

IOT-BASED SOLUTION FOR DETECTION DEFECTS IN INFRASTRUCTURE OBJECTS USING RASPBERRY PI

R. Mysiuk¹, V. Yuzevych^{1,2}

¹ *Department of System Design,
Ivan Franko National University of Lviv
50 Drahomanova St., UA-79005 Lviv, Ukraine
mysiukr@ukr.net*

² *Department of Electrophysical Methods of Non-Destructive Testing,
Karpenko Physico-mechanical Institute of the NAS of Ukraine
5 Naukova St., UA-79601 Lviv, Ukraine*

Monitoring the current condition of roads, materials and communication networks can often be difficult in the accessibility of the damaged place and difficult to visually assess the criticality of the condition. The current state of such infrastructure facilities makes it possible to assess data collected from sensors, and intermediate analysis is automated in microcomputers or microprocessors. For a visual assessment of the condition, a video with overlay filters was used to highlight areas of damage. The first possible type of object defect is corrosion, which is detected by applying a color filter, the second is cracks with key points filled with an edge filter. The transfer of the collected number of detected pixels of the filtered frame from the Raspberry Pi is performed using MQTT protocols in the used the Internet of Things system. The AWS IoT rule constantly reads logged messages from the subscription to AWS IoT to push the necessary data into the Elasticsearch database for further analysis.

Keywords: Internet of Thing, cloud technologies, defects, Raspberry Pi, corrosion, crack.

Overview

In many surrounding objects, various defects begin to appear over time under the influence of external factors and features of operation. The durability of the materials depends on the strength of such objects. Monitoring of the current state is used for maintenance. Metal and concrete are among the most popular objects in which defects can occur. The peculiarity of metal in comparison with concrete is that defects and cracks are formed after the corrosion process. Cracks in such objects often lead to the destruction of entire structures of larger objects.

The process of visual detection of such defects can be automated using a camera and software. Processing of image results and subsequent monitoring of changes will allow to assess the criticality of repair of a certain damaged area.

The purpose of this work is to detect defects in concrete and metal using image filtering and calculate the object damage in a Raspberry Pi-based IoT system.

Among the main tasks in the work is the integration of the Raspberry Pi microcomputer and software with cloud storage Elasticsearch and Amazon Web Service (AWS) S3 buckets for data about concrete and metal damage for future analysis.

In general, the Internet of Things in combination with Raspberry Pi is used in various fields such as transportation, medicine, agriculture, but the construction and maintenance of the current state of communication networks and roads are not very many solutions proposed based on Raspberry Pi [1]. Using the Internet of Things to process data on the state of corrosion and concrete to send to cloud storage [2, 3], but the main detection method is external sensors. Paper [4] provides a detailed literature analysis on low-cost sensors and the possibilities of their use for corrosion monitoring. Such an analysis is important for evaluating the possibility of the described integrated approach for detecting defects in various types of objects. Since the work uses a camera as a sensor on the Internet of Things in combination with a Raspberry Pi, it is worth considering existing methods for detecting moving objects [5]. Considering the topic of security cameras, some elements of a file storage system can be considered. Possibilities of data recovery in case of loss are considered [6], and localization of objects by pixel analysis [7]. For more efficient processing, the improvement of pixel processing should be taken into account [8].

Video processing and data storage in Raspberry Pi

There are different ways to detect external damage to metal and concrete. One of the ways to monitor the change of external changes in the material can be a video stream with further analysis. During the analysis, filters are applied to localize damaged areas. The number of detected pixels at a certain time of video recording is also calculated.

