

재난약자 및 취약시설에 대한 APC실증에 관한 연구

Research on APC Verification for Disaster Victims and Vulnerable Facilities

김승용¹ · 황인철^{2*} · 김동식³ · 신정재⁴ · 용승갑⁵Seungyong Kim¹, Incheol Hwang^{2*}, Dongsik Kim³, Jungjae Shin⁴, Seunggap Yong⁵¹Professor, Department of Management Information System, Korea National University of Transportation, Chungju, Republic of Korea²Director, Secuware Inc., Cheongju, Republic of Korea³Vice Chairman, Terra International Management Company, Anyang, Republic of Korea⁴Adjunct Professor, Department of Graduate School of Arts & Design, Kyonggi University, Suwon, Republic of Korea⁵Department Manager, Secuware Inc., Cheongju, Republic of Korea

*Corresponding author: Incheol Hwang, ichwang@secuware.co.kr

ABSTRACT

Purpose: This study aims to improve the recognition rate of Auto People Counting (APC) in accurately identifying and providing information on remaining evacuees in disaster-vulnerable facilities such as nursing homes to firefighting and other response agencies in the event of a disaster. **Methods:** In this study, a baseline model was established using CNN (Convolutional Neural Network) models to improve the algorithm for recognizing images of incoming and outgoing individuals through cameras installed in actual disaster-vulnerable facilities operating APC systems. Various algorithms were analyzed, and the top seven candidates were selected. The research was conducted by utilizing transfer learning models to select the optimal algorithm with the best performance. **Results:** Experiment results confirmed the precision and recall of Densenet201 and Resnet152v2 models, which exhibited the best performance in terms of time and accuracy. It was observed that both models demonstrated 100% accuracy for all labels, with Densenet201 model showing superior performance. **Conclusion:** The optimal algorithm applicable to APC among various artificial intelligence algorithms was selected. Further research on algorithm analysis and learning is required to accurately identify the incoming and outgoing individuals in disaster-vulnerable facilities in various disaster situations such as emergencies in the future.

Keywords: Disaster-vulnerable Facilities, Victims, Auto People Counter, AI, Image Recognition

요약

연구목적: 본 연구는 요양병원 등 재난취약시설에 재난이 발생할 경우 잔류한 요구조자를 정확하게 파악하여 소방 등 대응기관에 제공하는 APC(Auto People Counting)의 인식률 개선에 목적이 있다. **연구방법:** 본 연구에서는 실제 재난취약시설에 설치되어 운영 중인 APC를 대상으로 카메라를 통해 출입 인원의 이미지를 인식하는 알고리즘을 개선하기 위해 CNN모형을 활용하여 베이스라인 모델링을 하였다. 다양한 알고리즘의 성능을 분석하여 상위 7개의 후보군을 선정하고 전이학습 모델을 활용하여 성능이 가장 우수한 최적의 알고리즘을 선정하는 방법으로 연구를 수행하였다. **연구결과:** 실험결과 시간과 성능이 가장 좋은 Densenet201, Resnet152v2 모델의 정밀도와 재현율을 확인한 결과 모든 라벨에 대해서 정확도 100%를 나타내는 것을 확인할 수 있었다. 이 중 Densenet201 모델이 더 높은 성능을 보여주었다. **결론:** 다양한 인공지능 알고리즘 중 APC에 적용할 수 있는 최적의 알고리즘을 선정하였다. 향후 연무 등 다양한 재난상황에서 재난취약시설 내 출입인원을 정확하게 파악할 수 있도록 알고리즘 분석 및 학습에 대한 추가 연구가 요구된다.

핵심용어: 재난취약시설, 요구조자, 자동계수기, 인공지능, 영상인식

Received | 19 February, 2024

Revised | 11 March, 2024

Accepted | 11 March, 2024

OPEN ACCESS



This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0>) which permits unrestricted noncommercial use, distribution, and reproduction in anymedium, provided the original work is properly cited.

© Society of Disaster Information All rights reserved.

