ISSN: 0974-5823 Vol. 7 No. 1 January, 2022

International Journal of Mechanical Engineering

Classification of parking lot occupancy using deep learning

Jung Kyu Park[1], Eun Young Park[*2]

[1]Professor, Department of Computer Software Engineering, Changshin University, 51352, Korea

[*2]Professor, Rinascita College of Liberal Arts and Sciences, Shinhan University,11644, Korea

Abstract.
BACKGROUND/OBJECTIVES: Recently, due to urban development and rapid increase in the number of vehicles, a problem of lack of parking has occurred.
METHODS/STATISTICAL ANALYSIS: This causes parking problems in the city and affects the flow of vehicles. In order to solve the parking problem in front, a smart parking system is essential. In order to solve such a lack of parking lot problem, deep learning will be used to learn a large-capacity runner window image and check the availability of parking lot.
FINDINGS: Through preliminary research, we selected an algorithm that can be used for parking lot classification experiments among the existing deep learning and experimented using real images. The algorithm classified the image through tone change and color change. With this method, the parking lot classification result showed less than 90% accuracy. In this paper, we propose a parking lot classification method using deep learning. The deep learning proposed in the paper is composed of 11 layers of neural networks, and it learns and evaluates a database. The proposed deep learning algorithm learned many images captured in various weather environments. As a result of the experiment, it was found that the classification was classified with an accuracy of 94%. The proposed method showed similar performance to the Alexnet deep learning method.
IMPROVEMENTS/APPLICATIONS: In future research, the layer of deep learning will be improved. In addition, we will use various image databases to learn and measure the classification accuracy.

Keywords: Alexnet, Deep learning, Database, IoT, Smart Parking

1. INTRODUCTION

With the development of the city and the increase of single-person households, the number of cars has also increased, causing parking problems. Parking problems disrupt the flow of vehicles in the city and cause exhaust problems. In order to solve the parking problem in front, it is necessary to introduce a smart parking system. Smart parking systems can be more effective when linked with Intelligent Transportation Systems (ITS). In addition, IoT can be used to reduce traffic flow and parking problems. The most important issue in a smart parking lot is determining whether there are vehicles in the parking area [1-3]. As a previous study, Andrew proposed a smart parking system that can perform automatic parking and driver warning using Particle Filtering [4]. An automatic parking management system (APMS) that recognizes vehicle license plates using a template matching method was proposed [5].

In Kannan's study, a vehicle parking management system was implemented [6,7]. In the implemented system, an Arduino board and an ultrasonic sensor were used, and a Google map was used using Wi-Fi module in Android. The proposed system is designed to check whether the runner space is empty using IoT. The parking space information measured by IoT system with sensor and the space information is transmitted to the server for management. Rafael proposed a multi-camera system to map the parking space inside the parking lot and to check the presence or absence of a vehicle. The proposed system showed a classification accuracy of 90%. In addition, various smart parking systems have been studied [8,9].

In this paper, we look at the standard method used for parking lot classification [10,11]. It also displays the results of using the database and standard method used for learning. In this paper, we propose an algorithm that adds layers to improve the performance of *Alexnet*. And the layer used in the proposed algorithm will be described. Finally, to verify the performance of the proposed algorithm, we experimented with image data used in many experiments.

2. PKLOT DATABASE

The *pklot* database consists of approximately 12,000 photographs of two parking lots [11]. The photos were taken on cameras in two different parking lots. The photos are divided into various weather conditions and the image size is 1028x720. The first parking lot photos were taken from two locations. The total image file is provided as a compressed file and consists of a single file of about 4.6GB. The image file consists of three directories (parking1a, parking1b, parkking2). Each directory is divided into 3 sub-directories according to the weather (clear days, cloudy days, rainy days). In each sub-directory, directories are created by date. In each date directory, an XML file containing image files and parking space information (parking status, coordinates) is stored. About 700,000 parking space images can be obtained by navigating the parking space using the information in the XML file. Figure 1 shows the sample images of *pklot* database.

