

## SSC18-WKVII-01

### Development of Attitude Sensor using Deep Learning

Sho Koizumi, Yuhei Kikuya, Kenichi Sasaki, Yuto Masuda, Yohei Iwasaki, Kei Watanabe,

Yoichi Yatsu, Saburo Matsunaga

Tokyo Institute of Technology

#554 Isikawadai 1st building, 2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8552, Japan; +81-90-4916-0628

koizumi@lss.mes.titech.ac.jp

#### ABSTRACT

A new method for attitude determination utilizing color earth images taken with COTS visible light camera is presented. The traditional earth camera has been used for coarse attitude determination by detecting the edge of the earth, and therefore it can only provide coarse and 2-axis information. In contrast, our method recognizes the ground pattern with an accuracy of sub-degrees and can provide 3-axis attitude information by comparing the detected ground pattern and the global map. Moreover, this method has advantages in the size, mass and cost of the detector system which consists of a cheap optical color camera and a single board computer. To demonstrate the method in space, we have developed a sensor system named “Deep Learning Attitude Sensor (DLAS)”. DLAS uses COTS camera modules and single board computers to reduce the cost. The obtained images are promptly analyzed with a newly developed real-time image recognition algorithms.

#### INTRODUCTION

In these days, Nano satellites or CubeSats, which are much smaller than conventional satellites, provide space access to a broader range of satellite users. Recently, the importance of small satellites has increased and more than 100 small satellites are launched in a year. The low development cost and frequent launch opportunities enable variety of challenging missions. For those reasons, many countries have been putting much effort into the research and the development in this new space technology region. Moreover, recent innovations in the area of IoT has realized low cost and high-performance computers at the same time, the size of which are pretty small enough to be the on-board computers for Nano satellites. Therefore, we propose a multifunctional low-cost 3-axis attitude sensor using commercially available devices. Our attitude sensor has advantages in the size, mass and cost of the detector system which consists of a cheap optical color camera and an onboard computer. High-performance onboard computers enable us to perform real-time image recognition using machine learning on orbit, which is one of the main idea of our 3-axis attitude sensor. The camera and the onboard computer can also be used for the other purpose, satellite's selfie or earth observation, as well as the attitude determination, which might be convenient for CubeSats with limited payloads. The method of machine learning has become one of the essential ways for classifying satellite images. However, most of satellite image classifications are based on a ground level data analysis after downlinking the telemetry. Those image processing are driven by a high-performance computer which has a multiple GPUs inside. The real time image

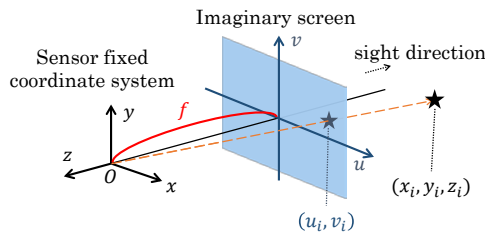
recognition makes it possible to reduce the downlink data drastically. In addition, the recognition result can also be used for disaster detection, ship detection and object detection such as for debris as well as being the key to the low-cost 3-axis attitude sensor. We named our 3-axis attitude sensor “Deep Learning Attitude Sensor (DLAS)” aboard a technology demonstration satellite of JAXA, which will be launched in 2019. DLAS uses COTS camera modules and single board computers to reduce the cost. The obtained images are promptly analyzed with a newly developed real-time image recognition algorithm customized for small satellite missions.

The attitude determination process consists of three steps: 2-axis attitude determination, image recognition, and 3-axis attitude determination. In this paper, we summarize the algorithm of our newly developed 3-axis attitude sensor focusing on the image recognition using machine learning.

#### METHODOLOGY

As we mentioned above, our 3-axis attitude determination consists of 3 steps. First step is 2-axis determination, which determine the nadir vector of a satellite. Next step, we utilize deep learning techniques to identify objects in satellite images and classify them into multiple categories such as sea, land or cloud. Finally, DLAS determines the attitude angle around the nadir vector using a simple matching technique for finding the classified land features from the global map, and then determines the complete attitude angle of the satellite.