서론

딥러닝의 성능을 입증하기 위해 AlexNet 모델은 대규모 이미지 데이터셋인 ImageNet에서의 이미지 분류로 가장 높은 성능을 달성한 모델이며 이후 딥러닝의 성능을 증명하고 활용 가능성을 높이는데 큰 영향을 미쳤다. AlexNet은 8개의 합성곱 계층과 3개의 완전 연결 계층으로 구성되어 있으며, ReLU 활성화 함수와 드롭아웃을 사용하여 과적합을 방지하도록 되어 있다(Krizhevsky et al., 2012). 또한 Karen Simonyan과 Andrew Zisserman에 의해 발표된 VGGNet이라는 딥러닝 아키텍처는 매우 깊은 신경망 구조로, 작은 필터 크기와 많은 합성곱(convolution) 계층을 사용하여 이미지 인식 분야에서 높은 성능을 달성했다. 특히, 이 연구에서는 3×3 크기의 작은 필터를 사용하여 신경망을 구성하는 것이 효과적임을 입증하였고 향후 지속적으로 VGGNet은 기존의 다양한 이미지 인식 작업에서 활발하게 사용되었다(Simonyan et al., 2014).

이 Inception 모델 연구는 GoogLeNet 보다 효율적인 계산과 더 깊은 신경망을 구현하기 위해 다양한 크기의 커널을 동시에 사용하는 모듈을 제안했다. 이 모듈은 네트워크가 다양한 크기의 특징을 추출할 수 있도록 하면서도 연산량을 효율적으로 줄이는 것이 목표이며 이미지 인식 및 다양한 컴퓨터 비전 작업에서 혁신적인 결과를 얻는데 사용되었다(Szegedy et al., 2014). 본 연구에서도 출입자의 이미지 처리가 빈번하기 때문에 이러한 검증된 연구가 필요하다.

네트워크를 더 깊게 만들면서도 그라디언트 소실 문제를 해결하기 위해 잔차 학습(Residual Learning) 개념을 도입한 ResNet(Residual Network)이라는 딥러닝 아키텍처가 연구되었다. 더 깊은 네트워크를 구축할 수 있고, 더 나은 이미지 인식 성능을 도출하였다. ResNet은 특히 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)와 같은 대규모 이미지 분류 대회에서 우수한 성과를 거두며 딥러닝 분야에서 큰 주목을 받았다(He et al., 2015).

DenseNet(밀집 연결 컨볼루션 네트워크)이라는 딥러닝 아키텍처는 ResNet의 개념을 확장하여 모든 레이어가 이전 레이어와 직접적으로 연결되는 구조를 의미하며 밀집한 연결 구조는 그라디언트 전파를 개선하고, 특징 재사용을 촉진하여 네트워크의 효율성과 성능을 향상시켰다. 또한 DenseNet은 네트워크의 매개 변수 수를 줄이고, 과적합을 방지하며, 더 깊은 네트워크를 효과적으로 학습할 수 있는 장점을 가지고 있어 딥러닝 분야에서 높은 인식을 받으며, 이미지 분류, 객체 감지, 분할 등 다양한 컴퓨터 비전 작업에서 널리 활용되고 있다(Huang et al., 2017).

EfficientNet은 네트워크의 깊이, 너비 및 해상도를 조정하여 모델을 확장하는 방법이다. 이를 통해 작은 모델부터 큰 모델까지 다양한 크기의 모델을 만들 수 있으며, 네트워크의 효율성과 성능을 동시에 개선한다. EfficientNet은 특히 리소스가 제한된 환경에서 효과적으로 사용할 수 있으며, 이미지 분류, 객체 감지, 분할 등 다양한 컴퓨터 비전 작업에서 뛰어난 성능을 보인다(Tan et al., 2019). 최근 화재탐지 분야에서 이미지 기반의 모델 YOLO와 동영상 기반의 모델 SlowFast 성능을 비교하여 화재 탐지 분야에 더 유효한 모델을 도출하는 연구가 진행되어 불꽃 이미지 탐지를 위한 모델을 평가하였다(Lee et al., 2023).

따라서 본 논문에서는 기존 이미지 데이터를 사용한 방식보다는 DenseNet이나 ResNet과 같이 이미지 처리에 탁월한 딥러닝 모델을 사용하여 이미지 파악의 정확성을 실험하고자 한다.

본론

본 연구에서 요양병원 등 재난취약시설 내 출입인원을 파악하기 위해 APC(Auto People Counter)를 적용하였다. Fig. 1과 Fig. 2와 같이 APC는 재난취약시설의 주요 출입문에 설치하였으며, 출입문을 통과하는 사람을 인식하여 건물 내 잔류인원을 파악하도록 하는 방식이다.

