Research Article Journal of Sustainable Engineering and Built Environment
Faculty of Engineering, Technology, and Built Environment, UCSI University

Developing Affordable IoT-based Body Temperature Screening

Rakan Emad Tabiah[1], M. Iwan Solihin[1,*], Affiani Machmudah[2]
[1]Faculty of Engineering, UCSI University, Kuala Lumpur, Malaysia
[2]Faculty of Advanced Technology and Multidiscipline, Universitas Airlangga, Indonesia
*Corresponding author: mahmudis@ucsiuniversity.edu.my

Keywords: thermal Imaging; body temperature screening, IoT

Abstract

Thermal infrared thermometers are nowadays being used everywhere to test the body temperature in areas with large numbers of people. This is because allegedly one very important symptom of COVID-19 is high body temperature. But this practice is tedious and time-consuming. An alternative to thermal body screening is using thermal camera-based measurement. However, the price is normally high. This research initializes the attempt to develop an affordable device capable of automatically detecting and monitoring the elevated body temperatures from the thermal image with fewer human interactions. The device uses a low-cost thermal sensor that can be incorporated into an IoT-based Mass Fever Screening (MFS) system. The system may be used for tracking the screening process to record the real-time data and take the necessary action when an infected person is suspected by triggering action such as a panic alarm or sending a notification or an email.

Introduction

In today's society of global mass transportation travel, contagious disease outbreaks will cross over national and international borders within hours. “The SARS”, “Bird Flu”, and “Swine Flu”, “H1N1”, “Ebola virus” and now “corona virus COVID-19)”, have infected thousands of people and businesses in all aspects of life with major change. Corona virus is a kind of virus that also spreads disease to animals [1] and individuals [1,2]. In January 2020, The World Health Organization (WHO) announced the new virus called by 2019-nCoV [3]. In most COVID-19 patients, the common symptoms such as fever, dry cough, and tiredness are easily known [4-6].

It has been widely implemented for the detection of suspected virus carriers among crowds by measuring the temperature in public places using an infrared thermometer. The method of monitoring the body temperature of people using an infrared thermometer gun has drawbacks such as only limited covered area and is time-consuming. The probability of spreading the virus will emerge from the health officer's system which has scanned through a lot of people queuing when one of them is likely to infect people around. An alternative technology with less human involvement is required to avoid this vulnerability to reduce the risk of virus spread [7]. Hence, there is a need to build an inexpensive device that uses thermal cameras and custom software with way less human interaction to detect elevated skins temperatures (EST) [8].

Internet of Things (IoT)-connected devices interact with people and other items and provide cloud storage and cloud computing services with sensor data where the data is processed and analysed to gain significant in- sights [9-11]. To enhance COVID-19's symptoms detection and diagnosis method, this study initializes an attempt to develop a system that has the capability of automatically detecting the high body temperatures from the thermal image rapidly with fewer human interactions. This is done by using an effective thermal sensor solution that can be incorporated into an IoT-based Mass Fever Screening ‘MFS’ system. In the future, the proposed device may be useful towards the development of a reliable and affordable, and effective fever screening system. This will be crucial to assist in the pandemic spread management such as COVID-19.

Research Methods

The flowchart as shown in Figure 1 explains the methodology of the system proposed in this research. The system has a thermal camera technology, and it is used in the proposed design and integrated with an IoT technology for tracking of the screening process to record the real-time data temperature obtained from the sensor solution, and then take the necessary action/trigger when an infected person is suspected, by triggering action such as a panic alarm or sending a notification or an email. Furthermore, the proposed device is fitted with facial recognition software, and it will automatically take the temperatures of the visitors.

The design proposed is intended to be integrated with Raspberry Pi, a Pi camera, and a thermal sensor, and Firebase IoT service will be used for communication with the real-time data recording. The device will inform the officer of the suspicious cases by giving a warning along with a taken image face along with body temperature. Inside the device, even the thermal camera needs to be configured. The “Haar feature-based cascade classifiers” for face detection method is used to perform face detection, and the method uses the images input as an integral image representation of the image. Depending on the features submitted, “Haar Cascade classifiers” will determine the true face of potential candidates. This form of face detection will produce data concerning all the faces present in a picture frame input.

The list of the components is as follows:

- Raspberry PI 4 model B – 4GB.
- Adafruit 8x8 Grid-EYE® Infrared Array Sensor AMG8833
- Raspberry Pi 8MP camera module V2.

