

The Devices of the Internet of Things Based on the Recognition of Handwriting Words with Mobile Assisted

<https://doi.org/10.3991/ijim.v14i04.12197>

Ghazi Shakah
Ajloun National University, Ajloun, Jordan
g.shakah@anu.edu.jo

Abstract—At the moment, all observed forms of communication are reduced either to a person-to-person scheme or person-to-device. But the Internet of Things (IoT) offers us a tremendous Internet future, in which will appear the communication type machine-machine (M2M). This makes it possible to integrate all communications into a common infrastructure, allowing not only to manage everything that is around us, but also providing information about the state of these things. The purpose of this paper is to create the client part of the client-server system for remote control of home appliances using cloud services through commands entered using handwritten words. For this we develop algorithms and methods for handwriting recognition using neural networks and implement a mobile application on the Android platform, which allows remote control of devices via cloud services based on commands entered using handwritten words. Anyway, this article will give a good understanding to other researchers who want to start their research on the IoT and will contribute to the effective accumulation of knowledge.

Keywords—Handwriting Recognition, Neural Networks, Internet of Things (IoT), Mobile

1 Introduction

At present, due to the increase in computer productivity, the field of application of technical vision systems, in particular, text recognition systems (TRS), which are relatively short sequences of characters from some a priori known alphabet, is significantly expanding. TRS are a special case of OCR-systems (Optical Character Recognition), represented by text recognition systems of different origin (printed, hand-printed, hand-written), but have a number of distinctive features.[1] [2] At the same time, the problem of the synthesis of TRS with given efficiency indicators remains unsolved. The effectiveness of TRS is usually characterized by a number of criteria. On the one hand, it is the quality of recognition, which includes probabilistic efficiency criteria, on the other hand, the time of the video image analysis working cycle, often limited by the requirement of real-time recognition. At present, the values of the performance criteria for most of the systems in operation are far from ideal, which is caused not only by the imperfection of the algorithms used. In practice, an

increase in the efficiency of the system is associated with the adjustment of the parameters of its algorithms, which makes it possible to adapt the system to the specified operating conditions (optical design, label type, lighting, etc.)[3] Usually TRS function as part of optoelectronic monitoring systems, which causes the following main differences from OCR text recognition systems: the low resolution of analyzed, complex background of text, presence of significant distortions introduced by the external environment, and high speed requirements. Another feature of TRS is that, as part of the optoelectronic control systems, TRS is used to identify objects with registration labels, such as labeled industrial products (steel slabs, pipes, etc.)[4]. The solution to the problem of text recognition is divided into a number of subtasks: separation of text from graphic information, removal of noise, selection of lines and individual words, recognition of letters and words. The most difficult is the task of recognizing handwritten words written together. There are a number of approaches to handwriting recognition, in particular, approaches based on the use of neural networks [5], on the use of hidden Markov models and structural features of letters and words [6].

2 Technologies Implementation System

2.1 Mobile platform analysis

For the development of a mobile application, the Android platform was chosen. Currently, Android is developing exponentially: every year the number of users of this operating system is constantly growing, thanks to the availability of mobile devices with this operating system. The Android operating system occupies 69.2% of the global mobile device market[7]. The Android platform is based on the Linux kernel and its own implementation of the Java virtual machine. Android allows you to create Java applications that control the device through the libraries developed by Google. Android Native Development Kit allows using libraries and application components written in C and other languages, which adds freedom to developers. Android freely provides a set of software for developing SDK software. It allows to quickly customize the application. The architecture of Android applications is based on the Model -View-View Model template (MVVM) template[8] [9]. This template consists of three components Model, View, View Model, the relationship between which is shown in Figure 1.