To explain the algorithm, a coordinate system along with its origin must be chosen. Figure 1 shows the sensor fixed coordinate system  $(x, y, z)$ , which is the right-handed one. The origin of the coordinate system corresponds to the earth camera's focal point and its sight direction corresponds to  $-z$  direction. It is reasonable to assume that the earth camera is fixed on a spacecraft's body frame. Therefore, coordinate conversion from the sensor fixed coordinate system to body fixed coordinate system can be determined uniquely. A projection of objects in three-dimensional space to two dimensional pictures can be modeled as follows (Figure 1). A point in 3D space  $(x_i, y_i, z_i)$  would be projected to two-dimensional plane  $(u_i, v_i)$ .

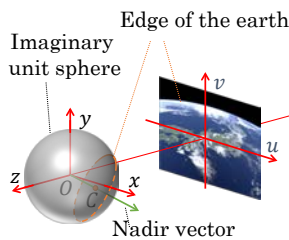


**Figure 1: Coordinate System**

### 2-axis Attitude Determination

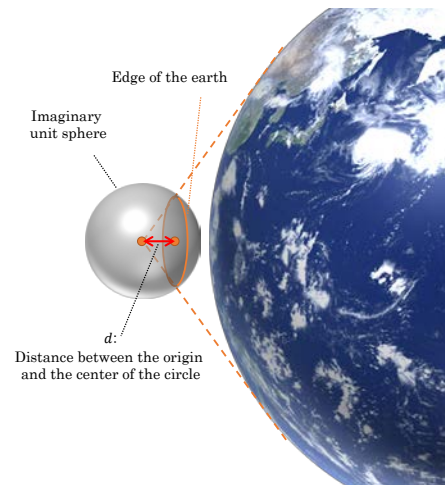
In the process of 2-Axis attitude determination, our sensor on a satellite automatically estimate the nadir direction vector by detecting the edge of the earth. Our DLAS project utilizes wide-field cameras consist of small CMOS sensors. The earth image took from the camera on a satellite may contain only a small part of the earth. Even in this case, our algorithm makes it possible to determine the nadir vector of the satellite.

We faced a couple of issues when we try to detect the edge of the earth. First, if the edge of the earth is not on the center of the picture, it is difficult to determine the center of the circle, the nadir direction, because of the distortion. Thus, we project the image to an imaginary unit sphere (Figure 2) and the circle on the unit sphere will not be distorted anymore, only if the ellipticity of the earth can be ignored.



**Figure 2: Nadir Direction Determination 1**

Detecting the edge between the earth and the space precisely was also one of the difficult problems. Sometimes earth images took from a satellite contains blackish land like sea, desert or lake. Since we use first-order-derivative for edge detection, sometimes misdetection occur depending on the image. We introduced the RANSAC (Random Sampling Consensus) algorithm [2] to accurately detect the nadir vector. The distance  $d$  between the origin and the circle on the imaginary unit sphere (see Figure 3) can be estimated from the diameter of the earth and the spacecraft altitude.



**Figure 3: The Distance d**

We, then, estimate the parameters as follows. The algorithm:

1. Selects 3 data (edge points) at random.
2. Determines a circle on the imaginary unit sphere.
3. Estimates the distance  $d$ .
4. Finds how many edge points fit the circle with parameter  $d$  within a given tolerance.
5. Repeats 1-4 several times and accepts the parameter if the number of the edge points are large enough.

