Advanced Engineering Informatics 50 (2021) 101380

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

An enhanced 3D model and generative adversarial network for automated generation of horizontal building mask images and cloudless aerial photographs

Kazunosuke Ikeno, Tomohiro Fukuda *, Nobuyoshi Yabuki

Division of Sustainable Energy and Environmental Engineering, Graduate School of Engineering, Osaka University 2-1, Yamadaoka, Suita, Osaka 565-0871, Japan

ARTICLE INFO

Keywords:
Deep learning
Generative adversarial network
Semantic segmentation
Mask image
Training data
Urban planning and design

ABSTRACT

Information extracted from aerial photographs is widely used in the fields of urban planning and design. An effective method for detecting buildings in aerial photographs is to use deep learning to understand the current state of a target region. However, the building mask images used to train the deep learning model must be manually generated in many cases. To overcome this challenge, a method has been proposed for automatically generating mask images by using textured three-dimensional (3D) virtual models with aerial photographs. Some aerial photographs include clouds, which degrade image quality. These clouds can be removed by using a generative adversarial network (GAN), which leads to improvements in training quality. Therefore, the objective of this research was to propose a method for automatically generating building mask images by using 3D virtual models with textured aerial photographs. In this study, using GAN to remove clouds in aerial photographs improved training quality. A model trained on datasets generated by the proposed method was able to detect buildings in aerial photographs with IoU = 0.651.

1. Introduction

Information extracted from aerial photographs is widely used in urban planning and design. For example, aerial photographs can be used for land surveys, building maintenance, and forest management [1,29,42]. As the use of unmanned aerial vehicle (UAV) technology has become more widespread, aerial photographs have become easier to take. Information that needs to be gathered in real time, such as damage to buildings during a disaster, can be grasped using aerial photographs taken by UAVs. In the past, obtaining information from a photograph required visual assessment by a human expert, which could take considerable time. However, it is difficult to extract information from a large number of photographs in this manner. Therefore, using artificial intelligence (AI) to recognize objects in images has been proposed as an efficient method for obtaining information from large numbers of photographs [26,35].

Using AI to obtain information from a large number of aerial photographs in a short time involves deep learning. In recent years, object detection and segmentation methods based on deep learning have been proposed, and these methods have made it possible to automatically detect objects in images in a short time. A deep learning segmentation method has been proposed for buildings in aerial photographs that enables the user to quickly grasp the state of the target area. Given that the building detection accuracy of these segmentation methods is strongly affected by the amount of data in the training dataset as well as the number of features used to train the deep learning model, it is necessary to train the deep learning model appropriately for each target area. However, in most cases, mask images of buildings used to train deep learning models are generated by manual operation, which requires considerable time and effort to generate a large number of mask images. Therefore, a method for efficiently generating building mask images is needed for deep learning.

Image editing software such as Adobe Photoshop and GIMP is often used to generate mask images of objects in an image because such programs can automatically clip objects. However, this method is not effective for buildings because they can be difficult to distinguish from other objects, such as roads and vehicles. Although a geographic information system can be used to generate mask images of urban structures,

Abbreviations: UAV, unmanned aerial vehicle; AI, artificial intelligence; PC, personal computer; VR, virtual reality; GAN, generative adversarial network; 3D, three-dimensional; GPU, graphics processing unit; IoU, intersection over union.

* Corresponding author.
E-mail addresses: fukuda@see.eng.osaka-u.ac.jp (T. Fukuda), yabuki@see.eng.osaka-u.ac.jp (N. Yabuki).

https://doi.org/10.1016/j.aei.2021.101380
Received 2 April 2021; Received in revised form 11 July 2021; Accepted 3 August 2021
Available online 12 August 2021

it is difficult to represent the generated masks in three dimensions, and the objects for which mask images can be generated are limited. To solve these problems, a method using virtual reality (VR) models to automatically generate building mask images for deep learning has been proposed [13]. However, commonly used VR models do not adequately represent real objects. Therefore, it is difficult to obtain accurate detection results for real objects when images obtained from these models are used to train deep learning models. High-precision rendering techniques are being developed to improve the representation of VR, but they require high-performance personal computers (PCs) and other equipment, making them impractical for general use [3]. Texture mapping is a method of defining surface texture, or color information, on a VR model. In texture mapping, it is possible to use photographs as textures, which can easily improve the representation of the VR model.

One of the challenges of using aerial photographs is the presence of clouds. When aerial photographs containing clouds are used as a supervisory dataset, there is a possibility that the learning accuracy will be degraded. Therefore, it is necessary to remove these clouds before using aerial photographs as texture. However, it takes a long time for a human to manually remove clouds from aerial photographs, and the resulting image has a less available area for use in a training dataset. In contrast, cloud removal can be achieved in a relatively short time by generating images using a generative adversarial network (GAN), which is an image processing technique based on deep learning [15].

It is also necessary to improve the inadequate representation of three-dimensional (3D) models, which is a problem in the automatic generation of training datasets using 3D models. By using a 3D model with aerial photographs as textures, it is possible to automatically generate a training dataset that easily solves this representation problem. In addition, the degradation of image quality in aerial photographs used for texture mapping can be avoided by using GAN to generate images. Therefore, the objective of this study was to propose a method to automatically generate horizontal building mask images by using three-dimensional (3D) models with textured aerial photographs for deep learning. Specifically, we aimed to improve the representation of VR models by using textured aerial photographs on 3D models. Some aerial photographs include clouds, which degrade image quality. In this study, we defined an enhanced 3D model as a 3D model whose appearance has been improved by applying aerial photographs as textures. The clouds on these aerial photographs were removed using GAN to improve training quality. The proposed method automatically generates mask images by using enhanced 3D models and GAN. An earlier version of this paper was presented at CAADRIA 2021 [20]. Significant developments in the research since then are described in the present work. The developed method makes it possible to detect buildings with high accuracy, even in urban areas for which no training data are provided. Accordingly, the technique can be used for land use surveys and to check for damage during disasters.

2. Literature review

In this section, we review previous research on object detection using deep learning, training datasets, and image completion techniques.

2.1. Object detection and segmentation by deep learning

In deep learning, features are automatically calculated from the provided training data, and object detection can be performed based on the calculated features. Object detection and semantic segmentation are deep learning techniques for identifying objects in an image. Object detection is a technique that detects the position and category of a given object in the image using a rectangle. AlexNet was proposed as a method for detecting the target region in an image as a rectangle using a convolutional neural network (CNN) [26]. The method was successful in the ImageNet Large Scale Visual Recognition Challenge, a large-scale image recognition competition [8,9]. Since then, several CNN-based object-detection algorithms have been developed, including R-CNN (Regions with CNN), YOLO (You Only Look Once), and SSD (Single Shot Multibox Detector) [14,40,34]. Semantic segmentation is a technique that classifies each pixel into several categories and segments objects in an image according to their silhouettes [35]. The technique was first proposed as a fully convolutional network (FCN), which replaces all the join layers in a CNN with upsampled convolutional layers. Since then, most semantic segmentation models have been based on FCNs.

2.1.1. Use in urban planning and construction

Object recognition methods based on deep learning have been widely used in the fields of urban planning and construction. Such methods have been proposed for streamlining the management of a large number of photos taken at construction sites and disaster-affected areas, to detect various construction equipment, workers, and signs in the photos and automatically sort them into folders [45]. A similar approach has been proposed for monitoring workmanship by using deep learning to check the construction status at a construction site from images captured by a camera [22]. Furthermore, in the field of structural management, object recognition has been proposed for efficiently detecting damage in tunnels [29], and in landscape studies, for simulating geometrically consistent and more realistic environmental design through dynamic occlusion processing [25]. Methods using deep learning models to detect waterway leakage from satellite images as well as land use and disaster damage from aerial photographs have also been proposed [5,1,39].

2.1.2. Building detection

A deep CNN-based method for automatically detecting suburban buildings from high-resolution Google Earth images has also been proposed as a building detection method that uses deep learning [47]. In addition, a fused fully convolutional network model has been proposed to perform building segmentation [4]. Some research is being conducted to improve the accuracy of Mask-R-CNN for detecting building footprint boundaries. Furthermore, a method combining Mask-R-CNN with building boundary regularization has been presented [49], and a method has been proposed for detecting different scales of buildings and segmenting buildings to have accurately segmented edges [50].

2.2. Training data for deep learning

Open source datasets are often used to train deep learning models. The PASCAL VOC dataset and the Microsoft COCO dataset have been proposed as training datasets for general objects [12,32]. A dataset that can be used in urban planning is the CityScapes dataset of images from 50 cities in Germany consisting of 5000 images with semantic segmentation information added at the detailed pixel level and another 20,000 images with semantic segmentation information added at the coarse pixel level [6]. Another dataset is the SpaceNet dataset, which contains aerial photos of Rio de Janeiro, Las Vegas, Paris, Shanghai, Khartoum, and Atlanta as well as their corresponding building locations [10]. However, most of the masked images in these datasets were generated manually, which required considerable time and effort. Accordingly, it is difficult to manually generate datasets for additional cities.

To overcome this challenge, a method has been proposed to automatically generate mask images of buildings by using VR 3D models for deep learning [13]. By using a 3D virtual model, we can reduce the time needed to create mask images compared to manual operation. We don't need the hassle of creating many building masks by hand in detail. Given that normal virtual models do not have the realism of a photograph, it is difficult to obtain highly accurate detection results in the real world even when the image is used for deep learning training. High-precision rendering methods have been developed but they are generally difficult to use because many computers do not have high enough specifications. Using textured 3D virtual models with photographs can overcome this challenge.

