



City Research Online

City St George's, University of London

Citation: Zhang, Z., Xu, G. & Song, J. (2018). CubeSat cloud detection based on JPEG2000 compression and deep learning. *Advances in Mechanical Engineering*, 10(10), 1687814018808178. doi: 10.1177/1687814018808178

This is the draft version of the paper.


This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: <https://openaccess.city.ac.uk/id/eprint/27162/>

Link to published version: <https://doi.org/10.1177/1687814018808178>

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to [City Research Online policy](#).

CubeSat cloud detection based on JPEG2000 compression and deep learning

Advances in Mechanical Engineering
2018, Vol. 10(10) 1–10
© The Author(s) 2018
DOI: 10.1177/1687814018808178
journals.sagepub.com/home/ade


Zhaoxiang Zhang , Guodong Xu and Jianing Song 

Abstract

In order to enhance the efficiency of the image transmission system and the robustness of the optical imaging system of the Association of Sino-Russian Technical Universities satellite, a new framework of on-board cloud detection by utilizing a lightweight U-Net and JPEG compression strategy is described. In this method, a careful compression strategy is introduced and evaluated to acquire a balanced result between the efficiency and power consuming. A deep-learning network combined with lightweight U-Net and Mobilenet is trained and verified with a public Landsat-8 data set Spatial Procedures for Automated Removal of Cloud and Shadow. Experiment results indicate that by utilizing image-compression strategy and depthwise separable convolutions, the maximum memory cost and inference speed are dramatically reduced into 0.7133 Mb and 0.0378 s per million pixels while the overall accuracy achieves around 93.1%. A good possibility of the on-board cloud detection based on deep learning is explored by the proposed method.

Keywords

Deep learning, cloud detection, JPEG2000, CubeSat, ASRTU mission

Date received: 8 March 2018; accepted: 10 September 2018

Handling Editor: ZW Zhong

Introduction

As far as the CubeSat satellites are concerned, the capability of the data transmission system would be largely limited by power consumption and covering time due to its low cost and lack of ground stations.

For example, The Association of Sino-Russian Technical Universities (ASRTU) satellite is designed by the Research Center of Satellite Technology (RCST) in Harbin Institute of Technology (HIT). The planned launch time is in 2019. The sketch of ASRTU is illustrated in Figure 1. And its hardwares and corresponding parameters related to optical payload are shown in Figure 2. Take the LilacSat-2 satellite as a reference which was also designed by HIT and launched on 20 September 2015. With downlink frequency of 437.2 MHz and orbit height of 524 km, the downlink data rate is less than 9600 bps, and the time window is around 10 min in every track, which means less than 3

images with size of 500×500 are able to be downloaded from satellite during one track. Thus, it highlights the importance of image processing and classification before the data transmission to improve its downlink efficiency.

On the other hand, over 66% of the land surfaces on Earth is covered by cloud, and it often appears and covers objects on the surface in remote-sensing (RS) images to make much difficulty for further image processing. Therefore, it is necessary to apply the on-board image cloud-detection algorithms in the ASRTU mission.

Harbin Institute of Technology, Harbin, China

Corresponding author:

Guodong Xu, Harbin Institute of Technology, B3 Building, HIT Science Park, Harbin 150001, China.
Email: xgdong_61@163.com



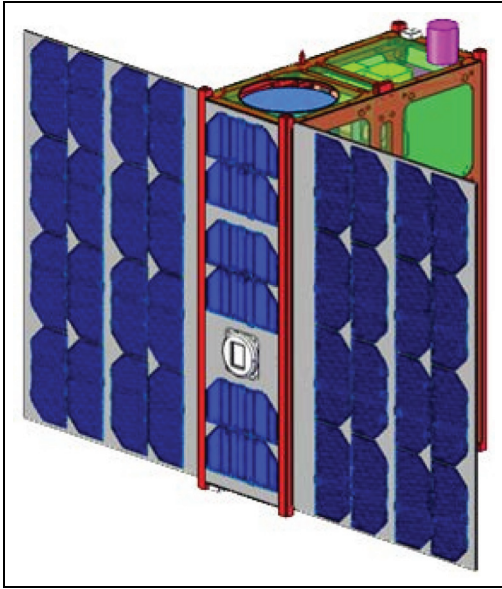


Figure 1. The sketch of the ASRTU satellite.