- WS2812B NeoPixel LED Stick - 8 LED.
- Official RPi 15W (5V/3A) Power Supply USB type C.
- 16 or 32GB Micro SD Card with NOOBS for RPI pre-installed.
- Raspberry Pi 4 Micro-HDMI to Standard HDMI cable.
- Female to Female Jumper Wires.
- 3D-printed plastic case.

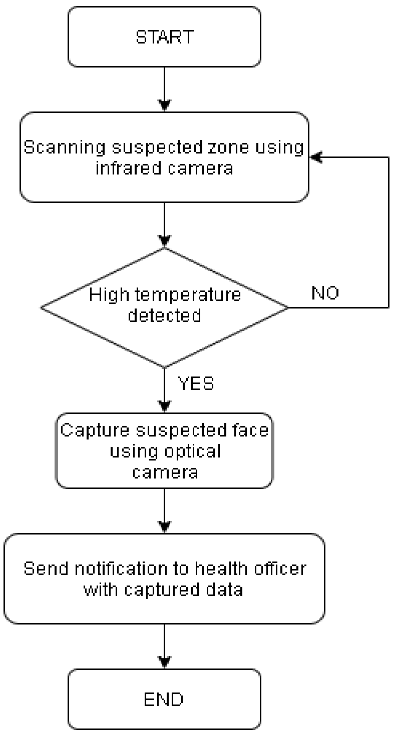

START
Scanning suspected zone using infrared camera
High temperature detected
NO
YES
Capture suspected face using optical camera
Send notification to health officer with captured data
END

Figure 1. Flowchart of the proposed system

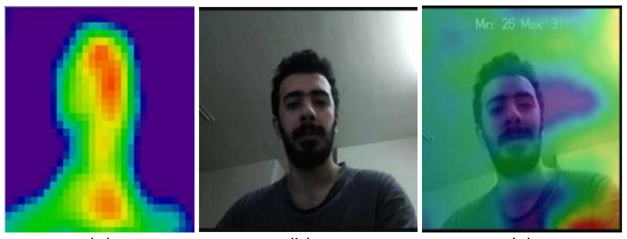
Initially, the Raspberry Pi was operated by connecting an external monitor, a keyboard, and a mouse for exploring the environment in which all the software development is going to take place. The code is written using Thonny IDE software, a Python Integrated Development Environment. The main procedure behind the project was to first, connect the pi camera and install all the essential packages and libraries. The next step was to connect the thermal camera sensor to the correct GIPO pins on the raspberry pi and install the "Adafruit" library for utilizing the sensor. With all the components set and ready, the following milestone was to produce the heat image. "Adafruit library" contained an example script for reading the sensor and mapping the temperatures to colours, However, the moving images it created could not be implemented. Therefore, the code was rewritten to a format supporting image processing, mainly for fusing two frames together. When a final image could be obtained, it was required to be sent to a web server. To do so a script from GitHub written by Antonio ALVES [12] was modified to suit this project's purpose.

The first step in the process was to gather data from the thermal camera. Adafruit library was utilised, which allows for easy reading of the sensor using the command "readPixels()", which generates an array containing temperatures in degree Celsius measured from the sensor's separate elements. Concerning the pi camera, the function command "picamera.capture()" generates an image with a specified output file format. In order to suit rapid processing and streaming purposes, a lower resolution was set to 500 x 500 pixels.

To visualise the thermal data in an informative and intuitive manner, the temperature values are mapped onto a colour gradient, ranging from blue to red with all the other colours in between. Every time the sensor is being read, the lowest temperature is mapped to 0 (blue) and the highest temperature to 1023 (red). All the other temperatures in between are assigned correlative values within the interval.

As previously stated, the resolution of the thermal sensor is fairly low, 8 x 8 pixels. Cubic interpolation made it possible to increase the resolution to 32 x 32, which results in a matrix 322 = 82 = 16 times larger. Interpolation works by constructing new data points between a set of known points. The accuracy may differ depending on which method is used. However, details occurring in the set where a picture is taken are not possibly to reproduce [13].