2.3. Image complementation

Image completion techniques have been extensively studied in the field of computer vision. These techniques include methods using luminance values and textures in the same image as well as methods based on deep learning.

2.3.1. Methods using luminance values and textures in the same image

As an image completion method using luminance values, a method for accurately recovering luminance values of still images by using optical flow has been proposed [37], and it has shown improvements in accuracy, especially at the boundaries in image completion. A method for image completion by histogramming local luminance features in an image and learning them statistically has also been proposed [28]. Such completion methods using luminance values are effective for completing thin lines in an image, but achieving clear image completion is difficult when the target area is large.

In contrast, image completion using textures in the same image is effective for large areas. An image completion method that uses pattern similarity and considers the brightness variation and locality of the texture has been proposed, as has a method that uses the pattern similarity of the texture for large defects and a one-dimensional pattern for fine linear defects [24,7]. These methods are effective for images with few changes in pattern structure, such as man-made objects. However, for images with many pattern changes (e.g., outdoor landscapes) and low texture similarity in the same image, completion performance is greatly reduced. To address this problem, a method has been proposed that matches the textures in an image and uses the statistical distribution of the relative positions of similar patches as a similarity measure to provide highly accurate completion even for images with highly variable pattern structures, including outdoor landscapes [17]. In recent years, image processing software such as Adobe Photoshop, GIMP, and the open-source image processing framework G'MIC have also provided image completion methods using textures in images.

2.3.2. Methods using deep learning

GAN is a deep learning architecture that has been proposed for image processing [15]. A GAN is a generative model that uses deep learning. GANs enable the generation of non-existent data and the transformation of existing data according to their features. GLCIC (globally and locally consistent image completion) is an image completion method that uses convolutional neural networks and considers the global and local consistency of a scene [19]. To form the completion network, whose layers consist of convolutional neural networks, these methods perform image completion while considering scene consistency and construct a global and local discrimination network to discriminate between the real image and the completed image. The global discriminative network evaluates whether the whole image is natural, whereas the local discriminative network evaluates the image based on the more detailed consistency around the completion region. By training the complete network to "trick" both of these two discriminative networks, we can output a completed image that is consistent across the scene and locally natural. In addition, a method based on deep learning has been proposed that removes rain and snow from photographs containing rain and snow and complements the background image [46,11]. In the field of remote sensing, an adversarial generation network has been used to remove clouds from satellite images [30]. The GAN can be used to remove clouds from aerial photographs to improve the quality of training data for deep learning applications. By adjusting the data for training, it is no longer necessary to pre-select images that include or do not include clouds. Therefore, all images can be used as input images regardless of whether they contain clouds or not, reducing the burden on the user.

3. Method for automatic generation of training datasets

In this section, we present our proposed method for the automatic generation of training datasets.

3.1. Overview of the proposed method

Our proposed method automatically generates building mask images and aerial photographs. The generated mask images are used to train the deep learning model for semantic segmentation. The proposed method loads 3D models that include terrain and building objects, classifies by building class and others class, switches between a model with all objects and one with only buildings, and finally generates two upper-view images of the models from multiple viewpoints. The game engine used in this method must be able to import enhanced 3D models, classify objects into the two classes, and output images while switching between display and non-display. Aerial photographs that include clouds are regenerated as images without clouds by using a GAN that can change from an image with one feature to another. At this stage, it is necessary to use manual operations to identify aerial photographs that contain clouds. This method can generate multiple sets of mask images and aerial photographs without clouds from an enhanced 3D model. A flowchart and conceptual diagram of the method are shown in Figs. 1 and 2, respectively.

3.2. Creation of an enhanced 3D model of the target city

In this method, it is necessary to create an enhanced 3D model of the target location with aerial photographs applied as textures in advance. In addition, the 3D model must include the location information of the buildings corresponding to the aerial photographs to be pasted as textures, and the model must be able to classify these objects when it is loaded into the game engine. Therefore, it is necessary to use software and web services to create 3D models that can be classified into building classes and other classes as attribute information for each object. The texture mapping of aerial photographs can be done either at the time of 3D model creation or by defining it on the game engine. However, when pasting aerial photographs as textures on a 3D model in a game engine, it is necessary to adjust the scale and position.

3.3. Generation of mask images on a game engine

To automatically obtain aerial photographs and mask images from the generated enhanced 3D models, we use a game engine. The game engine used in this method must be able to read the enhanced 3D models, classify objects into building classes and other classes, and output images while switching between display and non-display.

3.4. Cloud removal by GAN

Image transformation by GAN is used to remove clouds in aerial photographs used for texture mapping. We use a GAN based on pix2pix and Cycle GAN, which can change features between images and can change from one image with one feature to another image with another feature [21,51]. Here, we summarize the various types of GAN. GAN was proposed in 2014 as an architecture enabling two networks to learn by competing with each other. The generator network generates data and the discriminator network inputs random noise corresponding to the seeds of the features of the generated data and maps this noise closer to the target data. The discriminator network compares false data generated by the generator network with ground truth data to determine their authenticity. By alternately training these two networks, the generator network can generate images that are closer to the ground truth data.

In addition, there is conditional GAN, which can distinguish the type of data generated and output the results [36], and SRGAN, which can restore low-resolution images to high-resolution [27]. Other GANs, such as pix2pix and Cycle GAN mentioned above, transform image features. In this study, we use a GAN that transforms features between images because it is sufficient for outputting an image with the feature of

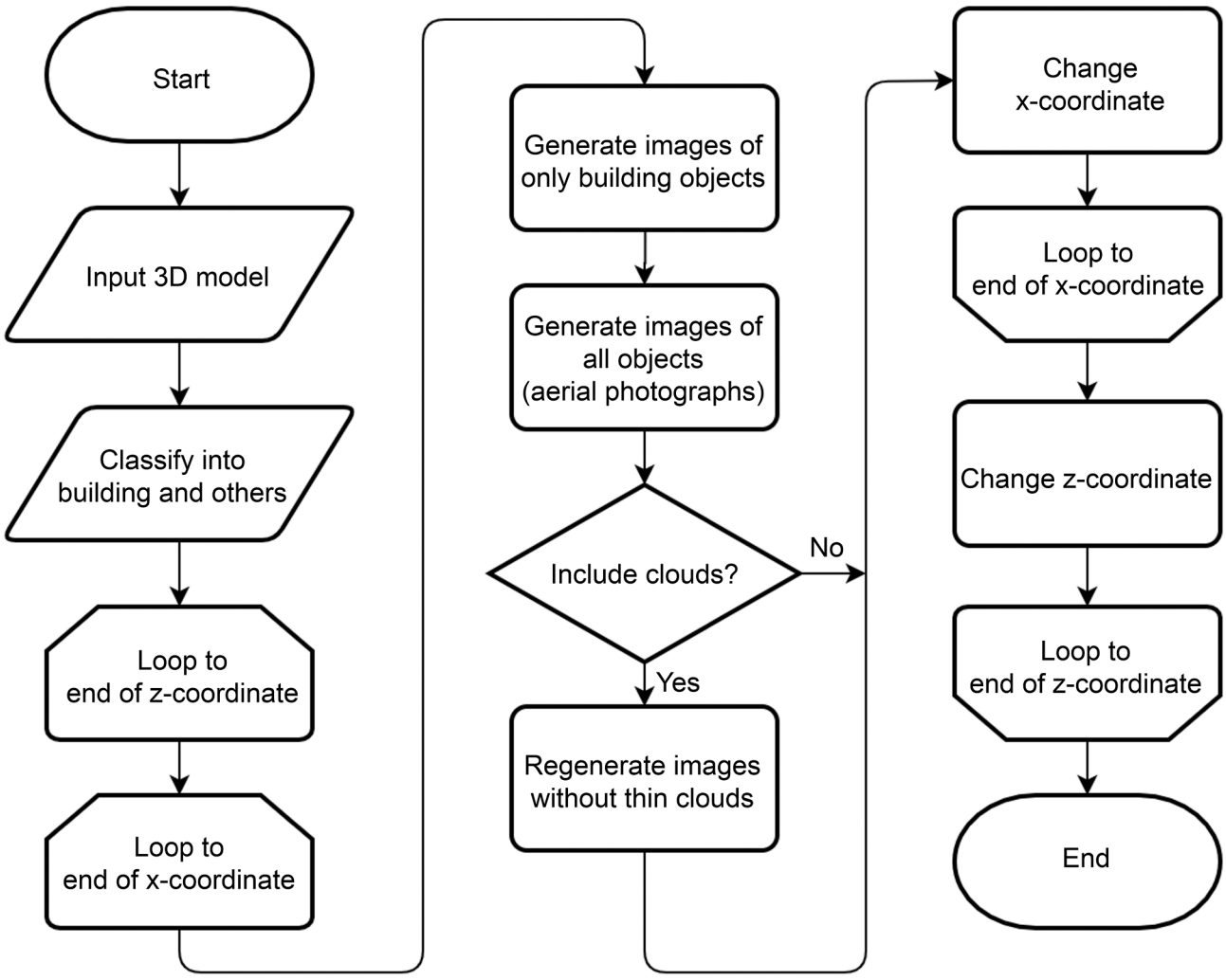

Fig. 1. Flowchart of the proposed method.

