Real-time image analysis in IoT-based home security system

A. Villarreal[a], M. Mehrubeoglu*[a], L. Davila[a], L. McLauchlan[b]

[a]Department of Engineering, Texas A&M Univ.-Corpus Christi, Corpus Christi, Texas, USA
[b]Department of Electrical Engineering and Computer Science, Texas A&M University-Kingsville, Kingsville, Texas, USA

ABSTRACT

Internet of Things (IoT) uses cloud-enabled data sharing to connect physical objects to sensors, processing software, and other technologies via the Internet. IoT allows a vast network of communication amongst these physical objects and their corresponding data. This study investigates the use of an IoT development board for real-time sensor data communication and processing, including images from a camera, as part of a custom-made home security system intended for the elderly for easy access.

Keywords: Internet of things, IoT, Arduino, BeagleBone, home security

1. INTRODUCTION

Internet of things (IoT) enabled devices continue to be developed for many commercial and research applications. On the development side, Molins-Benlliure *et al.* described a small-sized low-temperature cofired ceramic (LTCC) chip antenna for use in size-limited IoT devices [1] that can function without the need for placement clearance when connected. Chen *et al.* discussed a chip-free RFID strain sensor to be used in an IoT sensing system that utilizes an embedded tag and associated sensing capability to measure magnitude and direction of strain of the metal sample under investigation [2]. Many example IoT applications exist in the literature [3]-[9]. Example IoT applications include security systems, monitoring systems, unmanned aerial vehicles (UAVs), and remote sensing, to name a few. Ghamari *et al.* reviewed the UAV related applications of IoT including aspects of remote sensing, communication, and security [10], identifying the many issues still remaining in large-scale IoT applications, one of which is cybersecurity against cyberattacks [11]. Xie *et al.* investigated use of binocular vision sensing in IoT for detecting road conditions real-time [12]. The group trained their system to detect road debris, manhole covers, speed bumps and road surface structure for autonomous driving implementation. Falcon *et al.* utilized a drone-based surveying system through IoT connectivity to detect cracks on concrete structures [13]. Sangeethalakshmi *et al.* developed a health monitoring system [6] and Moore *et al.* developed and tested an IoT-based a knee sleeve for potential rehabilitation applications [7].

Other IoT applications that are closer to the work presented here involve those related to surveillance and security. Sattaru *et al.* describe a small-scale camera-based home surveillance system connected to Arduino UNO, and NodeMCU and ESP8266 for connecting to the Internet [14]. The system achieved motion detection through simple circuitry. Similarly, Rao *et al.* developed a small-scale camera for surveillance using Raspberry Pi and IoT [15]. The same lead author looked at expanding the same system by installing the camera on a mobile robot for with Raspberry Pi and IoT connectivity [16] to detect the presence of individuals. Ghosh *et al.* developed and tested a personal security device that would send information such as location via SMS [3]. Peng *et al.* tested multi-agents such as unmanned ground vehicle (UGV), and unmanned aerial vehicle (UAV) to monitor and chase intruders in a prototype intelligent home security system [4]. The system monitored the state of an entry point for changes and then the agents performed tasks such as recording video once an intruder was detected at the entry point. Another example of a home security system is addressed by Wang, Yuan and He who developed a system that monitored smoke, temperature and gas sensors [5].

*ruby.mehrubeoglu@tamucc.edu; phone 1 361 825-3378;

As IoT devices are becoming more common, skills to design and test such IoT enabled systems and applications are also becoming necessary. In [8]-[9] the authors discussed exercises to learn more about IoT to be able to utilize IoT in design projects as part of undergraduate engineering and computer science programs. With decreased computing costs and processor/microcontroller size, IoT will continue to grow as a research field. Here, we present real-time image capture, tracking and classification in an IoT-based home security system. The rest of the paper includes methodology and results in Section 2. Conclusions and future work are presented in Section 3.

