electronics

MDPI

Article

Building Façade Style Classification from UAV Imagery Using a Pareto-Optimized Deep Learning Network

Rytis Maskeliūnas [1,*], Andrius Katkevičius [2], Darius Plonis [2], Tomyslav Sledevič [2], Adas Meškėnas [3] and Robertas Damaševičius [4]

[1] Department of Multimedia Engineering, Kaunas University of Technology, 51423 Kaunas, Lithuania
[2] Department of Electronic Systems, Vilnius Gediminas Technical University, 03227 Vilnius, Lithuania
[3] Department of Reinforced Concrete Structures and Geotechnics, Vilnius Gediminas Technical University, 10223 Vilnius, Lithuania
[4] Department of Applied Mathematics, Silesian University of Technology, 44-100 Gliwice, Poland
* Correspondence: rytis.maskeliunas@ktu.lt

Abstract: The article focuses on utilizing unmanned aerial vehicles (UAV) to capture and classify building façades of various forms of cultural sites and structures. We propose a Pareto-optimized deep learning algorithm for building detection and classification in a congested urban environment. Outdoor image processing becomes difficult in typical European metropolitan situations due to dynamically changing weather conditions as well as various objects obscuring perspectives (wires, overhangs, posts, other building parts, etc.), therefore, we also investigated the influence of such ambient "noise". The approach was tested on 8768 UAV photographs shot at different angles and aimed at very different 611 buildings in the city of Vilnius (Wilno). The total accuracy was 98.41% in clear view settings, 88.11% in rain, and 82.95% when the picture was partially blocked by other objects and in the shadows. The algorithm's robustness was also tested on the Harward UAV dataset containing images of buildings taken from above (roofs) while our approach was trained using images taken at an angle (façade still visible). Our approach was still able to achieve acceptable 88.6% accuracy in building detection, yet the network showed lower accuracy when assigning the correct façade class as images lacked necessary façade information.

Keywords: Pareto; optimization; deep learning; building segmentation; building façade recognition; drones

Citation: Maskeliūnas, R.; Katkevičius, A.; Plonis, D.; Sledevič, T.; Meškėnas, A.; Damaševičius, R. Building Façade Style Classification from UAV Imagery Using a Pareto-Optimized Deep Learning Network. *Electronics* **2022**, *11*, 3450. https://doi.org/10.3390/electronics11213450

Academic Editors: Singara Singh Kasana, Ben Soh and Geeta Kasana

Received: 2 October 2022
Accepted: 16 October 2022
Published: 25 October 2022

1. Introduction

Remote data collection has become a need in industrial design [1], architecture [2], and city planning [3]. Object tracking has been used both in industrial applications [4] and for pedestrian tracking in smart city applications [5]. Another common application is the automatic inspection of building surfaces [6]. The usage of unmanned aerial vehicles (UAVs) may be emphasized to support construction management tasks, particularly with relation to construction site logistics, monitoring and follow-up of the work's evolutionary process, and permitting visual inspections in difficult spots and because of its capacity to record inaccessible regions [7] such as bridges [8], wind turbine rotor blades [9], and featureless tunnel-like settings [10], the UAV has become an ideal instrument for visual data collecting [11]. These applications are enabled by deep learning technologies, which surpass standard classification algorithms in terms of performance and feature transferability to new datasets [12]. It has been also demonstrated to contribute substantially to solving various sustainable-development problems [13].

In the architectural, engineering, construction, and facility management industries, unmanned aerial vehicles have become an almost mandatory visual data collecting tool for analyzing building façades, leading to collections of sequence datasets, potentially used to benchmark the computer vision systems [14,15]. The building façade is also a

major distinguishing characteristic from the initial eye contact since it often becomes a complicated interaction between the inside of buildings and the outside [16]. The UAV can collect a comprehensive high-quality image of the façade from every angle. As a result of the image processing techniques used in digital photogrammetry, extra visual assets such as 3D or orthomosaic models can be created [17]. The engineer, architect, or even AI itself [18] may then examine the visual assets created and flag potential faults, make maps, design buildings and building information modeling (BIM) applications, etc. [19]. This method has also become a very sustainable alternative for improving accessibility and providing an enriched analysis [20] and experience of city life due to their ability to capture images remotely and provide interesting and attractive perspectives to the urban landscape and buildings [21]. They have been effectively used for structural health monitoring of buildings [22]. However, the effective use of remotely sensed images and data captured by drone cameras requires the effective use of image processing, segmentation, and classification techniques [23].

We can potentially use the images of the urban landscape captured by drones to identify visual landmarks and defects on historical buildings, objects of urban heritage, etc. [24], or in city planning [25]. The problem is very complex when images in question vary in style, are obstructed by numerous other objects, and there is a low possibility of a clear, unobstructed view, which is the case when given the task to analyz e the unique buildings of the city of Vilnius (Wilno), a UNESCO heritage place, varying in style, from the Gothic church towers to the most modern glass structures[26]. Such a task requires the adoption of intelligent solutions supported by AI for the detection and identification of buildings in the overall urban background. The objective of our research, therefore, was set to detect the borders of a building's façade accurately under changing weather and lighting situations, as well as to determine the real building façade style based on façade taxonomy [27].

The remaining sections of the paper are organized as follows. Section 2 goes through the relevant works and highlights the most typical issues encountered while processing building façade images. Section 3 introduces the proprietary dataset and outlines the approaches employed. Section 4 summarizes the findings. Section 5 concludes the paper.

2. Related Works

Image processing is a broad field of research with a multitude of diverse methods available for researchers. Numerous studies have addressed the topic of object identification and detection [28–30]. Image processing becomes particularly complex in open spaces, where many extraneous noises that affect image processing occur: the sun and the resulting shadows, wind, rain, snow, and water surfaces and reflections from them [31–33]. All of the above factors influence the level of illumination and make the characteristic features of objects less visible. The problem of image processing in open spaces with dynamic changes in weather conditions is especially problematic [34]. Currently, there are no effective image processing methods suitable for identifying objects with different characteristics in different environments surrounding the object [35–37]. When considering the specific problem of building detection and identification, the current research focuses on:

- Detection of buildings and determination of their shape from high-resolution satellite images [38–41]. As an alternative, drones or manned drones can be used. In this case, high-resolution photos are obtained for a selected time and further analyzed separately. Such a system can be classified as both static and dynamic because the distance between the camera and the surface is distant and the camera does not move much relative to the object during the selected period. The most common task is to update the maps of buildings to assess the relationship between the territory of buildings and green areas and assess the roof of specific buildings, otherwise referred to in the literature as the form of building foundations.
- Detection and identification of buildings from building façade elements [42,43] in a purely static context. In this case, the camera does not move relative to the image, so

the captured characteristic features of the building allow a building to be distinguished from other buildings.

- Identification of buildings from a video stream in real time [44]. The image is processed by a camera, which is installed, for example, in front of a moving bus. This option can be used for building contour detection, but it is rarely used for building identification. To improve the accuracy, Zhang et al. [45] recommends using semantic segmentation to extract building façades. Dai [46] offers a model based on deep learning-based semantic segmentation technology and an ensemble learning technique, with object detection technology incorporated as a magnifier to increase model performance on small objects and border predictions.

We discuss the related works related to computer vision of building analysis below.

Semantic segmentation of building foundations is discussed in [47]. To solve the problem, researchers used deep neural networks (DNNs) based on the convolutional neural network (CNN) U-Net. A database of satellite images of cities was used to train the network. The Triplet Loss Function was used for training. The methodology allowed improving the segmentation of building foundations by 2% compared to the latest methods proposed by other authors. The authors of [48] used Siamese's fully CNN U-Net (SiU-Net) to segment buildings from shared satellite images of the area. The 512 × 512 images of individual buildings, parts of large buildings, and groups of buildings were used for training. Images obtained by an infrared camera are processed in [49] to segment the shapes of buildings viewed from above using monochrome images obtained with an infrared camera. The 2D adaptive image filter was used to determine the straight edges of contrasting areas in the image and reject the lines intersecting at non-straight angles as noise. The work [50] also examines the segmentation of buildings using satellite imagery. In this case, the authors focus on building projections using CNNs with rotational equivariant properties. However, CNNs are not sufficient as the quality of the segmentation is often determined by the position of the camera relative to the building one might want to segment. The authors of [51] state that CNNs are not fully suitable for processing synthetic aperture radar (SAR) images. The authors propose a new multitasking structure for a full CNN for segmenting buildings from SAR images, which allows for improving the quality of segmentation. Similar work has been done in [52]. Before processing satellite images with a CNN, additional image processing filters are first applied. The authors of [53] use already-created city maps from online mapping platforms in their research. They divided their work to extract 2D and 3D map fragments and distinguish the contours of buildings using various filters. A multilayer perceptron was used to determine the height of the building from 3D images and the building itself was segmented in the image using the MASK R-CNN network. The same task was performed in [54], only the graph convolutional network (GCN) was used in this case. The authors argue that reducing the contrast of satellite images degrades the quality of segmentation, so additional image processing is required before training a CNN. The situation is well-summarized in [55]. It states that there is no one-size-fits-all solution for image processing and segmentation of buildings. The accuracy of segmentation depends both on the data, in this case, the images, and on the chosen methodology, network type, structure, and additional image processing before paying for the network. Additional image processing by custom filters is also required. Tao et al. suggested using façade elements' spatial arrangement regularity and appearance similarity in a detection framework [56].

Vision transformer (ViT) techniques are fast gaining traction in computer vision and remote sensing, offering a viable alternative to CNNs and their variants. ViT's primal concept was a model of encoder–decoder sequence transduction based on self-attention and capable of encoding long-distance interactions between sequence components, originally applied to natural language processing [57], but soon after also converted for use as image categorization by Dosovitskiy et al. [58]. ViTs allow not using convolution layers in the same way that typical CNNs do, but instead, ViT utilizes multi-head attention processes as the primary building block to determine long-range contextual relationships between pixels

in the image data. ViT has several drawbacks over CNNs as it necessitates additional GPU memory and computational overhead [59]. This issue is exacerbated when dealing with large-sized inputs such as fine-resolution remote-sensing photos. Second, spatial details are not properly retained throughout the ViT's feature extraction, making it unsuitable for all scenarios of fine-grained building segmentation, but can be remedied by dedicated context paths, such as in [60]. Chen et al. [61] recommended employing cross-attention solely because it needs linear time for both computational and memory complexity rather than quadratic time, assisting in the development of a dual-branch transformer to merge picture patches of varying sizes to generate stronger image characteristics. Guo et al. [62] presented an automated method for extracting building information from satellite and street view images, which is based on a novel transformer-based deep neural network with a multidomain learning approach that was used to develop a compact model for multiple image-based deep learning information extraction tasks using multiple data sources. Chen et al. [63] proposed an efficient dual-pathway transformer structure that learns token long-term dependence in both spatial and channel dimensions and achieves state-of-the-art accuracy on benchmark building extraction datasets. Bazi et al. [64] demonstrated experimentally that their approach can condense the network by reducing half of the layers while maintaining competing classification accuracy. Bashamal et al. [65] proposed reshaping the picture and its enhanced version into a sequence of flattened patches, which would then be sent to the transformer encoder. The latter derives a compact feature representation from each picture using a self-attention technique that can manage global interdependence between image areas. The authors of [66] present a CNN+ViT hybrid search space with searchable down-sampling locations, converting the search space into blocks with a self-supervised training strategy to train each block individually before searching for the population center as a whole.

