applied sciences

MDPI

Article

Automatic Building Height Estimation: Machine Learning Models for Urban Image Analysis

Miguel Ureña-Pliego [1,*,†], Rubén Martínez-Marín [1,†], Beatriz González-Rodrigo [2,†] and Miguel Marchamalo-Sacristán [1,*,†]

[1] Department of Land Morphology and Engineering, Civil Engineering School, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[2] Department of Environmental and Forestry Engineering and Management, Civil Engineering School, Universidad Politécnica de Madrid, 28040 Madrid, Spain
* Correspondence: miguel.urena@upm.es (M.U.-P.); miguel.marchamalo@upm.es (M.M.-S.)
† Current address: ETSI Caminos, Canales y Puertos Universidad, Universidad Politécnica de Madrid, Calle del Prof. Aranguren, 3, 28040 Madrid, Spain.

Abstract: Artificial intelligence (AI) is delivering major advances in the construction engineering sector in this era of building information modelling, applying data collection techniques based on urban image analysis. In this study, building heights were calculated from street-view imagery based on a semantic segmentation machine learning model. The model has a fully convolutional architecture and is based on the HRNet encoder and ResNexts depth separable convolutions, achieving fast runtime and state-of-the-art results on standard semantic segmentation tasks. Average building heights on a pilot German street were satisfactorily estimated with a maximum error of 3 m. Further research alternatives are discussed, as well as the difficulties of obtaining valuable training data to apply these models in countries with no training datasets and different urban conditions. This line of research contributes to the characterisation of buildings and the estimation of attributes essential for the assessment of seismic risk using automatically processed street-view imagery.

Keywords: artificial intelligence; semantic segmentation; convolutional neural networks; building height estimation; seismic exposure; street view imagery

Citation: Ureña-Pliego, M.; Martínez-Marín, R.; González-Rodrigo, B.; Marchamalo-Sacristán, M. Automatic Building Height Estimation: Machine Learning Models for Urban Image Analysis. *Appl. Sci.* **2023**, *13*, 5037. https://doi.org/10.3390/app13085037

Academic Editors: Agostino Forestiero and Antonio Mannella

Received: 21 March 2023
Revised: 5 April 2023
Accepted: 14 April 2023
Published: 17 April 2023

1. Introduction

Digital transformation in the construction sector can lead to the introduction of disruptive technologies and tools that can result in innovative business models, materials, and solutions that benefit the entire value chain [1]. Artificial intelligence is considered within one of the three categories of digital technologies in construction, namely data information and analysis [2]. It has been observed that the adoption of artificial intelligence in the construction industry has been limited to pilot projects, with tests being conducted in structural analysis, design, and optimisation. Results show that the implementation of artificial neural networks can be used for structural damage assessment (e.g., detecting structural damage after earthquakes) or structural health monitoring (e.g., identification of damage and nonlinearities in wind turbine blades based on a pattern recognition technique).

In recent decades, managers and scientists have faced the challenging task of developing tools to assess the seismic vulnerability of existing building parks to define reliable risk-mitigation plans. Several large-scale approaches have been proposed to identify the most vulnerable sets in a building stock, using established procedures [3,4]. In this field, urban image analysis is being used to estimate the seismic exposure of buildings, an essential step to calculate their seismic vulnerability. There is a need for new automated tools capable of rapidly estimating the vulnerability of existing buildings from pictures, such as VULMA, a tool for the evaluation of the vulnerability response of individual buildings and for large-scale seismic risk estimation [5]. This line of research necessarily

includes the development of specific training datasets for this purpose, as proposed by Cardellicchio et al. [6].

The determination of seismic exposure is based on the characterisation of the attributes of the structure with different levels of detail to define the class of exposure of each building [7], as in the methods proposed by the Global Earthquake Model (GEM) foundation [4]. Previous research identified that one of the most influential attributes for seismic exposure classification is the height of the building or the number of floors [7,8]. As the attribute that is needed is the number of floors, a building height estimation with a maximum absolute error of one floor would be sufficient. There is a need for automated machine learning models to estimate this parameter in cities with no available data, as is the case of the capitals in Central America, subjected to high potential seismic hazard and high physical and social vulnerability [9].

Street view images are readily available in cities in developing countries or can be obtained at little cost. It is necessary to develop models capable of obtaining building heights from single-view images, allowing researchers to obtain this parameter in cities in developing countries from images available on the Internet or captured using mobile phones, creating the necessary databases for obtaining the seismic exposure of the buildings in a city.

The aim of this research is to develop low-cost methods for building height data collection for the purposes of seismic risk assessment in cities in developing countries in Central America. This research contributes to determining the exposure of buildings using street view images processed by various convolutional neural networks. This will provide valuable tools that accurately estimate relevant seismic exposure features using free imagery, thus being easily applicable to the monitoring of large areas of cities in developing countries.

This research applies technology originally developed for autonomous car driving to the gathering of building data and automatic building surveying. A new model is proposed combining RestNext and HRNet, and a method is developed for building height estimation using semantic segmentation and a single street view image.

2. Background

Research on the seismic vulnerability of buildings using imagery to extract basic parameters such as height or facade material is increasing worldwide [7]. Surveys are traditionally performed manually through visual inspection, which is time consuming and expensive. A current trend is moving towards the automation of these processes, as in the recently published VULMA model, which performs surveys automatically through a categorical classification machine learning model [10]. The reviewed experiences involve taking an image of the facade of a single building and applying the required labels using a classifier model. In real-world applications using street view images, multiple buildings and other elements such as the street are visible; hence, the classifier model would not function correctly, as it is not able to focus on the building of interest.

A building is defined in the INSPIRE scheme [11] as a construction, above and below ground, for the purpose of accommodating people, animals and things, or production and distribution of goods or services, this construction being a permanent structure on the ground. In the definition of a building, one or many BuildingParts are characterised, as each of the constructed areas in a cadastral parcel having homogeneous volume, either above and/or below ground. A BuildingPart element hosts the attributes related to height, namely, number of floors above ground, below ground level height in meters, and number of floors below ground [11].