2. METHODOLOGY AND RESULTS

This study investigates the use of an IoT development board (IoT Kits, Keysight, Santa Rosa, California) for real-time sensor data communication and processing, including images from a camera (Pixi2, Charmed Labs, Austin, Texas), as part of a custom-made home security system intended for the elderly for easy access and useability. Figure 1 shows a basic block representation of an IoT system. IoT-based application blocks could include applications such as monitoring systems, controlling unmanned aerial vehicles (UAVs), surveillance, remote sensing, and any other implemented by the user.

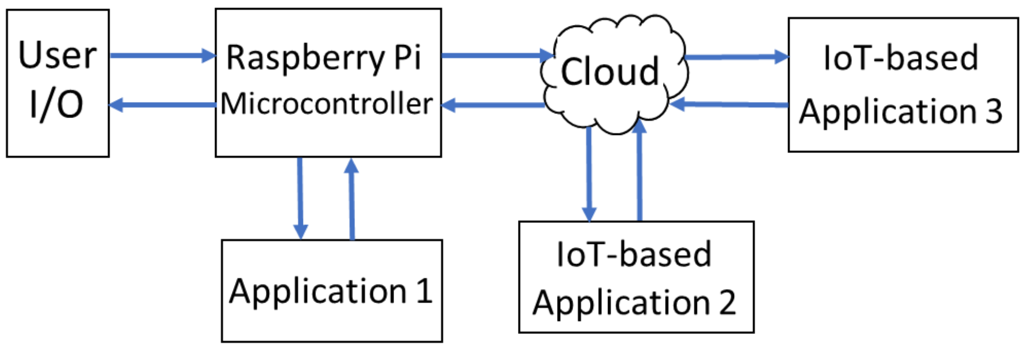

Figure 1. Representation of an IoT System [6]

For this research, the application is a surveillance system designed as part of a custom-made home security system intended for easy access. A high-level block diagram is shown in Figure 2 as well as pictures for each of the main components retrieved from [17]-[20].

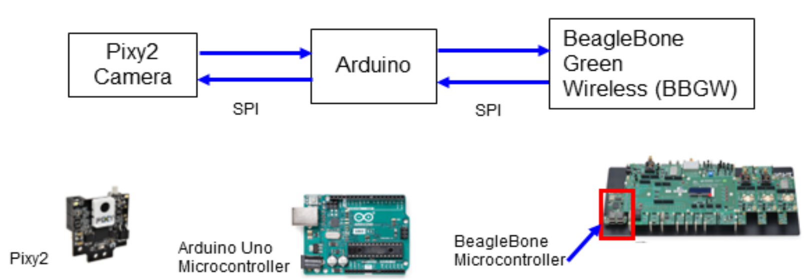

Figure 2. Connections for System – includes (left) Pixy2 camera [17], (center) Arduino Uno board [18], and (right) BeagleBone Green Wireless board [19] as part of the Keysight U3810A IoT Kit [20]

Since the IoT kits utilized had a BeagleBone Green Wireless (BBGW) microcontroller, which did not directly work with the Pixy2 camera, the Pixy2 camera needed another device to bridge the connection between the two devices. In this case an Arduino microcontroller was utilized. As seen in Figure 2, connections between the Pixy2 camera and Arduino microcontroller were made over one of the Serial Peripheral Interfaces (SPI) and a second SPI was used to connect the Arduino microcontroller to the BeagleBone Green Wireless microcontroller. Figure 3 shows the overall high-level description of the process for capturing frames.

