PRE- AND POST-PROCESSING OF SPACE IMAGING FOR WILDFIRE DETECTION USING CONVOLUTION NEURAL NETWORK APPROACH

by

Muhammad Hasif bin Azami

195F5005

Summited to the

Department of Engineering Electrical and Space Systems

Graduate School of Engineering

Kyushu Institute of Technology, Kitakyushu, Japan

Thesis submitted in fulfillment of the requirement

for the degree of

Doctor of Philosophy in Engineering

September 26th, 2022

ABSTRACT

An increasing number of wildfire cases every year has caused fear around the world. Scientists and researchers agreed that this catastrophe occurred due to climate change. Dry and windy conditions had worsened the situation in the affected area. Properties and life losses have created serious concerns for the authority to find a solution for preparing and fighting the fire promptly. Since the late '70s, leveraging satellite technology has brought helpful insight to monitor, detect, and assess wildfire events. NOAA AVHRR is one of the oldest Earth Observation (EO) satellites with the main objective of detecting and mapping forest fires. The MODIS fire product regularly upgrades the sensor technology and launches the satellites into space. However, with the advancement of current technologies, a miniaturized satellite called CubeSat creates a novel mission design by reducing the satellite development time, increasing the launching batch in a constellation method, and enhancing the detection result wildfire. The prime limitations of CubeSat are the size, weight, and power (SWaP), which lead to the optimization design of the payload and the communication subsystem. The big image data acquired by the CubeSat creates a bottleneck effect between the satellite and the ground station due to the low downlink data rate.

Deep learning (DL) techniques are improving in the computer vision area. Image classification, detection, and segmentation are used in neural network architecture designed by artificial intelligence researchers. In this study, the convolution neural network (CNN) algorithm was chosen for the pre-processing onboard CubeSat for wildfire detection as well as for the graphical user interface (GUI) used on the ground post-processing. The first and crucial step was to develop a custom dataset for wildfire images by leveraging satellite imagery. Defining the specifications of the CubeSat payload to which the CNN was implemented could support selecting the accurate resolution and bands for acquiring the satellite images. The KITSUNE satellite is a 6-unit CubeSat platform implementing the CNN onboard for wildfire image classification. It serves as the secondary mission to support the main mission of a 5-m class EO. The on-ground testing revealed that the CNN could classify wildfire occurrences on the satellite system using the MiniVGGNet network with an overall accuracy of 98 % and an F1-score of 97% success rate in 137 seconds. Other models were also compared, such as ResNet and MiniGoogLeNet implemented on the GUI with 97% and 96% F1-score, respectively. Overall, this research showed the feasibility of CubeSat of executing CNN onboard in orbit, particularly for wildfire detection.

Keywords: wildfire, convolution neural network, onboard classification, optical payload, CubeSat.

ACKNOWLEDGEMENT

First and foremost, I would like to give a heartily thankful to my supervisor, Professor Mengu Cho for his guidance and support since the beginning of my research while giving me the opportunity to be a part of KITSUNE member. This credit is also given to Dr Necmi Cihan Orger, Dr Victor Hugo Schulz, and Dr Jose Rodrigo Cordova Alarcon for their full support during the KITSUNE project.

This research, which was related to the KITSUNE mission, would not have been possible without the technical support of the Laboratory of Spacecraft Environment Interaction Engineering, Kyushu Institute of Technology. Secondly, I wish to express my sincere gratitude to my supportive KITSUNE members and other partners. The great teamwork makes the KITSUNE project has been operated in space, and well-coordinated with all KITSUNE and BIRDS Ground Stations Network operators.

Special appreciations are dedicated to my beloved wife, Fadhilah Humaira' binti Muhamad Rahimi and family members, especially my beloved father and mother, for their support and prayers. Lastly, many thanks to those who helped in my research directly and indirectly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Overview

Natural disasters like wildfire (or forest fire) have become a serious hazard that affects most of Earth's ecosystem. Earth Observation (EO) satellites are being utilized to provide an informative picture of wildfire coordinates and size. A miniaturized satellite called CubeSat offers an advantage of launching in the constellation method due to the less development cost and fast delivery. Daily fire images and a high ground resolution of the CubeSat constellation assist the EO traditional satellite services. However, the limitation of CubeSat size creates significant challenges to downlinking numerous image data, which leads to a study of pre-processing onboard CubeSat and post-processing on the ground station.

1.2 Problem statements

The EO satellites are launched into space in different orbits. Low Earth orbit (LEO), sun-synchronous orbit (SSO), and geostationary orbit (GEO) are the locations of active EO satellites. The higher the altitude of the satellite, the more challenging it is to capture a small fire size (<1 km^2), however, it could cover a large area of Earth. The trade-off of this issue shows the study of deploying a swarm of EO CubeSats at the LEO that can support the EO traditional satellite image data.

Furthermore, the CubeSat is known for its constraint of the communication subsystem. Considering the size, weight, and power (SWaP), the developers have difficulties downloading large image data captured by the payload to the ground station. The specification of imaging payload could be increased as well as the communication of downlink speed. Another option is implementing an algorithm to filter the image data before retrieving it to the users. Therefore, the CubeSat operator would only increase the chances of acquiring useful data.

In addition, machine learning (ML) or deep learning (DL) is an effective method for image processing nowadays. This subset of artificial intelligence (AI) could be implemented onboard CubeSat. However, the problem is the lack of high-resolution satellite images for training the dataset. The open and freely available images are from MODIS (250 m resolution), Landsat-8 (30 m resolution), and Sentinel-2 (10 m resolution) that have been used by MCD64A1C6, FireCCI51, and BA-Net datasets. Nevertheless, these data are limited and not daily images. Therefore, an account should be registered for Doves satellite images of 3-m resolution to acquire high-resolution daily data.

1.3 Research motivations

Wildfire cases have been reported every year. The authorities and experts are struggling to search for new solutions to overcome this catastrophe. In this study, the first research motivation is to help Malaysian authorities fight forest fire events by providing a start-of-art solution using CubeSat. The main goal is not only to monitor the occurrences happening in Malaysia, but the satellite images could also cover Southeast Asian countries such as Indonesia, Singapore, Thailand, and the Philippines. The haze produced by the wildfire repeatedly affects the neighbouring countries. Therefore, a constellation of CubeSat is an ideal option to detect and monitor across the Asian countries in low-cost development.

CubeSat under the nanosatellite category is typically classified and considered an under-rated imaging mission. The bus system's limitation creates a boundary for the developers to enhance the capability of CubeSat. The second motivation is to expand the potential of the CubeSat imaging mission for remote sensing applications. The size and power consumption of CubeSat have defined the selection of camera payload. In addition to the hardware design, the image processing algorithm onboard design is an innovative method. The software development would not affect the limitations of CubeSat, but it would increase the imaging mission's effectiveness.

The community of computer vision has widely used image processing tasks implementing AI. The main benefit of AI is fewer humans operating a particular work. Therefore, it could create an AI ecosystem to process satellite images on the ground as well as onboard satellites. Large image data captured by the satellite will burden the ground station operator to analyze. Hence, deploying the AI methods would considerably boost the work speed efficiently. For instance, remote sensing images such as natural disaster monitoring requires rapid actions to provide insights for the respected authorities.

1.4 Research objectives

This research aims to develop a wildfire dataset implementing the deep learning classification algorithm onboard CubeSat for fire detection and extend the development of the ground station graphical user interface (GUI) as the software for analysis. The convolution neural network (CNN) approach was utilized in this study for the pre-and post-processing image classification. The CNN pre-trained models could classify any input images as wildfire, cloud, land, or sea.

The main objectives of this research are stated as follows:

1) To investigate the wildfire problem in remote sensing applications.
2) To design a robust and high accuracy deep learning technique implemented in nanosatellite class.

3) To test, troubleshoot, and evaluate the performance of pre and post-processing imagery data for nanosatellite application.

1.5 Research scopes

This study has been defined into several scopes. Firstly, the recorded wildfire events are more than 5 hectares worldwide. The database consists of the date, coordinate, burned area, cause, suppression cost, and casualty between 2010 and 2020. Smaller fire sizes and the latest wildfire event data were used to analyze the accuracy of the classification algorithm.

Secondly, the images used for the dataset are visible spectrum bands (RGB) only. The three channels of daytime images are commonly implemented in the CubeSat imaging payload. However, the shortwave infrared (SWIR) range is the recommended spectrum for fire monitoring. Due to the limitation of the CubeSat payload, this study focused on leveraging the visible bands only for wildfire detection.

In addition, the deep learning approaches were applied onboard CubeSat as the pre-processing and on-ground computer as the post-processing. Image recognition is the technique applying the CNN network. Furthermore, image segmentation could provide a more valuable analysis of the ground segment, but the generated dataset can only work using the chosen technique. Thus, the feasibility of the CNN implementation was discussed thoroughly in this study.

Remote sensing has a wide area of applications, including atmosphere, ocean, land, and cryosphere. Forest fire in the land application was the main focus of this study. The EO satellites are the best tool to detect, monitor, and assess wildfire. Nonetheless, the study aimed to detect active fires only. Smokes and red pixels were the keys to classifying wildfire accurately. Hence, the colour images could be utilized optimally to run the image classification onboard for this land application.

1.6 Research novelty

The novelty of this research was creating a new dataset consisting of four labels utilizing the visible bands and multi-resolution: (1) wildfire, (2) cloud, (3) land, and (4) sea. Other researchers have developed the wildfire dataset but are limited to the burned area size and particular countries using single satellite imagery. Thus, the unique dataset created in this study would benefit the existing datasets available generally for the remote sensing application and mainly for wildfires.

In addition, the pre-trained CNN models using the dataset could reach a significantly high accuracy (> 95%) in classifying wildfire images onboard CubeSat in space. The energy required to execute the CNN algorithm was tested and verified with the limitation of CubeSat specification. Other deep learning techniques have been tested on ground-based applications,

but they are not feasible to be implemented onboard due to the computational costs. Searching for an optimum CNN model is certainly crucial in implementing onboard CubeSat.

This study would also improve the amount of valuable image data to be downlinked to the ground station. The KITSUNE CubeSat is the platform to analyze the efficiency of the CNN implementation. A major issue of launching CubeSat has been identified as downlink speed. As the imaging payload increases the performance, the communication speed should also be agile. Therefore, the onboard image processing could help filter the data before downloading it without upgrading the CubeSat transceiver.

Lastly, the research believes the CubeSat could help reduce forest fire damage by using the CNN approach onboard. Active fire data from space is valuable for the respective authority to create an effective plan to reduce the damage and contain the fire. Even though the study is limited to visible band data, the CubeSat payload could prove its capability to overcome wildfire cases.

1.7 Thesis organization

This dissertation will be divided into 6 chapters. Chapter 1 discussed the general overview of the research, problem statements, motivations, objectives, and scopes. Chapter 2 focused on the literature review that serves as basic knowledge and background study. Several recent studies were reviewed to better understand the most recent developments in Earth observation satellites for detecting wildfires by applying deep learning techniques.

Chapter 3 introduced the KITSUNE project and the development process of camera payload implementing convolution neural network onboard satellite. This part will also explain the dataset development, training and classification tests, as well as the graphical user interface (GUI) development, which were the important aspects of the proposed method in this research. Chapter 4 presented the results of the space environment tests, long-duration operational tests, training and classification tests. The space environment tests consist of the sun simulator test, total ionization dose (TID) radiation test, thermal vacuum test, and vibration test. The preliminary in-orbit data was also shown and investigated in this chapter. Since the CubeSat was in the initial operation, the flight data was limited to be further investigated.

Chapter 5 discussed the study results and compared the other studies. Later in this chapter, a case study of developing a dedicated 6U CubeSat for wildfire detection was presented based on the experience and lessons learn from the KITSUNE project. Chapter 6 summarized the research conclusion on the proposed method, contribution to the research community, and recommendations for further future works.

CHAPTER 2

RESEARCH BACKGROUND AND LITERATURE REVIEWS

2.1 CubeSats

2.1.1 History

CubeSats are cubic-shaped nanosatellite classes extensively developed by universities and public agencies. The satellite is normally categorized in mass and reflected on unit (U) size, from 1U to 27U. For instance, 1U is 1.33 kg and 10 cm cubic. The miniaturized CubeSat was invented by Professor Bob Twiggs (Standford University) and Professor Jordi Puig-Suari (California Polytechnic State University) in 1999. The first CubeSat successfully launched and operated is XI-IV from The University of Tokyo. Since 2003, the XI-IV has been orbiting and sending valuable data [1,2].

Currently, more than thousands of CubeSats have been launched into space from various identities globally. The main reasons are low production costs and rapid development (Fig. 1). 'Lean concept' has been introduced in the development of CubeSat to enhance its reliability [3]. Practically, the optimum production time should be less than two years or could increase the cost. Therefore, the simple design of CubeSats showed advantage compared to traditional satellites.

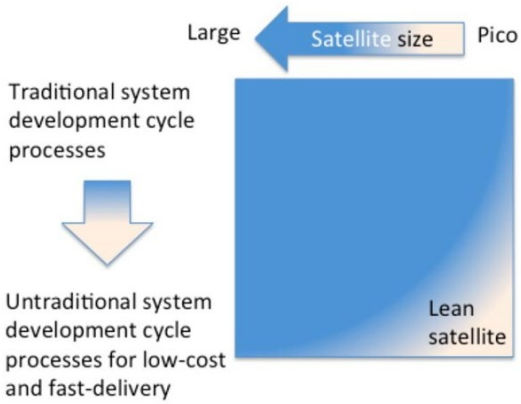


Figure 1. Lean satellites against traditional satellites. [3]

Furthermore, uncomplicated CubeSats development has established the concept of a satellite constellation. For instance, BIRDS projects launched a constellation of 1U CubeSat for technology demonstration (Fig. 2) [4], Planet and Spire have more than hundreds of CubeSat fleets orbiting Earth for remote sensing applications [5,6]. The price of launching a swarm of CubeSats is also reasonable compared to a single unit of a big satellite. CubeSats have options to be launched in soft or hard methods. The soft launch is when the CubeSats are sent to the International Space Station (ISS) and deployed from the payload orbital delivery (POD). The later

method is similar to traditional satellites released from the rocket fairing, known as the piggyback payload.

Figure 2. BIRDS-1, 2, 3, 4, and 5 projects from multi-nations.

2.1.2 Type of missions

Traditional satellites and CubeSats have been developed to have specific missions. Remote sensing, communications, education, scientific, and technology development are the mission types for CubeSat, which depend on organizations (Fig. 3). For example, schools and universities commonly choose to go with education and technology demonstrations.

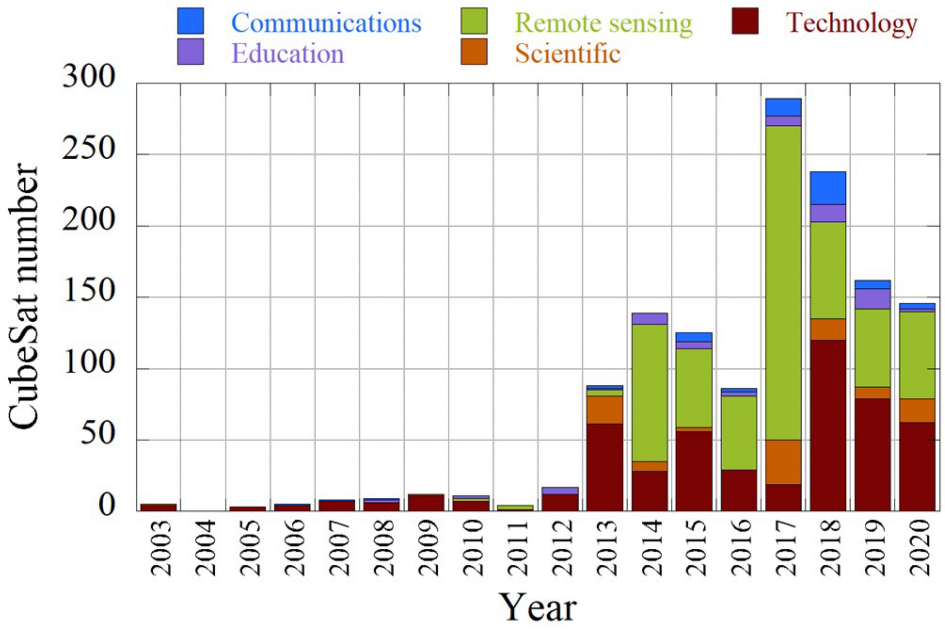


Figure 3. Kyutech database of CubeSat mission classification (1 – 10 kg) between 2003 and 2020.

On the other hand, space companies and agencies launch complex interplanetary missions. Mars Cube One (MarCO) is the first deep-space CubeSats beyond the low Earth orbit (LEO), which launched in 2018 [7]. It consists of two 6U CubeSats given mission as a relay data in real-time during InSight landing on Mars.

Therefore, the CubeSats show a promising platform for future complicated missions, which have been implemented in the traditional satellites.

Lunar and asteroid CubeSats are other state-of-the-art missions. Most of these planned missions used satellite platforms between 6U to 12U. The challenging task is to maintain the orbital, which requires a high-precision of attitude and orbit control system (AOCS), which 6U platform and above are the feasible options. For instance, to support the NASA Artemis program, a 12U CubeSat named CAPSTONE was planned to launch to the Moon in June 2022 [8]. Furthermore, LICIACube is a 6U CubeSat travelled with a DART satellite and will deploy in September 2022 [9]. Finally, when these missions are fully demonstrated, it would confirm the reliability of developing CubeSat for any missions similar to the big satellites.

2.2 Remote sensing satellites

2.2.1 History

The term ‘remote sensing’ comes from Evelyn Pruitt in the 1960s. Remote means to imply a measurement indirect contact sensor. It is mainly applied to the instruments of satellites and aircraft. Since the 1960s, these sensors have been miniaturized and fit into small satellite classes. Optical and microwave payloads work through the reflection and emission of electromagnetic radiation from Earth. The first EO satellite was TIROS (Television and Infrared Observation Satellite), launched by the US in April 1960 [10]. The 122 kg of satellite mass was able to record the first TV images of Earth from space.

Remote sensing (RS) satellites, also known as EO satellites, are important applications utilized in space after communication and navigation. Without the data from RS satellites, scientists could not predict weather conditions such as air pressure, temperature, speed, and humidity. Currently, several weather satellites are actively monitoring Earth in different orbits. NOAA-15/18/19, 20 and Suomi-NPP satellites are in the near-polar orbit, while GEOS satellites are stationed at GEO (Fig. 4). Hence, different altitudes complement each of the EO satellites for particular missions.

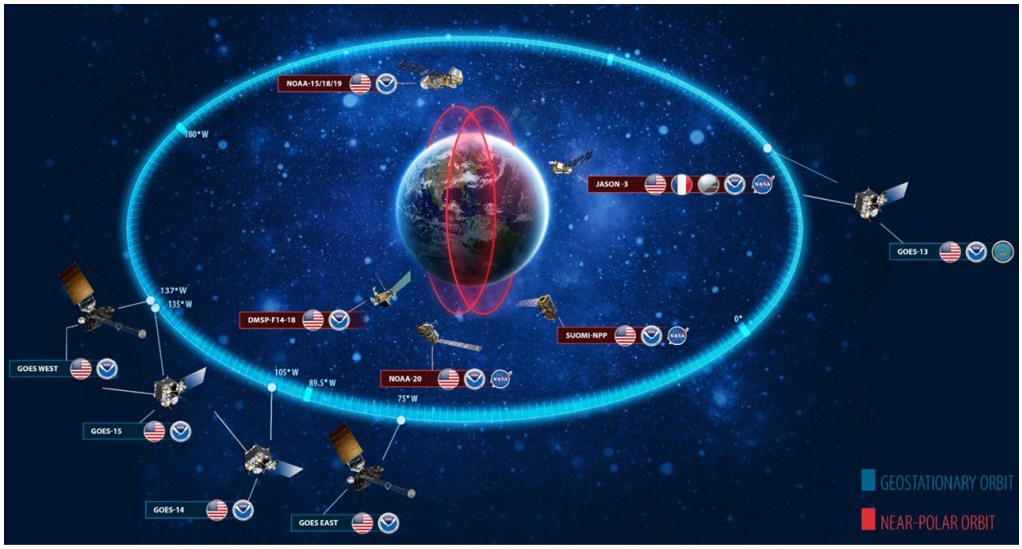


Figure 4. Active weather satellites orbiting in near-polar and geostationary orbits. [11]

2.2.2 Optical imaging sensor

Generally, optical imaging sensors have been installed as the payload of EO satellites. The electromagnetic spectrum of the sensor is within the visible (VIS) and the infrared (IR) range. Visible light spectrum wavelength covers the red, green, and blue bands (RGB) between 0.4 and 0.7 µm. The advantages of the VIS sensor are the high-resolution, small size, lightweight, and reasonable price, which lead to the first choice of implementing onboard satellite. However, this wavelength has limitations in penetrating the cloud for optimum ground sensing.

