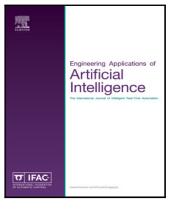
Engineering Applications of Artificial Intelligence 96 (2020) 103952

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



CubeSatNet: Ultralight Convolutional Neural Network designed for on-orbit binary image classification on a 1U CubeSat

Abhas Maskey *, Mengu Cho

Kyushu Institute of Technology, Kitakyushu-shi, Fukuoka, 804-8550, Japan

ARTICLE INFO

Keywords:
Convolutional Neural Network
Machine learning
Imaging payload
CubeSat

ABSTRACT

A 1U CubeSat has severe limitations on size, power and downlink capabilities. Transferring images taken by imaging payload from orbit can be a challenging. To save time and effort in downlinking and post-processing, a mechanism must be in place to sort quality data before transmitting it to the ground station. This paper presents an innovative approach combining a novel CubeSat image dataset and a lightweight Convolutional Neural Network architecture for automatically selecting images for downlink on a 1U CubeSat. Coined as CubeSatNet, the neural network is trained on 60,000 augmented images, is tiny enough to run on an ARM Cortex MCU and is tested on on-orbit data from Kyushu Institute of Technology's BIRDS-3 CubeSats with an accuracy of 90% and F1 score of 0.92. CubeSatNet outperformed SVM, DBN and AE trained to classify CubeSat images. If implemented, the CNN could cut down operation time by about 2/3 while significantly improving the quality of received data.

1. Introduction

Since the nanosatellite (1–10 kg) standard was first proposed in 1999 (Heidt et al., 2000), CubeSats have enabled educational institutions, small commercial entities and countries with limited to resources to build and launch very small satellites piggybacked as secondary payloads to space. As of 2020, more than a thousand such satellites have been launched (Maskey et al., 2019). The smallest form of CubeSat is a 1 Unit (1U). A 1U CubeSat is roughly the size of a Rubik's cube with $10 \times 10 \times 10$ cm^3 dimension, weighs about a kilo and consumes about a watt. The size, weight and power (SWaP) requirement for 1U CubeSat is stringent and developers must come up with creative to ways to implement newer missions. Any mission that can deploy on a 1U, can scale for larger satellites.

BIRDS-3 project from Kyushu Institute of Technology (Kyutech), Japan has a constellation of three 1U CubeSats; NepaliSat-1 from Nepal, Raavana-1 from Sri Lanka and Uguisu from Japan. The satellites have been operational since June 2019. Fig. 1 shows the 1U CubeSats before launch. Each satellite is equipped with a simple RGB camera to take images from space. The downlink is limited to 4800 bps in the amateur Ultra High Frequency (UHF) band with 35–45 min of communication window with one ground station (GS) per day. The image resolution is selected in such a way that it is large enough for media distribution while being small enough to downlink the image in a single day. Therefore, the highest image resolution is limited to VGA (640×480). For International Space Station (ISS) orbit of 400 km, the Ground Sampling Distance (GSD) of the image is about 1200 m.

The mission statement of the imaging payload for BIRDS-3 is to take images primarily for outreach and media purposes. For that, images that look "good" are necessary. Since BIRDS-3 satellites have active attitude stabilization but no control, there is no definite way of knowing whether the camera is pointed to space or the Earth when an image is captured. In the first three months of operation, a total of thirty images were downlinked. Of that, twelve are "good" earth-facing images. The rest are categorized as "bad". Fig. 2 shows some examples that include space, sunburnt cloudy and saturated pictures.

1.1. Downlink limitation of 1U CubeSat

Limited downlink leads to limited data. "CubeSats do not get more than a few MB over the course of a mission" (Palo et al., 2012). 3U CSSWE and RAX-2, which used the 9600 bps amateur UHF band, downloaded 60 and 242 MB of data respectively (Palo et al., 2012). They are isolated cases. BIRDS-3 have downlinked less than 1 MB of images in the first three months of operation from three different satellites. Ideally, more data could have been downlinked but due to factors of weather, ground station sharing between projects and antenna repairs, only a fraction of what can be downlinked is being accessed. Besides this, for every 50 packets of data downlinked, 10% to 15% are observed missing. Each set must be downlinked 2–3 times before a complete set can be merged and image is reconstructed. The same image must be downlinked 2–3 times to compensate for the loss in packets. BIRDS-3 team completes one image every 2–3 days.

* Corresponding author.
E-mail addresses: maskey.abhas481@mail.kyutech.jp (A. Maskey), cho@ele.kyutech.ac.jp (M. Cho).

https://doi.org/10.1016/j.engappai.2020.103952
Received 23 January 2020; Received in revised form 26 July 2020; Accepted 8 September 2020
Available online 16 September 2020
0952-1976/

Abbreviations

ADCS	Attitude Determination and Control
CNN	Convolutional Neural Network
COM	Communication Subsystem
CubeSat	Nanosatellite (1–10 kg) standard
DBN	Deep Belief Network
GS	Ground Station
GSD	Ground Sampling Distance
GSN	Ground Station Network
ICU	Image Classification Unit
MCU	Microcontroller
ML	Machine Learning
MTQ	Magnetorquer
NN	Neural Network
PWR	Power
qTFlite	quantized Tensorflow lite
AE	Autoencoders
SVM	Support Vector Machines
SWaP	Size, Weight and Power
TF	Tensorflow
UHF	Ultra-High Frequency

A few solutions have already been in place to address the issue of low downlink rate and missing packets. To reduce the image data size,

Fig. 1. Kyutech's BIRDS-3 CubeSat constellation are single watt systems of 1U CubeSat standard. Image from JAXA (2019).

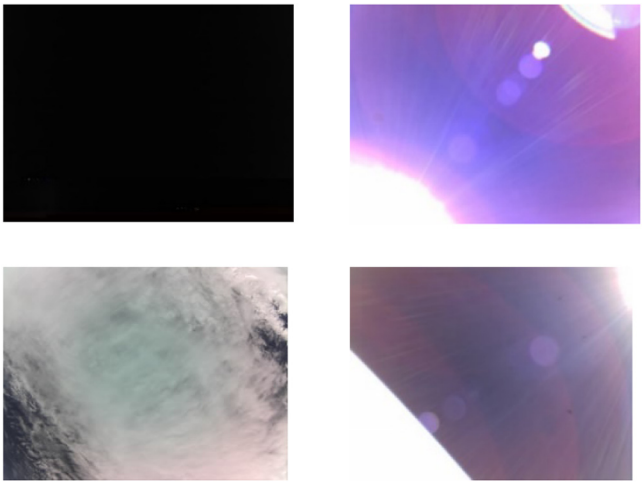


Fig. 2. Examples of images that are labelled "bad". From top-left clockwise: space, sunburn, saturated and cloudy.