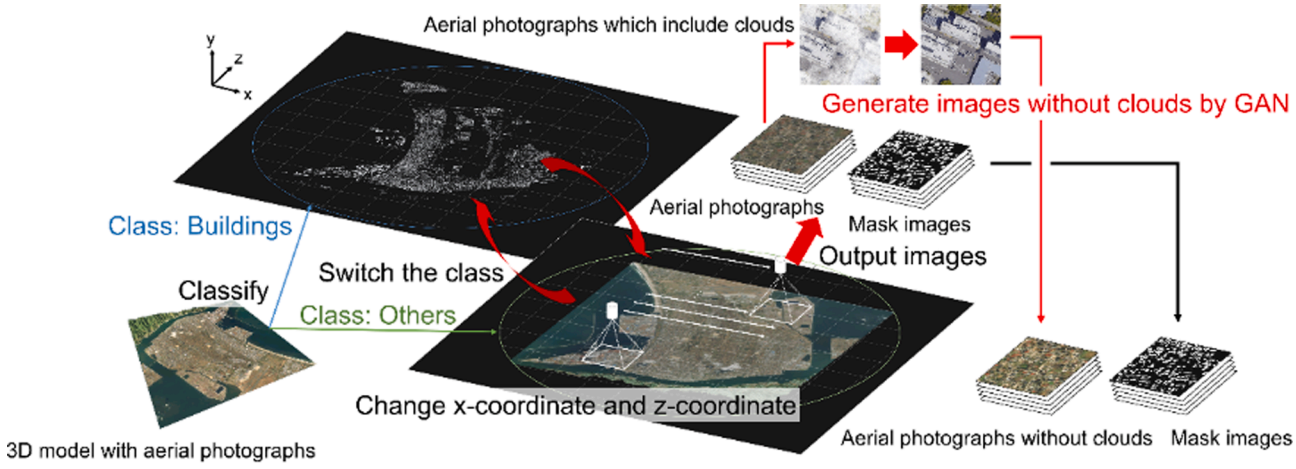

Fig. 2. Conceptual diagram of the proposed method.

containing clouds from an image with the feature of not containing clouds.

4. Development of a prototype system for a verification experiment

We developed a prototype system to verify the accuracy of building detection when a deep learning model is trained on training datasets generated by the proposed method. We also show the building detection

results of the deep learning model trained on the training datasets automatically generated by the prototype system. The data used in this study are summarized in Table 1.

4.1. Development of prototype system

We developed a prototype system to automatically generate training datasets to verify the method proposed in Section 3.

4.1.1. Creation of an enhanced 3D model

The enhanced 3D models of the target areas were generated using Autodesk InfraWorks. The building placement was determined according to fundamental geospatial data provided by the Geospatial Information Authority of Japan. The aerial photographs are pasted on objects in the terrain. The resulting enhanced 3D model and its classification are shown in Fig. 3 and Table 2, respectively.

4.1.2. Generation of aerial photographs and mask images

For the development of the prototype system, we used Unity, a game engine that can load enhanced 3D models, to define a camera object to output images on top of the enhanced 3D model loaded into the game engine. The enhanced 3D model is classified into building classes and other classes, and each image can be output while switching between active and inactive states. The specifications of the PC used for the development and execution of the prototype system are shown in Table 3. When the target area is small, the amount of training data that can be created may be insufficient. In such a case, data augmentation can be used to increase the amount of the training dataset. In this prototype system, rotation is used to output images from the enhanced 3D model because it is easy to implement, and the original image is rotated 90, 180, and 270 degrees.

The aerial photographs and mask images automatically generated by the prototype system are shown in Fig. 4. The left column shows the input images, the middle column shows the mask images automatically generated by the prototype system, and the right column shows the manually generated mask images as ground truth. The white areas are the building masks. The prototype system generated 6956 sets in 438 s. For comparison, it took the author between 5 and 20 min to manually create mask images for each of six aerial photographs, depending on the density of buildings.

4.1.3. Cloud removal by GAN

To generate cloud removal images, we used spatial attention GANs (SpA GAN) [38,31], which use a spatial attention network as the generator. SpA GAN generates an image by converting the features of an image to other types of features. Accordingly, SpA GAN can be used to generate backgrounds obscured by rain and to remove clouds from aerial photographs. The SpA GAN model trained on the open-source RICE dataset [33] was used to remove clouds from aerial photographs containing clouds. The prototype system uses GAN to process all aerial photographs. The color tone of the generated image was corrected to match that of the original aerial photograph. The specifications of the PC used to perform these tasks are shown in Table 4.

Fig. 5 shows the resulting images. The left column shows the input aerial photographs with clouds and the middle column shows the output images without clouds. The time required for GAN to generate 192 cloud removal images was 58 s. The ground truth images used for comparison are shown in Fig. 6. To evaluate the accuracy of the completion, we calculated SSIM (Structural Similarity) [44], a measure of image similarity, which is also used to evaluate the accuracy of image completion by GAN.

Table 1
The data used in this study.

SpaceNet	Features	The dataset containing aerial photographs, building mask images, and road mask images
	Target area	Rio de Janeiro
	Image size	400 × 400
	Number of images	6956
RICE dataset	Features	The dataset containing the reference picture without clouds, the picture of the cloud, and the mask of its cloud.
	Image size	512 × 512
	Number of images	RICE1: 500, RICE2: 450

4.2. Accuracy verification of generated datasets

To verify the accuracy of the training datasets for deep learning automatically generated by the prototype system, we trained U-Net [41], which is a semantic segmentation model that has been proposed for medical image segmentation. Given that the model was designed to detect images of microscopic objects and such as cells, it can accurately detect objects at the pixel level, for example, by increasing the loss at the boundary between the detected object and the background. Accordingly, U-Net can be used for building segmentation.

Other semantic segmentation models include SegNet, ICNet, and Mask-R-CNN. SegNet is a deep learning model capable of segmenting objects in a landscape image in a fast and memory-saving manner. It employs the encoder-decoder structure to perform the segmentation process quickly [2]. ICNet is also a deep learning model that can segment objects in a fast and memory-saving manner [48] but additionally employs an image cascade network structure that incorporates multiple resolution analysis, rather than pixel-by-pixel label inference. Mask-R-CNN is an instance segmentation method that divides the region of an object into pixels while distinguishing between objects of the same category [18]. It is capable of distinguishing between objects of the same category and performing pixel-wise instance segmentation; however, it is not practical because it can only run at about 5 fps on a single GPU and requires high processing power.

We chose U-Net for this study because we only need to detect buildings in aerial photographs, which are static images, with high accuracy. We do not consider real-time processing at high speed to be important for the verification of the prototype system. We selected the city of Sakaiminato in Tottori Prefecture as the target area because it has mostly small buildings, low building density, and a large proportion of the ground surface for which no training dataset is provided by the existing SpaceNet dataset.

To verify the accuracy of the trained U-Net model in detecting buildings, we used intersection over union (IoU; [12], which is an index for evaluating how well the detected area corresponds to the actual area. IoU is calculated by dividing the product set of the actual area and the predicted area by the sum of the two areas, as in Equation (1). The confusion matrix used in the calculation and the schematic diagram representing each value are shown in Table 5 and Fig. 7, respectively, in the form of the building to be detected in this study. Fig. 8 shows an example of a diagram for understanding the accuracy of the IoU. The value of IoU, which is an evaluation index of segmentation, becomes larger when the overlap between the correct and predicted regions is larger. However, the object detection accuracy index is very strict, and even a small deviation between the predicted and correct answers can cause a significant decrease in the value. The example shown in Fig. 8 shows that even when a square of the same shape is shifted by 1/9 vertically and horizontally, (8 × 8) / (100–2) ≈ 0.65 according to the IoU calculation method. As can be seen, even though the detection result is good, the value of IoU is greatly reduced.

$$IoU = \frac{TP}{TP + FP + FN} \tag{1}$$

The training conditions and the specifications of the PC used for training, performing building detection using the training model, and

Fig. 3. Enhanced 3D model generated by Infraworks.

Table 2
Classification of the created enhanced 3D model.

Objects		Types	Classes
Object 1		Building	Building masks
Object 2		Road	Others
Object 3		Water	
…		…	
Object N	Mesh a	Terrain	
	…	…	
	Mesh n	Terrain	
	…	…	

Table 3
Prototype system specifications.

OS	Windows 10 Education 64 bit
CPU	Intel® Core™ i5-7400 CPU @3.00 GHz
GPU	Geforce GTX 1060
RAM	16.0 GB
Storage	HDD (1 TB)
Motherboard	MouseComputer IStDxi-R027

calculating the IoU are shown in Tables 6 and 7, respectively.

4.2.1. Results of building detection by the trained U-Net model

Here we show the building detection results of the U-Net model trained on a set of supervised data generated by the prototype system. The U-Net model trained on unprocessed aerial photographs and mask images are shown below as AGBM-3DMP (automatic generation of horizontal building Mask images by using a 3D model with aerial photographs) Model_baseline. AGBM-3DMP Model_thin cloud removal is a U-Net model trained using aerial photographs and mask images with cloud removal by GAN. Fig. 9 shows the building detection results for aerial photographs of Sakaiminato by AGBM-3DMP Model_baseline and AGBM-3DMP Model_thin cloud removal. The red areas are the correctly detected areas of the buildings. For AGBM-3DMP Model_baseline and AGBM-3DMP Model_thin cloud removal, the IoU was calculated using a set of aerial photographs and mask images of the validation class not used for training; the results were 0.622 and 0.651, respectively.