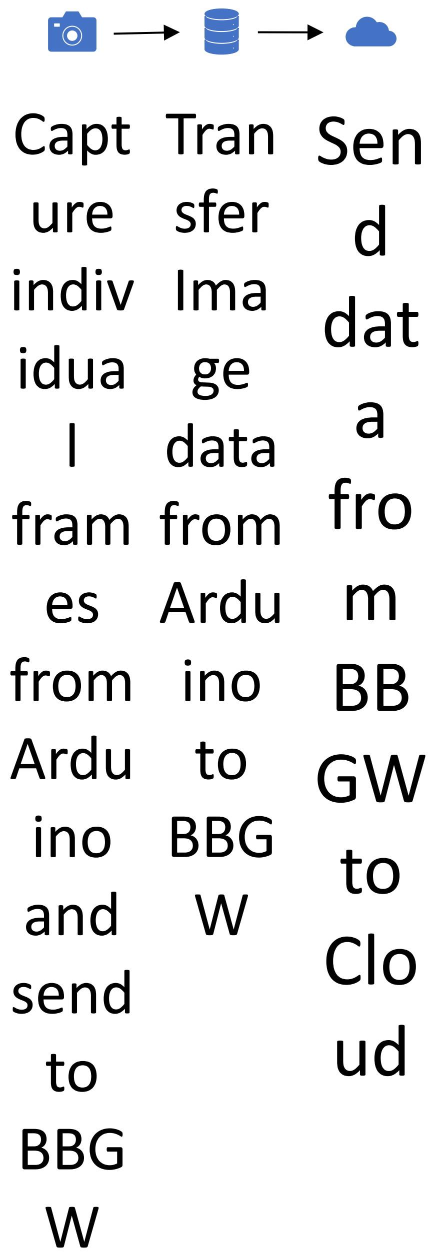

Figure 3. Process for Capturing Frames (BBGW: BeagleBone Green Wireless)

The Pixy2 camera has built-in processing power which off-loads some of the processing requirements for the video frames allowing the system to utilize a microcontroller such as an Arduino and a BeagleBone. A simplified block diagram model for this IoT application is displayed in Figure 4.

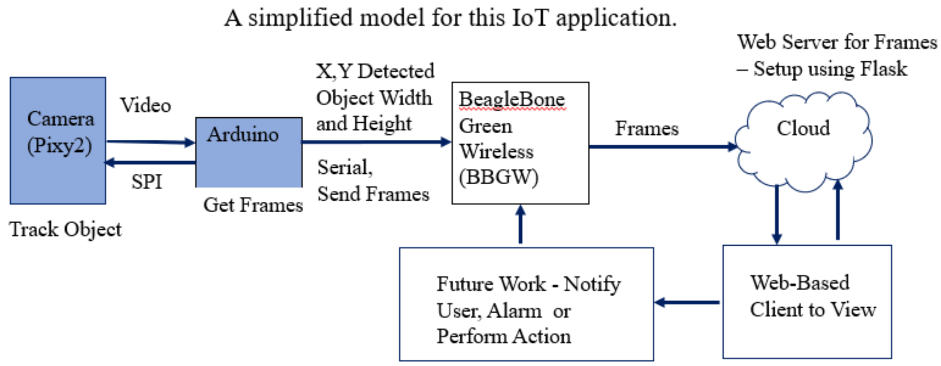

Figure 4. Simplified block diagram model for this IoT application – Blocks for capturing image frames are highlighted.

Block detection code in the Arduino IDE is shown below in Figure 5. The Pixy2 camera has libraries that can be utilized in the Arduino integrated development environment (IDE) to facilitate coding and control of the Pixy2 camera. Pixy2 and SPI library were included in the code. In this case, the blocks of interest were detected for an object to be tracked in the video and then a color connected components algorithm, a form of blob detection algorithm, was implemented to track an object. The *x,y* coordinates, height and width for the detected blocks were recorded. The SPI allows the Arduino and the Pixy2 to communicate over a serial communication port. This is highlighted in Figure 6.

```
#include <Pixy2.h>
#include <SPI.h>

Pixy2 pixy;

void setup() {
  Serial.begin(9600);
  pixy.init();
}

void loop() {
  static uint32_t last_ms = 0;
  uint32_t now_ms = millis();

  if (now_ms - last_ms > 100) {
    last_ms = now_ms;

    int blocks = pixy.ccc.getBlocks();
    if (blocks > 0) {
      for (int i = 0; i < blocks; i++) {
        int x = pixy.ccc.blocks[i].m_x;
        int y = pixy.ccc.blocks[i].m_y;
        int width = pixy.ccc.blocks[i].m_width;
        int height = pixy.ccc.blocks[i].m_height;
        // Send the block data over serial
        Serial.print(x);
        Serial.print(",");
        Serial.print(y);
        Serial.print(",");
        Serial.print(width);
        Serial.print(",");
        Serial.println(height);
      }
    }
  }
}
```