The next step is to resize the 32 x 32 image to a one with the size 500 x 500 in order to match the resolution of the picamera. Python Image Library (PIL) has an anti-aliasing filter when resizing images, meaning it will "smooth out" the edges between the pixels when enlarged up, which is shown to the left in Figure 2. The camera and thermal image are then blended into one final image with a pre-written function from PIL (Python Image Library), adding them together with 50% transparency on each. Worth to notice is when an image is fused from two sensors with a parallel distance between them, they will not completely overlap. Now the distance is fairly short and causes no problem as soon as objects are not right next to the sensors. The final result is shown to the right in Figure 2 where the image on the left is the one captured by the PI Camera with an overlaying text displaying the minimum and maximum temperature.



(a) (b) (c)
Figure 2. The heat image (a) is blended with the person image (b) from the PI camera to produce the final frame (c)

To obtain the above shown results, a local server is which is established in conjunction with running the scripts and is ready to receive the stream of the JPEG- images from the pi camera and the thermal camera sensor. To expose the local server to the Internet, making it accessible for other units, a free service called "Serveo" is used. "Serveo" has the feature of creating the user's own sub-domain (depending on availability) and will provide a static URL to reach the local server [14].

After running the compiled scripts using written in Python on the Raspberry Pi, the user must go to any browser and enter the URL associated with the localhost address (127.0.0.1) followed by the port predefined in the code. when accessed, a video stream from the Pi camera and the thermal sensor should appear on the monitor screen.

Firebase (Google's platform) is a "mobile-backend-as-a-service" that provides powerful features for building mobile apps. It has three core services: a "Realtime database", "user authentication" and "hosting". So, by using this service, it can be modified to allow it to send the temperature recordings from the thermal sensor to the Realtime database on Google's Firebase service. Three important things must be noted to enable such service and make it work as wanted, first after creating an account, the user should take note of the "API key" under the "Database secrets" tab and the "Project ID" all under "project settings". And lastly, is the database URL under the "Data" tab in "Realtime Database". Furthermore, an addressable RGB LED stick will be used which will be responsible to blink either red if a suspected person has a high body temperature detected, or green if the person has normal or below the threshold value of the body temperature.

For this project, another feature will be added which is to send alert emails only if the temperature exceeded the fixed threshold value, along with a screenshot of that detected person as an attachment in the email. The python software performs all of the operations required to read the sensor data, compare it to the predefined threshold, approve the email, and then send it with an image attached. All that is required is for the email to be arranged with a topic, body, as well as other attachments, which is accomplished using **"**Multipurpose Internet Main Extensions (MIME)".

In the final phase of the project, a 3D-designed case was manufactured to fit all the components explained above and to demonstrate how a complete platform could look like. It consists of three main parts shown in Figure 3 namely the base, the cover, and the front panel. The idea behind the design of the case is to achieve a robust arrangement. Mainly since one frame from the video stream is fused from the two sensors and therefore it is essential that they lay in the same plane without any angle between them. Another aspect of the design was to make room for ventilation slots to transfer away heat from the Raspberry Pi.

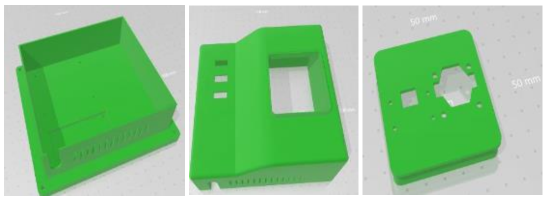


Figure 3. Prototype casing with 3-D printed material

Results and Discussion

The end product of this project is a device that will send a live stream to a website displaying a thermal view of the scenery as well as the minimum and maximum temperature. When the Raspberry Pi is powered on, it automatically executes the scripts that use the sensor modules, processes each frame, and creates a web server link with a live stream. It must be wired to a WIFI network to do so, and it can even run with a 4G USB modem plugged in. To navigate to the web page, the user must type "mjao.serveo.net" through the browser's address bar. Eventually, all readings will be uploaded and updated in real-time to Google's Firebase IoT program, which can always be available and saved at any time to make monitoring easier. Of course, if any high temperature is detected, a screenshot of that suspected person will be taken with details and sent via email to the person in charge of the monitoring station.

After running the compiled script using Thonny IDE for Python on the Raspberry Pi, the user should go to any browser and enter the URL associated with the localhost address (127.0.0.1) followed by the specified port in the code. In this case, the full URL for this project is following: (http://127.0.0.1:8080/cam.mjpg), when accessed, a video stream from the Pi camera and the thermal sensor should appear, and it should look something like in Figure 4. After setting up all the needed parameters mentioned above as well as for the sensor code, The output of the database created should look something like in b Figure 5.

