International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Artificial Intelligence-Powered Development of Location Image Analysis Algorithm using Image Crawling and Deep Learning

Jin-Wook Jang [1]

Submitted: 11/01/2024 **Revised**: 17/02/2024 **Accepted**: 25/02/2024

Abstract: The research study was conducted for development of the advanced image analysis service system based on deep learning. CNN (Convolutional Neural Network) is built in this system to extract learning data collected from Google and Instagram. The service gets a place image of jeju island as an input and provides relevant location information of it based on its own learning data. The process in 6 primary parts starts by collecting image data, converting it into appropriate format, and perform training and prediction, filtering invalid training data and repeats the Learning Part. Accuracy improvement plans are applied throughout this study. In conclusion, the implemented system shows about 79.2% of prediction accuracy. When the system has plenty of learning data, it is expected to predict various places more accurately.

Keywords: *Deep learning, Unsupervised Learning, Image Analysis, Convolutional Neural Network, location information, Location Image*

1. Introduction

This research study aims at providing a service system that itself collects and learns location images through a deep learning model. The system designed from this study is mainly categorized into three parts: Image Crawling, Image Learning, and Prediction parts. During the Image Crawling part, unfiltered data is gathered from Google and Instagram. Once the data collection is complete, it starts studying the place images through CNN, classifying them by appropriate location tags. Based on the learning model from CNN, it filters out irrelevant images within the collected data and repeats the learning and prediction procedures to extract the best model. Once the learning model is ready, the prediction procedure can also be utilized to provide location information of other place images. This research study focused on ten famous places in Jeju Island to test the performance of the system. Furthermore, statistical analysis was conducted to figure out how to further develop the program with better accuracy.

2. Relevant Research

2.1. Supervised Learning

Supervised Learning is one of three Machine Learning Types. Unlike other learning methods, it takes labeled dataset and predicts a correct label of unseen data [1]. Such learning model discovers the characters of each label and identifies/predicts unknown data by classifying it as appropriate label. Therefore, it usually manages classification problems and regression problems. If a supervised learning model has image data with labels, it is classified as a Deep learning model [2].

Since this study deals with image analysis to predict specific location, Supervised Learning is chosen to train the model.

2.2. CNN

CNN is a kind of Deep Learning Neural Network. The advantage of this neural network is that it generates a feature map of given images independently. Its learning result is calculated by matrix multiplication between the feature map filter and the trained image file converted to a matrix form.

Distinguished from other neural networks, CNN can process image learning without serialization. Since a color image is three-dimensional data, there would be loss or damage on the original file when it is serialized as one dimension: CNN does not cause such problem. The learning result from CNN has clearer feature map and higher accuracy than a learning model with any other neural network [3]. In other words, CNN is the most suitable neural network for image training. In this study, accuracy of image analysis is the key indicator to the system's performance, hence CNN was used.

[1] Department of Cooperative Digital Management, Agricultural Cooperative University, Republic of Korea
ORCID ID : 0000-0002-3605-9157
** Corresponding Author Email:jjw@nönghyup.ac.kr*

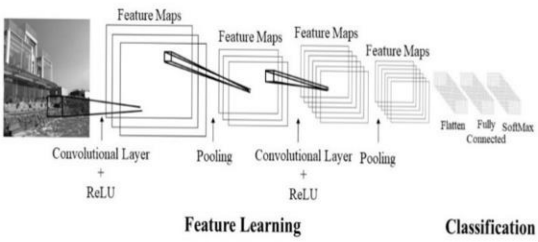

Fig. 1. Process of an Input Image Learning Process on CNN

The process of image learning on CNN can be summarized as the figure below, Fig. 1. This process requires various layers, with each layer allowing the deep learning model to train efficiently. Since the role of each layer differs, training process and accuracy would be changed depending on the structure of the CNN. Therefore, it is essential to understand how individual layers work and to concisely design an optimal network.

2.3. Max Pooling

The Pooling Layer is what reduces the risk of overfitting problem during the learning process. Max Pooling is one of the pooling methods, it extracts the maximum value of n*n matrix on an activation map (where n = stride size). Max Pooling Layers of CNN reduces the dimensions of an image, leaving the most important features only. This gives an optimized learning result [4].