Figure 5. Block detection code in Arduino IDE

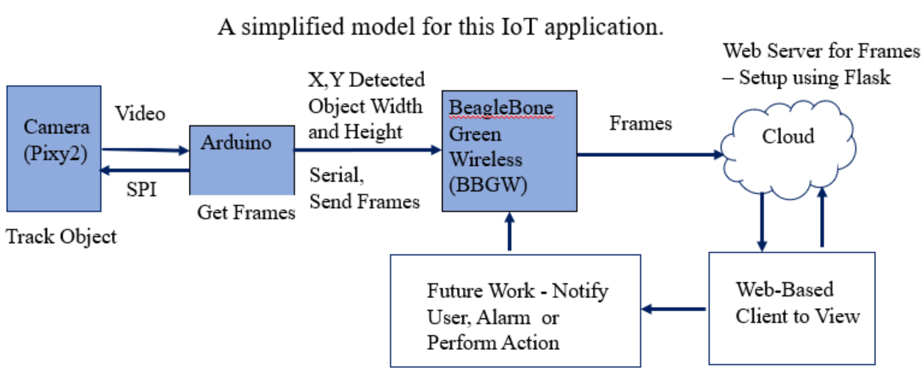

Figure 6. Simplified block diagrams model for this IoT application – Detected blocks and image data sent to BeagleBone

A Python script used to read the video frames from the serial port and display them using the 'opencv-python' library is shown below in Figure 7. 'serial' library has been used to read the binary data from the serial port and the cv2.imshow()' function is used to display the frames

```
import serial
import cv2
# Open the serial port
ser = serial.Serial('/dev/ttyACM0',
  115200)
# Create a window to display the frames
cv2.namedWindow 'Video',
  cv2.WINDOW_NORMAL)
# Read and display the video frames
while True:
    # Read the binary data from the
  serial port
    data = ser.read(1024
        # Decode the compressed video
  frame (e.g., using JPEG)
    frame = decodeJPEG(data)
      # Display the frame in a window
    cv2.imshow('Video', frame)
# Exit if the 'q' key is pressed
    if cv2.waitKey(1) & 0xFF ==
  ord('q')
        break
# Release the resources
cv2.destroyAllWindows()
ser.close()
```

Figure 7. Python script used to read the video frames from the serial port and display them

The next step in the process is to send image data to the Cloud. For this application, Flask was utilized to setup a web server as highlighted in Figure 8. Coding for setting up this communication is shown in Figure 9.

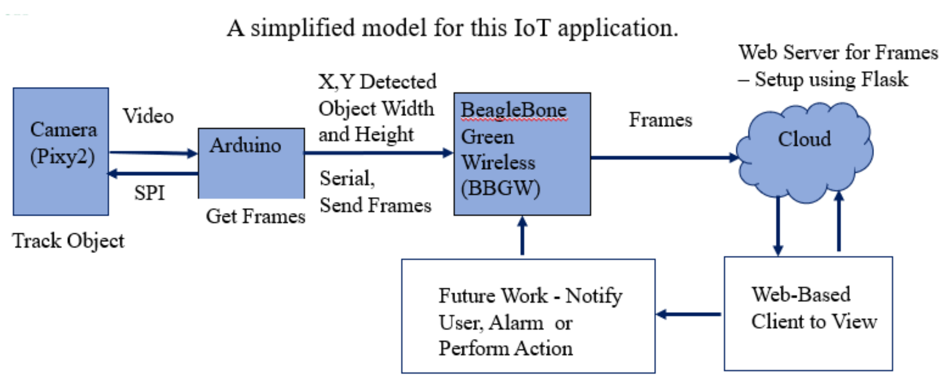

Figure 8. Simplified block diagrams model for this IoT application – Using Flask as webserver to store image data.

In this research, one can utilize a web server on the BeagleBone Green Wireless to stream the video frames over a network connection. Then one can use Flask [20] to create a web server that receives the video frames from the serial port, encodes these frames as a video stream using a format like MJPEG or H.264, and serves the stream to a web client.