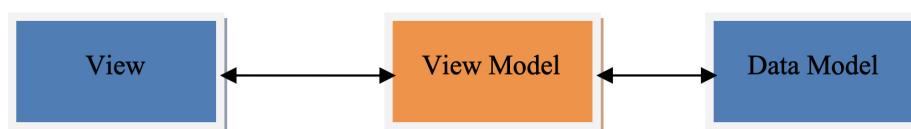


Fig. 1. MVVM template

2.2 Cloud service analysis

Cloud platforms internet of things (IoT) provide software that integrates everything in the system of the Internet of things. IoT can use the resources and functionality of cloud services. Such services can manage devices, collect and analyze data from devices, coordinate interaction with other services. Such companies as Microsoft, Amazon, Oracle develop their cloud platforms for this purpose. Consider the cloud service Microsoft Azure IoT Hub The interaction with the device is carried out using its device twin. Device twin is an abstraction for determining the state of a device. A device twin is a JSON – document that includes tags that are available only to devices, and properties used to synchronize settings between a device and its twin[10].In the Microsoft Azure IoT Hub service, there is no concept of an event as a notification mechanism between a device and a cloud service. But processing of messages at the level of applications deployed in the cloud service is available. Smart devices transmit data, called messages, through a device gateway. This data is used to update the shadow of the device. The device shadow is an object containing information about the device, which is updated when the state of the device changes. Figure 2. shows the mechanism for interacting with the shadow devices. The presence of this object eliminates the need to request information from the device to obtain its state and reduces the load on the network.

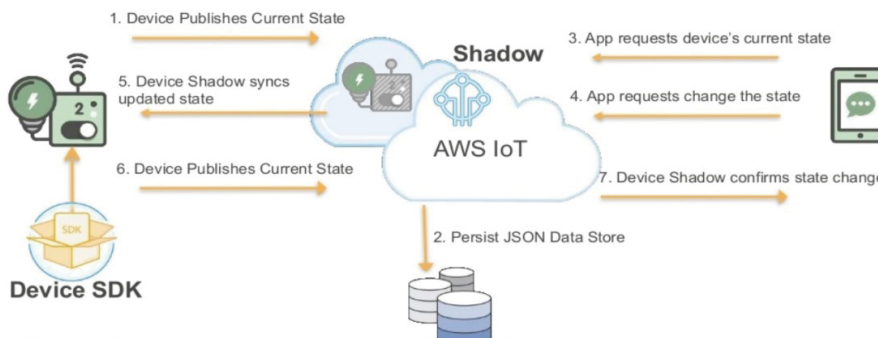


Fig. 2. The mechanism of interaction with the devices shadow

Let us take a closer look at the mechanism of device interaction using the MQTT protocol [11]. AWS IoT devices use the MQTT protocol to transfer messages between the cloud and other devices. MQTT is one of the most popular M2M – communication protocols for the Internet of things at present. The advantages of this channel are data packets of a minimum size and the ability to distribute packets to several applications at the same time [12] [13][14]. MQTT was designed to implement a publisher-subscriber message exchange pattern. This approach allows applications to operate independently over the network and provide scalable solutions for IoT applications. The Publisher / Subscriber Template is used when multiple clients generate data to be sent to an intermediary who coordinates incoming and outgoing messages. In this case, the client does not know about the recipients of his message. When devices

communicate, messages go through what are called themes. Themes are part of the MQTT protocol. By topic, data is transferred between publisher devices and subscribing devices. A message can be sent only if the device publishes a message to a subject, and this message can be received by another device only if it is subscribed to this theme. Figure 3. shows the scheme of interaction of devices using the MQTT protocol.