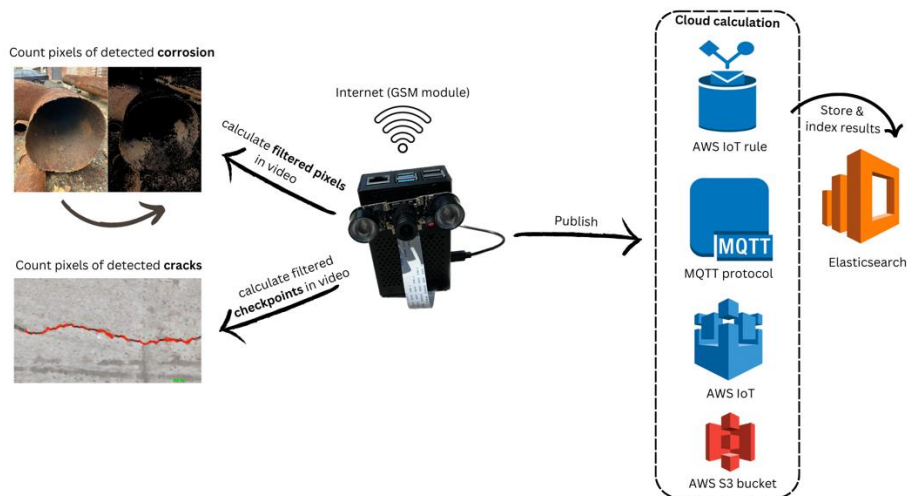


Fig. 1. Scheme of operation of the crack detection system

As shown in Fig. 1, the received information about the detected number of damaged areas of the object in the Raspberry Pi is sent to the Amazon Web Service (AWS) cloud environment for storage and analysis. The collected data from the video stream is collected in JSON format for sending to the AWS IoT subscription. This subscription works in the mode of reading the data of all messages that are sent from the device. A certain rule of AWS IoT is configured, which allows you to write only the most important information to Elasticsearch data store. Videos that contain more than a set value (more than 25% corrosion damage or more than 20 key points) are stored in S3 bucket file storage. To synchronize the time recorded in the non-

relational Elasticsearch database with the video, the recording time and date are added to the video. In this way, the comparison of the numerical results with the concrete segment of the video coincides. Message Queuing Telemetry Transport (MQTT) protocol is a messaging protocol for publishing and subscribing in machine-to-machine communication.



Fig. 2. Raspberry Pi 4 with camera

The central node in the system for processing data on object damage is the Raspberry Pi 4 based on Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit 1.8GHz. The used model contains 4GB SDRAM, 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0. In Fig. 2 this device is shown in a case with a cooling fan and a camera with infrared tracking in the dark. As an additional component, it is planned to add a GSM module for working on the Internet in real time. The existing model sends data using a shared network from a mobile phone through a Wi-Fi connection. External power is provided by several power batteries from a power bank. The camera is connected using the CSI interface located on the Raspberry Pi board. Since the microcomputer contains many connectors, interfaces and GPIO for connecting various additional devices.

It is proposed to use a Raspberry Pi 4 microcomputer with a camera with a built-in infrared filter for night mode and a 5-megapixel OV5647 sensor.

Since the camera is the main sensor of the system, the main characteristics of this element of the system are as follows:

- infrared illumination module with a light sensor, which allows you to take high-quality photos at night / in twilight and in daylight, without affecting the color rendering.
- 5-megapixel sensor OV5647.
- Photo resolution: 2592 x 1944 pixels
- Maximum resolution: 1080p

Since the Raspberry Pi microcomputer has a special Linux-like operating system installed, it allows the user to work with all commands and programs in graphical display mode. In the case of working with video, this type of work as with desktop programs allows you to

view the recorded video immediately on the device. In the future, improve the algorithm of work directly on the Raspberry Pi, without downloading it to the cloud storage.



Fig. 3. A fragment of the saved video for the investigated area of corrosion on metal after applying filters in Raspberry Pi

The system can save recorded videos, but to save memory resources, this option is disabled by default. Videos are deleted from the Raspberry Pi as soon as the bucket is successfully sent to AWS S3 file storage. It is also configured to automatically clean videos from the cloud file storage every day. Part of the recorded video is shown in Fig. 3 for detected corrosion in pipelines. Video recording is important to establish the specific location of damage to the object. A similar system is implemented on the basis of the Jetson nano Kit [9], and the system shown can be applied to this microcomputer in the following expansion steps.

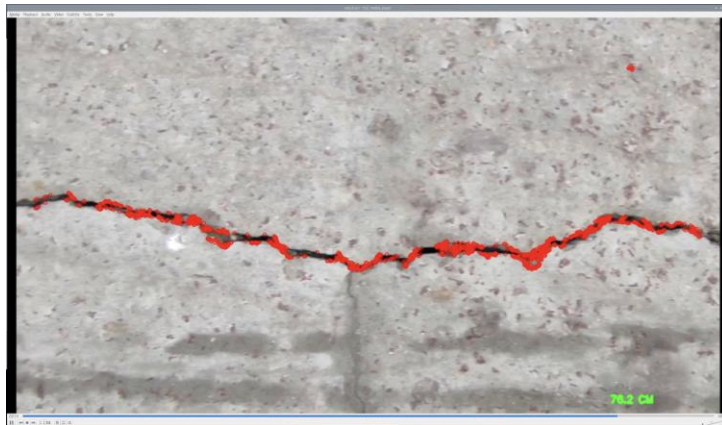


Fig. 4. A fragment of the saved video for the investigated area of cracks in concrete after applying filters in the Raspberry Pi

The process of storing data on cracks in concrete takes place in the same way as for detecting corrosion. A fragment of the video is shown in Fig. 4 with the indicated number of points, which fills the detected cracks.

