Person detection with deep learning and IoT for smart home security on Amazon Cloud

Nazir, Sajid; Poorun, Yovin; Kaleem, Mohammad

Published in:
Proceedings of the International Conference on Electrical, Computer, Communications and Mechatronics Engineering 2021 (ICECCME 2021)

Publication date:
2021

Document Version
Author accepted manuscript

Person Detection with Deep Learning and IoT for Smart Home Security on Amazon Cloud

Sajid Nazir
Dept of Computing
Glasgow Caledonian University
Glasgow, UK
sajid.nazir@gcu.ac.uk

Yovin Poorun
Dept of Computer Science
African Leadership College
Pamplemousses, Mauritius
ypoorun@alueducation.com

Mohammad Kaleem
Dept of Electrical and Computer Engineering
COMSATS University
Islamabad, Pakistan
mkaleem@comsats.edu.pk

***Abstract*— A smart home provides better living environment by allowing remote Internet access for controlling the home appliances and devices. Security of smart homes is an important application area commonly using Passive Infrared Sensors (PIRs), image capture and analysis but such solutions sometimes fail to detect an event. An unambiguous person detection is important for security applications so that no event is missed and also that there are no false alarms which result in waste of resources. Cloud platforms provide deep learning and IoT services which can be used to implement an automated and failsafe security application. In this paper, we demonstrate reliable person detection for indoor and outdoor scenarios by integrating an application running on an edge device with AWS cloud services. We provide results for identifying a person before authorizing entry, detecting any trespassing within the boundaries, and monitoring movements within the home.**

Keywords— embedded programming, remote monitoring, edge computing, motion detection, communications protocol, artificial intelligence, false positive

I. INTRODUCTION

A smart home has a large deployment of sensors to collect data and provide control of connected devices for improving health, conserving energy, and ensuring security of the occupants. The number and types of devices for smart home applications is increasing at a very fast rate and so are the possible applications. There was much hype created around the IoT applications with refrigerators automatically ordering groceries, and kettles being automatically switched on based on a calculated arrival time of the home owner. These esoteric applications although technically feasible, haven't achieved much traction and following yet.

The sensors generate an enormous amount of data that needs to be processed, analyzed and visualized. Cloud platforms provide an ideal set of services to meet IoT application needs. Cloud platforms also have a global presence whereby data and applications can be accessed at a low latency because of geographical closeness of cloud servers, both for the data producers and consumers. Cloud based platforms are preferred over local processing if the latency is acceptable [1]. However, with recent technological advances in communications, latency is no longer a concern even for time critical applications. The use of cloud platforms for processing big data and artificial intelligence (AI) workloads is increasing due to the virtually unlimited storage, compute, and processing capabilities available on cloud platforms. This makes it possible to upload data to the cloud platform for storage, analysis and generating notifications of any anomalous behaviour in near real time.

A comparison of cloud based IoT architectures, such as AWS IoT, Samsung SmartThings, and open source platform offerings such as FIWARE, OpenMTC, is provided in [1]. A cost and performance comparison of IoT architectures on AWS, Google Cloud Platform (GCP) and Microsoft Azure concluded that each platform provided integration of other services with IoT services [2].

The IoT applications for smart homes cover a range of applications. An implementation combining fan speed control, email alert, and garage door control using Raspberry Pi [3] over the home Wi-Fi network is described in [4]. A smart home security system to control the lighting, generate fire warning, and monitor gas leakage utilising the AWS IoT services is described in [5].

A review of smart home applications with IoT covering the challenges hindering IoT adoption and recommendations is provided in [6]. An overview of IoT for smart cities along with the challenges and risks is provided in [7]. The challenges of integrating IoT for smart home applications are described in [8]. The evolution of IoT technologies from a security perspective is described, highlighting the vulnerabilities and mitigation techniques [9]. The security of the devices used in the smart home must be ensured to protect them against malicious use of data or operation.

AWS cloud is the most mature cloud platform with the largest offering of cloud services. AWS provides many services that can be integrated with IoT to provide an end-to-end smart home security application [10].