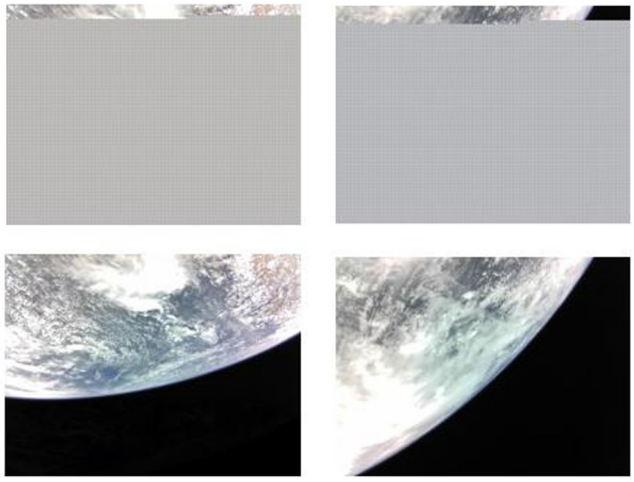


Fig. 3. Shows how operators of BIRDS-3 decide on downlinking an image. First 50 packets are downlinked and observed. If it is earth-facing, the downlink continues.

JPEG compression algorithm is in place. An eight-bit RAW RGB image for 640 × 480 has 307.2 kB of data. Depending on what the camera has captured and the quality of compression, BIRDS-3's 640 × 480 JPEG compressed data is anywhere between 7 kB to 74 kB which is between 2.3% to 24.1% of the original size. To increase the communication time, BIRDS Ground Station Network (GSN) of fifteen countries spread across in South America, Africa and Asia (Jirawattanaphol et al., 2016) including Kyutech is supporting the operation. To select "good" images, a manual downlink supervision process is also in place. The first set of 50 packets are downlinked and recreated. The operator can make an educated guess whether to proceed further with the downlink or move on to the next image. An example is shown in Fig. 3. However, this does not ensure that the final image will be "good" enough for media purpose as part of image could be saturated.

One way to ensure the quality would be to implement thumbnails created from the original image. This provides a preview of the image and can be used to make a sound decision whether or not to move forward with the downlink. The disadvantage of this would be the active involvement of an operator and the time required to select images from ground.

For further improvements, BIRDS-3 on-orbit lessons have been passed on to the BIRDS-4, the next project which involves satellites from Paraguay, the Philippines and Japan. BIRDS-4 have improved on the Attitude Determination and Control System (ADCS). A three-axis physical Magnetorquer (MTQ) is placed along with a Z-axis reaction wheel for active camera pointing. Furthermore, a software implementation on the camera allows burst mode. Several images are taken at once. The probability of taking a "good" image increases through the combination of active pointing and image volume.

Larger image volume, however, requires improved downlink speed. BIRDS-3 uses amateur UHF with an uplink/downlink bandwidth of 4800 bps. During downlink, the communication subsystem (COM) is transmitting at 0.8 W. The power consumption is 4.56 W giving a COM transmitter efficiency of 17.5%. Table 1 summarizes the COM parameters for BIRDS-3 satellites. The power consumption of 4.56 W means that 0.95 mJ (= 0.00026 mWh) per bit for 4800 bps downlink.

BIRDS-3 has NiMH batteries placed at three parallel, two series (3P2S) and has a total capacity of 3800 mAh rated at 3.8 V. The solar panel can generate an average of 1480 mWh per orbit. The limitation is not on the storage but on the power generation. Additionally, losses are to be expected while supplying. With 0.00026 mWh per bit, 1480 mWh can downlink 5.6 Mb or 700 kB per orbit at maximum. In a day, this is equivalent to 10 MB which is the theoretical limit of the amount of data downlinked per day assuming continuous communication with

Table 1
BIRDS-3 power (PWR) and COM characteristics.

Parameters	Details
Energy generation per orbit	1480 mWh
Transmission PWR	0.8 W
Supply voltage	3.8 V
Transmission Efficiency	17.5%
Over current protection	4 A

Table 2
Power consumption and ideal data downlink per orbit by increasing bitrate assuming that the COM is implemented on BIRDS-3 bus.

Frequency	Bitrate	PWR consumption (W)	Energy/bit (mWh)	Ideal data/orbit
UHF (BIRDS-3)	4.8 kbps	4.56	0.00026	700 kB
UHF	19.2 kbps	11.4	0.00017	1.12 MB
S-band	20 Mbps	11.4	1.6×10^{-7}	1.17 GB
X-band	120 Mbps	68.6	1.6×10^{-7}	1.17 GB

GS, full usage of the packet (i.e. no header and footer) and 100% power usage for communication. Therefore, the maximum amount of data that can be downlinked is probably 1 MB or less. In reality though, BIRDS-3 downlinks 25–30 kB of real data per day per satellite using the GSN which is far less than the theoretical limit.

Table 2 shows ideal power calculations and total communication time for an orbit if BIRDS-3 satellites were to increase the bitrate using commercially available COM. The calculations are based on EnduroSat's UHF (EnduroSat, 2019a), S-band (EnduroSat, 2019b) and X-band (EnduroSat, 2019c) COM modules. EnduroSat is used for reference. Table 2 shows that, in theory, it is still possible to increase bitrate up to 20 Mbps if S-band is implemented. Goliat (Cristea et al., 2009) has shown that it is feasible for a 1U CubeSat to transmit in S-band.

While placing a higher bitrate and frequency COM board into the bus system is an option, it is not as straight forward. The BIRDS satellites should have better attitude control for S-band patch antenna pointing. Licence needs to be applied where necessary and may extend the project time. The solution thus shifts away from hardware improvements. A more practical solution would be through software by classifying images on board and improving the quality of image downlinked through Machine Learning (ML).

1.2. Machine learning solution

For a classification problem that the paper is attempting to solve, supervised ML techniques such as Support Vector Machines (SVM) and Convolution Neural Network (CNN), a special type of Neural Network (NN), can be effective. SVM has the advantage of using less computational power, training dataset and memory. This is ideal for constraints imposed by a 1U CubeSat. On-orbit demonstration of SVM has been conducted on an AT91SAM9 processor on-board IPEX 1U CubeSat (Chien et al., 2017). IPEX could segment a 3 MP image into cloudy, clear, planetary limb or outer space (Thompson et al., 2015). Kyutech's BIRDS-4 has Image Classification Unit (ICU) mission in place using SVM to classify cloudy, non-cloudy, space and sunburnt image. The satellites are scheduled to be launched in 2020.

The idea of using NN for space applications is not new. The 1989 Goddard Conference on Space Application of Artificial Intelligence (AI) has papers where NN has been explored for planning and scheduling tasks, fault isolation and diagnosis, and cataloguing satellite imagery (Rash and Dent, 1989). However, researchers had difficulty working with NN as "the time and effort required to develop neural network architectures and training is very high" (Lee, 1991). That has changed with the recent advances in parallel computing, high-level open source ML libraries and big data through plethora of interconnected devices. NN is now being applied on remote sensed data to detect ships (Sonoke and Demura, 2019) and estimate ground-level Particulate Matter 2.5 ($PM_{2.5}$) (Zheng et al., 2020). NN is also now being applied for nanosatellite interplanetary autonomy (Feruglio and Corpino, 2017) and attitude control (Silva et al., 2019).