### Image recognition using deep learning

After determining nadir vector, we introduced deep learning techniques to identify objects in satellite images and classify them into multiple classes. The classification of images using deep learning has developed rapidly in recent years. We developed our own software that could classify images under the conditions of limited performance and electricity. We propose simple neural network called single-hidden-layer Multi-Layer-Perceptron (MLP). We developed this

MLP network without using any framework. This network was mounted on the on-board computer “Raspberry Pi 3 Model B” and going to be launched as DLAS through the program of JAXA’s innovative satellite technology demonstration. These image recognition processes eliminate the disturbance such as cloud, which leads to the 3<sup>rd</sup> axis attitude determination.

### MLP

MLP is the simplest neural network inspired by the structure and function of the brain. Our image recognition algorithm is the combination of MLP and sliding window approach, which is common in the area of image recognition.

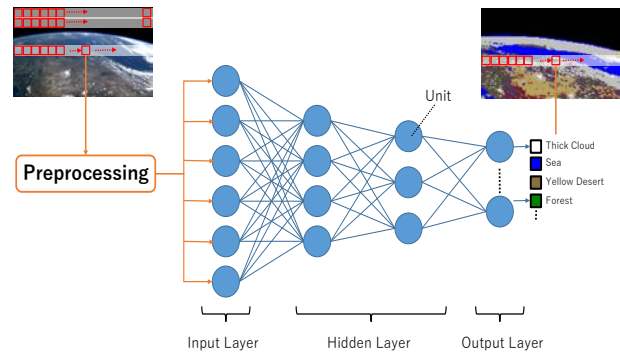
For the reasons mentioned below, we intentionally avoid using Convolutional Neural Networks (CNNs) in the process of sliding window.

1. CNN has a multiple convolution layer, which increase the processing time.
2. Since  $16 \times 16$  pixel window does not have much information, it does not make much difference between MLP and CNN.
3. In our DLAS project, we plan to perform relearning after acquiring the images took from the satellite. Therefore, we must be careful with the number of weights (parameter) in the network. The fewer parameter it becomes, the less command we needed.

### A Sliding Window Approach

The classification of images using deep learning has developed rapidly in recent years. The appearance of a computer with high-performance GPU has led to rapid developments in the area. Using these types of computer enabled pixel-based classification [3] in the image recognition area. However, the on-board computer we have selected in DLAS project is Raspberry Pi 3 Model B. The performance of this types of computer is usually much lower than a computer that has GPU inside. For those reasons, the network and algorithm on our DLAS project must be simple enough to achieve both speed and accuracy on the on-board computer. We therefore introduced sliding window approach in the process of image recognition using MLP network. A “window” is a fixed rectangular region that slides across an image. We determined the best size of a window,  $16 \times 16$  which is suitable for processing time and accuracy for attitude determination. For each of these windows, we apply an image classification using MLP to determine the land types such as sea, city, cloud, etc. (See figure 4). Since

we use visible light cameras, the number of input is  $16 \times 16 \times 3$  pixels. However, we performed some preprocessing such as selecting the feature or color conversion to reduce the processing time and to improve accuracy.



**Figure 4: Image Recognition Algorithm**

### Preprocessing

In order to remove irrelevant and redundant data in the image, we performed preprocessing before putting them into the neural network. Since we avoid using CNN, feature selection becomes a difficult problem. It is almost impossible to select feature which is the most useful for our dataset. Therefore, we selected a couple of features and tested them by using the ISS images in the course of training and classification (see figure 5).



**Figure 5 : Image took from the International Space Station [4]**

The feature we finally selected is the histogram. We calculate the histogram in each window and reduce the dimension by dividing it into 16 pieces. Therefore, the dimension of the input for the MLP network has reduced 256 to 16, which is suitable for the processing time, the parameter and the accuracy. In addition, we performed color conversion to the entire training image before the preprocessing.