The focus of this paper is to investigate the use of deep learning and IoT services on the AWS cloud platform to design an indoor and outdoor security system for smart home applications. The application captures an image using Raspberry Pi camera on being triggered by a motion detection event. The three scenarios being considered are (i) Person identification, for a person approaching a controlled entry point, (ii) person detection, for detecting intruders entering a restricted area in an outdoor environment, and (iii) intrusion detection, in an indoor environment based on comparing a motion detection event against the normal detection pattern. In all these cases, an alert

can be generated automatically to notify the homeowner, or security agencies.

Rest of the paper is organized as follows: Section II describes related work. Section III covers the related deep learning and IoT services on the AWS platform. The materials and methods are described in Section IV. The experiments and results are provided in Section V. Finally, the discussion and conclusion are provided in Section VI and VII.

II. Related Work

A home security system utilising machine learning and IoT is described in [11]. The designed device can be installed at the house entrance and is coupled with motion and distance sensors, and a camera [11]. The images taken by the camera were compared against a database of images using a pre-trained model [11].

Architecture of a door sensor that can inform the user of an 'open' event is described in [12]. The architecture utilised a magnetic reed switch, Arduino and Raspberry Pi for implementation, and transmission of the event to a mobile application [12]. A home security system using PIR sensor, IR sensor, piezoelectric sensor, and sound sensor is described in [13] that identified an intrusion and sent a notification to the user.

Message Queuing Telemetry Transport (MQTT) is the protocol commonly used for publishing messages from IoT devices [14], [15]. An image notification system for smart home using MQTT protocol is described in [14]. A room temperature control and fire alarm system utilising MQTT on AWS platform integrated a fire sensor, fire alarm, and sprinkler actuator on Wi-Fi [15].

An MQTT based object detection and home appliance control integrated AWS cloud, and GSM modem for application control in smart cities applications [16]. The model used deep neural networks for recognition and classification under different environmental conditions [16]. A smart home system using Raspberry Pi for remote monitoring and surveillance is described in [17]. The system utilised AWS services such as Simple Email Service (SES) and Simple Notification Service (SNS) to notify the home user in order to manage energy consumption [17]. A cloud-based system for fall detection and activity monitoring using AWS cloud is described in [10]. The system used multiple inputs and devices for fall detection with data transferred to the AWS cloud. The data was accessible to the caregivers and was also used for automated notifications in case of fall detection [10]. A home security system for detecting an intruder using IoT and AWS Rekognition services used face recognition and comparison algorithms on AWS cloud [18]. The images of authorised persons were pre-stored in AWS storage and used for face comparisons as a person approached the controlled entry point generating notification to home owner [18].

In comparison to above, our work on person identification is different as we use AWS Rekognition service for a known face search against images stored in server-side container termed as *collection*. The results are provided for person identification using a publicly available image dataset and invoking AWS Rekognition service from an edge device. We also provide results for person detection in outdoor environment with challenging low light conditions from a public dataset and indoors using a PIR sensor sending motion detection events to AWS IoT core. We demonstrate that consistent, reliable and reproducible results are possible using AWS services. Thus, the chances of any false positives, where it is wrongly determined that a person is in an image, are reduced significantly. This is critical for security applications as otherwise unnecessary actions may be initiated causing loss of time, effort and resources.

Fig. 1. AWS Deep Learning Framework.

Our focus is to show that sophisticated security applications can be designed by using the fully managed AWS deep learning and IoT services.

III. AWS Deep Learning Framework and I0T Architecture

A. Deep Learning Framework

The AWS machine learning stack (Fig. 1) provides a layered architecture of machine learning services at different levels of abstraction. The services in the top layer require just an Application Programming Interface (API) call and abstract away all the model's complexities. Rekognition is one of the AI services and provides facilities for object detection, image labelling and face comparison etc.

Face recognition and object detection can also be achieved through the platforms and frameworks in the two lower layers in Fig. 1. However, these layers require more implementation efforts and provide greater control over the models.

AWS Rekognition service also provides a facility to store images of faces in a container known as a *collection*. The face image features stored in the *collection* can then be used to lookup or match a face against another image or video.

A particular use case could be to determine if a person should be allowed entry to a house. The facial images of all the house members or allowed visitors are pre-stored in a *collection* stored on AWS. As any person approaches the main entrance to a house, an image is captured based on motion detection. The image can then be uploaded to AWS cloud for a match against the faces in the *collection*. The result could be a match or no match with the images in *collection*, determining the entry to authorised only in case of a match.