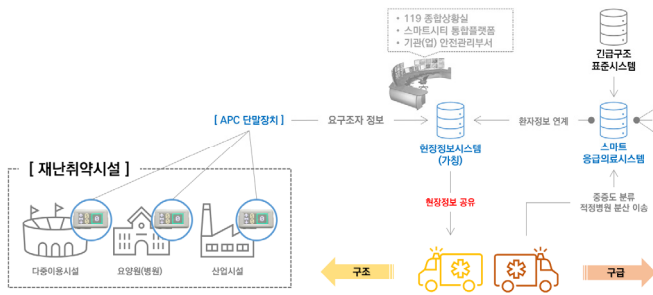


Fig. 1. Flowchart of rescue and rescue activities using APC

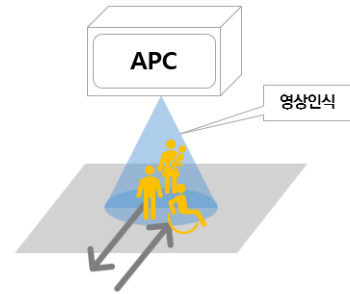


Fig. 2. In/Out peoples counting methods

출입문을 통과하는 다양한 이미지(사람, 휠체어, 기구 등) 중 사람을 특정하여 인식할 수 있도록 하기 위한 알고리즘이 매우 중요한 역할을 수행한다. 즉, 알고리즘의 인식률이 높아야 정확한 잔류인원을 파악할 수 있으며, 이는 실제 잔류인원이 없음에도 잘못된 정보로 인한 불필요한 구조작업으로 소방대원의 안전을 위협하는 것을 막을 수 있다.

출입영상 데이터로부터 직접 학습하는 딥러닝의 신경망 아키텍처인 CNN(Convolutional Neural Network) 모델을 이용하여 Table 1과 같이 베이스라인 모델링을 생성하여 모델의 성능을 확인한 결과 Fig. 3과같은 결과를 확인하였다.

Table 1. Baseline modeling with deep learning CNN model

Model: “Sequential”

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 32)	0
flatten (Flatten)	(None, 41472)	0
dense (Dense)	(None, 128)	5308544
dense_1 (Dense)	(None, 8)	1032

Total params: 5,319,720
 Trainable params: 5,319,720
 Non-trainable params: 0

데이터를 불러오고 이미지 전처리 작업을 거쳐서 베이스라인 모델을 설정 후 모델링을 완성하였다.

Initializing the CNN → # Step 1 - Convolution → # Step 2 - Pooling → # Adding convolutional layer → # Step 3 - Flattening → # Step 4 - Full Connection → # Step 5 - Output Layer → # Compiling the CNN

상기 모델의 성능을 높이기 위해 데이터 증강(Data Augmentation)으로 이미지를 로드하여 회전각도, 확대축소, 좌우이동, 상하이동, 좌우반전 등을 데이터 증강의 많은 기법을 사용하여 Train과 Val 데이터에 적용했고 Test 데이터는 학습에 사

용되지 않기 때문에 증강 기법을 적용하지 않았다.

```

Epoch 1/10
173/173 [=====] - 71s 408ms/step - loss: 0.3905 - accuracy: 0.8571 - val_loss: 0.3849 - val_accuracy: 0.8620
Epoch 2/10
173/173 [=====] - 72s 416ms/step - loss: 0.3130 - accuracy: 0.8848 - val_loss: 0.4181 - val_accuracy: 0.8526
Epoch 3/10
173/173 [=====] - 72s 416ms/step - loss: 0.2762 - accuracy: 0.9018 - val_loss: 0.3455 - val_accuracy: 0.8787
Epoch 4/10
173/173 [=====] - 72s 419ms/step - loss: 0.2331 - accuracy: 0.9142 - val_loss: 0.3913 - val_accuracy: 0.8686
Epoch 5/10
173/173 [=====] - 72s 417ms/step - loss: 0.2100 - accuracy: 0.9218 - val_loss: 0.2798 - val_accuracy: 0.9027
Epoch 6/10
173/173 [=====] - 73s 422ms/step - loss: 0.1925 - accuracy: 0.9263 - val_loss: 0.2551 - val_accuracy: 0.9136
Epoch 7/10
173/173 [=====] - 73s 419ms/step - loss: 0.1765 - accuracy: 0.9317 - val_loss: 0.2838 - val_accuracy: 0.8998
Epoch 8/10
173/173 [=====] - 73s 424ms/step - loss: 0.1598 - accuracy: 0.9346 - val_loss: 0.3475 - val_accuracy: 0.8903
Epoch 9/10
173/173 [=====] - 73s 422ms/step - loss: 0.1699 - accuracy: 0.9341 - val_loss: 0.3775 - val_accuracy: 0.8932
Epoch 10/10
173/173 [=====] - 73s 421ms/step - loss: 0.1365 - accuracy: 0.9488 - val_loss: 0.3338 - val_accuracy: 0.8889
<keras.callbacks.History at 0x1f0feac59aD>
    
```