4.2.2. Building detection results for multiple cities

To verify the accuracy of the U-Net model trained with aerial photographs and mask images automatically generated by the prototype system in detecting buildings on aerial photographs of cities other than Sakaiminato, which was the target of the training, we performed building detection using aerial photographs of several other cities. The target cities are shown in Table 8. Chikugo, Fukuoka Prefecture, was selected because it has the same level of building density as Sakaiminato, whereas Akashi, Hyogo Prefecture, was selected. After all, it has a higher building density compared with Sakaiminato among regional cities and also has mid-rise buildings. To verify the detection accuracy in cities outside of Japan, we selected Bad Wörishofen, Germany, and Homestead, FL, USA, which have different building designs from those in Japan but similar building densities. A sample of the detection results for each city is shown in Fig. 10.

5. Discussion

In this section, we discuss the mask images and aerial photographs generated by the prototype system, cloud removal by GAN, and building verification of the deep learning model.

5.1. Automatic generation of mask images

Our prototype system generated 6956 sets of mask images and aerial photographs without clouds in 438 s. The time to generate the mask images was reduced by automatically generating them from enhanced 3D models in comparison to the manual generating method. The mask images generated by our prototype system are nearly the same as those generated manually. This method generates mask images with detailed shapes. However, it was not able to generate mask images of small warehouses. It is therefore necessary to pre-screen the generated mask images.

5.2. Cloud removal by GAN

In the cloud removal image generated by GAN, the outlines of buildings that were covered by thin clouds in the original image are visible. The training quality was expected to be improved because the

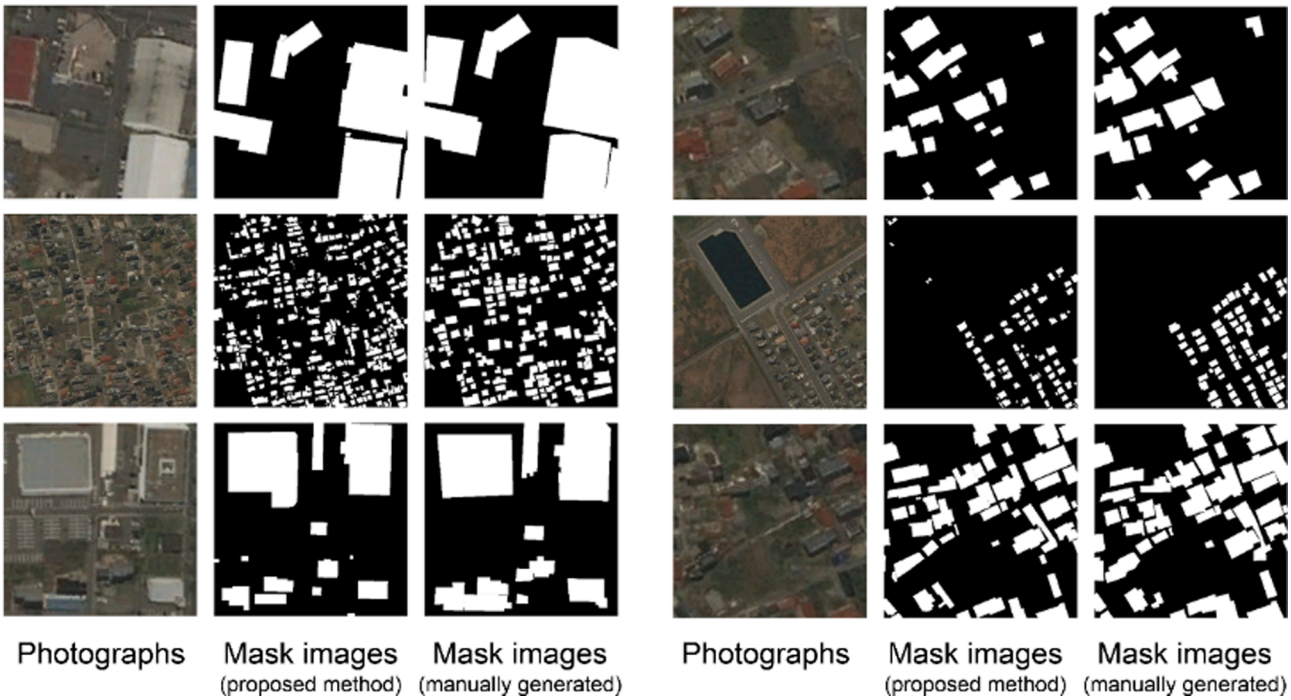

Fig. 4. Aerial photographs and the corresponding mask images.

Table 4
GAN system specifications.

OS	Ubuntu 16.04 LTS
CPU	Intel® Core™ i7-3770 K CPU @3.50 GHz
GPU	Geforce GTX 1060
RAM	28.0 GB
Storage	SSD (2 TB)
Motherboard	ASUS P8H77-V

contours of buildings can be recognized when the model is trained. However, when buildings were covered with thick clouds, they remained hidden below the clouds. Areas in which thick clouds have been removed are complemented as the ground surface. This is because the RICE dataset used for training contains many images of the ground surface. Referring to the interpretation of the bits of each manual mask in the Landsat 8 cloud coverage evaluation validation data [43], buildings cannot be completed in images with a Value greater than 192. Therefore, it is better to remove images in which buildings are

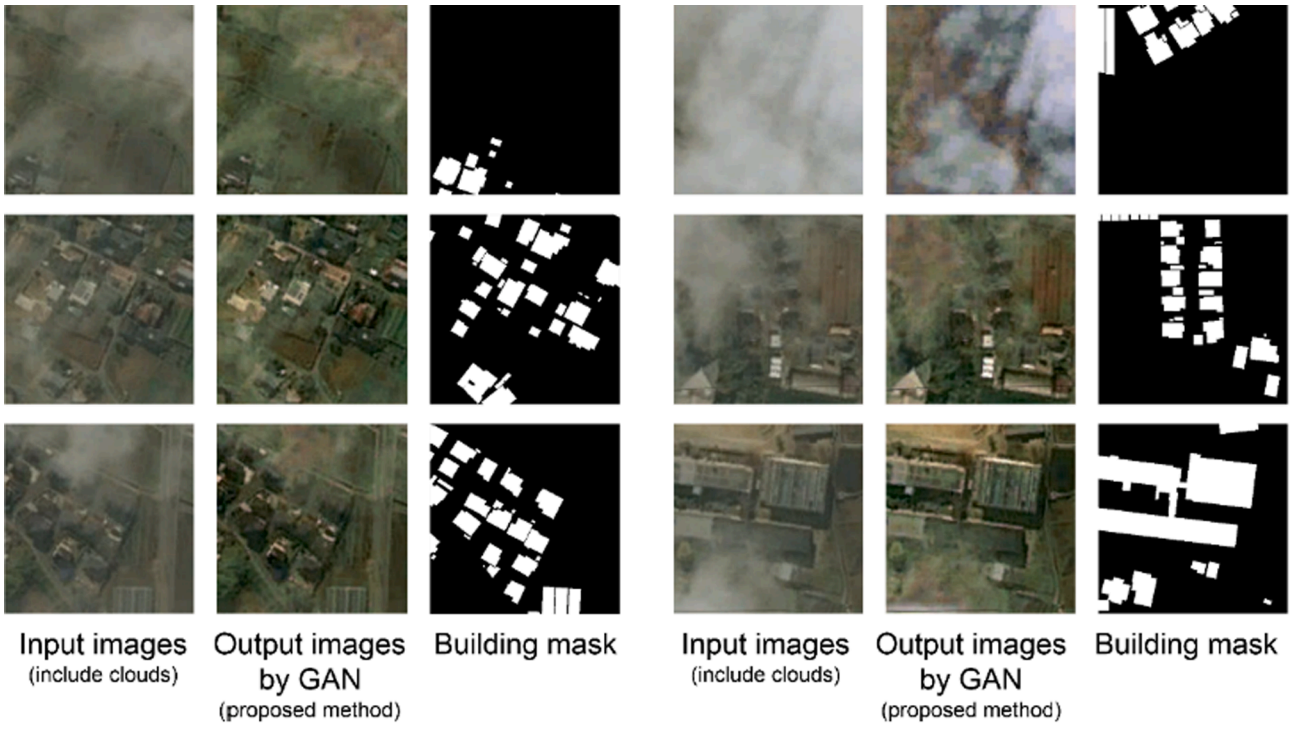

Fig. 5. GAN cloud removal results.