The problem of building segmentation, i.e., finding it in an image, is relevant when considering not only satellite images when the building is visible from above, but also images of the building façade when the building is photographed from the side view. For example, in [67], feature extraction is used for the automatic identification of buildings for tourism purposes. SIFT (scale-invariant feature transform) is used to determine characteristic points of the object in the image; assign an orientation to the object, compile a description of characteristic points, and perform an alignment of characteristic points. The result was greatly influenced by the ambient lighting and the accuracy was not sufficient. In [68], the authors use texture and color properties of a building to determine the position of an object and propose multi-scale neighborhood sensitive histograms of oriented gradient (MNSHOG) and automatic color correction to preserve the texture and color properties of building images. Next, they combine the texture and color characteristics of buildings into a single array and use the extreme learning machine (ELM) for classification. The paper [69] discusses localizing discriminative visual landmarks to identify a building and thus locate it. The study used CNN with algorithms for extracting their characteristic properties. Segmentation of building façades using procedural shape priors is presented in [70]. The authors used process modeling to determine the geometric and photometric variation of buildings, and random forests (RF) to optimize the processing time and resources used.

Morphological segmentation of building façades is presented in [71]. The authors use directional color gradients to isolate the façades of buildings from the general background of the city. The sky is then detected based on the segmentation method and the acquisition of color markers. Finally, the façade area is subdivided according to floor levels and morphological filters are used to alleviate the effect of reflections from windows, textured balconies, and other small surface irregularities on the quality of segmentation. Similar methods are also used in dynamic building façade detection and identification systems, where the video stream or photos are taken from vehicles moving on the road. The study [72] used building façade detection, segmentation, and feature extraction procedures to locate a mobile robot based on the accurate segmentation of building façades. The authors classify the points of the figure into planes using RANSAC (random sample

consensus) algorithm. Next, the authors construct a model of Markov random fields until all of the individual points belonging to the individual surfaces of the building are classified into separate planes. The method allowed the detection of buildings with 90% accuracy and the detection of individual façade surfaces with 85% accuracy.

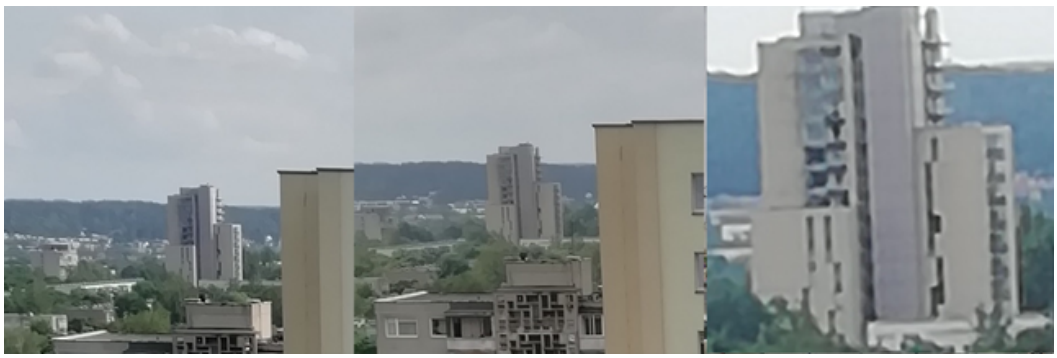
According to [73], segmenting an image into separate areas is a complex procedure. It is usually necessary to distinguish objects of complex shapes from the general background surrounding the building. Detecting the edges of individual areas is one of the main tools for finding and identifying objects in an image [74]. Often, authors first find the edges of objects and then create their various filters to get rid of the noise and leave only the edges of the objects that are relevant to them.

To summarize, the problems of the task of building detection and identification are obvious: changing environmental conditions greatly affect the quality of recognition. The complexity of the task is highly dependent on the shape of the building façade, the natural conditions, and the urban background behind the building. Using elementary mathematical models, researchers are usually able to identify only buildings with a very simple façade structure. Moreover, the classified buildings must differ significantly from each other. However, there are many problems in identifying buildings with a similar façade structure. In particular, due to distance and environmental factors, it is difficult to distinguish the contours of individual façade elements. Incorrect contours can be obtained with shadows or reflections. Separating the correct form alone does not guarantee successful further classification. In addition to the shape of the façade, the background behind the building has a significant impact on the tasks of detecting and identifying the façade of a building. In solving complex façade detection problems, researchers typically choose artificial neural networks. In this case, modern methods such as CNN and ViT can single out features that the researcher will simply not notice with the naked eye.

In this paper, we present a novel method for the segmentation and identification of specifically selected buildings against a noisy and cluttered city background. We discuss the common problems faced while solving this task in the following section.

Common Problems of Building Façade Image Processing

Images obtained by a remote camera often need to be denoised [75] and despeckled [76]. Additionally, some of the objects might appear in front of the building and the in-painting algorithm should be applied [77]. Then there might be a factor of image resolution [78]. The reasons why the image quality needs to be improved can be seen from the sample images of the building façade presented in Figure 1a–c.



(**a**) (**b**) (**c**)

Figure 1. Examples of building images: (**a**) image from a natural distance; (**b**) a view with a slight "floating effect" due to direct sunlight; (**c**) a captured image with a specific departure of points.

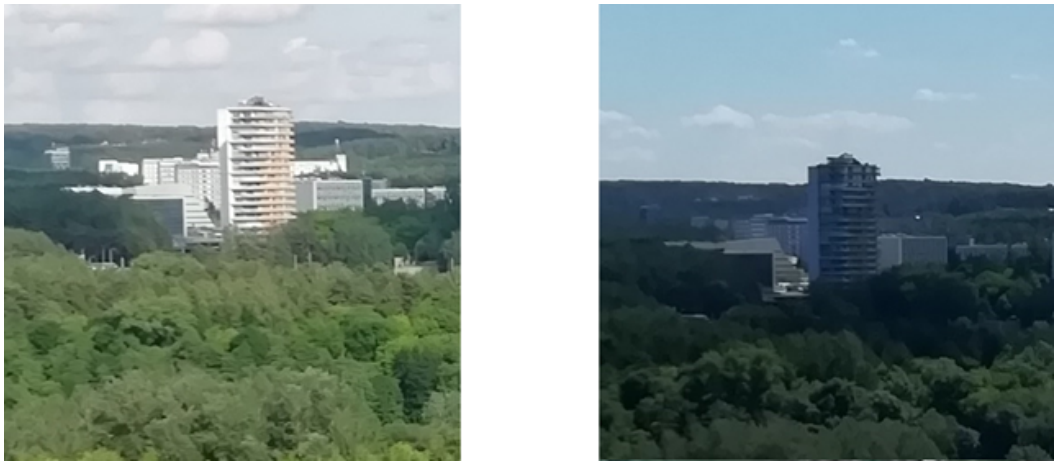
As the position of the sun changes with respect to the camera's lens, the image quality also changes. In addition, when shooting from a greater distance, the resolution of the zoomed façade decreases. Then there is an impact of wide camera angle, and also very common fisheye effect [79]. In this case, you need to use filters to get better results [80].

In the specific case, Figure 1 shows an example of a building façade in the evening when the lighting level is already lower. The resulting image is very noisy and quite blurry. By using the unsharp masking method, the thresholds of individual areas can be highlighted. Figure 2a presents the original image while Figure 2b shows the processed image with increased contrast between the colors of the individual areas. The results achieved in the figure are visible to the naked eye. In this case, the image in Figure 2b will allow obtaining better contours of the building and thus better results should be obtained by training an ANN.

(**a**) (**b**)

Figure 2. (**a**) Original and (**b**) color image of increased contrast between individual areas.

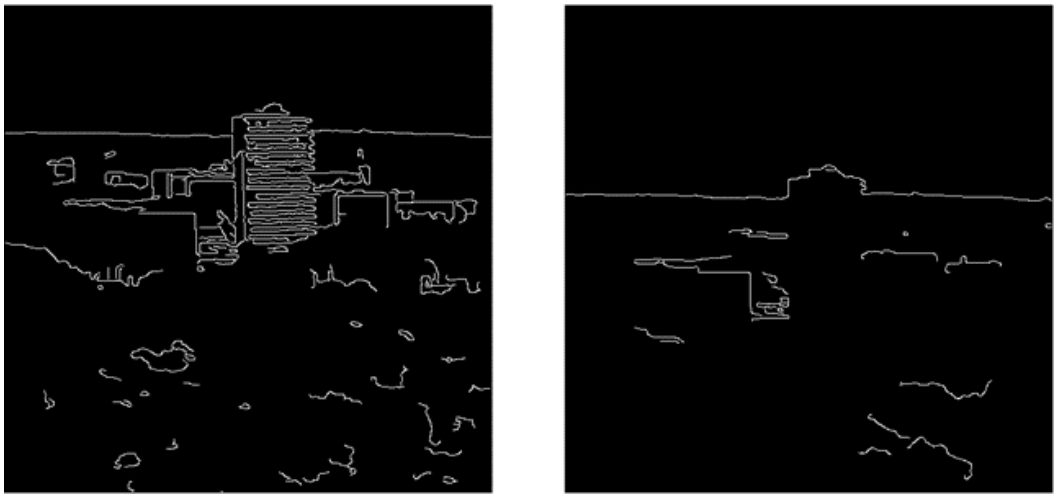
Another obvious problem is varying weather conditions [81]. In Figure 3a the building is illuminated by the sun, and in Figure 3b the same building is covered by rain.



(**a**) (**b**)

Figure 3. Example of building view (**a**) in the sunshine; (**b**) covered by clouds and rain.

