ISSN: 2508-7894 © 2021 KODISA & KJAI.
KJAI website: http://www.kjai.or.kr
DOI: http://dx.doi.org/10.24225/kjai.2021.9.2.7

Implementation of CNN-based Masking Algorithm for Post Processing of Aerial Image[*]

Eunsoo CHOI[1], Zhixuan QUAN [2], Sangwoo JUNG [3]

Received: August 10, 2021. Revised: August 21, 2021. Accepted: December 05, 2021.

Abstract

Purpose: To solve urban problems, empirical research is being actively conducted to implement a smart city based on various ICT technologies, and digital twin technology is needed to effectively implement a smart city. A digital twin is essential for the realization of a smart city. A digital twin is a virtual environment that intuitively visualizes multidimensional data in the real world based on 3D. Digital twin is implemented on the premise of the convergence of GIS and BIM, and in particular, a lot of time is invested in data pre-processing and labeling in the data construction process. In digital twin, data quality is prioritized for consistency with reality, but there is a limit to data inspection with the naked eye. Therefore, in order to improve the required time and quality of digital twin construction, it was attempted to detect a building using Mask R-CNN, a deep learning-based masking algorithm for aerial images. If the results of this study are advanced and used to build digital twin data, it is thought that a high-quality smart city can be realized.

Keywords : Aerial Image, Mask R-CNN, CNN, Digital Twin, Smart City

Major Classification Code : Artificial Intelligence, Deep Learning

1. Introduction

In the era of the 4th industrial revolution, research is being conducted focusing on the use of smart cities to solve urban problems (Park, 2019). Representatively, he is researching problem solving using ICT technologies such as ICBMA (IoT, Cloud, Bigdata, Mobile, AI), and further devising a local reality platform that can discover and develop new growth engines (Lee, 2017).

Digital twin is needed to implement effective monitoring, analysis, and artificial intelligence through smart cities. The need for digital twin has emerged in the past U-city research, but it was difficult to implement due to the lack of hardware infrastructure performance at the time. However, based on the introduction of high-performance GPUs (Graphics Processing Unit), high-performance sensors, and 5G, the reality of building digital twins is re-emerging (Rasheed, San, & Kvamsdal, 2020). A digital twin is a virtual environment that is visualized based on 3D multidimensional data of the real world and refers to a virtual environment created identical to the real world (Lee, 2021). The digital twin is implemented on the premise of the convergence of GIS data and BIM data.

[*] This Work was Supported by the Research Grant of the KODISA Scholarship Foundation in 2021.

[1] First Author, Assistant Manager, Smart Geospatial Research Center, ALLFORLAND Co.Ltd, Korea,
Email: ces951030@all4land.com

[2] Corresponding Author, Professor, Department of Alternative Medicine, Kwangju Women's University, Korea,
Email: 3700qzx@gmail.com

[3] Co-Author, General Manager, ALLFORLAND Co.Ltd, Korea,
Email: cki723@all4land.com

In particular, a lot of time is spent on data preprocessing and labeling in GIS and BIM construction. Therefore, the quality and duration of digital twin construction is determined by the quality and quantity of data.

A representative object to construct a digital twin consists of 2D and 3D objects. Specifically, 3D objects include indoor, outdoor, building, road, vegetation, and water facility, and this study was conducted on a building (Kim, Lee & Choi, 2020).

Therefore, this study tried to detect houses using Mask R-CNN, a deep learning-based masking algorithm, targeting deep learning-based aerial images to improve the quality and time required for digital twin construction.

To implement Mask R-CNN, data collection, data augmentation, data labeling, model training and validation were performed. Finally, the implications of the verification results are described.

2. Literature Review

2.1. Machine Learning

Machine learning is a technology that develops algorithms and techniques that enable computers to learn, focusing on representation and generalization. Basically, it can be said that data is analyzed using algorithms, learning through analysis, and judgment or prediction based on learning. Machine learning is divided into supervised learning, unsupervised learning, and reinforcement learning, respectively, depending on the case with and without labels. Classification and regression algorithms are representative methods of supervised learning (An, Yeo, & Kang, 2021).

A representative classification algorithm is Multiclass Decision Tree, which learns based on the ensemble method (Choi, Yoo, Kang, & Kim 2020). The regression algorithm typically uses linear regression and has the characteristic of predicting continuous values (Mun & Jung, 2021). A representative unsupervised learning algorithm is K-means as a clustering algorithm, which is an algorithm that groups the given data into number of *K* (Yoo, Lim, Ihm, Choi, & Kang, 2017). Finally, there is Q-Learning as a representative reinforcement learning algorithm, and it is an algorithm in which Agent and Environment are learned while Action, Reward, and State interact (Jung, Kim, Im, & Ihm 2021).

2.2. CNN

A Convolution Neural Network (CNN) is a type of deep learning network that is used to find patterns to analyze images (Jeong & Zhang, 2017). It learns images directly from data and uses the patterns to classify images. CNN is an advantage in learning while maintaining spatial information of images (Kang & Choi, 2021).