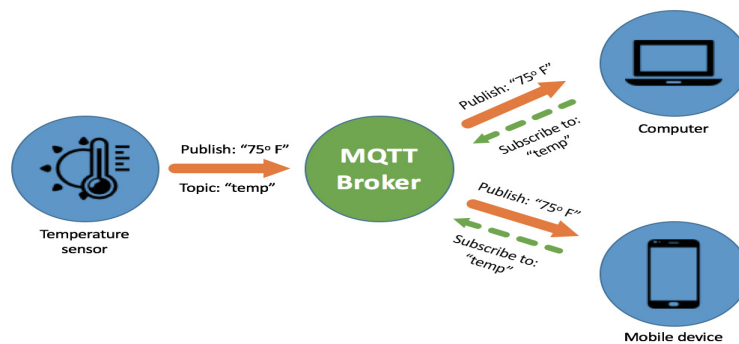


Fig. 3. MQTT protocol for device interaction

2.3 Word recognition task

The template method is one of the classic methods used to recognize handwritten characters[15] [16] [17]. It is based on comparing recognizable characters with a certain set of template images. Since in this case it is required that the dimension of the images be the same, the input image is normalized. For each input image, correlation coefficients are calculated, which reflect the difference between this image and the template. This method is simple to implement, has a relatively high recognition rate, but a significant drawback is the need to write characters in accordance with a predetermined pattern. The feature recognition method is to select a certain feature vector for the symbol image. The constructed vector is compared with some set of reference vectors by counting, for example, the Euclidean distance between these vectors. The advantage of this method lies in its immunity to character drawing, inclination and proportions. The disadvantage is the instability in relation to various image defects. In the structural method of recognition, the image is converted into a topological representation, which contains information on the relative position of the structural elements of the symbol. The disadvantages of this method are similar to those of the indicative one. The use of neural networks is based on the learning process. This method requires a large amount of data for training. The neural network based on the training sample and tags adjusts its weights. This approach is highly efficient in handwriting recognition. Next, we consider an algorithm based on a combination of neural networks, allowing to recognize handwritten text.

2.4 Handwriting recognition algorithm based on neural networks

An algorithm has been developed for handwriting recognition, which is a composition of neural networks. It consists of the sequential application of a convolutional neural network and a recurrent neural network. The algorithm is shown in Figure 4.

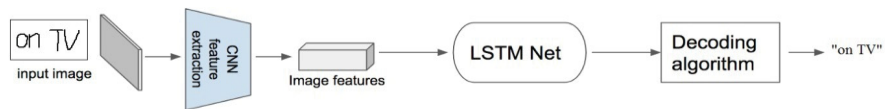


Fig. 4. Algorithm for handwriting recognition

The first step is to input the image to the input of the convolution neural network to extract features that are most characteristic of this image. The next step is the application of recurrent neural networks to the tensors of the received features. Next comes the decoding algorithm. This decoding algorithm takes the output of the recurrent neural network from each time step and produces final labeling. The IAM dataset, which consists of 79 different symbols, was used to train this neural network. The IAM dataset contains handwriting images in English [18] [19]. It includes 115,320 images with words and corresponding labels. Figure 5. Shows an example of an image from the IAM database.



Fig. 5. Image in IAM database

Thus, a black and white image with a size of 128×32 pixels is input to the model. Typically, images from a dataset do not have the desired size, so for learning, their size changes until it is 128 pixels wide and 32 pixels high. Next, the image is normalized, which simplifies the recognition task for the neural network. On the basis of the image obtained, the neural network is trained. The concept of neural networks is based on attempts to reproduce the human nervous system [20] [21]. Namely, the ability of the nervous system to learn and correct mistakes. The human nervous system is a complex network of structures that ensures the interconnected behavior of all body systems. A neuron receives signals from other neurons through several input channels. Each input signal passes through the weights. The weighted sum of the input data is calculated, the activation function is applied to it, and the signal from this neuron to the synapses of other neurons is determined.[22] [23] [24].

In Figure 6. an algorithm for recognizing handwritten words is presented the black and white image transmitted to the input of the neural network has the following form: height is 32, width is 128 and the number of channels is 1. This image is transferred to the convolution neural network to extract the signs. The output is a sequence of 32 elements, the element corresponds to the step. Each sequence is a vector with a length of 256 elements. Consider the architecture of convolution neural networks. CNN is a class of neural networks that specialize in processing data like a matrix. Images has just such a structure.[25] [26] [27] In the convolution neural network-CNN there are 3 layers: (1). Container layer (2). Sub-sampling layer (3). Completely connected layer.[28] [29] [30] [31].