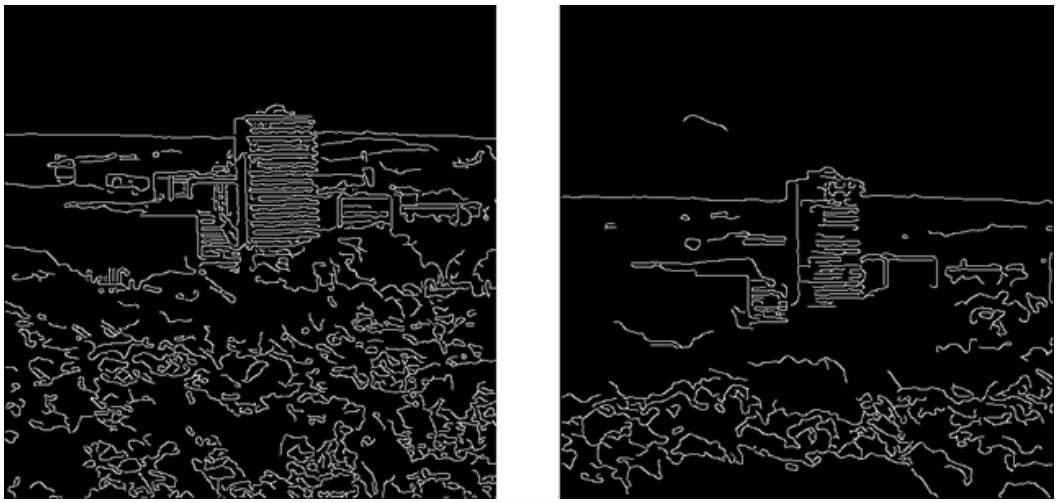
The overall level of illumination varies greatly between the façade and the surrounding environment. For these reasons, using the same image processing methodology will result in completely different results in both cases [82]. Figures 4 and 5 show how applying the same threshold values results in completely different contour separation results when the sun is shining and it is covered before the rain. For example, Using a Canny filter to separate the contours and using a threshold value [83], e.g., that of 0.45, would make it possible to find the area of the building and see it quite clearly, as illustrated in Figure 4a. Meanwhile, in a case of covering, the same threshold value gives a completely bad result (Figure 4b).



(**a**) (**b**)

Figure 4. Separation of area contours with a Canny filter with a threshold value of 0.45, when (**a**) the sun is shining, (**b**) overcast.

If we focus on the image overlaid and choose a threshold value of 0.16, we can distinguish the building (as can be seen in Figure 5b. However, at this threshold, the noise level (contours and trees of other buildings) increases significantly and it is difficult to detect the building in Figure 5a. These images show that there is no single threshold value for all cases. In similar types of image processing tasks, the threshold is adjusted by an expert.



(**a**) (**b**)

Figure 5. Separation of area contours with a Canny filter with a threshold value of 0.16 when (**a**) the sun is shining, (**b**) overcast

A common problem is the need for the transformation of the building façade when in the frame we see the building not directly in front, but rotated or pushed [84]. Viewed at an angle, the façade of the building no longer has 90° angles and is stretched. In this case, before processing, we need to transform the building so that we can see the façade exactly from the front. In Figure 6a, we see the originally obtained frame, and in Figure 6b we see the transformed image. In Figure 6b, the façade of the building is shown exactly from the front. Transformations of this kind make it possible to improve the visibility of the individual areas of the building façade and thus guarantee a better separation of the contours [85] and can further increase the quality of image processing systems. Such transformation operations are also useful in training artificial neural networks. Using push,

rotate, and other transformations, it is possible to obtain the façade of a building at different angles from different projections and thus increase the size of the training data.

(**a**) (**b**)

Figure 6. Example of building façade detection: (**a**) the original image obtained, and (**b**) the transformed image in order to have only a view of the façade exactly from the front.

3. Materials and Methods

3.1. Data Acquisition

We have used a DJI Mavic Air [86] (Sensor: 1/2.3″ CMOS, Effective Pixels: 12 MP, Lens FOV: 85°, 35 mm Format Equivalent: 24 mm, Aperture: f/2.8, Electronic Shutter: 8-1/8000 s), flown above picturesque locations in the city of Vilnius (Wilno), to capture numerous locations of different architecture as recommended by the Vilnius Architectural Guide [87] (see Figure 7 illustrating the dynamic variety in styles). We have captured 611 buildings at multiple angles (minimum of 4 (north, east, south, west), with additional variations if the façade was not visible at the exact angle) and in different conditions (in total 8768 images from 8 a.m. to 5 p.m.) The shape of the buildings themselves varied from quite ordinary squares to multiangle shapes, also with differences in stylistic details, such as the number and layout of windows and doors, the colors of façades as well as decorative elements, roof slopes, and style. The buildings are spaced at different distances and located in different directions around the city. The photos were taken during the periods of 2021 and 2022, so different shadows fell on the building façades at different times of the day, thus worsening the separation of building features and the identification of buildings. The UAV was flying in different permissible weather conditions. Some of the images were obtained of the sun shining in the morning, some during noon, and some in the evening with façades falling into shadows (approximately 22%, 69%, and 9% of each category). Some images were also obtained at medium and high cloud cover present at the time of flying (around 35%). The dataset also included an additional 1435 images taken during light rain (in non-windy conditions permissible to fly our UAV). The number of images in each class was not equal, nor was the number of buildings with distinct façade types. To balance the dataset, we used standard picture augmentation methods (rotation, brightness, etc.) from the Albumentations library [88]. The dataset was pre-annotated by the same R-CNN network developed by us (trained on the [89] dataset), then every image was manually supervised and corrections were made if required by the authors of this paper. We have left 30% of the dataset untouched (for validation and testing). In total, 70% of the photos were used for training. Validation was performed with 20% of the photos and the remaining 10% of photos were used for testing.

Figure 7. Famous Vilnius (Wilno) buildings used in the research.

3.2. Methodology

In this chapter, we present the operation algorithm of our proposed methodology (Figure 8). The methodology has three stages: collection and processing of training data (stage I), training of the model (stage II), and building façade style classification (stage III).

1. Stage I requires the collection and processing of training data for network training. In particular, the façades of buildings are photographed using UAVs in various weather conditions (step 1). Furthermore, the specific façades of the buildings are cut from the obtained photos (step 2) with a bounding box. When cutting building façades, keep in mind that all photos for network training should be the same dimensions. This is a disadvantage because the shape of the buildings does not always allow the entire façade of the building to be cut out and placed in a picture of a certain size. In order to accommodate the entire façade of the building, a larger area has to be cut out, in which case part of the city background remains in the image of the educational data.
2. Once the data has been collected for training, the process proceeds to stage II. First of all, the network model is developed (R-CNN) (step 3). The network is further trained with façade data (step 4). The network then performs a validation function (step 5). Finally, a network robustness check is performed with images not yet used in the training (step 6).
3. Finally, we classify the building façades (stage III). The network assigns each building in the image to one of the different façade style groups. Next, we evaluate how many images the R-CNN has assigned to the correct groups. We preset a certain threshold as a percentage of the classification reliability we aim for. If the obtained reliability is greater than or equal to the threshold, we consider that the CNN model was successful (step 7).

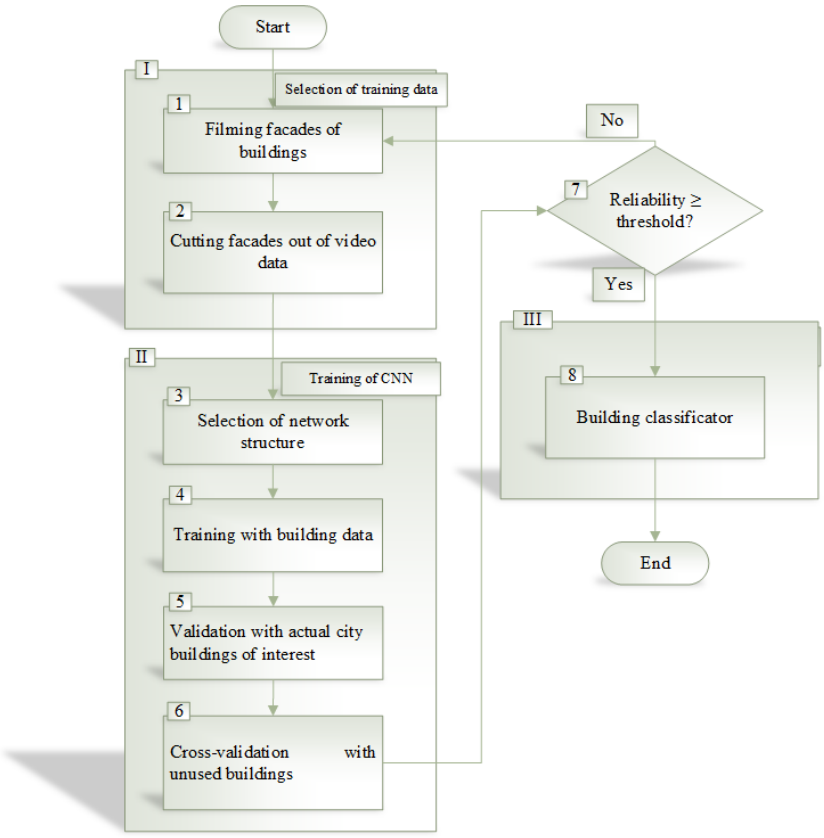


Figure 8. Deep learning-based methodology to identify building façades.

The remainder of this section aims to explain the heuristic building detection method used in data processing, explain the data augmentation process, as well as to provide more details on the network architecture and the optimization applied.

3.3. Heuristic Building Detection Method

First, the original color image with a resolution of 2080 × 4160 (step 1) is loaded. Next, the resolution of the image (step 2) is changed to 512 × 512. The same resolution will be used by artificial neural networks. Next, the color image I_c is changed to gray I_g with brightness information (step 3). To compute gradient of the image, the derivatives I_x and I_y are calculated by convolving I_g with Sobel kernels K_x and K_y:

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \tag{1}$$

$$K_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \tag{2}$$

$$I_x = K_x * I_g, \tag{3}$$

$$I_y = K_y * I_g. \tag{4}$$

The magnitude G of the gradient is calculated as follows:

$$|G| = \sqrt{I_x^2 + I_y^2}. \tag{5}$$

The threshold for gradient T_g (step 4) is selected. Selecting a T_g value distinguishes contours between contrasting areas. The contours are assigned white if the gradient pixel

$|G(x,y)| > T_g$ and the background is assigned black (step 5). The sizes of the objects are estimated as long as there are unprocessed objects (step 6). The sizes of the objects are estimated by a sum of pixels vertically P_y and horizontally P_x. If the object size $P_x + P_y$ is larger than the selected size threshold T_s (step 7), the i- th object is assigned a tag $O(i)$ (step 8) and the next $i + 1$ object is processed:

$$O(i) = \begin{cases} 1, & \text{if } P_x + P_y \geq T_s, \\ 0, & \text{if } P_x + P_y < T_s. \end{cases} \tag{6}$$

If the object size is less than the threshold T_s, the object is removed (step 9). After processing all of the objects and leaving only the largest ones according to the set threshold T_s, the rectangular objects are searched further (step 10). If there are rectangular shape (step 11), the largest rectangular shapes (step 12) is extracted.

3.4. Image Augmentation

To improve the training performance of the deep learning model, we have adopted image augmentations, which are useful when a dataset of images is too small as in our case.

We have adopted the Albumentations image augmentation library [88]. We adopted elastic deformation augmentation using the RGB adaptation of elastic transform as suggested in [90].

