*remote sensing*

MDPI

*Article*

# TCDNet: Trilateral Change Detection Network for Google Earth Image

**Junhao Qian [1], Min Xia [1,*], Yonghong Zhang [1], Jia Liu [1] and Yiqing Xu [2]**

[1] Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China; 20181223052@nuist.edu.cn (J.Q.); zyh@nuist.edu.cn (Y.Z.); liujia@nuist.edu.cn (J.L.)

[2] College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China; yiqingxu@njfu.edu.cn

* Correspondence: xiamin@nuist.edu.cn

Received: 27 June 2020; Accepted: 17 August 2020; Published: 19 August 2020


**Abstract:** Change detection is a very important technique for remote sensing data analysis. Its mainstream solutions are either supervised or unsupervised. In supervised methods, most of the existing change detection methods using deep learning are related to semantic segmentation. However, these methods only use deep learning models to process the global information of an image but do not carry out specific trainings on changed and unchanged areas. As a result, many details of local changes could not be detected. In this work, a trilateral change detection network is proposed. The proposed network has three branches (a main module and two auxiliary modules, all of them are composed of convolutional neural networks (CNNs)), which focus on the overall information of bitemporal Google Earth image pairs, the changed areas and the unchanged areas, respectively. The proposed method is end-to-end trainable, and each component in the network does not need to be trained separately.

**Keywords:** change detection; convolutional neural networks; deep learning

## 1. Introduction

Change detection is an important technique in remote sensing image analysis. It compares two images at the same position but of different time periods to locate the changed areas. Change detection has been widely used in urban sprawl tracking [1], resource exploration [2,3], land utilization detection [4], and post-disaster monitoring [5]. The resolution of a remote sensing image is extraordinarily large, whereas the proportion of the changed areas in the whole image is very small. Therefore, it is often time-consuming and laborious to compare such large images manually [6]. In the past few decades, people have proposed many methods for change detection.

A key problem of change detection is modeling temporal correlations between bitemporal images. Different atmospheric scattering conditions and complicated light scattering mechanisms make change detection problems highly nonlinear. Thus, a task-driven and learning-based approach is required to solve detection problems [7]. According to whether the dataset has sufficient prior knowledge, a detection problem can be divided into unsupervised [8,9] and supervised [10,11] methods. Unsupervised learning methods do not need to obtain prior knowledge from labeled data, whereas supervised learning methods have to infer the detection areas through labeled training data.

The unsupervised learning methods have been applied to change detection in many recent studies [12,13]. Generally, these methods mainly focus on the generation and analysis of differential images, extracting information from either original images or differential images to detect which areas change. Common methods include principal component analysis (PCA) based on k-means

clustering [14], multivariate alteration detection (MAD), and the iteratively reweighted multivariate alteration detection (IR-MAD) [15]. The PCA is one of the most famous subspace learning algorithms [16]. As a linear transformation technique, this method performs decorrelation on images. However, the PCA relies on statistical characteristics of images. If the data in the changed areas and unchanged areas are unbalanced, the model performance will be seriously affected [17]. The MAD is another unsupervised multitemporal image change analysis method. Its main mathematical essence is canonical correlation analysis (CCA) and band math in multivariate statistical analysis. However, this algorithm still cannot completely improve current multi-element remote sensing image methods. Based on the MAD algorithm, the IR-MAD algorithm was studied and put forward in combination with expectation-maximization (EM) algorithms [18]. The core idea of the algorithm is that the initial weight of each pixel is 1, and each iteration gives a new weight to each pixel in the two images. The unchanged pixels have larger weights through calculation, and the values of final weights are the basis for determining whether a pixel changed. After several iterations, the weight of each pixel would be stable when the change is less than a set threshold and the iteration will stop if it changes no more. Another classic method is the change vector analysis (CVA) proposed by Malila [19]. It uses a simple method to perform differential operation on image data of various wavebands in different periods, in this way, it evaluates the change of each pixel, and then forms a change vector of various wavebands. In addition, many algorithms were improved based on the CVA. Nevertheless, unsupervised learning methods cannot make use of prior knowledge of marked data, and relay on the assumptions of some models or similar rules to distinguish changed areas. Furthermore, unsupervised methods require targeted tuning of models to adapt to different environments, which is very time-consuming and laborious. Overall, unsupervised methods have certain limitations in change detection research.

Supervised learning methods use labeled training data to learn which areas changed. Traditional supervised learning methods include random forests (RFs) [20], convolutional neural networks (CNNs), etc. Because of the rapid development of graphics processing units (GPUs), deep learning methods can be applied in various fields [21–23]. In the field of change detections, the general end-to-end 2-D convolutional neural network [24] effectively learned distinguishing features from higher levels by a 2-D CNN and introduced a hybrid affinity matrix fused with subpixel representations to improve its generalization ability. Yang [25] performed change detection and land cover mapping simultaneously and used land cover information to help predict changed areas. Wang [26] proposed region-based CNNs to extend object detection for image change detection tasks. Zhan [27] proposed a change detection method based on deep Siamese convolution networks. This method obtained different feature maps by inputting two images into the same CNN, and then the method processed data from feature sets to detect changes between the images by the knowledge that the feature vectors of changed pixel pairs were far apart from each other.

Because most of the change detection tasks were performed at the pixel level, which is naturally associated with semantic segmentation tasks, a change detection task could be transformed into a two-class semantic segmentation problem. Semantic segmentation tasks in deep learning were mostly based on fully convolutional networks (FCNs) [28], which could classify each pixel in the image independently and quickly. Zhang [29] and Arabi [30] proved that FCN methods could be well applied to image change detection tasks. However, these works only optimized the structure of the original model; features of image change detection were not improved. For example, images acquired in different time periods might have some deviations in building angles, which would lead to a low accuracy of the model in identifying changed areas.

Most of the images studied by the above deep learning methods has low spatial resolution, when the spatial resolution and the height of the object increase, those methods do not work well as they have long calculation time and low accuracies, so this paper proposes a trilateral change detection network (TCDNet). The proposed model is consisted of three branches: The main branch is responsible for multiscale extraction of the overall information of the bitemporal Google Earth images to obtain a raw change prediction map. The other two auxiliary branches are the difference module and the

assimilation module, which carry out weight trainings on changed and unchanged areas, respectively. The overall framework of the model is shown in Figure 1. Through the cooperation of the two auxiliary modules, the main network improves its prediction accuracy. The main contributions of this work are as follows. (1) The proposed network is end-to-end trainable, and each component in the network does not need separate trainings, and (2) according to the characteristics of change detection tasks, targeted optimization is carried out. A main module and two auxiliary modules are arranged to cooperate with each other to promote information exchanges between the changed and unchanged areas and thus to improve the accuracy of the prediction map. Two auxiliary modules are trained together with the main module, and no additional separate training is required. (3) A new bitemporal Google Earth image dataset was collected, including different categories of roads, railways, factories, buildings, farmland, and so forth. There are enough change categories to ensure that the model can be trained to cope with most changing environments. All ground truth of data is manually annotated to ensure the accuracy of data. Experiments show that the established dataset is feasible for quantitative evaluation of change detection tasks.

The rest of paper is as follows. Section 2 introduces the proposed model. Section 3 analyzes the composition of the data set. Section 4 discusses the experimental results in details. Section 5 summarizes this paper and puts forward the future work.

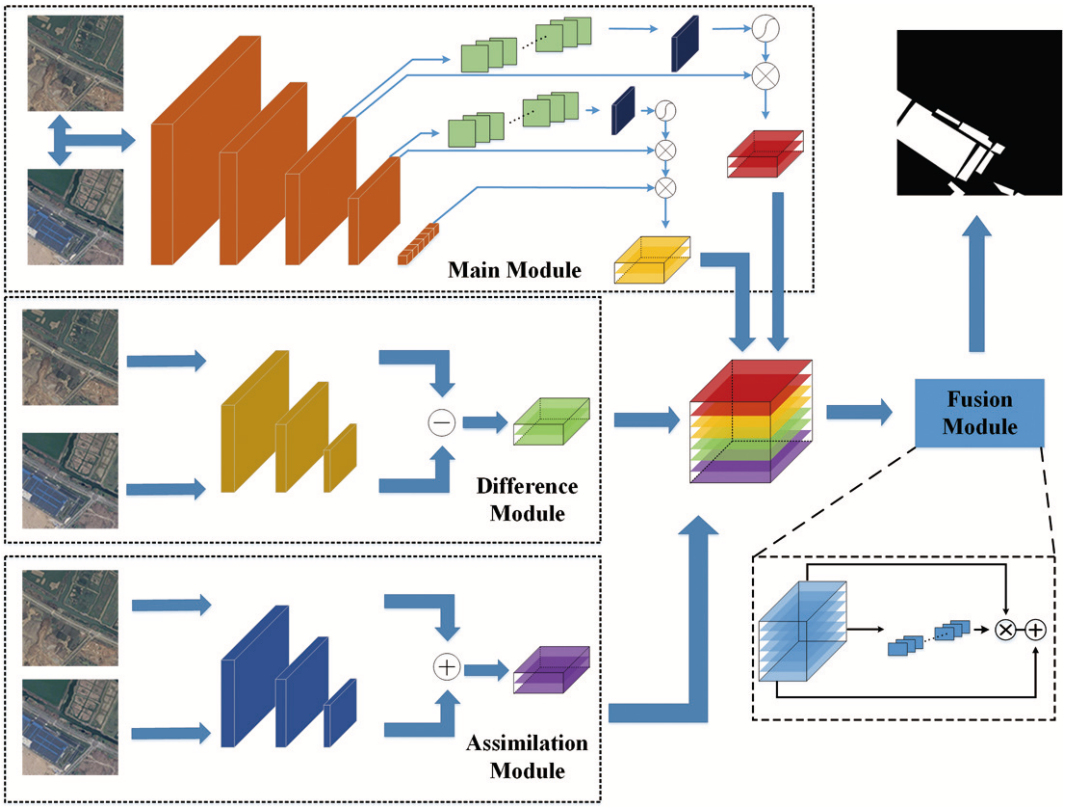


**Figure 1.** Trilateral change detection network (TCDNet) framework. The model consists of four parts: main module, difference module, assimilation module, and fusion module.