CNN is a subclass of NN where one or more convolutional layers are present besides the fully connected layer (FCL). Since AlexNet in 2012 (Krizhevsky et al., 2012), CNN has been the state-of-the-art technique for image classification tasks. Papers comparing SVM to CNN to classification problems in different fields show that CNN has performed better in the field of leaf identification (Hedjazi et al., 2017), lung cancer classification (Wang et al., 2017), facial recognition (Islam et al., 2018) and ship detection in satellite imagery (Marfu'ah and Kurniawardhani, 2020). While traditional ML algorithms such as SVM rely on hand crafted feature extraction, CNN's convolution layers (CL) automatically extracts features during training creating an optimized feature map for that problem.

In the field of remote sensing, feature extraction from hyperspectral images shows "supervised techniques provide better accuracy than their unsupervised techniques" (Kumar et al., 2020). CNN is the most popular algorithm in deep learning for remote sensing (Ma et al., 2019; Parikh et al., 2020). Other algorithms include Autoecoders (AE), Deep Belief Network (DBN), Recursive Neural Network (RNN) and Generative Adversarial Networks (GAN) (Ma et al., 2019). CNN has higher overall accuracy and kappa values for the University of Pavia dataset as compared to Deep Belief Network (DBN), Autoencoder (AE) and Residual Net (ResNET) (Ozdemir and Polat, 2020). The downside of CNN is that the training requires a very large dataset, is computationally and memory expensive.

These challenges have not prevented researchers to explore the possibility of using CNN on small satellites. Buonaiuto et al. (2017) used a CNN model deployed on Nvidia Jetson TX1 for the purpose of identifying small satellites in orbit. Manning et al. (2018) used CNN deployed on Xlinix Zynq7020 FPGA to classify images taken from their ISS SHREC payload. The paper explored the possibility of deploying the model in small satellite platforms. Arechiga et al. in (2018) implemented CNN on Nvidia Jetson TX1 for the purpose of on-board image processing for small satellites. Greenland et al. (2018) used Zynq7020 FPGA to classify clouds for small satellite on-board applications. Bappy and Siddique (2019) proposed to use "onboard deep neural computing and machine learning model to analyse and process multispectral images" for a 3U CubeSat. Adam Braun (2018) investigated CubeSat hardware that can run CNN models. Giuffrida et al. (2020) proposed CloudScout CNN deployed on a Myriad 2 Visual Processing Unit for on-board cloud detection on hyperspectral images. The payload is called HyperScout-2 and is designed for Φ-SAT-2 challenge from European Space Agency (ESA). However, none of these literatures specifically target 1U CubeSats.

The past research closest to this paper is by Zhang et al. (2018) which used CNN model deployed on ARM9 microcontroller (MCU) to detect cloud for CubeSat applications. The limitation of the research is (1) the CNN model is not trained on dedicated CubeSat imagery dataset (2) transfer learning through MobU-Net instead of building a customized, light CNN architecture specifically designed based on 1U CubeSat constraints and the use of (3) ARM9 MCU which is an older version to the popular ARM Cortex MCUs.

1.3. Paper purpose

The purpose of the present paper is to provide a novel method that prioritizes quality data over quantity without changing the constraints of size, power, volume, downlink and pointing requirements imposed by a 1U CubeSat such as BIRDS-3. The method uses an ultralight CNN called CubeSatNet to filter "good" images trained on an entirely new CubeSat image dataset. The present paper's novelty is in the following points; a training dataset of 60,000 augmented images collected from all available sources and a new CNN architecture with 90% accuracy that is small enough to fit inside an ARM Cortex MCU. The CNN has

Table 3
Source for CubeSat imagery dataset.

Parameter	Source	Count
CubeSat imagery	Internet/Internal Database	1106
Nadir pointed	Sentinel 3A	100
Horizon images	ISS	196
Total		1402

the highest F1 score as compared to SVM, DBN and AE classification methods for CubeSat imagery. This paper benefits the small satellite, especially CubeSat community, by automatically selecting useful image data in orbit and cutting down operation time significantly.

1.4. Paper outline

The paper consists of 4 sections including the introduction. Section 2 describes the methodology of creating the dataset, selecting hardware, building the CNN and training the model. Section 3 describes the results followed by a discussion of those results. Finally, Section 4 concludes the paper with future study items.

2. Methodology

This section provides details on how an entirely new CubeSat dataset of 60,000 images is created, what framework is selected for the training, what MCU is selected to load the model, how a new CubeSatNet CNN architecture is designed and the process in which the network is trained and optimized.

2.1. Data collection

An extensive search through the web is conducted to collect images taken by CubeSat teams from across the globe to create the training dataset. The search targeted social media sources including Instagram, Facebook and Twitter accounts, official websites, blogs, and third-party websites. Data from Kyutech's small satellite image database is also integrated. After exhausting all the sources, additional images are extracted from live feed of International Space Station (ISS). These are similar to the Earth horizon images taken by CubeSats. Nadir-pointed thumbnail images from Sentinel 3A are created from Google Earth Engine (Google, 2019a). Table 3 shows the number of images obtained from each category. A total number 1402 of pre-augmented images are collected.

To test the accuracy of the model, a separate dataset is needed. BIRDS-3 on-orbit image data is used to create the test dataset. A total of thirty images that had been collected for the first three months are included. For both the training and test dataset, each image is labelled "good" or "bad". Any image that is suitable for publishing in social or mainstream media is labelled as "good". Any image that is cloudy, space-faced, sunburnt or saturated is labelled as bad. Fig. 4 shows examples of test dataset for the two labels. 822 are classified as "good" and 581 are classified as "bad".

2.2. Data conditioning and augmentation

Any logos, spots or image defects on the dataset are manually corrected using a standard image processing software. A python script based on OpenCV library is used to resize images to 100 pixels by 100 pixels and output them on .jpg. The conditioned data is then fed through a data augmentation python script based on Augmentor library created by Bloice et al. (2019). Resulting images are generated from combined random application of rotation, flipping, distortion and skewing. Table 4 shows the parameters for each case. A total of 60,000 images are generated from the process where 30,000 are "good" and 30,000 are "bad". Fig. 5 shows some examples of the image augmentation result.

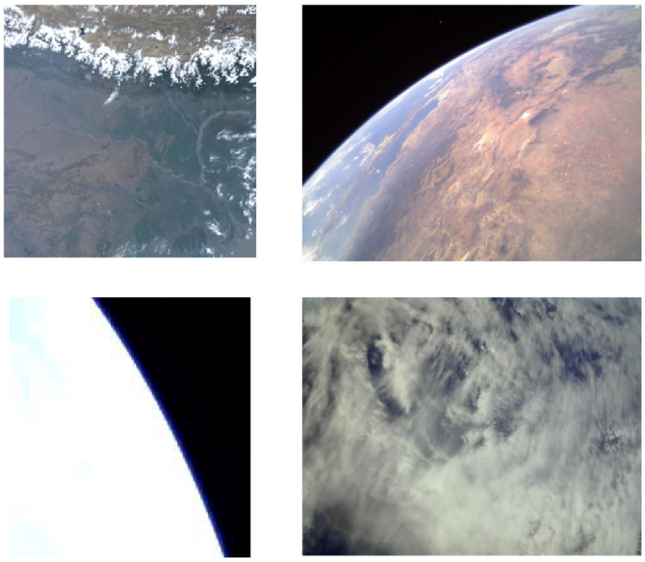


Fig. 4. Top images illustrate "good" while the bottom two images illustrate "bad".

