JCEC
THE JOURNAL OF ENGINEERING AND EXACT SCIENCES

OPEN ACCESS

The Journal of Engineering and Exact Sciences – jCEC, Vol. 08 N. 07 (2022)
journal homepage: https://periodicos.ufv.br/jcec
eISSN: 2527-1075
ISSN: 2446-9416

System for acquisition and transmission of thermal images to monitor equipment of an electrical substation

Sistema de aquisição e transmissão de imagens termográficas para monitoramento de equipamentos de uma subestação de energia

Article Info:
Article history: Received 2022-08-14 / Accepted 2022-09-20 / Available online 2022-10-13
doi: 10.18540/jcecvl8iss7pp14750-01e

Odair José Sanson Neto
ORCID: https://orcid.org/0000-0002-7930-3244
Universidade Tecnológica Federal do Paraná, Brazil
E-mail: odairneto@alunos.utfpr.edu.br
Leonardo Göbel Fernandes
ORCID: https://orcid.org/0000-0002-3242-4671
Universidade Tecnológica Federal do Paraná, Brazil
E-mail: leonardofernandes@alunos.utfpr.edu.br
Alceu André Badin
ORCID: https://orcid.org/0000-0003-3243-4092
Universidade Tecnológica Federal do Paraná, Brazil
E-mail: badin@utfpr.edu.br
Winderson Eugenio dos Santos
ORCID: https://orcid.org/0000-0003-0135-5703
Universidade Tecnológica Federal do Paraná, Brazil
E-mail: winderson@utfpr.edu.br
Eduardo Félix Ribeiro Romaneli
ORCID: https://orcid.org/0000-0003-0592-5195
Universidade Tecnológica Federal do Paraná, Brazil
E-mail: felix@professores.utfpr.edu.br

Resumo
Sistemas de monitoramento termográfico têm obtido resultados satisfatórios ao identificar pontos com temperaturas anômalas, antecipando a ocorrência de falhas. No desenvolvimento de uma metodologia para análise automática de imagens térmicas, faz-se necessário um sistema automatizado de aquisição e transmissão de dados. O presente artigo apresenta um comparativo entre os métodos de aquisição de imagens de uma câmera FLIR® A700: Atlas SDK, interface web e Rest API, entre os quais destaca-se este último devido a sua facilidade de implementação e simplicidade. Além disso, o artigo apresenta o software desenvolvido para captura automatizada de imagens térmicas e o sistema de comunicação elaborado para transmitir essas imagens.
Palavras-chave: Aquisição de imagens térmicas. Transmissão de dados. Subestação de energia elétrica.

Abstract
Thermographic monitoring systems are achieving satisfactory results in identifying points with anomalous temperature, preventing faults to occur. In the development of a methodology for automatic analysis of thermal images, an automated system for data acquisition and transmission is necessary. This article aims to compare the different ways to acquire images from a thermal camera FLIR® A700: Atlas SDK, web interface and Rest API, among which the latter stands out because of its simplicity and easy implementation. The article also describes the developed software for

automated capture of thermal images and the communication system developed to transmit these images.

Keywords: Thermal image acquisition. Data transmission. Electrical substation.

1. Introduction

Electrical substations are facilities responsible for distributing electrical energy. For this purpose, several equipment such as transformers and protection devices are needed. These equipments can suffer damage over time, causing failures. To meet the demands of continuity in energy supply and to comply with the quality and availability indexes imposed by the National Electric Energy Agency (ANEEL), it is convenient for electrical systems operators to reduce the failures that can cause service interruption in an electrical substation (Pinto et al, 2008; Souza, 2008; Tarrago, 2019).

Thermographic monitoring systems have been achieving satisfactory results in identifying points with anomalous temperature, preventing faults to occur. As there are more electrical substations than qualified manpower to perform thermal inspections, several alternatives are being developed for automated thermal image acquisition (Guo et al, 2010), such as the project PD 2866-0528/2020 – Development of a methodology for automatic analysis of thermal images, carried out by the Federal Technological University of Paraná (UTFPR), financed with resources from the P&D executed by COPEL-DIS (*Companhia Paranaense de Energia - Distribuição*) and regulated by ANEEL.

In this project, in order for the monitoring and analysis processes to be performed in an automated way, it was necessary to develop an image acquisition system and a data transmission system.