## 2. Methodology

In previous research, CNNs were proven to be effective for the remote sensing image segmentation [31,32]. In many similar visual tasks, it is usually necessary to use an extractor to extract feature information of diverse scales from images. CNNs' structures are similar to pyramid structures with multi-scales and multi-level characteristics. Thus, it was natural to use them in our image change detection research. The network automatically learned multiscale features from rough to fine through a series of convolutions from shallow to deep. CNNs directly use typical image

classification as the backbone network of a model, which is helpful to improve the accuracy of the algorithm. Furthermore, this paper is aimed at pixel-level classification; a single pixel depends on its surrounding pixel information heavily, and pooling layers and convolution layers in CNNs can handle this problem well. To sum up, we choose the neural network as the underlying framework of our model.

The TCDNet proposed in this paper is consisted of three modules: one main network module and two auxiliary modules. The main network module is responsible for systematic feature extraction of bitemporal Google Earth images pairs, and it takes advantage of convolution networks to obtain rich semantic information. The other two auxiliary modules are a difference module and an assimilation module, which focus on changed and unchanged areas, respectively, assisting the main network in carrying out change detections. To fuse the outputs of these three modules, a fusion module is designed at the end of the model to complete the task. The overall framework of the model is shown in Figure 1.

### *2.1. Main Module*

This module is mainly used for detailed feature extraction of bitemporal Google Earth image pairs. In selection of the backbone network, a mainstream strategy is to use the model that performs well in image classification tasks (Large Scale Visual Recognition Challenge [33]), such as the Visual Geometry Group Network (VGG) [34], Residual Network (ResNet) [35], Densely Connected Convolutional Networks (DenseNet) [36], and so forth. Because of the particularity of change detection tasks, where the depth of the neural network has an influence on the final result, the general idea is to design a network as deep as possible. However, as the depth of the network increases, the phenomenon of gradient disappearance manifests, and thus the training of the network will not be very effective, and the accuracy will even decrease. Nevertheless, a shallower network cannot fully extract the feature information of the image; therefore, the ResNet is chosen as the backbone network because it can well solve the gradient disappearance problem by using residual modules.

The core of residual modules lies in constant mapping. These modules solve the degradation problem in deep learning very well, with the increasement of network layers, the accuracy on the training set is saturated or even decreased. It is relatively difficult to directly fit convolution layers to a potential constant mapping function $H(x) = x$, but converting it to $h(x) = f(x) + x$ will make it easier. When $F(x) = 0$, an identity mapping $h(x) = x$ will be obtained [31]. The main function of residual networks is to remove the same parts and highlight minor changes in data.

It is well known that the most commonly used strategy is to combine low-level semantic features with high-level semantic features [28] to improve final prediction results. Thus, the backbone network uses 16 downsampling layers and 32 downsampling layersto extract features of different levels to construct a feature pyramid structure.

In order to extract more complete feature information, operations are as follows. The module uses global average pooling to get the global information of 16 times and 32 times of downsampling layers, then it uses a $1 \times 1$ convolution network and the sigmoid function, finally it multiplies the obtained feature vector with the original 16 times and 32 times of downsampling layers to achieve feature learning [37]. This weight vector can adjust features, which is equivalent to feature selection and combination. The sigmoid function can normalize the elements of the eigenvector to between 0 and 1. If the vector value is closer to 1, the channel would be more important. On the contrary, if the vector value is closer to 0, it would be less important.

### *2.2. Auxiliary Module*

The essence of the image change detection algorithm lies in determining changed areas through bitemporal images. This task can be divided into two interrelated parts: one module to detect changed areas, and another to detect unchanged areas. The outputs of the three branches are put together to

improve the accuracy. To accomplish this task, two auxiliary modules are designed: the assimilation and the difference modules.

Because some objects have certain deviations from different viewing angles of sensors, not all pixels of the two images taken in different periods are one-to-one paired. Especially when the observation distance is short, the height of the object is too high and the spatial resolution is very high as well, this will become a problem. As shown in Figure 2, these areas were taken at different time periods of the same place. According to the figures, these parts (boxed in red on Google Earth images) do not change, the visual differences are caused by viewing angles of sensors. Traditional methods (such as PCA-means and IR-MAD) could not solve this problem well. However, deep learning methods could help solve this issue. Deep learning methods convert area parameters into single values for comparison. Taking the difference module as an example, two images pass through three convolution layers, and the size of the obtained feature map is only one-eighth of the original image. That is, the information corresponding to an $8 \times 8$ original image is mapped into a single value after convolutions.

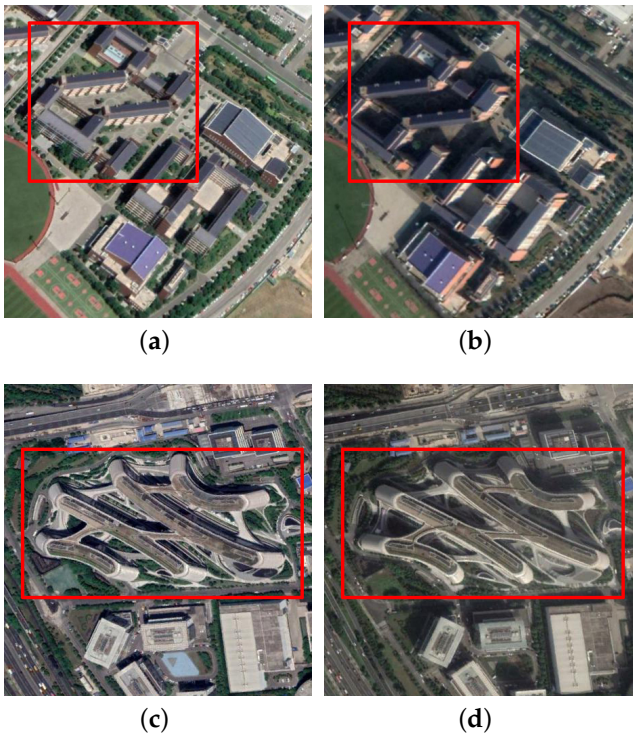


**Figure 2.** Bitemporal Google Earth images. (**a**) Taken in 2019 in Nanjing, China. (**b**) Taken in 2018 in Nanjing, China. (**c**) Taken in 2019 in Shanghai, China. (**d**) Taken in 2018 in Shanghai, China.

In Figure 3, the $4 \times 4$ receptive field in the blue feature layer is mapped to the $2 \times 2$ parameter in the orange feature layer through convolutions, and then it is mapped to the $1 \times 1$ parameter in the green feature layer through convolution operations. After two convolution layers, the parameters of a $4 \times 4$ area in the original image are finally mapped to a $1 \times 1$ parameter. This series of operations enables a single parameter in the final output feature layer to represent a small $4 \times 4$ image in the original image. This manipulation could extract global information from a small area, and convert the area parameters into single values for comparison, it can well handle the position deviation problem caused by different viewing angles of sensors.

Using deep learning methods to solve change detection problems is similar to image segmentation, which generates pixel-wise output. In image segmentation, the image is normally input to the CNN and FCN. A CNN first performs convolutions and poolings, reducing the size of the image and increasing the receptive field at the same time; image segmentation is to predict the output at pixel-wise, so it

is necessary to upsample a smaller image size after pooling to the original image size for prediction. There are two key points in image segmentation: one is pooling to reduce the image size and to increase the receptive field, and the other is upsampling to expand the image size. In the procedure of reducing and increasing the size, some information is lost, so a new operation is needed to design to get more information without pooling. Dilated convolution [38] meets all the above requirements, and it is widely used in computer vision. The extended convolution can make the receptive field to grow exponentially, and it will not increase the model parameters or computation while expanding the receptive field. In general, image pooling and undersampling operations will lead to information loss. The expanded convolution can replace pooling, and the expanded receptive field enables the output of each convolution to contain a larger range of information. In addition, dilated convolution used dilated factor (DF) to indicate the size of convolution expansion. Instead of filling blank pixels between pixels, dilated convolution skips some pixels the input remains the same, by carrying out zero-filling operation on the convolution kernel, a larger perception field is achieved.