Generally, it is difficult to deploy the creative and novel classification algorithms in the satellite platform for several reasons, such as the requirement of high reliability in satellite engineering and the real-time demands for numerous downlink data. The development of the CubeSat satellites demonstrates a solid edge for the verification of the novel missions such as on-orbit RS justification,¹ space debris identification,^{2,3} visual navigation aids,^{4,5} and others.

In recent years, the impressive results of deep-learning networks for computer vision applications brought fresh air to the satellite applications. Real improvements in several applications could be observed such as image classification and scene recognition, image retrieval, and many others. Deep-learning method offers a great opportunity for breakthroughs

of the on-board optical sensor data processing, especially in CubeSat missions. However, the mainstream deep-learning networks require extremely high computation capability of hardware, which is intolerable for the platform of the small satellites. Therefore, two strategies are applied here to improve the efficiency of the network. First, considering that the JPEG2000 compression method is utilized in ASRTU mission as demonstrated in Figure 2, a careful compression strategy is required to create the low-resolution images for image classification and detection. Furthermore, the image processing algorithm would be deployed on the compressed images to reduce the required memory of the network. Second, several improvement such as lightweight network layers, depthwise separable convolutions are introduced to shrink the network and improve its inference speed.

Many different machine-learning algorithms have been applied in cloud detection for recent years,⁶⁻⁹ while few of them considers on-board cloud detection. The IPEX mission by NASA in 2013 is the first time that a machine-learning system has been trained on a sub-orbital flight and then successfully utilized on orbit.¹⁰ However, the data compression problem is not under its consideration due to the low-resolution raw images. The STU mission launched on 25 September 2015 utilized the optical sensor data taken from Antarctic region to calibrate its exact position with the openly available Modis 250 data.⁴ In addition, random forest method is introduced into the hyperspectral images for object classification.^{11,12} An improved texton-based approach is demonstrated in article¹³ to categorize the cloud image patches. In addition, article by Lin et al.⁶ introduces a cloud-removal approach based on information cloning which could remove cloud-contaminated portions of a satellite image and then reconstruct the information of missing data utilizing temporal correlation of multitemporal images.

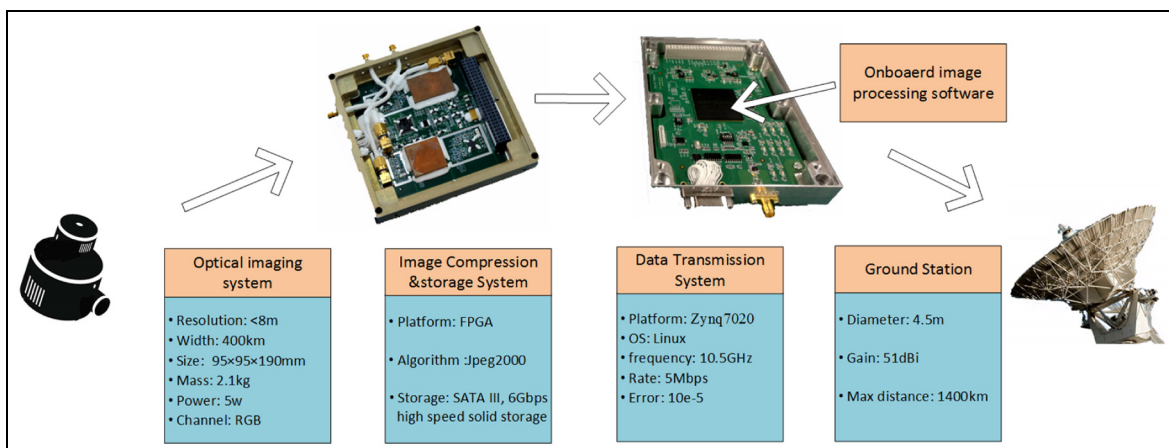


Figure 2. Hardwares and parameters of the ASRTU satellite.