2.4. ReLU

The Activation Layer calculates how well the CNN trains the learning model by selected loss function. When the system goes through each Activation Layer, loss value is calculated. The result from the function demonstrates the relevance of the training process and the input. Based on the result, the model determines whether to pass the input to the next node [5]. ReLU function is applied as its loss function in this study. The formula of the ReLU function is as Eq. (1). When an unnecessary value such as a negative integer is received, ReLU function returns a value of 0 unconditionally. This function allows the neural network to possess a significantly faster processing speed than the other [6, 7].

$$R(x) = \max(0, x) \tag{1}$$

2.5. TensorFlow

TensorFlow is a machine learning library and engine developed by Google. This has been used in various artificial intelligence fields such as Google search engine, speech recognition, translation, and automation features. By using TensorFlow, algorithms for image recognition, repetitive neural networks, and neural network learning can be easily implemented [8]. The library consists of arithmetic operations which is a useful library for this system as it needs to process matrix type files. In this study, Tensorflow was used for various calculation processing such as Max Pooling and Loss Function for image optimization and CNN for extracting feature maps.

2.6. Factorial Design and Two Way ANOVA Test

For further improvement, statistical experiment is designed and analyzed under Factorial Designs at two levels. To perform the design, there should be one dependent variable and two factors with a fixed number of levels of each factor. The experiment is repeated in all possible combinations of levels of each factor [9].

The Factorial Design shows the alteration of the dependent variable based on the two independent variables. Hence, the relationship between the interaction of two variables and the dependent factor should be analyzed. The Two way ANOVA test is a statistical test suitable to figure out such relationship.

The Two way ANOVA test has a quantitative response and two categorical factors. The initial null hypothesis would be that there is no relationship that affect the quantitative variable between the interaction of two categorical variables and a quantitative variable. Conversely, the alternate hypothesis would be the opposite, stating that there is such relationship. When the p-value on the test result is less than 0.05, the null hypothesis test can be rejected at 95% of significance level. The purpose of this test is to determine the joint relationship between the two factors with the response.[8] It tracks changes in the mean of the quantitative variable according to the levels of two categorical factors [10].

3. System Structure

The developed system consists of 6 major parts: Crawling, Preprocessing, Dataset, CNN, Filtering and Prediction Parts. Fig. 2 visualizes the algorithm of the model implemented in this research; it can be summarized into six steps as follows.

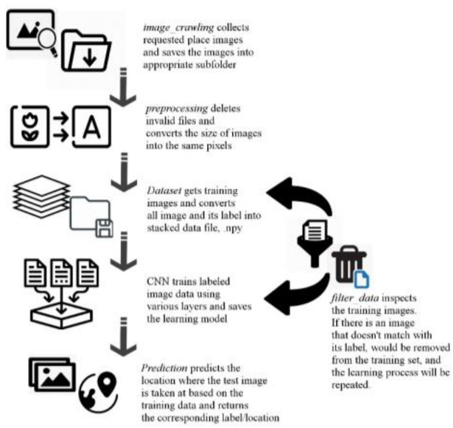

Fig. 2. Algorithm of Place Image Crawling, Learning, and Prediction Model

3.1. Image Crawling

The Crawling Part collects images which will be used in training data. Once the user inputs place names on an Excel file as Fig. 3, it creates new folders of each index and searches corresponding images of the places from Google and Instagram. Afterward, the images are saved into appropriate folders by their indices.

	A	B	C	D	E	F	G	H	I	J
1	0	1	2	3	4	5	6	7	8	9
2	하이엔드	원앤온리	몽상드애월	주상절리	천지연폭포	휴애리	성산일출봉	카멜리아힐	한라산	카페콜라

Fig. 3. Algorithm of Place Image Crawling, Learning, and Prediction Model

3.2. Preprocessing

This is where the system deals with the randomly collected images to avoid errors by data size or invalidation. The collected images are converted to an identical 240*240 pixels format while filtering invalid files. If the OS Error occurs while opening an image file with the error statement of "cannot identify image file", the dysfunctional file would be removed using try and except function, and an exception would arise to pass the error. Once the training data has the appropriate size and valid data, the next step can proceed.

3.3. Dataset

The Dataset Part converts the given image data to proper format for deep learning model. The system receives images which are labeled by the location. Once it loads the image files, these are stored as an .npy file with indices so that data can be easily used in the CNN Part. As MNIST file is provided by TensorFlow, the indices, training images and test images can be obtained separately from the .npy file which allows the deep learning model to conveniently call data for image learning.