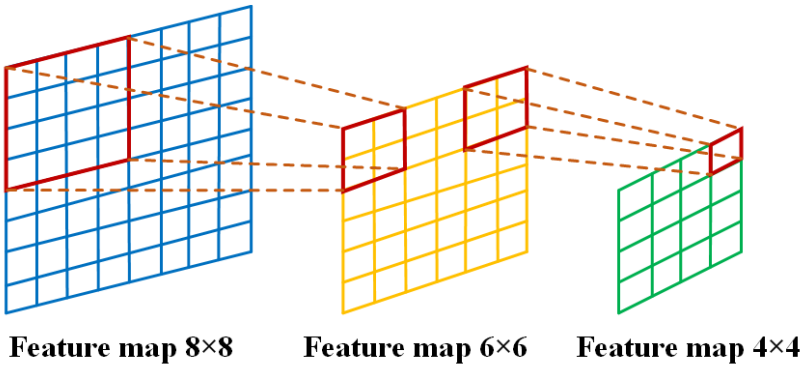


**Figure 3.** Schematic diagram of feature shrinkage.

As shown in Figure 4, three images show the schematic diagrams of receptive fields represented by different dilated factors, and all holes are filled with zeroes. When the factor is 3, the receptive field can reach 121. A wider receptive field of vision can be obtained and is equivalent to using a $11 \times 11$ convolution kernel. It can be seen that the parameters of the convolution kernel remain unchanged, and the size of the receptive field increases exponentially as the dilated factor increases. The growth formula of the two-dimensional dilated convolution receptive field can be expressed as

$$R_i = (4 \times i - 1)^2, \tag{1}$$

where $i$ denotes the dilated factor and $R_i$ is the receptive field size of the convolution kernel. The advantage of dilated convolution is that, without losing information due to poolings, the receptive field of the convolution kernel can be enlarged so that the output after each convolution could contain a larger range of information as far as possible.

Figure 5 illustrates how dilated convolutions increase the scope of the receptive field. Figure 5a and Figure 5b, respectively, represent the receptive fields obtained by ordinary convolutions and dilated convolutions after three convolution operations. It can be clearly seen from the figures that by using dilated convolutions, the information contained in the same parameter (red circle) is about 3.8 times that of ordinary convolutions (blue circle) in the feature map. Thus, it can be seen that by using dilated convolutions, the same parameter in the network can contain more surrounding information, and the building deviation caused by different viewing angles of sensors can be solved by a more friendly way.

Furthermore, the difference module and the assimilation module perform their respective functions. The assimilation module focuses on comparing the same areas, and the difference module is responsible for screening the characteristics of the changed areas. The three branches in the model

constitute a whole network, and supervised training is carried out at the same time to update the parameters in the network. The model judges whether a pixel is in a changed area by a pair of surrounding environment information, and all the information is obtained by automatic learning of the model. In the process of model training, we set up a loss function to assist the training. The outputs of these two modules are visualized and attached to the original images for display to understand the functions of auxiliary modules intuitively.

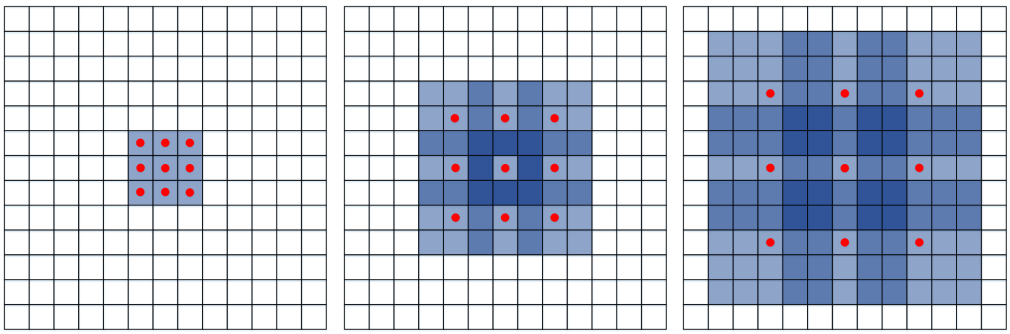


**Figure 4.** Schematic diagram of different dilated factors. (**a**) The original convolution kernel, with the dilated factor set to 1. (**b**) A convolution kernel with dilated factor of 2. (**c**) A convolution kernel with dilated factor of 3.

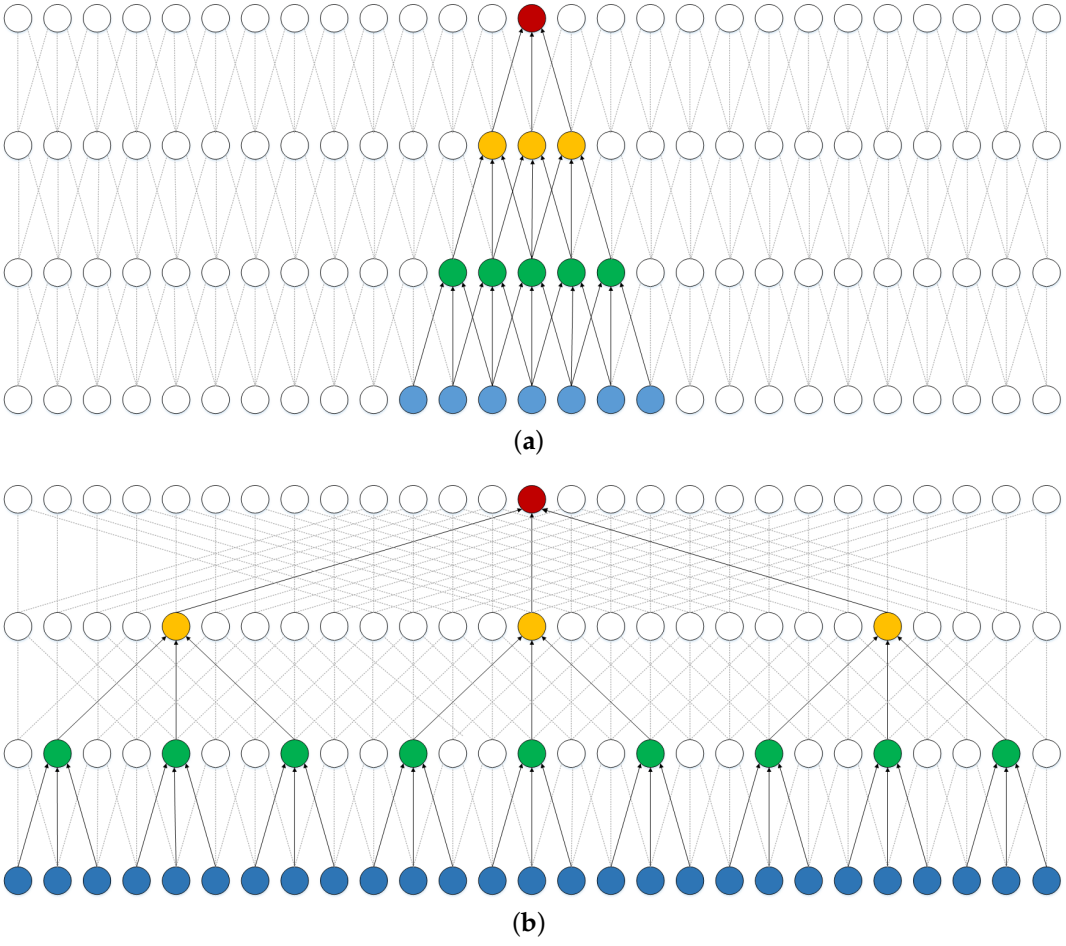


**Figure 5.** Receptive field of different convolution. (**a**) Ordinary convolution; (**b**) dilated convolution.

The images in Figure 6a,b are taken at the same place at different times. The green and blue areas in Figure 6c,d are areas with dense weights, indicating that these areas receive more attention. Figure 6c is the heat map output by the difference module. It can be seen from the diagram that the difference module focuses more on the changed areas in the bitemporal Google Earth images. Figure 6d is the heat map output by the assimilation module. Comparing Figure 6d with Figure 6c, it is seen that the assimilation module focuses on unchanged areas.