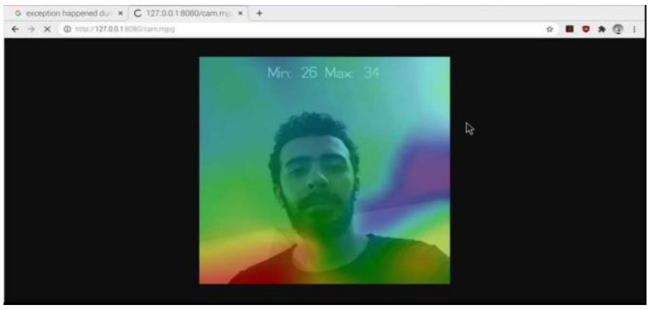
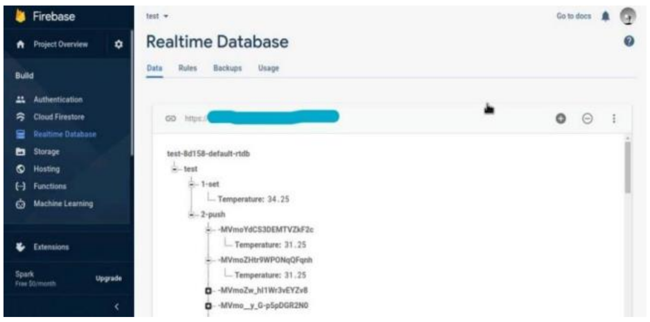


Figure 4. Thermal imaging streaming using a local web server.



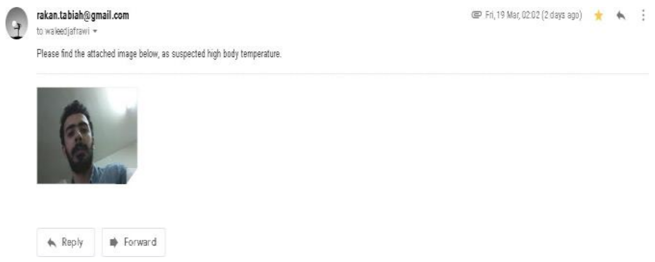
Realtime Database

https:/[redacted]

```
test-8d158-default-rtdb
 - test
   - 1-set
      └ Temperature: 34.25
   - 2-push
      - -MVmoYdCS3DEMTVZkF2c
         └ Temperature: 31.25
      - -MVmoZHtr9WPONqQFqnh
         └ Temperature: 31.25
      + -MVmoZw_hl1Wr3vEYZv8
      + -MVmo__y_G-p5pDGR2N0
```

Figure 5. Firebase real-time data sample

If the temperature of the object exceeds the threshold temperature, the python program can take an image from the camera, save it on the computer, then also send it via email. Figure 6 shows an example of an email sent from the Raspberry Pi.



rakan.tabiah@gmail.com
to waleedjafrawi

Fri, 19 Mar, 02:02 (2 days ago)

Please find the attached image below, as suspected high body temperature.

Figure 6. Sample of an email sent as an alert

Overall, the project initialization is achieved. The thermal camera functions as expected. It registers heat radiation from approximately 5 to 6 meters. The final cost for this prototype is estimated to be in the range of 600 to 700 MYR. If one wants to improve the platform so it could be used in a real scenario, more suitable components such as a different power supply setup, a thermal camera with higher performance and a GPS module will increase the price bus also increase the efficiency of the project. This low-cost prototype shows that if developed further, it could very well assist in the current pandemic situation and help can be distributed in every place.

One area of improvement that does not necessarily conflict with the application of such a device, is the rate at which the images get updated, i.e., the frame rate. The main issue that was faced was the technique behind fusing the camera image with the heat image and some overlaying text relied on working with single images at a time. To do so, a capture function from the 'picamera' library was used that had a relatively slow encoder which was the main factor that limited the frame rate. In a potential next version, deeper research can be done on how to capture continuous images and match them with the frames generated by thermal camera sensor.

The weight of the entire platform as well as the size of the case is possible to reduce if using a lithium-ion battery instead of the current power bank. Such a battery cannot be connected directly to the Raspberry Pi GIPO pins since they are lacking a built-in voltage regulator. Thus, another module controlling a steady-state power supply would be necessary which also will ensure that no power surge occurs that ultimately destroys the computer. Furthermore, those batteries can be dangerous if not cautiously handled.