Besides, a green channel background subtraction adaptive threshold (GBSAT) algorithm is applied in the green channel of the visual images¹⁴ to detect the cloud automatically.

To summarize, the main contributions of this work are: (1) a novel framework including special image-compression strategy and a lightweight U-Net combined with depthwise separable convolutional layers for cloud detection and (2) verification and comparison of two mainstream semantic segmentation networks and three wavelets for image compression.

The article is structured as follows. Section “JPEG2000 compression system” gives a brief recap of JPEG2000 compression and a carefully designed compression framework in order to provide the image input for the proposed deep-learning models, introduced in section “Deep-learning classification.” The experiment of the model training and test using different algorithms is illustrated in section “Experiment results,” while the conclusion and discussion in section “Conclusion and discussion.”

JPEG2000 compression system

JPEG2000 is an international standard compression technology based on wavelet transform, created and maintained by Joint Photographic Experts Group.¹⁵ It is a powerful and convenient tool which is widely utilized in RS image processing and storage. The JPEG2000 compression standard enables both lossless and lossy storage and became popular for its two major advantages: progressive transmission and region of interest coding.¹⁶ Progressive transmission means that in the image transmission process, the general content of the image would be transmitted first, before the details of the image information.¹⁷ With the gradual increase of the received image data, higher resolution pictures would be created. In this task, the compression system with progressive transmission would help the satellite platform judge and classify the RS images on orbit by using compressed images. Furthermore, the satellite bandwidth would be saved since the compressed images covered by thick cloud could be detected. The architecture of the basic JPEG2000 encoder and decoder are shown in Figure 3.

Basic discrete wavelet transform

In Figure 3, the discrete wavelet transform (DWT) would be utilized to reduce the image size without losing much of the resolution.¹⁸ It could be applied to a whole original image and provide a different level of decomposition with image coefficients blocks, and the block of the transformed coefficients are classified into types like High-High (HH), High-Low (HL), Low-High (LH), and Low-Low (LL). These types are described as

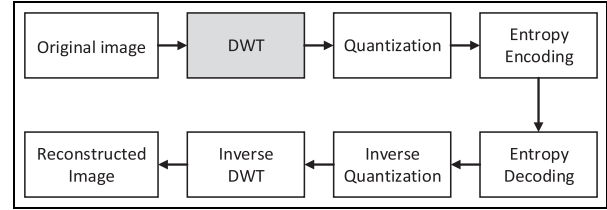


Figure 3. Basic architecture of JPEG2000 encoder and decoder.

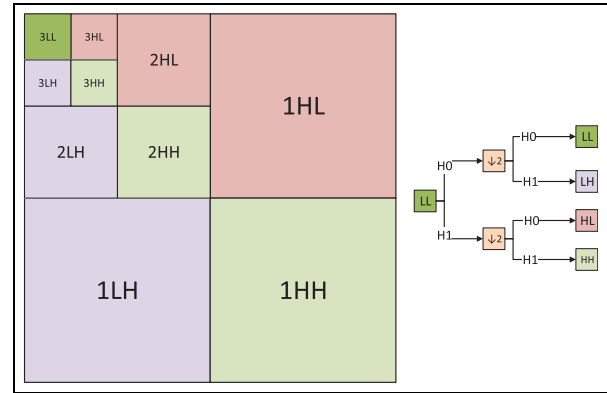


Figure 4. Three-level DWT sub-band structure.

diagonal, horizontal, vertical, and image approximation. The sub-band structure generated by a three-level DWT algorithm on an original image is depicted by Figure 4.

Where H0 and H1 represent the row transformation and column transformation of the DWT, respectively. After two transformations a new image with half resolution in two directions would be obtained, which is the LL in Figure 4. Furthermore, the decomposition process of the multi-level DWT and the results of each level decomposition are shown in Figure 4. The image block in upper left corner named by LL represents the low-frequency coefficient of the raw image, which could be treated as the image approximation of the original image, while other three different types contains the high-frequency coefficients along with the details of the raw image. When the DWT method reaches higher level, compressive images with lower resolution could be retrieved, which shows one of the good nature of DWT method: multi-resolution compression.