	
(a) parking lot 1a	**(b) parking lot 1b**

	`<parking id="pucpr">`
	`<space id="1" occupied="1">`
	`<rotatedRect>`
	`<center x="300" y="207" />`
	`<size w="55" h="32" />`
	`<angle d="-74" />`
	`</rotatedRect>`

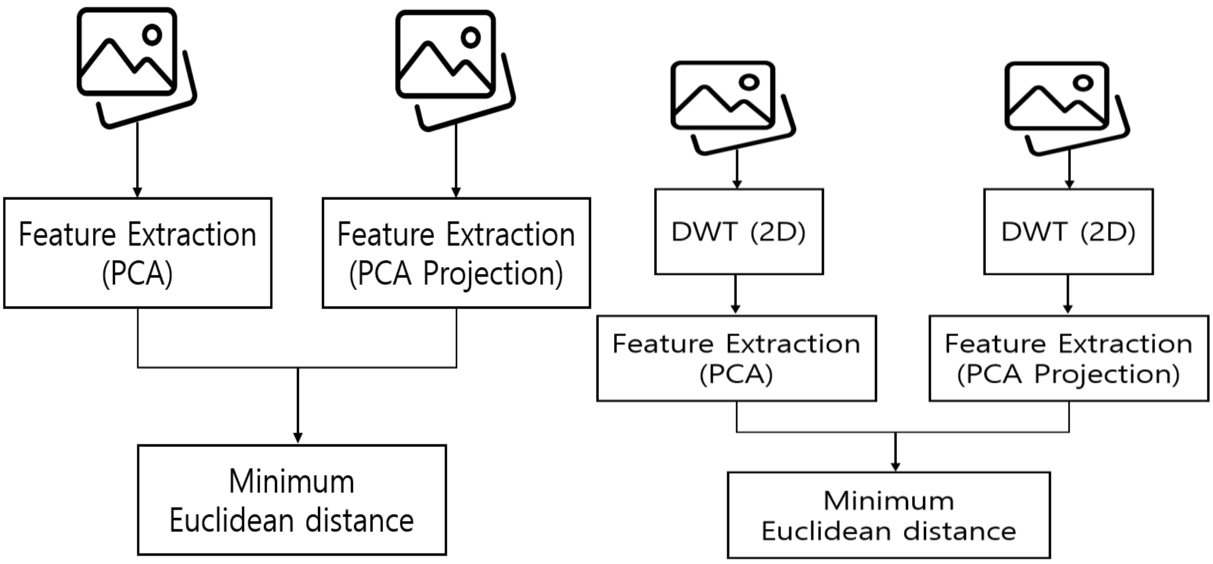
(c) parking lot 2	**(d) XML file of the image**

Figure 1. Sample images of *pklot* database

3. RELATED WORKS

Figure 2(a) shows how to train and test using PCA [12]. Figure 2(b) shows how to train and test using PCA and DWT. Table 3 shows different results when training and testing the previous two methods using different image databases. In the experiment, it was confirmed that the classification result was better when the first-stage DWT was used before PCA was applied. It was found that the classification rate increased by 82% or more when 80% of training images were learned from the image data.

In other experiments, an experiment was performed that omitted the training step. In this experiment, 10 was used as the threshold when converting the RBG image to grayscale. Then, it was converted into a black and white image using 12 as the next threshold. After that, the average value of the converted image is calculated, and if it is 0.581 or higher, it is classified as having an object in the area, and if it is below that, it is classified as an empty space. In this experiment, the classification accuracy was about 91% or higher. This classification experiment showed better results than the previously tested PCA and DWT. In addition, it was possible to reduce the total time by omitting the training step.



(a) PCA method **(b) DWT and PCA method**

Figure 2. Process of Training and Testing

Table 1: Test results of PCA and DWT

Image training (%)	Image testing (%)	PCA	PCA, DWT
10	90	62	72
20	80	74	74
30	70	40	59
40	60	74	65
50	50	25	76
60	40	33	50
70	30	74	52
80	20	64	81
90	10	41	72

4. PROPOSED METHOD

Deep learning is a field of artificial intelligence, which means a technology that can learn complex human intelligence such as vision and hearing with high accuracy. Recently, deep learning has been used to provide a level of accuracy similar to that of humans in image classification, speech recognition, and language processing. In the past, in order to solve the classification problem, a function to extract features that can identify an object from an image was used. The result of this feature extraction function is used as an input to the classification function and is used when determining an object. In order to classify a specific object, the following difficulties arise.

1. It is difficult to create a general, reliable and powerful function that can grasp a specific object.
2. It is difficult to create a function that responds to various shapes (transformation, rotation, size) of an object.
3. It is a difficult problem to select a set of elements that can determine the characteristics of each object.

