Design and Implementation of Smart Indicators based Emergency Evacuation System for Smart Buildings

Van-Quyet Nguyen
Faculty of Information Technology,
Hung Yen University of Technology and Education
16000 Hungyen, Vietnam
quyetict@utehy.edu.vn

Huy-The Vu
Faculty of Information Technology,
Hung Yen University of Technology and Education
16000 Hungyen, Vietnam
thevh@utehy.edu.vn

Van-Hau Nguyen
Faculty of Information Technology,
Hung Yen University of Technology and Education
16000 Hungyen, Vietnam
haunv@utehy.edu.vn

Thai-Bao Mai-Hoang
Faculty of Information Technology,
Hung Yen University of Technology and Education
16000 Hungyen, Vietnam
maihoangthaibao01@gmail.com

Kyungbaek Kim
Department of Artificial Intelligence Convergence, Chonnam National University
61186 Gwangju, South Korea
kyungbaekkim@jnu.ac.kr

ABSTRACT

During a fire event, an emergency evacuation system is an indispensable system in large buildings, which guides evacuees to exit gates as fast as possible by dynamic and safe routes. This paper presents a design and implementation of an emergency evacuation system that uses a dynamic evacuation routing approach and smart indicators as edge computing nodes and then directs the evacuees to exit gates. The proposed system is designed as a distributed system with multiple layers of computing that provides an efficient routing approach using *partial view information* which represents the hazard intensity and the crowd congestion information of a group of sections/floors in the building. We design smart indicators to capture people density using a pre-trained convolutional neural network model, track danger areas using temperature and smoke sensors, and show directions. The gathered information from smart indicators is provided to the smart guidance agents via a Web API for finding effective routes. To validate our approach, we implement a simulator to compare our evacuation routing approach with baseline approaches. Experimental results show that our approach reduces up to 25% of the total evacuation time compared with others. Furthermore, through the results of initial smart indicator implementation, which can interact with the simulator, we show the viability of our proposed system.

KEYWORDS

Smart Indicator, Smart Building, Emergency Evacuation System

1 INTRODUCTION

As a result of the development of the Internet of Things (IoT) as well as many achievements of modern technology, smart buildings have been coming to reality with the support of multiple smart devices such as smart indicators, smart sensors, and smart cameras. These smart devices can help building management systems to gather the essential information to make the right decisions in emergency situations such as fire building events. In which, emergency evacuation systems play an important role in safely evacuating people to the exit gate as soon as possible. One of the main roles of emergency evacuation systems is finding effective evacuation routes, which is not a trivial problem due to the uncertainty of the hazardous conditions and the possibility of congestions, to reduce evacuation time and help people pass through exit gates as fast as possible.

There have been several studies targeting the weighted graph based approaches using IoT data in smart buildings to dynamically find the evacuation routes more effectively as the situations can easily changes in such conditions [1][2][3]. Nikolaos et al. [1] proposed a building evacuation system that evaluated the optimal evacuation routes in real-time based on updating the hazard intensity between decision nodes (indicators). Sai-Keung et al. [2] proposed an optimized evacuation route algorithm based on crowd simulation using division points that divide evacuees into two groups and evacuate in opposite directions. Marin et al. in [3] proposed a distributed evacuation guidance for large smart buildings with building conditions and hazard intensity consideration based on a smart sensor network and personal mobile devices.

Most evacuation systems considered distributed approaches to find evacuation routes due to massive data generated by a large number of devices (e.g., sensors, cameras) [3]. Alfredo et al. [4] proposed a strategy using distributed client-server system to compute the suitable routes for individuals depending on their health and their locations in a small city. Recently, Van-Quyet et al. [5] proposed a distributed system by implementing the LCDT-based (Length, Capacity, Density, and Trustiness) algorithm and a caching strategy. The authors considered global view information, which is about the hazard intensity and the crowd congestion in the whole building, to build the evacuation routing algorithm. However, there is still room for improvement, because (1) using global view information might not efficient for finding evacuation routes in a high-rise building (e.g., the evacuation routes on the tenth floor might be not affected congestion at the second floor in a short period time) and (2) implementing of weight calculation in Smart Guidance Agents might take a high cost (i.e. processing time) due to handling of massive data gathered cameras and sensors in a short time.