B. IoT Architecture

The IoT architecture on AWS is shown in Fig. 2. The Message Broker is an MQTT broker and is an IoT endpoint for IoT devices to connect. The Device Shadow service is used to

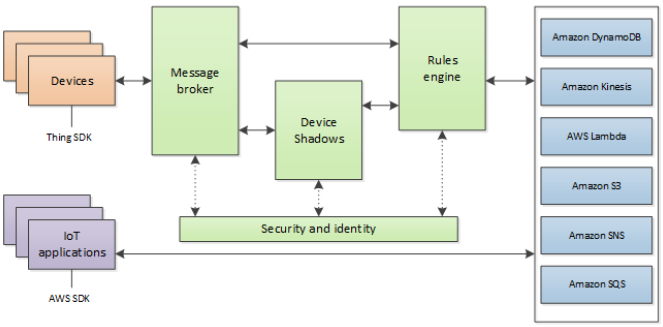


Fig. 2. AWS IoT Architecture.

maintain the current state of a device (thing). The Rules Engine connects the Message broker with the various services available on the AWS platform and rules provide for message processing and integration with other AWS services.

The IoT services on AWS platform use MQTT, HTTPS and Web Sockets as the communications protocols. MQTT however is a much lighter and favoured protocol compared to the other options [14].

Any IoT devices need to be configured and registered in order to connect with AWS IoT services for a secure data exchange and further processing of data.

IV. Materials and Methods

A. Image Datasets

We used different datasets according to their suitability for the various scenarios being considered.

1) Person Identification Scenario: The 'Celebrity Together' dataset [19] has multiple celebrities per image and comprises of 194k images with 546k faces in total containing 2622 labelled celebrities. The dataset was created using Google Image search and was later verified by human annotation [19].

2) Object Detection and Labelling Scenario: The dataset [20] contains a set of thermal images and videos for autonomous car research. The dataset has thermal images of people, vehicles, bicycles and dogs under both day and night lighting conditions. The dataset can help to develop machine learning algorithms to detect objects by combining optical, Light Detection and Ranging (LIDAR), radar and thermal sensor images.

The other thermal infrared dataset contains 4381 thermal images of 324x256 resolution captured with a Tau 320 thermal infrared camera [21]. The images of both 8 and 16 bit depth are available but we only use 8 bit images as 16 bit images appeared very dark.

We selected outdoor and low light condition images from the two datasets [20], [21] for object labelling and to determine if an object and specifically a person is in the image.

B. Sensor Nodes

We experimented with three configurations of sensor nodes built using Raspberry Pi and equipped with cameras and Passive Infrared (PIR) sensors to cater for the three application scenarios as described below.

Raspberry Pi [3] was used with Raspberry Pi camera which was connected using the Camera Serial Interface (CSI). The camera was configured to take a picture on motion detection event generated by a Passive Infrared (PIR) sensor.

PIR sensor detects motion in its field of view by detecting the movement of infrared radiation which is emitted by all bodies. However, in outdoor environments, a gust of air, car headlights etc. can trigger these, generating a false positive [22]. Thus, these sensors cannot be used as a reliable trigger to detect a person. We therefore propose to use a PIR sensor only for indoor scenarios where it is highly unlikely that it would be triggered erroneously. We also use the PIR motion trigger to capture an image which is then analyzed using a deep learning model to determine an intrusion and also to identify the intruder or the objects in the scene.

A PIR sensor keeps generating triggers if the motion event continues in its field of view. Thus, for any motion detection application, the PIR triggers can be ignored for a defined time window after the first trigger. We used a time window of 30 seconds to avoid repeated triggers.

The Raspberry Pi requires a connection to AWS Rekognition service for the person identification and detection scenario. We used Raspberry Pi 4 and Raspberry Pi zero as the sensor nodes. Connecting a Raspberry Pi to AWS platform requires downloading the security certificates onto the device and installation of AWS Python Software Development Kit (SDK) Boto3 [23], [24]. The motion detection required configuration of the Raspberry Pi to communicate with AWS IoT core. Therefore, additionally AWS IoT Device SDK [25] is also required for communication with the AWS IoT core.