The developed acquisition system is composed of a FLIR® A700 thermal camera, an AGV (Automated-Guided Vehicle) and a Pan-Tilt (electromechanical device capable of perform the angular positioning of cameras). The AGV and the Pan-Tilt were developed in partnership with local manufacturers. Both are responsible for positioning the thermal camera, which must capture images of each device of the electrical substation. Then, the thermal data captured must be transmitted to a remote server, where it will be analyzed by thermal segmentation algorithms.

Therefore, this article aims to describe the data acquisition and transmission system of a FLIR® A700 thermal camera from an electrical substation to a remote server.

2. Methodology

Exploratory research was conducted to develop the communication system architecture. Moreover, means of data extraction and control of the FLIR® A700 thermal camera were explored.

The enterprise Teledyne FLIR® provides technical documentation about the SDKs (Software Development Kits) and APIs (Application Programming Interfaces). Both of them are used to control FLIR® cameras. Several tests were performed with the FLIR® A700 camera, comparing the Atlas SDK, the Rest API, and the web interface of the camera, for the automated acquisition of thermal images. The tests included focus control, camera calibration, and optical and thermal image acquisition.

The focus control is essential because the equipment will be at different distances from the camera. Therefore, it is necessary to use autofocus before capturing each image. Furthermore, it is particularly important to often execute the camera calibration, a process performed through internal sensors to compensate the thermal measurement of the camera based on its encapsulation temperature.

In addition to the thermal images that contain the radiometric data of the devices, optical images must be captured to so that the equipment be identified by a neural network.

The thermal images are .jpg files that contain metadata in the EXIF format. EXIF is a specification from JEITA (Japan Electronics and Information Technology Industries Association), which defines how images of digital cameras are saved in files, including both the image and tags with capture information. The Exiftool is a Python library that was chosen to be used to extract data from the EXIF standard. It was the simpler solution found, and it is aligned with the project objective of finding the temperature value in each pixel of the image. The essential metadata for the thermal processing of the images of a FLIR® camera are shown in Table 1.

Table 1 – Essential metadata for the thermal image processing.

Metadata
RawThermalImage
Emissivity
Object Distance
Reflected Apparent Temperature
Atmospheric Temperature
IR Window Temperature
IR Window Transmission
Relative Humidity
Planck R1
Planck B
Planck F
Planck O
Planck R2
Atmospheric Trans Alpha1
Atmospheric Trans Alpha2
Atmospheric Trans Beta1
Atmospheric Trans Beta2
Atmospheric Trans X

2.1 Atlas SDK

Atlas SDK is a software development kit provided by FLIR® for the creation of applications of thermal camera control. The SDK contains several libraries in the C# programming language, which allow for actions as focus control, image and thermal data acquisitions, and even a post-processing of the images. Atlas was elaborated to develop applications for .NET framework of Microsoft, generally used in Windows operational systems. To use it with the Python programming language, libraries like Python.NET, which is a package that allows for the integration of .NET Common Language Runtime (CLR) and the Python language, must be employed. Furthermore, to be integrated in Linux operational systems, Mono environment must be used, which provides support to the .NET framework.

The SDK contains two main modules: Image, which is oriented to produce and process thermal images; and Live, which is used to control remote cameras (FLIR, 2022). The Live module presents a device class denominated Device, which contains several C# language classes to declare different types of cameras, as illustrated in Figure 1.