The method produces an augmented image from the original image by applying a displacement field to its pixels. The method defines for each pixel in the original image, the displacement field $\Delta x(x,y) = \alpha rand(-1,+1)$ and $\Delta y(x,y) = \alpha rand(-1,+1)$, where α is a scaling factor that depends on the size of the original image, and $rand(-1,+1)$ is a random value drawn from the uniform distribution in $[-1,1]$.

Because of the random displacement of each pixel, the augmentation introduces distortions in the augmented image. The horizontal Δx and the vertical Δy displacement fields are then filtered by the rotationally symmetric Gaussian lowpass filter.

3.5. An Enhanced R-CNN Model

We have chosen to utilize a modification of the region-based convolutional neural network [91] due to its reasonably low computational weight and proven good efficiency with patch analysis, further enhanced by our suggested Pareto optimization (see a dedicated section below). The enhanced R-CNN model for building façade detection is developed as follows. The size of the input layer depends on the size and type of the image. The network also includes a branch of the full convolutional network (FCN) to help improve the detection of the building shape as was suggested in a similar application by Wang et al. [92]. The size of the input layer is defined in our approach as the vector of $h \times l \times c$, where h is the height; l—width of the image; c— the number of channels in the picture (for grayscale, $c = 1$). In this case, data normalization is not used in the input layer. This is followed by a 2D convolutional layer. This layer applies convolution sliding filters to the input, that is, the visual information entering the input layer. The convolution operation between the filters and the image entering the input is performed by sliding the filter vertically and horizontally and calculating the product of the weights and the input point and adding the initial component (bias). The response y of the convolutional filter with the number of input channels n is calculated as follows:

$$y = \sum_{c=1}^{n} w_c \times x_c + b, \tag{7}$$

where w_c is a weight array of the 2D filter of the cth input channel; x_c is a 2D input to the cth filter; b is the bias.

In this case, we selected a hidden convolutional layer size equal to 16 filters of size 3×3. We also chose to match the output size of the layer to the input size without additional

overlap ("Padding", "same"). The filter shift step was left at the default level of 1×1. Initial weights were assigned equal to 1. The bias training factor was left equal to 1.

What follows is a data packet normalization layer that normalizes the data transmitted over each input channel as a small data packet. To accelerate CNN training and reduce the sensitivity to network initialization, we used batch normalization (BN) layers. The BN layer implements the following expressions:

$$\mu = \frac{1}{m}\sum_{i=0}^{m} y_i, \tag{8}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=0}^{m} (y_i - \mu)^2, \tag{9}$$

$$\hat{y} = \frac{y - \mu}{\sqrt{\sigma^2 + \epsilon}}, \tag{10}$$

$$\tilde{y} = \gamma\hat{y} + \beta, \tag{11}$$

where μ is a mean; σ^2 is the variance; ϵ is the numerical stability coefficient; γ is the scale factor; β is the offset calculated over a mini-batch; m is the number of images in a mini-batch; y is the response from multi-channel convolution filter; $\tilde{y}$ is the batch normalized output.

The rectified linear unit (ReLU) layer performs a nonlinear threshold operation for each input element when any value less than zero is set to zero:

$$y_r = \begin{cases} \tilde{y}, & \text{if } \tilde{y} \geq 0, \\ 0, & \text{if } \tilde{y} < 0. \end{cases} \tag{12}$$

In the max pooling layer, a sample reduction operation is performed by dividing the input sample into rectangular concentration regions and calculating the maximum value for each region. Equation (13) describes the output after max pooling:

$$y_m = \max_{1 \leq j \leq M \times M} y_r(j), \tag{13}$$

where M is the size of the pooling region.

Next, the four layers already discussed are repeated, starting with the 2D convolution layer. Only in this case, 32 filters of size 2×2 are selected in the layer. The BN, ReLU, and max pooling layers are identical to those discussed earlier.

This is followed by a fully connected (FC) layer in which the input data is multiplied by weight vectors and a bias value is added. Equation (14) describes the ith neuron output $f_j^{(L)}$ in the Lth fully connected layer:

$$f_j^{(L)} = a\left(w_{0,j}^{(L)} + \sum_{i=1}^{N_{L-1}} w_{i,j}^{(L)} f_i^{(L-1)}\right), \tag{14}$$

where $a(\cdot)$ denotes an activation function; N_{L-1} is the number of neurons in the $L-1$ fully connected layer; L is the index of the fully connected layer; i is the synapse index; j is the neuron index in the L-th layer; $f_i^{(L-1)}$ is the output signal from ith neuron in the the $L-1$ layer; $w_{i,j}^{(L)}$ is the neuron weight in the L-th layer.

R-CNN optimizes the spatial location misalignment problem produced by the ROI pooling layer by using region of interest (ROI) aligning and introducing a bilinear interpolation approach. To achieve precise pixel-level target segmentation, each ROI is better matched to the position of pixels on the original input picture. Finally, a regression layer is added, in which the mean square regression error values are calculated. Our enhanced R-CNN is displayed in Figure 9.

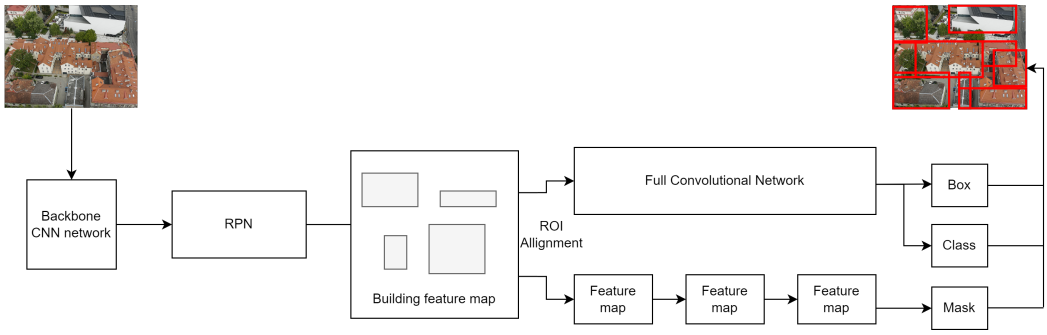


Figure 9. Architecture of the enhanced R-CNN network.

3.6. Pareto Optimized Layer

The Pareto-optimal layer [93,94] with quadratic polynomial optimization [95] was also added to help increase the overall accuracy of the algorithm, using an iterative population-based strategy that uses a single search procedure to estimate the Pareto front of optimum neural network solutions. This added feature was represented as a directed acyclic graph of cells, each of which is made up of 16 blocks. Each block is an 8-tuple that maps input vectors to output vectors. The output of the block was formed by applying a soft thresholding function and formed by concatenating unused blocks in the depth dimension.

The finite-dimensional convex optimization problem is described below. We define the quadratic module of a set of constraints G on a finite-dimensional vector space E as

$$Qc_{E,G} = \{\sum_{g \in G^0} g\,h + \sum_{g' \in G^+} g'\,h' \\ \mid h \in E, g\,h \in E \cdot E, h' \in \Sigma^2(E), g'h' \in E \cdot E\}.$$

If $G^\star$ is the set of constraints such that $G^{\star 0} = G^0$ and $G^{\star +} = \prod(G^+)$, the quadratic module $Qc_{E,G^\star}$ is the preordering of G, and is denoted $Qc^\star_{E,G}$.

By design, the $Qc_{E,G} \subset E \cdot E$ is a cone of polynomials which are non-negative on the semi-algebraic set S.

Definition 1. *Given a finite-dimensional vector space $E \subset R[xx]$ which contains* 1 *and a set of constraints G, we define*

$$Lc_{E,G} := \{\Lambda \in E \cdot E^* \mid \Lambda(p) \geq 0,\ \forall p \in Qc_{E,G}, \Lambda(1) = 1\}.$$

The convex set associated with the preordering $Qc^\star_{E,G} = Qc_{E,G^\star}$ is denoted $Lc^\star_{E,G}$.

The set $Lc_{E,G}$ is the intersection of the closed convex cone of semi-definite positive quadratic forms on $E \times E$ with a linear space S, thus it is a convex closed set. In multi-objective optimization, one considers $m > 1$ objectives $U_1, ..., U_m : S'$ over solution space S. A solution $s \subset S$ is called Pareto optimal *iff* it is not dominated by any other $s' \subset S$. The Pareto front S is the set of all Pareto optimal $s \subset S$. All other s are sub-optimal.

The conventional error backpropagation method is employed to train the neural network. The gradient descent approach is used to execute neural network training by updating the weight vector w in order to minimize the square error $E(w)$. It begins from an initial weight vector $w(0)$ and computes the weight vector difference $w(j)$ for each iteration. By using this technique, the weight vector w is moved in the direction where the function $E(w)$ decreases at the fastest pace.

Training is supervised and so we have a set of associations: $\mathbf{s}^{(q)} : \mathbf{t}^{(q)}, q = 1, 2, \ldots, Q$ is given. The training vectors $\mathbf{s}^{(q)}$ have N components,

$$\mathbf{s}^{(q)} = \begin{bmatrix} s_1^{(q)} & s_2^{(q)} & \ldots & s_N^{(q)} \end{bmatrix},$$

and their targets $\mathbf{t}^{(q)}$ have M components,

$$\mathbf{t}^{(q)} = \begin{bmatrix} t_1^{(q)} & t_2^{(q)} & \dots & t_M^{(q)} \end{bmatrix}.$$

Similar to the Delta rule, during training, the network is shown each training vector one at a time. Let us say that during the training procedure time step t, given a training vector $\mathbf{s}(q)$ for a certain q as input, to the network, $\mathbf{x}(t)$. To propagate the input signal forward through the network, then use the current weights and biases to generate $\mathbf{y}(t)$, the matching network output. The steepest descent algorithm is used to minimize the weights and biases. The error square for this training vector is:

$$E = \|\mathbf{y}(t) - \mathbf{t}(t)\|^2,$$

where $\mathbf{t}(t) = \mathbf{t}^{(q)}$ is the target vector that corresponds to the selected training vector $\mathbf{s}(q)$.

The network's total weights and biases determine this square error, E, as they are necessary for $\mathbf{y}(t)$. Based on the steepest descent method, we identify a set of updating rules for them:

$$w_{ij}^{(\ell)}(t+1) = w_{ij}^{(\ell)}(t) - \alpha \frac{\partial E}{\partial w_{ij}^{(\ell)}(t)}$$

$$b_j^{(\ell)}(t+1) = b_j^{(\ell)}(t) - \alpha \frac{\partial E}{\partial b_j^{(\ell)}(t)},$$

where $\alpha(>0)$ is the learning rate.

