

A Study on Pagoda Image Search Using Artificial Intelligence (AI) Technology for Restoration of Cultural Properties

ByongKwon Lee¹, Soo Kyun Kim², and Seokhun Kim^{3*}

¹ Department of Multimedia, Seowon University Republic of Korea
[e-mail: sonic747@seowon.ac.kr]

² Department of Computer Engineering, Jeju National University Republic of Korea
[e-mail: kimsk@jejunu.ac.kr]

³ Department of E-Commerce, Paichai University Republic of Korea
[e-mail: vambition@daum.net]

*Corresponding author: Seokhun Kim

*Received January 4, 2021; revised April 26, 2021; accepted May 19, 2021;
published June 30, 2021*

Abstract

The current cultural assets are being restored depending on the opinions of experts (craftsmen). We intend to introduce digitalized artificial intelligence techniques, excluding the personal opinions of experts on reconstruction of such cultural properties. The first step toward restoring digitized cultural properties is separation. The restoration of cultural properties should be reorganized based on recorded documents, period historical backgrounds and regional characteristics. The cultural properties in the form of photographs or images should be collected by separating the background. In addition, when restoring cultural properties most of them depend a lot on the tendency of the restoring person workers. As a result, it often occurs when there is a problem in the accuracy and reliability of restoration of cultural properties. In this study, we propose a search method for learning stored digital cultural assets using AI technology. Pagoda was selected for restoration of Cultural Properties. Pagoda data collection was collected through the Internet and various historical records. The pagoda data was classified by period and region, and grouped into similar buildings. The collected data was learned by applying the well-known CNN algorithm for artificial intelligence learning. The pagoda search used Yolo Marker to mark the tower shape. The tower was used a total of about 100-10,000 pagoda data. In conclusion, it was confirmed that the probability of searching for a tower differs according to the number of pagoda pictures and the number of learning iterations. Finally, it was confirmed that the number of 500 towers and the epochs in training of 8000 times were good. If the test result exceeds 8,000 times, it becomes overfitting. All so, I found a phenomenon that the recognition rate drops when the enemy repeatedly learns more than 8,000 times. As a result of this study, it is believed that it will be helpful in data gathering to increase the accuracy of tower restoration.

Keywords: Enter key words or phrases in alphabetical order, separated by commas.

This work was supported by the research grant of Pai Chai University in 2020.

1. Introduction

Each country in the world has various types of cultural properties. The current cultural assets are being restored depending on the opinions of experts (craftsmen). We intend to introduce digitalized artificial intelligence techniques, excluding the personal opinions of experts(workers) on reconstruction of such cultural properties. Many of the world's cultural properties are pagoda and are the most numerous in Southeast Asia [1]. it has been damaged and destroyed by war and robbery. Most of these cultural properties are restored and made with the technique of a professional engineer (as restore workers) However, a professional engineer may change the restoration form according to his or her taste and propensity. The first step in restoring cultural properties is to find the best pagoda. The optimal pagoda search is the process of learning an image containing pagoda and finding a pagoda [2]. This study is expected to be actively utilized in the part of restoring the pagoda. This paper analyzes data collection and related research on AI algorithms in Section 2, data organization and experiment environment in Section 3, and experimental results in Section 4. The last verse 5 concludes.

2. Related Works

2.1 YOLO(Only Look Once) algorithm

YOLO means you Only Look Once, a deep learning-based object recognition algorithm that guesses the type and location of an object by just looking at an image [3]. The YOLO algorithm is very fast compared to other deep learning algorithms and high mean average precision (mAP). However, it shows low accuracy for small objects and Used HD (1024*768) image to find cultural property pagoda. YOLO algorithm finds the location of the bounding box and classifies the class at the same time in the final output of the network [4]. In addition, The YOLO algorithm network simultaneously extracts features as creates bounding boxes and classifies them. The recognition speed is faster than other artificial intelligence algorithms. As you can see in Fig. 1. The left side of Fig.1 processing images with YOLO is simple and straightforward. (1) resizes the input image to 448 x 448, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence. The right side of Fig. 1 divides the image into an $S \times S$ grid and for each grid cell predict bounding boxes, confidence for those boxes, and class probabilities. These predictions are encoded as a tensor.