Fig. 3. Performance results of baseline model using CNN model

실험결과

전이학습(Transfer Learning)을 사용해서 모델 성능을 높이고 전이학습 모델을 다양한 모델 함수에 학습을 진행하여 Table 2와 같은 결과를 얻었다.

Table 2. Analysis model learning results using transfer learning

	Model	Val_Accuracy	Accuracy	Training Time(sec)
0	DenseNet201	1.0000	1.0000	49.20
1	ResNet152V2	0.9919	1.0000	58.44
2	ResNet50V2	0.9919	1.0000	21.12
3	DenseNet121	1.0000	0.9964	1197.90
4	MobileNetV2	0.9960	0.9964	16.69
5	InceptionV3	0.9798	0.9964	25.27
6	Xception	1.0000	0.9964	32.77
7	VGG16	0.9677	0.9674	46.58
8	MobileNetV3Large	0.5806	0.5326	18.55

학습한 모델에 Test셋을 통해서 전이모델별 성능을 확인했는데 정확도가 대부분 1에 수렴하는 것을 볼 수 있다. 정확도를 시각화해 보았을 때 Fig. 4와 같이 상위 7개는 1에 가까운 값을 나타냈다.

VGG19모델처럼 시간만 오래 걸리고 성능은 좋지 못한 모델이 있기 때문에 가장 좋은 성능을 내고 시간도 적게 드는 Densenet201과 Resnet152v2를 통해서 좋은 효율을 내는 모델 성능을 확인해 보았다(Fig. 5, Fig. 8).

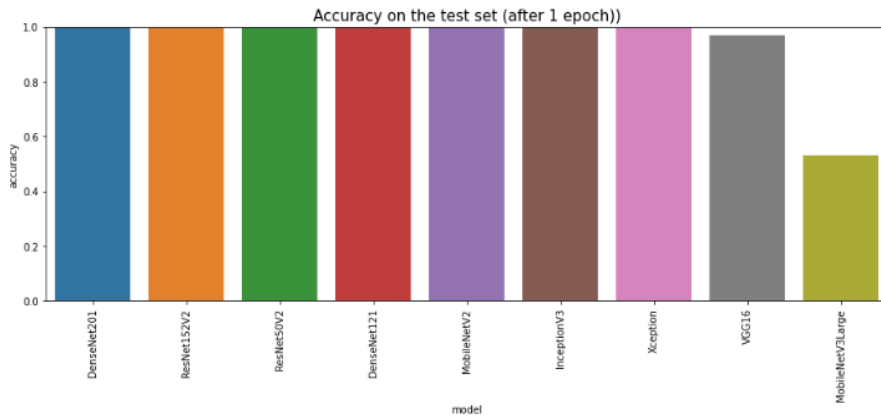


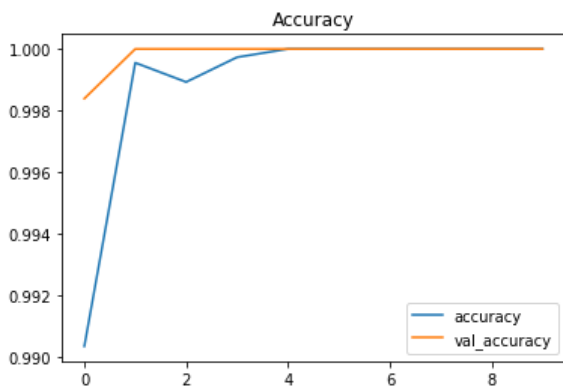
Fig. 4. Performance results by transfer model

```

Found 5589 validated image filenames belonging to 8 classes.
Found 620 validated image filenames belonging to 8 classes.
Found 690 validated image filenames belonging to 8 classes.
Epoch 1/7
175/175 [=====] - 24s 117ms/step - loss: 0.1121 - accuracy: 0.9707 - val_loss: 0.0133 - val_accuracy: 0.9952
Epoch 2/7
175/175 [=====] - 20s 111ms/step - loss: 0.0084 - accuracy: 0.9982 - val_loss: 0.0145 - val_accuracy: 0.9952
Epoch 3/7
175/175 [=====] - 19s 110ms/step - loss: 0.0130 - accuracy: 0.9961 - val_loss: 0.0137 - val_accuracy: 0.9935
Epoch 4/7
175/175 [=====] - 19s 111ms/step - loss: 0.0020 - accuracy: 0.9995 - val_loss: 0.0168 - val_accuracy: 0.9952
Epoch 5/7
175/175 [=====] - 19s 111ms/step - loss: 0.0017 - accuracy: 0.9995 - val_loss: 0.0296 - val_accuracy: 0.9952
Epoch 6/7
175/175 [=====] - 19s 111ms/step - loss: 2.648e-04 - accuracy: 1.0000 - val_loss: 0.0102 - val_accuracy: 0.9952
Epoch 7/7
175/175 [=====] - 19s 111ms/step - loss: 7.1476e-05 - accuracy: 1.0000 - val_loss: 0.0078 - val_accuracy: 0.9968
    
```