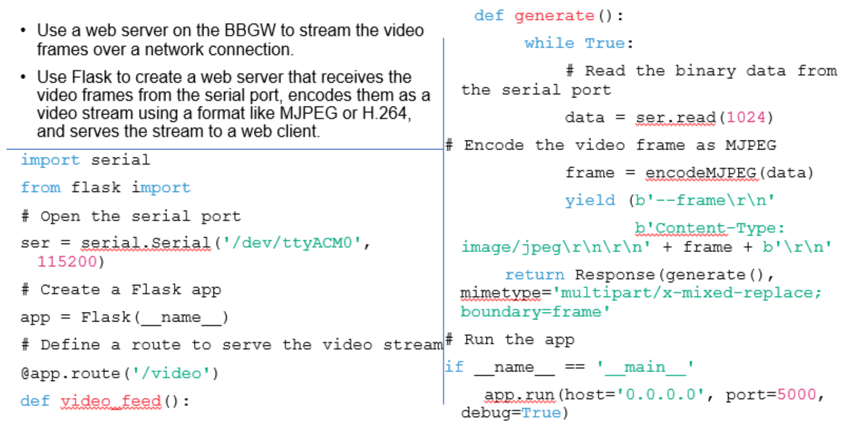

Figure 9. Coding to stream the video

A web-based client viewer for the security system would then allow the end-user to view the stream. In addition, the web-based client viewer can also be used to check other sensor data as well as set any alarms, as needed, as shown in Figure 10.

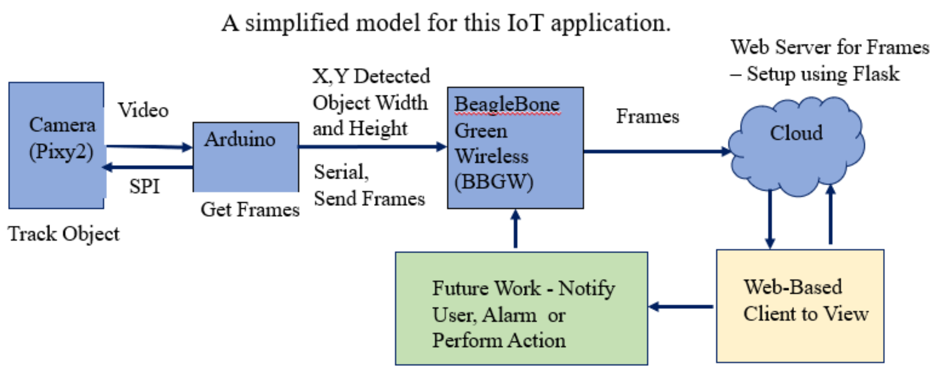

Figure 10. Simplified block diagram model for the presented IoT application – Web-based client to view images and future work to notify the user, set alarms and perform an action – highlighted blocks in tan (bottom right) and green (bottom left).

At this point the next step is to serve the video stream to a web client. In this case since this task was not completely finished and tested, a monitor program was run on a PC to display the scene Pixy2 camera was capturing, as shown in Figure 11. Example frames showing a blue cup that has been detected and tracked are displayed in Figure 12. Frames shown are 16.67 ms apart. This is possible due to the camera's on-board processing capability. Overall processing time to run the algorithm per frame and track objects across frames is feasible to run in real-time. More testing is needed to see what latency is encountered to stream the video to a web-based client for viewing the video stream.