As building heights are usually available through INSPIRE in Europe, new techniques were developed for automatic building height estimation, in part to aid data collection for cadasters in cities without data. Using ESA's Sentinel 1 and 2 data series and particularly Synthetic Aperture Radar (SAR) data [12,13], it was possible to extract building height data automatically on a large scale [13] with an error of less than 5 m for average-height buildings

but much larger error for higher buildings. Satellite techniques were improved using multi-view satellite imagery [14] achieving an error of around 2 m. Aerial imagery [15] and Aerial Laser Imaging Detection and Ranging (LIDAR) can be used to measure building heights [16], although this approach is expensive and requires manual work; thus, the viability of using it for large-scale data collection is questionable. Other recently developed methods use street-view imagery, that is, driver-view images taken from a car. The image is segmented using machine learning methods, and the top corners of the buildings are detected. The horizontal distance from the camera to the building is then calculated using the camera position and a database such as Openstreetmap, which registers the x and y coordinates of the corners of the building [17] and building rooflines [18]. Having determined certain parameters such as camera height, estimation of three vanishing points and the detection of the vertical lines of a building with the help of semantic segmentation, the height can be computed by considering the perspective of the image [19]. The relative error obtained using this method is 5%. Finally, the height can be obtained through trigonometry. Spherical images allow other more complex geometrical methods to be used which do not require 2D external corner coordinates [20].

In the field of computer vision, substantial advances have been made in recent years with regard to image classification tasks using new machine learning models such as the vision transformer [21] or its version for semantic segmentation [22]. These models are developed by large companies and contain a large number of parameters; hence, they are difficult to train. Other simpler models developed recently for image classification are RestNexts [23,24]. With regard to semantic segmentation, very good results have been achieved with HRNets [25], a model which processes an image in multiple resolutions. Computer vision is used primarily in the context of cities for tasks related to autonomous car driving, especially with regard to currently available public datasets, such as Cityscapes or Vistas [26,27]. Its use for other tasks such as building inspection or data collection has scarcely been developed to date.

3. Image Processing Methods

In this section, we present various image convolutions used for image processing, which constitute the core of our model.

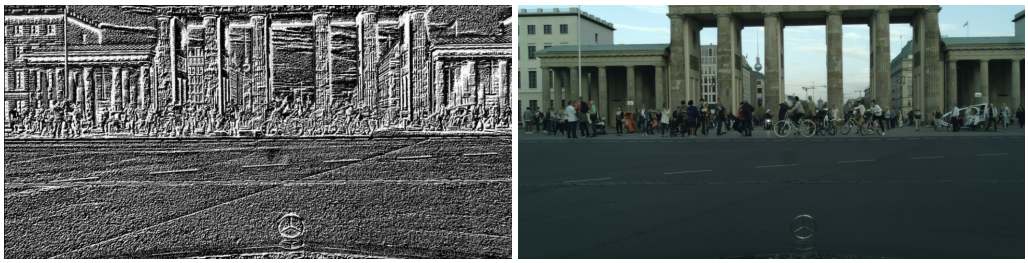
3.1. Image Convolutions

An image convolution is a mathematical operation that consists of convolving a linear kernel function over each pixel of an image. The image is interpreted as a matrix with w pixels in the width parameter and h pixels in the height parameter. A RGB image would have a third dimension containing three features, the red, green and blue channels. The image would therefore be a (w $\times$ h $\times$ 3) matrix. A kernel on an image convolution would be a matrix with a given width and height and a third dimension equal to the number of features in the image. A (500 $\times$ 250 $\times$ 3) RGB image could be convolved with a kernel of size (5 $\times$ 4 $\times$ 3) obtaining a (500 $\times$ 250 $\times$ 1) result. The convolution operation is performed by multiplying the kernel and the image pointwise and by sliding the kernel over each pixel, as shown in Figure 1.

3.2. Depthwise Separable Convolutions

A depthwise convolution consists of using one different two-dimensional convolution for each feature dimension. A (500 $\times$ 250 $\times$ 3) image could be convolved with three (5 $\times$ 4) convolutions, obtaining a (500 $\times$ 250 $\times$ 3) result. A pointwise convolution is a (1 $\times$ 1 $\times$ number of features) convolution. A (500 $\times$ 400 $\times$ 3) image could be convolved with a pointwise convolution (1 $\times$ 1 $\times$ 3), obtaining a (500 $\times$ 250 $\times$ 1) result. A depthwise separable convolution consists of applying first a depthwise and then a pointwise convolution. A layer of a convolutional network that uses depthwise convolutions applies a single depthwise and then multiple pointwise convolutions. This produces the same result as a normal convolution but requires significantly less computations [28]. This can be better

understood through an example: A ($500 \times 250 \times 3$) image is convolved with 32 normal (3×3) kernels. The number of multiplications that are performed is $500 * 250$ pixels in the input image, with $3 * 3 * 3$ parameters in each of the 32 kernels ($500 * 250 * 3 * 3 * 3 * 32 = 108$ million). However, when depthwise separable convolutions are used, a single depthwise (3×3) convolution would be applied, which has ($500 * 250$) pixels in the input image, $3 * 3$ parameters in the kernel and one different kernel for each input of the 3 input features. The pointwise convolution has $500 * 250$ pixels in the input image, with $1 * 1 * 3$ parameters in each kernel and 32 kernels. The number of computations with this type of convolution is ($500 * 250 * 3 * 3 * 3 + 500 * 250 * 1 * 1 * 3 * 32 = 15$ million) multiplications, which is more than 7 times less operations compared to normal convolutions.



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

Figure 1. Figure showing the effects of a convolution on an image. (**a**) Original image from the Cityscapes test dataset. (**b**) Result of applying a $\begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$ kernel to the grey-scaled original image. This kernel highlights edges on the image as it computes the local difference around every pixel.

3.3. Dilated Convolutions

A dilated or atrous convolution is an image convolution with spacing between the values in the kernel so that the field of view is widened, as shown in Figure 2. These convolutions are the basis for the spatial atrous pyramid layer [29]. This consists of applying a series of dilated convolutions with various dilation rates over the input and subsequently concatenating all outputs over the feature dimension. This gives the network a more global understanding of the image.

