Transactions on Engineering and Computing Sciences - Vol. 11, No. 4
Publication Date: August 25, 2023
DOI:10.14738/tecs.114.15155.
Sargsyan, S., Hovakimyan, A., & Antonyan, S. (2023). Creating a Security Assurance System Using IoT and ML Technologies. Transactions on Engineering and Computing Sciences, 11(4). 51-56.



Creating a Security Assurance System Using IoT and ML Technologies

Siranush Sargsyan
Yerevan State University, Yerevan, Armenia

Anna Hovakimyan
Yerevan State University, Yerevan, Armenia

Sveta Antonyan
Yerevan State University, Yerevan, Armenia

ABSTRACT

The work is dedicated to the development of security systems using the Internet of Things and Machine Learning technologies. The need for such systems use is associated with the increase in crimes, as well as with the increase in number of attacks on special areas. The authors of the article have created a security system that is located in a guarded area. The security system reacts in real-time and informs about the expected danger through a picture or video. The security device senses the movement of any object in the area through sensors, and the camera connected to the device captures the moving object. A Siamese neural network examines the image. If it is from the class of unwanted objects, the app sends it to the Telegram app of the guard of the monitored area. The authors used Raspberry Pi, various sensor components, and a Siamese neural network to create the security system. The system can be implemented in bank vaults, offices, and special-purpose buildings. We, being the professors of the university, attach more importance to the use of the created system on the premises of the educational institution.

Keywords: Internet of Things, Machine Learning, security system, Siamese Neural Network, classifier, Raspberry Pi.

INTRODUCTION

Nowadays, one of the primary problems in various fields of human activity is ensuring the safety of the person and the given area. In the world, including Armenia, the use of security systems become very necessary. People install personal security systems in their homes and at the entrances of residential buildings. The operation of service centers, shops, banks, and hotels is impossible to imagine without a security system. As a result of the expansion of the consumption of security equipment, there is a need to improve such systems that are already in operation, making them safer and more affordable.

The security system described in the article is intended to make the special purpose areas more secure and exclude any kind of illegal movement there.

To create the security system that uses the Internet of Things (IoT) technologies, the most suitable option from the types of single-board computers was selected, and the structure scheme of the device was modeled, taking into account the possibilities of its use in various fields.

We used a Siamese neural network to solve the face recognition problem using Machine Learning (ML) technologies. The security system software is written in the Python programming language. The created security system can be used in special-purpose areas of the educational institution, such as computer classrooms, laboratories, archives, etc.

STATEMENT OF THE PROBLEM

The purpose of the article is to describe the creation of a reliable and affordable security system using IoT and ML technologies.

IoT devices are designed to collect, analyze, process, and transmit data [1]. Thanks to this technology, it becomes possible to collect data in real-time, quickly respond to changing environments and make decisions without or with minimal human participation. First, devices collect data, for example, about the temperature in an apartment or a person's heart rate, then this data is sent to the cloud. The software handles them. If, for example, the temperature is too high or there is a burglar in the house, the system notifies the user about this or performs further actions itself, for example, calling the police. IoT systems operate in real-time and usually consist of a network of intelligent devices and a cloud infrastructure to which they are connected via WiFi, Bluetooth, or other forms of communication [1, 2, 3].

IoT devices collect data from various sources, supporting the learning process of an artificial intelligence (AI) system so that it knows how to properly automate the desired operations. Thanks to the IoT system, they can learn and make decisions about data management and analysis [9, 10].

Thus, IoT and ML technologies jointly create intelligent connected systems. IoT combined with the most dynamic AI technology can make the IoT system itself smarter and more easily mimic human activity [12, 13]. The use of ML algorithms with IoT in different applications depends on the field of application. These are often called smart IoT applications.

Analyzing data generated by smart IoT devices has applications in agriculture, healthcare, autonomous vehicles, wearable devices, etc.

In the presented work, a smart IoT device is created using the Siamese neural network (SNN) of ML [15].

HARDWARE AND SOFTWARE COMPONENTS OF A SECURITY SYSTEM

Raspberry Pi

A developed security system has hardware and software components. The hardware component consists of a Raspberry Pi single-board computer, which has USB, Ethernet, CSI Camera, Micro USB inputs, HDMI, Audio Jack outputs [4, 5].