C. Application Usage Scenarios

1) Person Identification: This requires an AWS Rekognition *collection* to be created which contains all the face images against which a face in the new captured image can be matched.

The captured image was uploaded to the AWS Rekognition service which can optionally be supplemented with a threshold for the face match. To demonstrate the functionality, we selected two images with multiple faces from the dataset [19] to create a *collection*. The corresponding test images were used to check for a face match through the Rekognition service.

2) Object Detection and Labelling: This scenario requires determining an intruder's presence in an image captured in an outdoor environment. We are specifically interested in object detection under low light and challenging conditions. The outdoor images may be captured using optical, thermal or infrared camera based on the nature of the application and normally a PIR would trigger the capture.

The problem of using a PIR sensor only for this scenario is that it would get triggered due to presence of animals etc. and many images will be false negatives or 'empty'. Instead we are interested in unambiguously determining a person in the scene.

Therefore, the captured image was analysed using the label detection feature of the AWS Rekognition service.

3) Indoor Person Detection: This scenario requires detecting the presence of a person in an indoor environment with a sensor node equipped with a PIR sensor only.

Although we only consider movement detection with PIR but this can also be supplemented with other sensors such as temperature sensor, DS1820B or magnetic switch to determine the opening of doors or windows. PIR sensor is very basic in nature and is privacy preserving but the analysis on the timing of captured motion pattern can reveal a wealth of information.

This scenario illustrates an end-to-end communication from the point of motion detection to the AWS cloud platform to notifying the registered user through email or Simple Notification Service (SNS).

D. Data Visualisation

For indoor motion detection, we transmit the sensor values to the AWS IoT services for data visualization. The IoT data that is reaching the IoT platform can be analyzed and certain actions can be defined and taken in case a threshold is reached.

More elaborate visualization dashboards can be created using AWS Elasticsearch service with Kibana.

V. Experiments and Results

This Section describes the results for the three scenarios that were considered to detect a person's presence to identify cases of trespassing or intrusion in the designated area.

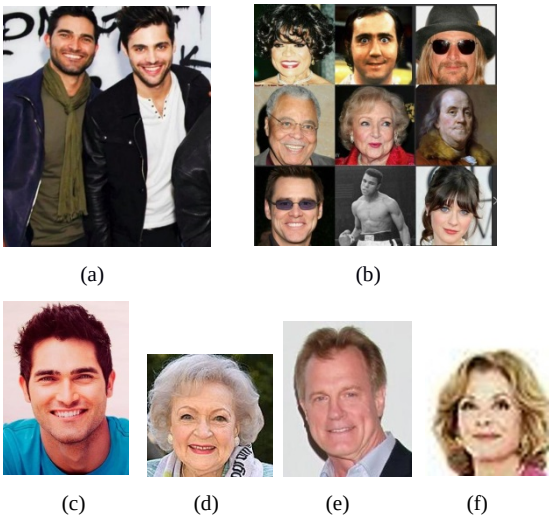
Although the Raspberry Pi sensor nodes were setup to communicate with AWS platform on detecting motion and subsequent image capture, we show the results for images from the publicly available datasets for enabling comparisons and supporting reproducible results.

A. Person Identification

The images in Fig. 3(a) and (b) were used to create the *collection* [26] for AWS Rekognition. After creating the *collection*, the images containing the faces to be added to the *collection* have to be indexed. We used Fig. 3(a) and 3(b) to index the 11 faces in the two images shown. The images for indexing and later matching need to be in the specified Simple Storage Service (S3) bucket.

The threshold for face match was set as 90 and a maximum of three images were to be matched. The 35 face images from the dataset of the person in Fig. 3(c) successfully matched with a similarity range of 99.11% to 100% with the same person in Fig. 3(a). Similarly, the 35 face images from the dataset of the person in Fig. 3(d) successfully matched with the face image of that person in Fig. 3(b). The similarity level ranged from 99.99% to 100%. For both sets of the images to be matched only one and the correct image got a match.

The images 3(e) and 3(f) and four other images (not shown) did not get matched with any of the faces in the collection. The fact that each match also provides a confidence level further helps in making an informed decision regarding allowing or disallowing a person.