We must comprehend how E is affected by the weights and biases in order to compute these partial derivatives. In the beginning, E explicitly depends on the network output $\mathbf{y}(t)$. (the final layer's activations, $\mathbf{a}(L)$), It is thereafter reliant on the net input into the $L-$th layer, $\mathbf{n}(L)$. Additionally, $\mathbf{n}(L)$ is determined by the weights and biases of layer L as well as the activations of the layer before it. For brevity, the reliance on step t is removed, which is the explicit relation as follows:

$$\begin{aligned} E &= \|\mathbf{y} - \mathbf{t}(t)\|^2 = \|\mathbf{a}^{(L)} - \mathbf{t}(t)\|^2 = \|f^{(L)}(\mathbf{n}^{(L)}) - \mathbf{t}(t)\|^2 \\ &= \left\| f^{(L)} \left(\sum_{i=1}^{N_{L-1}} a_i^{(L-1)} w_{ij}^{(L)} + b_j^{(L)} \right) - \mathbf{t}(t) \right\|^2. \end{aligned}$$

It is then easy to compute the partial derivatives of E with respect to the elements of $\mathbf{W}^{(L)}$ and $\mathbf{b}^{(L)}$ using the chain rule for differentiation.

4. Experimental Investigation

4.1. Training Procedure

Our dataset contained 8768 UAV photographs, shot at different angles aiming at 611 buildings in the city of Vilnius (Wilno), taken in varying weather and lighting conditions (see Section 3.1 for further details). A total of 70% of the photos were used for training. Validation was performed with 20% of the photos and the remaining 10% of photos were used for testing. All computational operations were made on a Linux Mint 22 machine, with a Geforce 1650 GPU with 16GB of RAM and a Ryzen 3500 CPU.

A total of 1000 epochs were used for training. One epoch occupied one iteration. We see that the value of the error drops from about the 40th iteration to about 0. The standard deviation also decreased consistently. Figure 10 illustrates full loss curves until the 1000th epoch, showing stable results for the total loss, bounding box loss, classification loss, and segmentation loss. We see that from about the 35th iteration the error value drops to about 0.5 and is minimized towards the end of the training.

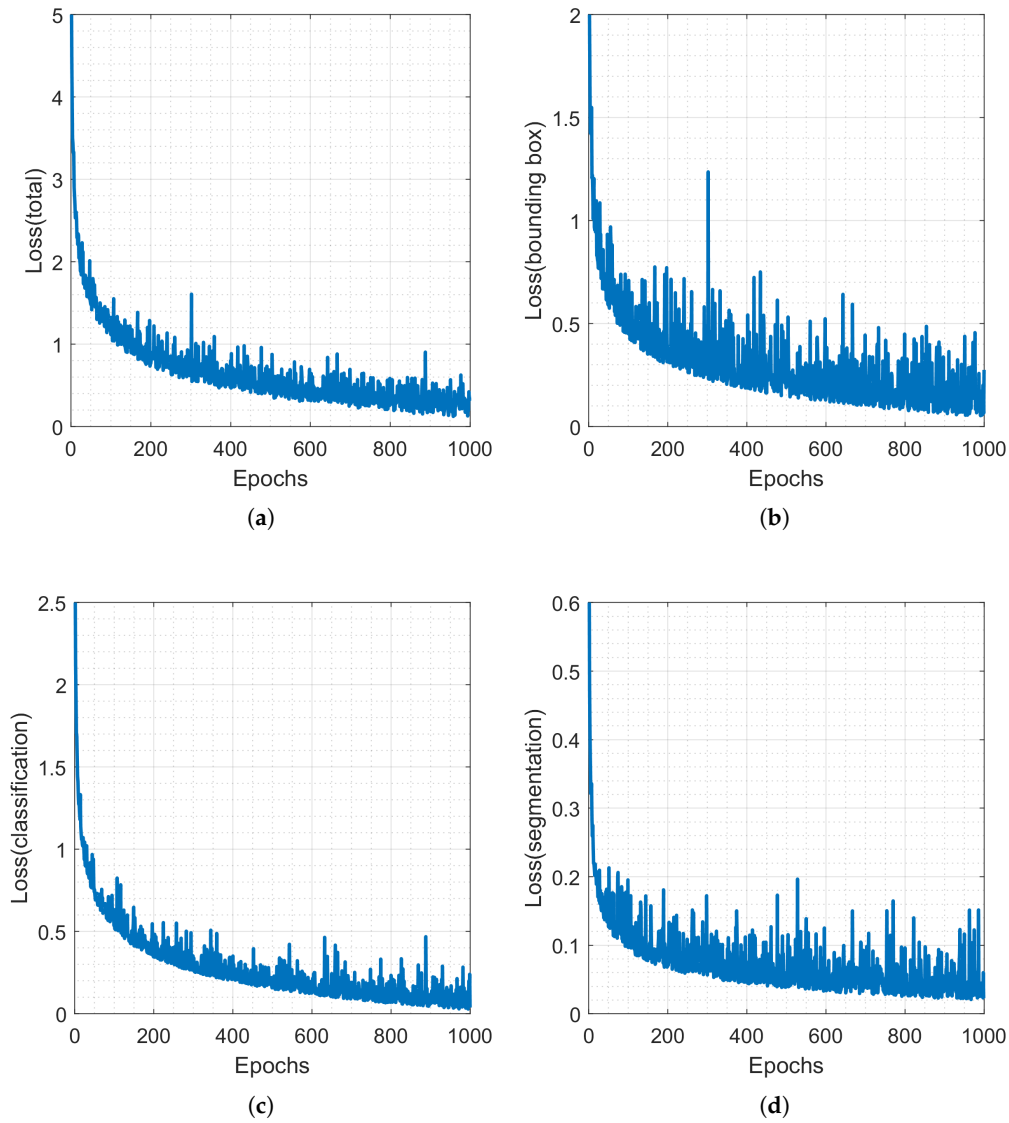


Figure 10. (**a**) Total loss; (**b**) bounding box loss; (**c**) classification loss; (**d**) segmentation loss.

4.2. Results

The buildings were classified into seven classes following the typology of building façades proposed in [27], which identified six types of building exteriors as follows:

1. transparent—using apertures, windows, doors, cracks, curtain walls, and other features that allow for unhindered communication between the interior and the outside,
2. opaque—where there is no clear visual distinction between the interior and the outside due to the usage of entire walls, large divisions, and closures;
3. blended compositions;
4. composed shape, which relies design on proportions, architectural detail composition, or rhythm;
5. decomposed shape refers to intentional distortion or omission of proportions, the composition of architectural elements, or rhythm;
6. mixed shape refers to the connection between distortion or absence of proportions, the composition of architectural details, or rhythm.

An additional class (7) is "Other", which includes all types of façades not covered by the aforementioned typology.

Classification results are illustrated in Figure 11, where confusion matrices indicated an average accuracy of 98.41% in clear view settings (top left), 88.11% in rain (top right), and 82.95% when the picture was partially blocked by other objects and was in the shadows (bottom left).

We have also experimented with the Harward UAV dataset containing images of buildings strictly from above [96]. None of the images were used in training, yet our approach was still able to achieve acceptable 88.6% accuracy in building detection (bottom right matrix in Figure 11). The network, unfortunately, failed to assign the correct class as the images from above lacked necessary information.

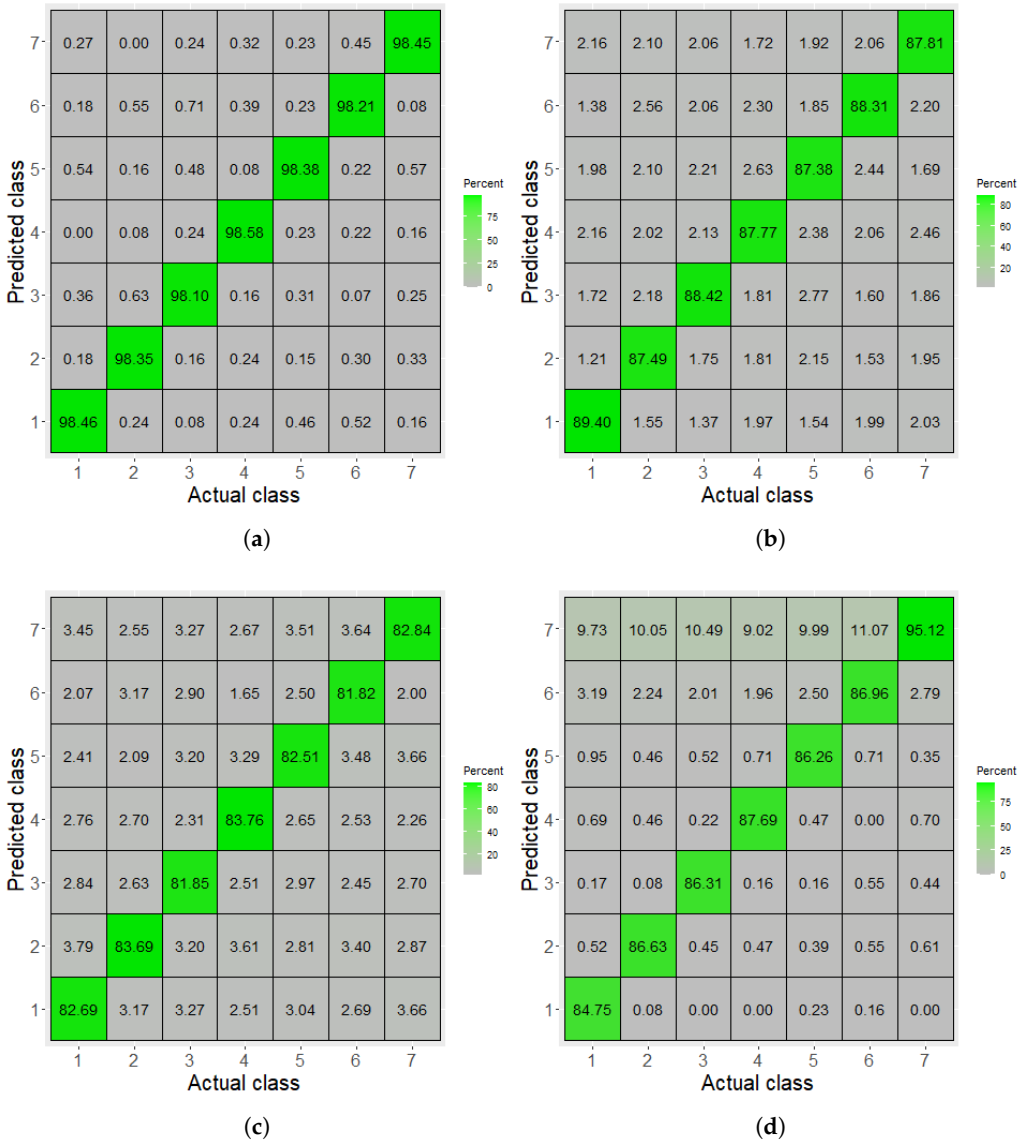


Figure 11. Confusion matrix of building façade classification in: (**a**) clear weather conditions; (**b**) rainy conditions; (**c**) partially blocked or in shadow. (**d**) shows the classification result from the Harward UAV dataset [96].