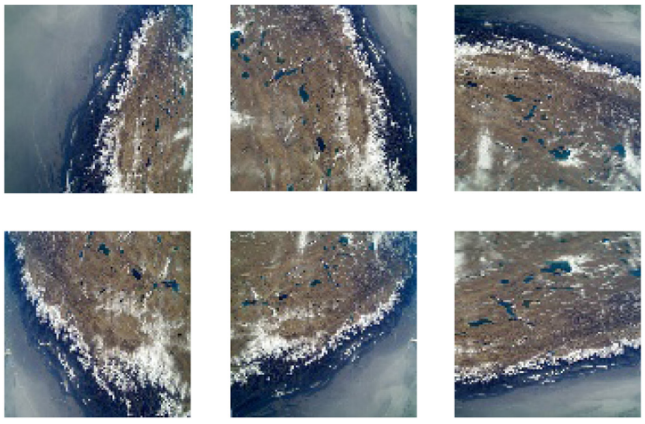


Fig. 5. Examples of data augmentation through combined implementation of rotation, flipping, distortion and skewing.

Table 4
Augmentation values for each parameter applied to images.

Parameter	Probability
Flip	0.5
90 Degree rotation	0.5
Distortion	0.4
Skewing (Standard)	0.4
Skewing (Left to Right)	0.5
Skewing (Tilt)	0.6

2.3. Model framework

Tensorflow (TF) 2.0 framework with Keras high level application programming interface (API) is used to train the model. TF is developed by Google and is open access. Keras has simplified python coding to access TF's library. Google Colaboratoy (Colab) is a free cloud service for ML education and research. Colab is based on Jupyter. "Jupyter is run on browsers and is open source that integrates interpreted languages, libraries and tools for visualization" (Carneiro et al., 2018). Colab provides a NVIDIA Tesla K80 12 GB with 2496 CUDA core GPU accelerated runtime to train ML models on TF and Keras (Carneiro et al., 2018). A maximum training time of twelve hours is allowed before the runtime is reset. The paper uses TF framework on Colab platform to train as the resource is free and the training time is short.

Table 5
STM32 processors used on CubeSats.

CubeSat	Type	Developer	Launch year	MCU
ESTCube-1	1U	Uni. of Tartu	2013	STM32F1
SNUSAT-1/1b	2U	Seoul National Uni.	2017	STM32F4
FOX-1C	1U	AMSAT	2018	STM32L1
PW-Sat2	2U	Warsaw Uni. of Tech.	2018	STM32F1
SNUSAT-2	3U	Seoul National Uni.	2018	STM32F4
PicSat	3U	Observatoire de Paris	2018	STM32F3
KrakSat	1U	AGH Uni. of Sci. and Tech., SatRevolution S.A.	2019	STM32xx
M6P	6U	Nanoavionics	2019	STM32H7

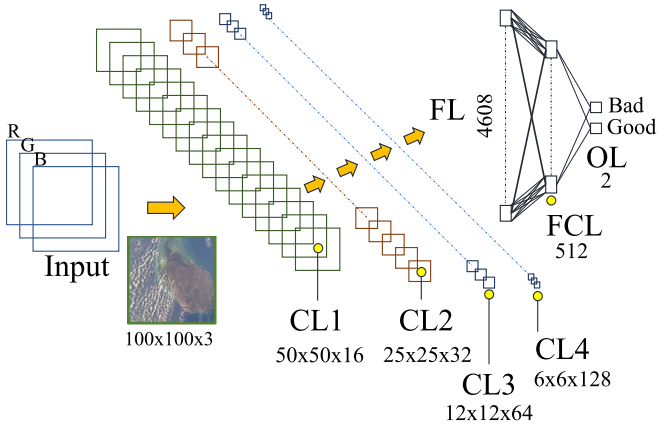


Fig. 6. CNN Architecture of CubeSatNet_v1.

2.4. Inference hardware

The trained model has to be deployed on an MCU. The MCU should be such that it fulfils the constraints imposed by power requirements of a 1U bus system of BIRDS-3. The MCU should also be general enough that it can be purchased easily, has technical support, is well documented and has a large user base. According to ASPENCORE 2017 Embedded Market Study published in April 2017, 2/3rd of the individuals surveyed were considering using 32 bit STM32 from STMicroelectronics for their next embedded project (AspenCore Global Media, 2017). Survey in 2015 had the same trend before (Quinnell, 2015). STM32 MCUs are known for its performance, fast clock speed, larger internal memory and low power consumption. Table 5 shows some of CubeSats which have used STM32 in either their bus or subsystem. In April 1, 2019, Nanoavionic's M6P mission carried SatBus 3C2 on-board which has a 400 MHz STM32H7 ARM Cortex M7 MCU (NanoAvionics, 2018). OpenMV Cam H7 processor is selected to run the model as (1) has the same STM32H7 processor with 1 MB internal memory and 2 MB flash memory (2) consumes 170 mA at 3.3 V (3) has direct technical support (4) has TF support and (5) has user friendly Integrated Development Environment (IDE).

2.5. CNN model architecture

The two CNN models explained in detail are named as CubeSatNet_v1 and CubeSatNet_v2. Fig. 6 shows that the CubeSatNet_v1 architecture has four Convolutional Layers (CL), a Flattening Layer (FL), a Fully Connected Layer (FCL) and an Output Layer (OL). CL has kernel of 3 × 3, same padding and ReLu activation layer. After each CL, a max-pooling layer of matrix 2 × 2 and stride 2 is applied. 16, 32, 64 and 128 features are extracted as the image passes through each subsequent CL. Feature size is reduced by the pooling layer to 50 × 50, 25 × 25, 12 × 12 and 6 × 6 respectively. Dropouts are imposed after FL and FCL. Softmax function is implemented in OL to provide probabilities for each class. During training, categorical cross-entropy calculates the loss and Adam optimizes during backpropagation. Table 6 summarizes each layer and its parameters. Table 7 enlists properties for each convolutional layer. Tables 10 and 11 later in Section 2.6. shows how accuracies change with changes in base reference values for inputs and architecture.

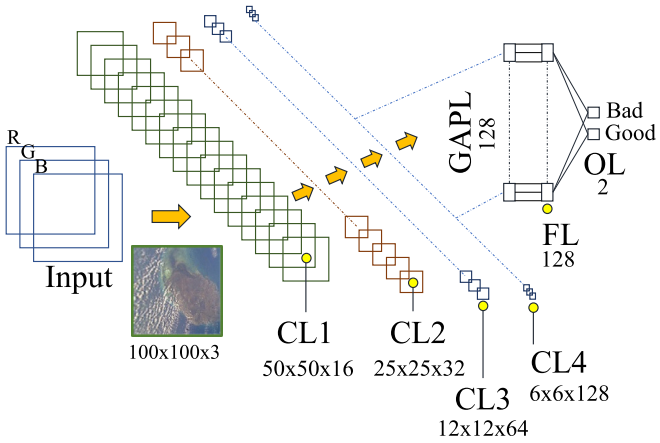


Fig. 7. CNN Architecture of CubeSatNet_v2.