USB ports enable the Raspberry PI to connect a keyboard, mouse, camera, display, and other equipment. Raspberry PI OS (Raspbian) operating system was used with Raspberry Pi. PIR motion sensors attached to the Raspberry Pi were used to detect objects with temperatures above zero, light radiating from them, and motion.

Two potentiometers are mounted on the PIR sensor board, one responsible for the sensitivity, and the other for the time of impulse delivery. This time is usually from two seconds to several minutes [6].

Raspberry Pi cameras work with visible light and infrared versions. After their physical connection, software settings were made.

Python RPI.GPIO module and the GPIO Zero library, available at www.raspberrypi.org, were used to manage the GPIO pins and hardware on the Raspberry Pi [7, 8].

A number of Python libraries were used to work with neural networks: TensorFlow, PyTorch, Numpy, Keras, Matplotlib, Pandas, etc. [11,12, 16,17].

Siamese Neural Network

Artificial neural networks are mathematical models built on the principle of biological neural networks [9-13].

Learning is the adaptation of a neural network to better solve a task by observing sample examples. In deep learning, neural networks are generally good at all tasks, but they need a large amount of data to work effectively [14]. For some tasks, including face recognition, we may not have much data. Therefore, SNN was used [15].

An SNN is a type of neural network, which in our case contains two identical subnetworks. They have the same weights and operate in parallel on two different input vectors to determine the similarity of the output vectors [15].

Typically, neural networks learn to predict a few predefined classes, which creates a problem when new classes need to be added or removed. To avoid the mentioned inconvenience, it is also appropriate to choose an SNN network, which enables the data to classify into new classes without additional training of the network.

An SNN network uses a similarity function and can distinguish whether two input images are similar. While training the model, the similarity function measures how different the feature vectors of two images are. With this architecture, the model can be trained without much data.

The feature vectors for similar image pairs for training the network must be close enough in distance, but the vectors for dissimilar pairs must not. By comparing the feature vectors of two images, the model learns what "similarity" is and how to recognize different images.

SECURITY ASSURANCE SYSTEM DESCRIPTION

The security system presented in the article makes up of various functional components.

1. The security system module detects motion, captures video, transmits to the real-time face clustering module, and sends the video to the user's Telegram if necessary.
2. The real-time face clustering module realizes image detection from video. It carries out the clustering process into classes.

Equipment Connections

To assemble a safety device, it was necessary to have some equipment. While working, several types of single-board computers were studied, such as Raspberry Pi, Arduino, Orange Pi, Rock Pi, etc.

After several studies and tests, it became clear that Raspberry Pi 4 B is the best solution to the problem. We used the PIR Motion sensor, RPi camera, Telegram app running gadget, LED, speaker, and connection wires. Note if connections are incorrect, the equipment may be out of order [5, 8].

Face Clustering System Description

Creating security device software had its peculiarities, as it was necessary to solve the problem of synchronizing individual devices, accurately designing an SNN network and real-time face recognition and classification from a picture or video, using the Telegram application, etc.

In face recognition, an SNN network detects the degree of similarity between two face images, determining whether they belong to the same class. An application performs real-time face recognition from video to segment frames from a continuous video stream and recognizes images from them [15].

We collected the dataset from pictures of people's faces guarding the area (we called them standard objects) and of other people's faces (non-standard objects).

Standard and non-standard objects have the structure $(x_i, y_i), i = 1,2, \dots, n$, where $x_i \in R^m$ is a feature, vector consisting of m values, $y_i \in \{1,2, \dots, C\}$ is the label of the vector x_i, C is the number of classes. In our case, the set of vectors is the set of face pictures belonging to the corresponding class, and $C = 2$.

We created database $S = \{(x_i, x_j, z_{ij}), i, j = 1,2,3, \dots, n\}$ of training and testing data. An element of the database consists of pairs of image vectors (x_i, x_j) and a binary marker z_{ij}. If both feature vectors x_i and x_j belong to the same class, then z_{ij} =0, otherwise $z_{ij} = 0$.

We divided the dataset S into two subsets, one consisting of 35 pairs of similar vectors (i.e., $z_{ij} = 0$) and the other of 40 pairs of different vectors (i.e., $z_{ij} = 1$). We used 80% of the set S for training of the network, and 20% for testing.