In the image matrix subjected to wavelet transform, the elements in the upper left corner (LL) show the average of the pixel values of the entire image, while the rest is the detail factor of the image block. According to this fact, if some high-frequency details of the coefficient are removed, it turns out that the reconstructed image quality is still acceptable. Furthermore, a large number of coefficients in the transformed matrix would become zero after the quantization process. Therefore, the matrix would become easier to be compressed

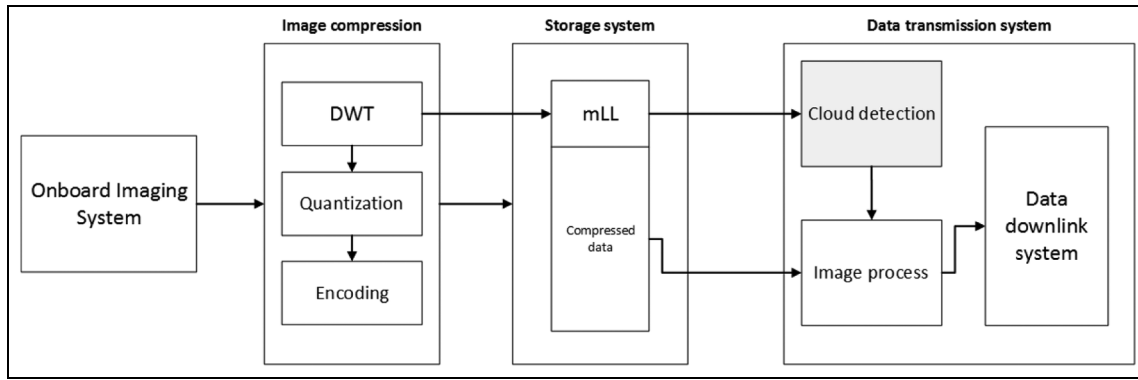


Figure 5. The proposed compression framework.

without losing the accuracy by using a proper encoding method.

Progressive transmission

The operation platform of the compression system is based on a field-programmable gate array (FPGA) processor in ASRTU satellite due to its efficiency requirement and power limit. However, the machine-learning algorithms are always deployed on a ARM processor, which means that the entropy decoding inverse quantization and inverse DWT are demanded in ARM processor to retrieve the raw image in low resolution. Therefore, the progressive transmission is introduced here to simplify the problem.

The definition of the progressive transmission is that reconstructing successively higher fidelity versions of an image along with the receiving data. The goal of the progressive transmission is thus not only efficient overall compression, but efficient compression at every step¹⁹ especially under the condition while the data rate available for image transmission is unexpectedly low or the volume of the compressed data exceeds expectations. The progressive transmission would provide an opportunity to make more efficient usage of the data channel, since the less-value data would be detected in advance and not be transmitted with the original resolution.

There are several different types of the progressive transmission orders supported by JPEG2000 standard in different application scenarios. Take the component-position-resolution-layer (CPRL) order as an example, the brightness component in the outermost compression loop would be transmitted prior to the color component in the multispectral image transformation, which would guarantee maximum recovery of the original data with only the bright image when the other component of the data is not available. In this article, the resolution-position-component-layer (RPCL) order becomes a good choice to create the low-resolution images and encode them first.

Compression framework

In order to retrieve the low-resolution images from the compressed file without decoding and inverse quantization, a new compression framework is introduced. Considering that the compression algorithms and the machine-learning methods are deployed in different platform, the framework is expressed in Figure 5.

In Figure 5, one can see that the raw image obtained from on-board imaging system is introduced into the compression system. The mLL image means the m-level low-frequency image data after applying DWT, and it would be directly introduced into the machine-learning algorithm to detect the cloud, m could be adjusted by the resolution of the raw data and the training data of the machine-learning methods. The result of the methods would determine whether the compressed image is qualified to be transmitted to the ground station.