Image-related deep learning techniques can be divided into before and after CNN. Before the advent of the CNN technique, image recognition was trained with a fully connected multi-layered neural network (FNN) after converting two-dimensional image data into a one-dimensional array. The disadvantage of FNN is that the correlation between adjacent pixels is ignored. Since FNN receives data expressed in vector form, the image must be vectorized. However, since image data generally has a very high correlation between adjacent pixels, information loss occurs in the process of vectorizing the image. On the other hand, since CNN receives matrix-type data to preserve the shape of the image, information loss that occurs in the process of vectorizing the image can be prevented (LeCun, Bottou, Bengio, & Haffner, 1998). Overfitting is often a problem with CNN. Therefore, dropout technique is mainly used to avoid overfitting. Overfitting refers to a state in which the neural network is over-adapted only to the training data and cannot properly respond to other data (Park, Choi, Kang, & Jung, 2017).

2.3. Computer Vision

Computer vision is academic field of artificial intelligence that enables computers to identify and understand objects and people in images and videos. The goal of computer vision is to program a computer that understands images or features of images (Szeliski, 2010).

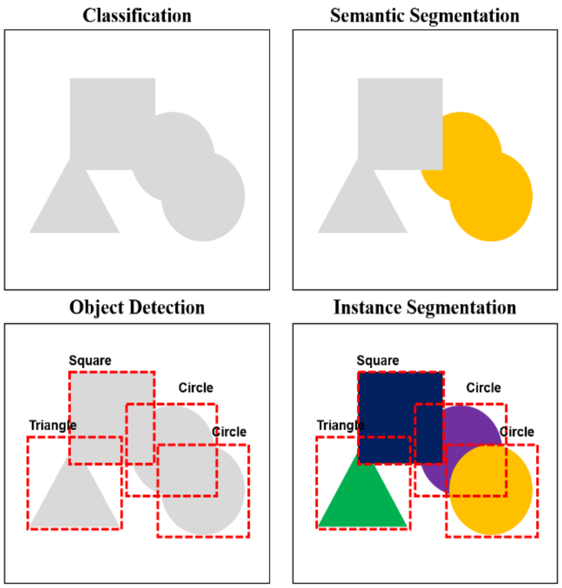


Figure 1: Type of Detection

Figure 1 shows the detection types of computer vision. Computer vision detection is divided into Classification, Semantic Segmentation, Object Detection, and Instance Segmentation. Classification refers to the classification of a photograph, which is a detection target, and the result that the picture is a figure can be expected through the Classification item in Figure 1 (Forsyth & Ponce, 2011) Object Detection classifies objects in the photo to be detected, and the results of the Bounding Box can be expected through the Object Detection item in Figure 1. In Semantic Segmentation, the result of masking can be expected by painting the pixels corresponding to the circle to be detected with the same color. Instance Segmentation is a fusion of Semantic Segmentation and Object Detection. By classifying each object and presenting a Bounding Box, various types of objects and results of the Bounding Box can be expected.

2.4. Mask R-CNN

Mask R-CNN is an instance segmentation algorithm proposed by Facebook (He, Gkioxari, Dollar, & Girshick, 2017).

Mask R-CNN is an improved algorithm from the previously implemented Faster R-CNN. As an improved item, a new mask branch was added to the classification and localization (bounding box regression) branches of the first Fast R-CNN. Before the second RPN (Region Pyramid Network), FPN (Feature Pyramid Network) was added. For the third image segmentation masking, RoI align has replaced RoI pooling. Figure 2 shows the prediction process of Mask R-CNN.

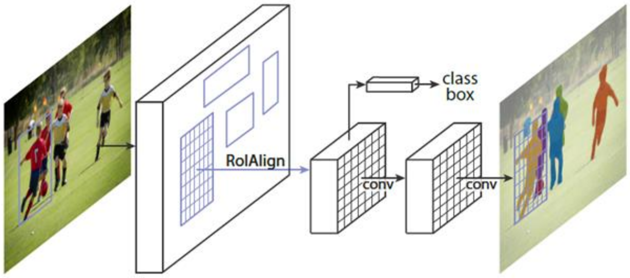


Figure 2: Mask R-CNN

Figure 3 shows the structure of Mask R-CNN. When an input image of size $N \times N$ is given, the process of Mask R-CNN is as follows.

Resize the image to size 800~1024 (Bilinear interpolation) and adjust it to the input size of 1024 x 1024 to enter the input of the Backbone network (Padding). Feature maps (C1, C2, C3, C4, C5) are generated in each layer (stage) through ResNet-101. P2, P3, P4, P5, and P6 feature maps are generated from the previously generated feature maps through FPN. create the classification and bounding box regression output values are derived by applying RPN to each of the finally generated feature maps. An anchor box is created by projecting the bounding box regression value obtained as an output to the original image. Delete all but the anchor box with the highest score among the anchor boxes created through non-max-suppression. Adjust the size of anchor boxes of different sizes through RoI align. We pass the anchor box value to the mask branch along with the classification and bounding box regression branch in Fast R-CNN.