Figure 12 illustrates the accuracy of classification in the same scenarios. The results suggest that our technique is relatively stable in our Vilnius (Wilno) dataset under all use cases, as the fluctuation between the seven groups classified is not significant. Unfortunately, the classification results of the Harward UAV dataset show a substantial distribution towards the seventh class of façades (others), as this dataset covers structures shot from the

top and lacks differentiating façade information. Deviation values and mIoU values are shown in Table 1.

Table 1. Calculated deviation and mIoU values.

Class	Deviation Value	mIoU Value	Scenario
1	0.91	0.51	clear weather conditions
2	1.00	0.71	
3	0.91	0.72	
4	0.72	0.70	
5	1.28	0.75	
6	1.00	0.77	
7	0.52	0.69	
1	0.77	0.61	rainy conditions
2	0.33	0.65	
3	1.48	0.62	
4	1.76	0.61	
5	1.31	0.64	
6	0.78	0.69	
7	0.53	0.61	
1	0.83	0.55	partially blocked or in shadow
2	0.72	0.60	
3	1.16	0.61	
4	1.81	0.59	
5	0.97	0.57	
6	0.84	0.59	
7	1.74	0.55	
1	1.87	0.67	Harward UAV dataset
2	0.72	0.71	
3	0.52	0.62	
4	0.06	0.78	
5	0.60	0.69	
6	0.96	0.71	
7	0.82	0.59	

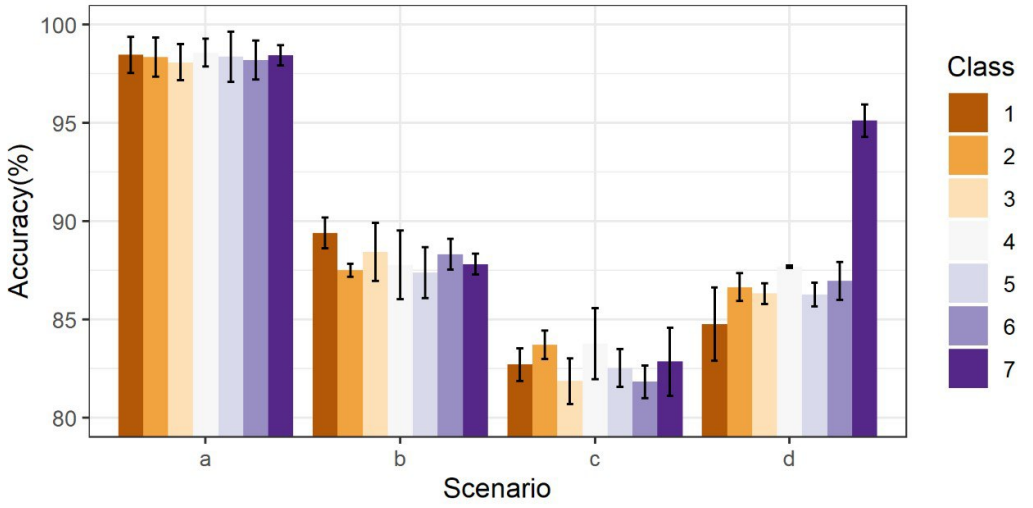


Figure 12. Accuracy of classification by class in different scenarios: (**a**) clear weather conditions; (**b**) rainy conditions; (**c**) partially blocked or in shadow. (**d**) shows the classification result from the Harward UAV dataset [96].

Figure 13 illustrates a sample result, showing the processing result of the algorithm, where every building had correctly marked boundaries even in the cases of multi-angled shots, yet there were some masking errors, which is not surprising, given the complexity of this particular area. Note that some of the large buildings shown are actually two or more separate entities with joint walls (in the example identifiable by balustrades and top floor decorations). Figure 14 shows an OpenStreetMap building overlay of the same location.

Figure 13. UAV camera images showing G⊙ center and M⊙ museum area at different angles. The red mark indicates a detected façade shape. Color highlights show assigned segment masks.

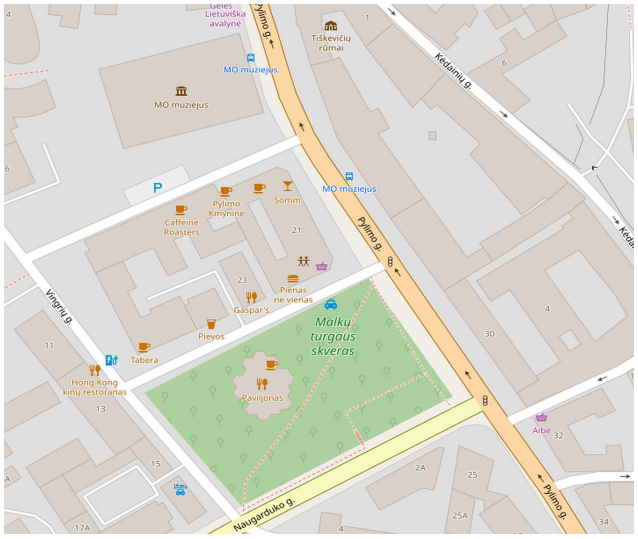


Figure 14. OpenStreetMap building overlay of the G⊙ center and M⊙ museum area [97].

5. Discussion and Conclusions

The main contributions of this study are a novel method for the segmentation of city buildings against a city background; the analysis of building features that allow for successful segmentation and identification of buildings against a city background; and the analysis of natural factors that influence the quality of building segmentation and identification. To assess the limitations of the methodology and threats to validity, we

assessed the impact of ambient noise. Outdoor image processing becomes problematic due to dynamically changing weather conditions. During the day, when the sun is in different positions, the contrast between the details of the building façades changes greatly in relation to the building façades and the lens. When the sun shines on the lens, the contrast of the photo decreases, and the image looks blurred. The image looks blurry even at dusk. To address this problem, we proposed a method to highlight individual parts of the building and to automatically calculate the illumination of the resulting photo. The proposed method allowed transforming the image of the building to have the façade of each building undistorted directly from the front.

As a result of this study, we can state that the proposed R-CNN-based deep neural network model was trained to identify the façades of the objects from two classes: the church and other buildings of the old town. Examination of the network with data not used for training and validation showed that the buildings were classified in the first class with 65% accuracy without using the image augmentation. With image augmentation methods with introduced transformations of images we were able to increase the number of images for training. This allowed us to improve our results by 18.74% to 87.16% compared to prior to optimization.

The applied Pareto optimization allowed achieving total accuracy of 98.41% in clear view settings, 88.11% in the rain, and 82.95% when the picture was partially blocked by other objects or was in shadow. This result is better than other state-of-the-art methods. The algorithm's robustness was also tested on the Harward UAV dataset containing images of buildings taken from above (roofs) while our approach was trained using images taken at an angle (façade still visible). Our approach was still able to achieve acceptable 88.6% accuracy in building detection, yet the network showed lower accuracy when assigning the correct façade class as images lacked necessary façade information. Naturally, such a comparison is not completely fair, as the other works more or less focused on datasets with different architectural style varieties, as well as different styles of pictures. Nonetheless, the RGB based models, such as MultiDefectNet were achieving around 62.7% in [98], Zhang's model achieved 81.6% [99],the K-means cluster algorithm 82% [100] accuracy, while point cloud-based models were exhibiting similar performance, e.g., Zolonvari's model achieved 86% [101], DLA-net was efficient up to 83% [102], and LFA-net up to 80.9% [103], as illustrated in Table 2.

Table 2. Comparison of classification accuracy with other approaches.

Method	Data type	Accuracy and Conditions
MultiDefectNet [98]	RGB	62.7% (varying)
DETR (transformer+FNN hybrid) [99]	RGB	81.6% (clear)
K-means [100]	RGB	82% (clear)
Pointcloud slicing [101]	Depth	86% (unspecified)
DLA-net [102]	Depth	83% (unspecified)
LFA-net [103]	Depth	80.9 % (unspecified)
R-CNN with Pareto optimization (ours)	RGB	**98.41% (clear), 88.11% (rain), 82.95% (partially blocked)**

Overall, the obtained results suggest that this algorithm would someday be suitable for recognizing the building façades in specific cases, with its accuracy still somewhat dependent on weather conditions and general illumination. Potentially, increasing the number of training images could improve the reliability of building identification. However, again, there is a high risk that the analysis of other buildings in another area will require a modification of the structure of the network itself, for example with art-deco buildings or those with circular shapes.

Author Contributions: Conceptualization, A.K., A.M.; Data curation, A.K., D.P., T.S., A.M. and R.D.; Formal analysis, R.M., A.K., D.P., T.S., A.M. and R.D.; Funding acquisition, R.M.; Investigation, R.M. and R.D.; Methodology, D.P.; Project administration, D.P.; Resources, A.K. and D.P.; Software, R.M.; Validation, R.M. and R.D.; Visualization, R.D.; Writing–original draft, R.M., A.K. and D.P.; Writing–review & editing, R.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data are available from the corresponding author upon reasonable request.

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

Abbreviations

The following abbreviations were used in this manuscript:

UAV	Unmanned Aerial Vehicle
BIM	Building Information Modeling
AI	Artificial Intelligence
UNESCO	United Nations Educational, Scientific and Cultural Organization
DNN	Deep Neural Network
CNN	Convolutional Neural Network
SAR	Synthetic Aperture Radar
U-Net	Variant of Convolutional Neural Network
SiU-Net	Variant of Convolutional Neural Network
MASK R-CNN	Variant of Convolutional Neural Network
GSN	Graph Convolutional Network
ViT	Vision Transformer
RF	Random Forest Algorithm
ELM	Extreme Learning Machine
RANSAC	Random Sample Consensus
CMOS	(Complementary Metal Oxide Semiconductor) Image Sensor
FOV	Field of View
R-CNN	Variant of Convolutional Neural Network
FCN	Full Convolutional Network
ReLU	Rectified Linear Unit
FC	Fully Connected
ROI	Region of Interest
LFA-net	Variant of Convolutional Neural Network
MultiDefectNet	Variant of Convolutional Neural Network
DLA-net	Variant of Convolutional Neural Network