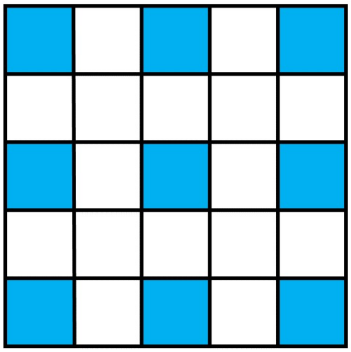


Figure 2. A (3×3) kernel with dilation rate $d = 2$ has a field of view of (5×5). Blue are the elements for the kernel and white are the skipped elements

4. Machine Learning Methods

In this section, we present a brief explanation of the machine learning algorithms and datasets available for urban image analysis.

4.1. Convolutional Neural Networks

The convolutional neural network is a type of neural network developed specifically for image processing [30]. Each layer in a convolutional network consists of applying an image convolution using various kernels and obtaining several new images. All images

are concatenated over the feature dimension. This means that, given an input image with three features, red, green and blue channels and size (width × height × 3), applying 32 different kernels would result in a matrix with 32 features and size (width × height × 32). The parameters that will be optimised are those of all kernels from all image convolutions in the network.

After a convolution is applied, a non-linear activation function is applied to each pixel of the output image. This makes the neural network a nonlinear function, achieving complex behaviour. ReLu is the most widely used activation function, as it is simple and fast to compute. On most layers, we use a variation of ReLu called GeLu, as shown in Figure 3, which performs better for classification tasks [23]. A depthwise separable convolution [28] is a type of convolution that uses even fewer parameters, achieving the same result as a normal convolution. Therefore, in our models, we almost exclusively use this type of convolution. Convolutional networks usually extract local information from an image, although it is not suitable for extracting global information, as two distant pixels in the image will not be processed by the same or by nearby kernels. To avoid this problem, models usually include a bottleneck. This consists of gradually reducing the resolution of the image and increasing the feature dimension. This brings all the pixels closer together. After running through the bottleneck for semantic segmentation tasks, the image enters the decoder, which gradually upsamples the image to its original resolution. This approach presents the disadvantage that detail is lost in the output image. The atrous spatial pyramid layer [29] is a convolutional layer that is also aimed at solving the global information problem by using dilated kernels, that is, kernels that skip proximate pixels and use those which are further away.

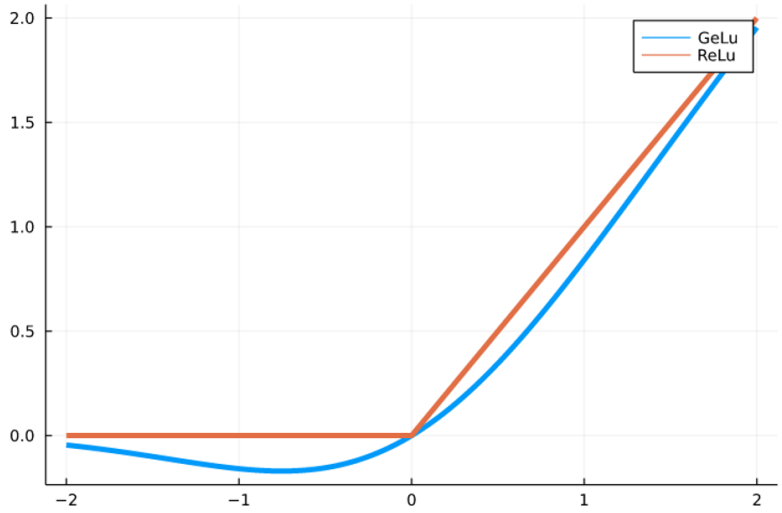


Figure 3. Comparison of the ReLu and GeLu activation functions. ReLu is simpler to compute, but GeLu avoids overfitting, as it is able to work better with negative-valued neurons.

4.2. Loss Functions

The aim of the training process is to minimise the loss function, thus achieving the best possible prediction. For classification tasks, the cross-entropy loss function (Equation (1)) is used [31]. It computes the difference between two probability distributions. Here, $.*$ denotes pointwise vector product, $\hat{y}$ is the model output and y is the real output.

$$crossentropy(\hat{y}, y) = -sum(y. * log(\hat{y})) \tag{1}$$

To obtain a probability distribution, the prediction is passed through the softmax function [32], which transforms an arbitrary vector into a vector whose sum is one and which

has elements ranging from (0) to (1) (Equation (2)). The target must also be transformed into a probability distribution.

$$softmax(x)_i = exp(x_i)/sum(exp(x_j)) \quad (2)$$

To obtain a single error value, cross-entropy is applied pointwise to the output image, and the mean is then computed.

4.3. Optimisation

Batch gradient descent [33] updates the model's parameters after computing the mean of the gradient of a certain number of training examples. This makes the steps more robust and precise but takes more computation effort and time. In our training procedure, we start updating the parameters in every single training example, and when the model is found not to be improving, the batch size is increased, updating parameters after a larger number of training examples. The ADAM optimiser [34] is used for all our models.

4.4. Categorical Classification

Categorical classification is a task in machine learning that consists of assigning a unique class to an image, thus identifying the general content of the picture [35]. For instance, in the case of a model developed for vehicle classification, when a street view image showing a car is inputted, the class "car" would be the output. If the image contained a car and a bus, the model would probably encounter difficulty, as it is not able to assign the image to a unique vehicle class. This is solved by building a model that outputs a vector with a length equal to the number of possible classes and by applying the softmax function, hence providing a kind of probability distribution expressing how likely the model considers the image to belong to a given class. The correct class must also be expressed as a probability vector. In particular, one with a length equal to the number of possible classes and which is one for the assigned class and zero for all other possible classes.

4.5. Label Smoothing

When training, the strategy of expressing classes as probability distributions can produce overfitting problems, as the model is trained to output probabilities near to one, which will usually not resemble the confidence level of the model, thus producing a model that outputs high probabilities for wrong classification. To correct this problem, label smoothing is usually applied, which consists of reducing the probabilities in the correct class probability distribution (Equation (3)). For instance, for three possible classes, the distribution used for training could be $[0.8, 0.1, 0.1]$ instead of $[1, 0, 0]$. Label smoothing is only applied for training.

$$ysmooth = 1 - \alpha y + \alpha k \quad (3)$$

α is the smooth factor, which is the probability that will be subtracted from the value of one, and k is the number of classes.