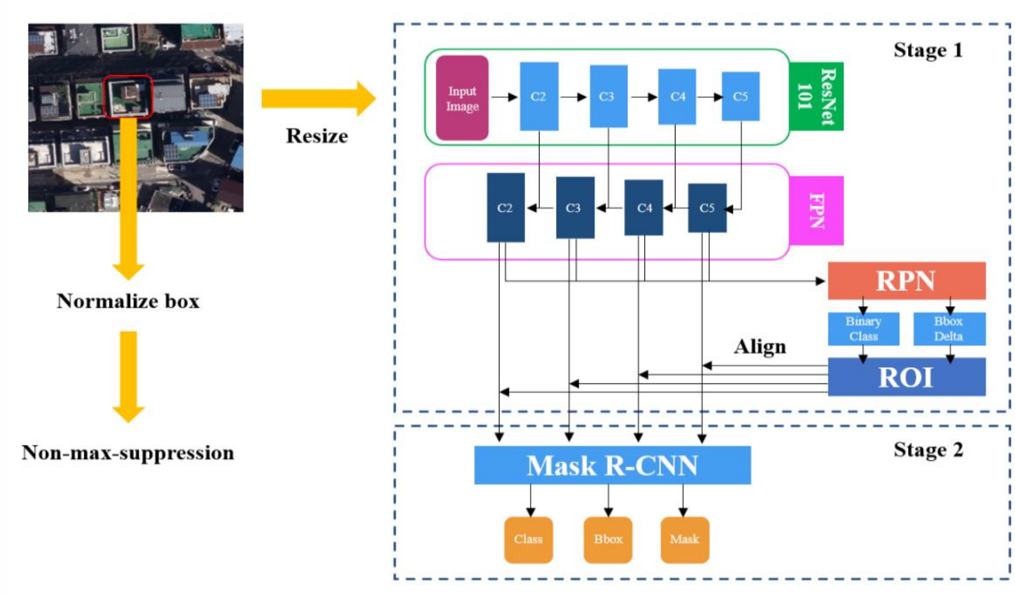


Figure 3: Structure of Mask R-CNN

3. Related Research

3.1. Aerial Image based Research

Table 1 shows Aerial Image based Research. Among them, it can be seen that Deep U-net, DeepLab V3+, FCN+ MLP, 2-levels U-Nets, Building-A-Nets, and Stacked ResUNet algorithms are applied in the case of object extraction for buildings.

Table 1: Aerial Image based Research

Title	Target	Algorithm
An empirical study on automatic building extraction from aerial images using a deep learning algorithm	Building	Deep U-Net, DeepLab V3+
Deep Learning Architecture for Building Extraction from Aerial Images	Building	FCN+ MLP, U-Nets, ResUNET

An empirical study on automatic building extraction from aerial images using a deep learning algorithm uses Deep U-net and DeepLab V3+ algorithms for buildings in Daejeon Metropolitan City, and then applies Deep U-net with excellent performance. As a result of the application, it was derived that there is some limitation in classifying single-family houses and buildings other than detached houses among buildings. In this study, object detection was applied compared to masking the object detected by applying instance segmentation (Seo, Oh, Kim, David, & Yang, 2019).

Deep Learning Architecture for Building Extraction from Aerial Images uses INRIA aerial imagery for buildings and uses FCN+ MLP, 2-levels U-Nets, Building-A-Nets, and Stacked ResUNet algorithms for overseas buildings. After learning, the predictive model was applied. As a result of application, it stacked ResUNet algorithm showed excellent performance (Na, Kim, & Choi, 2019).

3.2. Mask R-CNN based Research

Table 2 shows Mask R-CNN based Research. It is applied to various fields and objects such as clothing, tomatoes, and instance segmentation.

Mask R-CNN deep learning for fashion element detection is a Mask R-CNN deep mask for fashion image data set provided by iMaterialist Fashion Attribute Dataset to prepare a framework that can provide personalized AI fashion coordination service. A fashion element was detected using a learning algorithm. As a result of deep learning for detecting fashion elements with a total of 8 widths, the loss of training data was found to be Lcls 0.53, Lbox 0.38, and Lmask 0.35. And the loss of validation data was Lcls 0.54, Lbox 0.33, and Lmask 0.36. After adding various conditions such as color, season, material, trend, and brand name based on the fashion image data set owned by the individual. It is expected that fine tuning using the deep learning method implemented in this paper will be an effective individual AI fashion coordination service (Kim, 2021).

Table 2: Mask R-CNN based Research

Title	Target
Mask R-CNN deep learning for fashion element detection	Clothing
Instance Segmentation based Recognition System Tracking Tomatoes by Ripeness in Natural Light Conditions	Tomato
Automatic Dataset Generation of Object Detection and Instance Segmentation using Mask R-CNN	Instance Segmentation

Instance Segmentation based Recognition System Tracking Tomatoes by Ripeness in Natural Light Conditions, which can track tomatoes by maturity in natural light conditions, proposed a tomato recognition system for use in smart farms. The tomato recognition system consists of a deep learning model for performing tomato area and velocity recognition, a Kalman filter-based multi-object tracking algorithm, and a camera and lighting for image acquisition. The deep learning model used in the proposed system is Mask R-CNN, and a large-scale tomato image data set was collected using a camera system to train the model. The annotation of the collected data can transform the location information and ripeness information of each tomato expressed in the form of a mask into the form of a bounding box. It can also be used for learning. The tomato recognition result using the model learned in the proposed system was evaluated, and it was confirmed that the tomato in the robot's workspace was recognized with excellent performance. In addition, the performance of the Kalman filter-based multi-object tracking algorithm was also verified using MOTA. In the proposed paper, the goal was to recognize the location and ripeness of tomatoes included in the tomato harvesting robot's workspace, but the recognition target was set to other crops. If it is changed, it will be able to be used as a recognition system for the harvest robot for the crop, and if the object recognition range is extended to the entire frame rather than limited to the robot's workspace, it can be applied to the farm's inventory management system, crop status, etc. It can be extended to other fields as well. In addition, he said that he plans to conduct research on real tomato harvesting based on deep learning by composing a harvest robot by combining a tomato recognition system, an autonomous driving platform, and a manipulator (Lee, Ko, Kang, Park, & Jang, 2020).