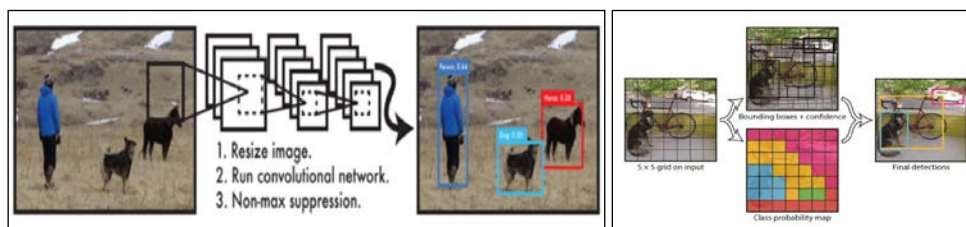


Fig. 1. YOLO algorithm process.

2.2 CNN (Convolution Neural Network)

CNN is especially useful for finding patterns to recognize images. CNN algorithms do not need to manually extract features by learning from data and classifying images using patterns [5]. The CNN algorithm is widely used in autonomous vehicles, face recognition and fields that require computer vision. A general image recognition method used a method of converting a two-dimensional image (three-dimensional including channels) into a one-dimensional array and then training it with a fully connected (FC) neural network. As you can see in Fig. 2.

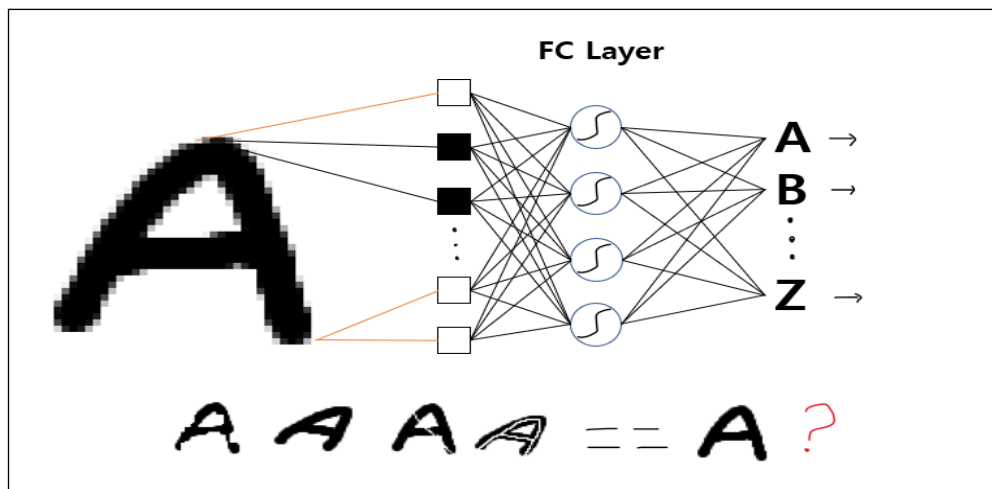


Fig. 2. FC neural network algorithms

As you can see in Fig. 2 recognition does not consider the image shape and processes raw data directly. However, the disadvantage is that the learning time is long. The FC algorithm must process new image data when the image rotates or moves. It is characterized by learning by looking at simple 1D data without understanding the characteristics of images, and spatial information is inevitably lost in the process of flattening image data. The FC algorithm is inefficient in extracting and learning features of a neural network and has limitations in increasing accuracy.

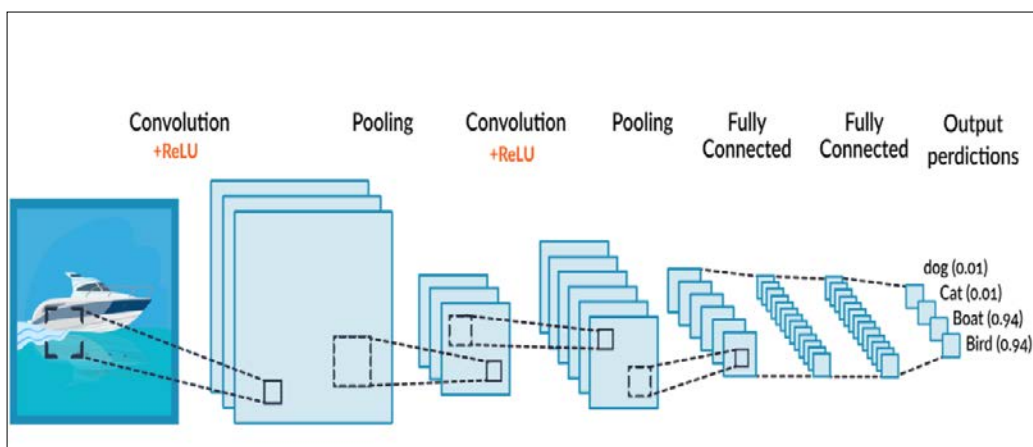


Fig. 3. CNN (Convolution Neural Network) procedure

CNN is a structure that extracts features of data and identifies patterns of features [6]. The CNN algorithm proceeds through the convolution process and the pooling process. An algorithm is created by composing the convolution layer and the pooling layer in combination. Fig. 3 shows the CNN algorithm procedure. This is the process of reducing the size of the layer that has undergone the convolution process [7]. This section describes the environment setting and implementation for artificial intelligence learning about the cultural property for the pagoda and also describes the process of finding the pagoda image based on the learned data.