4.6. Semantic Segmentation

Semantic segmentation is similar to categorical classification, with the difference that it assigns a unique class to each pixel in an image [36]. In the example used previously, when this type of model is inputted with a street view image showing a car and a bus, the model would assign the class 'car' to the pixels belonging to the car and the class 'bus' to the pixels of the bus. In this task, losing detail in the output image is an unwanted effect of the bottleneck of convolutional networks. A new type of machine learning model has recently been developed to address this issue, the HRNet [25]. Here, the image is copied, and multiple branches are processed at the same time. Each branch has a different resolution, thus having at least one high-resolution branch which helps maintain all the detail in the output image and a low-resolution branch that provides the bottleneck effect.

The branches are combined through cross-convolutional layers that upscale or downscale each resolution path and concatenate all branches together over their feature dimension.

4.7. Datasets

Datasets are essential for developing a working machine learning model. They must contain a sufficient number of correctly labelled images from a variety of contexts. The model will only be able to solve similar cases to those contained in the dataset used for training. A problem that we are facing and that occurs frequently in research dealing with machine learning is that the trained model is unable to generalise—to work with real-world data that are too different from the data used for training. This can happen with real-world imagery, as the camera and lighting conditions of the images are different from those of the training datasets. This problem is usually solved by changing the model architecture and training with more diverse data. The manual work needed to create a dataset is substantial; hence, it is often the case that the datasets already created are insufficiently diverse. In our case, we had a lack of images from countries in the developing world, as only large institutions can afford to create large datasets, especially in the semantic segmentation task, which requires the most manual work. This is a problem, as the light conditions and urbanism in the cities for which data are lacking are different from the cities where models are usually trained (mainly in Germany). Data augmentation addresses the issue of a lack of generalisation and consists of altering the input images of a dataset, thus creating new data from the existing dataset [37]. We used horizontal flipping—low image rotations as augmentation options.

5. Artificial Intelligence Applied to Building Height Estimation

In this section, neural network algorithms are applied to obtain building heights. The neural network developed for this purpose and the post processing algorithm are presented.

5.1. Facade Detection: Semantic Segmentation Model

The first step in any tool for data collection from images is to detect the spatial and categorical information of a certain object in the input images. In our case, the input images are street view images from the point of view of a car driver, from which we extract spatial information of buildings, cars, pavement and other objects. This is conducted with a semantic segmentation machine learning model that assigns a class to each pixel of the input image. We developed a fully convolutional neural network based on the HRNet encoder [25], which maintains a high-resolution representation of the input image during the whole model representation. The output labelled image has half the resolution of the input to enable faster runtime. The HRNet encoder Figure 4 splits the input image into two branches, one with half the original resolution and the other one with one sixteenth of the original resolution. The second branch acts as the bottleneck, while the first one acts as a high-resolution representation of the image. The downscaling process of the input image takes place gradually through the encoder, halving the resolution in the second branch in each encoder layer. Both branches are correlated together through cross-convolutions (Figure 5).

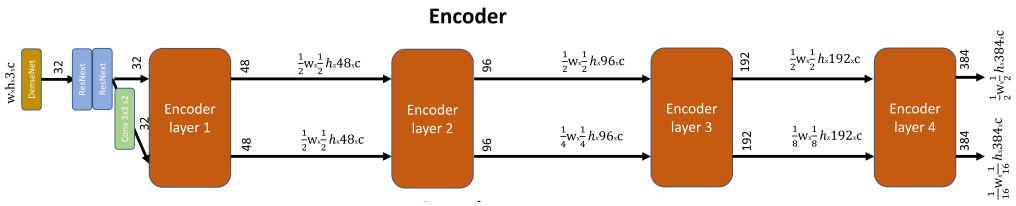


Figure 4. Model encoder.

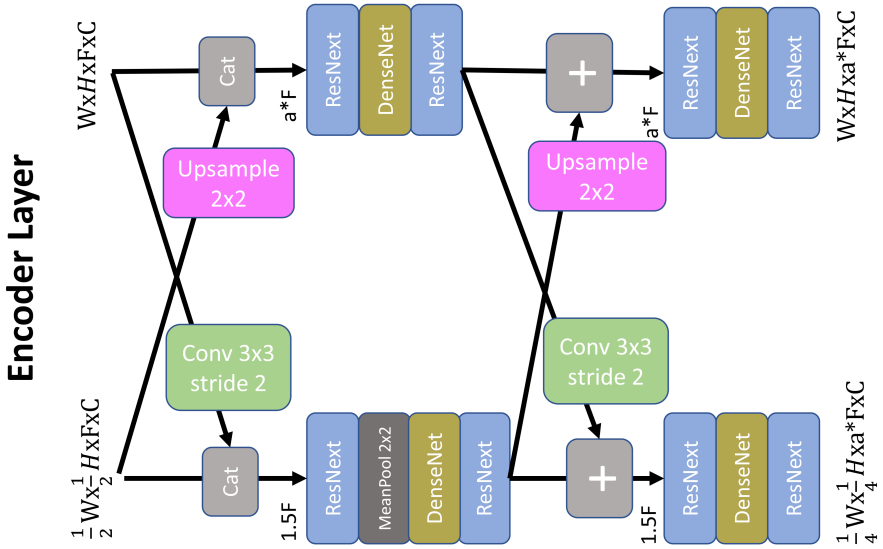


Figure 5. Encoder layer. The first cross-convolution is not applied, and mean pool is applied on branch 1 in the encoder layer 1. × denotes array dimension and * denotes multiplication.

In the encoder, ResNext (Figure 6) convolutions [23] are used and, subsequently, an atrous spatial pyramid (Figure 7) convolution [29] is applied. ResNext is a recently proposed model that uses depth separable convolutions [28] with an inverse bottleneck and achieves state-of-the-art results using less training parameters. DenseNet layers are layers that increase the feature dimension by concatenating outputs from convolutions to the input and by repeating this process multiple times [38].