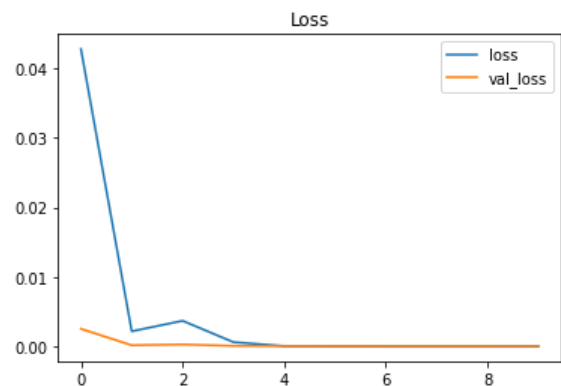
Fig. 5. DenseNet201 performance check

test 데이터를 통해서 성능을 확인한 결과 Densenet모델은 99.71%의 예측률을 나타내었다(Fig. 6, Fig. 7).



Accuracy on the test set : 99.71%

Fig. 6. Densenet201 accuracy



Accuracy on the test set : 99.71%

Fig. 7. Densenet201 loss

```

Found 5589 validated image filenames belonging to 8 classes.
Found 620 validated image filenames belonging to 8 classes.
Found 690 validated image filenames belonging to 8 classes.
Epoch 1/7
175/175 [=====] - 24s 117ms/step - loss: 0.1121 - accuracy: 0.9707 - val_loss: 0.0133 - val_accuracy: 0.9952
Epoch 2/7
175/175 [=====] - 20s 111ms/step - loss: 0.0084 - accuracy: 0.9982 - val_loss: 0.0145 - val_accuracy: 0.9952
Epoch 3/7
175/175 [=====] - 19s 110ms/step - loss: 0.0130 - accuracy: 0.9961 - val_loss: 0.0137 - val_accuracy: 0.9935
Epoch 4/7
175/175 [=====] - 19s 111ms/step - loss: 0.0020 - accuracy: 0.9995 - val_loss: 0.0168 - val_accuracy: 0.9952
Epoch 5/7
175/175 [=====] - 19s 111ms/step - loss: 0.0017 - accuracy: 0.9995 - val_loss: 0.0296 - val_accuracy: 0.9952
Epoch 6/7
175/175 [=====] - 19s 111ms/step - loss: 2.4648e-04 - accuracy: 1.0000 - val_loss: 0.0102 - val_accuracy: 0.9952
Epoch 7/7
175/175 [=====] - 19s 111ms/step - loss: 7.1476e-05 - accuracy: 1.0000 - val_loss: 0.0078 - val_accuracy: 0.9968
    
```