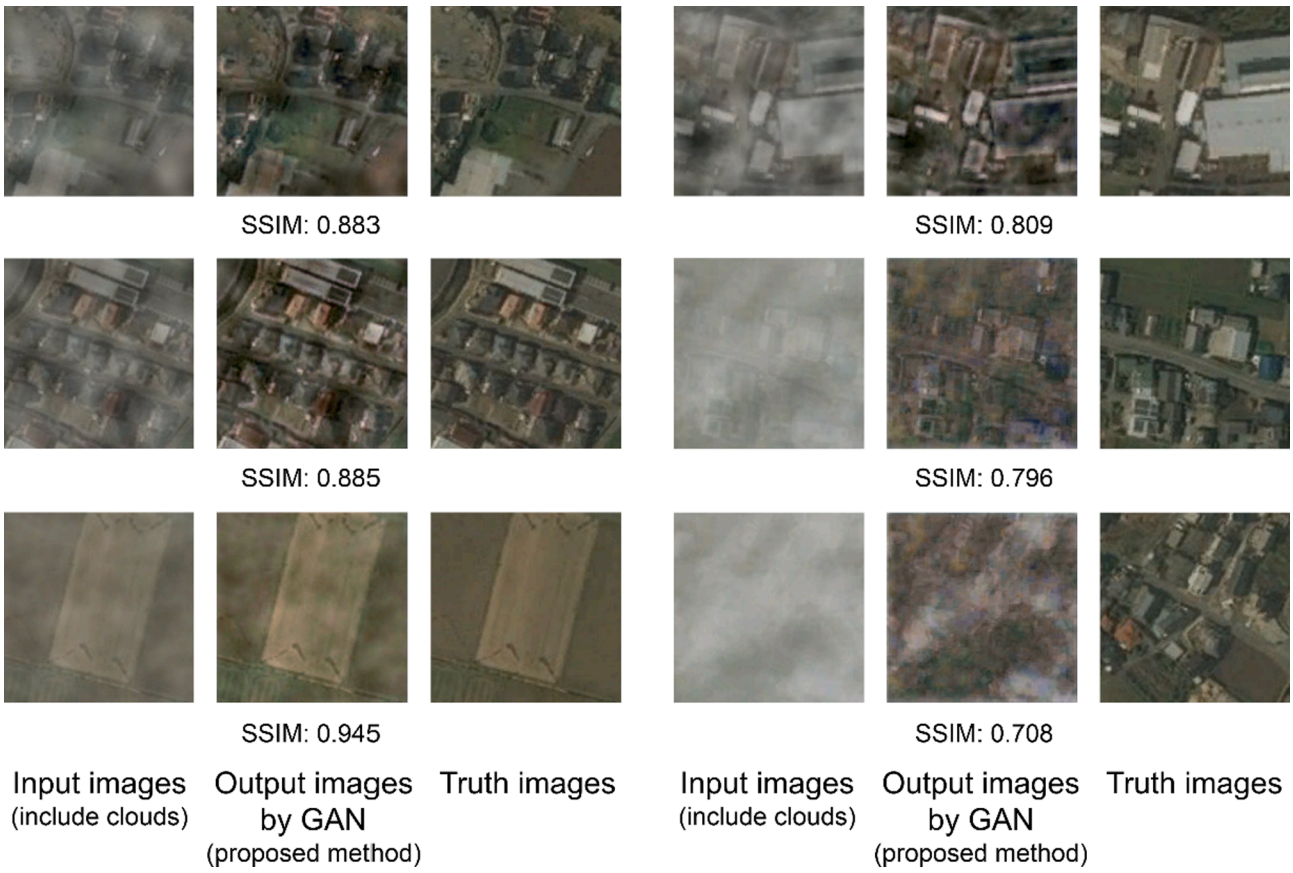

Fig. 6. Comparison of cloudless images and cloud removal images by using GAN.

Table 5
Confusion matrix used in building detection.

	Positive	Negative
True	TP (true positive) Correctly recognizes that it is a building	TN (true negative) Correctly recognizes that it is not a building
False	FP (false positive) Misrecognizes a non-building part as a building	FN (false negative) Misses the building part

completely hidden by thick clouds. For images in which the roofs of buildings were white and their boundary with the clouds was difficult to distinguish, the roofs were also removed.

5.3. Accuracy verification

We trained the model on mask images automatically generated by the prototype system and evaluated the accuracy of the trained model for segmenting buildings in aerial photographs of Sakaiminato. IoU was calculated for accuracy verification by using 1388 test images that were not used for training. The IoU of our trained model (AGBM-3DMP Model_thin cloud removal) was 0.651. Table 9 shows a comparison of the accuracy of the two models (AGBM-3DMP Model_baseline and AGBM-3DMP Model_thin cloud removal) and the detection accuracy of the U-Net model trained on an existing dataset (SpaceNet). The detection accuracy of AGBM-3DMP Model_thin cloud removal was improved compared with that of AGBM-3DMP Model_baseline. To validate the detection features, Accuracy, Precision, and Recall was calculated by Eqs. (2), (3), and (4), respectively. The Accuracy, Precision, and Recall of AGBM-3DMP Model_thin cloud removal were 94.3%, 84.4%, and 74.1%, respectively. AGBM-3DMP Model_thin cloud removal is a model with few false positives. In this verification experiment, U-Net was used for comparison with the existing dataset, but the accuracy might be improved by using a more accurate deep learning model. By validating the proposed method using a public dataset that includes images of buildings hidden by clouds and smoke, such as xBD [16], we can confirm the adaptive range of the method with higher accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

5.3.1. Improvement of accuracy by cloud removal

Figs. 11, 12, and 13 show the scatter plots of the detection accuracy for each image, which indicate that the detection accuracy of AGBM-3DMP Model_thin cloud removal was improved in most of the images compared with AGBM-3DMP Model_baseline, especially in images with a high ratio of buildings. Given that IoU = 0.5 is the threshold for the model to sufficiently detect buildings [23], the number of images in which buildings were sufficiently was 132 for AGBM-3DMP Model_baseline and 1177 for AGBM-3DMP Model_thin cloud removal.

A Chi-square test was used to check whether the improvement in accuracy due to cloud removal by GAN was significant. The calculated p-values met the significance level of 0.05, confirming that there was a significant difference in the detection accuracy of each model. This means that AGBM-3DMP Model_baseline can detect the contours of buildings in more detail for each image compared with AGBM-3DMP Model_thin cloud removal.

In addition to the IoU, which evaluates the degree of omission or

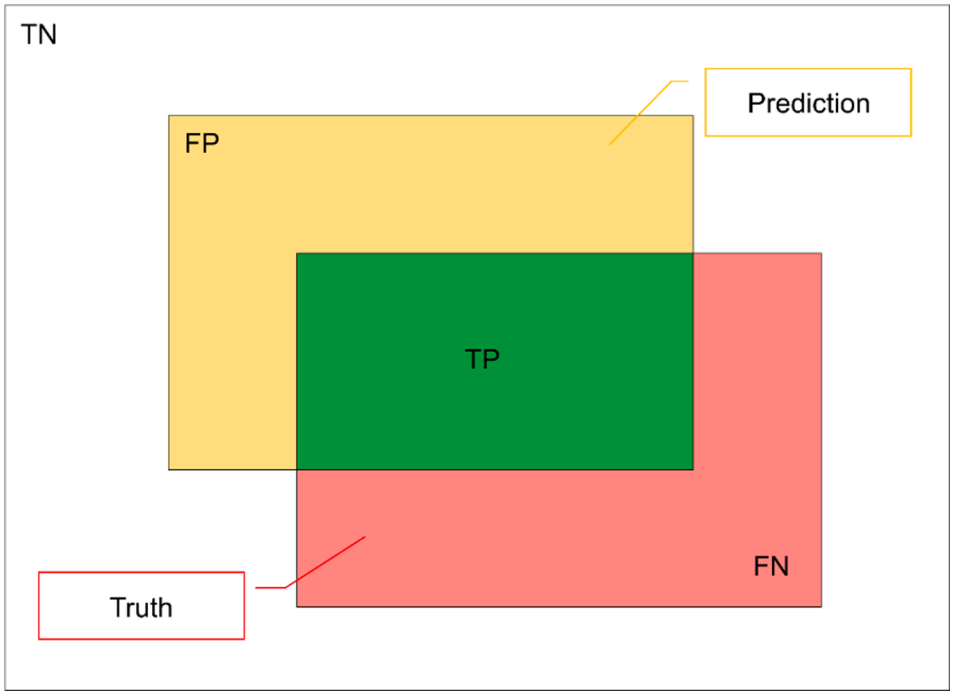

Fig. 7. Example of each region in a confusion matrix.

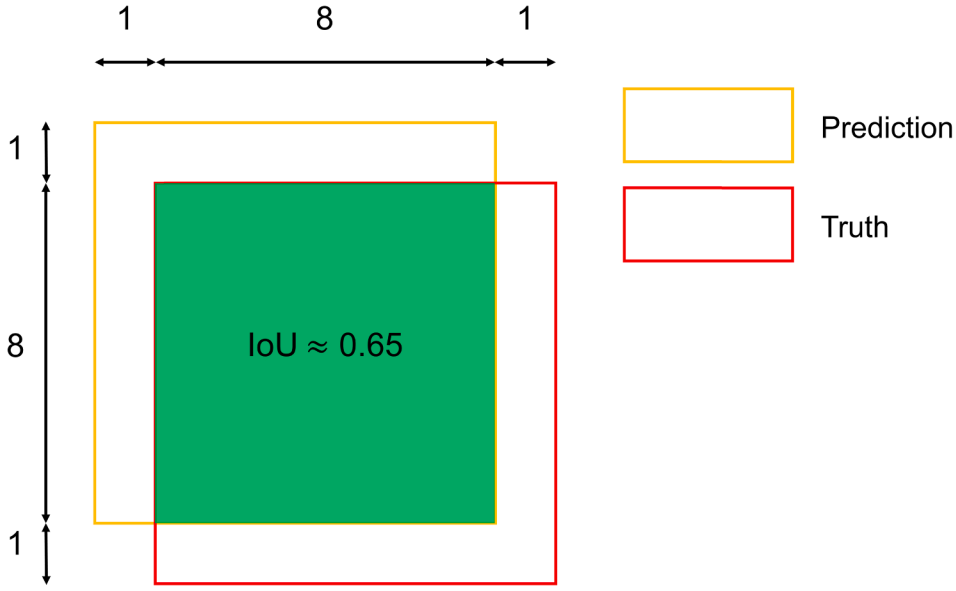

Fig. 8. Illustrative example of IoU calculation.