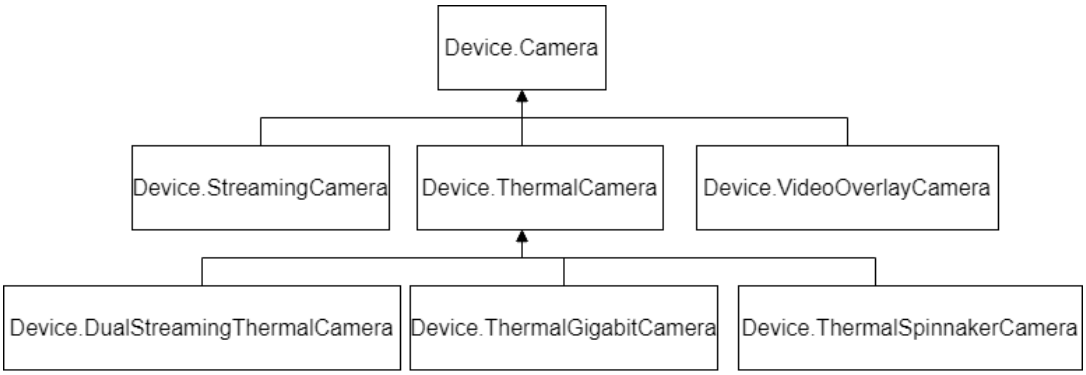

Figure 1 – Camera classes – Atlas SDK

2.2 Web Interface

The web interface of FLIR® A700 presents features like optical and thermal video streaming, focus control, and image acquisition.

Selenium is a framework that allows for action automation in browsers, and it is generally used to test web applications.

Using the selenium library of the programming language Python, it is possible to simulate the actions of a user of the FLIR® A700 web interface and then capture image and control the camera in an automated way.

2.3 Rest API

Rest API is an application programmable interface that works with HTTP request to send commands to a thermal camera. The Hypertext Transfer Protocol (HTTP) is a communication protocol used to transmit hypermedia documents. It is the base of any web data exchange, where the recipient initiates the requests, in this case, the client that desires to command the FLIR® camera. Actually, the FLIR® A700 web interface was constructed using the Rest API. In the HTTP header body, beyond the camera IP, the commands to control focus and to capture images, for example, are inserted.

With Rest API, it is possible to use the GET method of HTTP requests to capture thermal and optical images of different resolutions. And with the POST method, it is possible to control the focus and calibrate.

GET is a verb of the HTTP protocol and it is used just to receive data. The POST method, otherwise, is used to send information or data to a resource, making it possible to change the state of the resource. This way, the verb GET, unlike the POST method, does not present body in the request.

To incorporate Rest API with Python, the request library can be used, which allows to send HTTP requests in a straightforward way (Requests: HTTP for Humans v2.28, 2022).

2.4 Communication technologies

4G is the acronym for the fourth generation of mobile telephony. This technology has made significant advances in data transmission over previous generations. For example, with 4G, it is possible to reproduce high-definition videos and make videoconference calls with high stability. With a 4G modem, the 4G communication channel can be used not only to cell phones but also to computers and other network devices, substituting the cable internet.

VPN is a Virtual Private Network that establishes a secure connection through data cryptography. Through the VPN tunneling system, data can be sent and received only by the users of this network. Furthermore, as it is built over the infrastructure of a public network, it is a cheap privacy solution.

MikroTik is a router that has features for VPN, Proxy, Hotspots, Band control, QoS and Firewall configuration. Actually, MikroTik is a Latvian company that develops network devices.

Among these, there are the RouterBoard series devices, which use an operational system based on Linux called MikroTik RouterOS with its own hardware line.

The data transmission system contains a local network that works with the IP protocol. Figure 2 shows the system equipment.

The local network uses a VPN to communicate with the remote server. To provide internet access to the local network via VPN, the 4G technology was implemented. The access to the virtual private network is granted by a MikroTik.