A pair of images, with their similarity label, is the training sample given to the input of the SNN network. During training, the network learns whether the features of the two pictures are similar.

The constructed SNN network consists of two subnets with input vectors x_i andx_j, respectively. Note the SNN implements the mappings $h_i = f(x_i)$ and $h_j = f(x_j)$. The outputs h_i and h_j of the subnets are given as input to another software module, which produces a final result by computing the Euclidean distance $d(h_i, h_j)$. Based on this distance, the degree of similarity between $x_i$and x_j is estimated. $d(h_i, h_j)$ is as small as possible for $y_i = y_j$, and as large as possible for $y_i \neq y_j$.

In the training phase of the SNN network, we used the Contrastive Loss function.

The Contrastive Loss function updates the weights so that the Euclidean distance between two vectors with similar features is the minimum according to the following formula

$$(1 - y)\frac{1}{2}(D_w)^2 + (y)\frac{1}{2}\max(0, m - D_w)^2 ,$$

where D_w is the Euclidean distance between the vectors, y is the similarity measure between the two vectors ($y = 1$, if the vectors are different, and $y = 0$, if they are similar). The loss function, when the vectors are different, is minimized due to the second term, where D_w must be maximized. That is, a greater distance between different vectors is encouraged. If the vectors are similar, i.e. $y = 0$, then the loss function must minimize D_w [15].

Thus, the SNN learns not on learning errors, or wrong predictions, but rather on the similarity feature of objects.

The complexity of the real-time face recognition and classification problem is that capturing and extracting a face in the resulting video recording adds noise to the training images. For this reason, a developed software module allows to make face capture by slowing down the video recording. Since the solution to the problem required a quick response, we created appropriate software regulation modules. To build the SNN network, Tensorflow, Keras, and more Python libraries were used [16, 17]. After training, we tested the network on a testing dataset. The SNN neural network in the security system presented in the article provides about 75% accuracy. The created security device is affordable and easy to use. The device is of small and compact size, thanks to which it becomes invisible to dangerous targets.

CONCLUSION

Using IoT and ML technologies, a security system has been created. The system reacts in real-time and informs about the expected danger if being in the monitored area. The system senses the movement of any object in the monitored area through sensors, and the camera connected to the device captures the moving object. The Siamese neural network examines the image. If it is from the class of unwanted objects, the developed application sends the examined image to the Telegram of the guard of the monitored area.

The need to use the security system is related to the daily increase in crimes, as well as to the numbers of attacks on special purpose areas. Their use is more demand in educational institutions, to check the entrance and exit of students. Raspberry Pi, various sensor components, Siamese neural network were used to create the security system. The smart IoT-device programming package is in the Python programming language.

References

[1] Radouan Ait Mouha, Internet of Things (IoT), Journal of Data Analysis and Information Processing, 2021

[2] Holton J., Fratangelo T. Raspberry pi architecture: Rochester Institute,2017

[3] "Microchip/SMSC LAN9514 data sheet;" Microchip. Retrieved 15 July 2014.

[4] CODESYS Control for Raspberry Pi SL http://www.raspberrypi.org/

[5] How HC-SR501 PIR Sensor Works & Interface It with Arduino https://lastminuteengineers.com/pir-sensor-arduino-tutorial/

[6] https://docs.python.org/2/library/time.html

[7] https://learn.sparkfun.com/tutorials/raspberry-gpio/python-rpigpio-api

[8] https://gpiozero.readthedocs.io/en/stable/recipes.html

[9] Kelleher, John D., Mac Namee, Brian; D'Arcy, Aoife “Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies”. Cambridge, MA (2020).

[10] Bernard Etienne,” Introduction to Mashine Learning”, 2021.

[11] "Eager execution | TensorFlow Core". TensorFlow. Retrieved November 4, 2021.

[12] "Pandas 1.5.3". 18 January 2023. Retrieved 22 January 2023.

[13] https://www.ibm.com/topics/unsupervised-learning

[14] https://www.mathworks.com/help/deeplearning/ug/reinforcement-learning-using-deep-neural-networks.html

[15] https://www.projectpro.io/article/siamese-neural-networks/718#mcetoc_1gke10cloo

[16] https://ai.meta.com/tools/pytorch/

[17] https://numpy.org/doc/stable/user/whatisnumpy.html