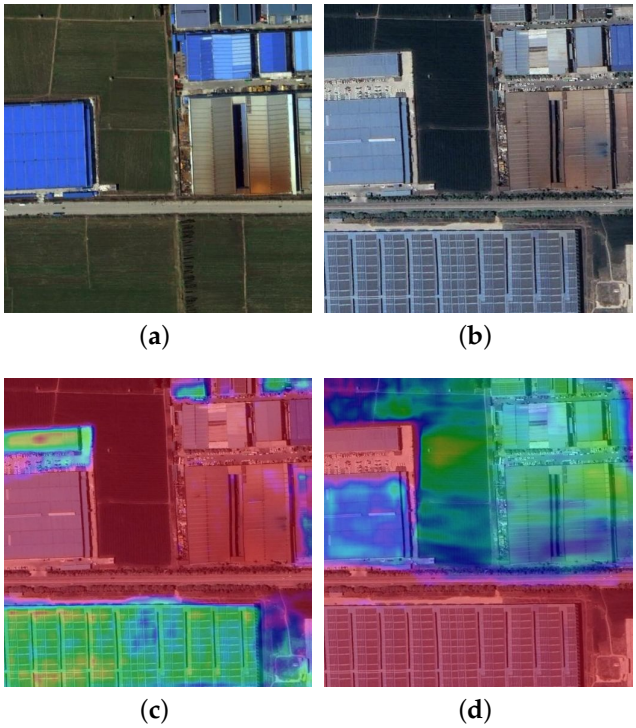

**Figure 6.** Heat maps of difference module and assimilation module. (**a**) Bitemporal Google Earth image. (**b**) Bitemporal Google Earth image. (**c**) Differential module heat map. (**d**) Assimilation module heat map.

### *2.3. Fusion Module*

The proposed algorithm consists of three branches: a main network and two auxiliary networks; if they are fused directly, the data information will inevitably affect each other. To solve this problem, a fusion module is added at the end of the network to process these three information flows.

As shown in Figure 7, $w$ and $h$ represent the width and height of the feature map, respectively, and the information of different branches is first stacked to obtain a feature layer $U$. Its equation can be written as

$$U' = \zeta(B(f^{3\times3}(U))). \tag{2}$$

$U'$ is obtained by convolutions. $f^{3\times3}$ denotes the convolution kernel of 3 × 3, $B$ means batch normalization and the calculation process can be described as

$$\mu_\beta = \frac{1}{m}\sum_{i=1}^{m} x_i, \tag{3}$$

$$\sigma^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2, \tag{4}$$

$$\bar{x}_i = \frac{x_i + \mu_\beta}{\sqrt{\sigma^2 + \varepsilon}}, \tag{5}$$

$$y_i = \gamma\bar{x}_i + \beta. \tag{6}$$

$\mu_\beta$ in Equation (3) denotes the average of the output data in the previous layer; $x_i$ and $m$ denote the parameters and the number of parameters in previous feature maps, respectively. The standard deviation $\sigma^2$ of the output data of previous layers can be acquired in Equation (4). In Equation (5), $\bar{x}_i$ is a normalized value and $\varepsilon$ is a very small value in order to avoid zero in the denominator. Finally, $y_i$ is batch normalized value in Equation (6). To make the normal distribution of each output data different, $\gamma$ and $\beta$ are introduced. $\zeta$ in Equation (2) denotes the function of ReLU, it can be written as

$$f(x) = max(0, x). \tag{7}$$

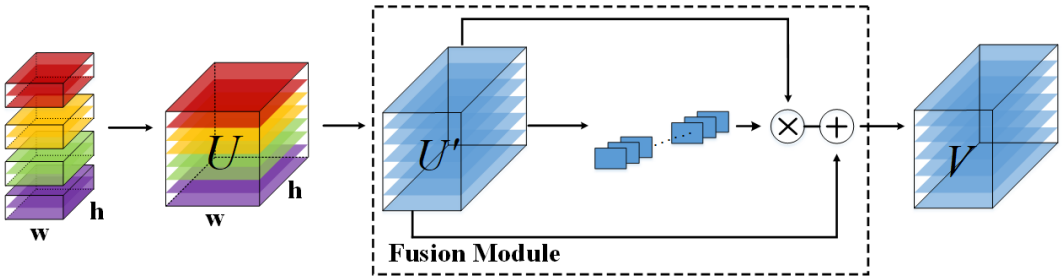


**Figure 7.** Fusion module.

In order to let the model better obtain global information and use vectors to guide feature learning, an attention mechanism is added to reorganize the data in the feature maps:

$$t = M_2(U') = S(f^{1\times1}(\zeta(f^{1\times1}(F_{gp}(U'))))), \tag{8}$$

$F_{gp}$ means global average pooling, it adds up all pixel values of the feature map to get average values, and then it uses these average values to represent the corresponding feature map. Its main function is to replace the fully connected layers, reducing the number of parameters and calculation, improving the robustness of the model. $S$ in Equation (8) denotes the function of Sigmoid, which can normalize values between 0 and 1, and its function can be written as

$$S(x) = \frac{1}{1 + e^{-x}}, \tag{9}$$

where $x$ represents the input to Sigmoid. $t \in \mathbb{R}^{1\times1\times C_n}$ in Equation (8), $C_n$ denotes the number of categories. In the change detection task, $C_n$ is set to 2 because only the changed areas are different from the unchanged. Finally, the number of corresponding channels in $t$ is multiplied with $U'$ to assign weight to the parameters, the obtained characteristic layer and $U'$ are added to obtain the final output $V$:

$$V = tU' + U'. \tag{10}$$

### *2.4. Build Loss Function*

Two auxiliary loss functions are added to the total output loss function of the model to improve the network performance and train each component better in the model. These two auxiliary loss functions enhance trainings of the assimilation module and difference module. The Softmax function is selected as the loss function because the output of the model divides the image into two categories: the changed and the unchanged areas, and its formula is shown in Equation (11). *loss* is the sum of the lost values with the real data for two branches, $p_i$ is the output prediction of the network, $q_k$ is the real value, and $N$ is the number of samples.

$$loss = \frac{1}{N}\sum_{i}^{N} -\log(\frac{e^{p_i}}{\sum_k e^{q_k}}), \tag{11}$$

$$L = \sum_{j=1}^{2} loss_j + loss_{\mathrm{m}}. \tag{12}$$

$L$ is the joint loss function in the network. $loss_j$ denotes the corresponding loss output of the assimilation module ($j = 1$) and the difference module ($j = 2$). $loss_{\mathrm{m}}$ denotes the loss value output from the end of main module's network.

## 3. Dataset

A dataset was created to train and verify the validity of the proposed model. Compared with other published datasets, the dataset used in this work contains more details of the objects and their environments due to a lower view altitude. Therefore, the requirements for the algorithm are higher. This dataset contains 3420 pairs of Google Earth images, the size of each image is 512 × 512, and the corresponding area is approximately $300^2$ square meters. These images from Google Earth (GE) include shots taken at different times from 2010 to 2019. GE is a virtual earth software developed by Google, which allows users to browse high-definition images of different time periods around the world for free. The satellite image of GE is not a single data source, but an integration of multiple satellite images, mainly from DigitalGlobe's QuickBird commercial satellite and EarthSat (images mostly from landsat-7). GE has a resolution of at least 100 m, usually 30 m. The view altitude is ~15 km. However, for big cities, famous scenic spots, or other hot spots, high-precision images with a resolution of 1 m or 0.6 m will be provided, with the viewing altitude of approximately 500 m or 350 m, respectively. The images of these places have been contained in the dataset. When collecting data, the bitemporal Google Earth images have been registered. The dataset has the following characteristics. (1) The proportion of positive to negative samples varies greatly. There are only small parts of changed areas in a complete Google Earth image. (2) The buildings in the images have some positional deviations due to different shooting angles. (3) Numerous scenes in changed areas, including farmland, roads, high-speed rail, factories, buildings, and mining areas. (4) Seasonal factors have great influence. For example, deciduous broad-leaved forests with seasonal changes have significant differences in winter and summer. (5) In order to obtain the most accurate changed area information, all ground truth were marked manually.

Some labeled images are shown in Figure 8, and these data contain changed areas of different proportions. The dataset contains a small number of bitemporal Google Earth images with unchanged areas, in order to evaluate the performance of the algorithm more thoroughly, as shown in Figure 8a.

Analyzing the proportion of the changed areas of each image in the dataset could have a better overall understanding of it. The graph after statistics is shown in Figure 9, where the abscissa is the proportion of the changed areas to the whole, and the ordinate is the number of images in this proportion. From this figure, the proportion of changed areas of most data is between 5% and 50%. There are 337 positive samples with the proportion of 0–2%, of which 242 0–1% and 95 1–2%. Increasing the imbalance between samples can better distinguish good from bad models.

In order to make the model more robust, it is inevitable to increase the amount of dataset. For example, the CNN is sensitive to translation, rotation flip, and so forth. Sometimes, the translation of a pixel can even lead to wrong classification. Rotation and scaling are even more devastating for the image position information. Therefore, data augmentation is very important and necessary. In order to solve the above problems, the general method is data augmentation; appropriate translation; inversion (horizontal, vertical); rotation; random shear; changing hue, saturation, and value (HSV) saturation; and so forth. The essence of data augmentation is to increase the data by introducing prior knowledge so as to improve the generalization ability of the model. These methods can effectively improve the generalization performance of the model. The specific parameter settings are shown in Table 1 and images after the data enhancement are shown in Figure 10. The translation distance does not exceed 20% of the length and width of the original image; HSV saturation and random rotation share similar settings. Each image has a 50% probability of flipping horizontally and vertically, respectively.

