019, 11, 403. [CrossRef]
35. Huang, Z.; Zhong, Z.; Sun, L.; Huo, Q. Mask R-CNN with pyramid attention network for scene text detection. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 764–772.

36. Hershberger, J.E.; Snoeyink, J. Speeding up the Douglas-Peucker Line-Simplification Algorithm; University of British Columbia, Department of Computer Science: Vancouver, BC, Canada, 1992.

37. Tzutalin. Labellmg. Git Code. 2015. Available online: https://github.com/tzutalin/labellmg (accessed on 20 February 2020).

38. Shetty, A.R.; Krishna Mohan, B. Building Extraction in High Spatial Resolution Images Using Deep Learning Techniques. In Computational Science and Its Applications—ICCSA 2018; Gervasi, O., Murgante, B., Misra, S., Stankova, E., Torre, C.M., Rocha, A.M., Taniar, D., Apduhan, B.O., Tarantino, E., Ryu, Y., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2018; Volume 10962. [CrossRef]

39. Lari, Z.; Ebadi, H. Automatic extraction of building features from high resolution satellite images using artificial neural networks. In Proceedings of the ISPRS Conference on Information Extraction from SAR and Optical Data, with Emphasis on Developing Countries, Istanbul, Turkey, 16–18 May 2007.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).