(a) (b) (c) (d) (e) (f)

Fig. 3. The two images (from the Celebrity dataset [19]) 3(a) and 3(b) are used to create an image *collection*, against which subsequent faces in captured images are matched. Image (c) and (d) are two samples from a set of 35images each that should match the corresponding indexed faces of these persons in the *collection*. Image (e) and (f) together with four other images (not shown) are of different persons and these faces didn't find a match.

B. Object Detection and Labelling

The images captured from the outdoor security camera can be uploaded to the AWS Rekognition service to find the labels corresponding to each of the objects in the scene. For a security application under consideration the primary interest is to determine if the image contains a person. This is easily accomplished as the response to AWS Rekognition service contains the label corresponding to each object with the confidence level, and from within the returned labels, the term 'person' can be searched.

The results of uploading an image to the AWS Rekognition service and the returned image with a bounding box and the returned labels are provided next.

The image in Fig. 4(a) was uploaded and the returned image with the bounding box and the labels is shown in Fig. 4(b). The corresponding labels were: Nature, Outdoors, Weather, Snow, Home Decor, Building, Pedestrian, Person (Confidence 68.23), Human (Confidence 72.57), Road, Urban, Storm, Winter, Tarmac, Asphalt, City, Town, and Ice.

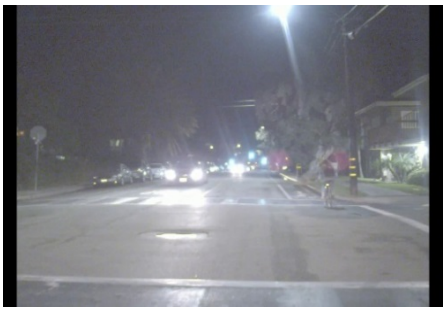
The image in Fig. 5(a) was uploaded and the image with the bounding boxes is shown in Fig. 5(b). The corresponding labels were: Road, Tarmac, Asphalt, Lighting, Nature, Cars (confidence 87.81), Transportation, Vehicle, Automobile, Pedestrian (84.48), Person (84.48), Human, Freeway, Intersection, Light, Lamp Post, Outdoors, and Highway.

The bounded boxes and the labels show a good delineation of a person and objects in the images. The results are similar to or better than a human observer. Obviously, the trained deep learning model looks for the image features for a match. Only a portion of the bicycle rider in the image has been labelled as a person, but even a human observer would have more chances of going wrong in such a challenging scene.

(a)

(b)

Fig. 4. Image (a) [21] is shown with a bounding box in (b) drawn using AWS Rekognition service.



(a)

(b)

Fig. 5. Image (a) [20] is shown with bounding boxes in (b) drawn using AWS Rekognition service.

The alert for a positive detection could be generated immediately at the AWS cloud, or it could also be sent out from the Raspberry Pi.

C. *Indoor Person Detection*

This scenario caters for the cases where a simple PIR sensor is deployed inside the smart home to cover either the areas of probable intrusion or where the movement detection is used as a method to infer the activities of the resident persons. After the first activation, we ignore any PIR motion trigger within a 30 seconds window to avoid sending continuous motion detection events.

The PIR motion trigger activations were sent to the AWS platform on occurrence. We used the value '1' in the message to signify motion detection.

The motion detection events as seen on the AWS IoT service for monitoring over a day are shown in Fig. 6. The timings and the motion activations show the trigger pattern and also the periods of inactivity. This can be used to flag a motion activity at an unauthorized time or place as an anomaly. Fig. 6 also shows the percentage distribution for different metrics.

VI. Discussion

Any monitoring system that is not capable of generating an automatic and near real time notification on detecting anomalous behavior is just good for recording historical incidents. Timely interventions can be enabled through a cloud based intelligent monitoring system based on IoT and artificial intelligence [27].

The simple scalar data from a PIR sensor can be analyzed to extract a wealth of information. For example, the time of activation can indicate the presence of an intruder. Similarly, the activation pattern can help to infer the activity or behavior of the person etc.

The cost of high-resolution thermal cameras is currently significantly high but these have become affordable in low resolutions. A thermal camera [28] of 24x32 pixel resolution costs around £ 50.0 and can be integrated in embedded applications. The application of deep learning techniques on thermal images is also an interesting research area with applications in many domains.