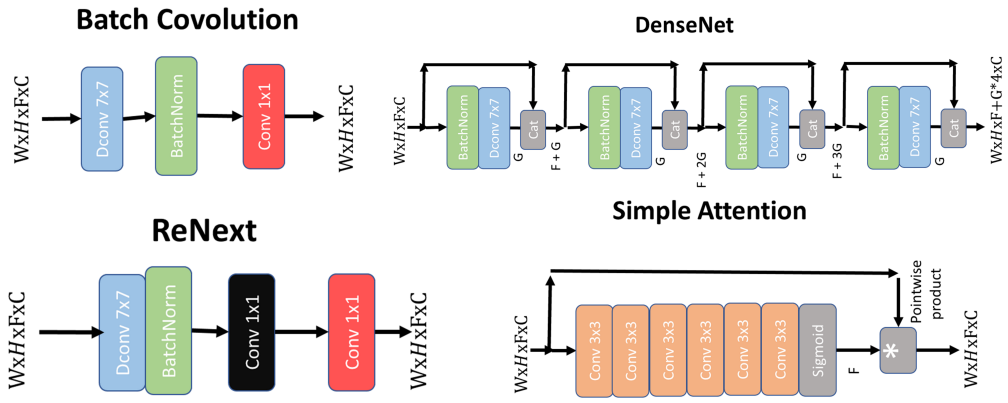


Figure 6. Basic layers of the model. × denotes array dimension and * denotes multiplication.

In the decoder (Figure 8), simple depthwise convolutions without a bottleneck are used. After applying an attention mechanism (Figure 6), skip connections coming from layers 5, 4 and 3 are fused with the decoder layers 1, 2 and 3, respectively, concatenating over the feature dimension of each branch. In every decoder layer, the second branch is upsampled until the size of the first branch is achieved in the last decoder layer, just before both branches are fused together using an attention mechanism. The attention mechanism assigns an attention, which establishes how important each pixel is and provides a criterion for fusing two images together.

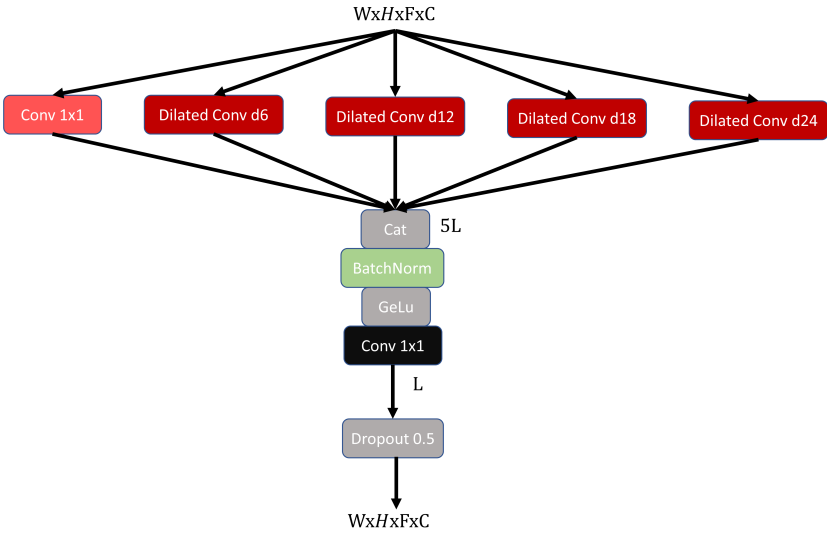


Figure 7. Atrous spatial pyramid layer.

The model is programmed in the Julia programming language [39] using Zygote [40] and Flux [41] libraries and cross-entropy [42] as the loss function for classification tasks. Optimisation is done with supervised learning using the ADAM optimizer [34]. Training is performed with a A100 Nvidia GPU on a Magerit supercomputer at Cesvima UPM. For model parameter initialization, a glorot uniform distribution [43] is used.

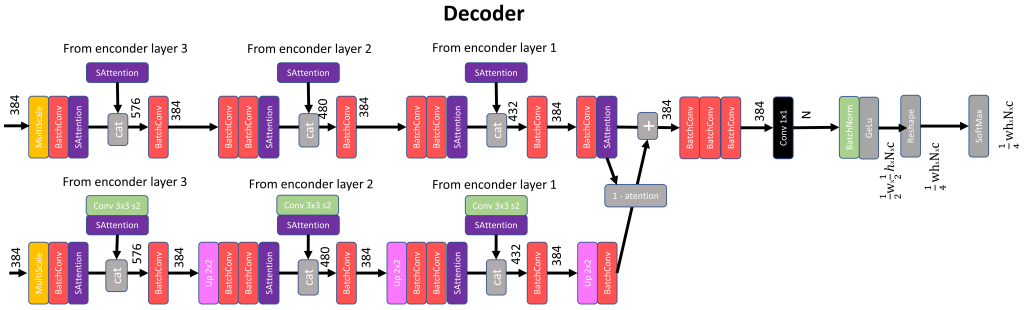


Figure 8. Model decoder. × denotes array dimension.

5.2. Extracting Spatial Information

This model provides 2D relative spatial information from the image. However, we would like to obtain 3D real-world spatial information, as we could then measure building height directly. This requires a point cloud to be generated from the input image. There are machine learning models that provide pixelwise depth estimation from single-view images. With the depthmap of the image and camera data such as focal length, we would be able to assign a distance and direction to every pixel of the image and thus create the point cloud. Unfortunately, the precision of the single-view depth estimation models [44] developed to date is insufficient for this purpose. In the KITTI benchmark, the best models present an error of around 8%. This means an error of 8 m in the depth estimation of a building at a distance of 100 m from the camera. We carried out tests with depth models trained on KITTI, but the point cloud we obtained was not accurate. As it was not possible to generate a point cloud from single view-images, we follow two alternative approaches. Firstly, a point cloud can be generated through photogrammetry from multiple images or a video [45]. This implies a high computational cost and takes away versatility from our method, as it is much more difficult to obtain multiple images with the necessary characteristics from a video. The second option is to develop a method that does not require a point cloud, which is the approach we have found to be the most feasible.

5.3. Height Comparison Method

In this section, building height is estimated using the 2D information from the image. On a given street, there are certain objects with known height, such as cars, people or the windows and doors of buildings. By comparing the size of the building with the size of a reference class such as those mentioned, we are able to obtain the building height. Windows and doors would be our best option as a reference class, but unfortunately, these are not labels contained in the training dataset. Hence, in order to use these objects as a reference class, we would need to create our own semantic segmentation dataset, which is a task beyond the capabilities of this research. We therefore decided to use cars and people as the reference classes, although cars only provide a good reference if they are parked near the building that is being measured.