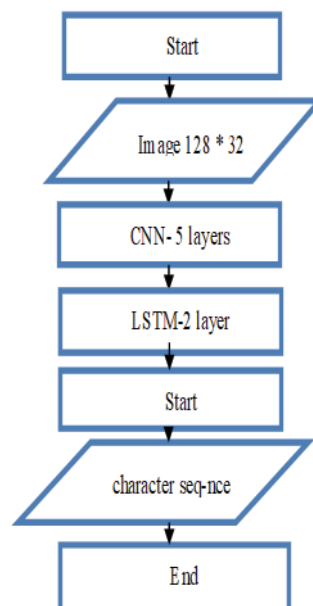


Fig. 6. Scheme of the algorithm for handwriting recognition

3 Development of Software Instrumentation

3.1 Architecture and composition of the mobile application

The mobile application for managing IoT devices interacts with the cloud service. Figure 7. shows the general interaction scheme with the cloud service AWS IoT.

To implement this interaction scheme, the following architecture is proposed, as shown in Figure 8. The interface for entering handwritten words allows you to enter commands. It is an area for drawing. This creates an image with handwritten text according to the following levels:

- The image analysis module recognizes text, which is a command to the IoT device, from the image.
- The device control command module provides the ability to create, edit, delete devices and their corresponding commands.
- The cloud service interaction module manages connections to AWS IoT using credentials and subscribes and publishes to the topics in the MQTT protocol corresponding to the commands.

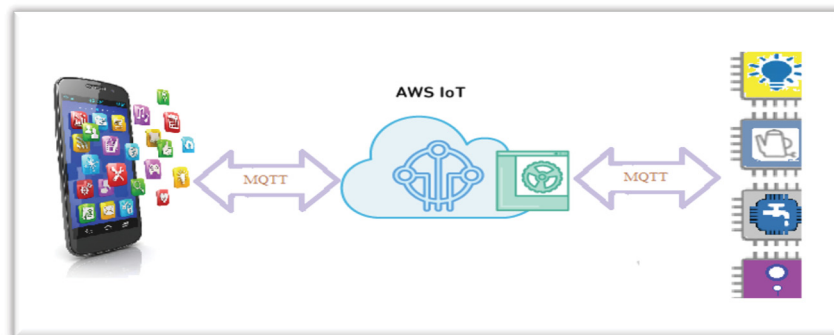


Fig. 7. The general scheme of interaction with the cloud service AWS IoT

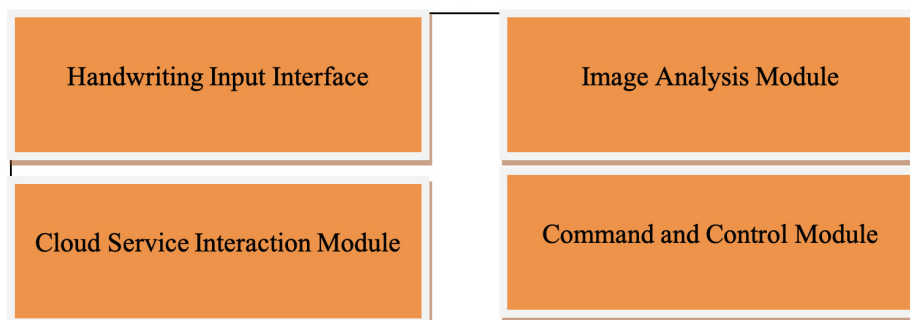


Fig. 8. Mobile Application Architecture

To start working with the system, the user must configure the credentials that will provide access to the cloud service using this application. You also need to enter devices and commands for them in the application settings.

- The user's communication with the application is as follows: the client writes text on the screen of the mobile device, then he presses the save button, after which the received handwritten text is saved as an image.
- The processed image gets to the input of the neural network and, as a result, we have the recognized text. There is a comparison of the received word with a set of commands laid down for controlling devices, for example, home appliances. If the command is found, the corresponding message is sent to the cloud service.

- If the command is not found, the user is notified that there is no recognized command.