Automatic Dataset Generation of Object Detection and Instance Segmentation using Mask R-CNN, uses Mask R-CNN algorithm learned by manipulating the structure of the dataset to automate image creation and labeling tasks, thereby recognizing objects. and a method that can significantly reduce the labeling cost for generating a dataset that can be used in the segmentation algorithm. Through experiments, object recognition algorithms and objects When the partitioning algorithm was trained with the data set generated by the proposed method rather than the actual data set, there was a difference of about 5% based on the AP. However, the data set generation efficiency was about 56 times higher when generating about 3,000 images, and it was shown that the efficiency increases as the amount of data sets generated increases. In addition to Mask R-CNN, which is an object segmentation algorithm, the proposed method was shown to be applicable to various algorithms by demonstrating that the proposed method can be applied to YOLO v3, an object recognition algorithm. Therefore, by using the proposed algorithm, it will be possible to increase the utility of object recognition and object segmentation by significantly reducing the data generation cost required for object recognition and object segmentation in fields that deal with various objects such as logistics sites (Jo, Kim, & Song, 2019).

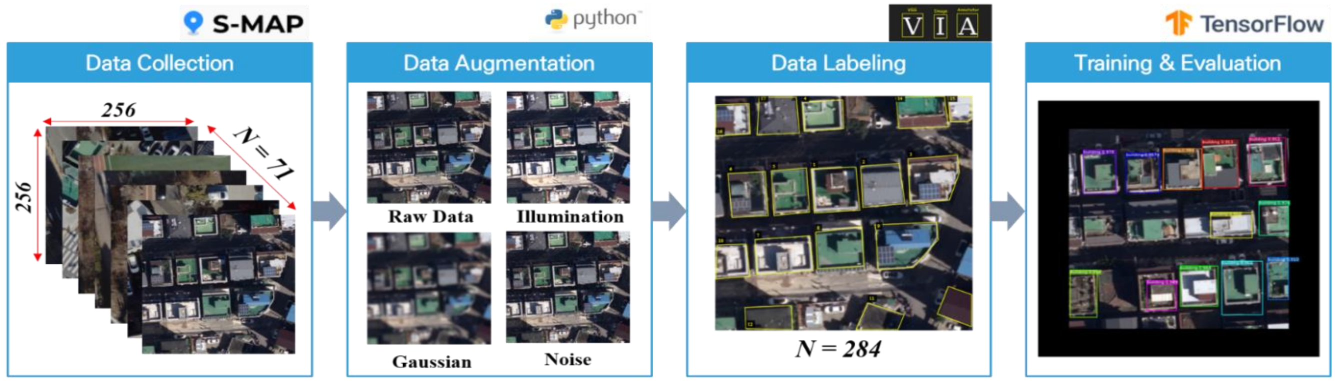


Figure 4: Process of Research

4. Experimental Model

4.1. Experiment Environments

4.1.1. H/W Environments

Table 3 shows H/W Environments. Implemented based on i5-8500(3.00GHz), 16GB RAM, GTX 1060(3GB).

Table 3: H/W Environments

Index	Spec
CPU	I5-8500, 3.00GHz
RAM	16GB
GPU	GTX 1060 3GB

4.1.2. S/W Environments

Table 4 shows S/W Environments. Implemented based on Window 10 Home, VIA Tool, Python 3.6, Tensorflow 1.5.0, Keras 2.2.0.

S/W for data preprocessing used VIA (VGG Image Annotator) developed by Oxford University. Window 10 Home, Python 3.6, Tensorflow 1.5.0, and Keras 2.2.0 were utilized as the S/W environment for the training model.

Table 4: S/W Environments

Index	Spec
OS	Window 10 Home
Preprocessing	VIA Tool
Language	Python 3.6
Module	Tensorflow 1.5.0, Keras 2.2.0

4.2. Experiment

Figure 4 shows the overall process of the experiment. Experiments were conducted with data collection, data augmentation, labeling, training, and validation.

4.2.1. Data Collection

Tile data was collected through S-Map of Seoul. Data collection targets include buildings, industrial buildings, buildings under construction, and vegetation. Table 5 shows examples of data to be collected.

Table 5: Examples of training data

Building	Industrial Building
Under Construction	Vegetation

4.2.2 Data Labeling

Labeling was performed on the target building using VIA. The region value of labeling is Polygon, and labeling was performed based on the vertices of each object. In addition, data augmentation was performed along with labeling. Figure 5 shows the status of labeling applied.