Local building height is obtained as the quotient on each column of pixels belonging to a building in the segmented image and pixels belonging to one of the reference classes multiplied by the height of the latter. The result of this operation is a vector with a length equal to the horizontal dimension of the image.

This method is only accurate if the reference object is next to the building. If this is not the case, then the reference object will appear larger on the image, and the measurement will not be valid. This may occur if street geometry is not standard, such as at an intersection, or where there is a moving car or a person crossing the street. Reference objects with non-standard sizes can also lead to inadequate measurements. As the output is the average height prior to calculating the mean of the local building height vector, invalid data can be statistically discarded. This is done by calculating the mean and standard deviation of the local height vector and by cleaning the vector of data that deviate excessively from the mean value. Invalid measurements tend to be lower than the true height, as inappropriate reference objects appear larger in almost every case. Hence, positive deviation over the mean of the height vector should be less tolerated than negative deviation. Should multiple images and reference classes be used, it is possible to perform more statistical computations to further clean up the height vectors.

5.4. Model Training Datasets

In semantic segmentation, most datasets with street-view imagery are created for tasks related to self-driving cars. These datasets usually contain a building class, a sky class and a road class, which are all useful to us. Unfortunately, other useful classes, such as windows and doors of buildings, building materials, etc., are not available in most datasets, as these classes are not relevant in the case of self-driving vehicles. To perform an initial, broad, low precision training, we used the Vistas [27], the GTA5 [46] and the Cityscapes coarse [26] datasets. The Vistas and Cityscapes coarse sets are coarsely annotated, which means that labels are only assigned to a small percentage of the pixels of an image. Vistas has around 25,000 images from countries all over the world, which makes it the only dataset with images from Central America. GTA5 is a dataset with around 25,000 images taken from the GTA5 videogame. The labels were automatically generated from game images from the city of Los Angeles, which appear in this videogame. The images, therefore, have different lighting conditions and even night scenes. Finally, Cityscapes is a dataset with images from German cities which has around 18,000 coarsely labelled and 5000 fine-labelled images. The coarse dataset was used for a broad, initial training, and the fine set was used as a last refinement. Through this training scheme, the model was trained with more diverse data, and to a certain degree, we were able to overcome the problems associated with real-world data.

The model was initially trained using the 18,000 images from the Cityscapes coarse dataset followed by 25,000 images from Vistas (adapting the labels of the latter to the Cityscapes labelling scheme), 25,000 images from GTA5 and 5000 images from the Cityscapes fine dataset. Data augmentation was applied through 10-degree rotations and horizontal flipping.

6. Case Study: Extraction of Building Height in Cityscapes

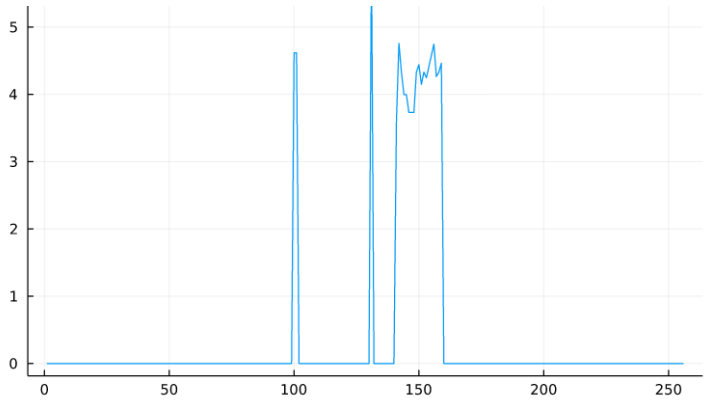
6.1. Results

The semantic segmentation model achieved an 86.13% IoU metric (intersection-over-union) on the Cityscapes fine val set [26]. To illustrate the results, a case study with a Cityscapes image is presented and discussed. An image from the Cityscapes val set was run through the model as a (width $\times$ height $\times$ 3) matrix with values ranging from 0 to 1. The result is a (classes $\times$ width $*$ height $\times$ 1) matrix ranging from 0 to 1. The values of the vector represent how likely the model considers a pixel to belong to a given class. In this case, there are three classes: sky, building and car. The vector is reshaped to the size (width $\times$ height $\times$ 1), there now being an integer representing the class with the highest probability for each pixel. The height vector is then obtained (Figure 9), comparing the pixels belonging to the building and the car class for each row, after discarding the columns with no pixels belonging to the sky class. In this example, having cleaned the height vector, the building height ranges between (3.8) and (4.7) with a mean of (4.34). This means that the buildings are around (4.34) times higher than a car in this street. Now, the predicted average building height for this street would be (4.34 $*$ car height). The car height that must be used would be calibrated through measurements of the city with buildings of known height. This method requires uniform street alignment and sufficient visible sky in the image. As street view images are cheap and simple to obtain, erroneous measurements from images with moving cars or a non-uniform street could be statistically discarded. This method could provide average building heights on a street with a maximum error of 3 m (around 1 floor) and a relative error of 9% in a given city with uniform streets.

(**a**)

(**b**)

Figure 9. *Cont.*



(**a**)

Figure 9. Example of the method with an image from the Cityscapes test set. (**a**) Input image. (**b**) Model output segmentation. (**c**) Building height vector considering car height 1.

6.2. Discussion

The image segmentation results are promising, achieving a high performance in the Citiscapes fine val set [26]. This opens the way to further automatisation of the detection of buildings and their parts and features.

The results obtained in the case study are satisfactory for the estimation of the building exposure attribute "height", which is defined as the height above ground in terms of the number of storeys (e.g., a building being three storeys high). The height class is usually defined as a range of storeys or floors, such as 1–3, 4–7 and >8 [4,47]. Thus, the determination of seismic exposure classes can be performed with one-floor-precision building height estimates. The proposed method fulfils this requirement with a maximum error of around one floor.

These results are comparable with those of other methods, such as those using satellite imagery with absolute errors of around 2–5 m, which usually require SAR data [13] or multi-view imagery [14,15] and are therefore more complicated.

Street view methods that use machine learning can achieve a lower relative error of 5% but require more data such as building footprints [17,18] or camera heights [19].

The proposed method works without the need for extra parameters, allowing for the use of images from the Internet or images captured by mobile phones for other purposes as long as the problem associated with an insufficiently diverse dataset can be overcome.