By utilizing the mLL images from the compression system, the cloud-detection system would calculate the cloud fraction of each image, and further decide the quality of the data. Considering the limited transmission resources for small satellites, the raw cloud data would be thrown away directly if its cloud fraction is above the default threshold. In addition, for images that are partly covered by cloud, the value in cloudy area would be transformed into zeros to improve the compression efficiency. The classification map combined with compressed images would be transmitted to the ground station in order to distinguish the cloud area.

It is worth mentioning that the mLL image block is retrieved without quantization and encoding. And the mLL data combined with other compression data are stored in the on-board hard disk in .rhh format. The .rhh format is a custom image format file whose header contains the necessary information of image-compression coding, such as types of color transform, coefficients of DWT, order of progressive transmission, and so on. Incomplete .rhh files can still be decoded correctly to restore a relatively blurred image.

Generally, the Le Gall 5/3 DWT would be the popular choice to obtain the mLL image data. The mLL data could be obtained by the following equation

$$\begin{aligned}
 & \text{for } 1 : m \\
 & \quad Y = D(X) \quad \text{in row direction} \\
 & \quad X \leftarrow Y \\
 & \quad Y = D(X) \quad \text{in column direction} \\
 & \quad X \leftarrow Y \\
 & \text{end}
 \end{aligned} \tag{1}$$

where $D(x)$ represents the Le Gall 5/3 DWT equation, that is

$$\begin{aligned}
 y(2n+1) &= x(2n+1) - \left[\frac{x(2n) + x(2n+2)}{2} \right] \\
 y(2n) &= x(2n) + \left[\frac{y(2n-1) + y(2n+1) + 2}{4} \right]
 \end{aligned} \tag{2}$$

Besides, other wavelets such as Haar and Symlet could also be introduced to the image-compression system. The function of the two equations is well known, and the details can be seen in a study by Yadaiah and Ravi.²⁰ Therefore, m -level low-frequency DWT result would be restored after m times DWT operation.

Deep-learning classification

A standard artificial neural networks consists of many simple, connected processors called neurons, each producing a sequence of real-valued activations. These neurons are heavily inspired by way biological nervous systems (such as the human brain) operate. The basic structure of artificial neural networks consists of input, hidden layers, and output. The hidden layers will make decisions from the previous layer and weigh up how a stochastic change within itself detracts or improves the final output, which is referred as learning.

Convolutional neural networks (CNNs) are analogous to traditional full-connected neural networks, from the input raw image vectors to the final output of the class score, the entire of the network will still express a single perceptive score function (the weight). The last layer will contain loss functions associated with the classes.²¹ Compared with full-connected neural networks, CNN becomes much more popular in image detection tasks due to its great performance in the field of pattern recognition within images.

Pixelwise semantic segmentation

As far as the RS community, the progress of the on-orbit image processing is difficult due to the power and time limit of the satellite. For the ASRTU satellite, several requirements of the algorithm are listed in order to guarantee a good result in space. First, the algorithm needs to be fast enough since the time window of the

downlink process is limited. Second, the algorithm operation platform is based on ARM9, which means algorithms that is vaguely complex and heavily depends on graphics processing unit (GPU) processors may be unacceptable. Third, the algorithm should be robust enough to deal with the complex electromagnetic environment in space. Deep-learning methods have been widely applied in different areas including RS image classification. However, most of the popular deep-learning networks contain more than 10 layers which reduce the possibility of on-board deployment due to the limited power and computation capability of the CubeSat satellite. Therefore, only the lightweight networks with fewer parameters should be considered here. In addition, unlike most cases, the size of the data set would exponentially reduce after image compression, and the existing deep-learning network combined with super pixel methods for cloud detection can hardly work. Therefore, the semantic segmentation neural network should be considered here to classify the cloud in the compressed data set.

Actually, there are two mainstream strategies for the pixelwise neural network. One is deconvolution network which was utilized in SegNet²² and Deconv-Net.¹³ Another is the upsampling layers which was utilized in fully convolutional network (FCN)²³ and U-Net.²⁴ Since the size of the cloud area varies from different scenes, the information from high-resolution layers should be remained and utilized in later layers. Therefore, U-Net with upsampling layers is introduced here as the main architecture of our network. Besides, the architecture of Deconv-Net with deconvolution layers is also introduced here to make a good comparison. The deconvolution layers is the transpose convolution layer to turn the convoluted images into their original size by reversing convolution. It would help the neural network achieve the pixelwise detection precision.