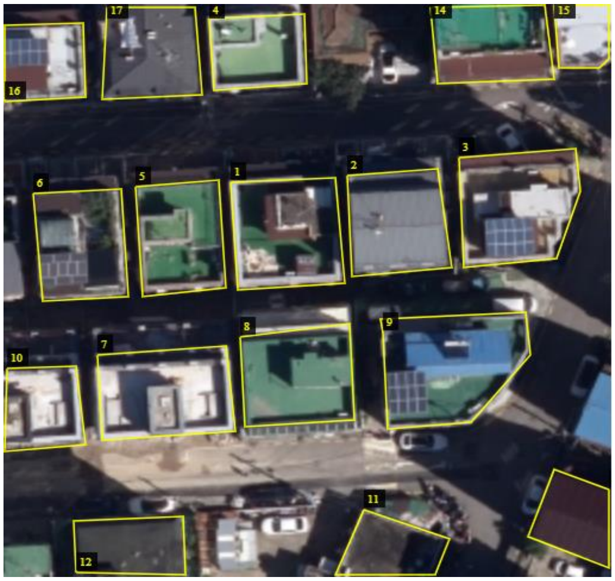
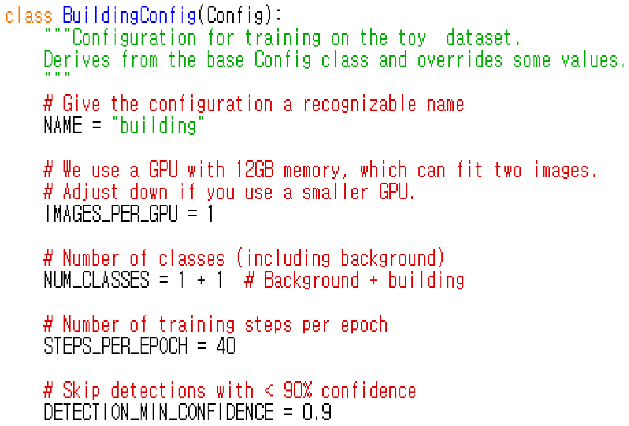


Figure 5: Labeling

4.2.3 Training

Mask R-CNN algorithm was used for learning. Epoch is a total of 5 times and Iteration is a total of 40 times, 200 times of learning, Training was performed based on GPU.

Figure 6 shows the model configuration. The prediction target was set to Building, and it was set not to detect if the confidence did not over 90%.



```
class BuildingConfig(Config):
    """Configuration for training on the toy  dataset.
    Derives from the base Config class and overrides some values.
    """

    # Give the configuration a recognizable name
    NAME = "building"

    # We use a GPU with 12GB memory, which can fit two images.
    # Adjust down if you use a smaller GPU.
    IMAGES_PER_GPU = 1

    # Number of classes (including background)
    NUM_CLASSES = 1 + 1  # Background + building

    # Number of training steps per epoch
    STEPS_PER_EPOCH = 40

    # Skip detections with < 90% confidence
    DETECTION_MIN_CONFIDENCE = 0.9
```

Figure 6: Model Configuration

4.2.4 Results

As a result of the prediction, it can be confirmed that learning about the shape was smoothly performed through the RPN generated for all buildings as follows. Figure 7 shows the RPN prediction result.

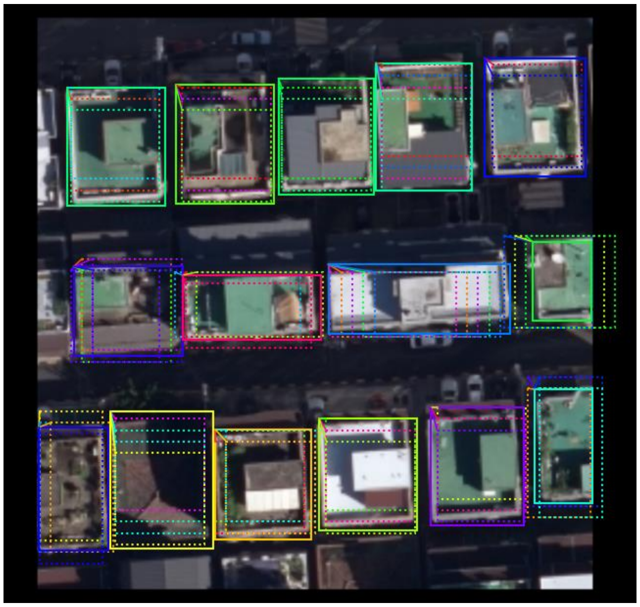


Figure 7: RPN Prediction Results

Figure 8 shows the results of classification based on the predicted probability (truncation of less than 90%). In general, it can be seen that the prediction probability is low when there is a shadow.

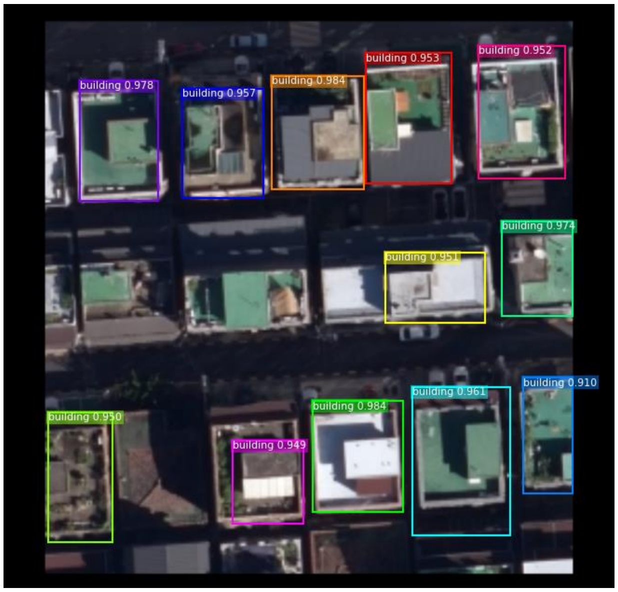


Figure 8: Bonduing Box Prediction Results

Based on the Bounding Box prediction result, the prediction result can be confirmed as shown in Figure 9 through masking.

