BUSINESS INFORMATICS INFORMATYKA EKONOMICZNA 3(61) • 2021

ISSN 1507-3858
e-ISSN 2450-0003

Ilhan Tarimer
Mugla Sitki Kocman University
e-mail: itarimer@mu.edu.tr
ORCID: 0000-0002-7274-5680

Buse Cennet Karadağ
Mugla Sitki Kocman University
e-mail: busecennet5@gmail.com
ORCID: 0000-0002-2488-1047

Michał Flieger
Adam Mickiewicz University in Poznań
e-mail: flieger@amu.edu.pl
ORCID: 0000-0002-8430-8883

MAKING DATA REVIEW BY A NOVEL CALCULATION ALGORITHM FOR DETECTING OBJECTS' VIEWS

WYKONYWANIE PRZEGLĄDU DANYCH PRZEZ NOWOCZESNY ALGORYTM OBLICZENIOWY DO WYKRYWANIA WIDOKÓW OBIEKTÓW

DOI: 10.15611/ie.2021.3.05
JEL Classification: A

Quote as: Tarimer, I., Karadag, B. C., and Flieger, M., Making data review by a novel calculation algorithm for detecting objects' views. *Business Informatics*, (3).

Abstract: In this study, an image processing was performed with a software developed in Python programming language by capturing object views. In this context, an algorithm was created to calculate the amount of substance in a particular container at a fixed camera angle; the measured substance amount as information was sent to smart devices operating with Android Systems. The amount of information transmitted was shown in the user interface as the form of a percentile, thanks to an algorithm and the developed program. The algorithm in this software performs cropping and conversion to black and white format on the captured photo using image processing steps, and it calculates by scanning the black areas and proportionally these values to the area of the container, giving information about the level of the substance.

It is possible to use the developed software as an alternative detection and evaluation system for object detection, image processing, and level control processes.

Keywords: software, algorithm, data, image, objects, things.

Streszczenie: W badaniu przetwarzanie obrazu zostało wykonane za pomocą oprogramowania opracowanego w języku programowania Python poprzez przechwytywanie widoków obiektów. W tym kontekście stworzono algorytm do obliczania ilości substancji w konkretnym pojemniku przy stałym kącie kamery; zmierzona ilość substancji jako informacja została wysłana do inteligentnych urządzeń działających z systemami Android. Ilość przesyłanych informacji została pokazana w interfejsie użytkownika w postaci percentyla dzięki opracowanemu algorytmowi i programowi. Algorytm w tym oprogramowaniu wykonuje kadrowanie i konwersję do formatu czarno-białego na przechwyconym zdjęciu za pomocą etapów przetwarzania obrazu, oblicza, skanując, czarne obszary i proporcjonalnie te wartości do obszaru pojemnika, podając informacje o poziomie substancji. Możliwe jest wykorzystanie opracowanego oprogramowania jako alternatywnego systemu detekcji i oceny procesów detekcji obiektów, przetwarzania obrazu i kontroli poziomu.

Słowa kluczowe: oprogramowanie, algorytm, dane, obraz, obiekty, rzeczy.

1. Introduction

Today, with the development of technology, image processing applications have increased visibly, especially in security, agriculture, health and other field applications. It is known that they make various contributions to the Internet of Things processes (Ağin and Malasli, 2016; Çavdaroğlu, 2017; Kırkaya, 2020; Osmanoğlu, Mutlu, Gürsoy, and Şanlisoy, 2019). In studies using image processing techniques, first images of the environment or object to be worked on are taken with a camera. After performing some operations on these images, various inferences are made about the object being studied.