In this paper, we presents a design and implementation of an emergency evacuation system that uses a dynamic evacuation routing approach and smart indicators as edge computing nodes, and then direct the evacuees to exit gates. The proposed system is designed as a distributed system with multiple layers of computing that provides an efficient routing approach using *partial view information* which represents the hazard intensity and the crowd congestion information of a group of sections/floors in the building. We design smart indicators to capture people density using a pre-trained convolutional neural network model, track danger areas using temperature and smoke sensors, and show directions. The information gathered from smart indicators is provided to the smart guidance agents via a Web API for finding effective routes.

Our work as described in this paper makes the following contributions.

- We propose a design of an emergency evacuation system with multiple computation layers using the information provided by smart indicators.
- We propose a design and implementation of smart indicators for smart buildings using IoT devices and applying a CNN-based image processing technique.
- We propose an approach for dynamic evacuation routing, so-called LCDT&PV, which improves the LCDT-based approach [5], by using partial view information.
- We implement a Web API for communication between smart indicators and smart guidance agents.
- We implement a simulator to compare our evacuation routing approach with baseline approaches. Experimental results show that our approach reduces up to 25% of the total evacuation time compared with others.

We next present related work on emergency evacuation systems in the following section.

2 RELATED WORK

There have been several techniques targeting the smart evacuation system. Nikolaos et al. [1] proposed a building evacuation system that computes the optimal evacuation routes in a real-time manner. The authors use smart sensors to obtain physical length and hazard intensity, then use them to make a weighted graph. Based on network flow techniques, Dressler et al. [6] proposed a method to determine the right exit gates for evacuees during emergency situations. Abdelghany et al. [7] proposed a framework to evacuate people in large-scale pedestrian facilities with multiple exit gates. Several designs of intelligent indoor emergency evacacuation systems are presented in [8] [9].

In order to reduce casualties because of lack of information due to suddenly changed situations in disaster events such as building ablaze, Lee et al. [10] proposed a new paradigm to develop assistance technology. Whereby the systems developed based on paradigm using many IoT sensors integrated inside the buildings to collect data about hazard signs such as smoke or fire. The system then used a machine learning algorithm to determine the safe routes for evacuees. Alfredo et al. [4] took advantage of a distributed client-server system to create a strategy to compute the suitable routes for individuals depending on their health and their locations in a small city amid disaster events like flood, storm or fire. In which, information of each individual is collected by client software and server computes best routes and recommend to evacuees via the client software. However, this strategy did not consider hazardous conditions as well as congestion routes. Zualkernan et al. [11] proposed an IoT-based emergency evacuation system to obtain information about situations inside a building during fire emergency in order to reduce turmoil and guides evacuees to safe exits. The system utilizes smartphones to recommend the safe routes to evacuees. In case evacuees have no smartphone, the exit sign can dynamically change their state in accordance with the situation. Thus, although there have been many studies focusing on the design and implementation of emergency evacuation systems, the development of smart evacuation systems is still in its infancy stage.

3 PROPOSED EVACUATION SYSTEM

This section presents a design and implementation of our proposed emergency evacuation system for smart buildings.

3.1 Design of System Architecture

Figure 1 presents the architecture of our proposed emergency evacuation system. The system consists of three main modules: *Smart Indicator*(s), *Smart Guidance Agent*(s) (SGAs), and *Smart Coordinator*. In which, the *Smart Indicators* interact *with Smart Guidance Agents* via a Web API using the HTTP protocol. While *Smart Guidance Agents* work with *Smart Coordinator* in the same application, namely *Smart Guidance Application*. We describe more details about these modules as follows.

Smart Indicators. A smart indicator is responsible for calculating the weight of a road segment (e.g., corridor segment) where it is located and showing the direction(s) to guide evacuees to pass through a safe exit gate. Here, the weight of a road segment represents the impact of building conditions and disaster conditions on that segment. It is defined in [5] as the following:

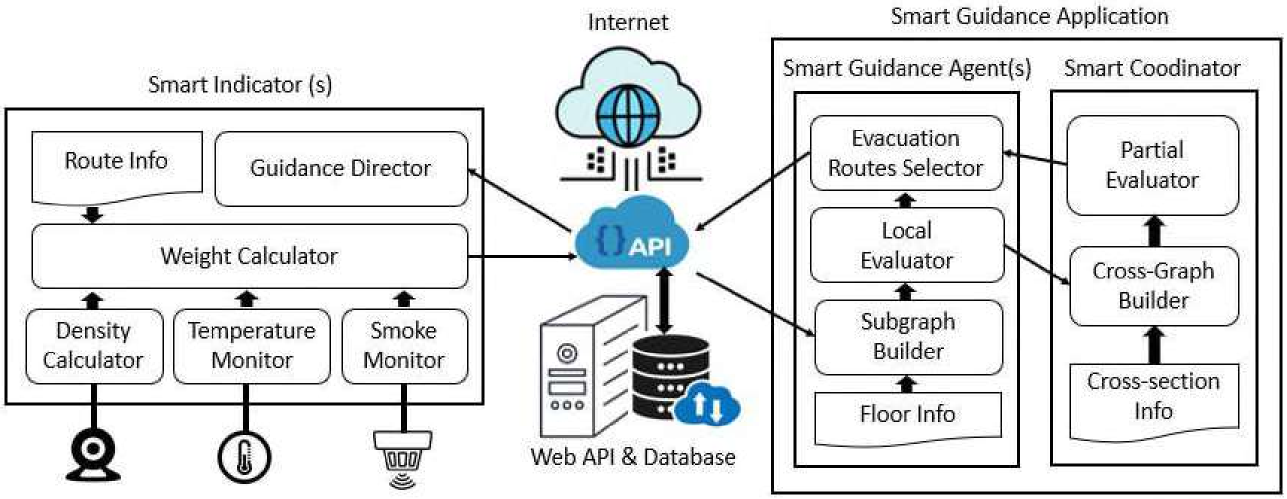

Figure 1: System Architecture

$$\mathbf{W_{(i,j)}} = \frac{\mathbf{L_{(i,j)}}}{\mathbf{T_{(i,j)}} \times (\mathbf{C_{(i,j)}} - \mathbf{D_{(i,j)}} + \mathbf{1})} \qquad \mathbf{(1)}$$

where $T_{(i,j)}$ is the hazard intensity road segment (i,j) ; $L_{(i,j)}$ is the physical length of the road segment (i,j), $D_{(i,j)}$ is the people density on the road segment (i,j), and $C_{(i,j)}$ is the capacity of the road segment. Here, $T_{(i,j)}$ can be obtained by using our early work, a sensor-based approach [12].

We use the temperature and smoke information gathering by *Temperature Monitor* and *Smoke Monitor* module. The people density on the road segment can be calculated by implementing a convolutional neuron network based image processing [13]. The people density calculating is performed by the *Density Caculator* module. For the capacity of the road segment, we can obtain it by estimating the maximum number of people who can move on the given segment at the same time based on the physical length and width of that segment information stored in the *Floor Info* file. Supposing that a person needs at least 1.0 square meters (m^2) of space for moving. Thus, a given road segment with 10m of length and 2m of width has a capacity of 20 (people). After calculating the weight of the road segment, smart indicators update the weight to the database by calling a REST API. This information is used by SGAs to build a weighted graph for finding efficient evacuation routes.

Another important role of a smart indicator is to show the direction provided by a Smart Guidance Agent. To do this, the *Guidance Director* module frequently checks the sign sent by the SGA from the Web API every a period of time (e.g., 5 seconds in our case).

Smart Guidance Agents. A smart guidance agent is designed to control the direction of smart indicators in a specific region (e.g., a floor). To do this, the SGA first builds a weighted subgraph relying on the structure of a network of smart indicators in the given floor using the information stored in the *Floor Info* file and the weights of road segments sent by smart indicators under its management. Then, it runs a shortest path algorithm (e.g., Dijkstra algorithm) to find efficient routes from every smart indicator to stairs of the floor its management. This step is performed by the *Local Evaluator* module. The *Local Evaluator* also calculates and sends the weights among stairs to the *Smart Coordinator* module for building weighted cross-graphs. Finally, the SGA use the *Evacuation Routes Selector* module to choose the best evacuation route for its smart indicators by combining the weight from smart indicators to stair-nodes in the floor (i.e. smart indicators located at stairs) and the weight among stair-nodes of several floors sent by *Partial Evaluator* module.

Smart Coordinator. A smart coordinator has two main functions: (1) building weighted cross-graphs based on the structure of network stair-nodes among several floors (e.g., each of five floors) using the information stored in the *Cross-section Info* file and the weighted among stair-nodes sent by the *Local Evaluators;* (2) finding the shortest path from a stair-node to other stair-nodes in the lower floors for every weighted cross-graph. This information is used by SGA to select efficient routes for smart indicators.