Table 6
Layer definition for CubeSatNet_v1.

Layer	Type	Shape	Param #
CL1	Conv	(100,100,16)	448
CL2	Conv	(50,50,32)	4640
CL3	Conv	(25,25,64)	18,496
CL4	Conv	(12,12,128)	73,856
FL	Flattening	4608	0
FCL	Dense	512	2,359,808
OL	Output	2	1026
Total parameters			2,458,274

Table 7
Kernel, activation and pooling properties of CubeSatNet_v1.

Parameter	Type	Padding	Stride
Kernel	3 × 3	Same	1
Activation	ReLu	–	–
Pooling	Max (2 × 2)		2

Table 8
Layer definition for CubeSatNet_v2. FL has been replaced by GAPL.

Layer	Type	Shape	Param #
CL1	Conv	(100,100,16)	448
CL2	Conv	(50,50,32)	4640
CL3	Conv	(25,25,64)	18,496
CL4	Conv	(12,12,128)	73,856
GAPL	**Global Avg.**	**128**	**0**
FL	Flattening	128	0
OL	Output	2	258
Total parameters			97,698

Table 7 shows that there are almost 2.5 million parameters that need to be trained in CubeSatNet_v1. The network size is proportional to the number of parameters. To reduce the number of trainable parameters, CubeSatNet_v2 has Global Average Pooling Layer (GAPL) instead of FCL. The total number of parameters is now under 100,000. Fig. 7 shows the CNN architecture and Table 8 enlists the layers of CubeSatNet_v2.

2.6. Training

As mentioned on Section 2.3., model training is done on TF framework on Colab platform. The training dataset is further divided into

Table 9
Summary of the total number of parameters for hyper-parameter tuning.

Hyper-parameter	Values	No.
Batch Size (BS)	32, **64**, 128, 512, 1024, 2048	7
Learning Rate (LR)	0.1, 0.01, 0.001, **0.0001**, 0.00001	5
Dropout (DO)	0.2, **0.3**, 0.4, 0.5	4
Total search space	BS*LR*DO	140

Table 10
Test accuracy of different CubeSatNet versions with different input size of image and layers.

CNN model	Layer selection	Input size	Parameters	Accuracy
CubeSatNet_v1	CLx4, FCL	100 × 100	2,458,274	90.12%
CubeSatNet_v2	**CLx4, GAPL**	**100 × 100**	**97,698**	**90.47%**
CubeSatNet_v3	CLx4, GAPL	75 × 75	97,698	88.06%
CubeSatNet_v4	CLx4, GAPL	50 × 50	97,698	88.04%
CubeSatNet_v5	CLx3, GAPL	50 × 50	23,714	86.92%
CubeSatNet_v6	CLx2, GAPL	25 × 25	5154	85.05%

Table 11
Test accuracy of different CubeSatNet_v2 with different depth values for CL.

CubeSatNet_v2	CL1	CL2	CL3	CL4	Parameters	Test accuracy
Type A	4	8	16	32	6282	89.88%
Type B	8	16	32	64	24,658	89.47%
Original	**16**	**32**	**64**	**128**	**97,698**	**90.47%**

training and validation dataset in 80:20 ratio. 48,000 images are used for training while 12,000 is used for validation. Three key hyper-parameters are selected for optimization; dropout (DO) before OL, batch size (BS) of images during training and Adam's learning rate (LR) for backpropagation. Table 9 shows the number of search parameters for DO, BS and LR are 4, 7 and 5 respectively. The total number parameters in the search space is 140. Liashchynskyi and Liashchynskyi (2019) states that grid search is a better approach than genetic algorithm when search space is small. For BS, LR and DO, the optimum values through grid search are 64, 0.0001 and 0.3 respectively. The values have been bolded in Table 9. Each training took about an hour using the platform explained in 3.1.3. TF model is generated in the process.

Fig. 8 shows that the model achieved a final accuracy of 90.47% at 43 epochs. After 43 epochs, the plots for training and validation diverge. A red line is marked on Fig. 8 to illustrate the diverging point which is the optimum value. The divergence after the red line in training and validation losses is because of overfitting. Bigger fluctuations are observed as the overfit model is sensitive to minor changes in the training data.

Table 10 shows a comparison of CubeSatNet_v2 with different versions of CNN model. CubeSatNet_v1 took 100 × 100 input with 3 channels (RGB). CubeSatNet_v2 reduced the number of parameters significantly while improving accuracy when FCL is replaced by GAPL. However, with progressive removal of CL and reduction of input size showed accuracy to reduce. Table 11 shows how original selection of base reference values of 16, 32, 64, 128 feature extraction for subsequent CL in CubeSatNet_v2 showed highest accuracy. Reducing depth of CL showed lower accuracies as documented in Type A and Type B where 4, 8, 16, 32 and 8, 16, 32, 64 respectively are selected as the depth of CL. The final model of CubeSatNet_v2 is about 104 kB. The details of the results are provided in Section 3.

3. Results and discussion

This section shows the performance of quantized CubeSatNet_v2 model. A comparative performance analysis between CubeSatNet, SVM, DBN and AE based on accuracy and F1 score is given. All algorithms are tested on a new test dataset that is created by taking images from a CubeSat that is currently in orbit. A discussion is provided on a hypothetical scenario where CubeSatNet_v2 is applied on-orbit on BIRDS-3 1U CubeSats.

Table 12
Confidence value of the image that flipped classification while changing models.

Bad			Good		
TF	TFlite	qTFlite	TF	TFlite	qTFlite
0.543	0.543	**0.497**	0.456	0.456	**0.502**

3.1. Quantization and model performance

A separate dataset of first thirty on-orbit images from BIRDS-3 are used to create the test dataset. As the dataset is not used for training or validation, the performance of the model can be better scrutinized. The testing has been done on three different TF models; the original generated from training, a Tensorflow lite (TFlite) model and an 8-bit quantized Tensorflow lite (qTFlite) model through post-training quantization process. TFlite is designed for inference on embedded systems. The 8-bit qTFlite is the actual model that is deployed on the MCU of the CubeSat. The TFlite conversion and the quantization process is done following the documentation provided by Google (Google, 2019b).

The visual representation of the image classification is shown in Fig. 9. TF is firstly converted to TFlite. TF and TFlite have same test accuracy of 86.67% and is about 4% lower than the validation accuracy during training. Interestingly, qTFlite model performs better than TFlite and TF with 90% accuracy (see Fig. 9). This is unexpected as factorial reduction in size should have maintained or reduced the accuracy. Table 12 shows how the probability calculation from output layer (confidence values) changed for the image boxed in Fig. 9. The model's prediction for "bad" and "good" for the image is borderline. Initially, the model is slightly more confident that the image is "bad". Post-training quantization changed the value in such a way that qTFlite is slightly more confident that the same image is "good". Since the original classification was incorrect, the qTFlite classified it as correct. This explains the increase in accuracy. The inference time for qTFlite model is about 0.25 s per image. The final size of CubeSatNet_v2 qTFlite is 104 kB. The model is small enough to fit inside proposed MCU.