3.1 CNN learning environment

The artificial intelligence algorithm to find the cultural property pagoda was implemented by installing DNN [8] to use the CNN learning algorithm. The implementation environment was installed with the CUDA platform provided by N-Vidia and the cuDNN library software platform and the software implementation environment consisted of OpenCV for the Visual Studio 20xx. The CNN implementation environment procedure is as follows for pagoda as digital image. (1) The visual studio platform for AI includes a development language to windows SDK 10 and the base language C++ [visual studio 20xx]. (2) CUDA (Computed Unified Device Architecture) is a GPU development tool developed by NVIDIA [9]. CUDA requires a completely basic H/W approach that is based on c and c++ and many researchers have opened It can be easily installed and used for deep learning for AI as supported library anyone. Currently, we have tested cuda and cuDNN by installing an AI learning platform provided by Nvidia. [<https://developer.nvidia.com/rdp/cudnn-archive>]. (3) OpenCV (OpenSource Computer Vision Library) is a programming library for the purpose of real-time computer vision. It is focused on real-time image processing and supports IPP (Intel Performance Primitives) which can increase speed when used on Intel CPUs. The openCV library is cross-platform available for Windows, Linux, etc. and can be used free of charge under the BSD license. OpenCV supports deep learning frameworks from TensorFlow/Torch / PyTorch and Caffe[<https://opencv.org/releases/>]. In this study, openCV and cuDNN were installed on a PC using Intel Core I7, RAM 32Gbyte and GPU GTX 3070 to learn pagoda search using artificial intelligence.

3.2 Implementation Yolomark-Master for pagoda finding

We experimented with learning to find pagoda by downloading the yolomark-master application from the Github website. We used Yolomark-master to mark the square shape to create the data to be trained before learning the pagoda. The process of marking is as follows. (1) Download yolomark from github and import it to your PC. (Link: https://github.com/AlexeyAB/Yolo_mark) [10]. (2) 64bit and IPP are set to increase data processing speed. (3) Added OpenCV lib path to linker library for saving marked images. (4) When the above process is completed, "yolo_mark.cmd" and "yolo_mark.exe" are automatically created, and pagoda data set for AI learning is created.



Fig. 4. Yolomark implementation for Pagoda and process

Fig. 4 shows Pagoda digital image was imported to YoloMark application. As shown in the figure, one pagoda object id:0-tower is marked in a square shape. If we display the background and other objects in the form of object id:1-sky... we can improve the accuracy. The next step is to set up the trainset using Yolo-Darknet [11]. Darknet is the name of a neural network framework developed independently by Joseph Redmon and provides a framework for learning and executing DNNs (Deep Neural Networks). YOLO is one of the trained neural networks and is currently open to YOLO 5. [coco dataset <https://ukayzm.github.io/cocodataset/>]. Fig. 5 shows the process of learning after marking using Yolo Maker.

Performance on the COCO Dataset							
Model	Train	Test	mAP	FLOPS	FPS	Cfg	Weights
SSD300	COCO trainval	test-dev	41.2	-	46		link
SSD500	COCO trainval	test-dev	46.5	-	19		link
YOLOv2 608x608	COCO trainval	test-dev	48.1	62.94 Bn	40	cfg	weights
Tiny YOLO	COCO trainval	test-dev	23.7	5.41 Bn	244	cfg	weights
SSD321	COCO trainval	test-dev	45.4	-	16		link
DSSD321	COCO trainval	test-dev	46.1	-	12		link
R-FCN	COCO trainval	test-dev	51.9	-	12		link
SSD513	COCO trainval	test-dev	50.4	-	8		link
DSSD513	COCO trainval	test-dev	53.3	-	6		link
FPN FRCN	COCO trainval	test-dev	59.1	-	6		link
Retinanet-50-500	COCO trainval	test-dev	50.9	-	14		link
Retinanet-101-500	COCO trainval	test-dev	53.1	-	11		link
Retinanet-101-800	COCO trainval	test-dev	57.5	-	5		link
YOLOv3-320	COCO trainval	test-dev	51.5	38.97 Bn	45	cfg	weights
YOLOv3-416	COCO trainval	test-dev	55.3	65.86 Bn	35	cfg	weights
YOLOv3-608	COCO trainval	test-dev	57.9	140.69 Bn	20	cfg	weights
YOLOv3-tiny	COCO trainval	test-dev	33.1	5.56 Bn	220	cfg	weights
YOLOv3-spp	COCO trainval	test-dev	60.6	141.45 Bn	20	cfg	weights