On the other hand, the IR or thermal bands have three different wavelengths: short-wavelength infrared (SWIR), medium-wavelength infrared (MWIR), and long-wavelength infrared (LWIR), which are from 1.3 to 15 µm range (Fig. 5). Compared to the VIS sensor, the resolution of IR is considerably low. The size, weight, and power consumption are affected due to the requirement of installing a cooling system on the IR sensor. It also has the advantage for various remote sensing applications, but the cost is quite expensive.

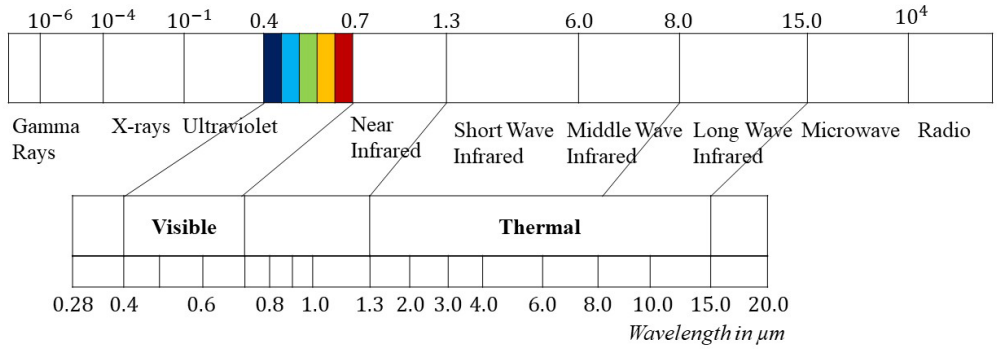


Figure 5. Electromagnetic spectrum in wavelength.

2.2.3 Land applications

EO satellites are built to detect, monitor, and assess changes on Earth. The applications could be divided into atmosphere, ocean, land, and cryosphere. Examples of land applications are monitoring vegetation, deforestation, floods, drought, and forest fire. Climate change increases disaster cases, particularly wildfires during the summer [12-14]. According to the United Nations Environment Programme and GRID-Arendal report, wildfires will become 50% more frequent globally by the end of the century [15]. Experts have warned the governments to focus on preventing and preparedness for wildfires. Therefore, EO satellites are the best tool for providing information on this catastrophe.

The 2020 Food and Agriculture Organization of the United Nations report states that Earth is covered by 31% (4.06 billion hectares) of the global land. Most of the forest area is in Russia (20.1%), Brazil (12.2%), Canada (8.5%), the USA (7.6%), China (5.4%), and Australia (3.3%). The green lands are gradually demolished due to unintentional occurrence that burns in grassland or forest. Wildfires (or forest fires) have two potential causes, which are natural phenomena or sparked by human actions. Lightning and dry weathers, such as drought are the major causes of the catastrophe. Over the past two decades, an average of 70,072 wildfires occurred annually in the USA, of which 2.8 million hectares were affected [16]. Furthermore, the extreme weather events in 2021 have increased the incidence of wildfires around the globe, such as in Turkey, Greece, Italy, France, Algeria, Lebanon, Jerusalem, and Siberia, which were recorded as the worst in decades in several countries [12-14]. Several studies have been conducted and analyzed the wildfire in Brazil forest (2001-2019), Russia, Canada, the USA, China, and Australia (2001-2018) [17,18]. Nonetheless, the criminal act of arson has been found in most incidents involving extraordinary wildfires. For instance, the Siberian wildfire in 2019 was caused by illicit logging that destroyed 31,000 km^2 of the forest area [19]. The consequences of wildfire are on the green forests, environments, economic losses, and casualties [12]. Hence, the EO satellite technology

has been improved to ensure the authorities can utilize aerial images to prevent wildfires from worsening.

Bird view could bring intelligence of ground variation by utilizing aircraft, unmanned aerial vehicles (UAV) and satellites. A study has been conducted to determine the merit and demerit of using these platforms [10, 20]. Finally, satellites are considered to have more advantages because of (1) large area and coverage, (2) frequent and repetitive coverage of area interest, (3) quantitative measurements of ground features, (4) relatively low-cost per unit area of coverage, and (5) ability to do semiautomated processing and analysis. The debates have been continuously involving satellite development and launching costs, but researchers have continued utilizing satellite data since the 1960s.

Historically, one of the oldest EO satellites for forest fire detection, and mapping is the NOAA AVHRR. The MODIS fire products began in 1978 with the launch of TIROS-N [10]. NOAA has launched the updated payload onboard satellites, with NOAA-15, 18, 19, and 20 orbiting Earth. Other space agencies are also sending the EO satellites in different orbits. For instance, Himawari-8 (JAXA), Meteosat-8 (ESA), and GOES-16 (NASA) in the GEO, while Sentinel-2 (ESA) and Landsat-8 (NASA) in the sun-synchronous orbit (SSO).

2.2.4 Earth observation satellites and CubeSats

Nowadays, the EO CubeSats have been launched to produce RS data besides the traditional EO satellites. Several studies discussed the limitations of these EO satellites [21]. The comparisons are on the size, weight, power generation, imaging payload type, and fire size detection. The traditional EO satellites win for most specifications, except that they only detect the smallest fire size of 2 km using a thermal sensor [22]. A constellation of CubeSats, such as Planet's Doves has the capability to capture in-meter fire size utilizing multispectral imaging payload [23]. Therefore, combining traditional EO satellites and CubeSats would benefit the authorities in acquiring knowledge of fire ecology. The advantages and disadvantages of the traditional EO satellites that are actively used for fire detection are summarized in Table 1.

Table 1. List of traditional EO satellite sensors used in fire ecology research.

Sensor(s)	Spatial resolution	Advantages	Disadvantages
IKONOS	0.8-4 m	High spatial resolution	Decommissioned, limited spectral resolution, high cost
Sentinel-2	10-60 m	Free, relatively high spatial and temporal resolution, multiple near infrared bands	Lack of canopy penetration
Landsat MSS, TM, ETM+, OLI	15-30 m	Free and easily accessible	Lack of canopy penetration, low temporal resolution
ASTER	15-90 m	Free and easily accessible, hyperspectral sensor, several short-wave infrared bands	Lack of canopy penetration, low temporal resolution
MODIS	250 m – 1 km	Free and easily accessible, high temporal resolution, large area analysis	Lack of canopy penetration, coarse spatial resolution limits analysis of smaller areas
GOES	1-4 km	High temporal resolution	Lack of canopy penetration, coarse spatial resolution limits fine scale analysis
Suomi NPP	1.5 km	Free and easily accessible	Lack of canopy penetration, low temporal resolution
MGS-SERIVI	3 km	Very high temporal resolution, large area of analysis	Lack of canopy penetration, coarse spatial resolution limits fine scale analysis

2.3 Machine learning and deep learning

Deep learning (DL) is a subfield of machine learning (ML) and a subfield of artificial intelligence (IA). Both DL and ML are similar in using a classifier between input and output. The dissimilarity is the approach where ML applied algorithms of a decision tree, k-nearest neighbour (kNN), or random forest (RF). The handcrafted feature extraction algorithms (SIFT and SURF) could be included before the ML classifier to increase the accuracy of output prediction. However, the DL operates by the feature extraction (edges, corners, and object parts) with the classification method. The DL algorithm could learn automatically from the training process based on experiences and examples. A dataset was labeled and categorized first before the training and validation operations. Increasing the amount of data could significantly increase the performance, which is true for the DL and not ML (Fig. 6) [24]. Hence, the DL approach overcomes the ML in terms of accuracy.

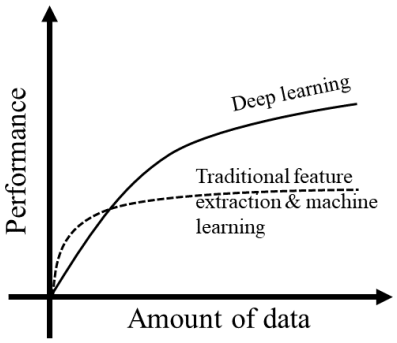


Figure 6. Deep learning against traditional feature extraction and machine learning.

The DL consists of three learning types; supervised, unsupervised, and semi-supervised. Supervised learning is teaching the algorithm with the labeled dataset; otherwise, it is unsupervised. Learning without a teacher is more challenging as the algorithm randomly learns and classifies the output with an unlabeled result. Ultimately, semi-supervised learning is a unique approach where the algorithm is taught and self-learning throughout the outcome. Thus, the best practice is to start with direct learning, supervised.

Computer vision's most common deep learning applications include image classification, detection, and segmentation. What is in the image? What is there in the image, and where is it? Which pixels belong to which object? These are questions related to deep learning utilization, respectively. The computer vision for EO satellite image data has been analyzed for various applications, especially wildfire images [25]. It could be a pixel depth of one (grayscale image) or three (colour image). The number of channels definition is crucial in the first step of creating a dataset.

2.3.1 Neural networks (NN)

Neural networks are the blocks in the DL systems. The architecture, node types, and training algorithms would define the network. Neural is an adjective form of neuron, and a network presents a graph-like structure. Therefore, an artificial neural network (ANN) is defined as a computation system that resembles or inspire by the neural connections in human nervous systems. A simple NN architecture could be expressed as inputs, hidden layers (weights), and output (figure). Each node in the figure performs simple computation, while each connection carries a signal. The hidden layers are labeled by weight to indicate whether the signal is amplified or diminished from one node to another. Thus, the connection weights are modified using a learning algorithm.

The inputs could be in vectors to systematically present the contents of an image, which are the raw pixel intensities of an image in the context of DL. Each input, x, is connected to a neuron through a weight vector, W. An activation function, f, could be applied by simply taking the weighted sum of inputs to generate output values. The most straightforward activation function is the step function implemented by the Perceptron algorithm.

The activation function of sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU) are other options in the DL network. The first two activation functions have a considerably smooth graph, but the gradients are ruined when the neurons become saturated because the delta of the gradient will be severely small. On the other hand, ReLu, known as a ramp function, was introduced by Hahnloser et al.

[26]. The advantages of the ReLU function are not saturable and exceedingly computationally efficient.

The most familiar NN architecture is the feedforward network. For instance, a network has three input nodes; the first hidden layer with two nodes, the second hidden layer with three nodes, and the output layer with two nodes, as illustrated in Fig. 7. The network only allowed nodes in the first layer to nodes in the second layer without backward or inter-layer connections. Each node in the input layer connected to each output node is called a fully-connected (FC) layer. Meanwhile, it is called a recurrent neural network in the case of feedback connections. Hence, the feedforward network is applied in the convolution neural networks (CNN) architecture for simplicity.

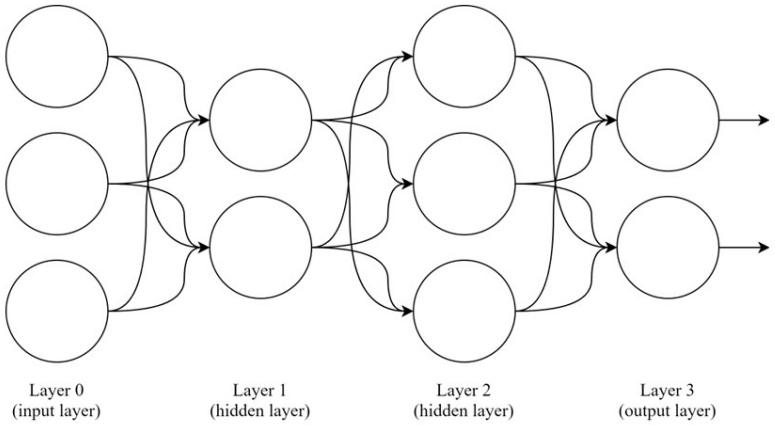


Figure 7. Example of 2 hidden layers of neural networks.

Convolution neural networks (CNN) are the most important building blocks in image processing that multiply two matrices followed by a sum of the elements. Blurring, smoothing, and sharpening an image are several examples of applying convolution. Each image pixel is convolved with a kernel, and the output is stored. For example, Laplacian or Sobel kernels are used to detect edge-like regions of an image [24]. The advantage of employing CNN is that it could overcome the challenge of manually hand-define kernels for various image processing operations.

The building of the CNN blocks consists of an input image that transforms it through several hidden layers. Convolutional filters (CONV), nonlinear activation functions (ReLU), pooling (POOL), backpropagation, batch normalization (BN), dropout (DO), and fully connected (FC) are examples of blocks that could be implemented in the CNN model. The CONV layer is the basis of building a CNN block comprising several filters K, receptive field size F, stride S, and amount of zero-padding P, where the output is shown in Equation 1. Later, an activation function like

ReLU is used after a CONV layer. The output of an activation layer is regularly similar to the input dimension.

$$W_{output} \times H_{output} \times D_{output} \quad (1)$$

where,

$W_{output} = ((W_{input} - F + 2P)/S) + 1,$

$H_{output} = ((H_{input} - F + 2P)/S) + 1,$

$D_{output} = K$

The methods to reduce the size of input volume are by implementing CONV layers with a stride > 1 and pooling (POOL) layers. The POOL layer has the main function of continuously reducing the spatial size of the input, the parameters and computation time, as well as controlling the overfitting. The parameters required for this layer are F (known as pool size) and S, which gave an output as shown in Equation 2.

$$W_{output} \times H_{output} \times D_{output} \quad (2)$$

where,

$W_{output} = ((W_{input} - F)/S) + 1,$

$H_{output} = ((H_{input} - F)/S) + 1,$

$D_{output} = D_{input}$

Batch normalization (BN) layers significantly decrease the number of epochs in the training process and increase the classification accuracy [27]. The placement of the BN layer is normally after the nonlinear activation layer. Furthermore, the dropout (DO) layer is a regularization form to avoid overfitting by raising the testing accuracy. For example, the DO layer is set with probability, p = 0.5, given a result of randomly dropping 50% of the connections. To generalize the network, DO will ensure that multiple and redundant nodes will remain active in similar inputs. The most common placing of the DO layer is between the fully connected (FC) layers. Finally, FC layers (one or two) are applied at the end of the network using a softmax classifier. The main function of FC layers is to fully connect all activations in the previous layer, which shows the final output for each class in probability value.

Generally, a deeper network is applied due to many labeled training data and challenging classification problems. Various network architectures have been introduced with considerably high accuracy results. Simple to complex CNN models were used in this study to feed with the new dataset created.

(1) ShallowNet

As the name represents, the architecture of ShallowNet consists of a few layers only. The network was introduced by Adrian Rosebrock, which can be

summarized as $INPUT \rightarrow CONV \rightarrow RELU \rightarrow FC$ (Fig. 8) [24]. The ShallowNet is a considerably simple CNN model that could achieve 90% classification accuracy with low computational cost.

ShallowNet	
Layer type	Output
Input	100 x 100 x 3
Conv	100 x 100 x 32
Act	100 x 100 x 32
FC	4
Softmax	4

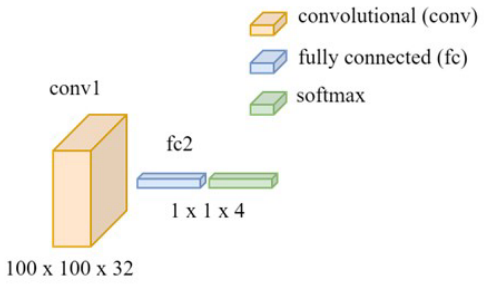


Figure 8. ShallowNet architecture.

(2) LeNet

The LeNet architecture was first introduced by Yann LeCun in 1998, mainly for optical character recognition (OCR) [28]. The network comprises $INPUT \rightarrow CONV \rightarrow RELU \rightarrow POOL \rightarrow CONV \rightarrow RELU \rightarrow POOL \rightarrow FC \rightarrow RELU \rightarrow FC$ (Fig. 9). In the current LeNet implementation, the ReLU layers were used instead of the original TANH layer. For instance, LeNet could obtain 98% classification accuracy using the MNIST dataset. Hence, this network was considered a 'shallow' deep learning model for image classification applications.

LeNet	
Layer type	Output
Input	100 x 100 x 3
Conv	100 x 100 x 20
Act	100 x 100 x 20
Pool	50 x 50 x 20
Conv	50 x 50 x 50
Act	50 x 50 x 50
Pool	25 x 25 x 50
FC	500
Act	500
FC	4
Softmax	4

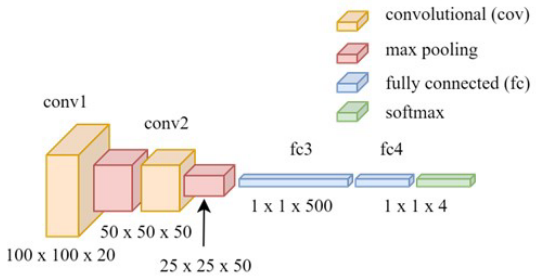


Figure 9. LeNet architecture.

(3) MiniVGGNet

Simonyan and Zisserman created the VGGNet in 2014, with higher depths (16 and 19 layers) [29]. The smaller and simpler version named MiniVGGNet was developed by Adrian Rosebrock, where the network can be summarized as

$INPUT \rightarrow CONV \rightarrow RELU \rightarrow CONV \rightarrow RELU \rightarrow POOL \rightarrow CONV \rightarrow RELU \rightarrow CONV \rightarrow RELU \rightarrow POOL \rightarrow FC \rightarrow RELU \rightarrow FC$ (Fig. 10) [24]. The BN and DO layers were also added to the network to decrease the overfitting as well as increase the classification accuracy. Overall, this network increases in depth compared to ShallowNet and LeNet.

MiniVGGNet			
Layer type	Output	Layer type	Output
Input	100 x 100 x 3	Conv	50 x 50 x 64
Conv	100 x 100 x 32	Act	50 x 50 x 64
Act	100 x 100 x 32	BN	50 x 50 x 64
BN	100 x 100 x 32	Pool	25 x 25 x 64
Conv	100 x 100 x 32	Dropout	25 x 25 x 64
Act	100 x 100 x 32	FC	512
BN	100 x 100 x 32	Act	512
Pool	50 x 50 x 32	BN	512
Dropout	50 x 50 x 32	Dropout	512
Conv	50 x 50 x 64	FC	4
Act	50 x 50 x 64	Softmax	4
BN	50 x 50 x 64		

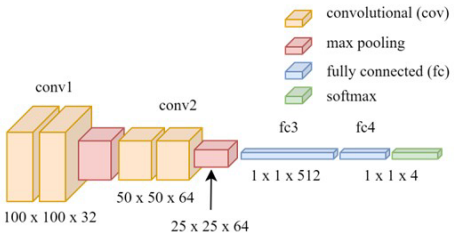


Figure 10. MiniVGGNet architecture.

(4) AlexNet

The original work that introduced the AlexNet architecture was Krizhevsky et al. in 2012 [30]. The network applies five CONV layers. The architecture consists of $INPUT \rightarrow CONV \rightarrow RELU \rightarrow POOL \rightarrow CONV \rightarrow RELU \rightarrow POOL \rightarrow CONV \rightarrow RELU \rightarrow CONV \rightarrow RELU \rightarrow CONV \rightarrow RELU \rightarrow POOL \rightarrow FC \rightarrow RELU \rightarrow FC \rightarrow RELU \rightarrow FC$ (Fig. 11). However, the pre-trained model generated using this network was considerably large.

AlexNet					
Layer type	Output	Layer type	Output	Layer type	Output
Input	100 x 100 x 3	BN	5 x 5 x 384	FC	4096
Conv	25 x 25 x 96	Conv	5 x 5 x 384	Act	4096
Act	25 x 25 x 96	Act	5 x 5 x 384	BN	4096
BN	25 x 25 x 96	BN	5 x 5 x 384	Dropout	4096
Pool	12 x 12 x 96	Conv	5 x 5 x 256	FC	4
Dropout	12 x 12 x 96	Act	5 x 5 x 256	Softmax	4
Conv	12 x 12 x 256	BN	5 x 5 x 256		
Act	12 x 12 x 256	Pool	2 x 2 x 256		
BN	12 x 12 x 256	Dropout	2 x 2 x 256		
Pool	5 x 5 x 256	FC	4096		
Dropout	5 x 5 x 256	Act	4096		
Conv	5 x 5 x 384	BN	4096		
Act	5 x 5 x 384	Dropout	4096		

Figure 11. AlexNet architecture.

(5) MiniGoogLeNet

Szegedy et al. introduced the GoogLeNet architecture to reduce the network size while increasing the depth. They also implemented a technique of a network in a network (or micro-architecture) for the overall architecture. In addition, a simpler version of GoogLeNet named MiniGoogLeNet (also known as Miniception) was developed by Chiyuan Zhang in 2017 [31]. The network can be summarized as: $INPUT \rightarrow CONV \rightarrow Inception\ module \rightarrow Inception\ module \rightarrow Downsample\ module \rightarrow Inception\ module \rightarrow Inception\ module \rightarrow Inception\ module \rightarrow Inception\ module \rightarrow Downsample\ module \rightarrow Inception\ module \rightarrow Inception\ module \rightarrow POOL \rightarrow FC$. The inception module merges two CONV layers, while the downsample module combines CONV and POOL layers. Therefore, the MiniGoggleNet model could obtain more than 95% accuracy with a small network size.