Fig. 8. ResNet152V2 performance check

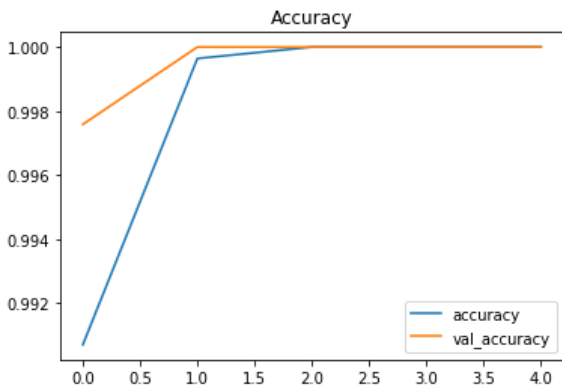


Fig. 9. ResNet152V2 accuracy

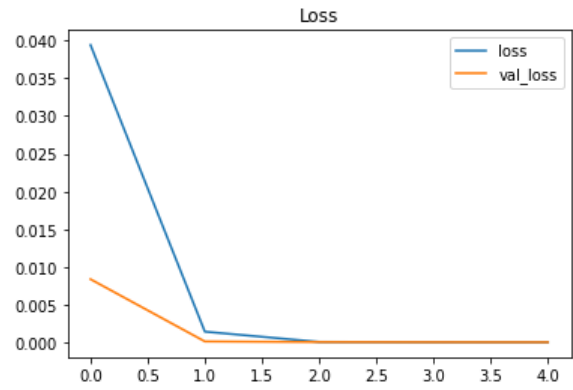


Fig. 10. ResNet152V2 loss

Fig. 9와 Fig. 10에서 보는 바와 같이 ResNet모델은 epochs 옵션을 5로 지정하고 학습을 진행하였고 epochs 옵션 2에서는 1, Loss는 0에 수렴하였다(Accuracy on the test set : 100.00%).

Table 3. Precision and recall results of Densenet201 and Resnet52v2 models

	Precision	Recall	F1-Score	Support
Airplane	1.00	1.00	1.00	167
Car	1.00	1.00	1.00	191
Cat	1.00	1.00	1.00	146
Dog	1.00	1.00	1.00	162
Flower	1.00	1.00	1.00	157
Fruit	1.00	1.00	1.00	210
Motorbike	1.00	1.00	1.00	159
Person	1.00	1.00	1.00	188
Accuracy			1.00	1380
Macro Avg	1.00	1.00	1.00	1380
Weight Avg	1.00	1.00	1.00	1380

시간과 성능이 가장 좋은 Densenet201, Resnet152v2 모델의 정밀도와 재현율을 확인해 보면 Table 3과 같이 모든 라벨에 대해서 정확도 100%를 나타내는 것을 확인할 수 있다.

결론

본 연구는 재난취약시설 내 출입 인원을 자동으로 집계하여 화재 등 재난 시 대응기관에 정확한 정보를 제공하여 신속한 구조 및 구조대원의 안전을 확보하고자 설치·운영 중인 APC의 인식률 개선에 목적을 두었다. 출입 인원을 정확하게 인식하기 위한 다양한 알고리즘을 선정하고 학습한 결과 최적의 인공지능 알고리즘을 선정할 수 있었다.

선정된 최적의 인공지능 알고리즘은 실제 운영 중인 APC에 적용하여 출입 인원에 대한 정확한 인식과 집계를 통해 재난 시 요구조자에 대한 정보를 대응기관에 제공하여 신속하고 안전한 재난 대응이 가능할 것으로 기대한다. 향후 연무, 조명 등 다양한 재난상황에서 APC의 인식률을 개선하는 연구를 진행할 계획이다.

Acknowledgement

본 연구는 2023년 한국교통대학교의 지원을 받아 수행하였음.

References

- [1] He, K., Zhang, X., Ren, S., Sun, J. (2015). "Deep residual learning for image recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, pp. 770-778.
- [2] Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q. (2017). "Densely connected convolutional networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, pp. 4700-4708.
- [3] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). "ImageNet classification with deep convolutional neural networks." Communications of the ACM, Vol. 60 No. 6, pp. 84-90.
- [4] Kwon, H., Youn, C.J., Kim, H.D. (2021). "Necessity of applying lifesaving smart technology for vulnerable class in nursing facilities: Focus on introduction of RFID-based positioning system and 3D scanning technology." Fire Science and Engineering, Vol. 35, No. 2, pp. 143-154.
- [5] Lee, J.R., Lee, D.W., Jeong, S.H., Jeong, S. (2023). "Comparative study on artificial intelligence model performance between image and video recognition in the fire detection area." Journal of the Society of Disaster Information, Vol. 19, No. 4, pp. 968-975.
- [6] Simonyan, K., Zisserman, A. (2014). "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556.
- [7] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2014). "Going deeper with convolutions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, pp. 1-9.
- [8] Tan, M., Le, Q. (2019). "EfficientNet: rethinking model scaling for convolutional neural networks." Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, USA, Vol. 97, pp. 6105-6114.