Fig. 5. COCO Dataset

Fig. 5 is a prepared learning coco-dataset provided by yolo-darknet. The COCO dataset is a dataset for object detection and segmentation and key point detection [12]. The coco dataset is used every year in competitions where a different dataset is attended by several universities/corporations around the world. The TensorFlow Object Detection API released by Google also contains a model trained with the COCO dataset. This study was implemented by creating a Pagoda dataset as COCO dataset [13].

3.3 YOLO-mark learning image labeling for Pagoda

In this section, we edit the pagoda image based on the two files ("yolo_mark.cmd" and "yolo_mark.exe") created in section 3.2 for image labeling. YOLO-mark's labeling for pagoda on the image marks a rectangular shape. Fig. 6 is a marking of pagoda image data used in the study.

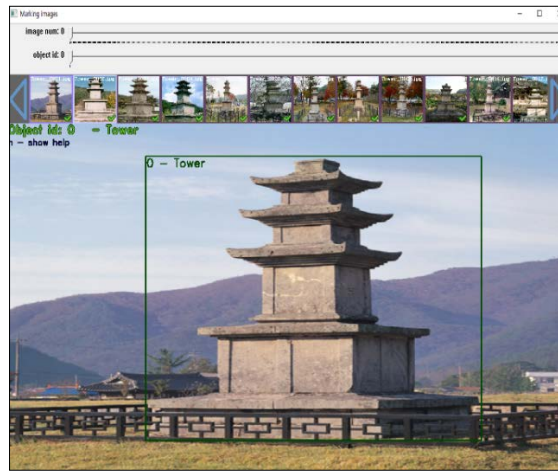


Fig. 6. marking of pagoda images

The pagoda image for marking needs to be modified in the "obj.names" and "obj.data" files. "obj.names" is a file that tags Pagoda (tower). The "obj.data" file configures the number of objects to be learned and the learning image data and the verification path(valid=data/train.txt). (Fig. 7)

```

File:obj.data
classes= 1           ←Number of objects to learn
train = data/train.txt ←image data path to learn
valid = data/train.txt ←Image data path to be verified
names = data/obj.names ←names file path
backup = backup/Tower/ ←Learned data storage path

File:obj.name      File: Tower_xxx.txt
Tower              0 0.544531 0.503472 0.598437 0.818056
    
```

Fig. 7. marking setting for AI learning

In the step, the marked information (id, square coordinate value:0 0.544 0.503 0.598 0.818) using Yolo Marker is recorded in "Tower_xxx.txt" and used as a dataset for AI learning. Fig. 7 is a file recording Yolo Marking information for pagoda images to file name "Tower_xxx.txt". The last step is to set up an AI learning environment using Yolo_darknet as a preparatory step. Environment settings modify the maximum number of learning repetitions (max_batches, filter, classes, etc). If the Yolo marker operation is normally executed the graph

window appears as shown Fig.8. The weight file is saved in the backup directory every specified number of repetitions (every 1000 times). Finally, the environment configuration file "yolo-obj.cfg" file for learning was set. [batch=64, subdivisions=8, height=416, width=310, channels=3, momentum=0.9, decay=0.0005, angle=0, saturation=1.5, exposure=1.5, hue=.1 etc]. "batch" determines how many sheets are processed at a time. "subdivisions" is processed by dividing the batch by this value. "height, width" means the size of the input image. The input image size is 416x310, and it is necessary to experiment by changing the size in the future.