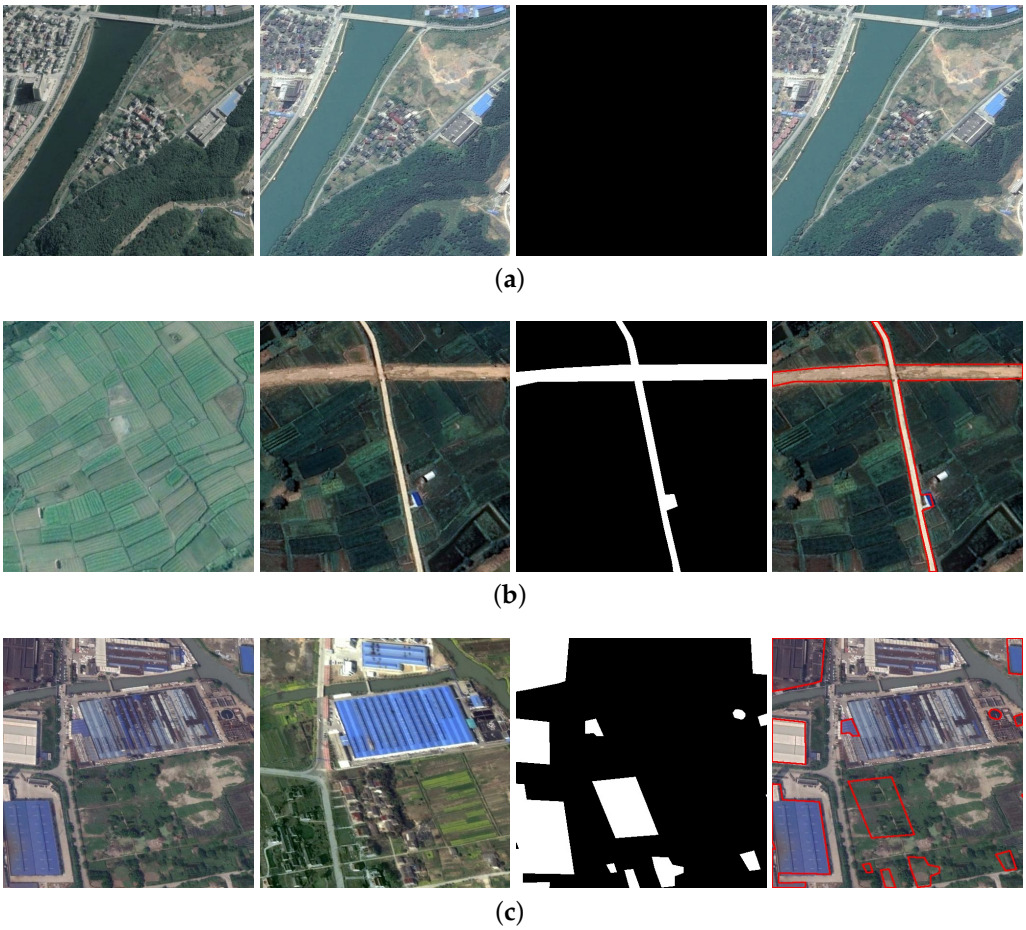

**Figure 8.** Display of some bitemporal Google Earth images in the dataset. In each line, from left to right, the first two Google Earth images are at the same place of different times. The third image is the ground truth, in which the black areas represent the unchanged, and the white represent the changed. The red lines in the fourth image are the parts of the changed areas. Panels (**a**–**c**) represent areas of changed in different proportions, respectively: (**a**) unchanged areas, (**b**) 9% of changed areas, and (**c**) 20% of changed areas.

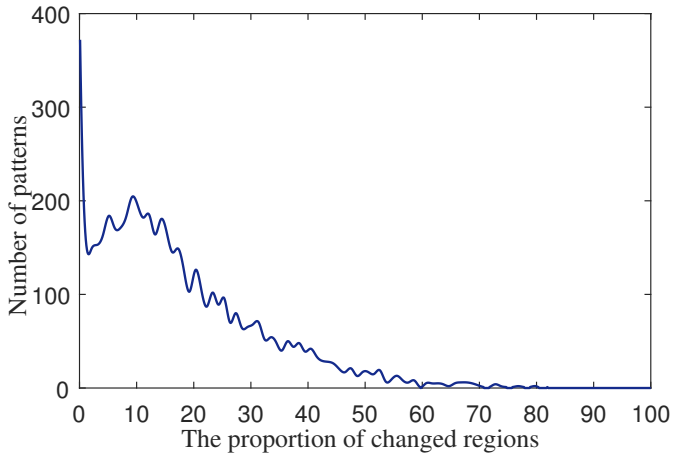


**Figure 9.** Dataset analysis. The ratio of the changed areas in each image in the data set to the whole.

**Table 1.** The parameters of data augmentation.

| Method | Parameter |
|---|---|
| Translation | ±20% |
| HSV Saturation | ±50% |
| Random Rotation | ±10° |
| Random Horizontal Flip | 50% |
| Random Vertical Flip | 50% |

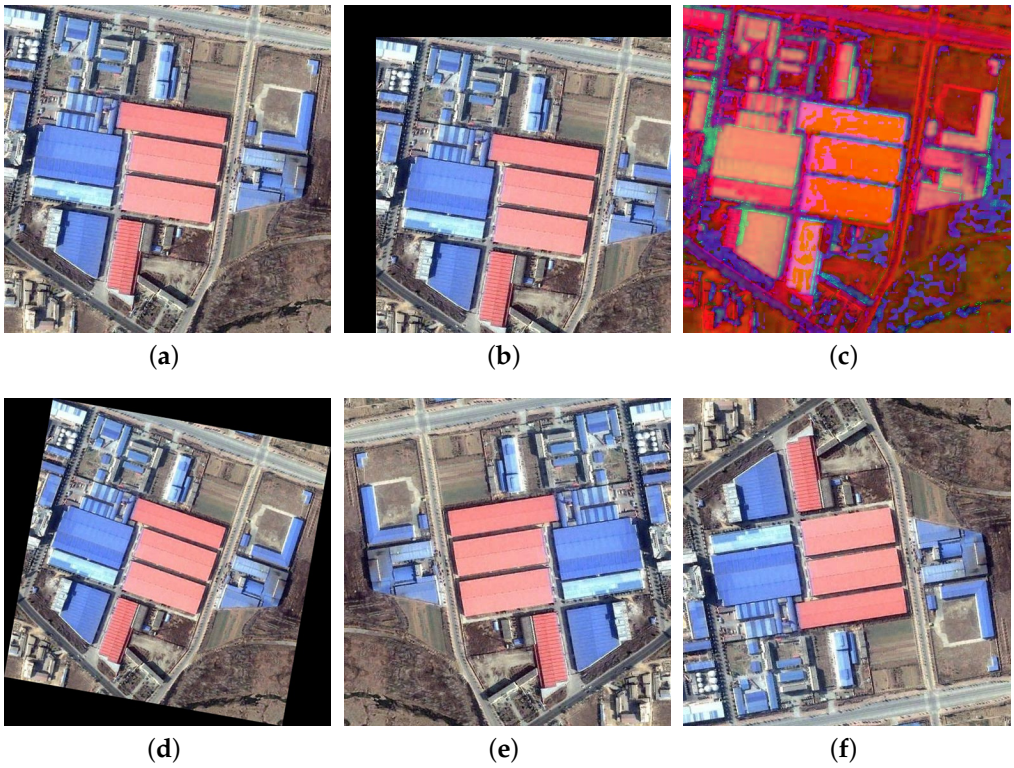

**Figure 10.** Diagrams of different data augmentation methods. (**a**) The original image; (**b**) the original image after being translated; panel (**c**) is the HSV transformed image; (**d**) the original image rotated by 10 degrees; panels (**e**) and (**f**) are images after horizontal and vertical rotation, respectively.

## 4. Experiment

To verify the effectiveness of the proposed algorithm, three experiments are designed. The quantitative indexes in these three experiments are the pixel-accuracy (PA), the recall (RC), the precision (PR), and the mean intersection over union (MIoU). PA is the simplest evaluation index, indicating the proportion of correctly marked pixels to the total pixels. RC is the ratio of all correctly predicted changed areas in the prediction map to the changed areas in the ground truth, and the PR is the ratio of correctly predicted changed areas to all predicted changed areas in the prediction map. MIoU is to calculate the ratios of intersection and union of two sets, and to represent the changed and unchanged areas in a change detection task. A combination of different evaluation indicators can better evaluate the comprehensive performance of the model.