References

1. Al Ridhawi, I.; Bouachir, O.; Aloqaily, M.; Boukerche, A. Design Guidelines for Cooperative UAV-supported Services and Applications. *ACM Comput. Surv.* **2022**, *54*, 1–35. [CrossRef]
2. Vizvári, B.; Golabi, M.; Nedjati, A.; Gümüşbuğa, F.; Izbirak, G. Top-down approach to design the relief system in a metropolitan city using UAV technology, part I: The first 48 h. *Nat. Hazards* **2019**, *99*, 571–597. [CrossRef]
3. Ramazan, E.; Oya, E.; Niyazi, A. Accuracy Assessment of Low Cost UAV Based City Modelling for Urban Planning. *Tehnicki Vjesnik* **2018**, *25*, 1708–1714. [CrossRef]
4. Bai, Z.; Li, Y.; Chen, X.; Yi, T.; Wei, W.; Wozniak, M.; Damasevicius, R. Real-time video stitching for mine surveillance using a hybrid image registration method. *Electronics* **2020**, *9*, 1336. [CrossRef]
5. Masood, H.; Zafar, A.; Ali, M.U.; Hussain, T.; Khan, M.A.; Tariq, U.; Damaševičius, R. Tracking of a Fixed-Shape Moving Object Based on the Gradient Descent Method. *Sensors* **2022**, *22*, 1098. [CrossRef]
6. Tan, Y.; Li, S.; Liu, H.; Chen, P.; Zhou, Z. Automatic inspection data collection of building surface based on BIM and UAV. *Autom. Constr.* **2021**, *131*, 103881. [CrossRef]
7. Freimuth, H.; König, M. Planning and executing construction inspections with unmanned aerial vehicles. *Autom. Constr.* **2018**, *96*, 540–553. [CrossRef]
8. Bolourian, N.; Hammad, A. LiDAR-equipped UAV path planning considering potential locations of defects for bridge inspection. *Autom. Constr.* **2020**, *117*, 103250. [CrossRef]

9. Denhof, D.; Staar, B.; Lütjen, M.; Freitag, M. Automatic Optical Surface Inspection of Wind Turbine Rotor Blades using Convolutional Neural Networks. *Procedia CIRP* **2019**, *81*, 1166–1170. [CrossRef]
10. Özaslan, T.; Shen, S.; Mulgaonkar, Y.; Michael, N.; Kumar, V. Inspection of Penstocks and Featureless Tunnel-like Environments Using Micro UAVs. In *Field and Service Robotics*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 123–136. [CrossRef]
11. Vetrivel, A.; Gerke, M.; Kerle, N.; Nex, F.; Vosselman, G. Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 45–59. [CrossRef]
12. Fu, G.; Liu, C.; Zhou, R.; Sun, T.; Zhang, Q. Classification for High Resolution Remote Sensing Imagery Using a Fully Convolutional Network. *Remote Sens.* **2017**, *9*, 498. [CrossRef]
13. Okewu, E.; Misra, S.; Fernandez Sanz, L.; Ayeni, F.; Mbarika, V.; Damaševičius, R. Deep neural networks for curbing climate change-induced farmers-herdsmen clashes in a sustainable social inclusion initiative. *Probl. Ekorozwoju* **2019**, *14*, 143–155.
14. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking, 2018. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018. [CrossRef]
15. Lyu, Y.; Vosselman, G.; Xia, G.S.; Yilmaz, A.; Yang, M.Y. UAVid: A semantic segmentation dataset for UAV imagery. *ISPRS J. Photogramm. Remote Sens.* **2020**, *165*, 108–119. [CrossRef]
16. Hosseini, S.M.; Mohammadi, M.; Rosemann, A.; Schröder, T.; Lichtenberg, J. A morphological approach for kinetic façade design process to improve visual and thermal comfort: Review. *Build. Environ.* **2019**, *153*, 186–204. [CrossRef]
17. Barrile, V.; Fotia, A.; Candela, G.; Bernardo, E. Integration of 3D model from UAV survey in Bim environment. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *XLII-2/W11*, 195–199. [CrossRef]
18. Wojtkowska, M.; Kedzierski, M.; Delis, P. Validation of terrestrial laser scanning and artificial intelligence for measuring deformations of cultural heritage structures. *Measurement* **2021**, *167*, 108291. [CrossRef]
19. Tan, Y.; Li, G.; Cai, R.; Ma, J.; Wang, M. Mapping and modelling defect data from UAV captured images to BIM for building external wall inspection. *Autom. Constr.* **2022**, *139*, 104284. [CrossRef]
20. Ruiz, R.D.B.; Lordsleem Jr., A.C.; Rocha, J.H.A.; Irizarry, J. Unmanned aerial vehicles (UAV) as a tool for visual inspection of building facades in AEC+FM industry. *Constr. Innov.* **2021** . [CrossRef]
21. Templin, T.; Popielarczyk, D. The use of low-cost unmanned aerial vehicles in the process of building models for cultural tourism, 3D web and augmented/mixed reality applications. *Sensors* **2020**, *20*, 5457. [CrossRef]
22. Kang, D.; Cha, Y.J. Autonomous UAVs for Structural Health Monitoring Using Deep Learning and an Ultrasonic Beacon System with Geo-Tagging. *Comput. Civ. Infrastruct. Eng.* **2018**. *33*, 885–902. [CrossRef]
23. Zhou, B.; Duan, X.; Ye, D.; Wei, W.; Woźniak, M.; Połap, D.; Damaševičius, R. Multi-level features extraction for discontinuous target tracking in remote sensing image monitoring. *Sensors* **2019**, *19*, 4855. [CrossRef] [PubMed]
24. Saska, M.; Kratky, V.; Spurny, V.; Baca, T. Documentation of dark areas of large historical buildings by a formation of unmanned aerial vehicles using model predictive control. In Proceedings of the 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus, 12–15 September 2017. [CrossRef]
25. Liu, C.; Li, W.; Lei, W.; Liu, L.; Wu, H. Architecture planning and geo-disasters assessment mapping of landslide by using airborne lidar data and UAV images. In Proceedings of the SPIE, Nanjing, China, 24 October 2011. [CrossRef]
26. Wilno—A Unique Unesco Historical City. Available online: https://whc.unesco.org/en/list/541/ (accessed on 25 September 2022).
27. Jabłońska, J.; Telesińska, M.; Adamska, A.; Gronostajska, J. The Architectural Typology of Contemporary Façades for Public Buildings in the European Context. *Arts* **2022**, *11*, 11. [CrossRef]
28. Wang, X.; Wen, K.L.; Ying, X.H.; Chen, H.C. Appling semantic method to windows detection in facade. In Proceedings of the 2011 IEEE/SICE International Symposium on System Integration (SII), Kyoto, Japan, 20–22 December 2011. [CrossRef]
29. Prathap, G.; Afanasyev, I. Deep Learning Approach for Building Detection in Satellite Multispectral Imagery. In Proceedings of the 2018 International Conference on Intelligent Systems (IS), Funchal, Portuga, 25–27 September 2018. [CrossRef]
30. Martinovic, A.; Gool, L.V. Hierarchical Co-Segmentation of Building Facades. In Proceedings of the 2014 2nd International Conference on 3D Vision, Tokyo, Japan, 8–11 December 2014. [CrossRef]
31. Bischke, B.; Helber, P.; Folz, J.; Borth, D.; Dengel, A. Multi-Task Learning for Segmentation of Building Footprints with Deep Neural Networks. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019. [CrossRef]
32. Hascoet, N.; Zaharia, T. Building recognition with adaptive interest point selection. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 8–10 January 2017. [CrossRef]
33. Noronha, S.; Nevatia, R. Detection and description of buildings from multiple aerial images. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 17–19 June 1997. [CrossRef]
34. Liu, Y.; Zhang, Z.; Zhong, R.; Chen, D.; Ke, Y.; Peethambaran, J.; Chen, C.; Sun, L. Multilevel Building Detection Framework in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2018**, *11*, 3688–3700. [CrossRef]
35. Ivanovsky, L.; Khryashchev, V.; Pavlov, V.; Ostrovskaya, A. Building Detection on Aerial Images Using U-NET Neural Networks. In Proceedings of the 2019 24th Conference of Open Innovations Association (FRUCT), Moscow, Russia, 8–12 April 2019. [CrossRef]

36. Gadde, R.; Jampani, V.; Marlet, R.; Gehler, P.V. Efficient 2D and 3D Facade Segmentation Using Auto-Context. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1273–1280. [CrossRef]
37. Vakalopoulou, M.; Karantzalos, K.; Komodakis, N.; Paragios, N. Building detection in very high resolution multispectral data with deep learning features. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015. [CrossRef]
38. Zhang, W.; Tang, P.; Zhao, L.; Huang, Q. A Comparative Study of U-Nets with Various Convolution Components for Building Extraction. In Proceedings of the 2019 Joint Urban Remote Sensing Event (JURSE), Vannes, France, 22–24 May 2019. [CrossRef]
39. Hui, J.; Du, M.; Ye, X.; Qin, Q.; Sui, J. Effective Building Extraction From High-Resolution Remote Sensing Images With Multitask Driven Deep Neural Network. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 786–790. [CrossRef]
40. Chen, G.; Li, C.; Wei, W.; Jing, W.; Woźniak, M.; Blažauskas, T.; Damaševičius, R. Fully convolutional neural network with augmented atrous spatial pyramid pool and fully connected fusion path for high resolution remote sensing image segmentation. *Appl. Sci.* **2019**, *9*, 1816. [CrossRef]
41. Xia, L.; Zhang, X.; Zhang, J.; Wu, W.; Gao, X. Refined extraction of buildings with the semantic edge-assisted approach from very high-resolution remotely sensed imagery. *Int. J. Remote Sens.* **2020**, *41*, 8352–8365. [CrossRef]
42. Zhuo, X.; Monks, M.; Esch, T.; Reinartz, P. Facade Segmentation from Oblique UAV Imagery. In Proceedings of the 2019 Joint Urban Remote Sensing Event (JURSE), Vannes, France, 22–24 May 2019. [CrossRef]
43. Jampani, V.; Gadde, R.; Gehler, P.V. Efficient Facade Segmentation Using Auto-context. In Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 5–9 January 2015. [CrossRef]
44. Recky, M.; Leberl, F. Windows Detection Using K-means in CIE-Lab Color Space. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010. [CrossRef]
45. Zhang, J.; Fukuda, T.; Yabuki, N. Development of a City-Scale Approach for Façade Color Measurement with Building Functional Classification Using Deep Learning and Street View Images. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 551. [CrossRef]
46. Dai, M.; Ward, W.O.; Meyers, G.; Densley Tingley, D.; Mayfield, M. Residential building facade segmentation in the urban environment. *Build. Environ.* **2021**, *199*, 107921. [CrossRef]
47. Bischke, B.; Helber, P.; Hees, J.; Dengel, A. Location-Specific Embedding Learning for the Semantic Segmentation of Building Footprints on a Global Scale. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019. [CrossRef]
48. Ji, S.; Wei, S.; Lu, M. Fully Convolutional Networks for Multisource Building Extraction From an Open Aerial and Satellite Imagery Data Set. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 574–586. [CrossRef]
49. Gorobets, A.N. Segmentation for detecting buildings in infrared space images. In Proceedings of the 2017 XI International Conference on Antenna Theory and Techniques (ICATT), Kyiv, UKraine, 24–27 May 2017. [CrossRef]
50. Lin, K.; Huang, B.; Collins, L.M.; Bradbury, K.; Malof, J.M. A simple rotational equivariance loss for generic convolutional segmentation networks: Preliminary results. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019. [CrossRef]
51. Zhang, Z.; Guo, W.; Yu, W.; Yu, W. Multi-task fully convolutional networks for building segmentation on SAR image. *J. Eng.* **2019**, *2019*, 7074–7077. [CrossRef]
52. Zorzi, S.; Fraundorfer, F. Regularization of Building Boundaries in Satellite Images Using Adversarial and Regularized Losses. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019. [CrossRef]
53. Li, Y.; Ding, Z.; Miao, W.; Zhang, M.; Li, W.; Ye, W. Low-cost 3D Building Modeling via Image Processing. In Proceedings of the 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 18–20 October 2019. [CrossRef]
54. Shi, Y.; Li, Q.; Zhu, X. Building Footprint Extraction with Graph Convolutional Network. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019. [CrossRef]
55. Michaelsen, E.; Soergel, U.; Schunert, A.; Doktorski, L.; Jaeger, K. Perceptual grouping for building recognition from satellite SAR image stacks. In Proceedings of the 2010 IAPR Workshop on Pattern Recognition in Remote Sensing, Istanbul, Turkey, 22 August 2010. [CrossRef]
56. Tao, Y.; Zhang, Y.T.; Chen, X.J. Element-Arrangement Context Network for Facade Parsing. *J. Comput. Sci. Technol.* **2022**, *37*, 652–665. [CrossRef]
57. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**.
58. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**.
59. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A Survey on Vision Transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022** . [CrossRef]
60. Wang, L.; Fang, S.; Meng, X.; Li, R. Building Extraction With Vision Transformer. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–11. [CrossRef]
61. Chen, C.F.R.; Fan, Q.; Panda, R. CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10-17 October 2021; pp. 357–366.