Yaman and Aktürk calculated the densities of passengers waiting in metro transmission lines and stations as a ratio by using image processing techniques in their papers (Yaman and Aktürk, 2001, pp. 12-14). Banerjee et al. proposed and implemented the design of a secure sensor node prototype that communicates between a mobile phone and monitoring equipment via Bluetooth using the RC4 encryption algorithm (Banerjee, Sethia, Mittal, Arora, and Chauhan, 2013). Albayrak et al. designed an embedded system that will control temperature from Android devices via a web service (Albayrak, Koçer, and Uslu, 2013). Vijayalaxmi et al. interfaced the video/picture data they took continuously in less than 10 ms with the help of a webcam to Raspberry Pi to perform object detection and tracking by color (Vijayalaxmi, Srujana, and Kumar, 2014). Nguyen et al. wrote a motion detection algorithm in Python, explaining its design and implementation in Raspberry Pi (Nguyen, Loan, Mao, and Huh, 2015). In their joint paper, Canh and Chon proposed using city data to detect mobile customers and objects in HoChiMinh city with Raspberry Pi (Khuu Minh and Le Trung, 2014). In the joint article by Ferdoush and Li, a wireless sensor network system was developed using Arduino and Raspberry Pi (Ferdoush and Li, 2014). In a study by Ujjainiya and Chakravarthi, the images taken by the USB webcam were comparatively analysed with a program written in OpenCV for Raspberry Pi (Ujjainiya and Chakravarthi, 2015). Vujović and Maksimović proposed a sensor web node application that monitors the Raspberry Pi for a possibility of fire in a building (Vujović and Maksimović, 2015). Jindarat and Wuttidittachotti presented a chicken farming management application for an enterprise using Raspberry Pi, Arduino Uno, an embedded system and a smartphone (Jindarat and Wuttidittachotti, 2015). In an article by Simaria and Papila, a video stream is taken that detects and monitors an object in real time with a camera with a rotation angle of 360° (azimuth) (Simaria and Papila, 2015). Yüzgeç and Aba implemented a smart home application for various scenarios using Raspberry Pi (Yüzgeç and Aba, 2017). Anandhalli and Baligar proposed a video image processing algorithm that detects, tracks and finds the number of vehicles on a road (Anandhalli and Baligar, 2018). In the joint articles of Dursun and Çuhadar, a representation environment created for working with secure data link and a new method for video streaming for UAVs developed in this environment are presented. In the representation environment, a data link system was designed with Raspberry Pi 3, Picamera and https for video streaming and the method was verified by experiments (Dursun and Çuhadar, 2018). In Prokopov's thesis, a real-time and robust background extraction was performed with the Gaussian Mixture model to analyse the traffic. For this, the background removal algorithm was investigated and applied (Prokopov, 2016).

In another study, an algorithm and a program were written for a level measurement application by using image processing techniques on a picture taken via Raspberry Pi. In the written program, provided that the prepared card and hardware can see the substance, users can learn the amount of the substance simultaneously via the mobile interface. In the second part of this study, the materials used, image processing and the methods used to store the image are given, in the third part, information about the developed application is presented and in the fourth part, the results of the experiments with the application are shown.

2. Material and method

With a mechanism established in this study, the increase and decrease in the amount of the substance in the container is recorded instantly. In the established system, the area occupied by the substance in the container is determined by applying image processing on the images taken with the camera, and the amount of this substance is recorded as a percentage value.

2.1. Image processing

From the moment the image is digitized and transmitted to the computer environment, the pictorial information is passed through some processes such as orientation and analysis in order to obtain the desired information from the image. In the first stage, the reflected light from a light source is transferred to the camera in optical form to provide the image. These rays, which hit and return to the object, identify the object and are converted into electrical signals within the camera. Meanwhile, the image is in the form of analog signals, which are digitized with the help of a digital converter and transferred to the computer environment for processing.

After the digital image is taken, the preprocessing step is started. At this stage, the image may be subject to being improved so that it can be processed more easily and work with the least error (errors that may occur due to scanning) in interpretation and analysis. Thanks to this the image is clarified, the impurities in the image are filtered and the structural defects on the image are eliminated or minimized.

At the final stage of processing (which is aimed to reduce visual pollution and ensure that the area to be processed is revealed more clearly) the image is divided into sub-segments (segmentation process).

2.2. Software platforms used

Wireless Internet environment was used for inter-object communication. Data sending and receiving data processes were carried out using wifi. The level of the substance in the container was sent to the database in the ThingSpeak cloud environment. The ThingSpeak application is an API that uses the HTTP protocol to store and retrieve data from objects or sensors (Babayiğit and Büyükpatpat, 2019). At the same time, it can be easily integrated with other platforms thanks to its applications and API keys (Sazak and Albayrak, 2017).