The problems described above explain the difficulty of grasping and accurately classifying objects. Deep learning builds an artificial neural network to make decisions based on self-learning and learned content by organizing algorithms into layers. In deep learning, feature classification and tracking model are used to classify each object. In order to develop the model described above, we use data with a large amount of data. Models that learn large amounts of data can be used to classify not only objects learned through training, but also new objects that have not been trained. In particular, Convolutional Neural Network(CNN) is used a lot when learning and classifying images in deep learning. Figure 3 shows an architecture of CNN.

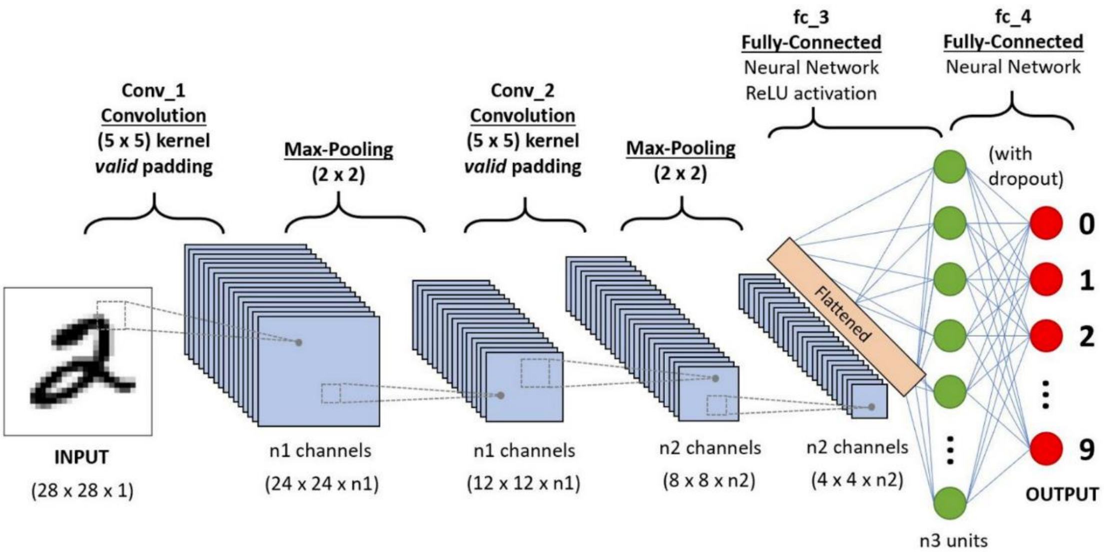


Figure 3. Architecture of CNN

CNN is a developed network for classifying images. So, be aware that it is optimized for images. And the image is made of pixels. After changing the color of the pixel into a number so that the computer can understand it, it is calculated through a neural network. The image classification sequence using machine learning before deep learning begins with catching the feature extractor of the image. HOG, SIFT, FAST, SURF, BRIEF... and so on were used to find features. After finding the features of the image, classification was conducted. Decision tree random forest, support vector machine, K near neighbor, Bayes, etc. were used for classification, and support vector machine was most used for image analysis prior to deep learning. It can be said that it took a lot of time because a person wrote an algorithm and proceeded with this method.

4.1. ALEXNET

AlexNet is a convolutional neural network (CNN) structure that won the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) competition held in 2012. It can be said that it is a structure that played a very large role in the revival of CNN. The basic structure of *AlexNet* is not very different from LeNet-5. The biggest change is that it is designed in a parallel structure in order to perform parallel operation with two GPUs. Figure 4 shows the structure of *AlexNet*. *AlexNet* consists of 8 layers. It consists of 5 convolutional layers and 3 full-connected layers. The second, fourth, and fifth convolution layers are connected only to the feature maps of the same channel in the previous step, while the third convolution layer is connected to all of the feature maps of the two channels in the previous step. Here are some of the important elements of *AlexNet*.