3.2 Modules of mobile application

The mobile application implemented in this paper consists of the following modules:

- Module for writing handwritten words
- Module for recognizing handwritten words
- Module converts text into a command to the device
- Module for accessing the server

The handwriting input module provides an interface that allows you to enter commands. It includes the logic of drawing words. When implemented, the input area corresponds to the width of the screen. The Path object stores the coordinates of the gesture on the screen. When a finger touches the screen, the current state of the Path is cleared, and instead we translate the Path to the current touch point. After the motion effect is detected, the delta of motion along the x and y axes is calculated. Lines are Bezier curves. When a finger is lifted from the screen, the final version of the line is saved and the current path is reset so that there are no failures. The application allows you to draw words with several gestures. In addition, the current location of the finger on the screen, which will be needed after drawing this part of the word.

3.3 Results and discussion

The user can add a new device to the system, as shown in Figure 9. Add the device "TV".

When clicking on any element, a transition to a list of commands that are feasible for the device in the system. The created device does not yet have commands. Add a command to turn on the TV, as shown in Figure 10. To do this, enter the name of the command, the topic through which the device will interact, and a message for publication in the topic. Thus, a command is set to control a device connected to the AWS IoT cloud service.

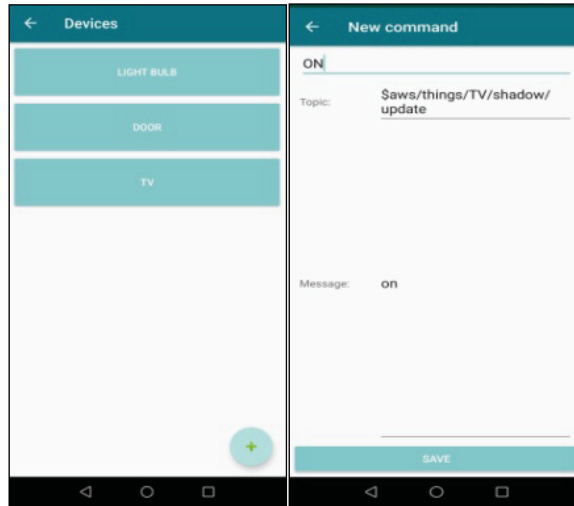


Fig. 9. Adding a device

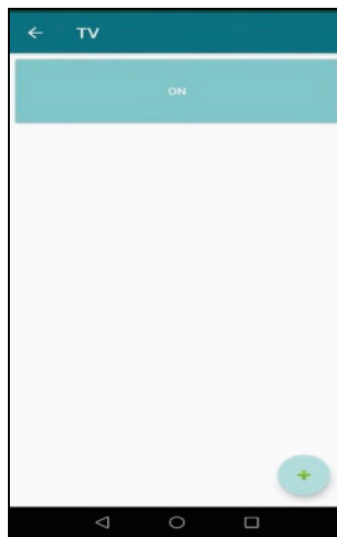


Fig. 10. Adding a command

The system is configured. A command is activated by a device when it is entered by hand on the main screen of the application. An example is shown in Figure 11. The user specifies the command and device, then presses the button on the right at the bottom of the screen. Enter the command to turn on the TV.



Fig. 11. Message with successful recognition and sending commands



Fig. 12. Message with no command entered in the system

The system recognizes the received command and establishes compliance with the command that allows to control the Internet of Things devices.

If you enter a command or device that is not in the system, as shown in Figure 12, a message appears on the screen stating that this command was not found.

4 Conclusion

Developing a system to recognize handwritten characters remains an urgent task. Many software solutions that are widely available on the market do not solve the problem in full. To develop algorithms and methods for controlling devices based on handwritten words, this will establish a correspondence between the entered command and the command specified in the system, and interact with the cloud service via messages in the MQTT protocol. Moreover, most of the existing programs are paid, which makes them difficult to use. Text recognition systems can be applied in areas that require the processing of a large number of handwritten documents such as filled out forms, questionnaires, and in public institutions.