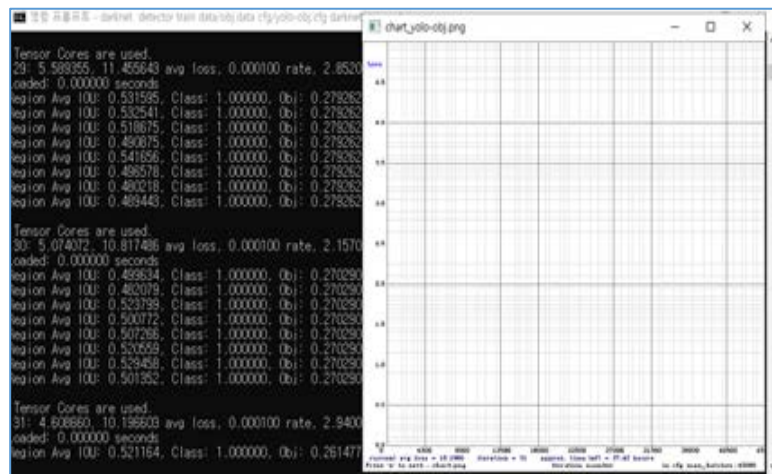


Fig. 8. Artificial intelligence learning processing for the Pagoda

4. Experiment and analysis results

The Pagoda dataset was used with Yolo Marker-Darknet. However, the recognition rate was lowered due to the lack of Pagoda image and Data set for AI learning. As a result, an error of recognizing images other than the Fig. 9 Pagoda image as a tower was issued (250 sets of Pagoda dataset and 1,000 repetitions). Fig. 9 show (1) is the original Pagoda dataset input and (2) the result recognized as an artificial intelligence learning. The number of datasets was increased and the background of the Pagoda image was removed and the experiment was conducted again as Fig. 10.

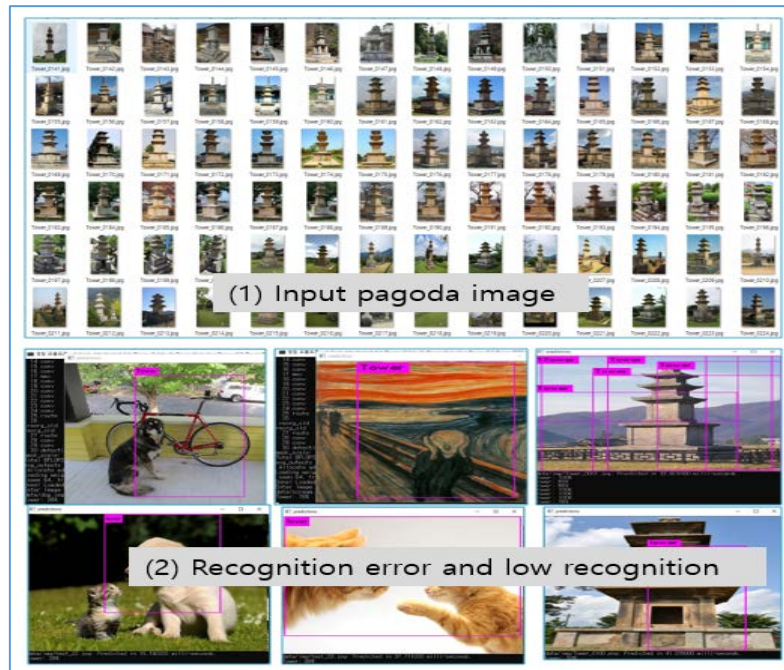


Fig. 9. Pagoda artificial intelligence learning

The dataset used for Pagoda AI training was used with 1,000 images. In Fig. 10, as a result of removing the background of pagoda, it was confirmed that the Tower search rate increased. The reason for this is that the reference data is small, but the accuracy of searching pagoda was high, over 95%. In conclusion, it is judged that pagoda's learning through artificial intelligence learning needs to correct the data.

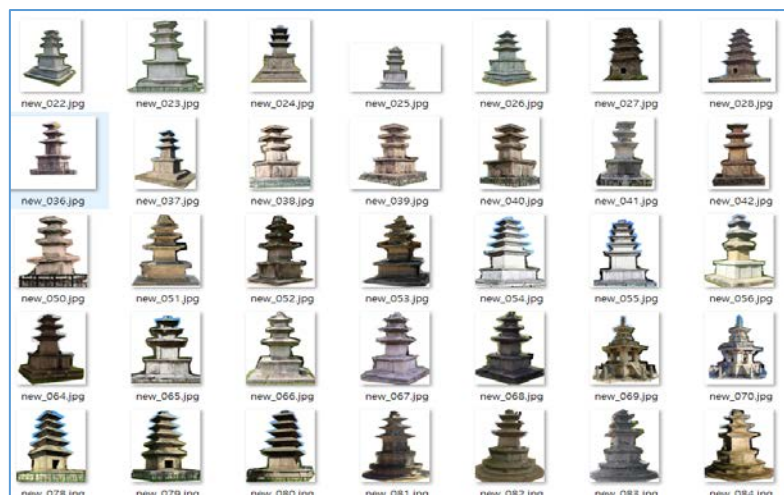


Fig. 10. Pagoda AI Learning Dataset

The experiment YOLO_Darknet for pagoda shape searching created a weight file (number of AI learned file) in units of 1,000 from in training 1,000 to 22,000 times. Fig. 11 is a weight file learned for Pagoda recognition as the generated weight file. In addition, the average loss rate for 1,000 repetitions is convergence to 0.0045.