### Training

JAXA’s innovative satellite will orbit low altitude of 500km around the earth and it is close to the orbit of ISS, which is around 408km. Hence, we use images sent from

the ISS as training data as shown in figure 5. In the process of training, we first need to create huge dataset from the ISS images. Dataset is an integral part of the field of deep learning and it influences the result of the training. In this network, we categorize the image into 10 classes as shown in Table 1. A total of 100000 data (window) was collected from the ISS's images. We then normalized each image and trained MLP network with these data. After we finished training, we passed the trained data to the classifier and execute classification on the on-board computer.

**Table 1: Dataset Description**

Class No.	Number of Training Data	Number of Testing Data	Classes
1	8000	2000	Thick Cloud
2			Thin Cloud
3			Foggy Cloud
4			Sea
5			Yellow Land(Desert)
6			Red Land(Desert)
7			Black Land(Desert)
8			Forest
9			City
10			Space

During the training, we trained a few different types of MLP (See Table 2). We changed the number of the nodes, and the color Space. We then evaluated these networks by generating Confusion Matrix.

**Table 2: Trained Model**

Model	Unit Number of Hidden Layer	Color Space
MLP (1 Hidden Layer)	20,40,60,100,120	RGB, HSV

**Classification**

In the classification process, the classifier repeats following 1-3 process for every single window. The algorithm:

1. Cuts out every  $16 \times 16$  pixels rectangular region.
2. Performs preprocessing.
3. Categorize each of the window into 10 classes.

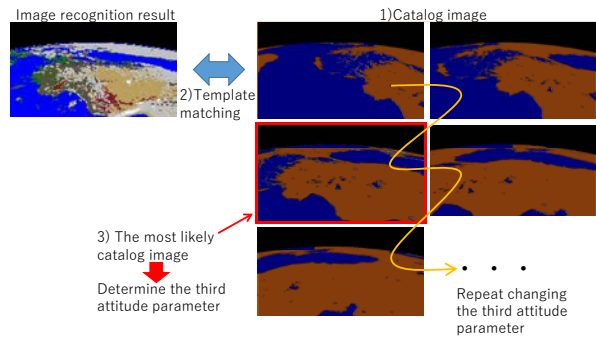
Since the resolution of the DLAS camera is  $3280 \times 2464$  pixel, the picture can be divided into 31570 windows. It is remarkable that the categorization of the 8M pixels image can be done only in 3 seconds on the computer "Raspberry Pi 3 Model B" which has no GPU inside. In the preprocessing, as we mentioned above, we extracted histogram from each window as a feature of the window.

It is also possible to change the size of a hidden layer, nodes and an output layer even if it is orbiting around the earth.

**3-Axis Attitude Determination**

We can finally determine the attitude angle around the nadir vector using a simple matching technique for finding the classified land features from the global map. We applied a template matching method to search and find the location of the classified image, which determines the attitude angle. The disturbances, such as clouds, interrupt us to introduce characteristic point matching-based methods such as SIFT. The procedure to determine the angle around the nadir vector are shown in Figure 6. The algorithm:

1. Generates imaginary pictures with all the possible angles around the nadir vector from the stored catalog earth image (figure 6). Position data from GPS and 2-axis attitude determination results enable us to set a limit to the relative attitude.
2. Calculates the accurate similarity of the image recognition result with the generated catalog images by using a template matching technique. In this process, we eliminate the disturbances such as cloud.
3. Ends by obtaining the angle with the highest value in all the generated catalog images.



**Figure 6: Procedure of 3 axis Attitude Determination**

## RESULTS

### The test result of the Image recognition

We first tested the classifier by using the ISS images. Table 3 shows the accuracy and the processing time. We used Raspberry Pi 3 Model B to calculate the processing time. As shown in Table 1, the calculation of the accuracy was conducted with testing data.