3.2 Design of Smart Indicators

To show the possibility of actualization of our smart indicators based emergency evacuation system, in this section, we present a design of a smart indicator by using IoT devices. Figure 2 shows our design model of a smart indicator. In our design, we use a Raspberry PI as a smartboard to implement the modules of a smart indicator, which is presented in Section 3.1. A digital camera is connected to the smart board via a USB port. We implement a Python program using a pre-train CNN model in [13] for counting people in the images captured by the camera. We use a DHT11 temperature sensor and a MQ-135 gas sensor to measure the

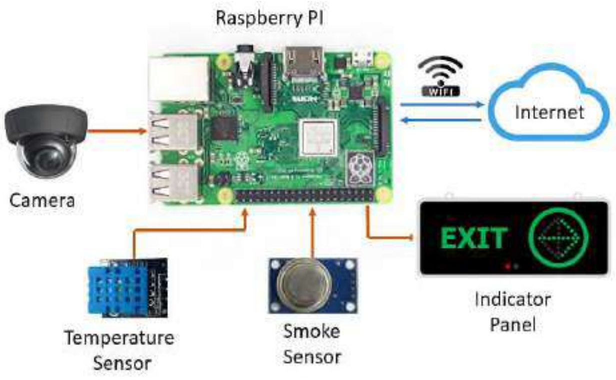

Figure 2: Hardware Components of a Smart Indicator

temperature and CO2 concentration in the air around the road segment where the smart indicator is located, respectively. To show different directions (e.g., four directions in our case), namely *east*, *west*, *north*, *south* correspond to the sign sent by SGA, we use 37 super bright LEDs, whose status (on/off) are automatically controlled by a Python program running on the smartboard. We also implement a program to send the calculated weight of the road segment and receive the sign via REST APIs every 5 seconds during a fire event happen. Note that, the smartboard can connect to the Internet via Wifi or Ethernet connection. Furthermore, we attach two LEDs (Red and Green) to the smart board to represent the status of disaster events. In which, the Red LED will be ON status during the disaster event, otherwise, it will be OFF and the Green LED will be ON.

3.3 Implementation of Partial Evacuation Routing Algorithm

In this section, we present a strategy using LCDT-based evacuation routing approach without global view information (i.e. using the hazard intensity and the crowd congestion in the whole building) but with partial view information (i.e. using the hazard intensity and the crowd congestion information in a group of sections in the building). We have a key observation is that the status of hazard intensity and crowd congestion of floor ***A*** might not affect finding the evacuation routes in a floor ***B*** far from ***A*** (e.g., the floor 10th is far from the floor 2nd). Meanwhile, several adjacent floors are affected by each other. Therefore, we improve the LCDT-based evacuation routing approach by using the partial view information instead of the global view information. The main idea of the LCDT-based partial evacuation routing algorithm is summarized in four main steps as the following:

Step 1: Choose a parameter K ($2 \leq K \leq N$) which indicates how many sections/floors (i.e. a large floor can be split into several sections) can be used as a group of sections/floors for partial evaluation in the Smart Coordinator, in which N is the number of sections/floors.

Step 2: Build weighted cross-graphs for each group of sections/floors. The number of weighted cross-graphs, C_g, can be calculated by using a simple equation below:

$$|C_g| = \begin{cases} N/K + 1 & \text{if } N\%K \geq K/2 \\ N/K & \text{otherwise} \end{cases} \quad (2)$$

Thus, each cross-graph is constructed using K adjacent sections/floors except the last cross-graph has $K \pm N\%K$ sections/floors.

Step 3: For each weighted cross-graph, we run a shortest path algorithm to find effective routes among stair-nodes. It is performed by the *Partial Evaluator* module in the *Smart Coordinator.*

Step 4: Select effective routes from every smart indicator in each section/floor to the stair-node or exit-node on the lowest floor of each section. This step is performed by the *Evacuation Routes Selectors* module in SGAs.

3.4 Implementation of Communication Module

We implement a Web API application to provide functions to support updating the weight of road segments calculated by smart indicators as well as sending the guidance directions from SGAs to the smart indicators. The Web API application is published to a host on the Internet so that both smart indicators and SGAs can access through the HTTP protocol. The information is packed in JSON format before sending the request to the Web server.

4 EVALUATION AND RESULTS

We implement a simulator as a *Smart Guidance Application* to show how our proposed system work. In this application, we implement our LCDT-based evacuation routing algorithm using partial view information, namely LCDT&PV, and compare with baseline approaches including LCDT-based evacuation routing algorithm using global view information, namely LCDT&GV [5]; Shortest-Path-Length based approach using the nearest stairs to pass through exits, namely Nearest Stairs; Length-Capacity-based approach which mentioned in [2], namely LC; and Length-Trustiness-based approach [1], namely LT. We also implemented and deployed a smart indicator that interacts with the simulator.