### *4.1. Experimental Preparation and Super Parameter Setting*

A very high learning rate may cause the loss function to miss the global optimum, but if the learning rate is too slow, the loss function change slowly, which will greatly increases the convergence complexity of the network, and will be be trapped in the local minimum or saddle point. Therefore, the learning rate is generally set to 0.001 at the initial stage of training, and the learning rate decreases

by 25% every 20 iterations. At the same time, setting up a warm-up can help slow down the overfitting of minibatch in the initial stage of the model and keep the deep stability of the model. The learning rate of each iteration is set as

$$\eta_{rn} = \frac{n}{5}\eta_r, \tag{13}$$

where $n$ is the n'th warm-up training. The batch size is set to four. Root mean square prop (RMSprop) is chosen for the optimization algorithm; its advantage is that the AdaGrad is modified into an exponential weighted moving average, making it more effective in a non-convex setting. The algorithm optimization strategy can be expressed as

$$\begin{cases} E\left[g^2\right]_t = \alpha E\left[g^2\right]_{t-1} + (1-\alpha)g_t^2 \\ W_{t+1} = W_t - \frac{\eta_0}{\sqrt{E[g^2]_t + \varepsilon}} \otimes g_t \end{cases}, \tag{14}$$

$W_t$ denotes the parameters of the $t$-th iterative model and $g_t = \Delta J(W_t)$ denotes the gradient under the cost function after $t$ iterations with respect to $W$. $E[g^2]$ is the squared mean of the gradient of the first $t$ times. $\alpha$ denotes the momentum (usually set to 0.9) and $\eta_0$ is the global initial learning rate. $\varepsilon$ is a very small number (usually set to $10^{-8}$) in order to avoid a denominator of 0. The weighted average is adopted to avoid the problem of lower and lower learning rate, and the learning rate can be adjusted adaptively. The problem of the sharp decrease of AdaGrad gradient could be overcome and excellent learning rate adaptability could be achieved. In particular, it performs better under nonstationary target functions than stochastic gradient descent (SGD), Momentum, and AdaGrad.

### *4.2. Selection of Backbone Network*

The purpose of the first experiment is to study the performance of CNNs with different depths in the proposed algorithm. The experimental results are shown in Table 2 and the best results are shown in bold. The network is consisted of three parts: a main module and two auxiliary modules. All training parameters are set to be the same. The number following the method name indicates the number of convolutional network layers. For example, TCDNet_18 indicates that ResNet18 is used in the main network.

**Table 2.** Influence of convolutional neural networks (CNNs) with different depths on algorithm performance (bold represents the best result).

| Method | PA (%) | RC (%) | PR (%) | MIoU (%) |
|---|---|---|---|---|
| TCDNet_18 | 95.22 | 70.81 | 79.19 | 81.23 |
| TCDNet_34 | 95.39 | **71.18** | **80.64** | **81.95** |
| TCDNet_50 | **95.43** | 70.12 | 78.58 | 81.40 |

The indexes with the best performance are shown in bold. As seen in Table 2, TCDNet_34 has the best comprehensive performance. For different tasks, more convolutional layers is not always better, especially for change detection tasks. This task can be regarded as a two-category semantic segmentation task, and the number of categories is relatively simple. As a result, as the network deepened, the gradient disappeared, and the accuracy saturation even decreased. According to the above analysis, ResNet34 is used as the backbone network in this work.

### *4.3. Ablation Experiment*

The second experiment focuses on ablation research. Compared with complex CNNs, the whole network can be better understood by deleting some modules. Understanding the causality of a system is the most direct method to generate reliable knowledge and verify whether different modules can effectively improve the model accuracy. The computation time of all algorithms is performed on the CPU (AMD Ryzen 5 1600X).

In Table 3, the main module (MM) uses the ResNet34 as the backbone network. The training parameters of all models are set to be the same, and the indexes with the best performance are shown in bold. According to the information in Table 3.The best results of different evaluation indexes are shown in bold. The difference module (DM) and assimilation module (AM) can indeed help the main network improve its performance. The RC index of the model with the main module was only 64.99%, while adding two auxiliary modules increase the index by 6.19%. There are also 0.76%, 0.82%, and 3.08% increases in PA, PR, and MIoU, respectively, because these two auxiliary modules can make up lost details of the main module. Through the cooperative detection of the changed and unchanged areas by the difference and assimilation modules, some noise information is effectively removed, and the odds of incorrect classification is reduced. In terms of calculation time, the two auxiliary modules only slightly increase the detection time, but the accuracy is greatly improved.

**Table 3.** Ablation experiment of TCDNet (bold represents the best result).

| Method | PA (%) | RC (%) | PR (%) | MIoU (%) | Time (s) |
|---|---|---|---|---|---|
| MM | 94.63 | 64.99 | 79.82 | 78.87 | **2.44** |
| MM + AM | 94.88 | 66.77 | 78.50 | 79.70 | 3.02 |
| MM + DM | 95.04 | 68.23 | 78.28 | 80.37 | 3.02 |
| MM + AM + DM | **95.39** | **71.18** | **80.64** | **81.95** | 3.57 |

### *4.4. Comparative Experiments of Different Algorithms*

In this experiment, the algorithm proposed in this paper uses the ResNet34 as its main network and two auxiliary modules to assist the main network in change detection. Because the proposed model takes semantic segmentation as its core idea, semantic segmentation network (SegNet) [37], FCN8S [28], High-Resolution Net (HRNet) [39], UNet [40], and UNet_att [41] are selected as comparison algorithms, and IR-MAD [15] and PCA-means [17] are used as traditional comparison algorithms. All the dataset used by the algorithms in Table 4 were introduced in Section 3. The experimental results are shown in Table 4, and the best results of different evaluation indexes are shown in bold.

**Table 4.** Comparison between different algorithms (bold represents the best result).

| Method | PA (%) | RC (%) | PR (%) | MIoU (%) | Time (s) |
|---|---|---|---|---|---|
| IR-MAD | 79.59 | 11.19 | 13.30 | 42.89 | **1.59** |
| PCA-Means | 86.00 | 10.74 | 33.11 | 47.32 | 9.35 |
| SegNet | 92.94 | 58.08 | 71.67 | 73.29 | 14.54 |
| FCN8S | 93.38 | 64.41 | 70.76 | 75.35 | 7.32 |
| HRNet | 94.17 | 65.17 | 75.84 | 77.55 | 4.53 |
| UNet | 94.26 | 65.30 | 76.31 | 78.04 | 12.41 |
| UNet_att | 94.66 | 68.18 | 78.82 | 79.69 | 24.29 |
| TCDNet_34 | **95.39** | **71.18** | **80.64** | **81.95** | 3.57 |

As seen in Table 4, the quantitative indexes of the IR-MAD and the PCA-Means fall behind those of deep learning algorithms. Because the buildings are photographed at different times with angle deviations, they could not be matched well. Additionally, due to the influences of seasons and illumination, the colors of object surfaces differed, which lead to the low precision of traditional algorithms. In particular, there is a large gap between RC and PR. Furthermore, because a map only has a small part of changed areas, most of the maps are unchanged areas. Although the result maps of the two traditional algorithms are not satisfactory, the PA index is approximately 80. The proposed algorithm optimizes the change detection task. The main module obtains the overall information of the image pair, and the two auxiliary modules focus on the changed and unchanged areas to improve the details of the prediction map. Its RC, PR, and MIoU reach 71.18%, 80.64%, and 81.95%, respectively. In terms of calculation time, IR-MAD has the fastest calculation speed, but its accuracy is not good. The TCDNet_34 only takes 3.57 s to detect a pair of images, which is similar to the traditional learning

method, and it is the fastest in deep learning methods. UNet_att has a high accuracy, but its running time is 6.8 times that of TCDNet_34.

Figure 11 is a change detection effect diagram of different algorithms. The images in Figure 11a,b are taken at different times at the same location, and the image in Figure 11c is the ground truth; in this picture, white indicates changed areas and black indicates unchanged areas, and the changed areas are marked with red boxes in Figure 11d. Figure 11e–k displays the prediction maps of seven different comparison algorithms, and Figure 11l is the result of the proposed method. From these eight change detection maps, the two traditional algorithms can not deal with complex changed areas and only compare image pairs on basic pixels. The texture information of objects and the surrounding environment information of pixels are not fully utilized; therefore, their results are not satisfactory. The other five deep learning comparison algorithms barely identify the changed areas, but there are still some noise and some gaps compared with results in Figure 11l. The algorithm proposed carries out multidimensional detection through three branches and relies on the main module to determine locations of changed areas roughly. Then, it carries out special detections for the changed and unchanged areas through two auxiliary modules. This cooperative method effectively reduces noise and improves the accuracy of the model.