The confusion matrix is presented on Table 13 along with performance results. The total number of images classified is 30. Among them, 17 are True Positives (TP) where CubeSatNet_v2 qTFlite correctly classified "bad" images. Likewise, 1 is False Negative (FN) where the CNN incorrectly classified as "good" image. The CNN has 2 False Positives (FP) image where "good" images are classified as "bad" and 10 True Negative (TN) where images are correctly classified as "good". F1 score is calculated to be 0.92. The maximum achievable score is 1. The accuracy (A) of the model on the test dataset is 90%.

3.2. Comparative performance

For simplicity, CubeSatNet_v2 qTFlite is referred to as CNN in this section. The CNN performance must be compared to other machine learning classification methods. The CNN model is compared to SVM, DBN and AE trained on the same dataset. SVM is built by Kyutech's BIRDS-4 team. The model is currently part of the ICU on-board BIRDS-4's 1U CubeSat constellation which is scheduled to launch late 2020. The size is 500 kB, takes greyscale input and is trained in Matlab. DBN consists of two hidden layers and is based on the DBN classification library designed by DBNAlbert (albertbup, 2017). To minimize size, the DBN takes greyscale input, is limited to 325 kB and uses TF framework. AE is unsupervised and the design algorithm is based on Ardamavi (ardamavi, 2018). The size is 621 kB, takes RGB input and is trained using high level Keras API to access TF framework.

Table 14 lists out the results on BIRDS-3 test dataset mentioned in Section 3.1. Among the four models, CNN has the smallest size with 104 KB. CNN also takes RGB input like AE but opposed to greyscale inputs of SVM and DBN. CNN shows highest test accuracy (A) with 90% followed by DBN, SVM and AE with 76.67%, 73.33% and 66.67% respectively. CNN displayed the highest recall (R) and precision (P).

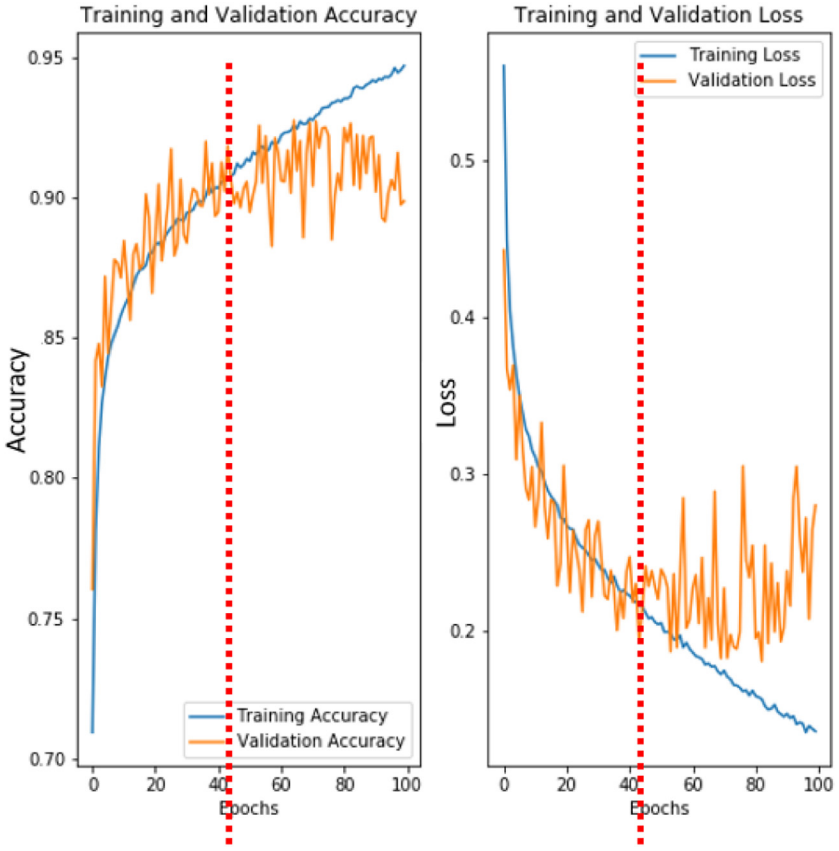


Fig. 8. Training and validation accuracy on the left and loss on the right. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

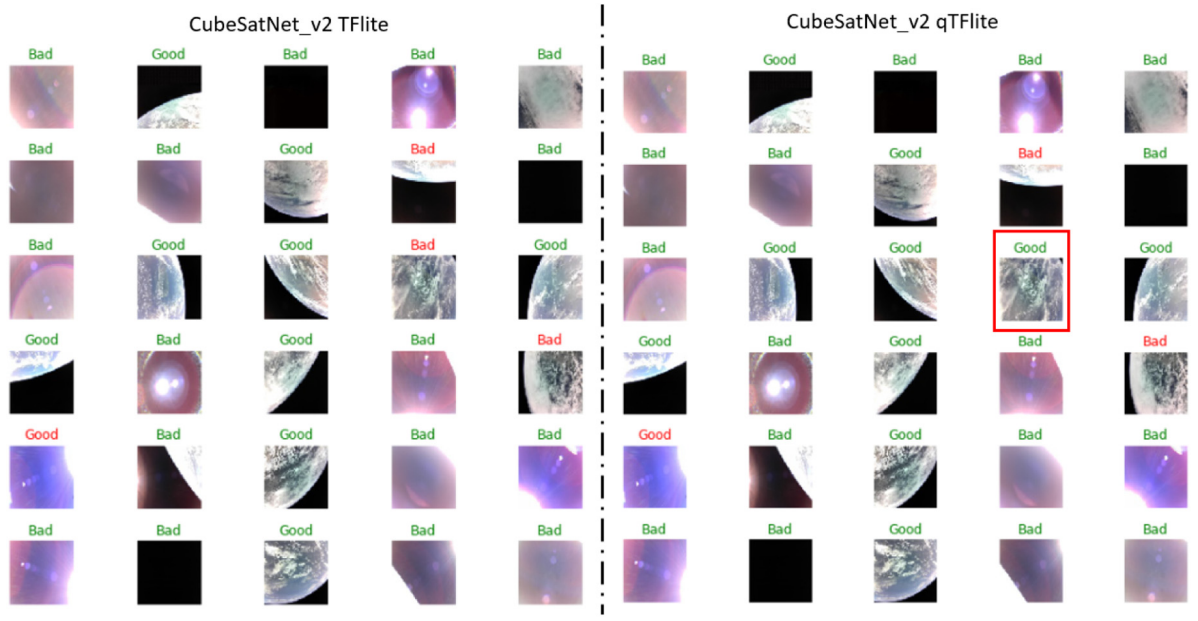


Fig. 9. CubeSatNet_v2 TFlite and qTFlite model's performance on BIRDS-3 images. Red shows changes in classification. TFlite's accuracy is 86.67% while qTFlite's accuracy increased to 90% which is unexpected. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The F1 score for CNN is 0.92. DBN, SVM and AE have F1 score of 0.80, 0.74 and 0.79 respectively. The results show that CNN outperforms DBN, SVM and AE for CubeSat image classification.