The data taken from the installed mechanism are saved in the ThingSpeak environment. This data is then accessed through the developed mobile applications.

2.3. Development of the application

Raspberry Pi, one of the assembly units prepared for this study, was programmed with the installed Python Image libraries. The designed setup was run by testing the connected camera module and libraries. A database was created in ThingSpeak to load and extract data. At the same time, the created database can be accessed via a smart device. The interface of the application on the smart device is given in Figure 1. Access to instant data is provided by entering the "Channel ID", where it is recorded, and the "Read API Key" of the channel to read the recorded data given in this interface.

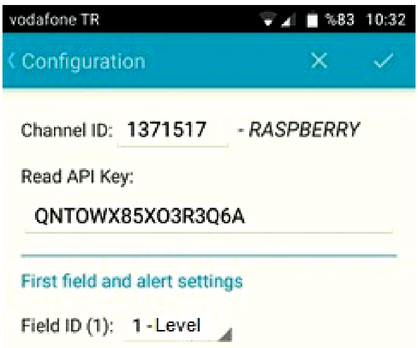


Fig. 1. Configuring the application for accessing data on the smart device

Source: own source.

This database records the percentage value of the area occupied by the substance in the container. The program updates the level value each time and saves the data along with the time the data were taken. Some of the received data are given in Table 1.

Table 1. Data taken from ThingSpeak

Date	Time	*Id*	*Level*	Date	Time	*Id*	*Level*
26.05.2021	11:37:07	1	50.61	26.05.2021	12:03:01	11	42.88
26.05.2021	11:37:26	2	49.55	26.05.2021	12:03:20	12	42.85
26.05.2021	11:37:46	3	49.33	26.05.2021	12:07:26	13	45.76
26.05.2021	11:38:10	4	48.9	26.05.2021	12:07:46	14	44.58
26.05.2021	11:57:43	5	43.53	26.05.2021	12:08:06	15	43.65
26.05.2021	12:00:05	6	45.15	26.05.2021	12:08:29	16	44.04
26.05.2021	12:00:22	7	44.2	26.05.2021	12:08:50	17	43.75
26.05.2021	12:01:59	8	43.41	26.05.2021	12:09:14	18	43.4
26.05.2021	12:02:20	9	42.95	26.05.2021	12:09:36	19	43.41
26.05.2021	12:02:40	10	43.01				

Source: https://docs.google.com/spreadsheets/d/1vjwObf2q7n_C3qZ9agFrh3f0kpmpv3NE/edit#gid=1337640980.

The amount of data given in Table 2.1 is around 1360, some of which are shown in the table, while others are registered in the database.

Data generated from the field are recorded in a channel created on the ThingSpeak platform. The data saved in this channel are shown in the chart given in Figure 2.

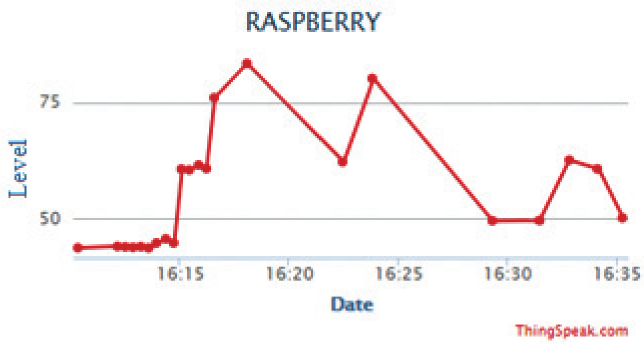


Fig. 2. Data saved in the Thingspeak

Source: https://thingspeak.com/channels/1371517/private_show.

In order to access the database established in this study, the Web service method was chosen. Then, data sending operations were performed on the Raspberry Pi platform with simple code phrases. The program's algorithm:

1) sets the background to be white in the setup (white background plays a big role in the separation of the substances in front of it),

2) takes pictures with the Raspberry Pi camera module at 10 second intervals and at the same resolution rates, then rearrange the image taken according to the image processing steps given in Figure 3.