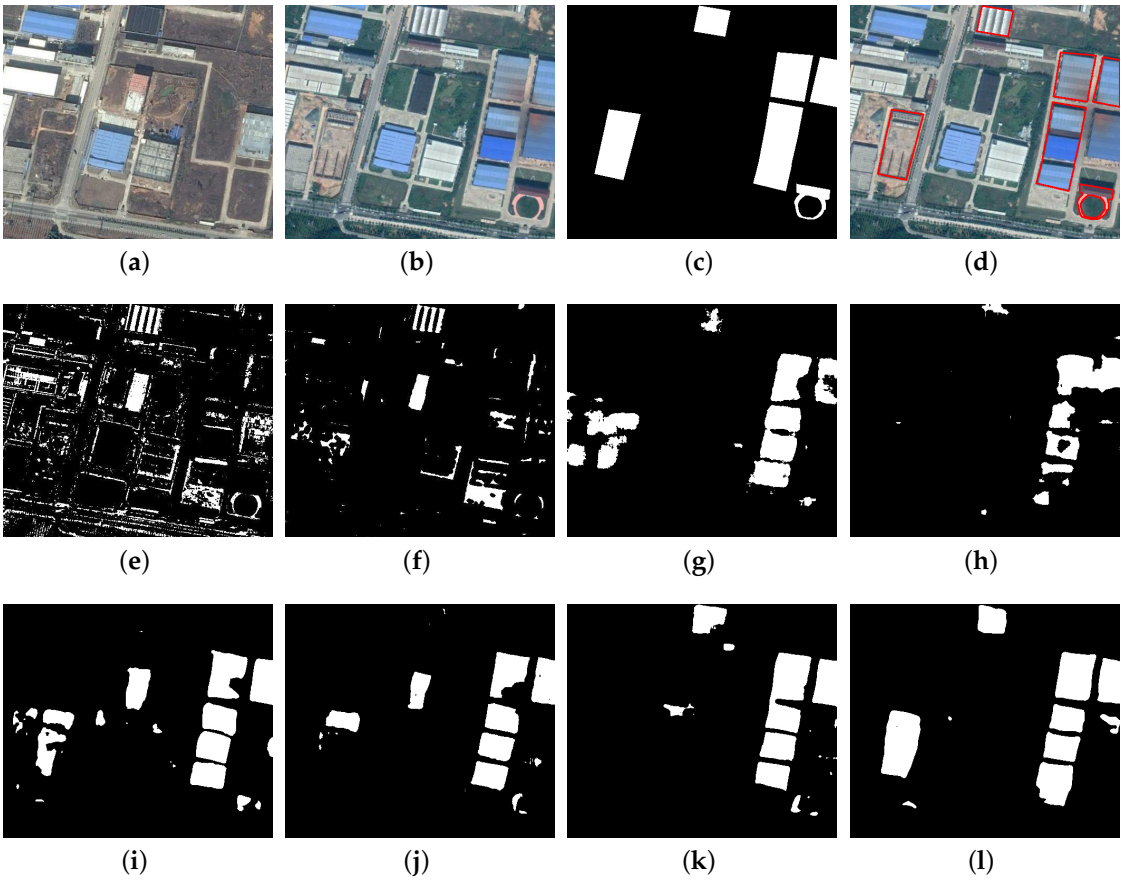


**Figure 11.** Comparison of prediction maps for different algorithms. Panels (**a**,**b**) are Google Earth images of the same place at different times; panel (**c**) is the ground truth, in which the black areas represent the unchanged, and the white areas represent the changed. The changed areas are framed by red lines in panel (**d**). Panels (**e**–**l**) are the prediction maps of different algorithms, which are, respectively, (**e**) IR-MAD, (**f**) PCA-means, (**g**) SegNet, (**h**) FCN8S, (**i**) HRNet, (**j**) UNet, (**k**) UNet_att, and (**l**) TCDNet_34.

Figure 12 is another set of comparison diagrams, from which we can better distinguish the changed and unchanged areas in detail. From the three maps Figure 12a–c, it can be seen that there

is an unchanged area between the factories. How to identify this part accurately will further test the accuracy of the algorithm. Although Figure 12g can roughly detect the changed areas, its edges are very rough; Figure 12h,k has some improvements on the edges, but they are not able to detect the unchanged areas between different changed areas. Figure 12i,j shows great improvements in details, but there are still interconnections between areas and the details are not accurate enough. Figure 12l is the TCDNet model. The algorithm firstly locates the changed areas roughly through the main module, and then repairs the details through the difference and assimilation modules. Compared with other algorithms, the accuracy and robustness of the algorithm are greatly improved, it can accurately identify the changed areas and reduce the occurrence of false detection.

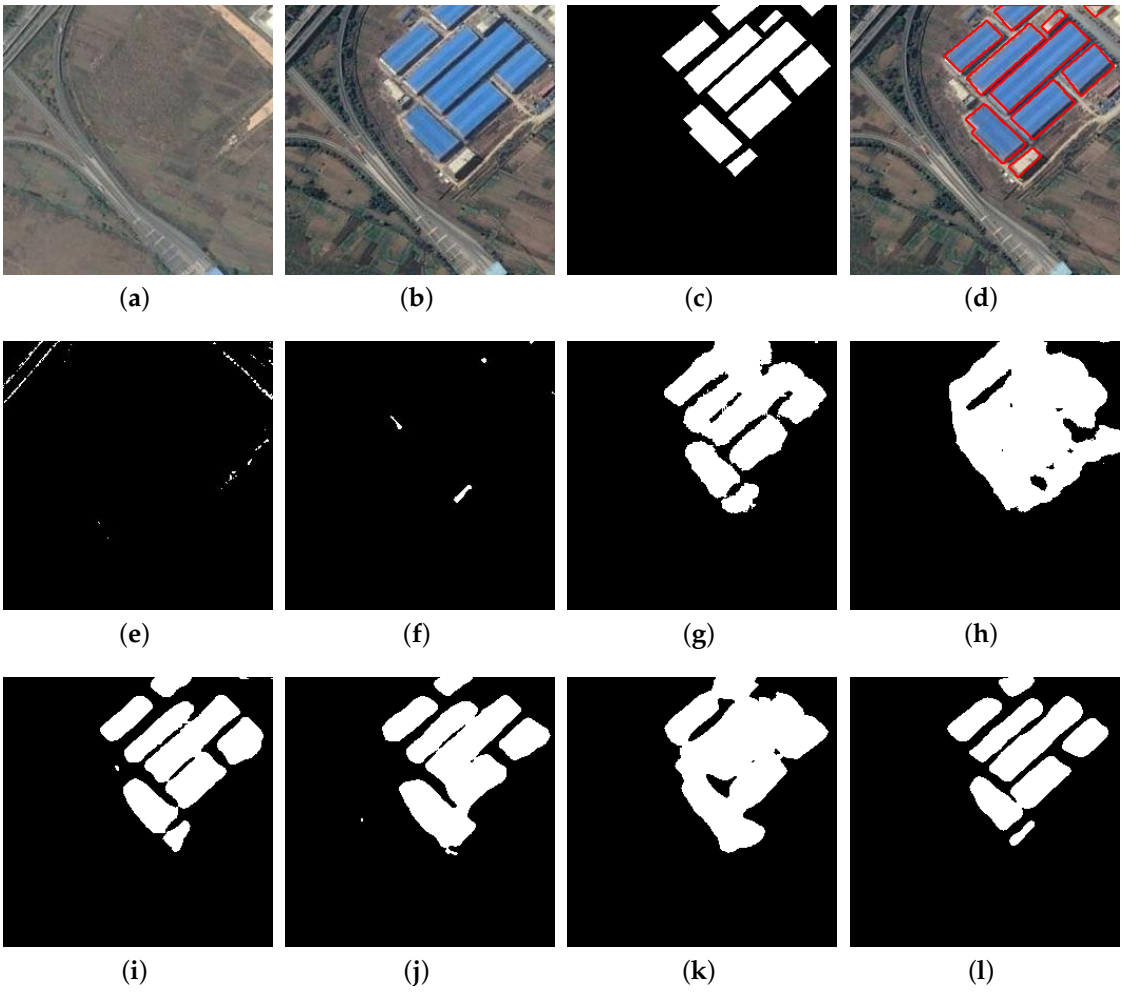


**Figure 12.** Comparison of prediction maps for different algorithms. Panels (**a**,**b**) are Google Earth images of the same place at different times. Panel (**c**) is the ground truth, in which the black areas represent the unchanged, and the white areas represent the changed. The changed areas are framed by red lines in panel (**d**). Panels (**e**–**l**) are the prediction maps of different algorithms, which are, respectively, (**e**) IR-MAD, (**f**) PCA-means, (**g**) SegNet, (**h**) FCN8S, (**i**) HRNet, (**j**) UNet, (**k**) UNet_att, and (**l**) TCDNet_34.

## 5. Conclusions

In this paper, a general framework called the TCDNet is proposed. It consists of three parts: one main module and two auxiliary modules, a difference module, and an assimilation module. These three branches output the final change detection map through a fusion module. First, the main module extracts different levels of information from bitemporal Google Earth image pairs, roughly distinguishes the changed and unchanged areas, and then the method combines the outputs of

these two auxiliary modules to refine the change detection map to remove redundant noise. Second, an independent loss function is added to guide weight updates in the training phase to better update auxiliary module weights. Compared with the other algorithms, the proposed method shows better generalization and robustness, and it is superior to the other algorithms in various indexes. The deep learning method depends on the number of categories in the dataset, the more categories, the better. If the type of changed area is not included in the training data, it will have a certain influence on the detection results. Plus, this paper only deals with changed areas of Google Earth images, future research will focus on the identification of categories in changed areas. At the same time, due to the huge cost of creating dataset, we will focus on the application of semisupervised and unsupervised methods in change detection in the future.