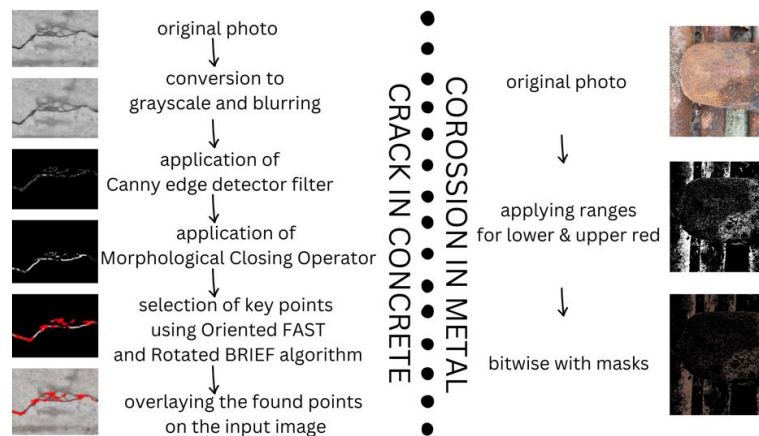


Fig. 5. A graphical sequence of steps for video processing on the Raspberry Pi in metal and concrete flaw detection model

The sequence and number of applied filters for different defects differs due to their features. As shown in Fig. 5, the Canny edge detector, the morphological closing operator and the determination of key points using the Oriented FAST and Rotated BRIEF algorithms were used to obtain a filtered image for the crack. At the first stage of filtering, averaging (blurring) with a 4x4 normalized box filter is used to hide small noises in the image. The next step is to define the edges with the Canny filter (value of 20 as a lower threshold value in Hysteresis Thresholding and 200 as an upper threshold value in Hysteresis Thresholding) and the closed parts of the fill with the Morphological Closing operator. Each of these filters is applied sequentially, the last filter is to fill the filtered part of the image with key points using the Oriented FAST and Rotated BRIEF filters according to brightness levels.

Instead, for corrosion, use the upper and lower ranges of red and bitwise with a mask. In this case, the HSL color model of the image is used. Such a model is presented in the form of a cylinder, this is a circle, this is hue, height - lightness, from the center of the circle - saturation. In the program, certain limits of the color of corrosion with value lower (0,70,70) and upper red (20,200,150) combinations of HSL values are set, which the filter passes. As a result, it is displayed a binary result from the missed and highlighted pixels. Thus, the program can distinguish an object from a video by a certain color. The last step is to apply the missing colors to the original image.

The described filtering videos are implemented in the Python programming language using the OpenCV and NumPy libraries. In addition, the same programming language is used to transfer data to the AWS cloud environment.

Integrating and sharing data in Raspberry Pi with the cloud environment

The Internet of Things (IoT) is based on the use of Internet protocols for data transmission as a tool for communication between a device and an external application. One of the ways to customize such a solution is to create your own web service, but this requires addition-

al protection during data transmission and ensuring the reliability of the created system. An alternative solution may be integration with existing services (AWS IoT, Google Cloud Platform IoT Core Azure IoT or others)

To start the integration of the Raspberry Pi with the cloud environment, keys are generated, the programming language is selected, and the installation script is downloaded to the Raspberry Pi. In this case, Python is chosen because the main video processing algorithms are written in the same language. In addition, you can download a starter script with Node.js and Java [10].

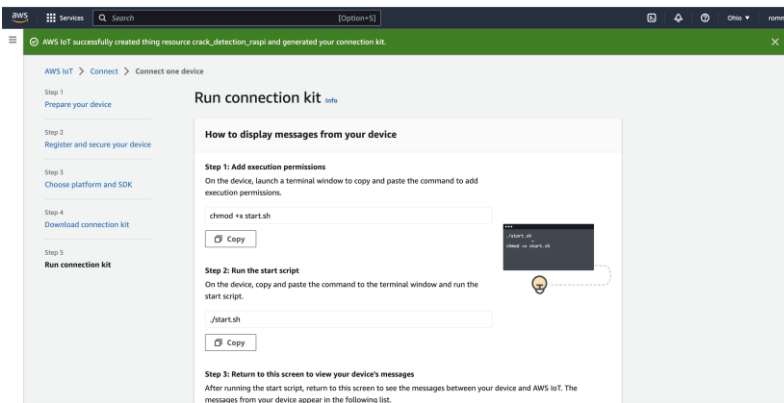


Fig. 6. Configuring access to the device with the generation of access keys and access certificates to the AWS cloud environment

As shown in Fig. 6, after downloading the package with generated certificates and examples of data transfer via the MQTT protocol, the start script should be run.