(6) ResNet

The extremely deep network of CNN is the ResNet, which has more than 50 layers. The architecture was introduced by He et al. in 2015 and won first place in ILSVRC 2015 challenges [32]. They also updated the architecture by implementing a new residual module concept, which is substantially deeper than the previous network. Finally, this model was the deepest network utilized in this study and the memory was limited during the training process.

In addition to the existing DL models, other parameters, such as pre-processors, data augmentation, optimizers, learning rate, and loss function are important for the model tuning. The image pre-processors could be simply re-sized fixed width and height (W x H), mean, patch or crop pre-processing. An optional method is adding the data augmentation into the model by shifting the W x H, flipping horizontal/vertical, rotational, shear or zoom to achieve higher classification accuracy. Furthermore, the available optimizers are stochastic gradient descent (SGD), Momentum, Adam, and RMSprop. A study has been conducted to compare the most effective optimizers used for image classification, concluding that the optimization method depends on the problem [33].

Moreover, the learning rate defines the step size. The objectives were to reduce overfitting and to obtain higher classification accuracy. The smaller the learning rate, the smaller the weight will be updated. Based on the loss and accuracy of the training/validation graph, the spikes are related to the learning rate. A learning rate scheduler could be implemented in the network to reduce the loss and the training time.

Lastly, the loss functions were binary or categorical cross-entropy. In the case of more than two labels, categorical cross-entropy was the appropriate option. The function determined how well the predicted class labels agreed with the ground-truth labels. The higher the level of agreement, the lower the loss, which means increasing the classification accuracy. Thus, the main goal was to minimize the loss function.

2.4 Single-board-computer (SBC)

2.4.1 Commercial-off-the-shelf (COTS)

Single-board-computer (SBC) or computer-on-module (COM) is a miniaturized and compact computer that consists of important embedded components such as flash memory, random access memory (RAM), central processing unit (CPU), and graphical processing unit (GPU). The tiny computer is available in various specifications at a reasonable price. In this study, the SBCs that work with deep learning environments mainly have been considered, which is the operating system (OS). An open OS is most likely to be selected as it could install the image processing libraries. For instance, Linux has been used in multiple onboard spacecraft as well as on the launch vehicle processors [34]. Therefore, the availability of source code and tools development of the OS with space use heritage is important in choosing the SBC.

The prominent SBCs for deep learning environments are Raspberry Pi, Nvidia Jetson, Beagle bone, Radxa, and Spresense development boards. A market study should be conducted to identify the most viable CubeSat platform, considering the limited requirements and space environment tolerance. The Nvidia Jetson has an advantage over the other SBCs because of GPU availability. An additional accelerator device such as Google Coral TPU (Tensor Processing Unit) and Intel Movidius VPU (Vision Processing Unit) enables compute-efficient deep learning inference at excellent performance but could consume additional power onboard CubeSat. These accelerators have space heritage that is optional to integrate with the SBC. Overall, the comparison of SBCs is summarized in Table 2.

Table 2. List of single-board-computers available for deep learning.

Board	CPU	GPU	USB	Ethernet	Flash memory	RAM	Size [mm]	Power [W]	Space heritage
RPi CM3+	BCM 2837B0	No	Yes	No	32 GB	1 GB	67.6 x 31.0	1.9	Yes
RPi CM4	BCM 2711	No	Yes	Yes	32 GB	8 GB	55.0 x 40.0	2.7	No
Jetson Nano	Quad-core ARM A57	Yes	Yes	Yes	16 GB	4 GB	69.6 x 45.0	5.0	No
Jetson TX2i	Quad-core ARM A57	Yes	Yes	Yes	32 GB	8 GB	87.0 x 50.0	10.0	Yes
BeagleBone black	AM3358	No	Yes	Yes	4 GB	512 MB	86.4 x 54.6	1.5	No
Radxa Zero	S905Y2	No	No	No	32 GB	4 GB	65.0 x 30.0	3.3	No
Spresense	ARM-M4F x 6 cores	No	No	No	8 MB	1.5 MB	50.0 x 20.6	0.3	Yes

2.4.2 Available AI boards for CubeSats

CubeSat-based companies are offering a readily developed board build-in with an AI environment. A Russian private company named SPUTNIX LLC was established in 2011, providing cost-effective solutions for microsatellite technology. OBC SXC-MB-04-RPI is a product that could integrate with RPi CM3 or CM3+ [35]. The board provides various interfaces such as UART, SPI, I2C, USB, and Ethernet. The board size and power consumption are within the 1U CubeSat platform. Ultimately, it has a space heritage where the CubeSat developer has a high-confident that would work in space.

On top of that, an AI miniaturized system named S-A1760 Venus was developed by Aitech company from the USA [36]. The company was founded in 1983 that provides commercial and military embedded computing solutions. The S-A1760 is aluminium shielded and integrated with NVIDIA Jetson TX2i that qualified for lower orbit space applications. The product provides additional interfaces and a mini SATA SSD for optional requests. The system consumed 1.5U space and 8 W power utilization. Overall, the S-A1760 datasheet does not mention space heritage but has been tested vigorously in the space environment tests.

A New Space company based in Poland, KP Labs, has developed a high-performance data processing unit for AI applications called Leopard [37]. The company offers several other products for space exploration by advancing autonomous spacecraft operations. The Leopard is embedded with quad-core ARM A53 and FPGA supporting CAN, LVDS, SPI, UART, and Ethernet interfaces. The product is equipped with 16 GB of flash-based data storage and 16 GB of RAM. Furthermore, the board fits in the 1U CubeSat, but the power consumption is considerably high (7.5 W), which is not feasible for a 1U platform. The Leopard has been used for Intuition-1 6U CubeSat, which is planned to be launched at the end of 2022.

2.5 AI CubeSats

The high-performance computing tools have made it difficult for CubeSat makers to execute image processing onboard in space. The CP8 IPEX 1U CubeSat was the first AI CubeSat that successfully demonstrated machine learning of the random forest (RF) model in orbit [38]. The CubeSat was orbiting in LEO from December 2013 to January 2015 with a mission to deploy onboard cloud detection. Five units of OV3642 cameras (200 m resolution) and a computer-on-module of Gumstix Earth Storm were integrated to classify 4-pixel categories (cloud, outer space, haze/limb, and surface). The IPEX had processed over 5000 images using the onboard classifier, which consumed less than 1 W of power and took about 29 sec to classify. Several downlinked IPEX images with the classification results were reported [38]. Thus, the successful mission by CalPoly and NASA JPL proved the feasibility of ML onboard CubeSat.

The second CubeSat equipped with AI onboard processing is a constellation of 2 Phi-Sat-1 6U CubeSat that was launched in September 2020 and is currently orbiting in LEO. The ESA mission was equipped with a hyperspectral camera to classify cloud-covered images [39]. The CubeSat uses a deep CNN model, which could classify images at about 325 msec. An Intel Movidius VPU was integrated to classify images rapidly that consumed about 1.8 W of power. The CubeSat managed to capture several images with different cloud types and apply DL onboard, giving the probability of cloud-covered on the captured images [39]. Hence, the Phi-Sat-1 CubeSat enhanced and verified the CubeSat capability of running a complex deep learning model onboard.

The latest successful AI CubeSat operated in orbit is the RSP-01 1U CubeSat (known as Selfie-sh). The CubeSat was developed by a private Japanese company named Rymansat that was launched in February 2021 and was deployed from the ISS. The main mission was to take a ‘selfie’ with Earth by extending its telescopic arm in space, while CNN's sub-mission of conducting image recognition and chatbot function [40]. Removing blur, noise, and sunlight were executed automatically using the RPi Zero onboard CubeSat. Overall, the missions were well-executed and confirmed the feasibility of CNN onboard the 1U platform.

Autonomous CubeSat projects have been proposed for future missions. Several AI CubeSats are planned to be launched in the end of 2022 such as Ke Ao (1U) [41], QlevEr-Sat (2U) [42], Intuition-1 (6U) [43], MANTIS (12U) [44], Hyperion 1 [45], and Skyris [46]. The previous successful missions of CP8 IPEX, Phi-Sat-1, and RSP-01 became practical examples of how machine learning and deep learning could be implemented onboard CubeSat with noticeable limitations. Therefore, the state-of-the-art image processing onboard has been verified, and more land applications should be investigated further.

CHAPTER 3
RESEARCH METHODOLOGY

3.1 Overview

This chapter explains the research methodology for developing the CubeSat platform, wildfire database, image dataset, CNN algorithm, and wildfire graphical user interface (GUI). The study started with the literature review in Chapter 2, which is important to determine the research field and identify the research novelty. Several studies have been published to investigate the feasibility of implementing CNN onboard CubeSats, but no such studies applied to classifying wildfire images. The components are illustrated in Fig. 12.

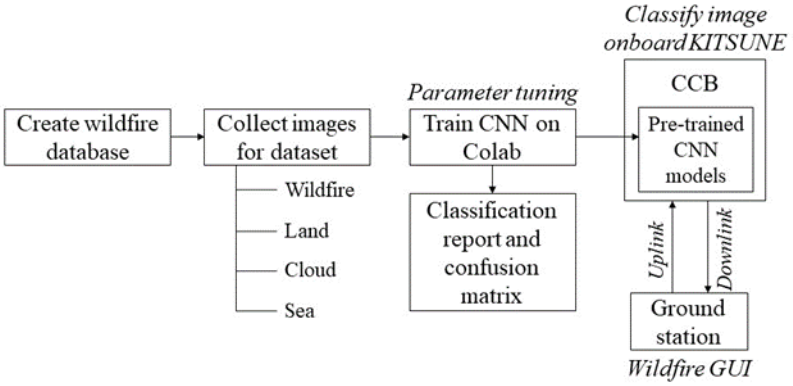


Figure 12. Flowchart of the study methods.

3.2 KITSUNE CubeSat

3.2.1 Overview

KITSUNE project collaborates between international academic institutions and the private sector in Japan: Kyushu Institute of Technology (Kyutech), Harada Seiki Co. Ltd. (HSK), and Addnics Corp. The 'fox' (in Japanese) utilized a 6U CubeSat platform, where the design and development were conducted in Kyutech. The team member comes from 18 multi-nation who are mostly postgraduate students. Furthermore, the development timeline started in September 2019 and deployed from the Japan Aerospace Exploration Agency (JAXA) KIBO module to the low Earth orbit (LEO) on 24 March 2022. Currently, the KITSUNE passed the initial operation phase, and the missions were executed accordingly.

3.2.2 Missions

KITSUNE is an acronym of the mission and development objectives; Kyutech standardized bus, Imaging Technology System, Utilization of Networking, and Electron content measurement. The 6U CubeSat platform comprises a 3U imaging payload, 2U main

bus system, and 1U SPATIUM-2 (space precision atomic-clock timing utility mission). Four main mission objectives are introduced as follows:

1) Earth observation with a 5-m-class resolution of visible images
2) Development of a 2U Kyutech standard bus system
3) Downlink of a low-resolution 2 MP image from a secondary camera by C-band uplink
4) Demonstration of C-band communication with the main and mobile ground station (GS)

The task management was divided between the collaborators. HSK engineers and the Kyutech camera team were responsible for the imaging mission. The Kyutech bus system team solely developed the 2U main bus systems. The Addnics engineers and Kyutech communication team were in charge of the C-band missions (3 and 4). Amateur radio frequencies (UHF and C-band communications) controlled the main bus system. Meanwhile, the SPATIUM-2 mission was not discussed in this study as it is out of the research scope.

The concept behind KITSUNE is to capture artificial coloured patterns or figures inside a 100 m^2 space for entertainment and social reasons in the early phases of development. The main goal is to deliver a 5-m-class imaging service in line with the mission statement. The wildfire image classification is incorporated as a secondary goal to optimize the usefulness of the image payload. As a result, the following success criteria were developed to categorize the mission outcomes:

1) Minimal success with downlinking an out-of-focus image to the ground station
2) Full success with capturing any letters or characters done by a group of people within a 100 x 100 m^2 space
3) Extra success with capturing images with 5-m-class resolution (6 m/pixel) and correctly classifying wildfire images

The payload system requirements were introduced concerning the mission statements and objectives. To support the listed system requirements, the design and verification requirements are shown in Table 3.

Table 3. The camera system, design and verification requirements.

No	System requirements	Design requirements	Verification requirements
1	Fit inside 3U size	1) Lens diameter < 90 mm 2) CCB size ≤ 90 x 90 mm 3) Full length ≤ 327.5 mm	Ensure the physical size
2	Within the weight budget	Total weight ≤ 7 kg	Ensure the mass of the payload
3	CCB should withstand the space and environment	1) Temperature: -20 to +50°C 2) Radiation: 200 Gy 3) Vacuum condition	1) RPi communicate with OBC, camera sensor, and C-band 2) RPi interfaces work stable
4	The camera should withstand the space environment	1) Operation temperature: -20 to +55°C 2) Vacuum condition	Ensure the camera takes clear photos in the worst hot and cold cases
6	Capture patterns	1) GSD: < 6 m/pixel 2) Swath: 20 km	Ensure the payload specification
7	Retrieve good-quality images	1) JPG compression > 90% 2) RGB images of no false colours	1) Images visual check 2) Retrieval of PNG and JPG files
8	Power compatibility	Total energy < 10 Wh/ orbit	1) Payload consumes less than total available energy per orbit 2) Rush current < OCP setting
9	Prepare images for downlink	1) Store > 5 images in C-band flash memory 2) Send thumbnail to OBC over UART	1) Uplink command over UHF 2) Image transfer to C-band 3) Ensure the payload is ready

3.2.3 Camera controller board (CCB) design and fabrication

Kyutech bus system team has developed the bus systems for 1U CubeSat, BIRDS project [4]. The 2U main bus systems were developed for KITSUNE CubeSat. Onboard computer (OBC), electrical power system (EPS), attitude determination and control system (ADCS), and communication system (COM) were well-arranged horizontally on a backplane board (BPB) (Fig. 13). The main purpose of integrating the BPB was to avoid failure from the harness connections. The OBC of KITSUNE has a pair of PIC microcontrollers that served as a command and data handling (C&DH) and communication to transmit the beacon of the CubeSat. In addition, the EPS has the important task of supplying sufficient power to the other bus systems and payload. It consists of 34 solar cells attached to each axis of the solar board, except the -Z-axis and the 14 W (maximum generation) of rechargeable batteries. Furthermore, a commercial off-the-shelf (COTS) active control module named MAI-401 was implemented in the KITSUNE bus system. The module has reaction wheels and sun sensors integrated and connected to a GPS and magnetometer from the other board. The GPS was installed on the solar panel, while the magnetometer was connected with a 10 cm harness distance from the ADCS module to avoid the electromagnetic noise of the reaction wheels. Finally, the C-band and the ultra-high-frequency (UHF) transceivers were installed in the KITSUNE for uplink and

downlink purposes. The C-band transceiver was used to downlink the full image data captured by the payload, while the UHF sends uplink commands and downlinks the thumbnail images, telemetry, and continuous wave (CW) beacon. Therefore, the bus systems are crucial for the successful imaging mission as they are related and connected, as shown in Fig. 14.

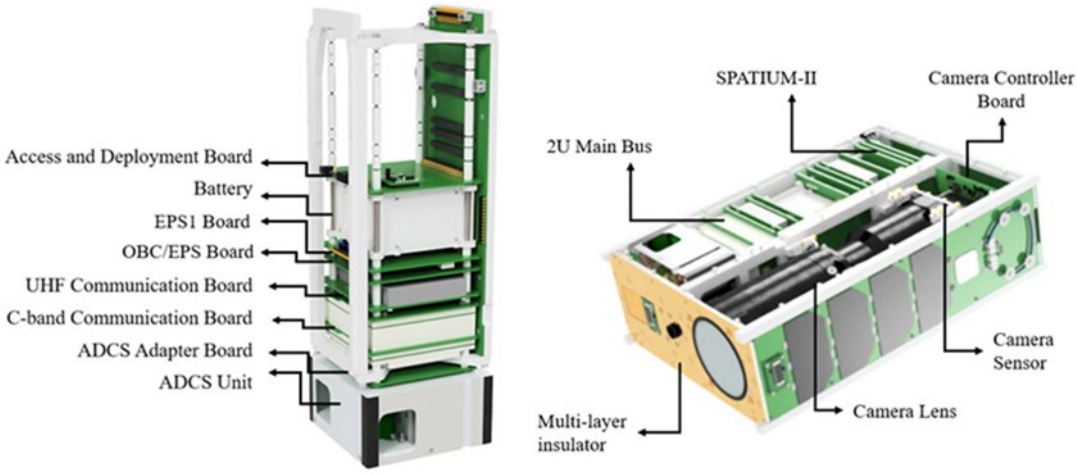


Figure 13. KITSUNE 2U main bus system configuration (left) and the satellite overview with 3U imaging payload (right).

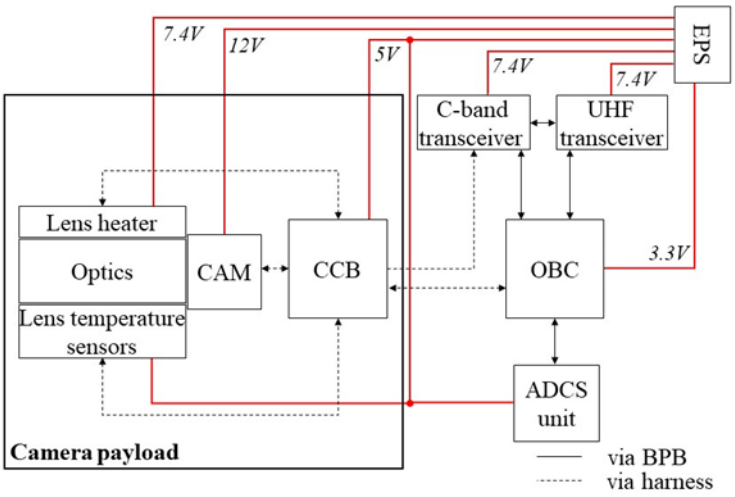


Figure 14. KITSUNE main mission subsystem block diagram.

Three important components in the 3U imaging payload were connected to the 2U main bus system via a harness. The camera controller board (CCB), camera sensor (CAM), and camera lens were well-designed to achieve the mission objectives and requirements (Table 4). A Raspberry Pi Compute Module 3+ (RPi CM3+) was installed on CCB, connecting and controlling a COTS CMOS camera sensor (31.4 MP). Meanwhile, a 300-mm custom-made RICOH lens could capture a full resolution of 6464 x 4852 pixels (Fig. 15). To achieve a 5-m resolution, the sensor pixels and optic focal length of this payload have been selected. The

ground spatial distance (GSD) could be calculated using Equation 3 to give the results of image output resolution and field of view (FOV):

$$GSD = \frac{h \times \tan(FOV_H)}{(\rho_H/2)} \quad (3)$$

where h is the satellite's altitude, FOV_H is the field of view for horizontal, and ρ_H is the pixel number of the image sensor for horizontal.

Table 4. Summary of imaging payload properties.

Components	**Details**
Sensor	
Number of pixels	31.4 MP
Sensor type	CMOS
Shutter method	Global shutter
Shutter speed	30 µs to 10 s
Interface	Ethernet
Data transmission speed	100 Mbps
Power supply	+12.0 V
Camera controller board	
Model	Customized board with Raspberry Pi Compute Module 3+
Operating system	GNU/Linux Ubuntu distribution version 18.04
CPU	ARMv8, 1.2 GHz
Memory	32 GB (eMMc), 1 GB (RAM)
Image capturing speed	0.42-8.75 frames per second (depending on image resolution)
Interface	Ethernet (sensor), USB (programming, UART (OBC and C-band board)
Power supply	+5.0 V
Optics	
Focal length	300 mm
Temperature control	Active control and multi-layer insulator
Heaters	Polyimide heaters
Heater power supply	7.4-8.4 V (unregulated power line)
Temperature sensors	Radial glass thermistor (G10K3976)

Figure 15. KITSUNE imaging payload.

The CCB was embedded with RPi CM3+ to control the CAM sensor through 100Mbps Ethernet and GPIO power cables. Three temperature sensors and two heaters attached to the lens were also connected to the bottom of CCB via harnesses (Fig. 16). The codes were saved in the RPi, interfacing with the OBC, C-Band, and ADB subsystems through 30-pin cables to the BPB. The parameter that can be set to the CAM sensor is shown in Table 5.