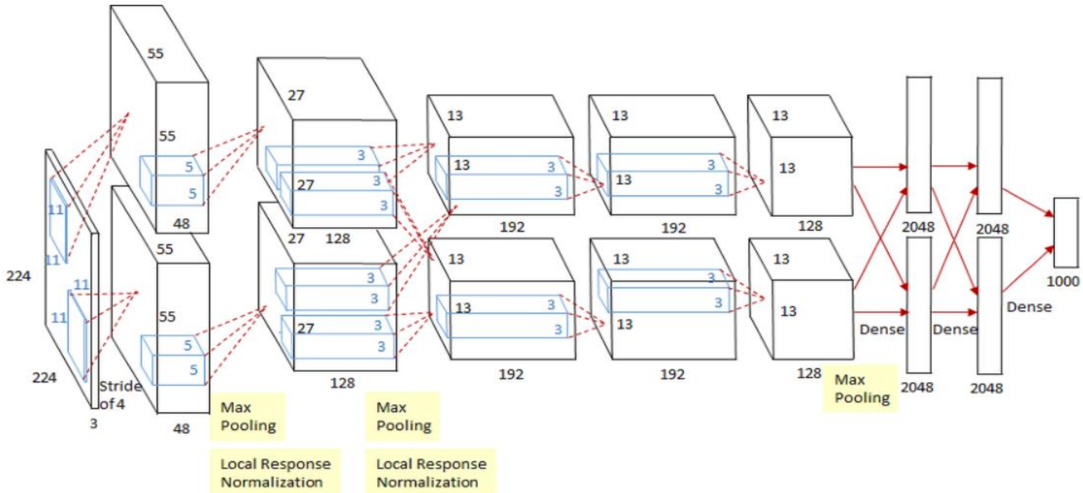


Figure 4. Architecture of AxlexNet

ReLU function: As the activation function, the ReLU function was used instead of the Tanh function used in LeNet-5. ReLU stands for rectified linear unit. It is said to be 6 times faster than using Tanh while maintaining the same accuracy. After *AlexNet*, it is preferred to use the ReLU function as an activation function. dropout: To prevent over-fitting, we used dropout, a type of regulatory technique. Dropout is the learning process while omitting some of the neurons in the fully-connected layer. It changes the value of some neurons to zero. Therefore, those neurons have no effect on forward pass and back propagation. The dropout is applied during training, and all neurons are used during testing. overlapping pooling: The role of pooling in CNN is to reduce the size of the feature map obtained through convolution. In the case of LeNet-5, average pooling was used, whereas in *AlexNet*, max pooling was used. In addition, in the case of LeNet-5, overlapping pooling was applied to make the stride, which is the stride of the pooling kernel, smaller than the kernel size. So, to be precise, LeNet-5 uses non-overlapping average pooling, and *AlexNet* uses overlapping maximum pooling.

4.2. PROPOSED METHOD

As a result of the pre-classification experiment, *AlexNet* showed low results, and this paper proposes a deep learning algorithm as shown in Figure 5. Layer 1 "input image" uses an image as an (160x160x3) input. Layer 2 "convolution" performs a 10x10 convolution of a 2d image. Next, 3x3 max pooling is performed at layer 4 "max pooling". In the other layer 5, 5x5 convolution is again performed, and in layer 6, 3x3 max pooling is performed. In layer 6, another 5x5 convolution is performed. In layer 7 "rectified liner", RelU activation function is executed for quick learning. In the next layer 8 "drop out", overfitting is improved by omitting a part of the network. In layer 9 "fully connected", neurons are turned off randomly in the middle to prevent learning. This prevents the phenomenon that learning is biased toward specific learning data. In layer 10 "soft max", when there are K values, the deviation of each value is enlarged so that large values are made relatively larger and small values are made relatively smaller, and then normalization is performed. In the last layer 11 "class output", it is decided whether or not an object is judged.