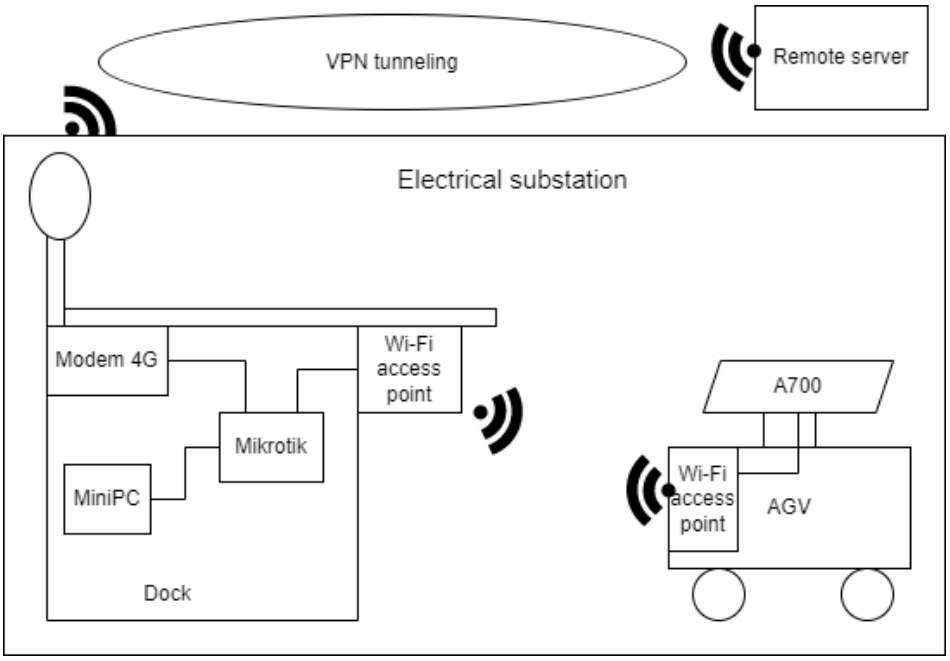

Figure 2 – Image transmission system architecture

3. Results and discussion

With the StreamingCamera class, contained in Atlas SDK, it is possible to perform optical image acquisition. ThermalCamera and the other classes that inherit its features – DualStreamingThermalCamera, ThermalGigabitCamera and ThermalSpinnakerCamera – made it possible to capture thermal images, which contain thermal metadata. With VideoOverlayCamera class, it is possible to capture the web interface streaming image. DualStreamingThermalCamera is the most promising class, as it made it possible to capture both the thermal and the optical image, which can be made in Python language using Python.NET library, as illustrated in the SysML (Systems Modeling Language) activity diagram of Figure 3.

The thermal images collected by any of these classes contain all the essential metadata of Table 1. However, although the images can be processed by FLIR® software programs, this does not happen with the Exiftool due to an error in extracting the RawThermalImage metadata, a binary file that contains information of each pixel temperature of the image. This error occurs due to an incorrect value of a GPS flag: "*ExifTool:Warning GPS pointer references previous ExifIFD Directory.*"

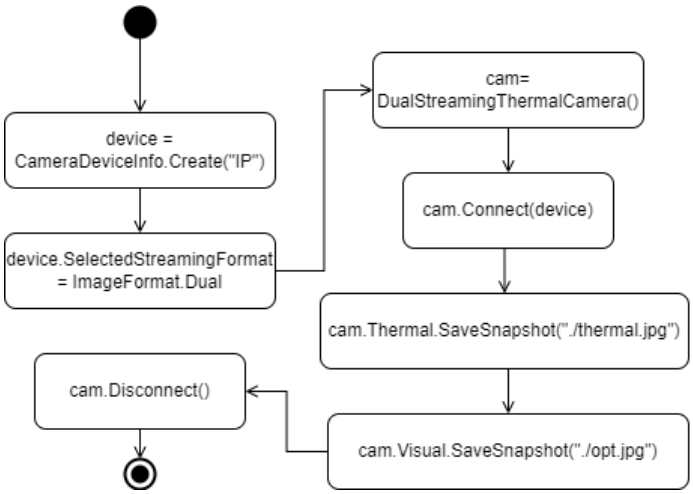

Figure 3 – SysML activity diagram representing the image acquisiton via Atlas SDK.

The web interface, despite having all the essential metadata for the project, became unfeasible as the only way to automate the image capture is by using libraries/modules of browser automation, like Selenium Python. However, the application presented several problems, such as major time for acquisition and inconsistency between interface versions. Each FLIR® A700 firmware version has an interface with different features, making it necessary to update the solution via Selenium, so that the system can run correctly.

Rest API made it possible to capture thermal image with resolution of 640x480, optical image with resolution of 1280x960 and optical image with resolution of 2592x1944, which one provided a major field of view. Also, it is possible to get an optical image with resolution of 640x480 which field of view matches exactly with the thermal image one. To perform the thermal image acquisition via Rest API using Python, the diagram illustrated in Figure 4 must be observed, where IP (Internet Protocol) is the camera IP address and snap_name is the file name where the image will be saved.

After making the HTTP request, it is necessary to open the file with the open() method and to save the content obtained with the write() method. The metadata of captured thermal images acquired by this capture method have all the essential information of Table 1, and the file RawThermalImage can be extracted by Exiftool with no problem.