For developing an AI application, it is preferable to integrate and use the solutions that already exist. This helps to quickly develop and launch an application and to try out new ideas for further development. The built-in AWS Rekognition service can provide better results quickly as compared to an in-house developed solution. An autonomous deep learning intrusion detection system is much preferable over remote monitoring by a human to avoid errors in object identification due to fatigue, boredom and lack of interest. For critical security applications, it is still recommended to have a human-in-the-loop.

The data being sent to the AWS IoT service can also be used to create sophisticated dashboards using AWS Elasticsearch service and the open source plug-in, Kibana.

VII. Conclusion

The security of smart homes and other critical infrastructures is important and can be ensured by utilizing the latest technologies. This paper describes the use of deep learning and Internet of Things services on Amazon Web Services (AWS) cloud for the design and implementation of a smart home security application. We demonstrate that an unambiguous person detection and identification is possible

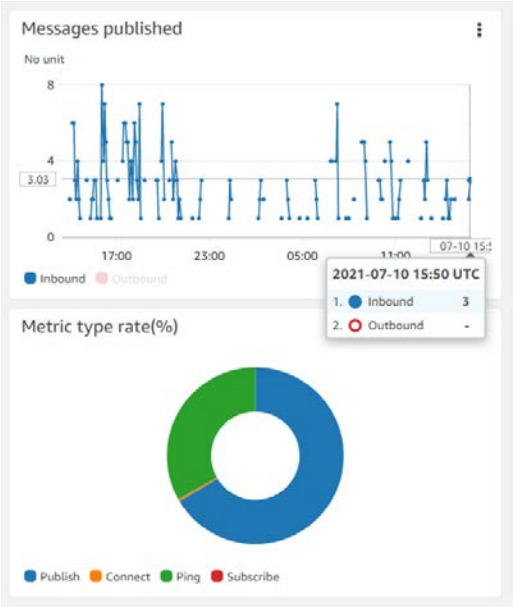


Fig. 6. Motion detection with IoT framework.

through integration of cloud services with applications running on an edge device, such as, Raspberry Pi. We consider three scenarios of person detection, that is, for authorizing entry based on person identification, to identify a trespasser in outdoor low-light images, and indoor person presence detection with Passive Infrared Sensor (PIR).

The proposed technology driven person detection for secure access and presence monitoring has a lot of potential applications for homes, offices, and healthcare infrastructures.

In our future work, we will investigate use of more types and number of sensors for security applications, and use of a unified dashboard using AWS Elasticsearch.

REFERENCES

[1] J. Guth *et al.*, "A detailed analysis of IoT platform architectures: concepts, similarities, and differences," B. Di Martino et al. (eds.), Internet of Everything, Internet of Things (Technology, Communications and Computing). Springer, Singapore. https://doi.org/10.1007/978-981-10-5861-5_4

[2] P. Pierleoni, R. Concetti, A. Belli and L. Palma, "Amazon, Google and Microsoft Solutions for IoT: Architectures and a Performance Comparison," in *IEEE Access*, vol. 8, pp. 5455-5470, 2020, doi: 10.1109/ACCESS.2019.2961511.

[3] Raspberry Pi. [Online]. Available: https://www.raspberrypi.org/help/what- is-a-raspberry-pi.

[4] M. Assim and A. Al-Omary, "Design and Implementation of Smart Home using WSN and IoT Technologies," *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, 2020, pp. 1-6, doi: 10.1109/3ICT51146.2020.9311966.

[5] A. Wang, Z. Yuan and B. He, "Design and Realization of Smart Home Security System Based on AWS," *2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, 2020, pp. 291-295, doi: 10.1109/ISPDS51347.2020.00067.

[6] M. Alaa, A.A. Zaidan, B.B. Zaidan, M. Talal, M.L.M. Kiah, "A review of smart home applications based on Internet of Things," *Journal of Network and Computer Applications*, vol. 97, 2017, pp. 48–65.

[7] B. Hammi, R. Khatoun, S. Zeadally , A. Fayad, L. Khoukhi, "IoT technologies for smart cities," *IET Netw.*, vol. 7, no. 1, pp. 1-13, 2018.