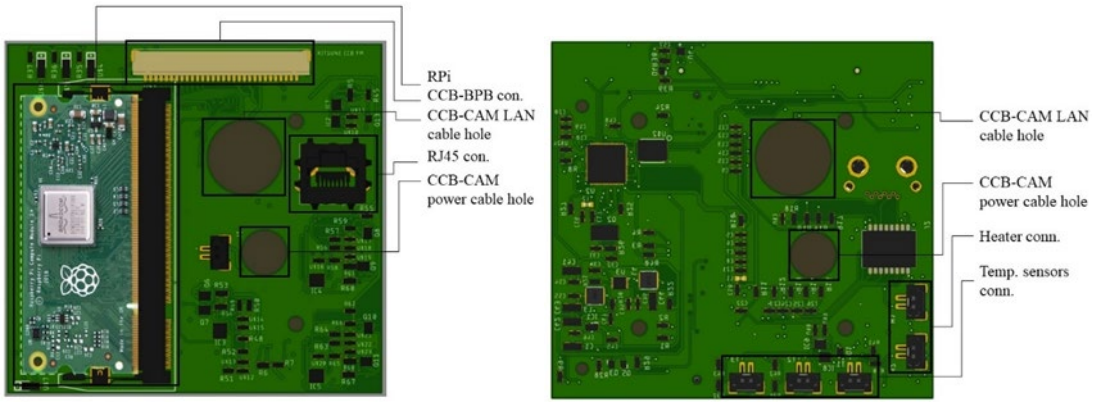


Figure 16. Camera Controller Board (CCB) design: (A) Raspberry Pi Compute Module 3+; (B) CCB-backplane board; (C) CCB-CAM sensor LAN cable hole; (D) RJ45 connector; (E) CCB-CAM sensor power cable hole; (F) temperature sensor connectors; (G) heater connectors.

Table 5. Summary of CAM sensor setting.

CAM sensor setting	**Details**
Resolution	6464 x 4852 (full) and 160 x 120 (thumbnails)
Image format	Jpeg and png
Automatic mode	Automatic gain and exposure
Gain	0-24 dB (analog) and 25-48 dB (digital)

Number of images	Up to 6 images/cmd
Compression quality	10 to 100%
Exposure time	47 µsec to 10 sec

A deep learning environment was installed on the RPi CM3+ with the necessary libraries. The operating system of Ubuntu 18.04 (aarch64-linux-gnu) was based on the ARM processor, which was matched with the RPi CPU. Due to memory limitation, a memory swapping method of 'zram' was installed. Other than OpenCV and TensorFlow, associated packages were also installed for deep learning development. The 60 steps of installation are listed as follows [47,48]:

1 $ sudo apt-get install zram-config
2 $ sudo nano /usr/bin/init-zram-swapping
3 *Edit line of 'mem=$(((totalmem / 2 / ${NRDEVICES}) * 1024 * 5))*
4 $ sudo reboot
5 $ sudo apt-get install build-essential cmake gcc g++ git unzip pkg-config
6 $ sudo apt-get install libjpeg-dev libpng-dev libtiff-dev
7 $ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev
8 $ sudo apt-get install libgtk2.0-dev libcanberra-gtk*
9 $ sudo apt-get install libxvidcore-dev libx264-dev
10 $ sudo apt-get install python3-dev python3-numpy python3-pip
11 $ sudo apt-get install python-dev python-numpy
12 $ sudo apt-get install libtbb2 libtbb-dev libdc1394-22-dev
13 $ sudo apt-get install libv4l-dev v4l-utils
14 $ sudo apt-get install libopenblas-dev libatlas-base-dev libblas-dev
15 $ sudo apt-get install liblapack-dev gfortran libhdf5-dev
16 $ sudo apt-get install libprotobuf-dev libgoogle-glog-dev libgflags-dev
17 $ sudo apt-get install libc-ares-dev libeigen3-dev
18 $ sudo apt-get install protobuf-compiler
19 $ sudo wget -O opencv.zip https://github.com/opencv/opencv/archive/4.5.0.zip
20 $ sudo wget -O opencv_contrib.zip
https://github.com/opencv/opencv_contrib/archive/4.5.0.zip
21 $ unzip opencv.zip
22 $ unzip opencv_contrib.zip

```
23 $ mv opencv-4.5.0 opencv
24 $ mv opencv_contrib-4.5.0 opencv_contrib
25 $ rm opencv.zip
26 $ rm opencv_contrib.zip
27 $ cmake -D CMAKE_BUILD_TYPE=RELEASE \
            -D CMAKE_INSTALL_PREFIX=/usr/local \
            -D
        OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/mo
        dules \
            -D ENABLE_NEON=ON \
            -D BUILD_TIFF=ON \
            -D WITH_FFMPEG=ON \
            -D WITH_GSTREAMER=ON \
            -D WITH_TBB=ON \
            -D BUILD_TBB=ON \
            -D BUILD_TESTS=OFF \
            -D WITH_EIGEN=OFF \
            -D WITH_V4L=ON \
            -D WITH_LIBV4L=ON \
            -D WITH_VTK=OFF \
            -D OPENCV_ENABLE_NONFREE=ON \
            -D INSTALL_C_EXAMPLES=OFF \
            -D INSTALL_PYTHON_EXAMPLES=OFF \
            -D BUILD_NEW_PYTHON_SUPPORT=ON \
            -D BUILD_opencv_python3=TRUE \
            -D OPENCV_GENERATE_PKGCONFIG=ON \
            -D BUILD_EXAMPLES=OFF ..
28 $ make -j4
29 $ sudo make install
30 $ sudo ldconfig
31 $ make clean
32 $ sudo apt-get update
33 $ sudo /etc/init.d/dphys-swapfile stop
34 $ sudo apt-get remove --purge dphys-swapfile
35 $ sudo apt-get install python-pip python3-pip
36 $ sudo pip uninstall tensorflow
37 $ sudo pip3 uninstall tensorflow
38 $ sudo -H pip3 install --upgrade setuptools
39 $ sudo -H pip3 install numpy==1.19.5
40 $ sudo -H pip3 install pybind11
```

```
$ sudo -H pip3 install Cython==0.29.21
$ sudo -H pip3 install h5py==2.10.0
$ sudo -H pip3 install gdown
$ gdown https://drive.google.com/uc?id=1X2mhbsG1p_RK_rKJrLCwh3cEA0Kwv6Zr
$ sudo -H pip3 install tensorflow-2.4.0-cp36-cp36m-linux_aarch64.whl
$ sudo rm -r /usr/local/lib/libtensorflow*
$ sudo rm -r /usr/local/include/tensorflow
$ sudo apt-get install wget curl libhdf5-dev libc-ares-dev libeigen3-dev
$ sudo apt-get install libatomic1 libatlas-base-dev zip unzip
$ gdown https://drive.google.com/uc?id=1c2-3HYH-HQdvczmRNKAKE89aPYTHziul
$ sudo tar -C /usr/local -xzf libtensorflow_cp36_64OS_2_4_0.tar.gz
$ sudo pip install opencv-contrib-python
$ sudo pip install scikit-image
$ sudo pip install pillow
$ sudo pip install imutils
$ sudo pip install scikit-learn
$ sudo pip install matplotlib
$ sudo pip install progressbar2
$ sudo pip install beautifulsoup4
$ sudo pip install pandas
```

The total time of installation could consume more than two hours depending on the clock frequency and the availability of the RAM. Finally, the installed libraries should be confirmed, as shown below:

```
$ python3
>>> import tensorflow as tf
>>> tf.__version__
‘2.4.0’
```

3.3 Wildfire database

An extensive study has been conducted to collect 715 wildfire cases from 2010 to 2020 around the world (Fig. 17). The online news, research papers, and fire department reports are reliable sources for collecting wildfire details [49-51]. Multiple sources on a particular wildfire

event are very helpful in generating a wildfire database, which consists of the dates, locations, coordinates, burned areas, causes, suppression costs, and casualties. Other wildfire databases are also being referred to and compared, such as the Monitoring Trends in Brun Severity (MTBS), National Interagency Fire Center (NIFC), Global Fire Emissions Database (GFED), Fire-CCI dataset, and TERN AusCover dataset [52-54]. The dissimilarities are the burned area size and only particular regions and countries. As stated in the research scope, the study covered global wildfire events with more than 0.05 km2 fire sizes. It is classified into alphabet values from A (> 0 km2) to L (> 4046 km2) based on the National Wildfire Coordinating Group (NWCG). Most of the wildfire cases recorded are from British Columbia (Canada), California (USA), Victoria and New South Wales (Australia). Frequent forest fires with complete wildfire reports at these locations make an excellent choice to include in the database. In addition, the smallest burned area is 0.05 km2 happened at Gyeongsanbukdo Pohang, South Korea, on 3 October 2013. Meanwhile, the largest fire that burned 338,000 km2 occurred in many parts of Australia between June 2019 and May 2020. The black summer bushfire was the worst Australia has recorded and the largest fire of the 21st century [55].

Figure 17. Wildfire events recorded in the database.

3.4 Dataset development

Influential criteria to achieve high accuracy of image classification is the solid dataset. The dataset could be binary (good or bad) or categorical (multi-labels). For instance, the algorithm aims to classify either wildfire image or not, which is easier to create the dataset of two labels only. Instead of having binary results, the dataset could be extended into several labels. This study mainly intends to classify wildfire images, but other labels such as cloud, land, and sea were also included. Therefore, having a 5-m resolution imaging payload on CubeSat, a mass of remote sensing applications was applicable to be classified.

Since the wildfire database has been developed, the satellite images could be retrieved to build a custom-made dataset. Doves (3-m resolution), Sentinel-2 (10-m resolution), Landsat-8 (30-m resolution), and MODIS (250-m resolution) image data have been downloaded accordingly. The images were acquired from the Sentinel Hub EO browser (free registered account) as well as the Planet Explorer (the Education and Research Standard registered account). The reason for acquiring multi-resolution image data is to have a robust dataset for KITSUNE and other CubeSats imaging missions within the resolution range. Furthermore, previous studies have shown the capability of utilizing multispectral bands, particularly for wildfire applications [18,56,57]. The results were excellent, with considerably high accuracy of fire detection. However, the approaches were limited to a particular spectral band of image data. In this study, the research scope utilizes the visible 3-band (RGB) images. Hence, only the visible data from the listed satellites have been utilized to match the imaging payload of KITSUNE.

3.5 Training dataset

After the dataset has been created, a DL algorithm could be trained properly. The four labels in the dataset were manually organized with 1500 images each, which were: (1) wildfire, (2) land, (3) cloud, and (4) sea. To compare the training accuracy, the dataset was divided into two types, 1000 and 1500 images/label. Furthermore, the configuration of the CNN models was defined in the training script. The training was run in Colab, which generated the pre-trained model in the '.hdf5' file, classification report, and the confusion matrix. Distinct parameters were tuned extensively, which were (1) input pixel, (2) data augmentation, (3) CNN models, (4) optimizers, and (5) learning rate, as illustrated in Fig. 18. Moreover, the dataset was split into training and testing sets, 75% and 25%, respectively. The process was written in the script and ran automatically and efficiently.

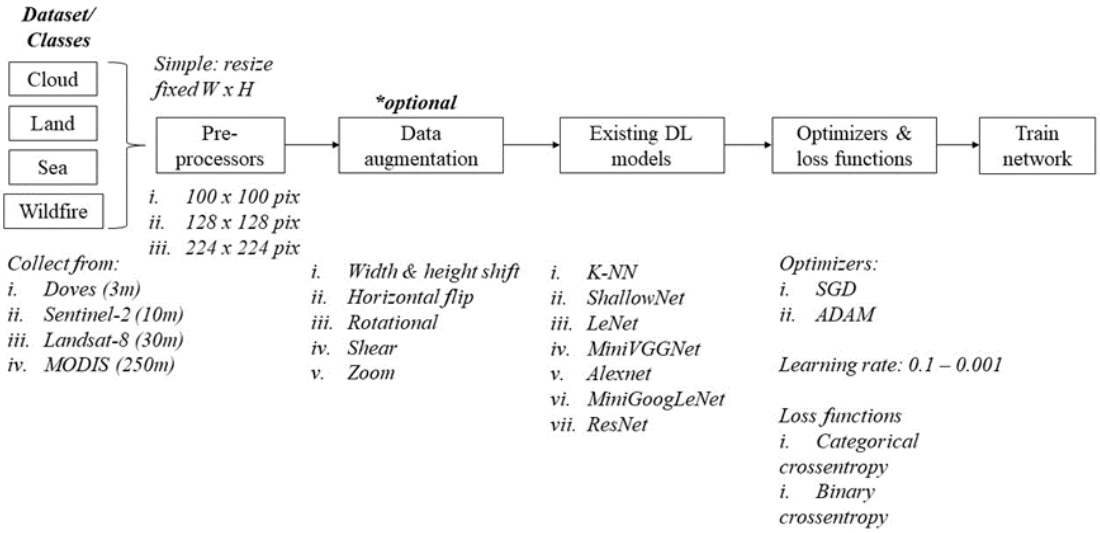


Figure 18. The CNN training algorithm pipeline.

The first parameter of tuning was the input pixel size. The dataset was pre-processed with a simple image resize of fixed width and height: 100 x 100, 128 x 128, and 224 x 224 pixels. Below the pixel size will reduce the valuable pixel of high-resolution data, while above the pixel size was not feasible due to the limited Colab memory. Secondly, data augmentation was implemented as it could increase classification accuracy. The method was shifting, flipping, rotating, shearing, and zooming the dataset. Despite an optional parameter, the rationale was that the satellite would capture the Earth's surface in different rotations and angles. Thirdly, the training dataset fed an ML model and six DL models. K-nearest neighbour (k-NN) was initially used to compare ML and DL models. Six CNN models: (1) ShallowNet, (2) LeNet, (3) MiniVGGNet, (4) AlexNet, (5) MiniGoogLeNet, and (6) ResNet were selected from low to a high number of layers. Finally, the optimizers used were stochastic gradient descent (SGD) and ADAM, while the learning rate was set between 0.1 and 0.001. Other researchers have suggested these tuning parameters and discussed them in Chapter 2. Furthermore, the loss function was only set to the categorical cross-entropy due to four labels in the dataset. The final output of these tuning parameters was the pre-trained model in the '.hdf5' file, which was then copied into the RPi CM3+ memory for further classification.

The performance of the CNN models depended on the number of correct and incorrect predictions. True positive (TP) referred to the pixels assigned as the target object in the ground truth and predicted result. If the pixels were identified as the target object in the prediction only, not in the ground truth, they are called false positive (FP). False negative (FN) and true negatives (TN) are vice versa from the FP and TP, respectively. These four parameters are important to illustrate the confusion matrix and calculate the performance report. Accuracy (A), precision (P), recall (R), and F1 score were the evaluation metrics that have been generally used to analyze the CNN networks thoroughly. The formula is shown in Equations (4)-(7), accordingly.

$$A = \frac{TP+TN}{(TP+TN+FP+FN)} \tag{4}$$

$$P = \frac{TP}{(TP+FP)} \tag{5}$$

$$R = \frac{TP}{(TP+FN)} \tag{6}$$

$$F1\ score = 2 \times \frac{P \times R}{P + R} \tag{7}$$

3.6 Classification test

The image classification test was conducted onboard the CubeSat. The script worked given two inputs: the image captured by the payload and the pre-trained CNN models. Separate two uplink commands (or schedule commands) have to transmit to the CubeSat for executing the classification script. The result was appended to the image captured, which will not damage

the image data retrieved on the GS. Nevertheless, it is also possible to downlink the classification result by only downloading the end of packet data for swift insight.

3.7 CoFFI graphical user interface (GUI)

To visualize the image data with analysis results, a graphical user interface (GUI) was created, as shown in Fig. 19. PySimpleGUI was chosen as the Python GUI framework. The GUI called classification of forest fire imagery (CoFFI) enables loading the raw captured image and running the six CNN models. Each model gave a different result in percentage, which depends on the accuracy of the pre-trained models.

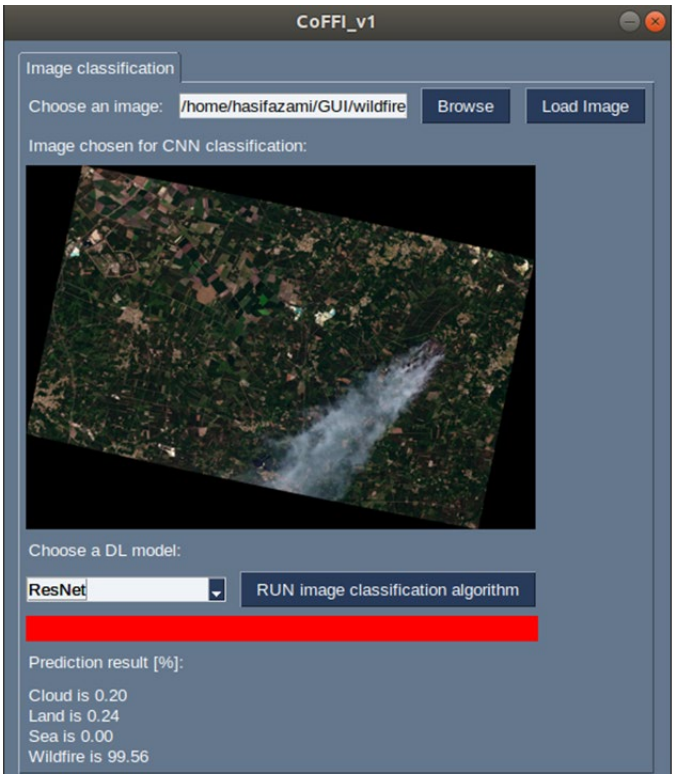


Figure 19. The CoFFI GUI window.

CHAPTER 4
RESULTS

4.1 Overview

This chapter shows the research results with descriptive analysis during the space environment tests (sun simulator, total ionization dose radiation, thermal vacuum, and vibration tests), long-duration operational, training and classification tests, and in-orbit data.

4.2 Space environment tests

Numerous functional tests (FT) were performed to establish the communications and interfaces between each subsystem and the imaging payload. To achieve a reliable result, the pre-, during, and post-FT of space environment tests should be conducted similarly to the mission satellite operation in orbit. The space environment tests consist of the sun simulator test, total ionization dose (TID) radiation test, thermal vacuum test (TVT), and vibration test (VT), while the long duration operational test (LDOT) was conducted afterwards. Hence, each test was significantly important for a successful mission as the CubeSat will orbit in a harsh space environment.

4.2.1 Sun simulator test

The camera sensor used in the KISUNE imaging payload was a complementary metal-oxide-semiconductor (CMOS) type. Normally, the sun simulator test is conducted for solar panels. However, in this study, the test aimed to study the effects of sunlight on the CAM sensor and verify the survival of the sensor facing directly towards the sun. The light intensity used for this test was 1367 W/m^2, which was measured by the pyranometer, as illustrated in Fig. 20. The exposure time was three minutes, increasing at 10-sec intervals. Even though the device under test (DUT) was not similar to the KITSUNE CAM sensor in terms of resolution, the pixel size, shutter type, and sensor manufacturer were identical. The main reason to use IMX296C (1.6 MP) instead of IMX342 (31.4 MP) was that the price is noticeably inexpensive.

Figure 20. Sun simulator test setup.

A thermocouple was placed behind the camera sensor to observe the temperature change during the sun simulator test. Before the test started, the distance between the sun simulator machine and the camera payload was measured using a reading from the pyranometer, which was 10 mV. The test procedures are listed as follows:

1) Take photos of a white background as the pre-functional test
2) Disconnect the CAM sensor from the RPi CM3+
3) Expose the CAM sensor to the sun simulator
4) Record the temperature reading using the digital multimeter
5) Turn off the sun simulator machine
6) Re-connect the CAM sensor to the RPi CM3+
7) Take photos of the white background again
8) Repeat steps 2 to 7 by increasing the exposure time to 10 s.

The black background was also captured to compare with the white colour. To have reliable results, the RGB images captured after exposure were converted to grayscale colour. Fig. 21 shows that the standard deviation value changed significantly after the CAM sensor was exposed to the sun simulator machine in less than a second, in addition to the temperature measured during the test. The intensity of grayscale images was analyzed through the histogram chart before, after a second, 180 sec, and six-day test. Fig. 22 shows that the intensity value shifted towards the dark region (to the left histogram) at 0.03% only after the CAM sensor was exposed to the sun simulator machine for 3 minutes. However, the CAM sensor recovered after the sixth day of the test showing the intensity value returned as before. Therefore, the sun simulator test verified the functionality of the CAM sensor, which was restored even after being exposed to the sun. The performance of the CNN algorithm was not investigated as it was unrelated to the test objective.