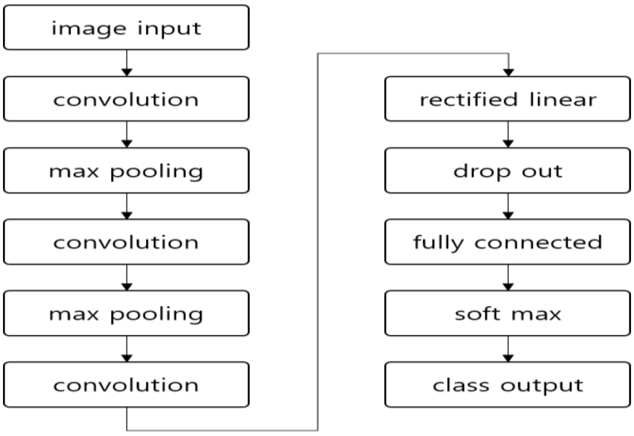


Figure 5. Proposed method for deep learning

Figure 6 shows the results of testing images on cloudy and rainy days after learning about 11,000 sunny days images from the pklot DB. In the case of sunny days, the accuracy of about 99% was achieved, but on cloudy and rainy days, the accuracy was 97%. Figure 3 is different from the experiment in Figure 7, and about 10,000 images of cloudy day were trained in the pklot database. After learning, image evaluation was performed on sunny, rainy, and cloudy days. When the algorithm proposed in the evaluation experiment was used, it showed an average of 93% classification accuracy on sunny, rainy, and cloudy days.

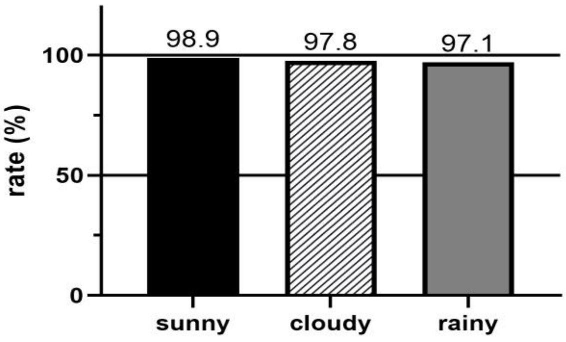


Figure 6. Performance test

(learning with 10,000 suuny images)

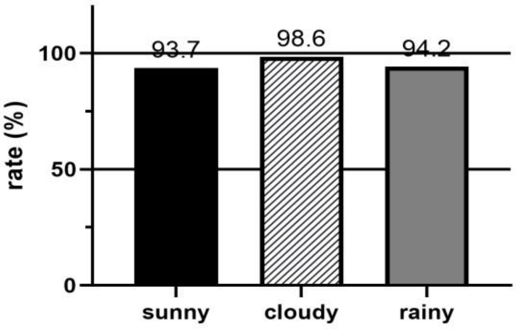


Figure 7. Performance test

(learning with 10,000 cloudy images)

In the following experiment, 11,000 images of rainy days were learned from the *pklot* database, and evaluations were performed on sunny, cloudy, and rainy days. Figure 8 shows the experimental results. Rainy days showed 94% classification results according to learning results, and cloudy days also showed high classification results. However, in the image evaluation on a sunny day, 88% of the fine classification results were shown.

In the last experiment, training was performed using the *Cnrall* database, not the *pklot* database. Figure 9 shows the experimental results. About 10,000 images were used for learning. After learning, the *pklot* database was evaluated. When evaluating the database of *pklot*, the average classification result was 83%. However, when evaluating the learned *Cnrall* database, an accuracy classification of about 94% was performed.

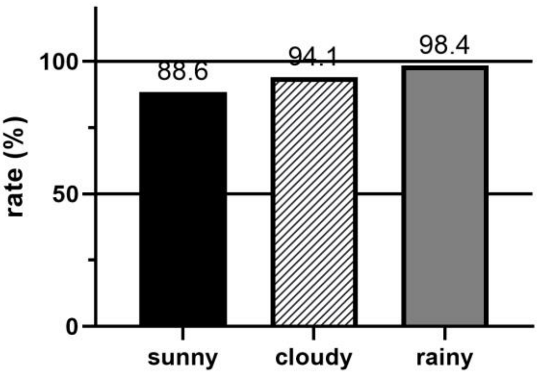


Figure 8. Performance test

(learning with 11,000 rainy images)

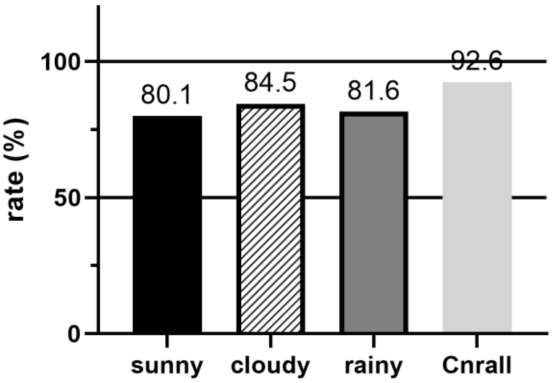


Figure 9. Performance test

(learning with 10,000 *Cnrall* images)