Table 6
Training conditions.

Number of images used to train the model	6956
Size (pixel)	400px × 400px
Batch size	8
Epoch	50

Table 7
Training device specifications.

OS	Ubuntu 16.04 LTS
CPU	Intel® Core™ i7-3770 K CPU @3.50 GHz
GPU	Geforce GTX 1060
RAM	32.0 GB
Storage	SSD (2 TB)
Motherboard	ASUS P8H77-V

overflow of contour detection, both Precision, which evaluates the strength against false positives, and Recall, which evaluates the strength against misses, was calculated for each image and compared to determine the degree of improvement in detection accuracy. The Accuracy, Precision, and Recall of AGBM-3DMP Model_baseline were 71.4%, 83.0%, and 71.4%, respectively. The Precision, Recall, and IoU of AGBM-3DMP Model_thin cloud removal was higher than those of AGBM-3DMP Model_baseline. Among them, the improvement rate of Recall was higher than that of Precision. This indicates that thin cloud

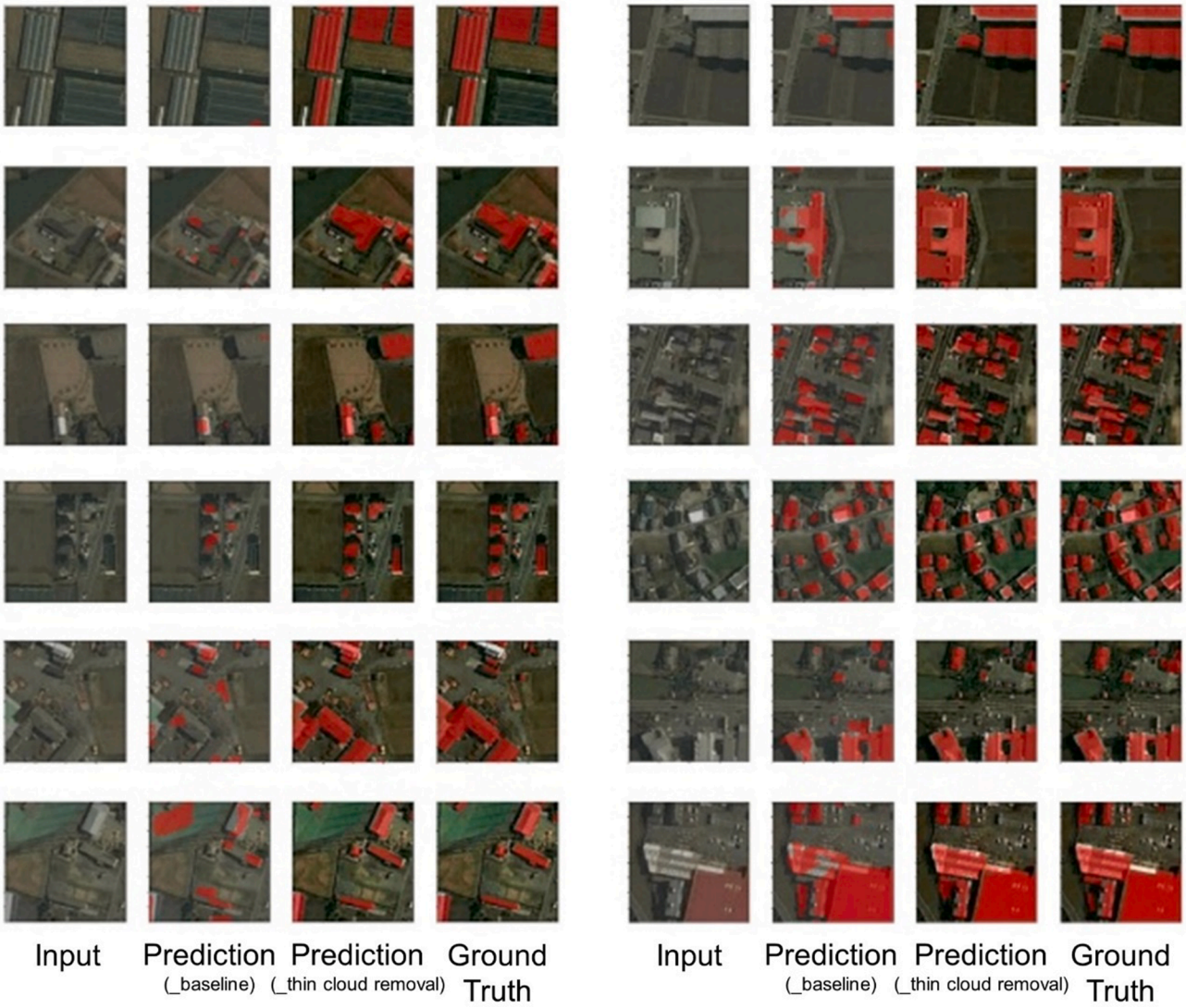

Fig. 9. Building detection results for aerial photographs of Sakaiminato City by each model.

Table 8
Target cities for accurate comparison.

Groups	Target cities
Low density	Sakaiminato, Japan
	Chikugo, Japan
Middle density	Akashi, Japan
Foreign cities	Bad Wörishofen, Germany
	Homestead, FL, USA

removal improved robustness to misses.

5.3.2. Building detection accuracy for multiple cities

Table 10 shows the IoU calculated using 30 aerial photographs of Chikugo, Japan; Akashi, Japan; Bad Wörishofen, Germany; and Homestead, FL.

The building detection results for aerial photographs of Chikugo, which has the same building density as Sakaiminato, showed that buildings were sufficiently detected in all of the 30 images used in the test. The reason for this is that most of the buildings in Chikugo City are low-rise houses with tiled roofs, and their features match those of the buildings in the aerial photographs of Sakaiminato used for training. This means that buildings in the aerial photographs of low-density low-rise cities, which have similar building densities and relatively the same building characteristics, are sufficiently detectable. In contrast, the building detection results for aerial photographs of Akashi, which was selected as a medium-density medium-rise city, showed insufficient detection accuracy for the 30 images used in the test. The IoU of the image with the highest detection result was 0.427. It is considered that the higher density of buildings and the lack of tiled roofs, which are common in Sakaiminato and Chikugo, contributed to the lower detection accuracy.

Detection accuracy was also insufficient in Bad Wörishofen and Homestead. This may be because roof design has a large influence on the identification of buildings in aerial photographs. Ceramic tiles are widely used as roofing materials in Bad Wörishofen whereas asphalt shingles are used in Homestead. Shingles are rarely used on buildings in Japan, including in Sakaiminato, which was the target of this study. Therefore, it is considered that the detection accuracy was not high enough, despite having the same building density. Because deep learning models are highly dependent on the features of the buildings included in the supervisory data used for training, we believe that it is possible to detect buildings with high accuracy in various cities by using this method to automatically generate a dataset of cities with the same features as the city to be detected and used to train the deep learning model. In this way, we can detect buildings in various cities where datasets are not provided with high accuracy.

Fig. 10. Building detection results for aerial photographs of cities chosen for comparison.

Table 9
IoU of the trained models.

Trained model	IoU
U-Net model trained using an existing dataset (SpaceNet dataset)	0.602
AGBM-3DMP Model_baseline (U-Net model trained using unprocessed images)	0.622
AGBM-3DMP Model_thin cloud removal (U-Net model trained using images in which thin clouds were removed by GAN)	0.651

6. Conclusions

Information extracted from aerial photographs is used widely in urban planning and design. It is effective to use AI to detect buildings in aerial photographs to understand the current state of a target area. In recent years, many object detection technologies involving deep learning have been developed. Training datasets for many cities and other areas are not publicly available, and must often be created manually, making it costly to add new targets. To resolve this issue, a method using deep learning was proposed to automatically generate mask images of buildings, roads, and other objects. Because the appearance of a conventional virtual model is similar to but not the same as a photograph, it is difficult to obtain highly accurate detection results in the real world, even when the image is used for deep learning training. Therefore, we aim to improve the representation of 3D models by applying textured aerial photographs to 3D models. However, some aerial photographs include clouds, which can degrade the image quality. In this study, the clouds in these aerial photographs were removed by using GAN to improve training quality.