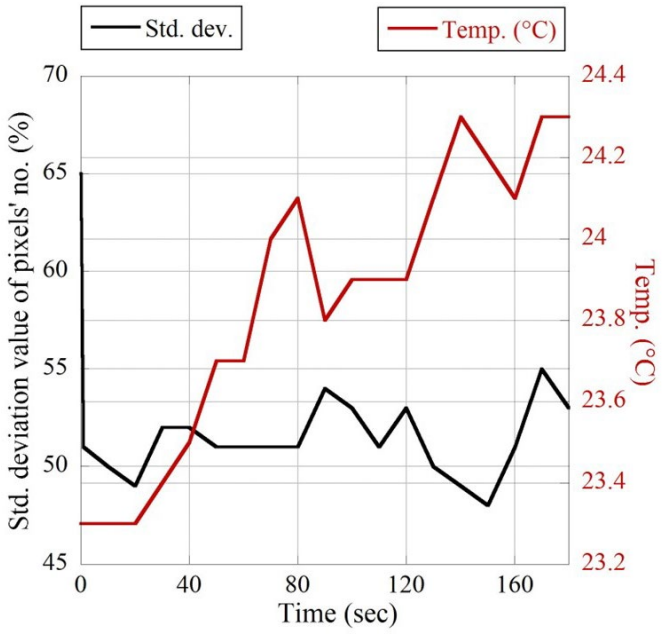


Figure 21. The standard deviation value of pixel number and sensor temperature reading during the sun simulator test.

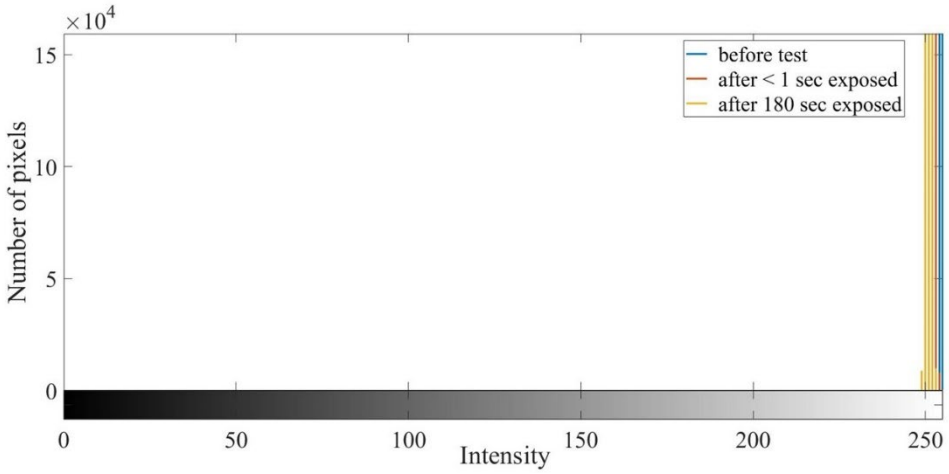


Figure 22. The number of pixels against the intensity before the test (blue), after less than 1 sec exposed (orange), and after 180 sec exposed (yellow).

4.2.2 Total ionization dose (TID) radiation test

The total ionization dose (TID) radiation test was the second space environment test. In space, the radiated particles would cause catastrophic damage to the electrical components of the satellites. Space radiation consists primarily of the ionizing radiation that exists in the form of high-energy and charged particles from the trapped radiation, galactic cosmic radiation (GCR), and solar particle events (SPE). The primary objective of the TID test was to demonstrate that the device under units (DUTs) remains operational and that the interface communication was stable in the radiated environment. The DUTs were RPi CM3+, RPi CM4, and Radxa Zero. The central processing unit (CPU) and the embedded multimedia card

(eMMC) were the vital components that needed to be observed continuously by executing the 'flash test' script. The additional DUTs were tested to acquire a comparison result. Moreover, the next objectives were to verify that the DL algorithm for training and classification tests could be executed, as well as to measure the power consumption (on RPi CM3+ only) during the TID test. The DUTs were directly exposed to a radiation source of Co-60 in the chamber facility at the Center for Accelerator and Beam Applied Science, Kyushu University, Japan, as shown in Fig. 23. The test consumed six hours, equivalent to 200 Gy simulated for two years in LEO. Several tests were also conducted by Toumbas [58] on RPi CM3 and by Slater et al. [59] on Jetson Nano GPU. The main differences between this study and previous studies were the type of DUTs, radiation source and radiated energy.

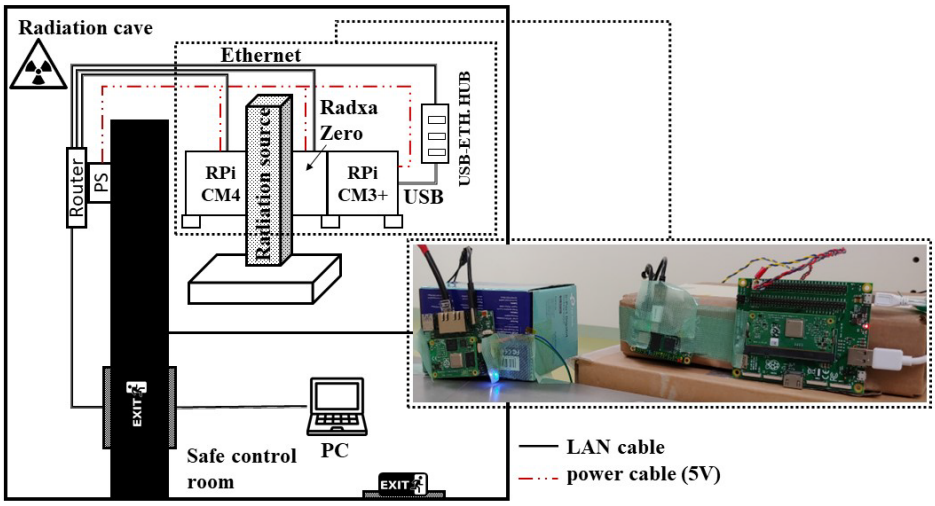


Figure 23. The total ionization dose (TID) radiation test setup.

The first objective was established by running a pseudo-random number generator named 'flash test' code. A known seed was utilized to produce files with equally spread byte values. Each file was seeded individually, resulting in files that were all distinct. The same test can be performed numerous times, and the test files will be identical each time. During the TID test, 75 files were created and compared to the same number of files made on the computer earlier. Later, the sha256 algorithm was used to hash each created file. This hash function had the benefit of providing a 256-bit result that was noticeably different, even if the change was only a single bit. Hash collisions across different files were also improbable, and none have been recorded for the sha256 technique. The time taken to create each batch of 15 files differed among the three DUTs: 32 min for RPi CM3+, 16 min for RPi CM4, and 17 min for Radxa Zero. After all 15 files were generated, a 30 min delay was added for manually running the training and classification algorithms. The three DUTs hashes were compared between the reference test data (before the TID test), the actual test data, and the post-functionality test data on the following TID testing day. The results showed no inconsistency for every 75 files for all

three DUTs. Nonetheless, the Radxa Zero found no communication after the flash test5 and did not recover until the test ended. Hence, the eMMC of the RPi CM3+ and CM4 could stay operationally stable for two years in orbit.

The important step of the TID test in this study was to observe the performance of training and classification algorithms using the ShallowNet model only for simplicity. Both scripts ran sequentially after flash test4 and flash test5. RPi CM3+ took 16 min (800 MB of memory) for the training test, while 36 sec (497 MB of memory) for the classification test. On the other hand, RPi CM4 consumed less time for the training and classification tests: 10 min (856 MB of memory) and 13 sec (517 MB of memory), respectively. Radxza zero attained some computational data before it failed (at the second training test), which was that 10 min (743 MB of memory) for the training test and 17 sec (429 MB of memory) for the classification test. Therefore, the memory of DUTs was consumed considerably similar to both execution codes, but the processing time was significantly distinctive. The RPi CM4 was the fastest in training and classification algorithms compared to RPi CM3+ and Radxa Zero due to considerably higher CPU clock speed, as summarized in Table 6.

Table 6. Summary of test results corresponding to the DUTs.

DUT	RPi CM3+	RPi CM4	Radxa Zero
Flash test	Δ Anomalies found after 4 hours TID test, but recovered	O No anomaly observed	X Anomaly after flash test5, cannot recovered
CNN training test	O Consumed 16 min	O Consumed 10 min	Δ Consumed 10 min, but failed at second time
CNN classification test	O Consumed 36 sec	O Consumed 12 sec	Δ Consumed 17 min, but failed at second time
Energy consumption (training/classification test)	O Less than 1 W.h	X Not tested	X Not tested
Memory consumption (training/classification test)	O 800 MB/ 497 MB	O 856 MB/ 517 MB	O 743 MB/ 429 MB

The second objective was verified by measuring the power consumption of RPi CM3+ only during the TID test. The flash test script was executed five times and two times for the CNN algorithms (training and classification tests) in a second of the time sampling. Fig. 24 shows the power reading throughout the test corresponding to the total dose exposed to the RPi CM3+. However, an anomaly was observed during the test, starting right before the flash test5 (at 4 h). Based on Fig. X, the RPi CM3+ consumed extra power without running any code. It revealed that a single event latch-up occurred at the time of 133 Gy being exposed to the DUT. The anomaly was recovered after executing the DL training test2 (at 4 h 55 min) as well as after rebooting (soft and hard resets) the RPi CM3+ at the end of the test. The issue was verified by observing the memory consumption and comparing the result with the pre-and post-functionality tests. Overall, the energy and memory consumed by the RPi CM3+ were

significantly similar between all functionality tests, about 1 W.h (800 MB) for the training test and 0.03 W.h (445 MB) for the classification test. Therefore, the mission design of turning on the RPi CM3+ only during the mission execution was acceptable as it could overcome the issue when the single event latch-up occurred.

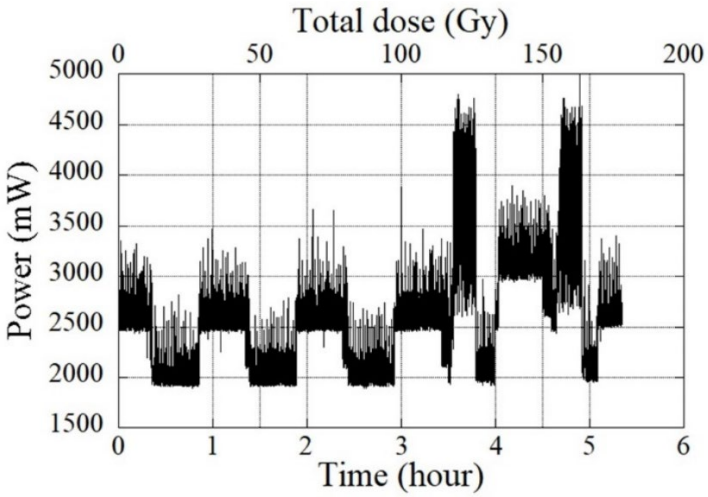


Figure 24. Power consumption of Raspberry Pi Compute Module 3+ during radiation test.

4.2.3 Thermal vacuum test (TVT)

The thermal vacuum test was conducted to simulate the satellite operation in extreme environments, the worst hot and cold in a vacuum condition. The KITSUNE payload has been tested in the unit system as well as in the satellite system levels. The objectives were to verify the functionality of the imaging payload and the feasibility of executing the CNN algorithms. The external solar panel boards of the KITSUNE flight model (FM) experienced two cycles between -30 and +50 degrees Celsius. The temperature range was based on the BIRDS project [4] and ISO 19683-design qualification and acceptance tests of small spacecraft and units. The CubeSat was precisely placed inside the thermal chamber, and a collimator lens was arranged at the outer chamber to test the image focus by pointing towards the pattern lens through the chamber window. An additional light source was also positioned facing the target lens to have a clear image captured during the test, as illustrated in Fig. 25. Hence, the TVT was a crucial test for the satellite developer to confidently demonstrate a workable mission in space.

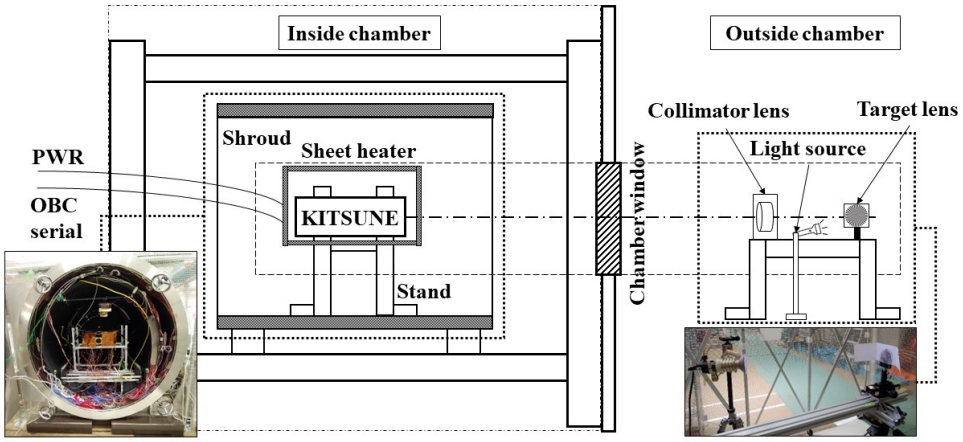


Figure 25. The thermal vacuum test setup.

The temperature readings of the camera controller board (CCB), camera sensor (CAM), and average external panels were plotted in Fig. 26. The panels' temperature verified that KITSUNE experienced two thermal cycles with a designated temperature range. The results also showed that the temperature corresponding to the CCB and CAM, and average external panels increased significantly in a short period during the FT of the imaging mission. However, during FT 5 and 6, the temperature reading of CCB was raised sharply. The reason was that the CNN model was trained for several minutes and correlated with the power consumption of the CCB. Thus, FT 5 and 6 were highlighted in this study for the deep learning execution onboard CubeSat, while the rest of the FTs were tested for other KITSUNE missions.

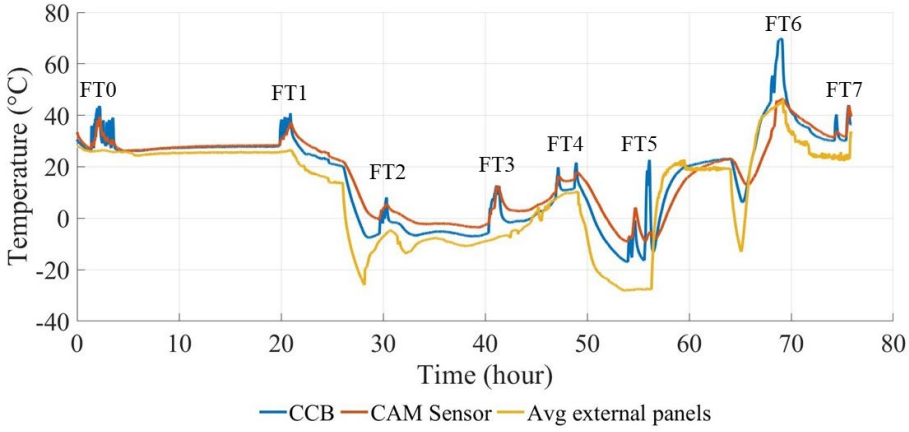


Figure 26. The functionality tests (FTs) correspond to the thermal cycles.

The first objective was carried out by executing the imaging mission in each FT point. Image focus was analyzed using several normalized focus measure operators: Gradient-based, Laplacian-based, wavelet-based, statistic-based, and discrete cosine transform-based. As discussed by Pertuz et al., the Gradient-based and Laplacian-based were the best measure operator for the image edges [60]. Five operators have been chosen in this study: (1) Gaussian

derivative, (2) Tenengrad variance, (3) modified Laplacian, (4) diagonal Laplacian, and (5) variance of Laplacian, which were the top in the ranking according to the Sony camera relative quality. Ultimately, the diagonal Laplacian operator has been chosen as the significantly suitable approach due to the consistency in TVT and VT results. Based on Fig. 27, the highest focus measured was at FT4, 100% focus at +12.5 degrees Celsius in vacuum conditions, with a number closer to 1.0 indicating better focus. Meanwhile, the FT0 image was the lowest focus measure, which was 74.3% at atmospheric room temperature. The camera focus was significantly affected by the pressure level (atmospheric or vacuum) and temperature differences, as shown in Figure 9, where FT1 (in vacuum condition) was higher in focus measured than FT0 (in atmospheric condition), increased by 6.3%. In the same vacuum condition, FT2 showed better focus, which increased by 17.8% from the FT1 result because of the different lens temperatures measured. Therefore, the KITSUNE FM camera payload could have an optimal focus (above 95.0% focus measured) between -1.8 and +12.5 degree Celsius, based on the TVT results in the vacuum conditions.

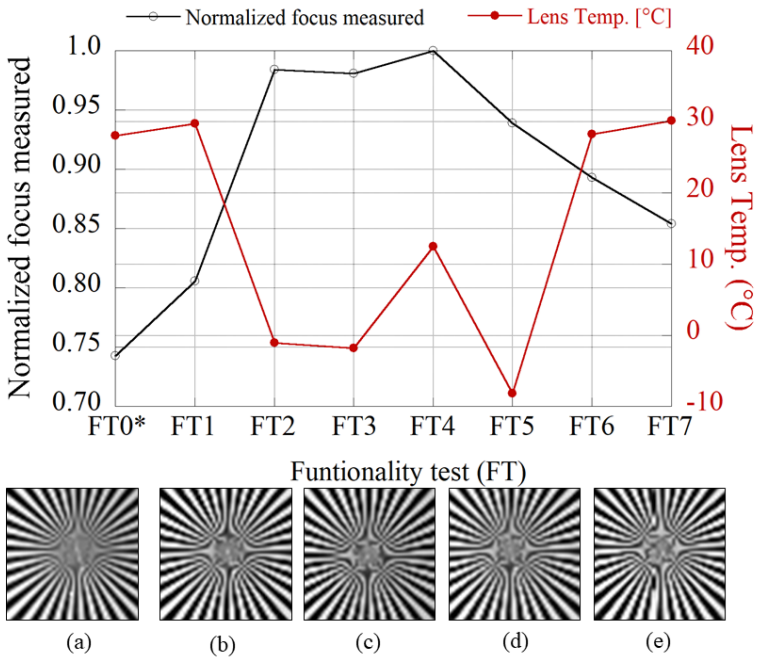


Figure 27. Result of normalized focus measured on the image of the target lens captured during each functionality test (FT) with the lens temperature measured during (a) FT0*(atmospheric condition), (b) FT1, (c) FT2, (d) FT3 and (e) FT4.

The TVT was also conducted to verify the feasibility of a deep learning algorithm in orbit. The power consumption readings at FT 5 and 6 were plotted: cold soak (blue line) and hot soak (orange line, respectively with the vertical dash lines indicating the start and end of execution time (Fig. 28). The training test during FT 5 consumed a shorter execution time than FT 6, 734 sec (667 MB) and 1664 sec (675 MB), respectively. On the other hand, the classification test of FT 5 and 6 took 26 sec (284 MB) and 36 sec (301 MB), respectively. Fig.

28 also shows the relationship between the power consumption and the temperature readings in KITSUNE FM during the training and classification test execution. The training test consumed about 1.12 W.h during the cold and 2.08 W.h during the hot soak environments. The CPU performance of RPi CM3+ will be affected during the worst hot condition. Overall, the image classification could be feasibly executed due to the shorter computational time, which will only be implemented in the KITSUNE mission.

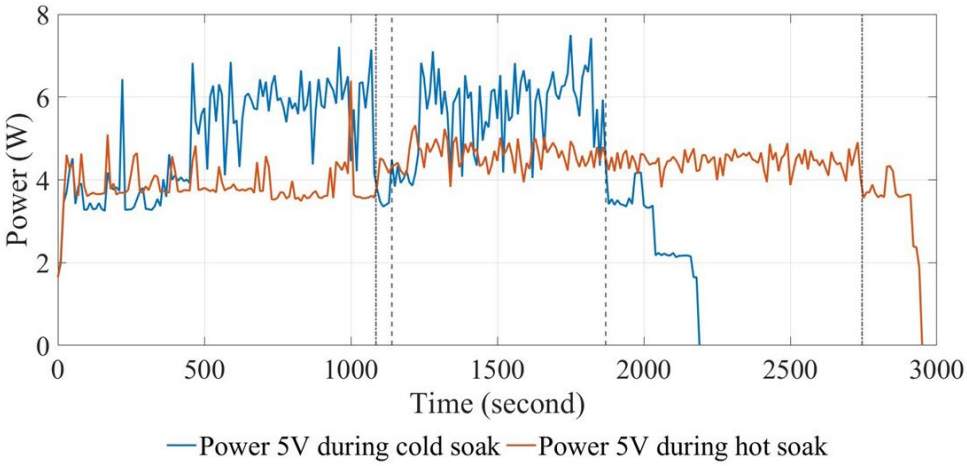


Figure 28. Power consumption of Raspberry Pi Compute Module 3+ (5 V) during the cold soak (blue) and hot soak (orange).