7. Conclusions

In light of this research, the notable potential of artificial intelligence in terms of automating the extraction of relevant parameters of buildings can be confirmed, although there is still a long way to go in the field of computer vision.

There is a lack of high-quality datasets with urban street imagery for tasks such as image segmentation, especially in the case of imagery from developing countries. Moreover, the datasets currently available have been created mainly for self-driving car technology, not for uses related to the construction sector.

It is possible to estimate the average height of the buildings in pilot images with errors of just a few metres, under conditions that are versatile and which allow for easy matching without the need for point clouds or any kind of image depth estimation. The semantic segmentation model achieves noteworthy results and is more versatile as well as faster than other models, successfully addressing the issues of lack of global information extraction and image detail loss. However, the model still needs improvements in order to function adequately for non-European cities due to the lack of suitable datasets.

Further research could address the lack of training data for the segmentation model through domain adaptation, with a model that converts street images to the style of a German city. However, it is crucial that datasets are created for other regions of the world.

A promising line of research includes the development of building height estimation classifier models. Classifier machine learning models are among the most studied fields in artificial intelligence. These models will assign the number of floors to each building as a categorical class. This information is essential for the assessment of seismic exposure and for the calculation of building vulnerability, an urgent issue in many regions of the world.

Author Contributions: Conceptualisation, M.U.-P., R.M.-M. and M.M.-S.; methodology, M.U.-P. and R.M.-M.; software, M.U.-P.; validation, M.U.-P. and M.M.-S.; writing—original draft preparation, M.U.-P.; writing—review and editing, B.G.-R., R.M.-M. and M.M.-S.; visualisation, M.U.-P.; supervision, R.M.-M. and B.G.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received public funds from a grant for the completion of industrial doctorates (IND2020/TIC-17528) funded by Comunidad de Madrid.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: Not applicable.

Acknowledgments: The authors gratefully acknowledge the KUK_AHPAN project (RTI2018-094827-B-C21/C22) and the Universidad Politécnica de Madrid (www.upm.es, accessed on 10 February 2023) for providing computing resources on Magerit Supercomputer.

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.

Abbreviations

The following abbreviations are used in this manuscript:

GEM	Global Earthquake Model
INSPIRE	European Commission: Infrastructure for Spatial Information in Europe
ESA	European Space Agency
LIDAR	Light Detection and Ranging
SAR	Synthetic Aperture Radar
ReLu	Rectified Linear Activation Unit
GeLu	Gaussian Error Linear Unit
HRNet	High Resolution Network
GTA5	Grand Theft Auto 5 videogame
KITTI	Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago
GPU	Graphics Processing Unit
UPM	Universidad Politécnica de Madrid

References

1. European Construction Sector Observatory. *Digitalisation in the Construction Sector*; Technical Report; Publications Office of the European Union: Luxembourg, 2021.
2. Baldini, G.; Barboni, M.; Bono, F.; Delipetrev, B.; Duch Brown, N.; Fernandez Macias, E.; Gkoumas, K.; Joossens, E.; Kalpaka, A.; Nepelski, D.; et al. *Digital Transformation in Transport, Construction, Energy, Government and Public Administration*; Publications Office of the European Union: Luxembourg, 2019.
3. Baggio, C.; Bernardini, A.; Colozza, R.; Pinto, A.V.; Taucer, F. *Field Manual for Post-Earthquake Damage and Safety Assessment and Short Term Countermeasures (AeDES) Translation from Italian: Maria ROTA and Agostino GORETTI*; Technical Report.Publications European Commission JRC: Luxembourg, 2007.
4. Brzev, S.; Scawthorn, C.; Charleson, A.; Allen, L.; Greene, M.; Jaiswal, K.; Silva, V. *GEM Global Earthquake Model GEM Building Taxonomy Version 2.0 Exposure Modelling*; Technical Report; GEM Foundation: Pavia, Italy, 2013.
5. Ruggieri, S.; Cardellicchio, A.; Leggieri, V.; Uva, G. Machine-learning based vulnerability analysis of existing buildings. *Autom. Constr.* **2021**, *132*, 103936. [CrossRef]
6. Cardellicchio, A.; Ruggieri, S.; Leggieri, V.; Uva, G. View VULMA: Data Set for Training a Machine-Learning Tool for a Fast Vulnerability Analysis of Existing Buildings. *Data* **2022**, *7*, 4. [CrossRef]