The proposed method automatically generates mask images and aerial photographs without clouds by using an enhanced 3D model and

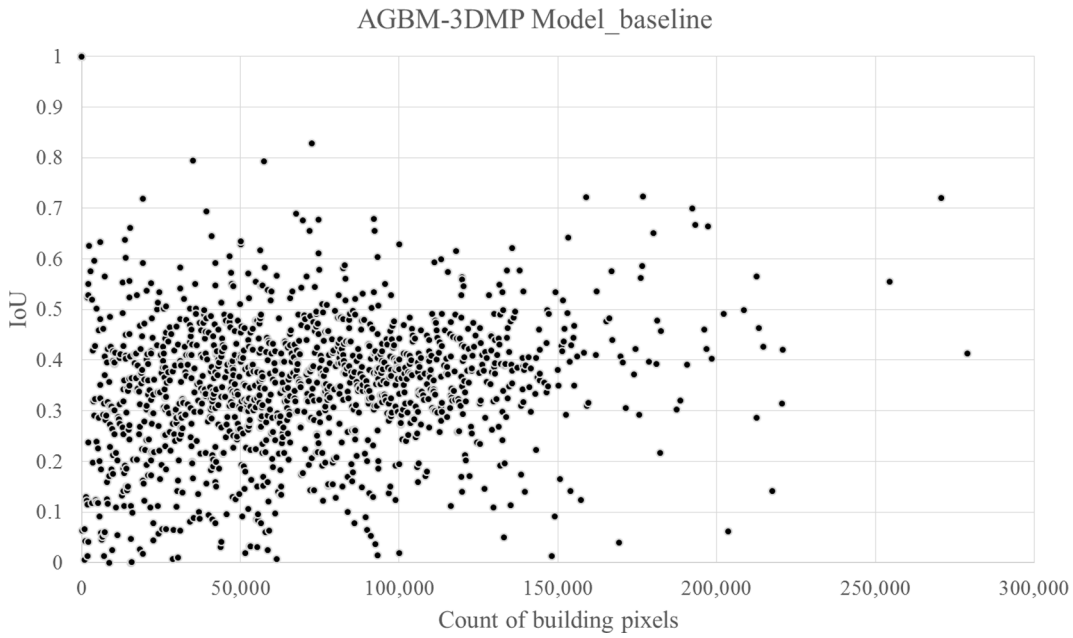

Fig. 11. Detection accuracy for each image (AGBM-3DMP Model_baseline).

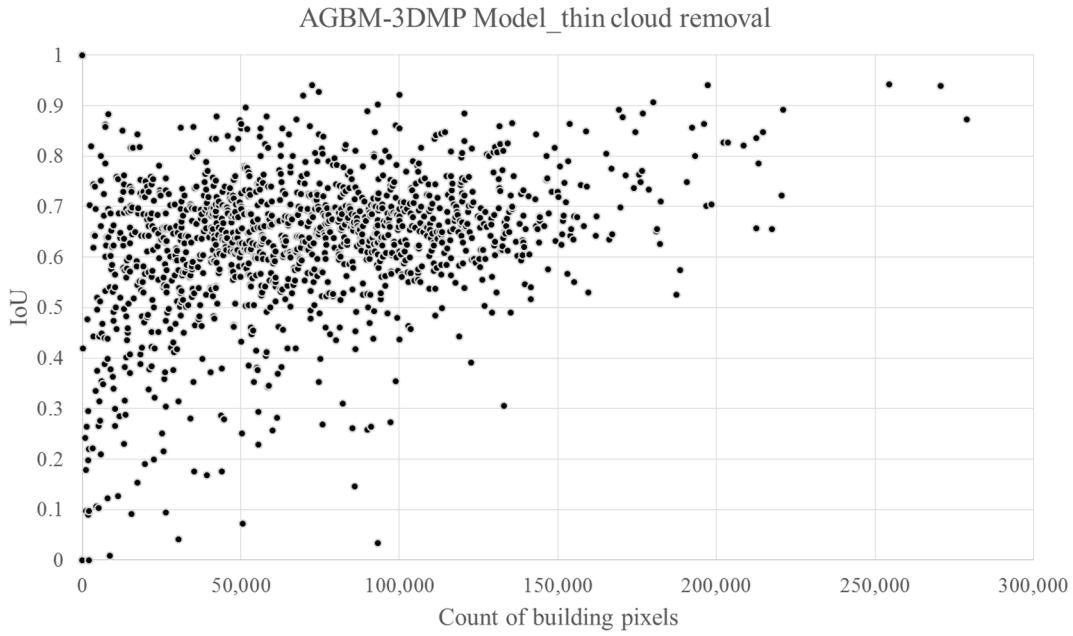

Fig. 12. Detection accuracy for each image (AGBM-3DMP Model_ thin cloud removal).

GAN. In this method, the enhanced 3D models are classified into building classes and other classes. To switch between these classes, the mask images and aerial photographs are taken from the same camera point. After that, the coordinates of the camera object are changed and the process is repeated. This system outputs pairs of mask images and aerial photographs automatically. Aerial photographs that include clouds are regenerated as images without clouds by using GAN.

The prototype system using the proposed method in this study can automatically generate training datasets for deep learning, including aerial photographs and mask images, in a short time compared with methods requiring the manual generation of mask images. The prototype system generated 6956 sets in 438 s. AGBM-3DMP Model_thin cloud removal could detect buildings in aerial photographs with IoU = 0.651. It was shown that cloud removal by GAN was effective at improving the training quality.

The method proposed in this study can be used in situations where buildings are hidden by thin clouds and are visible. However, in areas where buildings are completely hidden by clouds, there is currently a risk of generating false labels, which should be removed by manual operation. It is also necessary to generate the dataset from the year according to the building features in the area to be detected.

The conclusions of the present study are summarized as follows.

- Our prototype system can generate sets of aerial photographs in which clouds are removed by GAN as well as mask images from an enhanced 3D model.
- Cloud removal by GAN improves training quality.

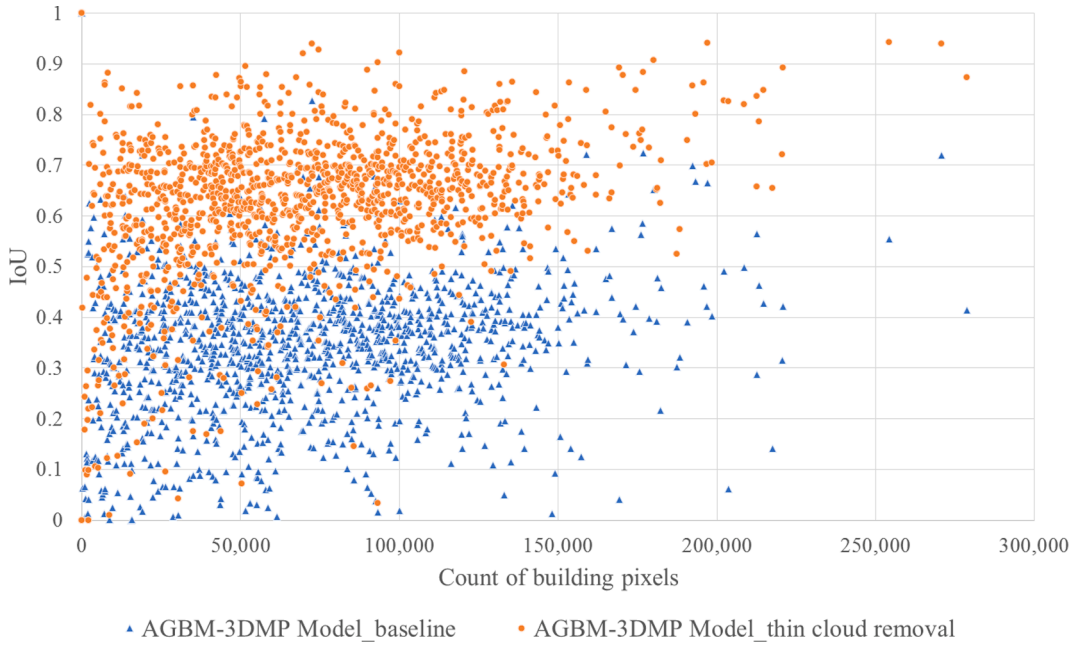

Fig. 13. Detection accuracy for each image (comparison of the two models).

Table 10
Building detection IoU in multiple cities.

Target cities	IoU
Chikugo, Japan	0.690
Akashi, Japan	0.265
Bad Wörishofen, Germany	0.430
Homestead, FL	0.151

- The model-trained datasets generated by our prototype system can detect buildings in aerial photographs with IoU = 0.651.
- This study makes it possible to detect buildings in areas where datasets are not provided with high accuracy and is expected to substantially contribute to land use surveys and disaster damage assessments.

The proposed method can be used not only for buildings in aerial photographs but also for other types of objects. This method might also be used to automatically generate supervised datasets for objects such as roads and rivers in aerial photographs as well as buildings seen from street level. In addition, the mask images generated by this method can be used not only as training data for deep learning but also for visualization to understand cities.

This method generates wrong labels for images in which buildings are completely hidden by thick clouds. Therefore, our future work is to consider the location information of the building as additional input data, so that the GAN can recover the completely hidden building. In addition, since the accuracy of building detection by the U-Net model trained on the dataset generated by our method is greatly affected by the target city for dataset generation and the building characteristics of the target city, it is necessary to select an appropriate city for verification.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Funding: This research was partly supported by JSPS KAKENHI Grant Number JP19K12681.