3.4. CNN

The CNN Part processes actual image learning and returns training data at the end. Its filter goes through all unit of input image shifting as the size of stride at a time. This process reduces the size of the volume and extracts low level features (lines or dots), following Eq. (2) [11].

$$n_{out} = \frac{n_{in} + 2p - k}{s} + 1 \tag{2}$$

The training images go through a total of 72 layers, extracting features of images. The 24 layers out of 72 are Max Pooling Layers. These layers prevent the overfitting problem and assist in the learning process which can extract clear feature map. Another 24 layers are Activation Layers with ReLU function; they quickly extract extremely accurate features. The last of the layers are convolutional layers.

In this step, the model returns the training data as a .h5 format, Hierarchical Data Format, which contains advantages of supporting random access, fast search speed, and large quantity of data storage. This learning data file contains feature maps of each location tag, and the maps are what actually make decisions to predict location of a test image on the next step, the Prediction Part.

3.5. Prediction

The Prediction Part requests for the learning model, which was saved as .npy at the Dataset Part, and the test images of the location that the user wants to know. Based on the data, the model returns the most relevant place label for the each given image.

3.6. Filtering

This part is where the system enhances the learning model by removing the irrelevant data among the training images. The Filtering Part performs prediction on the current training images with its learning model. If the prediction fails to match any images and their original labels, the image would be removed from the training data. Once the faulty images are filtered, the system repeats the learning parts to extract a clearer learning model.

4. Substantiation

For testing, images of ten popular places in jeju island were collected and labeled as Fig. 2. An exact total of 3368 images were collected and properly stored in indexed subfolders. The saved images were trained by CNN from the deep learning model. The accuracy of the learning was 0.9994 and its loss values was 0.0015 as shown in Fig. 4, resulting a high learning accuracy of the model.

```
1472/1775 [=======================>......] - ETA: 28s - loss: 0.0016 - accuracy: 0.9993
1504/1775 [========================>.....] - ETA: 25s - loss: 0.0016 - accuracy: 0.9993
1536/1775 [========================>.....] - ETA: 22s - loss: 0.0016 - accuracy: 0.9993
1568/1775 [=========================>....] - ETA: 19s - loss: 0.0016 - accuracy: 0.9993
1600/1775 [==========================>...] - ETA: 16s - loss: 0.0015 - accuracy: 0.9994
1632/1775 [==========================>...] - ETA: 13s - loss: 0.0015 - accuracy: 0.9994
1664/1775 [===========================>..] - ETA: 10s - loss: 0.0015 - accuracy: 0.9994
1696/1775 [===========================>..] - ETA: 7s - loss: 0.0015 - accuracy: 0.9994
1728/1775 [============================>.] - ETA: 4s - loss: 0.0015 - accuracy: 0.9994
1760/1775 [============================>.] - ETA: 1s - loss: 0.0015 - accuracy: 0.9994
1775/1775 [==============================] - 168s 95ms/step - loss: 0.0015 - accuracy: 0.9994
```

Fig. 4. Accuracy and Loss of the learning process

To test the learning model, test images were randomly collected from Google and Instagram, which are different sources from the training data. For accurate testing, test images that were completely irrelevant to a given location were filtered out manually. With 500 test images and training data from CNN, prediction was performed. As shown in Fig. 5, the probability of that the index of test image and the prediction result exactly matched among the 500 predictions was 79.2%.

```
-----------------------------------------
396 predictions success out of 500 tests
It has a success rate of 79.2%
Finished predictions on jeju_images_filtered_size50 with google_first1
-----------------------------------------
```

Fig. 5. Success rate of the prediction

However, prediction results for each place shows some difference. Fig. 6 indicates that the prediction accuracy of the places indexed 3 and 4 are significantly low. This is because these two places are natural heritage sites and look similar, so the learning image data could not accurately distinguish them. In this case, the accuracy can be improved if there are sufficiently large and clear data to allow the model to see the minor differences between the similar places.

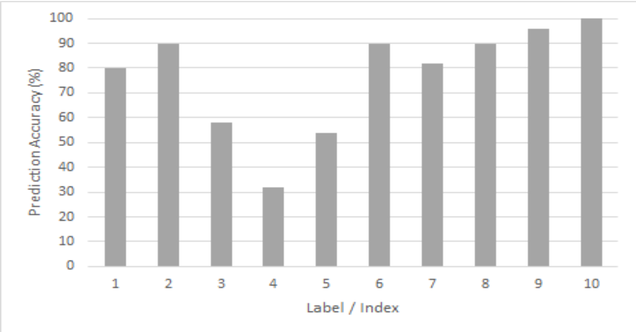

Fig. 6. Bar Graph of The Result of The Predictions

Theoretically, filtering irrelevant images with their labels was expected to improve the accuracy of the learning model. Contrary to the expectation, the Filtering Part does not change the accuracy. Removing more irrelevant images to the learning model causes the prediction accuracy to significantly fall. (see Fig. 7) This is because if the initial learning model gets the wrong data first and draws the draft of the feature map based on data, the system is unable to detect the irrelevant images and hence increasing the odds of deleting relevant images instead.