7. Esquivel-Salas, L.C.; Schmidt-Díaz, V.; Pittore, M.; Hidalgo-Leiva, D.; Haas, M.; Moya-Fernández, A. Remote structural characterization of thousands of buildings from San Jose, Costa Rica. *Front. Built Environ.* **2022**, *8*, 947329. [CrossRef]
8. Rodríguez-Saiz, J.; Marchamalo, M.; Esquivel, L.; Rejas-Ayuga, J.; García-Lanchares, C.; González-Rodrigo, B.; Benito, B. Exposición sísmica de los edificios por métodos geoespaciales. In Proceedings of the XIV Congreso Geológico de América Central & VII Congreso Geológico Nacional, San José, Costa Rica, 28 June–1 July 2022.
9. Benito, M.B.; Lindholm, C.; Camacho, E.; Climent, A.; Marroquín, G.; Molina, E.; Rojas, W.; Escobar, J.J.; Talavera, E.; Alvarado, G.E.; et al. A new evaluation of seismic hazard for the Central America Region. *Bull. Seismol. Soc. Am.* **2012**, *102*, 504–523. [CrossRef]
10. Cardellicchio, A.; Ruggieri, S.; Leggieri, V.; Uva, G. A machine learning framework to estimate a simple seismic vulnerability index from a photograph: The VULMA project. *Procedia Struct. Integr.* **2023**, *44*, 1956–1963. [CrossRef]
11. *INSPIRE Infrastructure for Spatial Information in Europe D2.8.III.2 Data Specification on Buildings-Technical Guidelines Title D2.8.III.2 INSPIRE Data Specification on Buildings-Technical Guidelines*; Technical Report; European Commission Joint Research Centre: Luxembourg, 2013.
12. Drusch, M.; Del Bello, U.; Carlier, S.; Colin, O.; Fernandez, V.; Gascon, F.; Hoersch, B.; Isola, C.; Laberinti, P.; Martimort, P.; et al. Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sens. Environ.* **2012**, *120*, 25–36. [CrossRef]
13. Frantz, D.; Schug, F.; Okujeni, A.; Navacchi, C.; Wagner, W.; van der Linden, S.; Hostert, P. National-scale mapping of building height using Sentinel-1 and Sentinel-2 time series. *Remote Sens. Environ.* **2021**, *252*, 112128. [CrossRef]
14. Cao, Y.; Huang, X. A deep learning method for building height estimation using high-resolution multi-view imagery over urban areas: A case study of 42 Chinese cities. *Remote Sens. Environ.* **2021**, *264*, 112590. [CrossRef]
15. Xiao, J.; Gerke, M.; Vosselman, G. Building extraction from oblique airborne imagery based on robust façade detection. *ISPRS J. Photogramm. Remote Sens.* **2012**, *68*, 56–68. [CrossRef]
16. Bonczak, B.; Kontokosta, C.E. Large-scale parameterization of 3D building morphology in complex urban landscapes using aerial LiDAR and city administrative data. *Comput. Environ. Urban Syst.* **2019**, *73*, 126–142. [CrossRef]
17. Ala, B. *An Open-Source System for Building-Height Estimation Using Street-View Images, Deep Learning, and Building Footprints Reports on Special Business Projects*; Technical Report. Statistics Canada: Ottawa, ON, Canada, 2020.
18. Zhao, Y.; Qi, J.; Zhang, R. CBHE: Corner-based building height estimation for complex street scene images. In Proceedings of the Web Conference 2019—World Wide Web Conference (WWW 2019), San Francisco, CA, USA, 13–17 May 2019; Association for Computing Machinery, Inc.: New York, NY, USA, 2019, pp. 2436–2447. [CrossRef]
19. Yan, Y.; Huang, B. Estimation of building height using a single street view image via deep neural networks. *ISPRS J. Photogramm. Remote Sens.* **2022**, *192*, 83–98. [CrossRef]
20. Díaz, E.; Arguello, H. An algorithm to estimate building heights from Google street-view imagery using single view metrology across a representational state transfer system. In Proceedings of the Dimensional Optical Metrology and Inspection for Practical Applications V SPIE, Baltimore, MD, USA, 19 May 2016; Volume 9868, p. 98680A. [CrossRef]
21. 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 16 × 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations ICLR, Online, 5 May 2021. Available online: https://iclr.cc/virtual/2021/session/4343 (accessed on 20 March 2023).
22. Strudel, R.; Garcia, R.; Laptev, I.; Schmid, C. Segmenter: Transformer for Semantic Segmentation. *arXiv* **2021**, arXiv:2105.05633.
23. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated Residual Transformations for Deep Neural Networks. *arXiv* **2016**, arXiv:1611.05431.
24. Hitawala, S. Evaluating ResNeXt Model Architecture for Image Classification. *arXiv* **2018**, arXiv:1805.08700v1.
25. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep High-Resolution Representation Learning for Visual Recognition. *arXiv* **2019**, arXiv:1908.07919.
26. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. *arXiv* **2016**, arXiv:1604.01685.
27. Neuhold, G.; Ollmann, T.; Rotabuì, S.; Kontschieder, P.; Research, M. *The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes*; Technical Report. IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5000–5009. [CrossRef]
28. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. *arXiv* **2016**, arXiv:1610.02357.
29. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *arXiv* **2016**, arXiv:1606.00915.
30. O'Shea, K.; Nash, R. An Introduction to Convolutional Neural Networks. *arXiv* **2015**, arXiv:1511.08458.
31. Janocha, K.; Czarnecki, W.M. On Loss Functions for Deep Neural Networks in Classification. *arXiv* **2017**, arXiv:1702.05659.
32. Bridle, J.S. *Training Stochastic Model Recognition Algorithms as Networks Can Lead to Maximum Mutual Information Estimation of Parameters*; In Proceedings of the 2nd International Conference on Neural Information Processing Systems, Cambridge, MA, USA, 1 January 1989; Volume 2, pp. 211–217.
33. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
34. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.

35. Lorente, O.; Riera, I.; Rana, A. Image Classification with Classic and Deep Learning Techniques. *arXiv* **2021**, arXiv:2105.04895.
36. Liu, X.; Deng, Z.; Yang, Y. Recent progress in semantic image segmentation. *Artif. Intell. Rev.* **2019**, *52*, 1089–1106. [CrossRef]
37. 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]
38. Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. *arXiv* **2016**, arXiv:1608.06993.
39. Bezanson, J.; Edelman, A.; Karpinski, S.; Shah, V.B. Julia: A Fresh Approach to Numerical Computing. *arXiv* **2014**, arXiv:1411.1607.
40. Innes, M.; Edelman, A.; Fischer, K.; Rackauckas, C.; Saba, E.; Shah, V.B.; Tebbutt, W. A Differentiable Programming System to Bridge Machine Learning and Scientific Computing. *arXiv* **2019**, arXiv:1907.07587.
41. Innes, M.; Saba, E.; Fischer, K.; Gandhi, D.; Rudilosso, M.C.; Joy, N.M.; Karmali, T.; Pal, A.; Shah, V. Fashionable Modelling with Flux. *arXiv* **2018**, arXiv:1811.01457.
42. Zhang, Z.; Sabuncu, M.R. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. *arXiv* **2018**, arXiv:1805.07836.
43. Glorot, X.; Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Montreal, QC, Canada, 13–15 May 2010; pp. 249–256.
44. Li, Z.; Wang, X.; Liu, X.; Jiang, J. *BinsFormer: Revisiting Adaptive Bins for Monocular Depth Estimation*; Technical Report. *arXiv* **2022**, arXiv:2204.00987.
45. Müller, T.; Evans, A.; Schied, C.; Keller, A. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.* **2022**, *41*, 102. [CrossRef]
46. Richter, S.R.; Vineet, V.; Roth, S.; Koltun, V. Playing for Data: Ground Truth from Computer Games. *arXiv* **2016**, arXiv:1608.02192.
47. FEMA. *Hazus -MH 2.1. Technical Manual*; Publications FEMA: Washington, DC, USA, 2001; pp. 1–139.