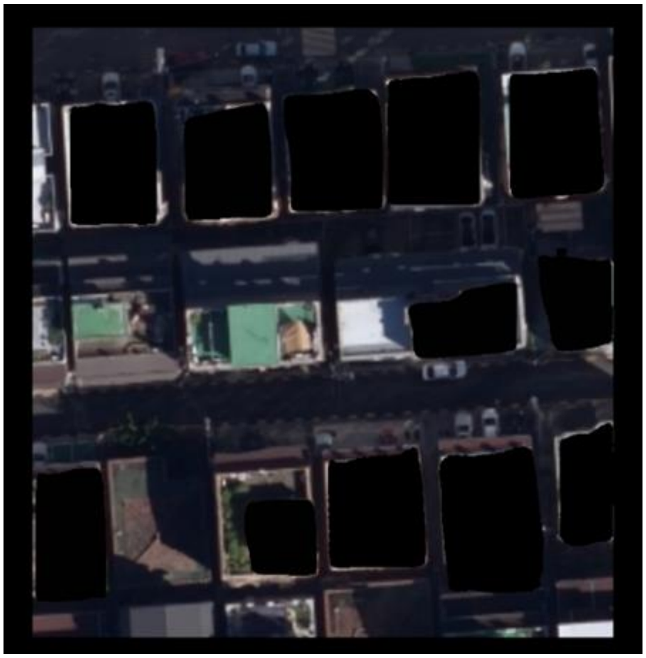
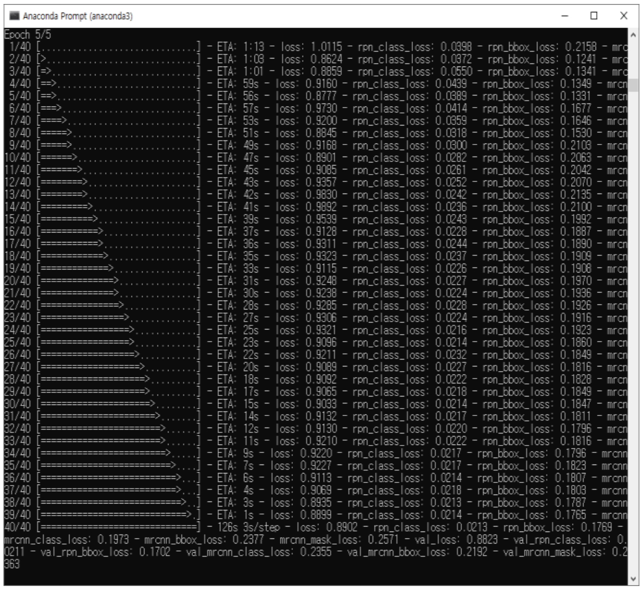