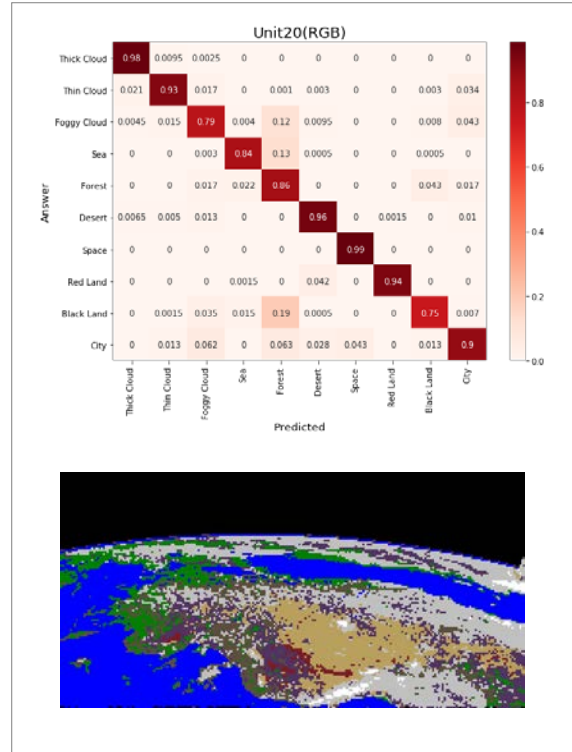
**Table 3: Test results of the Trained Model**

No.	PARAMETER			EVALUATION	
	Model	Unit Number	Color Space	Accuracy [%]	Time[s]
1	MLP (1 Hidden Layer)	20	RGB	98.85	2.83
2		40		90.23	3.46
3		60		90.35	4.18
4		80		90.56	4.89
5		100		90.55	5.60
6		120		90.50	6.18
7	MLP (1 Hidden Layer)	20	HSV	91.64	3.04
8		40		91.99	3.62
9		60		92.06	4.29
10		80		92.2	4.97
11		100		92.2	5.71
12		120		92.17	6.37

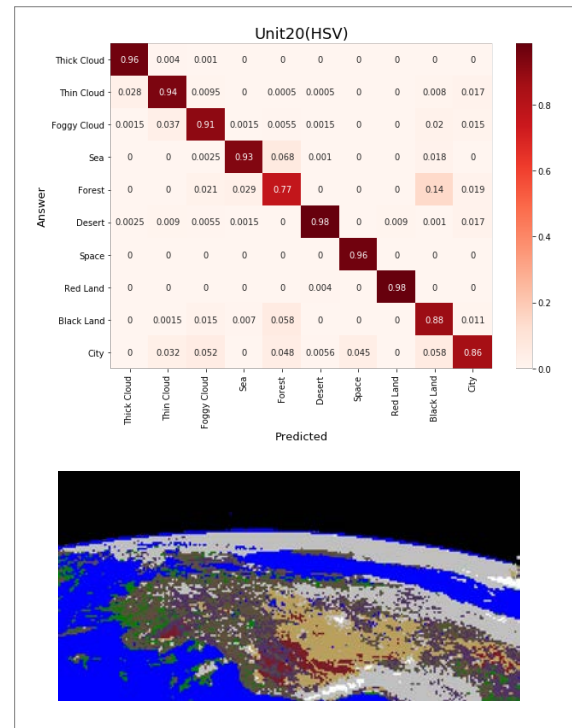
There is a clear difference between the two color spaces, RGB and HSV. Even though it requires processing time for the HSV classifier, the accuracy of that classifier excelled RGB classifier. We therefore selected the classifier MLP\_1 with the color space HSV and 20 units in a hidden layer. Figure 8 and Figure 9 are the confusion matrix and the scene classification result of the model 1 and 7 in Table 3. The tested image is the picture shown in Figure 5, which was not included in the training dataset.