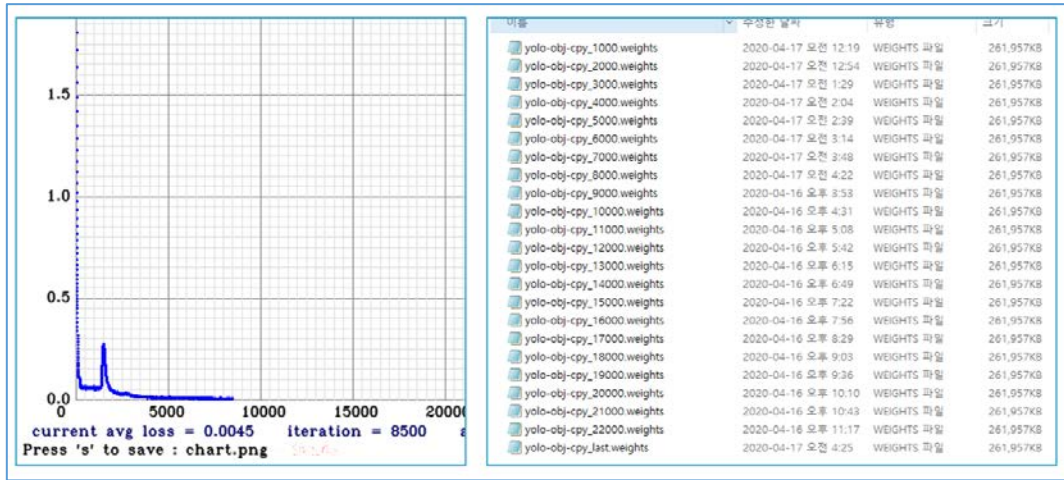


Fig. 11. Pagoda recognition as the generated weight file

This paper evaluated the performance of digital images of cultural property pagoda with the data without the background removed. A total of 22,500 cycles were repeated and proceeded until the error rate was minimized. The results shown in Fig. 12 were obtained. In Fig. 12. We experimented with two image types, one with the background of the pagoda image and one with only the pagoda with the background removed. It was confirmed that the data recognition rate increased as the number of repetitions increased. As a result of the experiment when entering 1,000 data sets. it was confirmed that the prediction rate was 98% when repeated 20,000 times. Fig. 13 shows the loss rate and prediction result for each repeated cycle in 1,000 cycles only the shape of the pagoda image was found, but there was a problem of recognizing to pagoda image. In 2,000 cycles started looking for a pagoda shape and 3,000 cycles did not recognize images a pagoda. There was a problem recognizing the pedestal stone of the 5,000 and 10,000 cycles pagoda. In 20,000 cycles recognized only Pagoda excluding the Pagoda's pedestal stone and supporters