Depthwise separable convolutions

In order to further reduce the peak memory cost of the network in ARM processor and improve the inference speed of the network, the depthwise separable convolution from MobileNet²⁵ is illustrated and combined with the proposed network. The standard convolution operation includes two steps, one is filtering image features based on the convolutional kernels, another is combining features to produce a new representation. And the filtering and combination steps can be split into two steps by utilizing the factorized convolutions called depthwise separable convolutions for substantial reduction in computational cost. Depthwise separable convolution is composed of two layers: one is depthwise convolutions which apply a single filter in each input channel and remains the same channels after the convolution. Another is pointwise convolution, which uses



Figure 6. WRS2 path/row data location of imagery utilized for training and test.

simple 1×1 convolution to create a linear combination of the output of the depthwise layer. Besides, the batchnorm and ReLU operation is applied after both layers. Compared with the standard convolutional layers, the computational cost of the depthwise separable convolution is changed from equations (3) to (4)

$$\text{Standard} = C1 \times M \times N \times C2 \quad (3)$$

$$\text{Depthwise} = C1 \times M \times C2 + M \times N \times C2 \quad (4)$$

where $C1$ and $C2$ are, respectively, the convolution computation cost of the input layers and the output feature map, respectively. It is obvious that the depthwise convolution is extremely efficient relative to standard convolution. Generally, the 3×3 kernel would be used in depthwise separable convolutions and its computation cost would be 8 or 9 times less than the standard convolutions, and the accuracy would decrease slightly. The details of the result are shown in the next section.

Combining with the depthwise separable convolution networks, the U-Net-based architecture MobU-Net and the Deconv-Net-based architecture MobDeconv-Net are illustrated in Tables 1 and 2, respectively. The conv in the tables means the standard convolution layers, and the dw/s represents the depthwise separable convolutional layers. a changes as the size of the input images changes. Considering that the compressed images are paid attention. a becomes 72, 36, and 16 when the 300×300 data set is compressed in level-2, level-3, and level-4.

Experiment results

The main objective of this article is to evaluate the generalization capacity of the effective deep-learning

Table 1. Architecture of the lightweight MobU-Net.

Layers	kernel size	output size	channels
Input	3×3	$a \times a$	4
conv1-1	3×3	$a \times a$	8
conv1-2	3×3	$a \times a$	16
pool1	3×3	$a/2 \times a/2$	16
dw/s2-1	3×3	$a/2 \times a/2$	32
dw/s2-2	3×3	$a/2 \times a/2$	32
pool2	3×3	$a/4 \times a/4$	32
dw/s3-1	3×3	$a/4 \times a/4$	64
dw/s3-2	3×3	$a/4 \times a/4$	64
unpool1	3×3	$a/2 \times a/2$	32
concat[unpool1, dw/s2-2]	3×3	$a/2 \times a/2$	64
conv4-1	3×3	$a/2 \times a/2$	32
conv4-2	3×3	$a/2 \times a/2$	32
unpool2	3×3	$a/2 \times a/2$	16
concat[unpool2, conv1-2]	3×3	$a \times a$	32
dw/s5-1	3×3	$a \times a$	16
dw/s5-2	3×3	$a \times a$	8
dw/s5-3	3×3	$a \times a$	2

a depends on the input image size.

Table 2. Architecture of the lightweight MobDeconv-Net.

Layers	kernel size	output size	channels
Input	3×3	$a \times a$	4
dw/s1	3×3	$a \times a$	8
dw/s2	3×3	$a \times a$	16
fc3	1×1	$a \times a$	32
fc4	1×1	$a \times a$	32
deconv2	3×3	$a \times a$	16
deconv1	3×3	$a \times a$	8
output	1×1	$a \times a$	2

a depends on the input image size.

Table 3. Cloud-detection result of different networks with Haar wavelet.