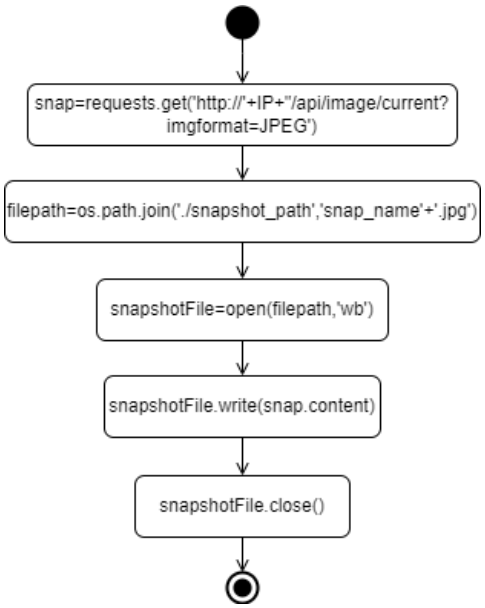

Figure 4 – SysML activity diagram of thermal image acquisition via Rest API with Python language.

A Python class was developed to control the camera via Rest API, which provides focus control, camera calibration and thermal, optical, and high resolution optical image capture. The class methods and properties are described in Figure 5.

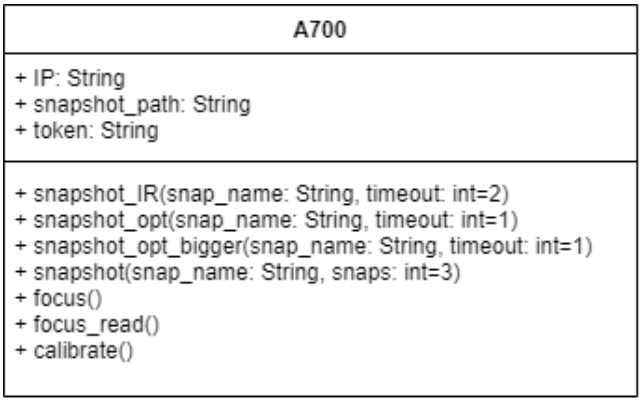

Figure 5 – A700 Rest API class – methods and properties

The token property is responsible for authentication and is inserted in the header of functions that use the POST method. This token must be created in the web interface of the FLIR® A700 camera. The focus() and calibrate() methods are represented, respectively, in Figures 6 and 7. It is possible to observe the use of the POST method with the requests library, where the data to be sent is in the data parameter and the authentication token is in the header.