Conclusion

While the Panasonic's thermal camera AMG8833 has a field of view considered sufficient, its resolution is not as high as high-resolution thermography (i.e., thermal images rich in detail), which is necessary for long-range thermal detection. FLIR offers the relatively low-cost radiometric LWIR camera FLIR Lepton 3.5®. Its field of view with 57 degrees is slightly narrower than that of the AMG8833, which is compensated by its superior resolution of 160 x 120, making it 300 times finer than the AMG8833. Considering the price, Panasonic's sensor became more desirable. "Sparkfun" offers a complete kit with a breakout circuit board for roughly MYR 200 while FLIR offers its LEPTON 3.5 for around MYR 800 to MYR 1000. However, this low-cost prototype shows that if developed further, it could achieve the level to meet expected results and objectives.

References

[1]. C. Xie, L. Jiang, G. Huang, H. Pu, B. Gong, H. Lin, S. Ma, X. Chen, B. Long, G. Si, H. Yu, "Comparison of different samples for 2019 novel coronavirus detection by nucleic acid am- plification tests.," Int. J. Infect. Dis., 2020.

[2]. M. Shen, Y. Zhou, J. Ye, A.A. AL-maskri, Y. Kang, S. Zeng, S. Cai, "Recent advances and perspectives of nucleic acid detection for coronavirus," J. Pharm. Anal., no. xxxx, 2020.

[3]. World Health Organization, "Laboratory testing for 2019 novel coronavirus (2019-nCoV) in suspected human cases," vol. 2019, no. January, pp. 1–7, 2020.

[4]. Q. Li, X. Guan, P. Wu, X. Wang, L. Zhou, Y. Tong, R. Ren, K. S. Leung, E. H. Lau, J. Y. Wong, X. Xing, "Early Transmission Dynamics in Wuhan, China, of Novel Coronavirus– Infected Pneumonia," N. Engl. J. Med., pp. 20–21, 2020.

[5]. Z. Y. Zu, M.D. Jiang, P. P. Xu, W. Chen, Q. Q. Ni, G. M. Lu, L. J. Zhang, "Coronavirus Disease 2019 (COVID-19): A Perspective from China," vol. 2019, 2019.

[6]. T. P. Velavan and C. G. Meyer, "The COVID-19 epidemic," Trop. Med. Int. Heal., vol. 25, no. 3, pp. 278–280, 2020.

[7]. Mohammed, M. N., Hazairin, N. A., Syamsudin, H., & Al-Zubaidi, S., "2019 Novel Coronavirus Disease (Covid-19): Detection and Diagnosis System Using IoT Based Smart Glasses". International Journal of Advanced Science and Technology, Vol. 29, 954-960, 2020.

[8]. Ng, E. Y., Kaw, G., & Chang, W., "Analysis of IR thermal imager for mass blind fever screening". 104-109, 2004.

[9]. M. N. Mohammed, S. F. Desyansah, S. Al-Zubaidi, E. Yusuf,

An internet of things-based smart homes and healthcare monitoring and management system, Journal of Physics: Con- ference Series 1450, 012079.

[10]. S. L. Fong, D. Wui Yung Chin, R. A. Abbas, A. Jamal, and F. Y. H. Ahmed, “Smart City Bus Application with QR Code: A Review,” 2019 IEEE Int. Conf. Autom. Control Intell. Syst. I2CACIS 2019 - Proc., no. June, pp. 34–39, 2019.

[11]. S. L. Fong, D. Wui Yung Chin, R. F. Y. H. Ahmed and A. Jamal, “Smart City Bus Applica- tion with Quick Response (QR) Code Payment,” in ICSCA '19 Proceedings of the 2019 8th International Conference on Software and Computer Applications, Pages 248-252, Penang, Malaysia — February 19 - 21, 2019.

[12]. GitHub: PiCameraStream. https://gist.github.com/nioto/9343730?fbclid=IwAR0fKY6UR G33SJDcCfZw3gMaugP0cf fTw4IsBwW3GIihC32uAkxgm0OcQrI Retrieved 03-04-2021.

[13]. Cambridge in colour: Image Interpolation https://www.cambridgeincolour.com/tutori- als/image-interpolation.htm. Retrieved 04-04-2021.

[14]. Serveo: How it works manual and alternatives self-host https://serveo.net/. Retrieved 04- 04-2021.