**Author Contributions:** Conceptualization, M.X., J.Q., and Y.Z.; methodology, M.X. and J.Q.; software, J.Q.; validation, J.Q. and Y.Z.; formal analysis, M.X., J.L., and Y.X.; investigation, Y.Z. and Y.X.; resources, M.X. and Y.Z.; data curation, M.X. and Y.X.; writing—original draft preparation, J.Q.; writing—review and editing, M.X. and J.L.; visualization, J.Q.; supervision, M.X. and Y.Z.; project administration, M.X.; funding acquisition, M.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of PR China of grant numbers 41875027, 61773219, and 41661144039.

**Conflicts of Interest:** The authors declare no conflicts of interest.

**Data Availability:** The data and the code of this study are available from the corresponding author upon request. (xiamin@nuist.edu.cn).

## References

1. Tison, C.; Nicolas, J.M.; Tupin, F.; Maître, H. A new statistical model for Markovian classification of urban areas in high-resolution SAR images. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 2046–2057. [CrossRef]. [CrossRef]
2. Yokoya, N.; Chan, J.C.W.; Segl, K. Potential of resolution-enhanced hyperspectral data for mineral mapping using simulated EnMAP and Sentinel-2 images. *Remote Sens.* **2016**, *8*, 172. [CrossRef]. [CrossRef]
3. Tian, B.; Wang, L.; Kashiwaya, K.; Koike, K. Combination of well-logging temperature and thermal remote sensing for characterization of geothermal resources in Hokkaido, northern Japan. *Remote Sens.* **2015**, *7*, 2647–2667. [CrossRef]. [CrossRef]
4. Mucia, A.; Bonan, B.; Zheng, Y.; Albergel, C.; Calvet, J.C. From Monitoring to Forecasting Land Surface Conditions Using a Land Data Assimilation System: Application over the Contiguous United States. *Remote Sens.* **2020**, *12*, 2020. [CrossRef]. [CrossRef]
5. Wieland, M.; Martinis, S. A modular processing chain for automated flood monitoring from multi-spectral satellite data. *Remote Sens.* **2019**, *11*, 2330. [CrossRef]. [CrossRef]
6. Singh, A. Review article digital change detection techniques using remotely-sensed data. *Int. J. Remote Sens.* **1989**, *10*, 989–1003. [CrossRef]. [CrossRef]
7. Mou, L.; Bruzzone, L.; Zhu, X.X. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 924–935. [CrossRef]. [CrossRef]
8. Bovolo, F.; Bruzzone, L.; Solano-Correa, Y. Multitemporal Analysis of Remotely Sensed Image Data. In *Comprehensive Remote Sensing*; Elsevier: Oxford, UK, 2017; pp. 156–185. [CrossRef].
9. Saha, S.; Solano-Correa, Y.T.; Bovolo, F.; Bruzzone, L. Unsupervised Deep Transfer Learning-Based Change Detection for HR Multispectral Images. *IEEE Geosci. Remote Sens. Lett.* **2020**. [CrossRef]. [CrossRef]
10. Peng, D.; Zhang, Y.; Guan, H. End-to-end change detection for high resolution satellite images using improved unet++. *Remote Sens.* **2019**, *11*, 1382. [CrossRef]. [CrossRef]
11. Ball, J.E.; Anderson, D.T.; Chan, C.S. Comprehensive survey of deep learning in remote sensing: Theories, tools, and challenges for the community. *J. Appl. Remote Sens.* **2017**, *11*, 042609. [CrossRef]. [CrossRef]
12. Liu, J.; Gong, M.; Qin, K.; Zhang, P. A deep convolutional coupling network for change detection based on heterogeneous optical and radar images. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *29*, 545–559. [CrossRef]. [CrossRef]

13. Hussain, M.; Chen, D.; Cheng, A.; Wei, H.; Stanley, D. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS J. Photogramm. Remote Sens.* **2013**, *80*, 91–106. [CrossRef]. [CrossRef]
14. Celik, T. Unsupervised change detection in satellite images using principal component analysis and *k*-means clustering. *IEEE Geosci. Remote Sens. Lett.* **2009**, *6*, 772–776. [CrossRef]. [CrossRef]
15. Nielsen, A.A. The regularized iteratively reweighted MAD method for change detection in multi-and hyperspectral data. *IEEE Trans. Image Process.* **2007**, *16*, 463–478. [CrossRef]. [CrossRef] [PubMed]
16. Zhang, H.; Gong, M.; Zhang, P.; Su, L.; Shi, J. Feature-level change detection using deep representation and feature change analysis for multispectral imagery. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1666–1670. [CrossRef]. [CrossRef]
17. Deng, J.; Wang, K.; Deng, Y.; Qi, G. PCA-based land-use change detection and analysis using multitemporal and multisensor satellite data. *Int. J. Remote Sens.* **2008**, *29*, 4823–4838. [CrossRef]. [CrossRef]
18. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B* **1977**, *39*, 1–22. [CrossRef].
19. Malila, W.A. Change Vector Analysis: An Approach for Detecting Forest Changes with Landsat. In Proceedings of the 6th Annual Symposium on Machine Processing of Remotely Sensed Data, West Lafayette, IN, USA, 3–6 June 1980; pp.326–335.
20. Sesnie, S.E.; Gessler, P.E.; Finegan, B.; Thessler, S. Integrating Landsat TM and SRTM-DEM derived variables with decision trees for habitat classification and change detection in complex neotropical environments. *Remote Sens. Environ.* **2008**, *112*, 2145–2159. [CrossRef]. [CrossRef]
21. Xia, M.; Xu, Y.; Wang, K.; Zhang, X. Dilated residual attention network for load disaggregation. *Neural Comput. Appl.* **2019**, *31*, 8931–8953. [CrossRef]. [CrossRef]
22. Xia, M.; Wang, K.; Zhang, X.; Xu, Y. Non-intrusive load disaggregation based on deep dilated residual network. *Electr. Power Syst. Res.* **2019**, *170*, 277–285. [CrossRef]. [CrossRef]
23. Xia, M.; Zhang, X.; Weng, L.; Xu, Y. Multi-Stage Feature Constraints Learning for Age Estimation. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2417–2428. [CrossRef]. [CrossRef]
24. Wang, Q.; Yuan, Z.; Du, Q.; Li, X. GETNET: A general end-to-end 2-D CNN framework for hyperspectral image change detection. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 3–13. [CrossRef]. [CrossRef]
25. Yang, M.; Jiao, L.; Liu, F.; Hou, B.; Yang, S. Transferred Deep Learning-Based Change Detection in Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6960–6973. [CrossRef]. [CrossRef]
26. Wang, Q.; Zhang, X.; Chen, G.; Dai, F.; Gong, Y.; Zhu, K. Change detection based on Faster R-CNN for high-resolution remote sensing images. *Remote Sens. Lett.* **2018**, *9*, 923–932. [CrossRef]. [CrossRef]
27. Zhan, Y.; Fu, K.; Yan, M.; Sun, X.; Wang, H.; Qiu, X. Change detection based on deep siamese convolutional network for optical aerial images. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 1845–1849. [CrossRef]. [CrossRef]
28. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [CrossRef].
29. Zhang, C.; Wei, S.; Ji, S.; Lu, M. Detecting large-scale urban land cover changes from very high resolution remote sensing images using CNN-based classification. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 189. [CrossRef]. [CrossRef]
30. Arabi, M.E.A.; Karoui, M.S.; Djerriri, K. Optical Remote Sensing Change Detection Through Deep Siamese Network. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 5041–5044. [CrossRef].
31. Xia, M.; Qian, J.; Zhang, X.; Liu, J.; Xu, Y. River segmentation based on separable attention residual network. *J. Appl. Remote Sens.* **2019**, *14*, 032602. [CrossRef]. [CrossRef]
32. Zhang, M.; Xu, G.; Chen, K.; Yan, M.; Sun, X. Triplet-based semantic relation learning for aerial remote sensing image change detection. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 266–270. [CrossRef]. [CrossRef]
33. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef].
34. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556 [CrossRef].

35. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef].
36. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708. [CrossRef].
37. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef]. [CrossRef]
38. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122. [CrossRef].
39. 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. [CrossRef].
40. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Cambridge, UK, 19–22 September 2015; pp. 234–241. [CrossRef].
41. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.; Heinrich, M.; Misawa, K.; Mori, K.; McDonagh, S.; Hammerla, N.Y.; Kainz, B.; et al. Attention u-net: Learning where to look for the pancreas. *arXiv* **2018**, arXiv:1804.03999. [CrossRef].