4.2.4 Vibration test (VT)

The last space environment test was the vibration test aimed to imitate the CubeSat during the rocket launching. For KITSUNE, the acceleration profile used was the Orbital Cygnus vehicle as planned during the development process. Three-axis were tested in the qualification test (QT) level using 5.77 Grms of random vibration for 120 seconds. To verify that the camera payload was in good shape, two methods were conducted by inspecting the lens and comparing the images captured between the pre-and post-functionality tests. Each axis change should be examined carefully for cracks or shifted screws (Fig. 29). A similar method was also used to check the focus changes on the captured images between pre-FT, after YZ-axis FT, and post-FT. The result showed no significant changed in the image focus of about ± 3%. Moreover, the CCB was working well without any errors with the communication interface. Overall, the VT was necessary for the successful mission and the safety of the launch vehicle.

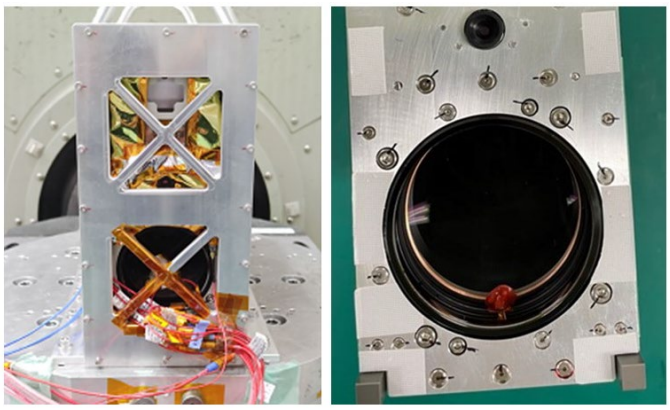


Figure 29. Vibration test on KITSUNE flight model at Z-axis (left) and lens and screws checked (right).

4.3 Long duration operational test (LDOT)

After the space environment tests were completed, the long-duration operational test (LDOT) was carried out. The LDOT was significantly important to validate the software design of the camera payload as well as the DL algorithm. Each mission scenario was executed with multiple uplink commands to the CubeSat in real wireless UHF and C-band communications. The KITSUNE FM was stationed in the clean room with the ground station radio and terminal node controller (TNC). Meanwhile, a serial cable was only connected to the CubeSat for monitoring and debugging purposes, as shown in Fig. 30. The test was conducted over several days to find software bugs and resolve them. The important point was to estimate the accurate timing of mission execution. The leading question was to verify whether the camera mission and DL algorithm could be executed sequentially in a pass window or later pass. In addition, the power consumption during the LDOT was observed and plotted for four passes of KITSUNE in orbit. The CubeSat was also connected to a power supply where battery charging and discharging conditions were programmed to demonstrate the mission feasibility.

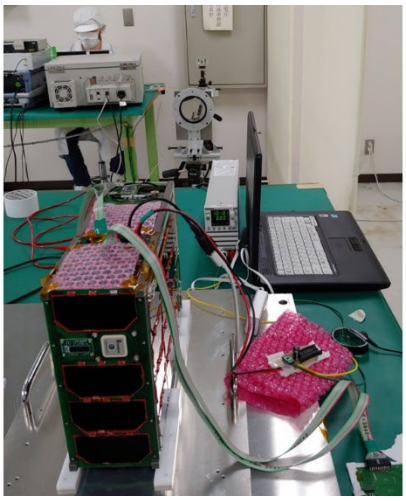


Figure 30. The long-duration operational test of the KITSUNE flight model in the clean room.

The entire imaging mission was demonstrated in four passes. The initial pass was programmed to take six photos at the desired time and coordinate. The ADCS was also set to the nadir-pointing mode an hour before the camera sensor was turned on. Fig. 31a shows that the power consumption during the first pass consists of image capturing and copying the data to the OBC shared flash memory. The 12 V power line (blue line) was designated for the camera sensor, the CCB used the 5 V line (orange line), and the overall power (yellow line), which indicated the total consumption, including from the main bus subsystems. The result showed that 2.90 Wh of the overall energy was consumed at this pass. In addition, the CCB was programmed to convert the PNG images into JPG format files as well as to generate the thumbnails. Furthermore, the mission scenario was to downlink the thumbnails through UHF communication before retrieving the complete targeted PNG/JPG image data via the C-band, conducted in the second pass. Referring to Fig. 31b, the commands were sent as follows:

1) Uplink command to get the last address of image data through UHF (green line) at 104 sec
2) Downlink six thumbnails through UHF between 160 and 192 sec
3) Downlink complete JPG data through C-band in real-time (red line) between 507 and 637 sec
4) Downlink corresponding ADCS housekeeping data from the C-band between 770 and 845 sec
5) Finally, copy the PNG image to the C-band flash memory between 924 and 2087 sec

The overall energy consumed during the second pass was 2.81 Wh. Moreover, the third pass showed the downlink of the PNG image data from the C-band flash memory (Fig. 31c). The short peak at the 39th second was due to the uplink command. Overall, the total energy consumed for the 512-sec mission was 2.33 Wh within the KITSUNE power generation.

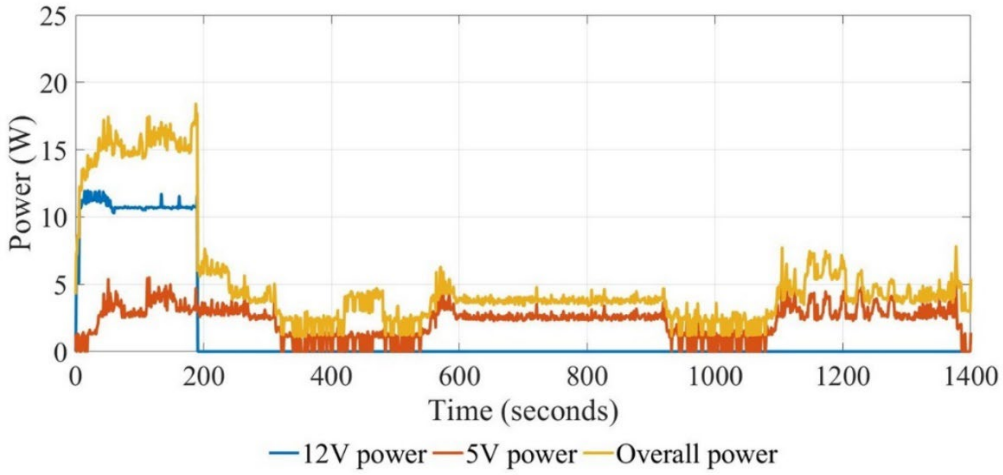


Figure 31a. Power consumption of KITSUNE camera capture mission (first pass).

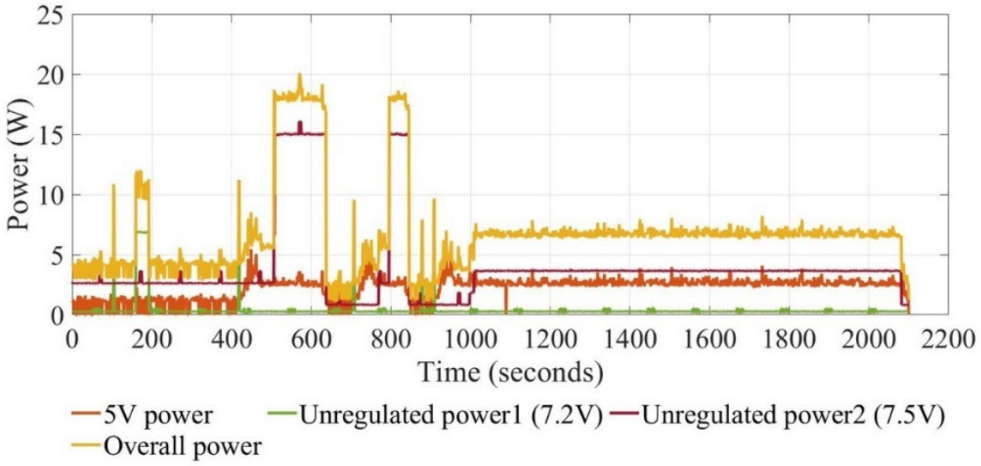


Figure 31b. Power consumption of KITSUNE downlink JPG image (second pass).

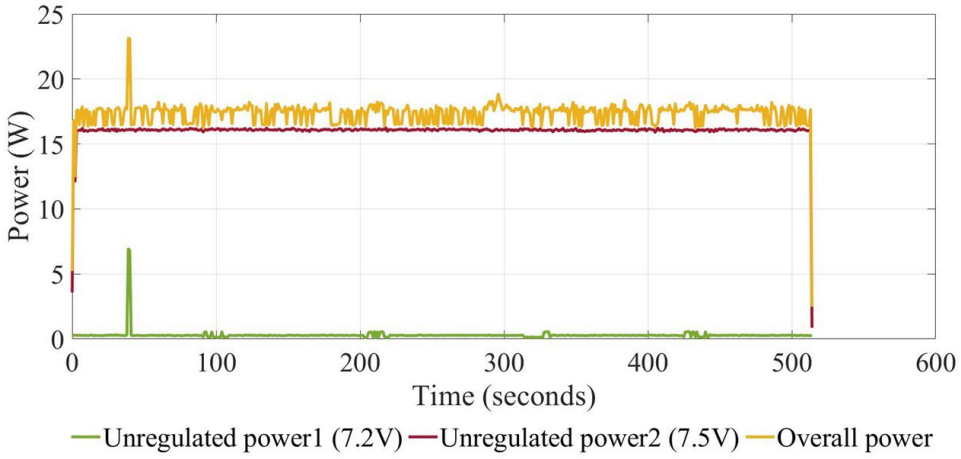


Figure 31c. Power consumption of KITSUNE downlink PNG image (third pass).

The final pass was scheduled for the deep learning algorithm execution. The wildfire classification script was tested using the pre-trained CNN models saved onboard KITSUNE: ShallowNet, LeNet, and MiniVGGNet. The objective of this pass was to identify which models are applicable and correctly classify the images captured by the imaging payload. Fig. 32 showed the complete DL execution cycle, from classifying images to saving the classified data into the C-band flash memory. The algorithm was executed five times using different CNN models and image combinations at the first 1142 sec. Six short peaks of 7 W unregulated power1 were observed due to the uplink command for this purpose. The result showed that the algorithm took about 137 sec to classify each image and consumed about 680 MB of RPi CM3+ memory, referring to the 5 V orange line. At the end of the test, the RPi CM3+ and C-band transceiver was turned on for transferring classified image data between the flash memories. In addition, the final classification results were appended to the image data for analysis at the GS.

Ultimately, the four passes verified the operation of the imaging mission and the demonstration of executing a DL-classification onboard KITSUNE, as summarized in Table 7.

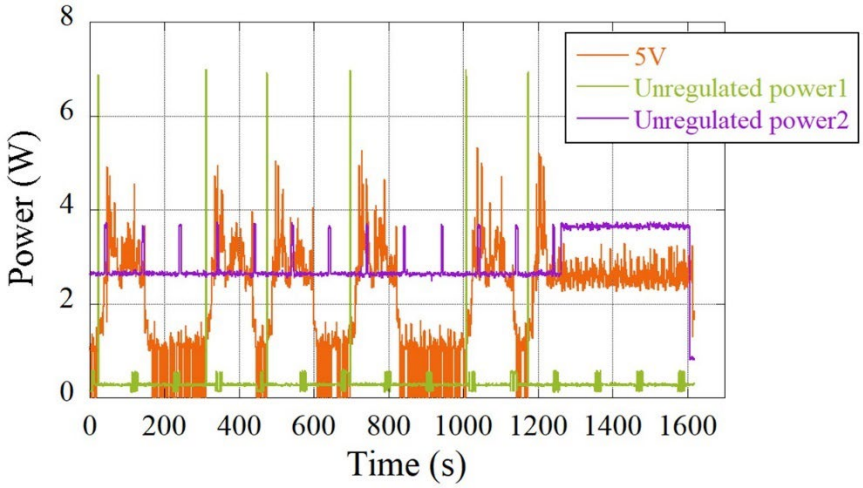


Figure 32. Power consumption of KITSUNE deep-learning execution (fourth pass).

Table 7. Summary of mission execution in four passes during the long duration operational test ("x" means turn on and "-" means turn off).

	Pass			
	1	**2**	**3**	**4**
Purpose	**Camera capture**	**Downlink thumbnails and JPG image**	**Downlink PNG image via C-band**	**Deep-learning execution**
12 V	+	-	-	-
5 V	+	+	-	+
Unregulated power1	-	+	+	-
Unregulated power2	-	+	+	-
Total duration (s)	1400	130	512	137
Peak power (W)	18.40	20.10	23.14	5.31
Energy consumption (Wh)	2.90	2.81	2.33	0.10

4.4 Training and classification test

The wildfire dataset was trained using Google Colab Pro. It allows researchers to write and execute deep learning algorithms through the browser with faster GPUs. However, the Colab has memory limitations when executing a higher CNN layer with large input sizes such as ResNet 224 x 224 pixels. Other CNN models that have been used in this study were well-suited using the Colab environment. The models were thoroughly trained with multiple parameters, including the input pixel size. Table 8 shows the summarized training result of the CNN models in the percentage of overall accuracy (OA) and F1-score of wildfire. The highest results corresponding to the CNN models were plotted in Fig. 33. The top three OA of the pre-trained CNN models were ResNet (RN) with 99%, MiniVGGNet (MVGGN), and MiniGoogLeNet (MGLN) with a similar 98%. The F1-score of wildfire also showed the highest

for RN, MVGGN, and MGLN with 97%. However, only MVGGN were implemented into the KITSUNE with ShallowNet (SN) and LeNet (LN) due to the medium range of computational costs for the RPi CM3+. The six pre-trained models implemented the CoFFI GUI for the post-processing image downlink. Several studies have shown that applying a Deep Neural Network for cloud detection could achieve 92% accuracy, while CNN for binary classification could obtain 90% accuracy [39]. Therefore, the main contributions of this study were the custom-made training dataset and the parameter tuning of the neural network.

Table 8. Summary of training CNN models with different input pixel sizes: overall accuracy (F1 score of wildfire) [model size].

Model/Input pixel size	100 x 100	128 x 128	224 x 224
ShallowNet	95 (92) [5 MB]	94 (92) [8 MB]	0.88 (0.74) [25 MB]
LeNet	97 (94) [60 MB]	97 (95) [98 MB]	0.97 (0.94) [300 MB]
MiniVGGNet	98 (97) [79 MB]	95 (91) [129 MB]	0.97 (0.94) [393 MB]
AlexNet	95 (90) [95 MB]	91 (88) [115 MB]	87 (68) [223 MB]
MiniGoogLeNet	98 (97) [7 MB]	98 (96) [7 MB]	97 (94) [7 MB]
ResNet	98 (97) [5 MB]	99 (97) [5 MB]	Memory limit

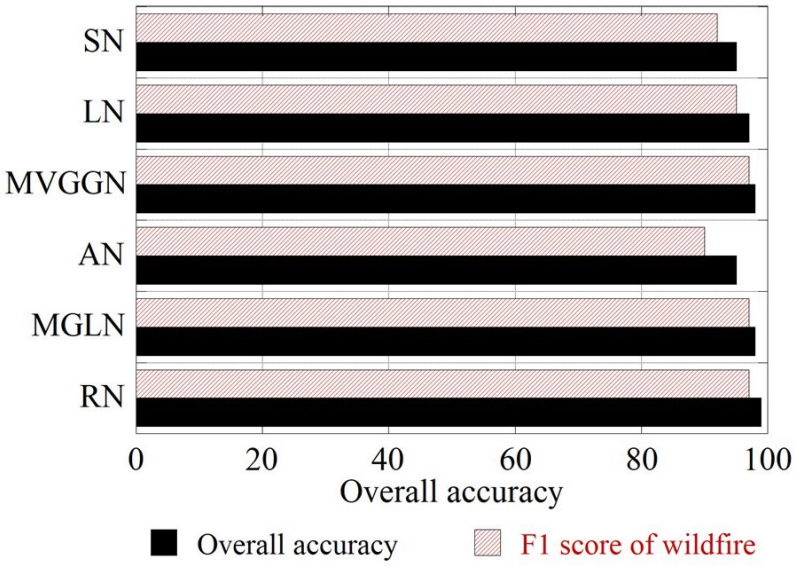


Figure 33. Comparison of ShallowNet (SN), LeNet (LN), MiniVGGNet (MVGGN), AlexNet (AN), MiniGoogLeNet (MGLN), and ResNet (RN) models in overall accuracy and F1 score of wildfire results.

In addition, the training loss and accuracy were plotted to correlate with the top three pre-trained CNN models in 100 epochs (Fig. 34). ResNet achieved 90% training accuracy at epoch six and less than 50% of training loss at epoch 70. Notably, these top three CNN models

were trained at different learning rates to gain the highest accuracy possible, 0.05 for MVGGN, whereas it was 0.005 for RN and MGLN. On the other hand, other parameters such as input pixel size and type of optimizers were trained to identify whether they would increase the F1 scores of the CNN models. Fig. 35 shows the differentiation results on the ResNet pre-trained model. The result verified that increasing the input pixel size and dataset would not increase the scores, while the best optimizer implemented on the dataset was the Stochastic Gradient Descent (SGD) combined with the data augmentation of rotation, zoom, shift, and flip the images. Therefore, the optimum combination of the ResNet model parameter was 128 x 128 pixels, SGD, and augmented the dataset.

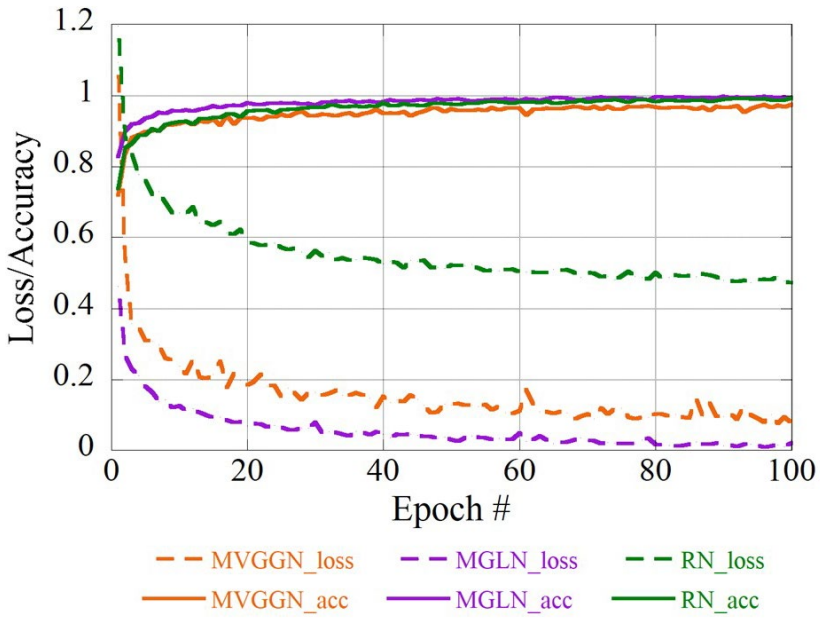


Figure 34. Training loss and accuracy of MiniVGGNet (MVGGN), MiniGoogLeNet (MGLN), and ResNet (RN) models.

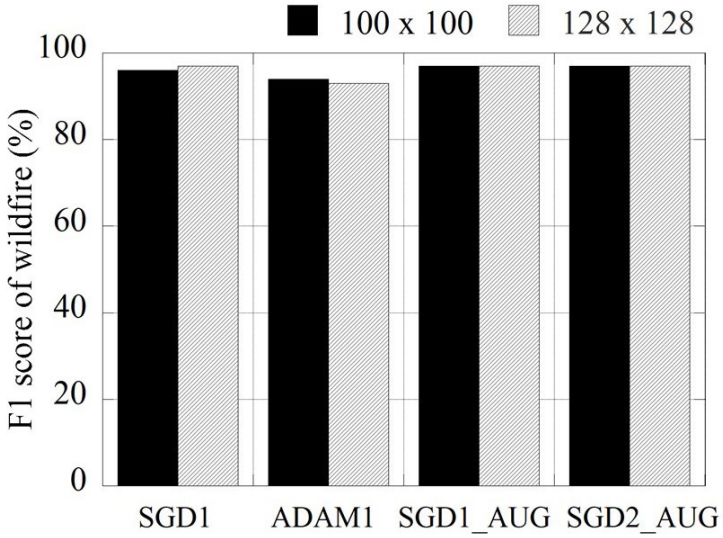


Figure 35. F1 score of wildfire results using ResNet model with different input pixel sizes and optimizer types ("1" and "2" indicate the learning rate of 0.005 and 0.05, respectively).