[8] L. Pan, and C. Lu, , "Challenges in integrating IoT in smart home," Report Number: 2019, *All Computer Science and Engineering Research*. https://openscholarship.wustl.edu/cse_research/1170

[9] R. Ande, B. Adebisi, M. Hammoudeh, J. Saleem, "Internet of Things: evolution and technologies from a security perspective," *Sustainable Cities and Society*, 54, 2020, 101728.

[10] Q. T. Huynh, U.D. Nguyen, and B. Q. Tran, "A Cloud-Based System for In-Home Fall Detection and Activity Assessment." In *International Conference on the Development of Biomedical Engineering in Vietnam*, pp. 103-108. Springer, Singapore, 2018.

[11] A. Chavan, S. Ambilpure, U. Chhapra, and V.Gawde, "Autonomous home-security system using Internet of Things and machine learning," In *International Conference on Innovative Data Communication Technologies and Application*, pp. 498-504. Springer, Cham, 2019.

[12] C. Davidson, T. Rezwana, and M. A. Hoque, "Smart home security application enabled by IoT," A. S. Pathan et al. (Eds.): *Smart Grid and Internet of Things*, *SGIoT*, LNICST vol. 256, pp. 46–56, 2019. https://doi.org/10.1007/978-3-030-05928-6_5

[13] P. A. Teja, A. A. F. Joe and V. Kalist, "Home Security System using Raspberry PI with IOT," *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2021, pp. 450-453, doi: 10.1109/ICACITE51222.2021.9404551.

[14] S. Nazir and M. Kaleem, "Reliable Image Notifications for Smart Home Security with MQTT," *2019 International Conference on Information Science and Communication Technology (ICISCT)*, 2019, pp. 1-5, doi: 10.1109/CISCT.2019.8777403.

[15] D. Kang *et al.*, "Room temperature control and fire alarm/suppression IoT service using MQTT on AWS," *2017 International Conference on Platform Technology and Service (PlatCon)*, Busan, Korea (South), 2017, pp. 1-5, doi: 10.1109/PlatCon.2017.7883724.

[16] S. Khan, S. Nazir and H. U. Khan, "Smart object detection and home appliances control system in smart cities," *Computers, Materials & Continua*, 67, pp. 895-915, 2021. DOI:10.32604/cmc.2021.013878

[17] S. Kayastha and P. Upadhyaya, "Design and Implementation of a Cost-Efficient Smart Home System with Raspberry Pi and Cloud Services," *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, 2019, pp. 1-7, doi: 10.1109/AITB48515.2019.8947439.

[18] M. Mehra, V. Sahai, P. Chowdhury and E. Dsouza, "Home Security System using IOT and AWS Cloud Services," *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, 2019, pp. 1-6, doi: 10.1109/ICAC347590.2019.9089839.

[19] Y. Zhong, R. Arandjelovic, and A. Zisserman, "Compact deep aggregation for set retrieval," In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[20] FLIR starter thermal dataset for autonomous vehicle testing. [Online]. Available: https://www.flir.com/news-center/camera-cores--components/flir-open-source-starter-thermal-dataset-for-autonomous-vehicle-testing/

[21] Thermal infrared dataset. [Online]. Available: https://projects.asl.ethz.ch/datasets/doku.php?id=ir:iricra2014

[22] S. Nazir, G. Fairhurst, and F. Verdicchio, "WiSE–a satellite-based system for remote monitoring," *International Journal of Satellite Communications and Networking*, 2017, 35(3), 201-214.

[23] AWS Python SDK Boto3. [Online]. Available: https://docs.aws.amazon.com/pythonsdk/

[24] Connect a Raspberry Pi or another device. [Online]. Available: https://docs.aws.amazon.com/iot/latest/developerguide/connecting-to-existing-device.html

[25] AWS IoT Device SDK for Python. [Online]. Available: https://github.com/aws/aws-iot-device-sdk-python/

[26] AWS Rekognition developer guide. [Online]. Available: https://docs.aws.amazon.com/rekognition/latest/dg/create-collection-procedure.html

[27] AWS IoT core pricing. [Online]. Available: https://aws.amazon.com/iot-core/pricing/

[28] Raspberry Pi thermal camera. [Online]. Available: https://www.raspberrypi.org/blog/raspberry-pi-thermal-camera/