As shown in Fig. 8 received messages on the AWS side are displayed in subscriptions with certain set names. For example, the test subscript has the name `sdk/test/python`. The date of receipt of the message and the number of the received message next to the message. In the same way, the program can be modified and integrated with existing pixel-filtered video analysis to detect the largest defects in objects.

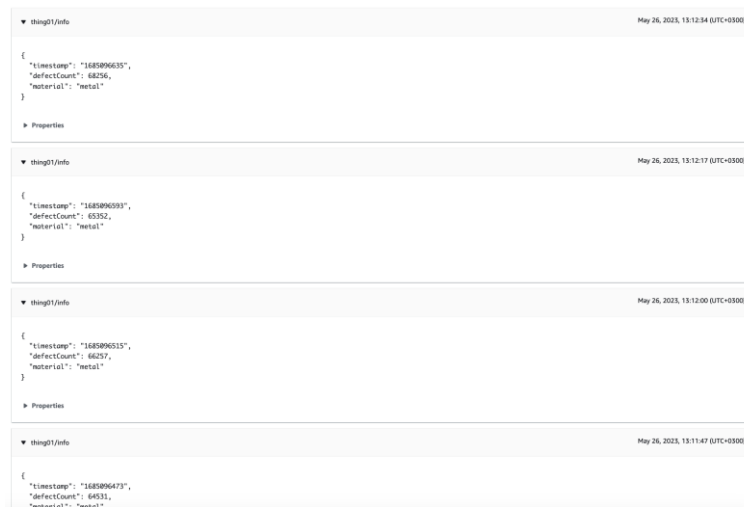


Fig. 9. Information received in the `thing01/info` topic in the AWS IoT cloud service about the number of detected pixels (metal corrosion), time and material obtained from video processing

In addition to the test format, the data can be converted into a convenient Json representation of information for processing in non-relational databases. In the key-value format, the name and value of a certain parameter are transferred, respectively. In Fig. 9 shows an example of information received from the Raspberry PI in the subscript `thing01/info`. Among the parameters, the timestamp, the number of detected pixels of the corroding image and the material are saved. The material can be either metal or concrete, accordingly, the quantitative ratios of “`defectCount`” parameter differ due to the use of different approaches to detecting the criticality of object damage.

During the exchange of device information with the cloud environment, it was observed that the delay between sending and receiving a message is minimal, it can be up to a few seconds. This may be due to the speed of the Internet and the location of the device and the cloud environment. The recording time in the video is transmitted, for more accurate detection of the location of the damage and viewing on the stored video in the AWS S3 bucket.

The number of pixels in a metal video or keypoints for a concrete video changes from second to second of the video due to the changing angle of the shot and the view of the damaged areas.

However, identifying the largest values among a large array of received logged data can be difficult during long-term video recording. Therefore, an effective way to monitor data is to establish certain rules for writing them to the database. The non-relational Elasticsearch database was chosen as a data store, in which you can perform flexible data searches with the creation of various queries.

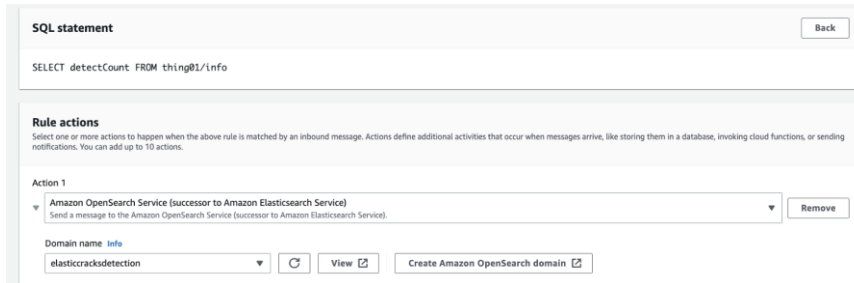


Fig. 10. An example of configuring data filtering rules in crack index

The AWS IoT rule is used to filter only the most necessary information to signal the largest amount of detected damage in the object.