5. CONCLUSION

In this paper, we propose a parking lot classification method using deep learning. Previous method, the parking lot classification result showed less than 90% accuracy. The deep learning proposed in the paper is composed of 11 layers of neural networks, and it learns and evaluates a database. In the proposed deep learning algorithm, it was created by adding 3 more layers than *AlexNet*. The proposed deep learning algorithm learned many images captured in various weather environments. In order to analyze the performance of the algorithm, verified image databases such as *pklot* and *Cnrall* were used in the experiment. As a result of the experiment, it was found that the classification was classified with an accuracy of 94%. The proposed method showed similar performance to the *Alexnet* deep learning method.

6. ACKNOWLEDGMENT

This work was supported by Changshin University Research Fund of 2021-014

7. REFERENCES

1. Mahmoud MB, Wesam AA, Mostafa MF; Mohamed MEAM, Abdulah JA, Waleed A. Smart Parking System With Privacy Preservation and Reputation Management Using Blockchain. IEEE Access, 2020 Aug;8:150823-150843. DOI:10.1109/ACCESS.2020.3016945.
2. Jiazao L, Shi-Yong C, Chih-Yung C, Guilin C. SPA: Smart Parking Algorithm Based on Driver Behavior and Parking Traffic Predictions," IEEE Access, 2019 Mar;7:34275-34288. DOI: 10.1109/ACCESS.2019.2904972.
3. Fabian B, Sergio DM, Antonio O. Smart Parking: Using a Crowd of Taxis to Sense On-Street Parking Space Availability. IEEE Transactions on Intelligent Transportation Systems, 2020 Feb;21(2):496-508. DOI: 10.1109/TITS.2019.2899149.
4. Andrew M, Petros S, Konstantinos NP. Smart Parking System Based on Bluetooth Low Energy Beacons With Particle Filtering, IEEE Systems Journal, 2020 Sep;14(3):3371-3382.
5. Rafael MN, Álvaro G, Alexander GH, José MM. Automatic Vacant Parking Places Management System Using Multicamera Vehicle Detection. IEEE Transactions on Intelligent Transportation Systems, 2019 Mar;20(3):1069-1080. DOI:10.1109/TITS.2018.2838128.
6. Tajudeen OO, Carlos EO, Luis DO, Kehinde OO, Ivica K. Path Loss Models for Low-Power, Low-Data Rate Sensor Nodes for Smart Car Parking Systems. IEEE Transactions on Intelligent Transportation Systems, 2018 Jun;19(6):1774-1783. DOI:10.1109/TITS.2017.2741467.
7. Lee M, Kim S, Lim W, Sunwoo M. Probabilistic Occupancy Filter for Parking Slot Marker Detection in an Autonomous Parking System Using AVM. IEEE Transactions on Intelligent Transportation Systems, 2019 Jun;20(6):2389-2394. DOI:10.1109/TITS.2018.2855183.
8. Yu Y, Moon S, Sim S, Park S. Recognition of License Plate Number for Web Camera Input using Deep Learning Technique, Journal of Next-generation Convergence Technology Association, 2020 Dec;4(6):565-572. DOI:10.33097/JNCTA.2020.04.06.565.
9. Vikas H, Vikas S, Vinay C, Fei RY. A Parking Slot Allocation Framework Based on Virtual Voting and Adaptive Pricing Algorithm. IEEE Transactions on Vehicular Technology, 2020 Jun;69(6):5945-5957. DOI:10.1109/TVT.2020.2979637.
10. Kim H, Song E. Malicious Packet Filtering Scheme of Network Parameter on Internetwork Topology Using Deep Learning. Journal of Next-generation Convergence Technology Association, 2020 Aug;4(3):250-257. DOI:10.33097/JNCTA.2020.04.03.250
11. Paulo RLAA, Luiz SO, Alceu SB, Eunelson JS, Alessandro LK. PKLot – A robust dataset for parking lot classification. Expert Systems with Applications, 2015 Jul;42(11):4937-4949. DOI:10.1016/j.eswa.2015.02.009.
12. Matthew T, Alex P. Eigenfaces for recognition. Journal of Cognitive Neuroscience, 1991 Jan;3(1):71-86. DOI:10.1162/jocn.1991.3.1.71.