62. Guo, Y.; Wang, C.; Yu, S.X.; McKenna, F.; Law, K.H. AdaLN: A Vision Transformer for Multidomain Learning and Predisaster Building Information Extraction from Images. *J. Comput. Civ. Eng.* **2022**, *36*, 04022024. [CrossRef]
63. Chen, K.; Zou, Z.; Shi, Z. Building Extraction from Remote Sensing Images with Sparse Token Transformers. *Remote Sens.* **2021**, *13*, 4441. [CrossRef]
64. Bazi, Y.; Bashmal, L.; Rahhal, M.M.A.; Dayil, R.A.; Ajlan, N.A. Vision Transformers for Remote Sensing Image Classification. *Remote Sens.* **2021**, *13*, 516. [CrossRef]
65. Bashmal, L.; Bazi, Y.; Rahhal, M.M.A.; Alhichri, H.; Ajlan, N.A. UAV Image Multi-Labeling with Data-Efficient Transformers. *Appl. Sci.* **2021**, *11*, 3974. [CrossRef]
66. Li, C.; Tang, T.; Wang, G.; Peng, J.; Wang, B.; Liang, X.; Chang, X. BossNAS: Exploring Hybrid CNN-Transformers With Block-Wisely Self-Supervised Neural Architecture Search. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10-17 October 2021; pp. 12281–12291.
67. Costache, A.; Popescu, D.; Ichim, L.; Mocanu, S. Building Recognition in Static Images. In Proceedings of the 2019 23rd International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania, 9–11 October 2019. [CrossRef]
68. Muthukrishnan, R.; Radha, M. Edge Detection Techniques For Image Segmentation. *Int. J. Comput. Sci. Inf. Technol.* **2011**, *3*, 259–267. [CrossRef]
69. Wan, G.; Li, S. Automatic facades segmentation using detected lines and vanishing points. In Proceedings of the 2011 4th International Congress on Image and Signal Processing, Shanghai, China, 15–17 October 2011. [CrossRef]
70. Hernandez, J.; Marcotegui, B. Morphological segmentation of building façade images. In Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009. [CrossRef]
71. Tian, S.; Zhang, Y.; Zhang, J.; Su, N. A Novel Deep Embedding Network for Building Shape Recognition. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 2127–2131. [CrossRef]
72. Delmerico, J.A.; David, P.; Corso, J.J. Building facade detection, segmentation, and parameter estimation for mobile robot localization and guidance. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011. [CrossRef]
73. Li, B.; Sun, F.; Zhang, Y. Building Recognition Based on Sparse Representation of Spatial Texture and Color Features. *IEEE Access* **2019**, *7*, 37220–37227. [CrossRef]
74. Sun, Z.; Zhao, G.; Woźniak, M.; Scherer, R.; Damaševičius, R. Bankline detection of GF-3 SAR images based on shearlet. *PeerJ Comput. Sci.* **2021**, *7*, e611. [CrossRef] [PubMed]
75. Shu, C.; Sun, L.; Li, J.; Gou, M. Remote sensing image restoration: An adaptive reciprocal cell recovery technique. *Inf. Technol. Control* **2018**, *47*, 704–713. [CrossRef]
76. Sun, Z.; Shi, R.; Wei, W. Synthetic-aperture radar image despeckling based on improved non-local means and non-subsampled shearlet transform. *Inf. Technol. Control* **2020**, *49*, 299–307. [CrossRef]
77. Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; Huang, T. Free-Form Image Inpainting With Gated Convolution. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019. [CrossRef]
78. Cao, J.; Metzmacher, H.; O'Donnell, J.; Frisch, J.; Bazjanac, V.; Kobbelt, L.; van Treeck, C. Facade geometry generation from low-resolution aerial photographs for building energy modeling. *Build. Environ.* **2017**, *123*, 601–624. [CrossRef]
79. Burochin, J.P.; Vallet, B.; Brédif, M.; Mallet, C.; Brosset, T.; Paparoditis, N. Detecting blind building façades from highly overlapping wide angle aerial imagery. *ISPRS J. Photogramm. Remote Sens.* **2014**, *96*, 193–209. [CrossRef]
80. Ali, D.; Verstockt, S.; Weghe, N.V.D. Single Image Façade Segmentation and Computational Rephotography of House Images Using Deep Learning. *J. Comput. Cult. Herit.* **2021**, *14*, 1–17. [CrossRef]
81. Kedzierski, M.; Wierzbicki, D. Radiometric quality assessment of images acquired by UAV's in various lighting and weather conditions. *Measurement* **2015**, *76*, 156–169. [CrossRef]
82. Liu, C.; Shirowzhan, S.; Sepasgozar, S.M.E.; Kaboli, A. Evaluation of Classical Operators and Fuzzy Logic Algorithms for Edge Detection of Panels at Exterior Cladding of Buildings. *Buildings* **2019**, *9*, 40. [CrossRef]
83. Veerashetty, S.; Patil, N.B. Novel LBP based texture descriptor for rotation, illumination and scale invariance for image texture analysis and classification using multi-kernel SVM. *Multimed. Tools Appl.* **2019**, *79*, 9935–9955. [CrossRef]
84. Zhang, Z.; Cheng, X.; Wu, J.; Zhang, L.; Li, Y.; Wu, Z. The "Fuzzy" Repair of Urban Building Facade Point Cloud Based on Distribution Regularity. *Remote Sens.* **2022**, *14*, 1090. [CrossRef]
85. Soycan, A.; Soycan, M. Perspective correction of building facade images for architectural applications. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 697–705. [CrossRef]
86. Mavic Air Drone Specification. Available online: https://www.dji.com/lt/mavic-air (accessed on 25 September 2022)
87. Marija Drėmaitė, Rūta Leitanaitė, J.R.E. *Vilnius 1900–2016. An Architectural Guide*; Lapas: Vilnius, Lithuania, 2016.
88. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and Flexible Image Augmentations. *Information* **2020**, *11*, 125. [CrossRef]
89. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 21–26 July 2017; pp. 1125–1134.
90. Nanni, L.; Paci, M.; Brahnam, S.; Lumini, A. Comparison of Different Image Data Augmentation Approaches. *J. Imaging* **2021**, *7*, 254. [CrossRef]

91. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014. [CrossRef]
92. Wang, Y.; Li, S.; Teng, F.; Lin, Y.; Wang, M.; Cai, H. Improved Mask R-CNN for Rural Building Roof Type Recognition from UAV High-Resolution Images: A Case Study in Hunan Province, China. *Remote Sens.* **2022**, *14*, 265. [CrossRef]
93. Plonis, D.; Katkevicius, A.; Gurskas, A.; Urbanavicius, V.; Maskeliunas, R.; Damasevicius, R. Prediction of Meander Delay System Parameters for Internet-of-Things Devices Using Pareto-Optimal Artificial Neural Network and Multiple Linear Regression. *IEEE Access* **2020**, *8*, 39525–39535. [CrossRef]
94. Michel, G.; Alaoui, M.A.; Lebois, A.; Feriani, A.; Felhi, M. DVOLVER: Efficient Pareto-Optimal Neural Network Architecture Search. *arXiv* **2019**, arXiv.1902.01654.
95. pereira da Silva, E. Modelos de Crescimento e das Respostas de Frangas de Postura Submetidas a Diferentes Ingestoes de Aminoacidos Sulfurados. Available online: https://acervodigital.unesp.br/handle/11449/104074(accessedon25September2022).
96. Liu, Y. UAV Building Segmentation Dataset. 2020. Available online: https://dataverse.harvard.edu/citation?persistentId=doi:10.7910/DVN/ACVYY9 (accessed on 25 September 2022).
97. Openstreetmap Building Overlay of the G Center and MO Museum Area. Available online: https://www.openstreetmap.org/search?query=mo%20museum#map=19/54.67934/25.27751 (accessed on 25 September 2022).
98. Lee, K.; Hong, G.; Sael, L.; Lee, S.; Kim, H.Y. MultiDefectNet: Multi-Class Defect Detection of Building Façade Based on Deep Convolutional Neural Network. *Sustainability* **2020**, *12*, 9785. [CrossRef]
99. Zhang, G.; Pan, Y.; Zhang, L. Deep learning for detecting building façade elements from images considering prior knowledge. *Autom. Constr.* **2022**, *133*, 104016. [CrossRef]
100. Mao, B.; Li, B. Building façade semantic segmentation based on K-means classification and graph analysis. *Arab. J. Geosci.* **2019**, *12*, 253. [CrossRef]
101. Zolanvari, S.I.; Laefer, D.F.; Natanzi, A.S. Three-dimensional building façade segmentation and opening area detection from point clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 134–149. [CrossRef]
102. Su, Y.; Liu, W.; Yuan, Z.; Cheng, M.; Zhang, Z.; Shen, X.; Wang, C. DLA-Net: Learning dual local attention features for semantic segmentation of large-scale building facade point clouds. *Pattern Recognit.* **2022**, *123*, 108372. [CrossRef]
103. Su, Y.; Liu, W.; Cheng, M.; Yuan, Z.; Wang, C. Local Fusion Attention Network for Semantic Segmentation of Building Facade Point Clouds. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]