5 References

- [1] Gyeonghwan Kim, Venu Govindaraju, Sargur N. Srihari (1999), International journal on Document Analysis and Recognition, An architecture for handwritten text recognition systems ,2(1), pp.37-44. <https://doi.org/10.1007/s100320050035>
- [2] D. Bouchaffra, et al, (1996), Incorporating Diverse Information Sources in Handwriting Recognition Postprocessing. International Journal of Imaging Systems and Technology (IJIST), 7.

- [3] Z. Razak, et al, (2008), Off-line handwriting text line segmentation: a review, *International Journal of Computer Science and Network Security*, ,8 (7) pp.12–20.
- [4] Abas Setiawan, Reza Pulungan. (2018), Deep Belief Networks for Recognizing Handwriting Captured by Leap Motion Controller, *International Journal of Electrical and Computer Engineering (IJECE)*, ,8(6), pp.4693-4707. <https://doi.org/10.11591/ijece.v8i6.pp4693-4704>
- [5] W. Wiharto, H. Kusnanto, and H. Herianto, (2017), “Hybrid system of tiered multivariate analysis and artificial neural network for coronary heart disease diagnosis,” *International Journal of Electrical and Computer Engineering (IJECE)*,7(2), pp. 1023. <https://doi.org/10.11591/ijece.v7i2.pp1023-1031>
- [6] L. Likforman Sulem, A. Zahour, B. Taconet, (2007) “Text line segmentation of historical documents: a survey”, *International Journal on Document Analysis and Recognition (IJ DAR)*, Vol. 9(2), pp. 123-138. <https://doi.org/10.1007/s10032-006-0023-z>
- [7] Md. Rashedul, et al. (2017) *Mobile Application and Its Global Impact*. *International Journal of Engineering & Technology (IJET)*,10 (6), pp.104-111.
- [8] Raffaele Garofalo . (2011) *Building Enterprise Applications with Windows Presentation Foundation and the Model View ViewModel Pattern*, Book.
- [9] S. Hyrynsalmi. (2014), *Letters from the War of Ecosystems — An Analysis of Independent Software Vendors in Mobile Application Marketplaces*. Doctoral dissertation, University of Turku, Finland.
- [10] Lukas Bereuter, et al, (2018), A Novel Communication Method for Wireless Pacemaker Synchronization, *Jacc: Basic to Translational Science*,3(6), pp.813-823.
- [11] Alexandre Schmitt, Florent Carlier, Valerie Renault , (2018), Dynamic bridge generation for IoT data exchange via the MQTT protocol ,The 9th International Conference on Ambient Systems, Networks and Technologies, pp.90-97, ANT. <https://doi.org/10.1016/j.procs.2018.04.016>
- [12] P. Zhu, J. Han, Y. Guo, and F. Lombardi. (2016), Reliability and criticality analysis of communication networks by stochastic computation. *IEEE Network*, 30(6):70–76, November. <https://doi.org/10.1109/mnet.2016.1500221nm>
- [13] Dinesh Thangavel, Xiaoping Ma, Alvin Valera, Hwee-Xian Tan, and Colin Keng-Yan Tan. (2014), Performance evaluation of MQTT and CoAP via a common middleware. In 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), pp.1–6, April. <https://doi.org/10.1109/issnip.2014.6827678>
- [14] Teddy Surya Gunawan, et al, (2018), Performance Evaluation of Smart Home System using Internet of Things, *International Journal of Electrical and Computer Engineering (IJECE)*,8(1), pp.400-411. <https://doi.org/10.11591/ijece.v8i1.pp400-411>
- [15] Vandana Reddy, Gayathri P, (2019), Integration of internet of things with wireless sensor network, *International Journal of Electrical and Computer Engineering (IJECE)*,9(1), pp.439-444. <https://doi.org/10.11591/ijece.v9i1.pp439-444>
- [16] Abdel Karim M. Baareh, et al, (2018), Performance Evaluation of Edge Detection Using Sobel, Homogeneity and Prewitt Algorithms, *Journal of Software Engineering and Applications*,11(11), PP. 537-551. <https://doi.org/10.4236/jsea.2018.1111032>
- [17] Ghazi Shakah, (2018), The problem of image segmentation and de-noising methods and various approaches to its solution, *International Journal of Engineering & Technology*,7(4),2018, pp. 5297-5301.
- [18] U. Marti and H. Bunke. (2002), The IAM-database: An English Sentence Database for Off-line Handwriting Recognition. *Int. Journal on Document Analysis and Recognition*,5, pp. 39 - 46. <https://doi.org/10.1007/s100320200071>
- [19] M. Zimmermann and H. Bunke. (2000), Automatic Segmentation of the IAM Off-line Database for Handwritten English Text. In *Proc. of the 16th Int. Conf. on Pattern Recognition*, 4, pp.35-39. <https://doi.org/10.1109/icpr.2002.1047394>