In the previously created database selected from the proposed list as shown in Fig. 10. In the next step, the index name, identifier, and type are specified in the database records. Since the feature of Elasticsearch is the placement of data in indexes, the storage of data for various purposes is divided according to the researched materials. In my case, the index name is "crack_index" for concrete damage and "corrosion_index" for metal damage. Type "data". It should be taken into account that the identifier must contain a unique value, so the `newuid()` function can be used in order not to overwrite the existing value. Additionally, security roles are selected to access the endpoint.

The filtering condition is as shown in Fig. 10 is an SQL query where you can specify how to save all fields from the subscription and specific ones to prevent redundant information in the data store. The advantage of such filtering is the possibility of using the "WHERE" condition in the SQL query to specify a certain limit of the required values.



Fig. 11. Stored data on the number of detected points in Elasticsearch AWS according to the rule in concrete and metal

In Fig. 11 shows stored values in the database viewed through an EC2 virtual instance in the AWS cloud environment.

By the percentage ratio of the number of pixels and the total number of pixels of the camera, we will get information about the criticality of defects and highlight the deepest cracks or the greatest corrosion. The next steps for analyzing such information will be evaluation capabilities and data analysis by means of artificial intelligence.

When working with images and videos, the quality of the video plays an important role due to the detail of the detected damage to the objects. Therefore, for more accurate and better evaluations, it is worth installing the device on a more stable platform and making smooth movements.

Conclusion

In this work, a Raspberry Pi-based IoT system is built to detect concrete and metal defects using image filtering and object damage calculation. The integration is described and implemented for a Raspberry Pi microcomputer and software with cloud storage Elasticsearch and file storage Amazon Web Service (AWS) S3 cloud storage for concrete and metal damage data for further analysis.

The described system can be used for hard-to-reach places, taking into account autonomy and visual assessment of the criticality of the state of objects. System flexibility and extensibility is possible for more damage and object condition monitoring by adding and improving Raspberry Pi-side filtering. The applied keypoint filtering for cracks in concrete and localization by color of corrosion from the video frames allows to quantify the damage changes with the help of the described set of steps.

Also, the work describes the specific steps for configuring the integration parameters of the Amazon Web Service cloud environment and the Raspberry PI microcomputer. Filtering of data from the MQTT protocol subscript to the Elasticsearch data store using the Amazon Web Service IoT rule is used.

References

- [1] *Hosny, Khalid & Magdi, Amal & Salah, Ahmad & Elkomy, Osama & Lashin, Nabil.* Internet of things applications using Raspberry-Pi: a survey. *International Journal of Electrical and Computer Engineering.* 13. 2023. P. 902-910, doi: <https://doi.org/10.11591/ijece.v13i1.pp902-910>
- [2] *Woubishet Zewdu Taffese, Ethiopia Nigussie, Jouni Isoaho,* Internet of Things based Durability Monitoring and Assessment of Reinforced Concrete Structures / *Procedia Computer Science // Vol. 155. 2019. P. 672-679, doi: https://doi.org/10.1016/j.procs.2019.08.096*
- [3] *Schiegg, Y.* A new IoT Corrosion Monitoring-System for concrete structures. *International Symposium on Non-Destructive Testing in Civil Engineering (NDT-CE 2022), 16-18 August 2022, Zurich, Switzerland // e-Journal of Nondestructive Testing. Vol. 27(9), doi: https://doi.org/10.58286/27339*
- [4] *Komary, M., Komarizadehasl, S., Tošić, N., Segura, I., Lozano-Galant, J.A., Turmo, J.* Low-Cost Technologies Used in Corrosion Monitoring. *Sensors 2023, 23, 1309. doi: https://doi.org/10.3390/s23031309*