Accuracy/recall/F1-score	Level 2	Level 3	Level 4
U-Net	0.9746/0.8778/0.9237	0.9499/0.875/0.9108	0.9479/0.9608/0.9543
MobU-Net	0.9706/0.8085/0.896	0.9546/0.7896/0.8646	0.9310/0.8422/0.884
Deconv-Net	0.953/0.829/0.887	0.9522/0.7886/0.8627	0.9352/0.7536/0.835
MobDeconv-Net	0.9530/0.8045/0.8725	0.940/0.737/0.8235	0.9279/0.7194/0.8103

Table 4. Cloud-detection result of different networks with LeGall-5/3 wavelet.

Accuracy/recall/F1-score	Level 2	Level 3	Level 4
U-Net	0.9708/0.8792/0.9227	0.964/0.881/0.921	0.9651/0.9272/0.9458
MobU-Net	0.9619/0.8190/0.8847	0.9416/0.7751/0.8503	0.9416/0.7751/0.8503
Deconv-Net	0.9559/0.8176/0.8814	0.9455/0.7371/0.8284	0.9254/0.7144/0.8063
MobDeconv-Net	0.9397/0.8387/0.8863	0.9316/0.7224/0.8138	0.9137/0.6385/0.7517

Table 5. Cloud-detection result of different networks with Symlet-2 wavelet.

Accuracy/recall/F1-score	Level 2	Level 3	Level 4
U-Net	0.9644/0.8828/0.9218	0.9436/0.8870/0.9145	0.9675/0.7803/0.8642
MobU-Net	0.9668/0.8520/0.9058	0.9567/0.7019/0.8097	0.9382/0.6838/0.7911
Deconv-Net	0.9074/0.8261/0.8649	0.9482/0.7555/0.8409	0.93086/0.6551/0.7690
MobDeconv-Net	0.9376/0.8236/0.8769	0.9427/0.8430/0.8901	0.8943/0.7746/0.8302

algorithms in on-board cloud detection for the ASRTU mission. In order to acquire better results, the open-source Landsat data SPARCS (<https://landsat.usgs.gov/sparcs>) is downloaded and utilized here as the training and test data set. The SPARCS data set was created by M. Joseph Hughes from Landsat 8 operational land imager (OLI) level-1B scenes. The overall images are separated into two groups: 80% as training set and 20% as test set. We carefully pick 12 representative scenes as test images and 64 scenes for training to make sure that every class is included and each class ratio in two groups keep approximately the same. The distribution of the scene for training and test is demonstrated in Figure 6. In order to make better simulation of the images from ASRTU satellite, several pre-processing steps are required here.

1. Considering that the ASRTU satellite products contains only four bands, in SPARCS data set only band 2, 3, 4, 5, which is red, blue, green, and infrared bands are utilized for training and testing, and the classes are reclassified as two classes including cloud and non-cloud.
2. All the images are splitted into sub-scene with size of 300×300 to simulate the result of the progressive transmission. Therefore, 1024

sub-scenes are used for training, and 256 sub-scenes are used for test.

All the networks are deployed in Windows-based TensorFlow environment with a single NVIDIA GTX-1060 GPU. The Google Cloud Platform is also utilized to train the data set.

The overall accuracy, recall, and F1-score are introduced here to evaluate the performance of different compression algorithms. F1-score could be retrieved from the following equation

$$F1 = \frac{2 \times Acc \times Rec}{Acc + Rec} \quad (5)$$

where *Acc* means the overall accuracy, and *Rec* represents the recall of the neural network. The results of the neural network with different wavelets are illustrated in Tables 3–5, respectively.

The architecture of the MobU-Net and MobDeconv-Net have already been demonstrated in Tables 3 and 4. The U-Net and Deconv-Net in the tables represent the lightweight U-Net and Deconv-Net with the same architecture in Tables 3 and 4, while the convolution process is achieved by the standard convolution layers. The results from the three tables show the following:

Table 6. Average iteration time cost in training process for different networks with Haar wavelet.