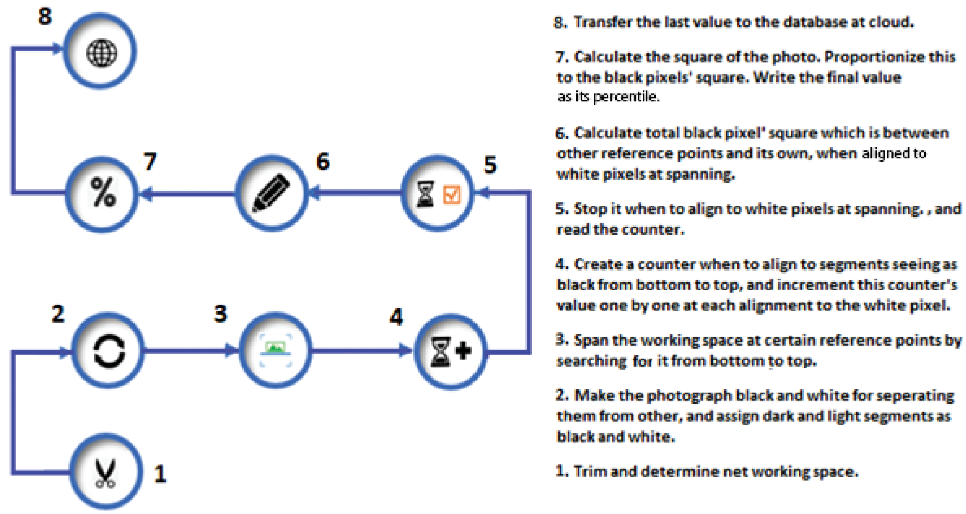


Fig. 3. Image processing flowchart

Source: own elaboration.

The processing of the received image is shown in the work flowchart in Figure 3.

2.4. Calculation algorithm

The calculation algorithm was used to determine the amount of substance contained in the container. In order to make the calculation, the image is first scanned from the top to the bottom. The scan area consists of the image processing steps and the dimensions of the final image obtained after cutting. In the picture where the recorded pixel value is zero, that is, converted to black-white, fields representing the black colour are recorded by the counter.

After the scanning process is completed, the total area occupied by the substance in the container is calculated. While calculating the ratio according to the container in which the item amount is located, the total value of the black pixels is multiplied by 100 and the obtained value is divided by the area of the picture. The mathematical expression of this ratio is given as in equation 1.

$$rate = \frac{(sum\ of\ black\ pixels \times 100)}{image.width \times image.height}.$$

In this equation the sum of black pixels variable represents the sum of the black areas of the image in black and white format, the image.width value represents the horizontal plane of the image, and the image.height value represents the vertical plane of the image. When the ratio of these values to each other is calculated, the information about how much the substance in the container covers is revealed. With this value, information can be obtained about the amount of the substance in the container. If the amount of the substance in the container is small, the substance can be added.

3. Findings and discussion

In this application, an algorithm that calculates the amount of substance in a certain container was written, and the user was informed with the obtained data. In the related studies, the Raspberry Pi and the container assembly were placed as necessary, and after the application was run, a photo frame as in Figure 4. was captured.

After this situation was achieved, the unnecessary parts in the resulting image were cut from the adjusted points so that the entire container was obtained. With the image processing steps given in Figure 3. With the image processing steps applied and given the foreground and background of the photograph, namely the substance and the container, are separated from each other and the final image obtained is given in Figure 5.

Fig. 4. The first photo captured

Source: own elaboration.

Fig. 5. Photo edited using image processing program

Source: own elaboration.

In the next step, the program was provided to continuously capture photos in an endless loop and apply image processing techniques. As a result of these techniques, the amount of substance in the container was calculated at 10-second intervals by scanning the photograph.

In order to control the operation of the calculation algorithm, the sum of the black pixels and the area of the container in which the substance is located were taken from the system. These data are given in Figure 6.