The confusion matrix and classification report were calculated and tabulated as standard techniques to verify the accuracy of CNN models. Similar analysis methods were also conducted to test the dataset concerning the CNN or ML classifiers [61-63]. The results were generated using the scikit-learn library included in the python script (Table 9). The gray color shows the true positive (TP) with and without the normalized values, respective to the four labels. The confusion matrix shows that the top three CNN models were chosen based on the small number of false-positive (FP) and false-negative (FN) results: 0 and 14 for MiniVGGNet, 5 and 9 for MiniGoogLeNet, and 3 and 10 for ResNet, respectively (Fig. 36). Therefore, the ResNet pre-trained model was considered the best CNN network compared to MiniVGGNet and MiniGoogLeNet, with minor type 1 errors (FP) and type 2 errors (FN).

Table 9. Confusion matrix of MiniVGGNet, MiniGoogLeNet, and ResNet (gray color means the TP value).

CNN Models	True/Predicted Labels	Cloud	Land	Sea	Wildfire
MiniVGGNet	Cloud	233 96.7%	8 3.3%	0 0%	0 0%
	Land	1 0.4%	256 99.6%	0 0%	0 0%
	Sea	0 0%	0 0%	267 100%	0 0%
	Wildfire	1 0.4%	13 5.5%	0 0%	221 94.0%
MiniGoogLeNet	Cloud	231 98.3%	1 0.4%	0 0%	3 1.3%
	Land	4 1.7%	235 97.5%	0 0%	2 0.8%
	Sea	0 0%	0 0%	257 100%	0 0%
	Wildfire	1 0.4%	7 2.6%	1 0.4%	258 96.6%
ResNet	Cloud	239 99.2%	0 0%	0 0%	2 0.8%
	Land	1 0.4%	255 99.2%	0 0%	1 0.4%
	Sea	0 0%	0 0%	267 100%	0 0%
	Wildfire	6 2.6%	4 1.7%	0 0%	225 95.7%

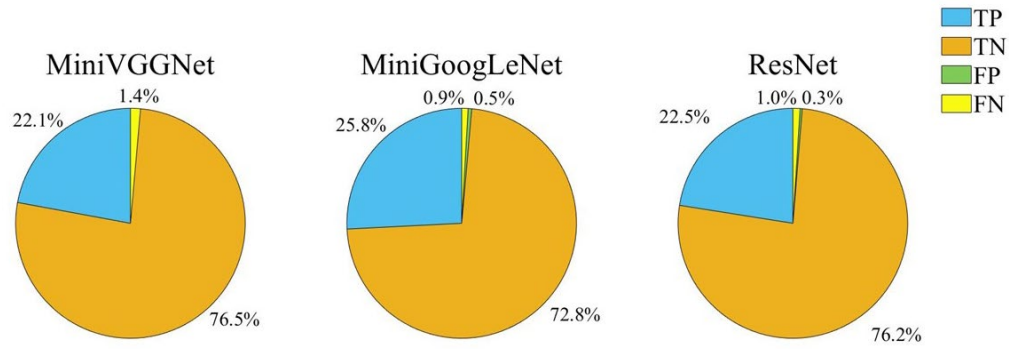


Figure 36. The true positive, true negative, false positive, and false-negative results correspond to the CNN models.

The CNN models also tested several unprocessed images with the classification script. The wildfire images were retrieved from the Doves, Sentinel-2, and Landsat-8 satellites. They were captured during the Kincade-Maria fire in the USA in October 2019 and the Turkey wildfire in July 2021 (Fig. 37). The results showed the correct and incorrect classification by implementing the pre-trained ResNet model. Most of the images from the Sentinel-2 and Landsat-8 were able to predict correctly, while fewer were from the Doves images. Even though the dataset was extensively trained using multi-resolution images, positive classification remained challenging, especially for the high-resolution images. The clouds were commonly characterized as misclassified outcomes. Therefore, the result could be improved if a visible and infrared camera payload were implemented and multi-band images were added to the dataset. Nevertheless, in this study, wildfire-detection processing onboard the CubeSat was verified using a visible dataset, with the best CNN model being ResNet, MiniVGGNet, and MiniGoogLeNet.

Figure 37. Classification test results using the pre-trained ResNet model: (a) correct; (b) incorrect predictions.

In the last experiment, a computational cost study was also conducted. Energy consumption, computational memory, and model size will impact the CubeSat mission design and operation. Because of the CubeSat constraints, training or re-training a CNN model onboard was challenging and nearly impossible (without the accelerator). However, loading the pre-trained CNN model for classifying images onboard was likely feasible based on the results shown in Fig. 38. The model size was not a serious issue to be implemented in the RPi CM3+ due to the large flash memory storage, such as for the LeNet model. It was also not affecting the energy consumption and computational memory. Moreover, the CNN classification results

indicate that the computational cost was practical because the energy required was below 100 mWh. However, for the safety and smooth KITSUNE operation, only the first three models were implemented (ShallowNet, LeNet, and MiniVGGNet). The RN model showed the highest energy consumption but was the smallest model size due to the very depth convolution layers with impressive performance. Overall, the state-of-art of CNN networks has a higher accuracy with great depth in a small-sized generated model.

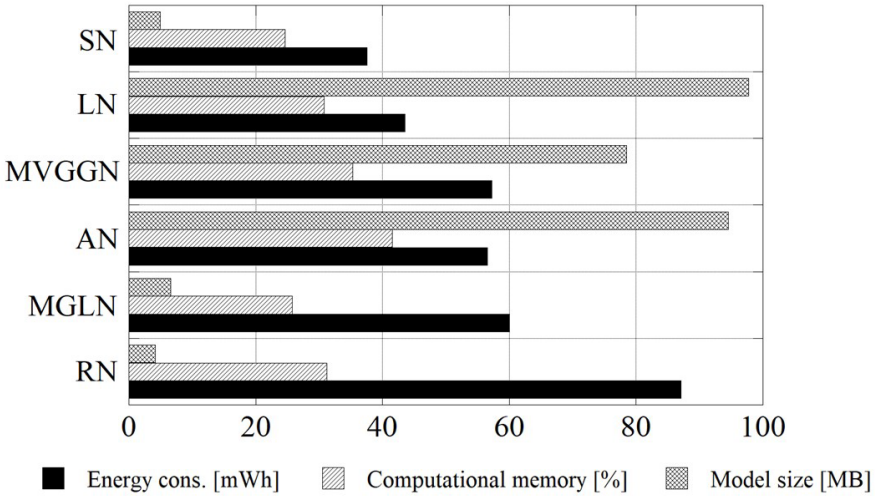
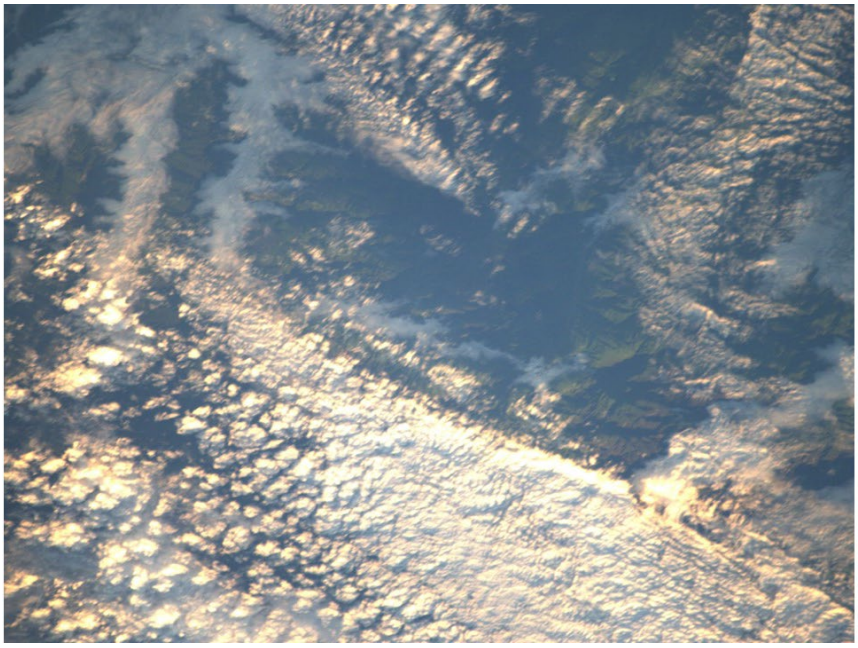


Figure 38. Result of energy consumption, computational memory, and model size for ShallowNet (SN), LeNet (LN), MiniVGGNet (MVGGN), AlexNet (AN), MiniGoogLeNet (MGLN), and ResNet (RN).

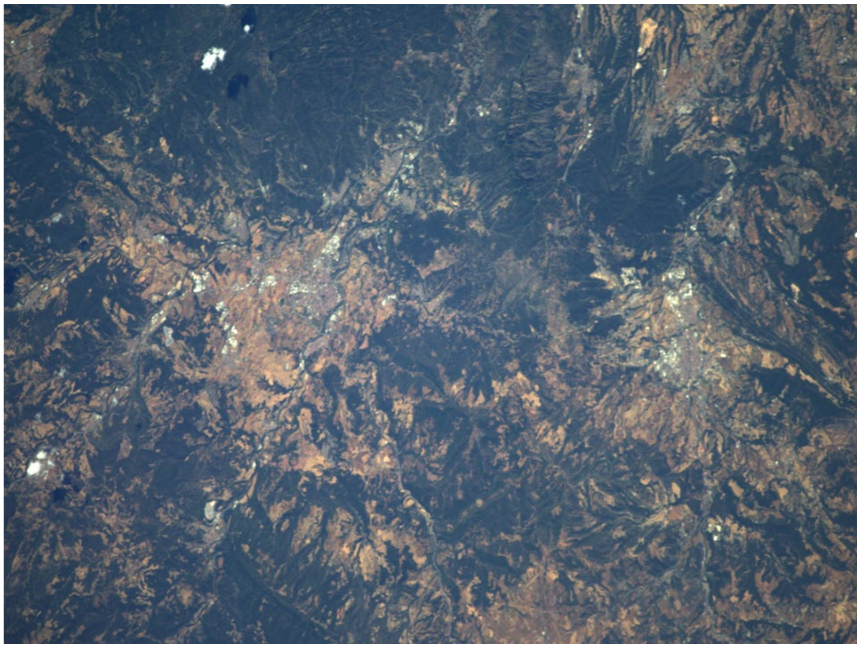
4.5 KITSUNE in-orbit data

To date, the KITSUNE is in the progress of completing the initial phase of operation. The main camera mission has been executed with several thumbnail data downloaded. However, only two full JPG images were able to be retrieved through the C-band GS (Fig. 39). The details of the image captured are as follows:

1) Image ID: 28 and 150
2) Date and time captured (JST): 10 May 2022 (07:18 AM) and 24 June 2022 (20:06 PM)
3) Location: Hubei (China) and Barcelona (Spain)
4) Size: 2.22 MB and 2.23 MB
5) Exposure time: 256 µsec and 70 µsec
6) Gain: 22 dB and 24 dB



(a)



(b)

Figure 39. The full JPG images downlinked by KITSUNE (a) image 28 and (b) image 150.

In addition, the image was also tested and classified using the CoFFI GUI on the ground computer. Table 10 shows prediction results in percentage applying six CNN pre-trained models. Most networks classified the image as land, while ShallowNet and LeNet were classified as wildfire images. However, the ground truth of the image consists of clouds and land. The results also showed that MiniVGGNet, AlexNet, MiniGoogLeNet, and ResNet could classify KITSUNE images correctly with a confidence level of more than 94%. Therefore, the

models half correctly predicted the image, however, more full JPG images should be downloaded to verify the accuracy of the CNN models.

Table 10. Summary of prediction results using the CoFFI GUI in different CNN models (gray color means the highest prediction) (ID 28 is image 28 and ID 150 is image 150).

Model/Labels	Cloud		Land		Sea		Wildfire	
	ID 28	ID 150	ID 28	ID 150	ID 28	ID 150	ID 28	ID 150
ShallowNet	9.33	0.00	0.01	23.07	0.00	5.84	90.66	71.09
LeNet	0.43	0.00	0.00	0.17	0.00	0.04	99.57	99.80
MiniVGGNet	0.73	0.00	99.04	99.94	0.00	0.06	0.22	0.00
AlexNet	0.00	0.00	99.98	99.77	0.00	0.00	0.02	0.23
MiniGoogLeNet	0.00	0.00	99.98	99.97	0.00	0.00	0.01	0.03
ResNet	0.60	0.06	94.75	98.89	0.00	0.01	4.64	1.04

CHAPTER 5
DISCUSSION

5.1 Additional discussion

An advanced camera sensor with more than multispectral (MS) is called a hyperspectral (HS) imager, which is widely available on a miniaturized scale. The hyper bands of a sensor in a unit are significantly helpful for land applications. An AI-based integrated with HS payload is considered challenging regarding processing power onboard a CubeSat. Moreover, the downlink capabilities of CubeSat could be another challenge in addition to the power generation from the solar panels per orbit. Danielsen et al. [63] showed how much power was consumed for the onboard processing of HYPSO-1 integrated with an HS imaging payload. The mission utilized the Self-Organizing Maps (SOMs) on sea scenes with an overall accuracy of above 90%, which took 380 sec to execute. Therefore, the accuracy and computational time could be improved by implementing a CNN model, as the results shown in this study for wildfire image classification reached 99% accuracy using the energy of 87.05 mWh. It also indicates that the computational costs could significantly affect the ML or DL implementation onboard CubeSat.

The PhiSat-1 6U CubeSat has also demonstrated the Deep Neural Network (DNN) for cloud detection using an HS imager. The test set achieved a 92% accuracy with a 1% FP using the dataset acquired from Sentinel-2. As a result, the CNN-based algorithm consumed only 0.16 mWh of energy because of the support by the integrated Myriad 2 VPU onboard PhiSat-1. Nevertheless, in this study, the MiniVGGNet model implemented onboard KITSUNE demonstrated a 98% accuracy, 0% FP, and consumed 0.10 Wh of energy without the accelerator installed. It could be claimed that the significant difference was in the power consumption measured from the RPi CM3+ turning on, the DL execution, and then shutting it off. The RPi CM3+ was designed not to be turned on constantly to prevent the SEL anomaly that was noticed during the TID ground test. However, the RPi CM3+ consumed only 56.56 mWh when it was always turned on. The other half of mWh was from the energy use of turning the RPi CM3+ on and off. The power could be lowered in the same manner if the CNN libraries were initialized and maintained during the CubeSat operation.

Furthermore, the deep learning networks were frequently implemented on the ground-based computer without resource limitations. The reasons and methods of processing image downloaded from the satellite has been discussed by Chen et al. [64]. Generally, the significant issue was comparing the traditional satellite and CubeSat regarding limited power and downlink capability. The state-of-the-art computer vision implemented onboard CubeSat mainly trained the satellite processor to classify the image before downloading the data. Larger satellites do

not argue to downlink a giga-byte of image data due to high-speed downlink communication. Nevertheless, the image classification onboard in this study has created an advantage for the nanosatellite class and was complementary to the low latency of downlink. Hence, the computational cost and power challenges could be improved in the mission plan strategy.

Moreover, the learning or re-training capability should be implemented to create an efficient and autonomous AI-based processor. The new images captured by the CubeSat payload can be extensively utilized to update the default dataset and increase the accuracy of the CNN models. In this study, the KITSUNE design was tested and showed promising results for re-training, but several issues were raised. The critical question would be regarding how much confidence there can be that the images captured are correctly classified and sorted according to the label for the re-training process. Human involvement is highly necessary for the training process of the CNN algorithm. At the end of this study, the image classification was only implemented onboard KITSUNE to reduce the risk of operation failure. However, a different approach was discussed by Mikuriya et al. [65], where the updated classifier will be shared between the CubeSat and ground learning. The concept was appealing, but the method would burden the uplink communication to the CubeSat, where most of the CubeSat fails to establish the communication link.

Despite testing only six CNN networks, other models have not been initialized yet with the wildfire dataset, such as Inception, Deeper-GooogLeNet, DenseNet, and U-Net. The image processing consists of three tasks: image classification/recognition, object detection and location, and image segmentation, as mentioned by Buonaiuto et al. [67]. The effectiveness level of image processing would significantly increase by applying these three tasks onboard CubeSat. In the KITSUNE design, only image classification was applied due to the limited resources for generating the dataset. Each task needs to train using different types of datasets. The availability of high-resolution and multi-band satellite images was limited but challenging to be downloaded for creating the dataset. Thus, most land applications required deeply high pixels for the analysis.

Overall, satellite technology for remote sensing applications is vital in reducing wildfire impacts. Combining image classification onboard CubeSat and segmentation of the data on the ground could truly benefit the authorities in handling forest fires. Szpakowski and Jensen [22] reviewed the techniques in fire ecology, which are fire-risk mapping, fuel mapping, active fire detection, burned area estimates, burn severity assessment, and post-fire vegetation recovery monitoring. The techniques were performed in three phases: before, during, and after the disaster. Meanwhile, this study only covers during an active fire by classifying images captured onboard CubeSat. The mapping and assessment should be conducted on the ground, utilizing the CNN image segmentation algorithm. Overall, the idea was to create an AI

ecosystem onboard CubeSat and ground segments to overcome the global wildfire consequences.

5.2 Case study of developing a 6U CubeSat for wildfire detection mission (AIWSat)

A conceptual design of 6U CubeSat has been studied as having the main mission of wildfire detection. The new CubeSat, Artificial Intelligence Wildfire Satellite (AIWSat) idea was based on the experience and lessons learned from the KITSUNE project. This section will reflect several improvements from mission design to operational planning.

5.2.1 Mission statement and objective

5.2.1.1 The CubeSat shall provide a 7-m class resolution imaging service for detecting wildfire using deep learning approaches onboard.

5.2.1.2 The CubeSat objective shall capture wildfire events within 10 km2 of the image area.

5.2.2 Mission requirements

The mission requirements consist of functional, operational requirements and constraints, as shown in Table 11.

Table 11. Summary of functional and operational requirements with the constraints.

Requirement	**Factors**	**AIWSat**
Functional requirements		
Performance for the primary objective	Orbit, altitude, inclination, and period	AIWSat orbit shall be in low Earth orbit with an altitude of 400 km, an inclination of 51.6°, and a period of 92 min.
	Payload	AIWSat payload shall be a short-wavelength infrared (SWIR) main camera and a visible (VIS) secondary camera.
	Operational modes	AIWSat payload shall have two operational modes: 1. Photo mode 2. Video mode
	Size	Wildfire payload size shall be within 327.5 mm (L) x 90 mm (W) x 90 mm (H)

	Pointing and mapping	Wildfire pointing and mapping shall be ±5 km for pointing error and ±0.5° for mapping error
	Resolution	Wildfire payload resolution shall be as follow: 1. Main camera: 7 to 9 m 2. Secondary camera 25 to 30 m
	Accuracy	AIWSat payload accuracy shall be 7 m
	Resources	The mission payload resources shall be as follows: 1. Weight: ≤ 7 kg 2. Size: ≤ 327.5 mm x 90 mm x 90 mm 3. Communication: a. Downlink transmission 10 Mbps b. Telemetry, tracking, and command system (TT&C) c. Mission data
	CubeSat configuration	AIWSat CubeSat's configuration shall be a 6U-wide structure of 340.5 mm x 226.3 mm x 100.0 mm, has a bus system of electrical power, TT&C, attitude control, and thermal control
	Weight	AIWSat weight shall be 14 kg
	Electrical power system	AIWSat EPS shall be provided 7 Wh generated from 5 solar panels on the system
	Attitude determination and control system (ADCS)	AIWSat ADCS shall be the earth sensor's accuracy of 0.1°, sun sensor's accuracy of ≤ 5°, global positioning system (GPS) accuracy of 35 m, magnetometer,

		gyroscope, reaction wheels, and magnetorquers.
Coverage	Orbit	AIWSat coverage shall be orbit in low Earth orbit (LEO)
	Swath width	AIWSat swath width shall be an area abound 10 km
	Number of satellites	AIWSat shall be a 1 CubeSat only
Responsiveness	Communications architecture	AIWSat communications architecture shall be telemetry, tracking and control subsystem (TT&C) using UHF and data relay and communication using C-band
	Processing delays	AIWSat processing delays should be ≤ 1 hour after the observation
	Data delivery time for users after observation	AIWSat data delivery time for users after observation shall be within 24 hours
Operational requirements		
Availability	Percentage of CubeSat availability for mission operations/duty	AIWSat availability shall be continuous imaging time of 10 minutes, orbital period of 92 minutes, and percentage of CubeSat availability for mission operations of 10.9% (could be improved with multiple GS networks)
Data distribution	Communications architecture up to end-users	AIWSat communications architecture up to end-users shall be: 1. Receiving and compiling observation requests 2. Operation planning 3. Satellite control and tracking operation 4. Observation data reception, processing, and delivery

Data content, form, and format	User needs	AIWSat user needs shall be satellite images and observation data
	Payload data form and format	AIWSat payload data form and format shall be in RAW, PNG, and JPG
Constraints		
Cost	R&D, the flight system design, development, manufacturing and tests, launch and initial checkout	AIWSat cost constraints shall be a total of JPY 10M
Schedule	Initial operating capability after the project start	AIWSat schedule constraints shall be within two years of initial operating capability after the project start
Launch vehicle	Rocket launcher and deploy from ISS	AIWSat launch vehicle shall be used any rocket launcher for ISS supplier
Operational duration	Mission operational	AIWSat operational duration shall have a mission operational at least one year, and the aim goal shall be two years
Reliability	The bus systems for one year	AIWSat reliability of the bus systems for one year shall be 0.9

5.2.3 CubeSat configuration

AIWSat 6U CubeSat configuration has been designed following the mission requirements, as illustrated in Fig. 40. The CubeSat will be embedded with the deep learning network installed on Raspberry Pi Compute Module 4 (RPi CM4). The main SWIR camera sensor (ATP013S-WC) was chosen based on the mission objective and requirements.