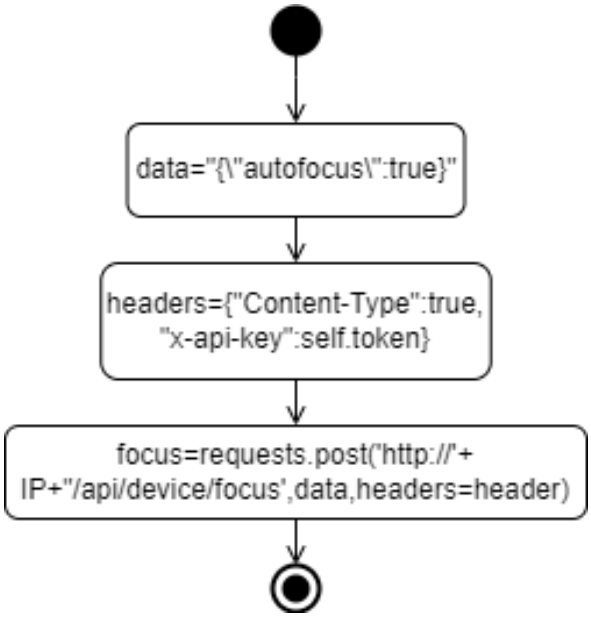

Figure 6 – SysML activity diagram of focus() method to perform autofocus

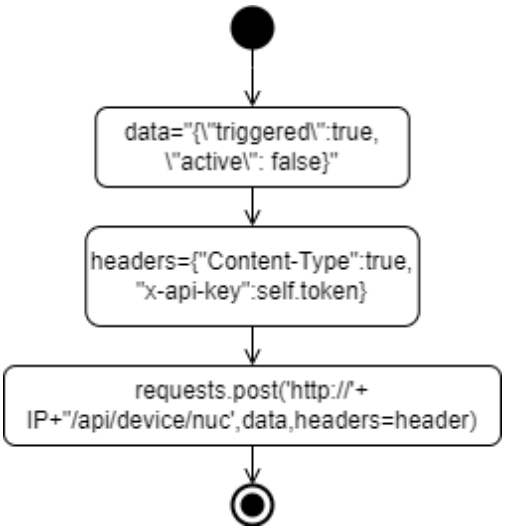

Figure 7 – SysML activity diagram of calibrate() method to calibrate the camera

The snapshot() method captures thermal, optical, and high resolution optical images. In addition, it solves possible connection problems that can delay the HTTP request dispatch and the reply receipt, through a capture retry logic, as illustrated in Figure 8 by a SysML state machine diagram. In this diagram, the rectangles with rounded corners represent states of the system. In each state, entrance activities (entry/), state activities (do/), and exit activities (exit/) are performed. The activities described in Python language are represented with "{Python}" in the beginning of the text. Moreover, the arrows show the transition between states (Friedenthal et al, 2015).

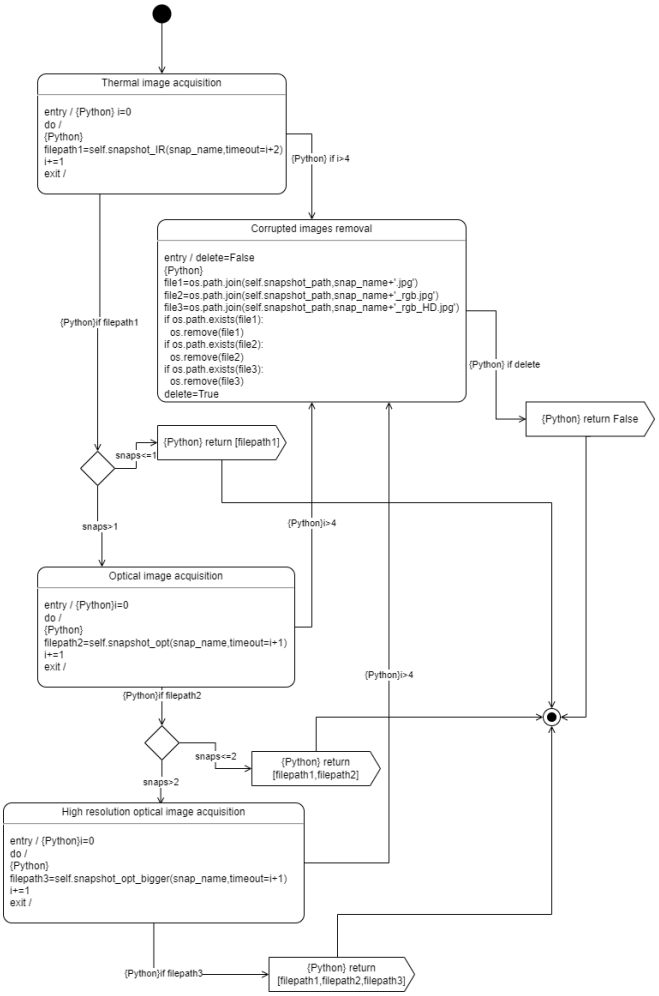

Figure 8 – SysML state machine diagram of the snapshot() method.

Table 2 presents a comparison between the different types of image acquisition method concerned about their main features.

Table 2- Comparison between image acquisition methods.

Acquisition Method	Operational System (OS)	RawThermalImage extracted by Exiftool	Easy automation
Atlas SDK	preferably Windows	No	Yes
Interface *web*	Windows, Linux	Yes	No
Rest API	Windows, Linux	Yes	Yes

In order to allow the free movement of the AGV, the data transmission of the thermal camera to the industrial miniPC must be via Wi-Fi. All the devices of the dock communicate via Ethernet protocol, as illustrated in Figure 2.

The miniPC is responsible for controlling all the system. Thus, it perform a HTTP request to the thermal camera to capture an image. Then, this image is saved in a file in miniPC, as illustrated in Figure 9.