**Figure 7: The Class Description in Figure 8 and Figure 9**

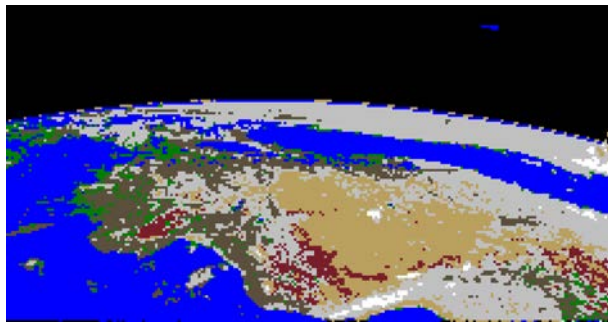


**Figure 8 : Confusion matrix and the result of scene classification by the model No.1 in Table 3**

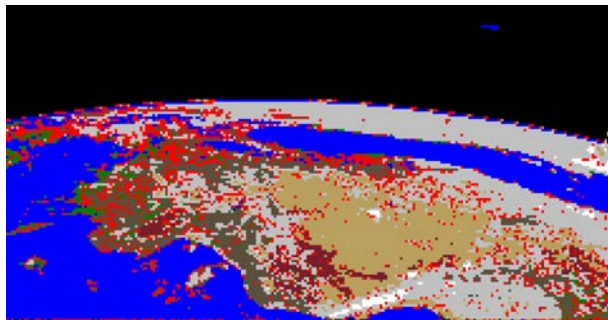


**Figure 9: Confusion matrix and the result of scene classification by the model No.7 in Table 3**

The both precision and recall of the HSV classifier exceed that of the RGB classifier except City and Forest. However, classifying the Forest and the Black Land are sometimes confusing even for humans. Classifying Forest as Black Land does not affect the 3-axis attitude determination since both of the classes are defined as land in the matching process. When it comes to City, the difference can be negligible. Therefore, we concluded that the classifier of the HSV exceeds that of the RGB. The results of scene classification also makes it clear that the misdetection of the city with clouds and vice versa happens frequently. To improve the accuracy, we eliminated Foggy Cloud and City. Not only that, we set a certain threshold value to the Softmax Function in the output layer and eliminated the classified result of each window with the output of low probability(The red dots in Figure 11). The result is shown in Figure 10 and 11.



**Figure 10: 7 classes' scene classification using model No.7 in Table 3.**



**Figure 11: 7 classes' scene classification using model No.7 in Table 3. (The threshold setting: 70%)**

### The performance test of 3-axis attitude determination

We also tested the entire sequence, the attitude determination, to make sure if our sensor, DLAS, works correctly on orbit. Since the images taken from the ISS do not contain the attitude information, we generated the simulated earth images using a 3D rendering software "Maya" with a high-resolution image of the earth "Blue Marble: Next Generation" provided by NASA. We, then, projected them on the screen as shown in Figure 12.



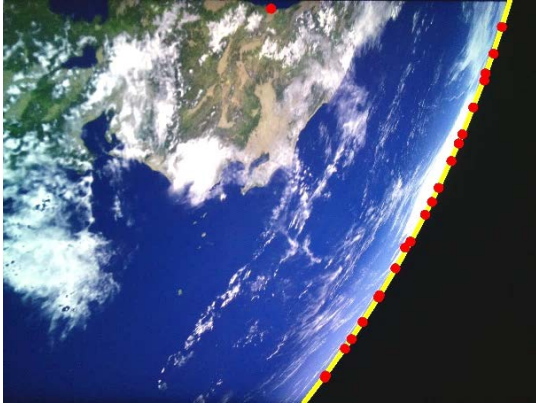
**Figure 12: Performance test image**

We tested 8 images and the results are shown in Table 4. The 2-axis attitude determination tends to fail by the misdetection of edges when the blackish sea or other blackish land is projected on an image. When it comes to image recognition, it fails when there is only a sea or land in an image as well as the existence of the blackish land. The image generated with "Blue Marble: Next Generation" was actually not a real image taken on orbit entirely. Therefore, the image recognition result tends to be less accurate than the real images taken from the satellite. The error value of the attitude from the performance test are listed in Table 4.