3.3. Model application discussion

Fig. 10 shows the images that would have been downlinked if hypothetically the CNN is applied on BIRDS-3 from deployment. All 18

Table 13
Confusion matrix and performance metrics for CubeSatNet_v2 qTFlite.

Confusion matrix	Actual		Performance on BIRDS-3 Test Dataset			
Predicted	Bad	Good	Accuracy (A)	Recall (R)	Precision (P)	F1 Score
Bad	17 [*TP*]	2 [*FP*]	$\frac{TP+TN}{TP+FN+FP+TN}$	$\frac{TP}{TP+FN}$	$\frac{TP}{TP+FP}$	$2 * \frac{P*R}{P+R}$
Good	1 [*FN*]	10 [*TN*]	0.90	0.94	0.90	**0.92**

Table 14
Summary of changes of size and accuracy of different models. CNN outperforms all other methods.

Model Information				Confusion Matrix Values				Performance on BIRDS-3 Test Dataset				
Model	Framework	Size	Input	TP	FP	FN	TN	Total	A	R	P	F1
SVM	Matlab	500 kB	B&W	15	5	3	7	30	73.33%	0.83	0.75	0.79
CNN	**Tensorflow**	**104 kB**	**RGB**	**17**	**2**	**1**	**10**	**30**	**90%**	**0.94**	**0.90**	**0.92**
DBN	Tensorflow	324 kB	B&W	14	3	4	9	30	76.67%	0.78	0.82	0.80
AE	Tensorflow	621 kB	RGB	14	6	4	6	30	66.67%	0.78	0.70	0.74

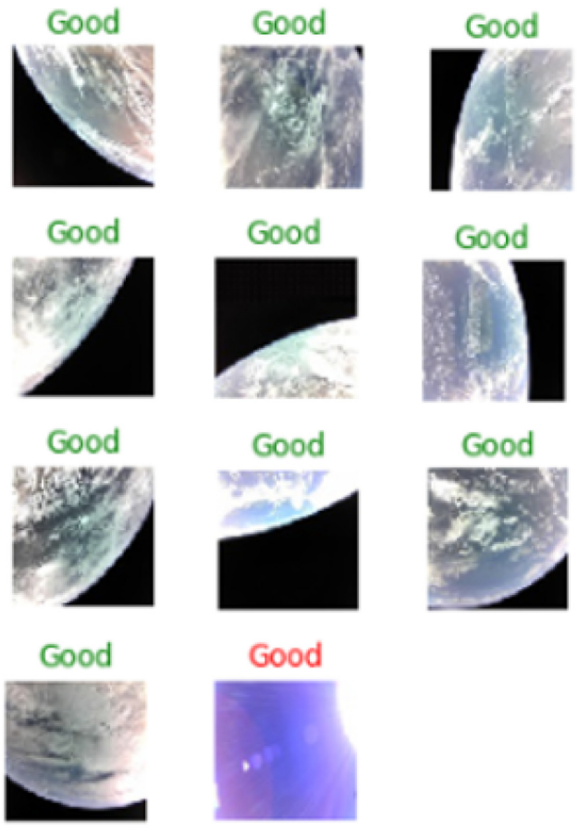


Fig. 10. Shows images that would have been downlinked had the CNN been applied on BIRDS-3 in first three months of operation.

images classified as "bad" would have been deleted. In that 2 actually "good" images would have been lost. The remaining 11 images would have been taken about a month to downlink cutting down the operation time to by 2/3rd. Out of the 11 images downlinked, 1 would have been incorrectly classified as good. The CNN has the potential to save operation time and manual work while significantly improving the quality of image downlinked.

4. Conclusion

This paper presented an innovative method to tackle the limited data downlink capability of a 1U CubeSat. An ultralight CNN architecture called CubeSatNet is proposed and trained on a novel CubeSat image dataset of 60,000 augmented images to prioritize quality image data for downlink. The final model is just over 100 kB in size and is small enough to load on an ARM Cortex MCU and has an accuracy of 90%. CubeSatNet had the highest F1 score as compared to SVM, DBN and AE trained to classify CubeSat images. Test is done on first thirty on-orbit images from Kyutech's BIRDS-3 CubeSats. The results showed that, if implemented, operation time could be cut by about 2/3 while significantly improving on quality of image received.

Future work will be to improve the baseline training accuracy by increasing the dataset and improving the architecture of the CNN. Images taken from recently launched CubeSats will be aggregated into the dataset including recent images from BIRDS-3. Additional convolutional layers can also be placed to extract features better. The challenge will be to reach the maximum accuracy while staying within the allowable MCU running memory size.

CRediT authorship contribution statement

Abhas Maskey: Conceptualization, Methodology, Visualization, Software, Validation, Writing, Editing. **Mengu Cho:** Supervision, Writing - reviewing & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

albertbup, 2017. A Python implementation of Deep Belief Networks built upon NumPy and TensorFlow with scikit-learn compatibility [WWW Document]. URL https://github.com/albertbup/deep-belief-network (accessed 7.22.20).

ardamavi, 2018. Using Autoencoders for classification as unsupervised machine learning algorithms with Deep Learning. [WWW Document]. URL https://github.com/ardamavi/Unsupervised-Classification-with-Autoencoder (accessed 7.22.20).

Arechiga, A.P., Michaels, A.J., Black, J.T., 2018. Onboard image processing for small Satellites. In: Proceedings of the IEEE National Aerospace Electronics Conference, NAECON. IEEE, pp. 234–240. http://dx.doi.org/10.1109/NAECON.2018.8556744.

AspenCore Global Media, 2017. 2017 Embedded Markets Study. EE Times.

Bappy, M.I., Siddique, S., 2019. AI-OBC: Conceptual Design of a Deep Neural Network Based Next Generation Onboard Computing Architecture for Satellite Systems.

Bloice, M.D., Roth, P.M., Holzinger, A., 2019. Biomedical image augmentation using Augmentor. Bioinformatics.

Braun, A.D., 2018. Investigation of Deep Neural Network Image Processing for Cubesat Size Satellites.

Buonaiuto, N., Kief, C., Louie, M., Aarestad, J., Zufelt, B., Mital, R., Mateik, D., Sivilli, R., Bhopale, A., 2017. SSC17-WK-56 Satellite identification imaging for small satellites using NVIDIA. In: Small Satell. Conf.

Carneiro, T., Da Nóbrega, R.V.M., Nepomuceno, T., Bian, G.-B., De Albuquerque, V.H.C., Reboucas Filho, P.P., 2018. Performance Analysis of Google Colaboratory as a Tool for accelerating deep learning applications. IEEE Access 6, 61677–61685.

Chien, S., Doubleday, J., Thompson, D.R., Wagstaff, K.L., Bellardo, J., Francis, C., Baumgarten, E., Williams, A., Yee, E., Stanton, E., Piug-Suari, J., Davies, M., 2017. Onboard autonomy on the intelligent payload experiment CubeSat mission. J. Aerosp. Inf. Syst. 14, 307–315. http://dx.doi.org/10.2514/1.I010386.

Cristea, O., Dolea, P., Dascăl, P.V., 2009. S-band ground station prototype for low-earth orbit nanosatellite missions. Telecomunicatii 6, 4–71.