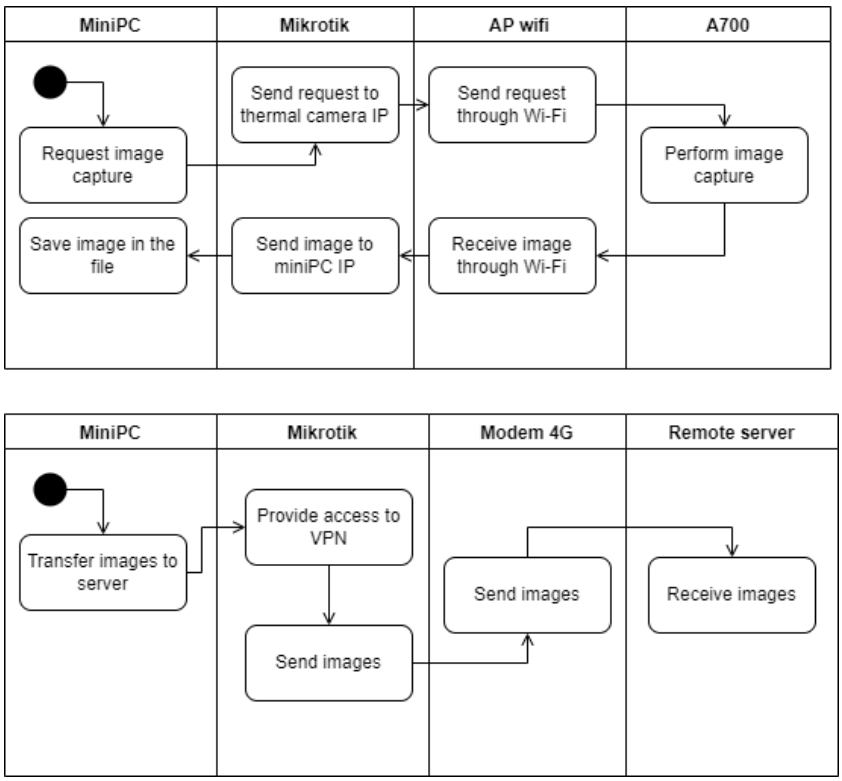

Figure 9 – Transmission system activity diagram.

When the miniPC requests the file transfer to the remote server, the MikroTik provides access to VPN, and through the 4G channel the images are sent to the server.

The company that developed the AGV provided a Python API, which contains methods and properties to control the position of the AGV, the position of the Pan-Tilt and to obtain several statuses of the system. Integrating the FLIR® A700 Python class developed to control the thermal camera with the AGV API, a script was developed to perform automated missions of image capture. This script is represented in Figure 10 by a SysML state machine diagram.

First, the script connects to the database of the mission and obtains the stop points of the AGV and positions of the Pan-Tilt to the mission with identifier mis_id. The Connect_database(mis_id) activity returns a list called mission[]. A list in Python is a sequence of objects and can be modified

through insertion or removal of items. In this case, the mission[] list is composed of several tuples. A tuple in Python is also a sequence of objects, but it is immutable. Each tuple of mission[] list represents an image acquisition position; therefore, it has three values: coordinate of the AGV stop point (x), Pan-Tilt azimuth angle (azi), and Pail-Tilt elevation angle (ele).

After the creation of the mission[] list, the connection between the miniPC, where this script is running, with the AGV and the FLIR® A700 camera is initiated.

Then, the mission starts; the AGV moves to the first stop point described by the tuple with index 0 in the mission list. When it arrives at the point, the Pan-Tilt is positioned, the camera performs autofocus and the snapshot() method is called to capture thermal, optical, and high resolution optical images. The AGV moves then to the stop point presented at the tuple of index 1. If this point is the same as the last one, the AGV stays in the same coordinate. The Pan-Tilt is then positioned according to the tuple values; autofocus and image acquisition are performed.