**Table 4: Performance test result**

No.	2-axis attitude determination		3 <sup>rd</sup> axis attitude determination	Time[s]
	Error [°]		Error [°]	
	$\theta$	$\phi$	$\psi$	
1	1.65	1.71	1.71	7.96
2	0.67	0.05	0.79	5.66
3	0.07	0.25	0.79	5.73
4	0.48	0.35	0.88	5.88
5	0.23	0.15	0.89	5.76
6	0.19	0.31	0.58	5.62
7	1.00	0.23	2.32	5.61
8	0.06	0.18	0.12	5.68


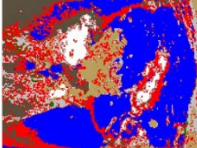

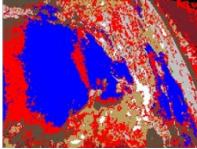

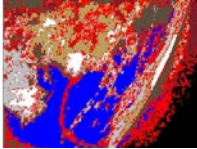

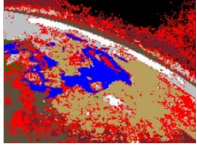

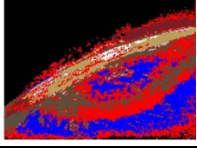
One of the 2-axis attitude determination simulation result (No.3 in Table 4) is shown in Figure 13. The red points are the detected edge of the earth, and the yellow curve line is estimated edge which is calculated on the imaginary unit sphere. Even if a small misdetection occurs, the algorithm can exclude them correctly.



**Figure 13: Simulation picture and the result of the earth edge detection**

The image recognition was then performed and the result is shown in Table 5. Since the monitor used in the test could not portray enough contrast, the results has become a bit rough. We set the threshold of the softmax function high, which is 85%, to avoid the noise from the display.

**Table 5: The result of the scene classification in Table 4dis**

No.	Error [°]	Captured image & the result of the image recognition	
	$\psi$		
1	1.71		
2	0.79		
3	0.79		
4	0.88		
5	0.89		

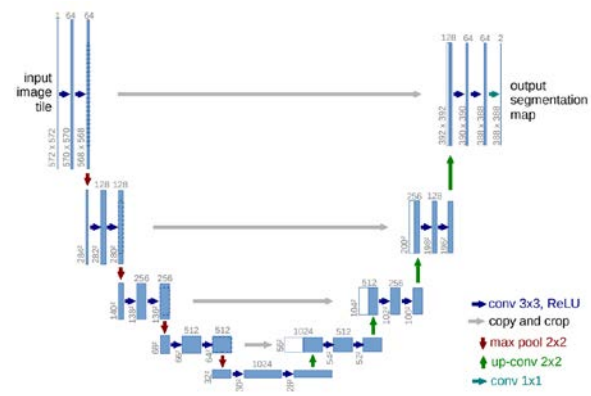
The red dots are the eliminated classified results with the output of low probability in the softmax function. Because of the noise and the brightness from the display, the classified results from the noise are successfully eliminated (The red dot in the figures). The misdetection of the Space is also because of the brightness. These results proves the validity of the proposed algorithm.

## CONCLUSION

The attitude determination algorithm we proposed archives good performance on various images. The processing time of the entire attitude determination is under 6 seconds. We have to remember that the network model has to be selected carefully since it depends on the performance of the on-board computers. We are sure that our algorithm can be developed deeper and become more precise in the era of IoT innovations.

## FUTURE WORK

The image segmentation using deep learning is now popular in the area of image recognition. We will apply one of the image segmentation techniques: U-net [5], which is convolutional network architecture for fast and precise segmentation of images, developed for biomedical image segmentation (see figure 14). When it comes to U-net, it is still on a test phase and is not mounted on our DLAS project. Therefore, we only introduce the idea of how we are planning to use the result of the U-net in this section.