When the data in Figure 6. were placed in the calculation algorithm given in equation 1, the result obtained was:

$$rate = \frac{(419191 \times 100)}{546975} = 76{,}8380.$$

The calculated occupancy value was sent to the ThingSpeak environment with the web service. The occupancy value sent to the database was displayed via the application installed on the smartphone. This value was displayed in the Android interface as shown in Figure 7.

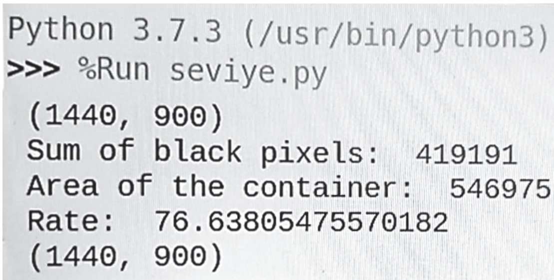


Fig. 6. Necessary data for the calculation algorithm

Source: own elaboration.

Fig. 7. View occupancy rate on smart device

Source: own elaboration.

In the application, the user is informed if the substance falls below the determined level. In addition to this information, the level values of the substance in the container are displayed graphically (Figure 8).

While testing the established system from different locations, it was observed that the algorithm calculated the item amount with an error margin of 6.21% . It was determined that this margin of error was caused by the light falling on the container. The first image taken from the camera is given in Figure 9 and the ratio calculated over this image is given in Figure 10. Values and visuals showing this margin of error are given in the figures below.

After this calculation, the container in which the substance was placed was moved a little (2cm back), and the amount of the substance was recalculated. The images taken from the camera in the new location are given in Figure 11 and the ratio calculated over this image is given in Figure 12.

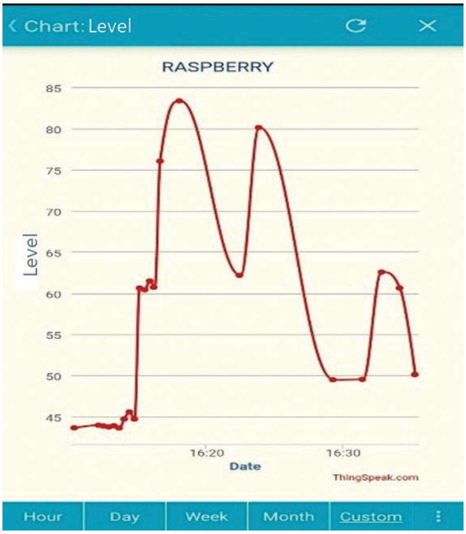
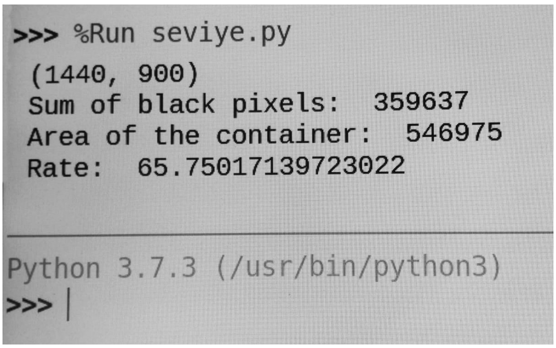


Fig. 8. Displaying data graphically on a smart device

Source: own elaboration.

Fig. 9. The first image taken from the camera and its processed version

Source: own elaboration.



```
>>> %Run seviye.py
 (1440, 900)
 Sum of black pixels:  359637
 Area of the container:  546975
 Rate:  65.75017139723022

Python 3.7.3 (/usr/bin/python3)
>>> |
```

Fig. 10. Ratio calculated over the first image taken from the camera

Source: own elaboration.