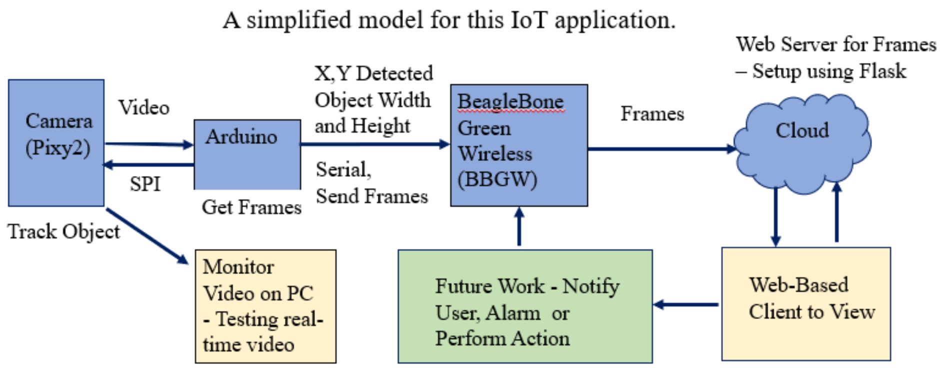

Figure 11. Simplified block diagram model for this IoT application – A PC has been added to view what the Pixy2 is capturing – highlighted block in tan (bottom left).

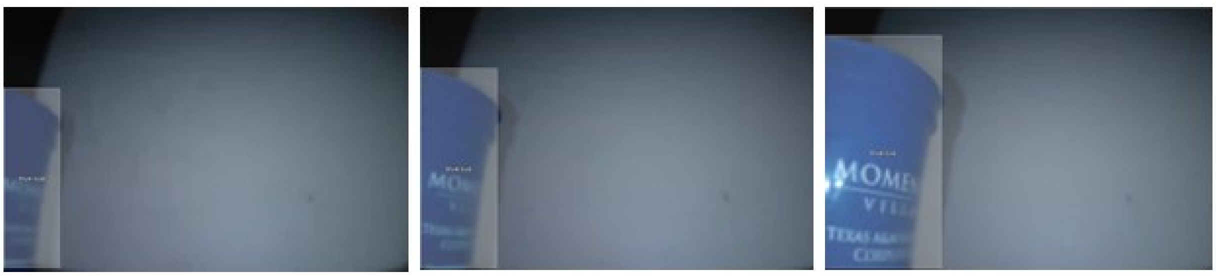

Figure 12. Example consecutive frames showing a cup that has been detected and tracked.

3. CONCLUSIONS AND FUTURE WORK

A BeagleBone Green Wireless microcontroller has been connected to a Pixy2 camera through an Arduino microcontroller. Detected objects were tracked and one can then post those frames to a webserver using Flask. The tested portions of the system will run in real-time. More testing is needed to see what latency is encountered to stream the video to a web-based client for viewing the video stream. More sensors such as infrared or acoustic sensors will also need to be added to the system to make this a more complete security system. Future work will entail adding notification via text, email or some other predetermined method when an intruder is detected. In addition, more data statistics, sensor data and added functionality will improve the overall system prototype.

A BeagleBone Green Wireless microcontroller has been connected to a Pixy2 camera through an Arduino microcontroller. Detected objects were tracked and one can then post those frames to a webserver using Flask. The tested portions of the system will run in real-time. More testing is needed to see what latency is encountered to stream the video to a web-based client for viewing the video stream. More sensors such as infrared or acoustic sensors will also need to be added to the system to make this a more complete security system. Future work will entail adding notification via text, email or some other predetermined method when an intruder is detected. In addition, more data statistics, sensor data and added functionality will improve the overall system prototype.

ACKNOWLEDGMENT

This research is supported in part by McNair Scholars program at Texas A&M University - Corpus Christi. The IoT Kits used in this work is funded through a grant from the National Science Foundation (#2044255).

REFERENCES

[1] J. Molins-Benlliure, E. Antonino-Daviul, M. Cabedo-Fabr and M. Ferrando-BatallerLife Fellow, "On-ground Small