We simulate a synthetic building with ten-story (ten floors), each has 4 stair gates and 187 smart indicators, and the first floor has 3 exit gates as shown in Figure 3. The area of each floor is approximately 8,300 square meters. We simulate a fire event that affected a stair on the first floor as well as 2 exit gates and their close regions. We assign randomly trustiness values in the range [0.2,0.8] for road segments affected by the fire event. We also simulate the fire spreading by randomly updating trusted values and expanding the affected regions. We generate 2000 objects simulating people in the building. The location of people is generated randomly nearby locations of smart indicators. We


Emergency Evacuation System for Smart Building

Simulation Time: 1200 Update Interval: 10

Running Time: 63 Total Civilians: 2000 Evacuated: 325 Remain: 1679

Figure 3: An Emergency Evacuation Simulator

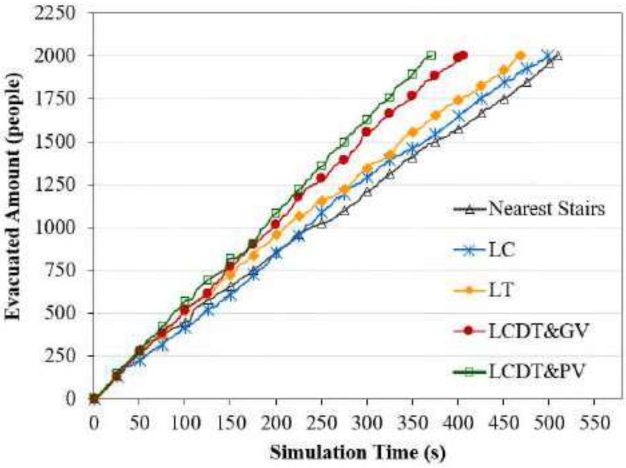

Figure 4: Comparison of the effectiveness between LCDT&PV and other baseline approaches

choose the interval for updating the status of every smart indicator equals 10 seconds. The moving step of people is updated every 200 milliseconds.

Figure 4 illustrated the comparison of our LCDT&PV approach and other baseline approaches. We found that our LCDT&PV approach is more effective than others in overall. Specifically, to guide all people pass through exit gates, the needed simulation times are 371s, 406s, 468s, 498s, and 509s corresponding to LCDT&PV, LCDT&GV, LT, LC, and Nearest Stairs, respectively. Thus, our LCDT&PV approach can save approximately 8% to 25% of the total evacuation time compared to other baseline approaches.

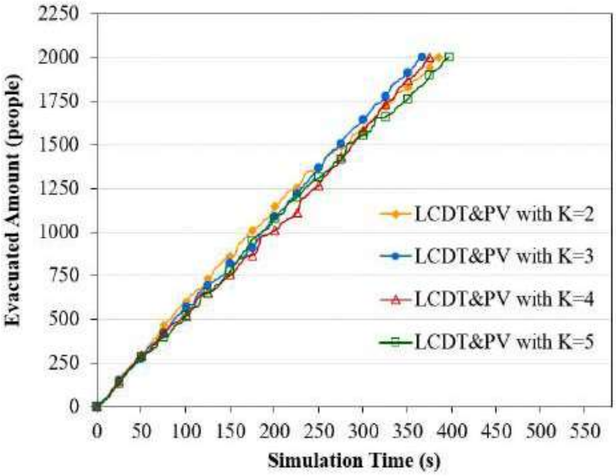

Figure 5: Comparison of the effectiveness with varied *K* in our LCDT&PV approach

We also run experiments with varied *K* using for partial evaluation in our approach. The experimental results are described in Figure 5. We can see that the effectiveness of our LCDT&PV

approach is slightly different between the values of K. In this case, K equals 3 is the best choice to run our LCDT&PV algorithm. In practice, identifying a suitable K for partial evaluation depends on several information such as the area of the buildings, the distribution of people in the building, and the impact of the fire event.