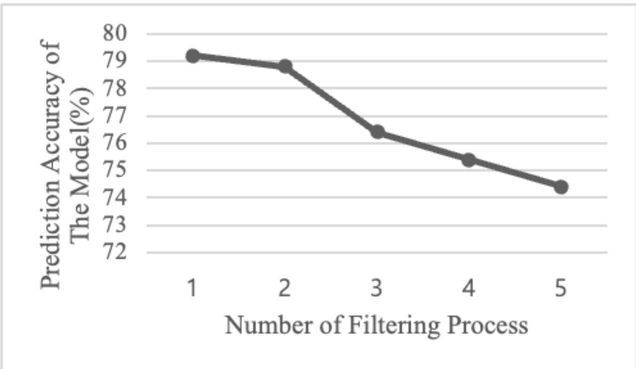

Fig. 7. Line Graph of The Prediction Accuracy Changes After n Times Filtering Procedures

To improve the prediction accuracy, initially well labeled dataset should be given to the learning model. Once clearly labeled dataset of more places is given, more place information and image data can be accurately classified. Consequently, as the amount of learning data increases and varies, the model can provide better prediction accuracy and more place information.

5. Analysis

To figure out how to improve the performance of the learning model, 2^k factorial experiment was conducted. The dependent variable is the accuracy of the prediction. The two independent factors are the number of its convolutional layers and the amount of learning image data. For each experiment, the number of the layers will be 72 or 144. The number of training image per a place will be 50 or 200 respectively. There are 4 different experiments, repeated at 5 times for higher reliability, using different test images.