LTCC Chip Antenna and its Placement on IoT Devices," in *IEEE Antennas and Wireless Propagation Letters*, doi: 10.1109/LAWP.2023.3255163.
[2] L. Chen, L. Liu, L. Kang, Z. Wan, G. Wan and L. Xie, "A Multibranch U-Shaped Tunable Encoding Chipless RFID Strain Sensor for IoT Sensing System," in IEEE Internet of Things Journal, vol. 10, no. 6, pp. 5304-5320, 15 March15, 2023, doi: 10.1109/JIOT.2022.3221938.
[3] Ghosh, P., Bhuiyan, T., Nibir, M., *et al*., "Smart Security Device for Women Based on IoT Using Raspberry Pi," *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, Dhaka, Bangladesh, 2021, pp. 57-60.
[4] Peng, Z., Kato,T., Takahashi, H., and Kinoshita, T., "Intelligent home security system using agent-based IoT devices," *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, Osaka, Japan, 2015, pp. 313-314.
[5] Wang, A., Yuan, Z. and He, B., "Design and Realization of Smart Home Security System Based on AWS," *2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, Xi'an, China, 2020, pp. 291-295.
[6] Sangeethalakshmi, K., Preethi Angel, S., Preethi, U., Pavithra, S. and Shanmuga, V., "Patient health monitoring system using IoT," *Materials Today: Proceedings,* Available online June 24, 2021.
[7] Moore, A., Mehrubeoglu, M., Mooney, A. and McLauchlan, L., "Application of IoT-based Sensing and Signal Processing for Rehabilitation," *Proc. of SPIE*, April 2022.
[8] McLauchlan, L., Hicks, D.. Mehrubeoglu, M. and Zimmer, B., "Work-in-Progress: Internet of Things Enabling Remote Student Learning," *2022 ASEE Annual Conference and Exposition,* Minneapolis, MN, USA June 26-29, 2022.
[9] Mehrubeoglu, M., McLauchlan, L., Hicks, D. and Zimmer, B., "Work-in-Progress: Developing an IoT-based Engaged Student Learning Environment and Tools for Engineering Programs," *2022 ASEE Annual Conference and Exposition,* Minneapolis, MN, USA June 26-29, 2022.
[10] M. Ghamari, P. Rangel, M. Mehrubeoglu, G. S. Tewolde and R. S. Sherratt, "Unmanned Aerial Vehicle Communications for Civil Applications: A Review," *IEEE Access*, vol. 10, pp. 102492-102531, 2022, doi: 10.1109/ACCESS.2022.3208571.
[11] Zhang, J., Chen, H., Gong, L., Cao, J. and Gu, Z., "The Current Research of IoT Security," *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, Hangzhou, China, 2019, pp. 346-353.
[12] Q. Xie, X. Hu, L. Ren, L. Qi and Z. Sun, "A Binocular Vision Application in IoT: Realtime Trustworthy Road Condition Detection System in Passable Area," in IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 973-983, Jan. 2023, doi: 10.1109/TII.2022.3145858.
[13] Jose Falcon, Mehrube Mehrubeoglu, and Pablo Rangel "Real-time image processing in a multi-UAV system for structural surveillance through IoT platform", Proc. SPIE 12102, Real-Time Image Processing and Deep Learning 2022, 121020F (27 May 2022); https://doi.org/10.1117/12.2622362
[14] P. K. Sattaru, K. V. Burugula, R. Channagiri and S. Kavitha, "Smart Home Security System using IoT and ESP8266," 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2023, pp. 469-474, doi: 10.1109/ICSSIT55814.2023.10061059.
[15] 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, pp. 1172-1176, doi: 10.1109/ICIRCA48905.2020.9182983.
[16] B. N. Rao, R. Sudheer, M. A. Sadhanala, V. Tibirisettti and S. Muggulla, "Movable Surveillance Camera using IoT and Raspberry Pi," 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2020, pp. 1-6, doi: 10.1109/ICCCNT49239.2020.9225491.
[17] BeagleBoard.org, BeagleBone®. Available: https://beagleboard.org/bone (Accessed: 2 May 2023).
[18] U3810A IoT Kit, Keysight. Available: https://www.keysight.com/us/en/product/U3813A/iot-system-designvalidation-training-kit-lab-sheets.html (Accessed: 2 May 2023).
[19] Arduino Uno. Available: https://store.arduino.cc/products/arduino-uno-rev3 (Accessed: 2 May 2023).
[20] Flask. Available https://flask.palletsprojects.com/en/2.3.x/ (Accessed: 2 May 2023).