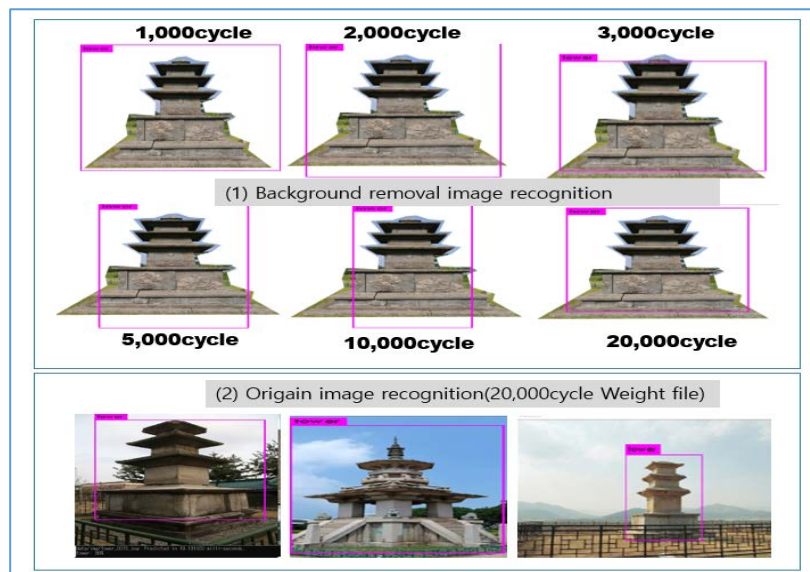


Fig. 12. Recognition rate by learning cycle

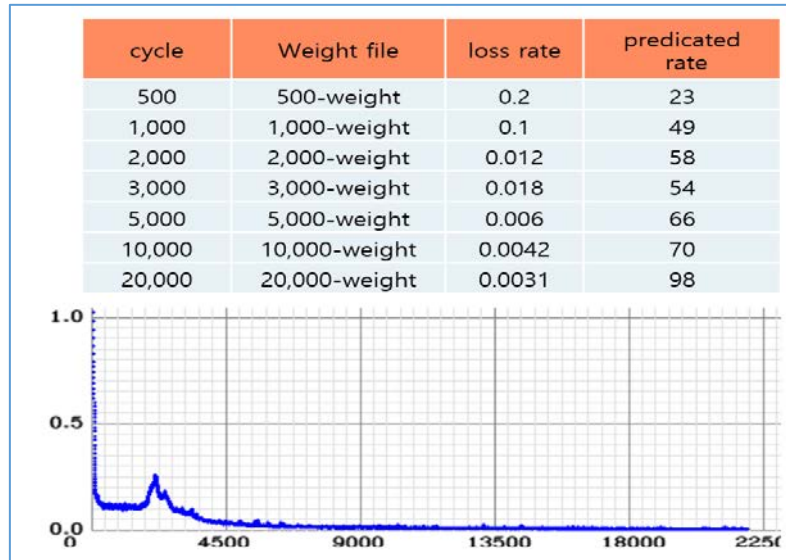


Fig. 13. Prediction rate by learning cycle

5. Conclusion

In this paper, a study was conducted to find the pagoda from the image using artificial intelligence learning about the traditional cultural property pagoda, which finds a large proportion of the cultural properties existing in various countries around the world. The process of finding a pagoda in the image of a cultural property has been accomplished through human intervention. Traditional cultural property restoration technology has studied the cornerstone of a method that can be restored from an objective standpoint, excluding the individual propensity of experts. First of all, Pagoda data was collected through an ancient archive database and internet gathering. The collected pagoda image data was classified by history and period. In the second step, the Pagoda image was marked using Yolo-Marker and the Pagoda dataset was constructed. In addition, in order to increase the recognition, rate the artificial intelligence learning environment was set to match the Pagoda dataset and the background of the Pagoda image was deleted to increase the predictability. As a final step, we experimented and evaluated using Darknet. As a result of the experiment, it was confirmed that artificial intelligence learning for Pagoda search is affected by input values and learning repetition as a method to reduce errors and increase recognition. it was confirmed that the prediction rate was 98% when repeated 20,000 times. This study is expected to be actively utilized in the part of restoring the pagoda. As a future research task, it is necessary to analyze the correlation between the original data and the matching rate with the learning data when restoring the pagoda.

Acknowledgement

This work was supported by the research grant of Pai Chai University in 2020.