Figure 9: Masking Prediction Results

4.2.5 Evaluation

In Figure 10, as the verification result, it can be seen that learning and verification were successfully performed through various loss values.



```
Epoch 5/5
 1/40 [..............................] - ETA: 1:13 - loss: 1.0115 - rpn_class_loss: 0.0398 - rpn_bbox_loss: 0.2158 - mrc
 2/40 [>.............................] - ETA: 1:03 - loss: 0.8624 - rpn_class_loss: 0.0372 - rpn_bbox_loss: 0.1241 - mrc
 3/40 [=>............................] - ETA: 1:01 - loss: 0.8859 - rpn_class_loss: 0.0550 - rpn_bbox_loss: 0.1341 - mrc
 4/40 [==>...........................] - ETA: 59s - loss: 0.9160 - rpn_class_loss: 0.0439 - rpn_bbox_loss: 0.1349 - mrcn
 5/40 [==>...........................] - ETA: 56s - loss: 0.8777 - rpn_class_loss: 0.0389 - rpn_bbox_loss: 0.1331 - mrcn
 6/40 [===>..........................] - ETA: 57s - loss: 0.9730 - rpn_class_loss: 0.0414 - rpn_bbox_loss: 0.1677 - mrcn
 7/40 [====>.........................] - ETA: 53s - loss: 0.9200 - rpn_class_loss: 0.0359 - rpn_bbox_loss: 0.1646 - mrcn
 8/40 [=====>........................] - ETA: 51s - loss: 0.8845 - rpn_class_loss: 0.0318 - rpn_bbox_loss: 0.1530 - mrcn
 9/40 [=====>........................] - ETA: 49s - loss: 0.9168 - rpn_class_loss: 0.0300 - rpn_bbox_loss: 0.2103 - mrcn
10/40 [======>.......................] - ETA: 47s - loss: 0.8901 - rpn_class_loss: 0.0282 - rpn_bbox_loss: 0.2063 - mrcn
11/40 [=======>......................] - ETA: 45s - loss: 0.9085 - rpn_class_loss: 0.0261 - rpn_bbox_loss: 0.2042 - mrcn
12/40 [========>.....................] - ETA: 43s - loss: 0.9357 - rpn_class_loss: 0.0252 - rpn_bbox_loss: 0.2070 - mrcn
13/40 [========>.....................] - ETA: 42s - loss: 0.9830 - rpn_class_loss: 0.0242 - rpn_bbox_loss: 0.2135 - mrcn
14/40 [=========>....................] - ETA: 41s - loss: 0.9892 - rpn_class_loss: 0.0236 - rpn_bbox_loss: 0.2100 - mrcn
15/40 [==========>...................] - ETA: 39s - loss: 0.9539 - rpn_class_loss: 0.0243 - rpn_bbox_loss: 0.1992 - mrcn
16/40 [===========>..................] - ETA: 37s - loss: 0.9128 - rpn_class_loss: 0.0228 - rpn_bbox_loss: 0.1887 - mrcn
17/40 [===========>..................] - ETA: 36s - loss: 0.9311 - rpn_class_loss: 0.0244 - rpn_bbox_loss: 0.1890 - mrcn
18/40 [============>.................] - ETA: 35s - loss: 0.9323 - rpn_class_loss: 0.0237 - rpn_bbox_loss: 0.1909 - mrcn
19/40 [=============>................] - ETA: 33s - loss: 0.9115 - rpn_class_loss: 0.0226 - rpn_bbox_loss: 0.1908 - mrcn
20/40 [==============>...............] - ETA: 31s - loss: 0.9248 - rpn_class_loss: 0.0227 - rpn_bbox_loss: 0.1970 - mrcn
21/40 [==============>...............] - ETA: 30s - loss: 0.9238 - rpn_class_loss: 0.0224 - rpn_bbox_loss: 0.1936 - mrcn
22/40 [===============>..............] - ETA: 28s - loss: 0.9285 - rpn_class_loss: 0.0228 - rpn_bbox_loss: 0.1926 - mrcn
23/40 [================>.............] - ETA: 27s - loss: 0.9306 - rpn_class_loss: 0.0224 - rpn_bbox_loss: 0.1916 - mrcn
24/40 [=================>............] - ETA: 25s - loss: 0.9321 - rpn_class_loss: 0.0216 - rpn_bbox_loss: 0.1923 - mrcn
25/40 [=================>............] - ETA: 23s - loss: 0.9096 - rpn_class_loss: 0.0214 - rpn_bbox_loss: 0.1860 - mrcn
26/40 [==================>...........] - ETA: 22s - loss: 0.9211 - rpn_class_loss: 0.0232 - rpn_bbox_loss: 0.1849 - mrcn
27/40 [===================>..........] - ETA: 20s - loss: 0.9089 - rpn_class_loss: 0.0227 - rpn_bbox_loss: 0.1816 - mrcn
28/40 [====================>.........] - ETA: 18s - loss: 0.9092 - rpn_class_loss: 0.0222 - rpn_bbox_loss: 0.1828 - mrcn
29/40 [====================>.........] - ETA: 17s - loss: 0.9065 - rpn_class_loss: 0.0218 - rpn_bbox_loss: 0.1849 - mrcn
30/40 [=====================>........] - ETA: 15s - loss: 0.9033 - rpn_class_loss: 0.0214 - rpn_bbox_loss: 0.1847 - mrcn
31/40 [======================>.......] - ETA: 14s - loss: 0.9132 - rpn_class_loss: 0.0217 - rpn_bbox_loss: 0.1811 - mrcn
32/40 [=======================>......] - ETA: 12s - loss: 0.9130 - rpn_class_loss: 0.0220 - rpn_bbox_loss: 0.1796 - mrcn
33/40 [=======================>......] - ETA: 11s - loss: 0.9210 - rpn_class_loss: 0.0222 - rpn_bbox_loss: 0.1816 - mrcn
34/40 [========================>.....] - ETA: 9s - loss: 0.9220 - rpn_class_loss: 0.0217 - rpn_bbox_loss: 0.1796 - mrcnn
35/40 [=========================>....] - ETA: 7s - loss: 0.9227 - rpn_class_loss: 0.0217 - rpn_bbox_loss: 0.1823 - mrcnn
36/40 [==========================>...] - ETA: 6s - loss: 0.9113 - rpn_class_loss: 0.0214 - rpn_bbox_loss: 0.1807 - mrcnn
37/40 [==========================>...] - ETA: 4s - loss: 0.9069 - rpn_class_loss: 0.0218 - rpn_bbox_loss: 0.1803 - mrcnn
38/40 [===========================>..] - ETA: 3s - loss: 0.8935 - rpn_class_loss: 0.0213 - rpn_bbox_loss: 0.1787 - mrcnn
39/40 [============================>.] - ETA: 1s - loss: 0.8899 - rpn_class_loss: 0.0214 - rpn_bbox_loss: 0.1765 - mrcnn
40/40 [==============================] - 126s 3s/step - loss: 0.8902 - rpn_class_loss: 0.0213 - rpn_bbox_loss: 0.1769 -
mrcnn_class_loss: 0.1973 - mrcnn_bbox_loss: 0.2377 - mrcnn_mask_loss: 0.2571 - val_loss: 0.8823 - val_rpn_class_loss: 0.
0211 - val_rpn_bbox_loss: 0.1702 - val_mrcnn_class_loss: 0.2355 - val_mrcnn_bbox_loss: 0.2192 - val_mrcnn_mask_loss: 0.2
363
```

Figure 10: Evaluation Results

5. Conclusion

To improve the required time and quality of digital twin construction, this study tried to detect buildings using deep learning-based aerial images.