To check the relationship between the two factors and the accuracy of the learning model, Two way ANOVA test is used on R Studio with the recorded data. The test result is shown as Fig. 7. According to it, only the number of learning image affects to the accuracy. The test between number of images and the accuracy only results significantly small p-value to reject the null hypothesis. In other words, neither the number of layers itself nor the two factors together have a strong relationship with the accuracy. However, there is a relation between the number of learning images and the accuracy.

```
                    Df Sum Sq Mean Sq F value  Pr(>F)
Num.Images           1 1748.5  1748.5  10.594 0.00497 **
Num.Layers           1  213.9   213.9   1.296 0.27174
Num.Images:Num.Layers 1   85.7    85.7   0.519 0.48155
Residuals           16 2640.6   165.0
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fig. 8. Result of Two way ANOVA Test

Consequently, larger image data gives higher accuracy to the learning model while the number of layers does not really change the result. Since the number of layers is not influential, we can decrease the running time without loss of the success rate of prediction by providing less layers to the deep learning model. This analysis concludes that the optimal deep learning model consists of a smaller number of layers, but with larger image data for the best performance.

6. Conclusion

The algorithm of the deep learning model is designed and implemented for the purpose of providing location information, in which jeju island was referenced for performance places. To implement a deep learning model via CNN, Max Pooling and ReLU functions are used for high accuracy and fast learning. In addition, image files are calculated and trained using TensorFlow. The process in 6 primary parts starts by collecting image data, converting it into appropriate format, and perform training and prediction, filtering invalid training data and repeats the

Learning Part.

In the testing, this model trains 3368 images of ten locations in Jeju and predicts 500 test images with an accuracy rate of 79.2%. The performance of the model could be enhanced when the model gets the correctly labeled dataset with more image data collected. Running time can be reduced, keeping the accuracy by setting a smaller number of layers. This model has great potential to be improved with better accuracy and be applied to any service where place image searching programs are needed.

Acknowledgements

This work was supported by The National Research Foundation of Korea in 2024(grant number 2022S1A5A8049255) and Cooperative Management Research Institude, Agricultual Cooperative University

Author contributions

Jin-wook Jang: Conceptualization, Methodology, Software, Field study, Data curation, Writing, Original draft preparation, Software, Validation., Field study, Visualization, Investigation, Writing, Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] AITUDE Homepage, https://www.aitude.com/supervised-vs-unsupervised-vs-reinforcement, last accessed 2021/05/13.

[2] Towards Data Science Homepage, https://towardsdatascience.com/what-is-machine-learning-a-short-note-on-supervised-unsupervised-semi-supervised-and-aed1573ae9bb, last accessed 2021/05/13.

[3] Michael A.: Neural Networks and Deep Learning. In: Chapter 6 Deep learning (2015).

[4] Prabhu Homepage, https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148, last accessed 2020/08/20.

[5] Torsa Talukdar Homepage, https://medium.com/@torsatalukdar11/why-do-we-need-activation-functions-in-neural-network-c72c340c78fa, last accessed 2020/08/22

[6] Pham, T.: CNN-based Facial Expression Recognition with New Loss Function., pp. 10. Department of Information & Telecommunication Engineering, Graduate School of Soongsil University, South Korea (2019).

[7] Machine Learning Mastery Homepage, https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/, last accessed 2020/08/20.

[8] Great Learning Homepage, https://www.mygreatlearning.com/blog/what-is-tensorflow-machine-learning-library-explained/#1.1., last accessed 2020/08/20.

[9] Taback, T.: Design of Experiments and Observation Studies, In: Factorial Design at Two Levels – 2^k Designs (2020).

[10] Evan, M. and Rosenthal, J.: Probability and Statistics: The Science of Uncertainty. 2nd edn. University of Toronto, Canada.Scribbr Homepage, https://www.scribbr.com/statistics/two-way-anova/, last accessed 2021/03/15.

[11] TensorFlow Homepage, https://www.tensorflow.org/tutorials/customization/basics., last ac-cessed 2021/01/23.

[12] C. Zhang and T. Akashi, "Fast affine template matching over Galois field," British Machine Vision Conference (BMVC), pp.121.1-121.11, 2015.

[13] M.A. Fischler and R.C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol.24, no.6, pp.381-395, 1981.

[14] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," SIAM Journal on Imaging Sciences (SIIMS), vol.2, no.2, pp.438-469, 2009.

[15] L.G. Brown, "A survey of image registration techniques," ACM Computing Surveys (CSUR), vol.24, no.4, pp.325-376, 1992.

[16] D. Nair, R. Rajagopal, and L. Wenzel, "Pattern matching based on a generalized Fourier transform," International Symposium on Optical Science and Technology, pp.472-480, International Society for Optics and Photonics, 2000.

[17] M.-S. Choi and W.-Y. Kim, "A novel two stage template matching method for rotation and illumination invariance," Pattern Recognit. (PR), vol.35, no.1, pp.119-129, 2002.

[18] H.Y. Kim and S.A. de Araújo, "Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast," Pacific Rim Conference on Advances in Image and Video Technology (PSIVT), pp.100-113, Springer-Verlag, 2007.

[19] A. Penate-Sanchez, L. Porzi, and F. Moreno-Noguer, "Matchability prediction for full-search template

matching algorithms," IEEE International Conference on 3D Vision (3DV), pp.353-361, 2015.

[20] C. Zhang and T. Akashi, "Simplifying genetic algorithm: A bit order determined sampling method for adaptive template matching," Irish Machine Vision and Image Processing Conference (IMVIP), pp.91-96, 2015.

[21] M. Gundam and D. Charalampidis, "Fourier transform-based method for pattern matching: Affine invariance and beyond," SPIE Defense+ Security, pp.94770I-94770I, International Society for Optics and Photonics, 2015.

[22] S. Korman, D. Reichman, G. Tsur, and S. Avidan, "FAsT-Match: Fast affine template matching," IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013, pp.2331-2338, 2013.

[23] F. Jurie and M. Dhome, "Real time robust template matching," British Machine Vision Conference (BMVC), pp.10.1-10.10, 2002.

[24] C. Zhang, Y. Yamagata, and T. Akashi, "Robust visual tracking via coupled randomness," IEICE Trans. Inf. & Syst., vol.E98-D, no.5, pp.1080-1088, May 2015.

[25] C. Zhang and T. Akashi, "High-speed and local-changes invariant image matching," IEICE Trans. Inf. & Syst., vol.E98-D, no.11, pp.1958-1966, Nov. 2015.

[26] J.H. Holland, Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence, University of Michigan Press, 1975.

[27] M. Hutter and S. Legg, "Fitness uniform optimization," IEEE Trans. Evol. Comput., vol.10, no.5, pp.568-589, 2006.

[28] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2010, pp.3485-3492, 2010.

[29] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," Int. J. Computer Vis. (IJCV), vol.88, no.2, pp.303-338, 2010.