5 CONCLUSION

This paper presented a design and implementation of an emergency evacuation system that uses a dynamic evacuation routing approach and smart indicators as edge computing nodes, and then directs the evacuees to exit gates. The proposed system is designed as a distributed system with multiple layers of computing that provides an efficient routing approach using *partial view information* which represents the hazard intensity and the crowd congestion information of a group of sections/floors in the building. We designed smart indicators to capture people density using a pre-trained convolutional neural network model, track danger areas using temperature and smoke sensors, and show directions. The gathered information from smart indicators is provided to the smart guidance agents via a Web API for finding effective routes. We implemented a simulator to compare our evacuation routing approach with baseline approaches. Experimental results showed that our approach reduces up to 25% of the total evacuation time compared with others. Moreover, through the results of initial smart indicator implementation, we showed the viability of our proposed system.

ACKNOWLEDGMENTS

This research is funded by Hung Yen University of Technology and Education under the grant number UTEHY.L.2020.07. This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program(IITP-2021-2016-0-00314) supervised by the IITP(Institute for Information & communications Technology Promotion). This research was supported by the Bio & Medical Technology Development Program of the National Research Foundation (NRF) & funded by the Korean government (MSIT) (NRF-2019M3E5D1A02067961)

REFERENCES

[1] Dimakis, Nikolaos, Avgoustinos Filippoupolitis, and Erol Gelenbe. “Distributed building evacuation simulator for smart emergency management.” The Computer Journal 53, no. 9 (2010): 1384-1400.

[2] Wong, Sai-Keung, Yu-Shuen Wang, Pao-Kun Tang, and Tsung-Yu Tsai. “Optimized evacuation route based on crowd simulation.” Computational Visual Media 3, no. 3 (2017): 243-261.

[3] Lujak, Marin, Holger Billhardt, Jürgen Dunkel, Alberto Fernández, Ramón Hermoso, and Sascha Ossowski. “A distributed architecture for real-time evacuation guidance in large smart buildings.” Computer Science and Information Systems 14, no. 1 (2017): 257-282.

[4] Cristóbal-Salas, Alfredo, Bardo Santiago-Vicente, Daniel Pérez-Castañeda, Carolina Solis-Maldonado, Raul Alejandro Luna-Sáncez, Andrei Tchernykh, and Sergio Nesmachnow. "A Customized Evacuation Route Planning Algorithm for Highly Risky Smart Cities." In 2019 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), pp. 1-5. IEEE, 2019.

[5] Van-Quyet Nguyen, Huu-Duy Nguyen, Quyet-Thang Huynh, Nalini Venkatasubramanian, and Kyungbaek Kim. "A scalable approach for dynamic evacuation routing in large smart buildings." In 2019 IEEE International Conference on Smart Computing (SMARTCOMP), pp. 292-300. IEEE, 2019.

[6] Dressler, Daniel, Martin Groß, Jan-Philipp Kappmeier, Timon Kelter, Joscha Kulbatzki, Daniel Plümpe, Gordon Schlechter, Melanie Schmidt, Martin Skutella, and Sylvie Temme. "On the use of network flow techniques for assigning evacuees to exits." Procedia Engineering 3 (2010): 205-215.

[7] Abdelghany, A., Abdelghany, K., Mahmassani, H. and Alhalabi, W., 2014. Modeling framework for optimal evacuation of large-scale crowded pedestrian facilities. European Journal of Operational Research, 237(3), pp.1105-1118.

[8] Zhang, Qian, Tao Chen, and Xian-zhi Lv. "New framework of intelligent evacuation system of buildings." Procedia engineering 71 (2014): 397-402.

[9] Liu, J. W. S., F. T. Lin, E. T. H. Chu, and J-L. Zhong. "Intelligent indoor emergency evacuation systems: Reference architecture and data requirements." In 2016 Future Technologies Conference (FTC), pp. 600-609. IEEE, 2016.

[10] Lee, John, Dan Koo, Dinaol Tadesse, Atharv Jain, Sushmitha Shettar, and Dongsoo Kim. "Emergency Evacuation Assistance." In 2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM), pp. 1-6. IEEE, 2020.

[11] Zualkernan, Imran A., Fadi A. Aloul, Vikram Sakkia, Hassan Al Noman, Salman Sowdagar, and Omar Al Hammadi. "An IoT-based Emergency Evacuation System." In 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), pp. 62-66. IEEE, 2019.

[12] Nguyen-Van, Quyet, and Kyungbaek Kim. "Study on location trustiness based on multimodal information." In Proceedings of the 4th International Conference on Smart Media and Applications, pp. 187-190. 2016.

[13] Zhang, Yingying, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. "Single-image crowd counting via multi-column convolutional neural network." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 589-597. 2016.