- [5] *B. N. Rao and R. Sudheer*. Surveillance Camera using IoT and Raspberry Pi, 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2020, P. 1172-1176, doi: <https://doi.org/10.1109/ICIRCA48905.2020.9182983>
- [6] *Mysiuk R., Yuzevych V.* Recover Data about Detected Defects of Underground Metal Elements of Constructions in Amazon Elasticsearch Service // Path of Science, 2023, Vol. 9, No 1. P. 1011-1019, doi: <https://doi.org/10.22178/pos.89-9>
- [7] *R. Mysiuk, I. Mysiuk, V. Yuzevych, G. Pawlowski* Determining the Place of Depressurization of Underground Pipelines (Gas Pipelines): New Solutions in Industry based on Thermal Image Analysis Using Computer Vision // Path of Science. 2022. Vol. 8, No 10, P. 1001-1010, doi: <https://doi.org/10.22178/pos.86-9>
- [8] *Al-Ghaili, A.M. Kasim, H. Hassan, Z. Al-Hada, N.M. Othman, M. Kasmani, R.M. Shayea, I. A.* Review: Image Processing Techniques' Roles towards Energy-Efficient and Secure IoT. Appl. Sci. 2023, 13, 2098, doi: <https://doi.org/10.3390/app13042098>
- [9] *R. Mysiuk, I. Mysiuk, G. Pawlowski, V. Yuzevych, M. Yasinskyi and Y. Tyrkalo.* Video-based Concrete Road Damage Assessment Using JetRacer Kit / 2023 17th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Jaroslaw, Poland, 2023, P. 1-4, doi: <https://doi.org/10.1109/CADSM58174.2023.10076528>
- [10] Connect a Raspberry Pi or other device - AWS IoT Core [Online]. URL: <https://docs.aws.amazon.com/iot/latest/developerguide/connecting-to-existing-device.html>
- [11] Streaming Sensor Data (Raspberry Pi) to AWS IoT [Online]. URL: <https://techblog.calvinboey.com/raspberrypi-aws-iot-python/>.
- [12] How to Connect Your Raspberry Pi to AWS IoT? [Online]. URL: <https://witekio.com/de/blog-de/connect-raspberry-pi-to-aws-iot/>

ІОТ РІШЕННЯ ДЛЯ ВИЯВЛЕННЯ ДЕФЕКТІВ В ІНФРАСТРУКТУРНИХ ОБ'ЄКТАХ НА ОСНОВІ RASPBERRY PI

Роман Мисюк¹, Володимир Юзевич^{1,2}

¹ кафедра системного проектування,
Львівський національний університет імені Івана Франка,
вул. Драгоманова, 50, м. Львів, 79005, Україна
mysiukr@urk.net

² Фізико-механічний інститут ім. Г. Карпенка НАН України,
вул. Наукова, 5, м. Львів, 79601, Україна.

Моніторинг поточного стану доріг, матеріалів та комунікаційних мереж часто може бути ускладнений в доступності до пошкодженого місця та важкий для візуальної оцінки критичності стану. Поточний стан таких об'єктів інфраструктури дають змогу оцінити зібрані дані з давачів, а проміжний аналіз автоматизований в мікрокомп'ютерах чи мікропроцесорах. 5-піксельна OV5647 камера використовується як пристрій для візуального зчитування даних з об'єкта, а мікрокомп'ютер Raspberry Pi 4 як пристрій для аналізу відео та надсилання до хмарного середовища Amazon Web Service (AWS).

Для візуальної оцінки стану використано відео з застосуванням фільтрів для виокремлення ділянок пошкодження об'єкта. Перший можливий тип дефекту об'єкта є корозія, яка виявляється методом накладання колірних фільтру, другий - тріщини заповнені ключовими точками алгоритмами Oriented FAST та Rotated BRIEF у виділених краях фільтром Canny.

Передача зібраних кількості виявлених пікселів фільтрованого кадру з Raspberry Pi здійснюється за допомогою протоколів MQTT в розроблений системі. Дані про час, кількість виявлених пікселів та матеріал для якого проводиться дослідження формуються у повідомлення JSON формату. AWS IoT правило постійно читає повідомлення з підписаного набору повідомлень в AWS IoT для надсилання в нереляційну базу даних Elasticsearch потрібних даних для подальшого аналізу. Створені два окремі індекси у базі даних Elasticsearch для зберігання даних про кількість виявлених пікселів корозії та кількість ключових точок в тріщинах для AWS IoT правил. У AWS S3 bucket зберігаються файли відео, які видаляються з Raspberry Pi після надсилання для збереження вільного місця у карті пам'яті. Проаналізовано можливості інтеграції Raspberry Pi 4 з хмарних середовищем AWS для зібраних даних з відео про кількість корозії та тріщини в об'єктах. В результаті реалізована система може бути використана для опрацювання дефектів об'єктів з відео та зберігатися у сховищах даних для подальшого аналізу та візуалізації.

Ключові слова: Інтернет речей, хмарні обчислення, дефекти, Raspberry Pi, корозія, тріщина.

Стаття надійшла до редакції 09.01.2023

Прийнята до друку 16.02.2023