References

- [1] Y. Lu, L. Zhang and W. Xie, "YOLO-compact: An Efficient YOLO Network for Single Category Real-time Object Detection," in *Proc. of 2020 Chinese Control And Decision Conference (CCDC)*, Hefei, China, pp. 1931-1936, 2020. [Article \(CrossRef Link\)](#)
- [2] W. Guo, Y. Li, W. Li and M. Sun, "Image-Based Modeling of Virtual Pagoda of China," in *Proc. of 2008 International Conference on Multimedia and Ubiquitous Engineering (mue 2008)*, Busan, pp. 9-14, 2008. [Article \(CrossRef Link\)](#)
- [3] W. Lan, J. Dang, Y. Wang and S. Wang, "Pedestrian Detection Based on YOLO Network Model," in *Proc. of 2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, Changchun, pp. 1547-1551, 2018. [Article \(CrossRef Link\)](#)
- [4] S. Chen and W. Lin, "Embedded System Real-Time Vehicle Detection based on Improved YOLO Network," in *Proc. of 2019 IEEE 3rd Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Chongqing, China, pp. 1400-1403, 2019. [Article \(CrossRef Link\)](#)
- [5] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. of 2017 International Conference on Engineering and Technology (ICET)*, Antalya, pp. 1-6, 2017. [Article \(CrossRef Link\)](#)
- [6] H. Ketout, J. Gu and G. Horne, "MVN_CNN and UBN_CNN for endocardial edge detection," in *Proc. of 2011 Seventh International Conference on Natural Computation*, Shanghai, pp. 781-785, 2011. [Article \(CrossRef Link\)](#)
- [7] Z. Xu, M. Strake and T. Fingscheidt, "Concatenated Identical DNN (CI-DNN) to Reduce Noise-Type Dependence in DNN-Based Speech Enhancement," in *Proc. of 2019 27th European Signal Processing Conference (EUSIPCO)*, A Coruna, Spain, pp. 1-5, 2019. [Article \(CrossRef Link\)](#)
- [8] Jin-Seob Kim, "A pitch-synchronous speech analysis and synthesis method for DNN-SPSS system," in *Proc. of 2016 IEEE International Conference on Digital Signal Processing*, pp.408-411, 2016. [Article \(CrossRef Link\)](#)
- [9] F. K. Noble, "Comparison of OpenCV's feature detectors and feature matchers," in *Proc. of 2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*, Nanjing, pp. 1-6, 2016. [Article \(CrossRef Link\)](#)
- [10] WY. Nie, P. Sommella, M. O'Nils, C. Liguori and J. Lundgren, "Automatic Detection of Melanoma with Yolo Deep Convolutional Neural Networks," in *Proc. of 2019 E-Health and Bioengineering Conference (EHB)*, Iasi, Romania, pp. 1-4, 2019. [Article \(CrossRef Link\)](#)
- [11] S. Carata, R. Mihaescu, E. Barnoviciu, M. Chindea, M. Ghenescu and V. Ghenescu, "Complete Visualisation, Network Modeling and Training, Web Based Tool, for the Yolo Deep Neural Network Model in the Darknet Framework," in *Proc. of 2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Cluj-Napoca, Romania, pp. 517-523, 2019. [Article \(CrossRef Link\)](#)
- [12] W. Blewitt, G. Ushaw and G. Morgan, "Applicability of GPGPU Computing to Real-Time AI Solutions in Games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 5, no. 3, pp. 265-275, Sept. 2013. [Article \(CrossRef Link\)](#)
- [13] X. Li et al., "COCO-CN for Cross-Lingual Image Tagging, Captioning, and Retrieval," *IEEE Transactions on Multimedia*, vol. 21, no. 9, pp. 2347-2360, Sept. 2019. [Article \(CrossRef Link\)](#)
- [14] A. M. Garcia, R. S. Requena, A. Alberich-Bayarri, G. García-Martí, M. Egea and C. M. Martinez, "Coco-Cloud project: Confidential and compliant clouds," in *Proc. of IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Valencia, pp. 227-230, 2014. [Article \(CrossRef Link\)](#).



Byong Kwon Lee received the B.S. degree in computer engineering from Hanbat University, Daejeon, Korea in 2000, and the M.S. degree in computer engineering from Hannam University, Korea in 2002, also Ph.D. degree in Dept. of IT engineering from Chungbuk National University, Korea in 2007. He has worked as an Assistant Professor in Dept. of Multimedia, College of Engineering Seowon University, in Korea. His current research interests AI computing, image processing, multimedia computing and XR(VR/AR/MR) System.



Soo Kyun Kim received Ph.D. in Computer Science & Engineering Department of Korea University, Seoul, Korea, in 2006. He joined Telecommunication R&D center at Samsung Electronics Co., Ltd., from 2006 and 2008. He worked a professor at Department of Game Engineering at Pai Chai University from 2008 to 2020. He is now a professor at Department of Computer Engineering at Jeju National University, Korea. Dr. Kim has published many research papers in international journals and conferences. Dr. Kim has been served as Chairs, program committee or organizing committee chair for many international conferences and workshops; Chair of ICCCT'11, ITCS'10, HumanCom'10, EMC'10, ICA3PP'10, FutureTech'10, ACSA'09, Em-Com'09, CSA'09, CGMS'09, ISA'09, SIP'08, FGCN'08 and so on. Also Dr. Kim is guest editor of the International Journal of "IET Image Processing" and "Multimedia Tools and Applications". His research interests include multimedia, pattern recognition, image processing, mobile graphics, geometric modeling, and interactive computer graphics. He is a member of ACM, IEEE, IEEE CS, KACE, KMMS, KKITS and KIIT.



Seokhun Kim received the M.S and Ph.D. degree in Computer Engineering from Hannam University in 2003 and 2006. He is an assistant professor Mobile Media at Suwon Women's University in from 2012 to 2017. Dr.Kim is currently an assistant professor in the Electronic Commerce at Paichai University. His teaching and research specialties are in the fields Mobile computing, Web-App programming, E-commerce System.