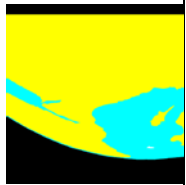
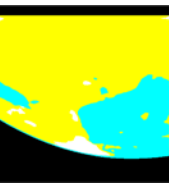

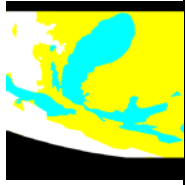
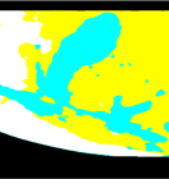
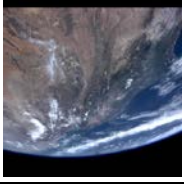
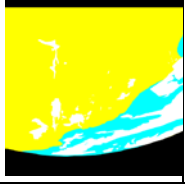
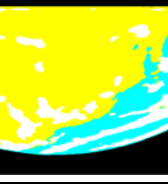
**Figure 14 : U-net architecture [5]**

## Methodology

We trained the network using ISS images. As a trial, we created Ground Truth data from 3 images ( $512 \times 512$  pixel) shown in Table 6. Since U-net uses image segmentation techniques, we do not need to classify the images into 7-10 classes. Therefore, we defined 4 classes: land, sea, cloud and space, which is the minimum required classes in the process of 3-axis attitude determination. This network has advantages in

the accuracy since it classify the image on a pixel bases and has the unique architecture. In addition, we do not need to separate the process of 2-axis attitude determination and the image recognition since we use the pixel based classification. The edge detection using first-order-derivative process can be eliminated if the image recognition process can detect the earth edge precisely.


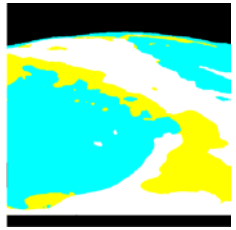

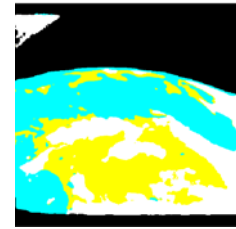
**Table 6: Trained images and GT for U-net**

Image	Ground Truth	Results
		
		
		

**Trial**

We tested 2 images and the results are show in Table 7. The trial was conducted by the high-performance computer which has multiple GPUs inside. The processing time of the U-net will certainly exceed that of the MLP. Therefore, we will introduce another single board computer which has GPU inside. It goes without saying that the processing time is the most important part for attitude determination and we will examine it in the future. We, therefore, have not yet reached the point where we examine and consider the validity of this network on orbit. However, we are sure that the pixel based classification enable more precise 3-axis determination. Our goal is a real time image recognition. We have to improve networks, the algorithm and the hardware to classify the image on orbit and perform 3-axis determination using image segmentation.

**Table 7: Test result using U-net**

Image	Result
	
	

**REFERENCES**

1. K. Tawara, S. Harita, Y.Yatsu, S. Matunaga, and Hibari project team (2017) “Technology Demonstration Microsatellite Hibari: Variable Shape Attitude Control and Its Application to Astrometry of Gravitational Wave Sources”, 31st International Symposium on Space Technology and Science, 2017-f-013, Matsuyama, Japan, June.
2. R. Hartley and A. Zisserman, “Multiple view geometry in computer vision”, Cambridge university press, 2003.
3. J.Long, E.Shelhamer, and T.Darrell,”Fully convolutional networks for semantic segmentation”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431-3440, 2015
4. NASA, “The-Earth-4K-Extended-Edition”, [https://archive.org/details/NASA-Ultra-High-Definition/The-Earth-4K-Extended-Edition\\_MP4.mp4](https://archive.org/details/NASA-Ultra-High-Definition/The-Earth-4K-Extended-Edition_MP4.mp4), Accessed: June, 2017.
5. Olaf Ronneberger, Philipp Fischer, Thomas Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, International Conference on Medical image computing and computer-assisted intervention, Springer, LNCS, Vol.9351:234-241, 2015