References

[1] A.M. Abdi, Land cover and land use classification performance of machine learning algorithms in a boreal landscape using Sentinel-2 data, GIScience Remote Sens. 57 (1) (2020) 1–20, https://doi.org/10.1080/15481603.2019.1650447.
[2] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: a deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495, https://doi.org/10.1109/TPAMI.3410.1109/TPAMI.2016.2644615.
[3] C. Barré-Brisebois, H. Halén, G. Wihlidal, A. Lauritzen, J. Bekkers, T. Stachowiak, J. Andersson, Hybrid rendering for real-time ray tracing, in: Haines, E., Akenine-Möller, T. (Eds.), Rayt gems, Berkeley, CA, 2019. https://doi.org/10.1007/978-1-4842-4427-2_25.
[4] K. Bittner, F. Adam, S. Cui, M. Korner, P. Reinartz, Building footprint extraction from VHR remote sensing images combined with normalized DSMs using fused fully convolutional networks, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (8) (2018) 2615–2629, https://doi.org/10.1109/JSTARS.460944310.1109/JSTARS.2018.2849363.
[5] J. Chen, P. Tang, T. Rakstad, M. Patrick, X. Zhou, Augmenting a deep-learning algorithm with canal inspection knowledge for reliable water leak detection from multispectral satellite images, Adv. Eng. Inf. 46 (2020) 101161, https://doi.org/10.1016/j.aei.2020.101161.
[6] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, https://doi.org/10.1109/CVPR.2016.350.
[7] A. Criminisi, P. Pérez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Trans. Image Process. 13 (9) (2004) 1200–1212, https://doi.org/10.1109/TIP.2004.833105.
[8] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, L. Fei-Fei, ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC 2012), 2012. http://www.image-net.org/challenges/LSVRC/2012/ (accessed 4 January 2021).
[9] J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), 2009, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848.
[10] DigitalGlobe, 2019. SpaceNet https://explore.digitalglobe.com/SpaceNet-Thank-You.html. (accessed 9 September 2019).
[11] H. Emami, M.M. Aliabadi, M. Dong, R.B. Chinnam, SPA-GAN: spatial attention GAN for image-to-image translation, IEEE Trans. Multimedia 23 (2021) 391–401, https://doi.org/10.1109/TMM.604610.1109/TMM.2020.2975961.
[12] M. Everingham, S.M.A. Eslami, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The PASCAL visual object classes challenge: a retrospective, Int. J. Comput. Vision 111 (1) (2015) 98–136, https://doi.org/10.1007/s11263-014-0733-5.

[13] T. Fukuda, M. Novak, H. Fujii, Y. Pencreach, Virtual reality rendering methods for training deep learning, analysing landscapes, and preventing virtual reality sickness, Int. J. Arch. Comput. 19 (2) (2021) 190–207, https://doi.org/10.1177/1478077120957544.
[14] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2014), 2014, pp. 580–587. https://doi.org/10.1109/CVPR.2014.81.
[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS), 2014, pp. 2672–2680.
[16] R. Gupta, R. Hosfelt, S. Sajeev, N. Patel, B. Goodman, J. Doshi, E. Heim, H. Choset, M. Gaston, xBD: A Dataset for Assessing Building Damage from Satellite Imagery, 2019. arXiv preprint arXiv:1911.09296.
[17] K. He, J. Sun, Image completion approaches using the statistics of similar patches, IEEE Trans. Pattern Anal. Mach. Intell. 36 (12) (2014) 2423–2435, https://doi.org/10.1109/TPAMI.2014.2330611.
[18] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, IEEE International Conference on Computer Vision, 2961–2969, 2017. https://doi.org/10.1109/ICCV.2017.322.
[19] S. Iizuka, E. Simo-Serra, H. Ishikawa, Globally and locally consistent image completion, ACM Trans. Graph. (Proc. SIGGRAPH) 36 (4) (2017) 1–14, https://doi.org/10.1145/3072959.3073659.
[20] K. Ikeno, T. Fukuda, N. Yabuki, Can a Generative Adversarial Network Remove Thin Clouds in Aerial Photographs? Toward Improving the Accuracy of Generating Horizontal Building Mask Images for Deep Learning in Urban Planning and Design, in: Proceedings of the 26th International Conference on Computer-Aided Architectural Design Research in Asia (CAADRIA 2021), vol. 2, 2021, pp. 377–386.
[21] P. Isola, J.Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1125–1134, https://doi.org/10.1109/CVPR.2017.632.
[22] R. Izutsu, N. Yabuki, T. Fukuda, As-built detection of steel structure using deep learning, in: Proceedings of the 4th International Conference on Civil and Building Informatics 2019, 2019, pp. 25–32.
[23] A. Jabbar, L. Farrawell, J. Fountain, S.K. Chalup, Training deep neural networks for detecting drinking glasses using synthetic images, Neural Inform. Process. 354–363 (2017), https://doi.org/10.1007/978-3-319-70096-0_37.
[24] N. Kawai, T. Sato, N. Yokoya, Image Inpainting Considering Brightness Change and Spatial Locality of Textures and Its Evaluation, in PSIVT2009, 2009, pp. 271–282. https://doi.org/10.1007/978-3-540-92957-4_24.
[25] D. Kido, T. Fukuda, N. Yabuki, Diminished reality system with real-time object detection using deep learning for onsite landscape simulation during redevelopment, Environ. Modell. Softw. 131 (2020) 104759, https://doi.org/10.1016/j.envsoft.2020.104759.
[26] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS 2012), 2012, pp. 1097–1105. https://dl.acm.org/doi/10.5555/2999134.2999257.
[27] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, J. Wang, W. Shi, Photo-realistic single image super-resolution using a generative adversarial, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4681–4690, https://doi.org/10.1109/CVPR.2017.19.
[28] A. Levin, A. Zomet, Y. Weiss, Learning how to inpaint from global image statistics, in: Proceedings Ninth IEEE International Conference on Computer Vision, 2003, pp. 305–312. https://doi.org/10.1109/ICCV.2003.1238360.
[29] D. Li, Q. Xie, X. Gong, Z. Yu, J. Xu, Y. Sun, J. Wang, Automatic defect detection of metro tunnel surfaces using a vision-based inspection system, Adv. Eng. Inf. 47 (2021) 101206, https://doi.org/10.1016/j.aei.2020.101206.
[30] J. Li, Z. Wu, Z. Hu, J. Zhang, M. Li, L. Mo, M. Molinier, Thin cloud removal in optical remote sensing images based on generative adversarial networks and physical model of cloud distortion, ISPRS J. Photogramm. Remote Sens. 166 (2020) 377–389, https://doi.org/10.1016/j.isprsjprs.2020.06.021.
[31] R Li, L Cheong F, R Tan T, Heavy rain image restoration: Integrating physics model and conditional adversarial learning, In IEEE Conference on CVPR (2019) 1633–1642.
[32] T.Y. Lin, M. Marie, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C.L. Zitnick, Microsoft COCO: common objects in context, in: European Conference on Computer Vision (ECCV 2014), 2014, pp. 740–755. https://doi.org/10.1007/978-3-319-10602-1_48.
[33] D. Liu, (BUPTLdy): 2019, RICE_DATASET. https://github.com/BUPTLdy/RICE_DATASET (accessed 19 September2020).
[34] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A. Berg, SSD: Single Shot MultiBox Detector, in: Proceedings of the 14th European Conference on Computer Vision (ECCV 2016), vol. 1, 2016, pp. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2.
[35] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965.
[36] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014. arXiv: 1411.1784. https://arxiv.org/abs/1411.1784.
[37] T. Ogawa, M. Haseyama, H. Kitajima, Restoration for missing intensity of still images using the optical flow, in: IEICE Transactions on Information and Systems (Japanese Edition), 2004, https://doi.org/10.1002/scj.20376.
[38] H. Pan, (Penn000): 2020, SpA-GAN_for_cloud_removal. https://github.com/Penn000/SpA-GAN_for_cloud_removal (accessed 19 September 2020).
[39] Y. Pi, N.D. Nath, A.H. Behzadan, Convolutional neural networks for object detection in aerial imagery for disaster response and recovery, Adv. Eng. Inf. 43 (2020) 101009, https://doi.org/10.1016/j.aei.2019.101009.
[40] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91.
[41] O. Ronneberger, P. Fischer, T. Brox, Unet: Convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.
[42] T. Tamura, A. Kato, H. Obanawa, T. Yoshida, Three height measurement from aerial images taken by a small Unmanned Aerial Vehicle using Structure Motion, J. Jpn. Soc. Reveget. Tech 41 (1) (2015) 163–168, https://doi.org/10.7211/jjsrt.41.163.
[43] U.S. Geological Survey, L8 Biome Cloud Validation Masks. U.S. Geological Survey, 2016. https://doi.org/10.5066/F7251GDH.
[44] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612, https://doi.org/10.1109/TIP.2003.819861.
[45] N. Yabuki, N. Nishimura, T. Fukuda, Automatic Object Detection from Digital Images by Deep Learning with Transfer Learning, in: Workshop of the European Group for Intelligent Computing in Engineering, Springer, Cham, 2018, pp. 3–15. https://doi.org/10.1007/978-3-319-91635-4_1.
[46] H.e. Zhang, V. Sindagi, V.M. Patel, Image de-raining using a conditional generative adversarial network, IEEE Trans. Circuits Syst. Video Technol. 30 (11) (2020) 3943–3956, https://doi.org/10.1109/TCSVT.7610.1109/TCSVT.2019.2920407.
[47] Q. Zhang, Y. Wang, Q. Liu, X. Liu, W. Wang, CNN based suburban building detection using monocular high resolution Google Earth images, Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS) 661–664 (2016), https://doi.org/10.1109/IGARSS.2016.7729166.
[48] H. Zhao, X. Qi, X. Shen, J. Shi, J. Jia, Icnet for real-time semantic segmentation on high-resolution images, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 405–420, https://doi.org/10.1007/978-3-030-01219-9_25.
[49] K. Zhao, J. Kang, J. Jung, G. Sohn, Building extraction from satellite images using mask R-CNN with building boundary regularization, in: Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018, pp. 247–251. https://doi.org/10.1109/CVPRW.2018.00045.
[50] K. Zhou, Y. Chen, I. Smal, R. Lindenbergh, Building segmentation from Airborne VHR Images Using mask R-CNN, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-2/W13 (2019) 155–161, https://doi.org/10.5194/isprs-archives-XLII-2-W13-155-2019.
[51] J.Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2223–2232, https://doi.org/10.1109/ICCV.2017.244.