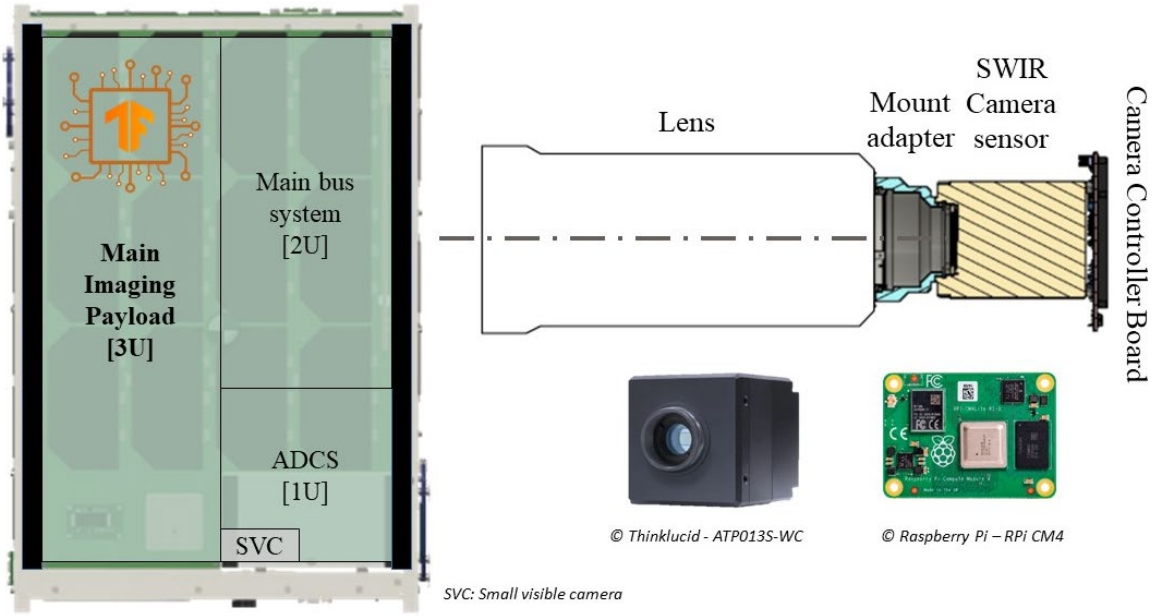


Figure 40. AIWSat configuration consists of 3U imaging payload, 2U main bus system, 1U attitude determination and control system (ADCS), and small visible camera (SVC).

CHAPTER 6
CONCLUSION AND RECOMMENDATION

6.1 Conclusion

It was concluded that implementing the deep learning technique in the nanosatellite class is feasible and has been tested in KITSUNE 6U CubeSat throughout its development. The KITSUNE imaging payload design was discussed in order to achieve the main objective of 5-m resolution. The wildfire image classification CNN algorithm was also tested as the secondary mission along with the functional test of the satellite system. The space environment tests were conducted thoroughly to verify the camera payload. Meanwhile, six CNN models were trained and half were implemented onboard the KITSUNE flight model. The highest overall accuracy from training the custom-made dataset was 99%, while 97% was the F1 score wildfire label using the ResNet model. However, the MiniVGGNet was implemented onboard KITSUNE with 98% overall accuracy and 97% F1 score wildfire, together with ShallowNet and LeNet. The MiniVGGNet showed considerably high results without any risks affecting the main mission. In addition, the space environment on-ground tests demonstrated the workability of the imaging mission with the deep learning algorithm. Nevertheless, the TID radiation result showed that the radiated environment in space might affect the image classification performance onboard by a single event latch-up during the test. Additional precautions should be planned before executing the main mission with deep learning. Ultimately, more KITSUNE in-orbit data could fully verify the mission's success by capturing images with the wildfire feature.

The wildfire problem in remote sensing applications was also investigated. KITSUNE CubeSat is the first CubeSat that implemented a deep learning approach for wildfire detection. The experiences learnt from the project could benefit future CubeSat development, such as the Artificial Intelligence Wildfire Satellite (AIWSat). A case study has been presented in chapter 5 for improvements needed to be made. The wildfire events could be detected more accurately using SWIR spectral range to discard the cloud image. Therefore, a fusion of visible and SWIR camera payload is an excellent design leveraging four spectral bands embedded with the deep learning algorithms for onboard image processing.

6.2 Contribution

As CubeSat launching has significantly increased each year, autonomous processing should be implemented onboard for particular applications. To date, only three CubeSats have shown an outstanding flight result of using machine learning onboard, two of which were applied for cloud detection. KITSUNE CubeSat will become the next nanosatellite to enable a

deep learning approach for wildfire detection. It could benefit the respective authority by retrieving a valuable image directly from the CubeSat. Furthermore, the lesson learnt from the KITSUNE project is important to develop an improvement for the future CubeSat. The case study could also be used for the next remote sensing satellite in Malaysia as forest fire events are occurring yearly due to climate change. Moreover, the deep learning network could also be applied for different remote sensing applications. Thus, the result will be a reference for the new AI CubeSat.

6.3 Recommendation for future work

Several recommendations have been identified for future work. Instead of only implementing the image classification technique, image segmentation is also beneficial for processing and analysing the captured image. It could be done onboard the CubeSat but post-processing on-ground computer is more reliable. The computational costs of image segmentation onboard CubeSat should be tested first to verify the feasibility of the implementation. The reason is that the image segmentation will use more layers, affecting the CubeSat operational power. Thus, an extended study is necessary to implement image segmentation onboard the CubeSat.

Furthermore, multispectral and hyperspectral imagers are pleasing candidates for camera sensors onboard CubeSat. The power consumption is within the allowable CubeSat specification. However, for wildfire applications, the spectral range is not covered by these imagers. SWIR is considered the accurate range to detect wildfire from space. Generally, a thermal sensor requires a cooling system to capture longer electromagnetic spectrum wavelengths. Nevertheless, the HS imager is not equipped with such a system due to the multi-band in a camera module. Comprehensive research should be made to verify that the new technology of HS imager could cover the SWIR range, mainly for wildfire detection.

REFERENCES

[1] Nakasuka, S.; Sako, N.; Sahara, H.; Nakamura, Y.; Eishima, T.; Komatsu, M. Evolution from education to practical use in University of Tokyo's nano-satellite activities. Acta Astronaut. 2010, 66, 1099–1105. https://doi.org/10.1016/j.actaastro.2009.09.029.

[2] Tsuda, Y.; Sako, N.; Eishima, T.; Ito, T.; Arikawa, Y.; Miyamura, N.; Tanaka, A.; Nakasuka, S. University of Tokyo's CubeSat project—Its educational and technological significance. In Proceedings of the the 15th Annual AIAA/USU Conference on Small Satellites, Logan, UT, USA, 13–16 August 2001.

[3] M. Cho, M. Hirokazu, and F. Graziani, "Introduction to lean satellite and ISO standard for lean satellite," RAST 2015 - Proc. 7th Int. Conf. Recent Adv. Sp. Technol., pp. 789–792, 2015, DOI: 10.1109/RAST.2015.7208447.

[4] Kim, S.; Yamauchi, T.; Masui, H.; Cho, M. BIRDS BUS: A Standard CubeSat BUS for an Annual Educational Satellite Project. JoSS 2021, 10, 1015–1034.

[5] Boshuizen, C.R.; Mason, J.; Klupar, P.; Spanhake, S. Results from the Planet Labs Flock Constellation. In Proceedings of the 28th AIAA/USU Conference on Small Satellites, Logan, UT, USA, 4–7 August 2014.

[6] J. C. Daniel Bryce and Manufacturing, "Smallsat Manufacturing, The Spire 'Constant NPI' Model. Daniel," Small Satell. Conf., pp. SSC19-I–01, 2019.

[7] A. Klesh et al., "MarCO: Early Operations of the First CubeSats to Mars," 32nd Annu. AIAA/USU Conf. Small Satell., pp. 1–6, 2018.

[8] T. Gardner et al., "CAPSTONE: A CubeSat Pathfinder for the Lunar Gateway Ecosystem," Small Satell. Conf., no. 720, 2021.

[9] E. Dotto et al., "LICIACube - The Light Italian Cubesat for Imaging of Asteroids In support of the NASA DART mission towards asteroid (65803) Didymos," Planet. Space Sci., vol. 199, no. January, p. 105185, 2021, DOI: 10.1016/j.pss.2021.105185.

[10] E. William and A. Camps, Introduction to Satellite Remote Sensing. Elsevier, 2017. DOI: 10.1016/C2015-0-04517-8.

[11] https://www.nesdis.noaa.gov/current-satellite-missions/currently-flying

[12] Centre for Research on the Epidemiology of Disasters, "CRED Crunch 62 – 2020 Annual Report," May 2021, <https://www.cred.be/publications> (12 October 2021).

[13] Centre for Research on the Epidemiology of Disasters, "CRED Crunch 64 – Extreme weather events in Europe," September 2021, <https://www.cred.be/publications> (12 October 2021).

[14] Haddad, M., Hussein, M., "Mapping wildfires around the world," Aljazeera, 19 August 2021, < https://www.aljazeera.com/news/2021/8/19/mapping-wildfires-around-the-world-interactive> (12 October 2021).

[15] https://www.unep.org/news-and-stories/press-release/number-wildfires-rise-50-2100-and-governments-are-not-prepared

[16] https://sgp.fas.org/crs/misc/IF10244.pdf

[17] J. Lizundia-Loiola, M. L. Pettinari, and E. Chuvieco, “Temporal Anomalies in Burned Area Trends: Satellite Estimations of the Amazonian 2019 Fire Crisis,” Remote Sens., vol. 12, no. 1, p. 151, 2020.

[18] Hislop, S.; Haywood, A.; Jones, S.; Soto-Berelov, M.; Skidmore, A.; Nguyen, T.H. A satellite data driven approach to monitoring and reporting fire disturbance and recovery across boreal and temperate forests. Int. J. Appl. Earth Obs. Geoinf. 2020, 87, 102034. https://doi.org/10.1016/j.jag.2019.102034.

[19] Balmforth, T., “Russia says Siberian wildfires started on purpose by illegal loggers,” Reuters, 7 August 2019, < https://www.reuters.com/article/us-russia-wildfires-idUSKCN1UW1WY> (12 October 2021).

[20] C. Toth and G. Jóźków, “Remote sensing platforms and sensors: A survey,” ISPRS J. Photogramm. Remote Sens., vol. 115, pp. 22–36, 2016, DOI: 10.1016/j.isprsjprs.2015.10.004.

[21] A. Poghosyan and A. Golkar, “CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions,” Prog. Aerosp. Sci., vol. 88, no. September 2016, pp. 59–83, 2017, DOI: 10.1016/j.paerosci.2016.11.002.

[22] D. M. Szpakowski and J. L. R. Jensen, “A Review of the Applications of Remote Sensing in Fire Ecology,” Remote Sens., vol. 11, pp. 1–12, 2019, DOI: 10.3390/rs11222638.

[23] Y. Michael, I. M. Lensky, S. Brenner, A. Tchetchik, N. Tessler, and D. Helman, “Economic assessment of fire damage to urban forest in the wildland-urban interface using planet satellites constellation images,” Remote Sens., vol. 10, no. 9, 2018, DOI: 10.3390/rs10091479.

[24] A. Rosebrock, Deep Learning for Computer Vision with Python, 3rd ed., vol. Starter Bu. PyImageSearch, 2019.

[25] M. M. Pinto, R. Libonati, R. M. Trigo, I. F. Trigo, and C. C. DaCamara, “A deep learning approach for mapping and dating burned areas using temporal sequences of satellite images,” ISPRS J. Photogramm. Remote Sens., vol. 160, no. November 2019, pp. 260–274, 2020, doi: 10.1016/j.isprsjprs.2019.12.014.

[26] Richard HR Hahnloser et al. “Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit”. In: Nature 405.6789 (2000), page 947 (cited on pages 128, 130).

[27] Sergey Ioffe and Christian Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: CoRR abs/1502.03167 (2015). URL: http://arxiv.org/abs/1502.03167 (cited on pages 191, 195).

[28] Yann Lecun et al. “Gradient-based learning applied to document recognition”. In: Proceedings of the IEEE. 1998, pages 2278–2324 (cited on pages 26, 197, 221, 229).

[29] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large Scale Image Recognition”. In: CoRR abs/1409.1556 (2014). URL: http://arxiv.org/abs/1409.1556 (cited on pages 115, 194, 197, 229, 231, 280).

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: Advances in Neural Information Processing Systems 25. Edited by F. Pereira et al. Curran Associates, Inc., 2012, pages 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deepconvolutional-neural-networks.pdf (cited on pages 16, 88, 97, 105, 131).

[31] Chiyuan Zhang et al. “Understanding deep learning requires rethinking generalization”. In: CoRR abs/1611.03530 (2016). URL: http://arxiv.org/abs/1611.03530 (cited on page 135).

[32] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” CoRR, 2015, DOI: 10.1109/CVPR.2016.90.

[33] Dami Choi et al. ON EMPIRICAL COMPARISONS OF OPTIMIZERS FOR DEEP LEARNING. ArXiv : 1910.05446, 2019

[34] H. Leppinen, “Current use of linux in spacecraft flight software,” IEEE Aerosp. Electron. Syst. Mag., vol. 32, no. 10, pp. 4–13, 2017, DOI: 10.1109/MAES.2017.160182.

[35] https://sputnix.ru/en/equipment/cubesat-devices/materinskaya-plata-s-raspberry-pi-soft

[36] https://aitechsystems.com/product/s-a1760-space-gpgpu/

[37] https://kplabs.space/leopard/

[38] Chien, S.; Doubleday, J.; Thompson, D.; Wagstaff, K.; Bellardo, J.; Francis, C.; Baumgarten, E.; Williams, A.; Yee, E.; Fluitt, D. Onboard Autonomy on the Intelligent Payload EXperiment (IPEX) CubeSat Mission: A pathfinder for the proposed HyspIRI Mission Intelligent Payload Module. In Proceedings of the 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space, Montreal, QC, Canada, 18–22 June 2014; pp. 1–8.

[39] G. Giuffrida et al., “CloudScout: A deep neural network for on-board cloud detection on hyperspectral images,” Remote Sens., vol. 12, no. 14, pp. 1–17, 2020, DOI: 10.3390/rs12142205.

[40] https://www.rsp01.rymansat.com/en

[41] L. D. Young et al., “Ke Ao: A Low-Cost 1U CubeSat for Aerospace Education and Research in Hawaii,” in 35thAnnual AIAA/USU Conference on Small Satellites, 2021, pp. 1–11. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2021/all2021/18/

[42] https://www.csug.fr/menu-principal/projets/qlever-sat/qlever-sat-751384.kjsp

[43] M. Ziaja et al., “Benchmarking Deep Learning for On-Board Space Applications,” Remote Sens., vol. 13, no. 19, p. 3981, Oct. 2021, DOI: 10.3390/rs13193981

[44] M. Pastena et al., “Overview of ESA’s Earth Observation upcoming small satellites missions,” 34thAnnual AIAA/USU Conf. Small Satell., pp. 1–8, 2020, [Online]. Available: https://digitalcommons.usu.edu/smallsat/2020/all2020/127/

[45] https://www.inovor.com.au/space-technology/hyperion-mission/

[46] https://www.inovor.com.au/space-technology/skyris-mission/

[47] https://qengineering.eu/install-ubuntu-18.04-on-raspberry-pi-4.html

[48] https://pyimagesearch.com/2019/12/09/how-to-install-tensorflow-2-0-on-ubuntu/

[49] https://www.9news.com.au/bushfires

[50] https://www2.gov.bc.ca/gov/content/safety/wildfire-status/about-bcws/wildfire-history/wildfire-season-summary

[51] https://www.fire.ca.gov/incidents/2020/

[52] Balch, J.K.; St. Denis, L.A.; Mahood, A.L.; Mietkiewicz, N.P.; Williams, T.M.; McGlinchy, J.; Cook, M.C. Fired (Fire events delineation): An open, flexible algorithm and database of us fire events derived from the modis burned area product (2001–2019). Remote Sens. 2020, 12, 3498. https://doi.org/10.3390/rs12213498.

[53] Pinto, M.M.; Libonati, R.; Trigo, R.M.; Trigo, I.F.; DaCamara, C.C. A deep learning approach for mapping and dating burned areas using temporal sequences of satellite images. ISPRS J. Photogramm. Remote Sens. 2020, 160, 260–274. https://doi.org/10.1016/j.isprsjprs.2019.12.014.

[54] Wang, J.; Zhang, X. Investigation of wildfire impacts on land surface phenology from MODIS time series in the western US forests.

[55] https://www.bbc.com/news/world-australia-50951043.amp

[56] Chen, Y.; Lara, M.J.; Hu, F.S. A robust visible near-infrared index for fire severity mapping in Arctic tundra ecosystems. ISPRS J. Photogramm. Remote Sens. 2020, 159, 101–113.

[57] Collins, L.; Griffioen, P.; Newell, G.; Mellor, A. The utility of Random Forests for wildfire severity mapping. Remote Sens. Environ. 2018, 216, 374–384. https://doi.org/10.1016/j.rse.2018.07.005.

[58] Toumbas, G. Raspberry Pi Radiation Experiment; University of Surrey: Guildford, UK, 2018.

[59] Slater, W.S.; Tiwari, N.P.; Lovelly, T.M.; Mee, J.K. Total Ionizing Dose Radiation Testing of NVIDIA Jetson Nano GPUs. In Proceedings of the 2020 IEEE High Performance Extreme Computing Conference, HPEC 2020, Waltham, MA, USA, 22–24 September 2020.

[60] Pertuz, S., Puig, D., Garcia, M.A., 2013. Analysis of focus measure operators for shape-from-focus. Pattern Recognit. 46, 1415–1432. https://doi.org/10.1016/j.patcog.2012.11.011

[61] Maskey, A.; Cho, M. CubeSatNet: Ultralight Convolutional Neural Network designed for on-orbit binary image classification on a 1U CubeSat. Eng. Appl. Artif. Intell. 2020, 96, 103952. https://doi.org/10.1016/j.engappai.2020.103952.

[62] Thanh Noi, P.; Kappas, M. Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery. Sensor 2017, 18, 18. https://doi.org/10.3390/s18010018.

[63] Danielsen, A.S.; Johansen, T.A.; Garrett, J.L. Self-organizing maps for clustering hyperspectral images on-board a cubesat. Re- mote Sens. 2021, 13, 4174. https://doi.org/10.3390/rs13204174.

[64] Z. Chen, T. Yang, F. Wang, W. He, B. Hua, and Y. . Wu, “Ship Detection From Remote Sensing Images Based On Modified Convolution Neural Network,” pp. 1–6, 2019.

[65] Mikuriya, W.; Obata, T.; Ikari, S.; Funase, R.; Nakasuka, S. Cooperative Learning Between On-board and Ground Computers for Remote Sensing Image Processing. In Proceedings of the International Symposium on Space Technology and Science Aerospace Engineering, Fukui, Japan, 15–21 June 2019.

[66] Buonaiuto, N.; Kief, C.; Louie, M.; Aarestad, J.; Zufelt, B.; Mital, R.; Mateik, D.; Sivilli, R.; Bhopale, A. Satellite Identification Imaging for Small Satellites Using NVIDIA. In Proceedings of the 31st AIAA/USU Conference on Small Satellites, Logan, UT, USA, 5–10 August 2017.

[67] Mohanty, S.P.; Czakon, J.; Kaczmarek, K.A.; Pyskir, A.; Tarasiewicz, P.; Kunwar, S.; Rohrbach, J.; Luo, D.; Prasad, M.; Fleer, S.; et al. Deep Learning for Understanding Satellite Imagery: An Experimental Survey. Front. Artif. Intell. 2020, 3, 1–21. https://doi.org/10.3389/frai.2020.534696.