EnduroSat, 2019a. UHF Transceiver II CubeSat Communication | CubeSat by EnduroSat [WWW Document]. URL https://www.endurosat.com/cubesat-store/cubesat-communication-modules/uhf-transceiver-ii/ (accessed 12.20.19a).

EnduroSat, 2019b. S-Band Transmitter - CubeSat Communication Module | EnduroSat [WWW Document]. URL https://www.endurosat.com/cubesat-store/cubesat-communication-modules/s-band-transmitter/ (accessed 10.12.19b).

EnduroSat, 2019c. X-Band Transmitter CubeSat Communication Module | EnduroSat [WWW Document]. URL https://www.endurosat.com/cubesat-store/cubesat-communication-modules/x-band-transmitter/ (accessed 12.20.19c).

Feruglio, L., Corpino, S., 2017. Neural networks to increase the autonomy of interplanetary nanosatellite missions. Robot. Auton. Syst. 93, 52–60.

Giuffrida, G., Diana, L., de Gioia, F., Benelli, G., Meoni, G., Donati, M., Fanucci, L., 2020. CloudScout: A deep neural network for on-board cloud detection on hyperspectral images. Remote Sens. 12 (2205), http://dx.doi.org/10.3390/rs12142205.

Google, 2019a. Google Earth Engine [WWW Document]. URL https://earthengine.google.com/ (accessed 12.20.19a).

Google, 2019b. TensorFlow Lite 8-bit quantization specification | TensorFlow [WWW Document]. URL https://www.tensorflow.org/lite/performance/quantization_spec (accessed 12.20.19b).

Greenland, S., Ireland, M., Kobayashi, C., Mendham, P., Post, M., White, D., 2018. Design & prototyping of a Minaturised forwards looking imager using deep learning for responsive onboard operations. In: The 4S Symposium. pp. 1–9.

Hedjazi, M.A., Kourbane, I., Genc, Y., 2017. On identifying leaves: A comparison of CNN with classical ML methods. In: 2017 25th Signal Processing and Communications Applications Conference (SIU). IEEE, pp. 1–4.

Heidt, H., Puig-Suari, J., Moore, A., Nakasuka, S., Twiggs, R., 2000. CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation.

Islam, K.T., Raj, R.G., Al-Murad, A., 2018. Performance of SVM, CNN, and ANN with BoW, HOG, and Image Pixels in Face Recognition. In: 2nd Int. Conf. Electr. Electron. Eng. ICEEE 2017. pp. 1–4. http://dx.doi.org/10.1109/CEEE.2017.8412925.

JAXA, 2019. Pre-launch operations begin for deploying four CubeSats from ISS : Experiment - International Space Station - JAXA [WWW Document]. URL https://iss.jaxa.jp/en/kiboexp/news/20190304_birds.html (accessed 7.21.20).

Jirawattanaphol, A., Kurahara, N., Cho, M., 2016. Design and development of ground station network for cubesats constellation, joint global multi-nation birds. Proceedings of 60th JSASS Space Science and Technology Conference. 60, p. 6.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. pp. 1097–1105.

Kumar, B., Dikshit, O., Gupta, A., Singh, M.K., 2020. Feature extraction for hyperspectral image classification: a review. Int. J. Remote Sens. 41, 6248–6287. http://dx.doi.org/10.1080/01431161.2020.1736732.

Lee, C.-C., 1991. Intelligent control based on fuzzy logic and neural net theory.

Liashchynskyi, Petro, Liashchynskyi, Pavlo, 2019. Grid search, Random Search, genetic algorithm: A big Comparison for NAS. arXiv Prepr. arXiv:1912.06059.

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., Johnson, B.A., 2019. Deep learning in remote sensing applications: A meta-analysis and review. ISPRS J. Photogramm. Remote Sens. 152, 166–177. http://dx.doi.org/10.1016/j.isprsjprs.2019.04.015.

Manning, J., Langerman, D., Ramesh, B., Gretok, E., Wilson, C., George, A.D., Mackinnon, J., Crum, G., 2018. Machine-learning space applications on SmallSat platforms with TensorFlow. In: 32nd Annu. AIAA/USU Conf. Small Satell.. pp. 1–8.

Marfu'ah, N.J.L., Kurniawardhani, A., 2020. Comparison of CNN and SVM for Ship detection in satellite Imagery. p. 1, AUTOMATA.

Maskey, A., Monowar, M.I., Cho, M., 2019. The first thousand nanosatellites: A study of operating systems used. In: 32nd International Symposium on Space Technology and Science & 9th Nano Satellite Symposium. Fukui. pp. 1–6.

NanoAvionics, 2018. High-performance Multi-purpose 6U nano-satellite Platform.

Ozdemir, A., Polat, K., 2020. Deep learning applications for hyperspectral imaging: A systematic review. J. Inst. Electron. Comput. 2, 39–56. http://dx.doi.org/10.33969/jiec.2020.21004.

Palo, S., Connor, D.O., Devito, E., Kohnert, R., 2012. SSC14-IX-1 Expanding CubeSat Capabilities with a Low Cost Transceiver.

Parikh, H., Patel, S., Patel, V., 2020. Classification of SAR and PolSAR images using deep learning: a review. Int. J. Image Data Fusion 11, 1–32. http://dx.doi.org/10.1080/19479832.2019.1655489.

Quinnell, R., 2015. Embedded Markets Study: Changes in Today's Design, Development & Processing Environments. ARM TechCon.

Rash, J.L., Dent, C.P., 1989. Space applications of artificial intelligence. In: Proceedings of the Annual Goddard Conference. Greenbelt, MD, May 16, 17, 1989.

Silva, M.A.C., Shan, M., Cervone, A., Gill, E., 2019. Fuzzy control allocation of microthrusters for space debris removal using CubeSats. Eng. Appl. Artif. Intell. 81, 145–156.

Sonoke, T., Demura, H., 2019. Identification and classification of ships SAR data using machine learning. In: International Symposium on Space Technology and Science. Fukui.

Thompson, D.R., Altinok, A., Bornstein, B., Chien, S.A., Doubleday, J., Bellardo, J., Wagstaff, K.L., 2015. Onboard machine learning classification of images by a cubesat in Earth orbit. AI Matters 1, 38–40. http://dx.doi.org/10.1145/2757001.2757010.

Wang, H., Zhou, Z., Li, Y., Chen, Z., Lu, P., Wang, W., Liu, W., Yu, L., 2017. Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18F-FDG PET/CT images. EJNMMI Res. 7. http://dx.doi.org/10.1186/s13550-017-0260-9.

Zhang, Z., Xu, G., Song, J., 2018. CubeSat cloud detection based on JPEG2000 compression and deep learning. Adv. Mech. Eng. 10, 1–10. http://dx.doi.org/10.1177/1687814018808178.

Zheng, T., Bergin, M.H., Hu, S., Miller, J., Carlson, D.E., 2020. Estimating ground-level PM2.5 using micro-satellite images by a convolutional neural network and random forest approach. Atmos. Environ. 230, 117451. http://dx.doi.org/10.1016/j.atmosenv.2020.117451.