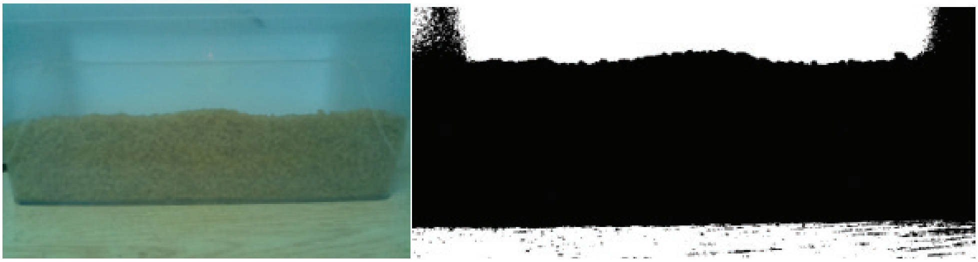


Fig. 11. Image taken from the camera in a new location and the processed state of image

Source: own elaboration.

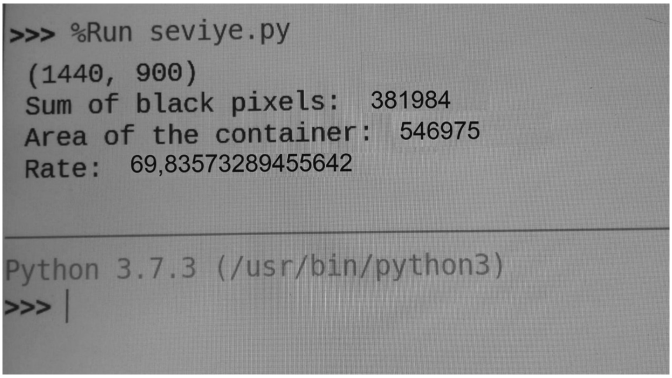


Fig. 12. Ratio calculated over image in a new location

Source: own elaboration.

4. Conclusion

In this study, Raspberry Pi worked with a ready-made camera module, Raspbian operating system and interface in the installed hardware. This hardware used software to enable the ThingSpeak environment to work on a mobile device as well.

In the developed system, the occupancy rate of the container imaged by the camera was determined by the calculation algorithm in this study. The resulting image was converted to the black-and-white format so that the algorithm could determine the ratio. The container was scanned in the horizontal/vertical plane on the new image. The number of black areas (areas with zero pixel values) detected in the scanning area with the scanning process was determined by a counter.

After the scanning process was completed, the value of the counter where the black areas were recorded and the total area covered by the substance in the container were determined. After finding the total area, the occupancy rate of the container was determined by looking at the ratio of the pixel values of the black areas occupied by the substance.

The fill rate of the container was recorded in a field holding the value "Level" in the ThingSpeak environment. The "Level" values in this environment were updated with the data from the program at intervals of 10 seconds. These updated data were saved to the system at intervals of 18-20 seconds due to the speed of the Internet. With the application on a smart device with a user interface, these values were displayed instantly.

According to the test results, the program calculates the same amount of substance from different locations with an error margin of 6.21%. The increase or decrease of this error depends on the movement of the camera used in the project and the intensity of light reflected on the material. In case of excessive light reflection on the item, the program cannot distinguish the item from the background and ignores a certain part of it and performs the calculation process accordingly. Although the resolution of the camera is sufficient for the project, cameras with focus adjustment or night vision options can be used by using the same CSI port.

As an IoT application, it can be used to measure kitchen components in domestic applications such as a kitchen inventory tracking system. A better determination of the amount of substance can be achieved by developing the system with light or distance sensor attachments. With this system, an online ordering application can be developed in cases of a decrease in the substance. For this purpose, further versions of this study are still being developed.

References

Ağın, O., and Malaslı, M. Z. (2016). The place and importance of image processing techniques in sustainable agriculture. *Journal of Agricultural Machinery Science*, *12*(3), 199-206.

Albayrak, Y., Koçer, A., and Uslu, S. (2013). Temperature control from android devices via web services. *Academic Informatics*, *1*(27).

Anandhalli, M., and Baligar, V. P. (2018). A novel approach in real-time vehicle detection and tracking using Raspberry Pi. *Alexandria Engineering Journal*, *57*(3), 1597-1607.