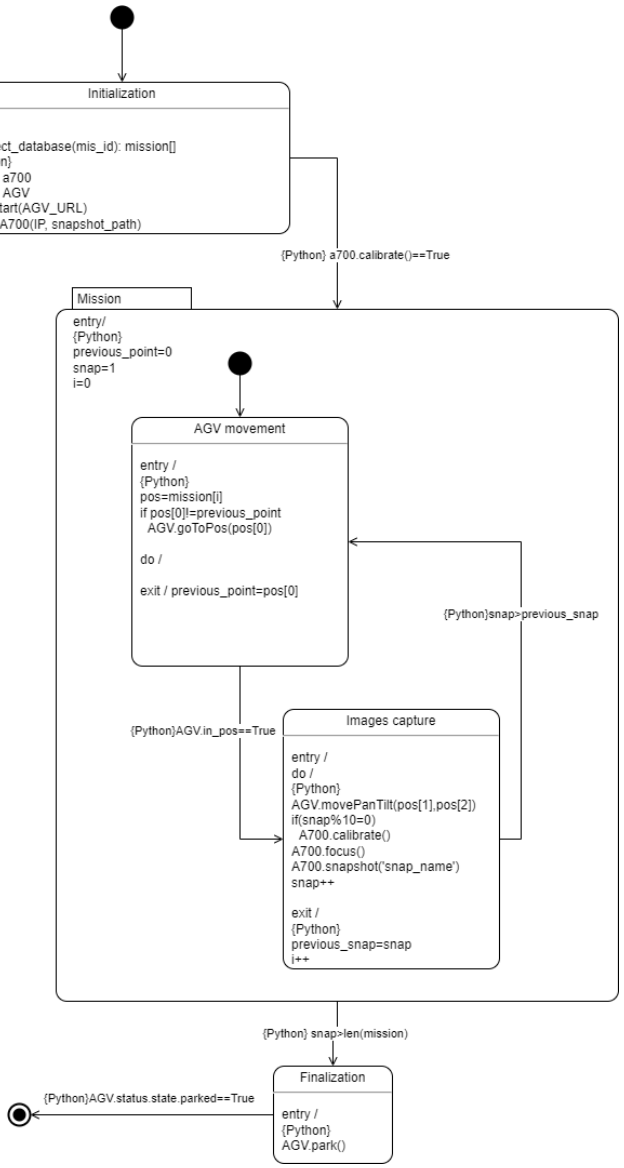

Figure 10 – SysML state machine diagram of an automated mission.

In the image capture state, for each 10 images captured, the camera is calibrated to ensure the measurement precision of the thermal sensor during the capture of thermal data.

The described process repeats itself until all the mission acquisitions are performed. After the conclusion, i.e., when the mission list length is smaller than the variable snap (image number), the AGV receives the command to park in the dock through the AVG.Park() method.

4. Conclusion

In the beginning of this work, the ideal method to capture images should be found, and then it would be implemented in a solution to perform automated thermal image acquisition missions.

Considering the thermal image processing, and due to its implementation simplicity in Python language, in addition to the facility of integration in the automated system, the acquisition method chosen was Rest API.

The development of the system was well concluded and, for future works, a practical user interface to supervise the missions in real time may be developed.

Notwithstanding, the developed class in Python to command the FLIR® A700 camera can be used in other applications where a custom solution of camera control and automated image acquisition is necessary.

Acknowledgements

The authors would like to thank the UTFPR for the support and infrastructure provided for the development of this research and the Copel for the support and resource financing to execute the project PD-2866-0528/2020 – Development of methodology for automatic analysis of thermal images, financed with resources from P&D executed by COPEL-DIS and regulated by ANEEL.

References

Pinto, J. K. C.; Masuda, M.; Magrini, L. C.; Jardini, J. A.; Garbelloti, M. V (2008). *Mobile robot for hot spot monitoring in electric power substation*, IEEE, Chicago, IL, USA.

Souza, F. A (2008). *Detecção de Falhas em Sistema de Distribuição de Energia Elétrica Usando Dispositivos Programáveis*. Dissertação (mestrado) - Universidade Estadual Paulista, Faculdade de Engenharia de Ilha Solteira, Ilha Solteira.

Tarrago, R. A (2019). *Confiabilidade de Subestações de Transmissão de Energia Elétrica com Aplicação de Equipamentos de Manobra não Convencionais*. Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul, Escola de Engenharia, Porto Alegre.

Guo, R.; Han, L.; Sun, Y.; Wang, M (2010). A mobile robot for inspection of substation equipments. 1st International Conference on Applied Robotics for the Power Industry, 2010.

FLIR (2022). *Atlas-Cronos SDK 7.0.2 Documentation*.

Reitz, K (2022). *Requests: HTTP for Humans*.

Friedenthal, S.; Moore, A.; Steiner, R (2015). *A Practical Guide to SysML : The Systems Modeling Language*. Waltham, MA: Morgan Kaufmann.