For deep learning-based building detection, aerial photos of buildings were collected through Seoul S-Map. In addition, data augmentation was performed by various methods such as illumination and Gaussian noise. The collected and augmented data were learned and verified through the Mask R-CNN algorithm along with data labeling using the VIA Tool. As a result of the verification, it can be confirmed that 100% accuracy detection is difficult due to the shadow of the aerial image, which is the fundamental data. This suggests that data quality is more important before AI can learn to make predictions.

Through this study, it is expected that it can be used as a preliminary study for preprocessing and labeling automation research for digital twin implementation.

References

An, S. H., Yeo, S. H., & Kang, M. S. (2021). A Study on a car Insurance purchase Prediction Using Two-Class Logistic Regression and Two-Class Boosted Decision Tree. *Korean Journal of Artificial Intelligence*, *9*(1), 9-14.

Choi, E. S., Yoo, H. J., Kang, M. S., & Kim, S. A. (2020). Applying Artificial Intelligence for Diagnostic Classification of Korean Autism Spectrum Disorder. *Psychiatry investigation*, *17*(11), 1090-1095.

Forsyth, D., & Ponce, J. (2011). *Computer vision: A modern approach.* Prentice hall.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. *In Proceedings of the IEEE international conference on computer vision*, 2961-2969.

Jung, I. C., Kim, Y. S., Im, S. R., & Ihm, C. H., (2021). A Development of Nurse Scheduling Model Based on Q-Learning Algorithm. *Korean Journal of Artificial Intelligence*, *9*(1), 1-7.

Jeong, B. J., Zhang, Fan. (2017). A Study on the Emoticon Extraction based on Facial Expression Recognition using Deep Learning Technique. *Korean Journal of Artificial Intelligence*, *5*(2), 43-53.

Jo, H. J., Kim, Dawit., & Song, J. B. (2019). Automatic Dataset Generation of Object Detection and Instance Segmentation using Mask R-CNN. *Journal of Korea Robotics Society*, *14*(1), 31-39.

Kang, M. S., & Choi, E.S. (2021). MACHINE LEARNING: Concepts, Tools and Data Visualization. World Scientific.

Kim, H. S. (2021). Mask R-CNN deep learning for fashion element detection. *Journal of Digital Contents Society*, *22*(4), 689-696.

Kim, S. Y., Lee, H. H., Choi, E. S., & Go, J. U. (2020). A Case Study on the Construction of 3D Geo-spatial Information for Digital Twin Implementation. *Journal of the Korean Association of Geographic Information Studies*, *23*(3), 146-160.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Lee, Y. S. (2017). Value Creation and Competitiveness Achievement Strategies of Smart Cities. *Journal of the Korean Regional Science Association*, *33*(1), 59-68.

Lee, I. S. (2021). A Study on Geospatial Information Role in Digital Twin. *Journal of the Korea Academia-Industrial Cooperation Society*, *22*(3), 268-278.

Lee, W. Y., Ko, K. E., Kang, J. H., Park, H. J., & Jang, I. H. (2020). Instance Segmentation based Recognition System Tracking Tomatoes by Ripeness in Natural Light Condition. *Journal of Institute of Control, Robotics and Systems*, *26*(11), 940-948.

Mun, J. H., Jung, S. W., (2021). A customer credit Prediction Researched to Improve Credit Stability based on Artificial Intelligence. *Korean Journal of Artificial Intelligence*, *9*(1), 21-27.

Na, Y. H., Kim, J. H., & Choi, J. P. (2019). Deep Learning Architecture for Building Extraction from Aerial Images. *Proceedings of Symposium of the Korean Institute of communications and Information Sciences, Korea Institute of Communication Sciences*, 396-397.

Park, J. G., Choi, E. S., Kang, M. S., & Jung, Y. G. (2017). Dropout Genetics Algorithm Analysis for Deep Learning Generalization Error Minimization. *International Journal of Advanced Culture Technology*, *5*(2), 74-81.

Park, Y. J. (2019). Strategy for Building Smart City as a Platform of the 4th Industrial Revolution. *Journal of Digital Convergence*, *17*(1), 169-177.

Rasheed, A., San, O. and Kvamsdal, T. 2020. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access* 8:21980-22012.

Seo, K. H., Oh, C. H., Kim, David., Lee, M. Y., & Yang, Y. J. (2019). An empirical study on automatic building extraction from aerial images using a deep learning algorithm. *Proceedings of Korean Society for Geospatial Information Science, Korea Spatial Information Society*, 243-252.

Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.

Yoo, J. H., Lim, M. K., Ihm, C. H., Choi, E. S., & Kang, M. S., (2017). A Study on Prediction of Rheumatoid Arthritis Using Machine Learning. *International Journal of Applied Engineering Research*, *12*(20), 9858-9862.