- [20] Keun Young Lee, Namil Chung, Suntae Hwang, (2016), Application of an artificial neural network (ANN) model for predicting mosquito abundances in urban areas, *Ecological Informatics*, 36, pp.172-180. <https://doi.org/10.1016/j.ecoinf.2015.08.011>
- [21] Victor Krasnoprosin, Ghazi Shakah, Aleksandr Valvachev, (2012), Monitoring and Management of Natural Territorial Complexes, *Advances in Computing*, 2(1): pp.1-5. <https://doi.org/10.5923/j.ac.20120201.01>
- [22] Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, (2012), Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, pp. 1097-1105. <https://doi.org/10.1145/3065386>
- [23] Ali.T.Al-Oqaily, Ghazi Shakah, (2018), Solving Non-Linear Optimization Problems Using Parallel Genetic Algorithm, 8th International Conference on Computer Science and Information Technology (CSIT). IEEE, pp.103-106, <https://doi.org/10.1109/csit.2018.8486176>
- [24] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. (2011), Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug): pp.2493–2537.
- [25] P. J. Sousa, R. Tavares, P. Abreu and M. T. Restivo, NSensor (2017), *International Journal of Interactive Mobile Technologies (IJIM)*, Wireless Sensor Network for Environmental Monitoring, vol 11, no 5. <https://doi.org/10.3991/ijim.v11i5.7067>
- [26] AI Kuzmich, G Shakah, AN Valvachev, (2011), Remote monitoring system for mobile objects, proceeding of 10-th International conference (PRIP2011), pp.427-430.
- [27] Rikiya Yamashita, et al, (2018), Convolutional neural networks: an overview and application in radiology, *Insights into Imaging*, 9, pp.611-629.
- [28] Guang-Bin Huang and Haroon A. Babri, Upper Bounds on the Number of Hidden Neurons in Feedforward Networks with Arbitrary Bounded Nonlinear Activation Functions, *IEEE Transactions on Neural Networks*, 9, (1), January. <https://doi.org/10.1109/72.655045>
- [29] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, Xiaohui Xie, (2018), Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp.3697-3707.
- [30] P. V. Dung, (2012), “Online handwriting recognition using multi convolution neural networks”. In *Proceedings of The Ninth International Conference on Simulated Evolution and Learning (SEAL 2012)*, Hanoi, Vietnam, pp. 310-319, December. https://doi.org/10.1007/978-3-642-34859-4_31
- [31] Md. Asifuzzaman Jishan, et al, (2019), Natural language description of images using hybrid recurrent neural network, *International Journal of Electrical and Computer Engineering (IJECE)*, 9(4), pp.2932-2940. <https://doi.org/10.11591/ijece.v9i4.pp2932-2940>

6 Author

Ghazi Shakah is a research scholar in Ajloun National University, Jordan. He is working as an associate professor in the Faculty of Information Technology, Department of Computer Science. He has more than 15 years' experience in teaching. His research interest is image processing, artificial intelligence, multi-agents systems. He is a member of the General Committee of the Association of Colleges of Computers and Information (CCIS). Of the Association of Arab Universities Colleges of Computer and Information Society.

Article submitted 2019-11-04. Resubmitted 2019-12-10. Final acceptance 2019-12-14. Final version published as submitted by the authors.