Babayiğit, B., and Büyükpatpat, B. (2019). Design and implementation of IoT-based irrigation and remote monitoring system. *Turkey Informatics Foundation Journal of Computer Science and Engineering*, *12*(2), 13-19. Retrieved July 12, 2021 from https://dergipark.org.tr/tr/pub/tbbmd/issue/50642 /635505

Banerjee, S., Sethia, D., Mittal, T., Arora, U., and Chauhan, A. (2013, November). *Secure sensor node with Raspberry Pi* (2013 International Conference on Multimedia, Signal Processing and Communication Technologies (IMPACT), pp. 26-30).

Çavdaroğlu, G. Ç. (2017). Processing of traffic density map images using image processing techniques. *Academic Platform Journal of Engineering and Science (APJES)*, *5*(2), 22-28. Retrieved July 21, 2021 from https://dergipark.org.tr/en/pub/apjes/issue/30323/289448

Dursun, M., and Çuhadar, İ. (2018). Secure video streaming implementation for unmanned air vehicle (UAV) data link with Raspberry Pi 3 over https. *Journal of Information Technologies*, *11*(1), 23-28.

Ferdoush, S., and Li, X. (2014). Wireless sensor network system design using Raspberry Pi and Arduino for environmental monitoring applications. *Procedia Computer Science*, *34*, 103-110. Retrieved September 6, 2021 from https://tr.wikipedia.org/wiki/Raspberry_Pi

Jindarat, S., and Wuttidittachotti, P. (2015, April). *Smart farm monitoring using Raspberry Pi and Arduino* (IEEE 2015 International Conference on Computer, Communications, and Control Technology, pp. 284-288).

Khuu Minh, C., and Le Trung, C. (2014). Proposal of the 3D and temporal GIS for detecting moving objects with Raspberry Pi moving client base in urban conditions, GeoInformatics for Spatial-Infrastructure Development in Earth & Allied Sciences (GIS-IDEAS). HoChiMinh City.

Kırkaya, A. (2020). *Smart Agricultural Technologies Applications* (3rd International Conference on Awareness, pp. 1-12).

Nguyen, H. Q., Loan, T. T. K., Mao, B. D., and Huh, E. N. (2015, July). Low cost real-time system monitoring using Raspberry Pi (7th International Conference on Ubiquitous and Future Networks (IEEE 2015), pp. 857-859).

Osmanoğlu, U. Ö., Mutlu, F., Gürsoy, H., and Şanlısoy, S. (2019). Use of image processing and analysis in medicine and an application. *Osmangazi Journal of Medicine*, *41*(1), 6-16. Retrieved July 12, 2021 from https://doi.org/10.20515/otd.426347

Prokopov, A. (2016). *Moving object detection and tracking using dynamic background and foreground separation for the purpose of traffic analysis on mobile devices (Bachelor's thesis)*. Tartu: University of Tartu Institute of Computer Science.

Sazak, T., and Albayrak, Y. (2017). A system design that collects and remotely tracks environment data on the Internet of Things (IoT) (19th Academic Informatics Conference, Aksaray/Turkey).

Simaria, R. D., and Pipalia, D. S. (2015, May). Real time object detection & tracking system (locally and remotely) with rotating camera. *International Journal on Recent and Innovation Trends in Computing and Communication*, *3*(5), 3058-3063.

Ujjainiya, L., and Chakravarthi, M. K. (2015). Raspberry-Pi based cost effective vehicle collision avoidance system using image processing. *ARPN J. Eng. Appl. Sci*., *10*(7).

Vijayalaxmi, K. A., Srujana, B., & Kumar, P. R. (2014). Object detection and tracking using image processing. *Global Journal of Advanced Engineering Technologies*, *9*(1), 113-115.

Vujović, V., and Maksimović, M. (2015). Raspberry Pi as a sensor web node for home automation, *Computers & Electrical Engineering*, *44*, 153-171.

Yaman, K., and Aktürk, N. (2001). *Determination of the person density with image processing*. Konya: UMTS, Selçuk University.

Yüzgeç, U., and Aba, Ö. (2017). Development of a smart home application using Raspberry Pi. *Bilecik Şeyh Edebali University Journal of Science*, *4*(1), 21-29.