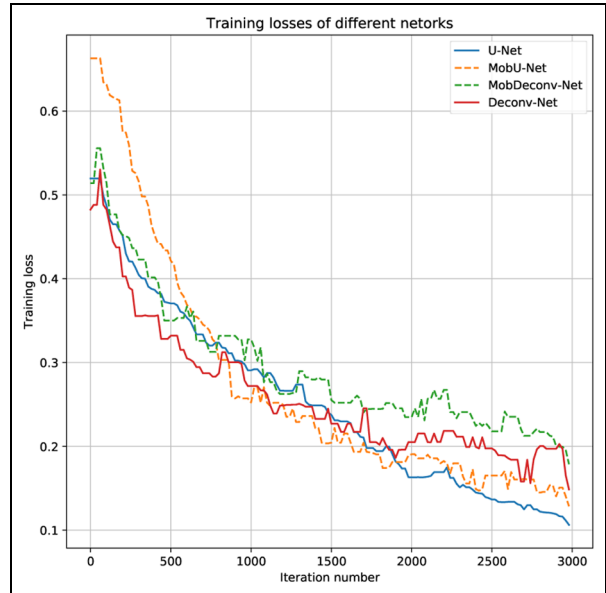
Average iteration cost	Level 2	Level 3	Level 4
U-Net	0.061	0.049	0.028
MobU-Net	0.125	0.048	0.031
Deconv-Net	0.065	0.024	0.015
MobDeconv-Net	0.081	0.020	0.016

1. The overall accuracy and the recall are decreased when the depthwise separable convolutional layers are introduced into the networks, and its influence on the recall is more obvious.
2. Generally, U-Net shows the best performance on the data set with different levels and different wavelets. And correspondingly, MobU-Net illustrate higher F1-scores than MobDeconv-Net.
3. Cloud-detection results on Haar wavelet data set shows slightly better overall performance than other wavelets. For example, the recall of U-Net in Table 3 achieves 0.9543, while with other two wavelets the recall could only achieve 0.9272 and 0.7803, respectively. Considering that the high recall represents the low rate of the misclassified cloud areas, Haar wavelet is therefore selected in the JPEG2000 compression system for further process.

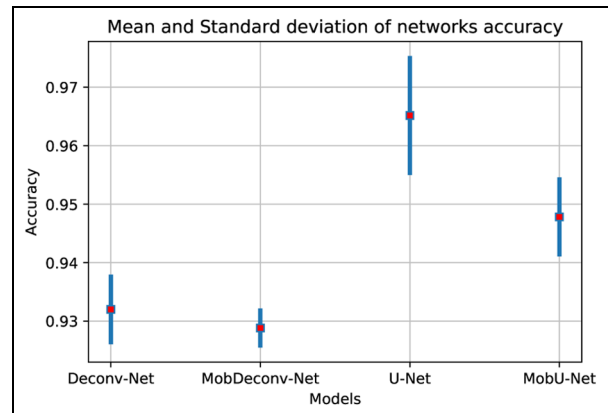
Table 6 demonstrates the average iteration time cost during training process for different networks with Haar wavelet. It is clear that the training time becomes shorter when the level of compression becomes deeper. Besides, the MobU-Net and MobDeconv-Net cannot save training time as we expected, which we think due to the reason that calculation for depthwise separable convolution layers in TensorFlow framework would not save too much time. However, the peak memory cost of the layers are much more economic, as shown in Table 7.

Figure 7 shows the training losses with iteration increasing for different neural network models. It is obvious that U-Net model achieves slightly better performance at the end of the tracks. And the four models have similar convergence trajectory. It is worth to mention that the training losses would still decrease after 3000 iterations, but the validation loss stops decreasing around 3000 iterations.

Figure 8 illustrates the statistic results of the classification accuracy on different models. The test data set was created on Haar algorithm with level-4 compression. U-Net in Figure 8 expresses out-performance compared with other methods, while MobU-Net have better performance than MobDeconv-Net. In summary, the statistic results keeps with the uniformity in Table 3.

**Figure 7.** Training losses of different networks.**Table 7.** Maximum memory cost and inference speed of the four networks.

	Maximum memory cost (Mb)	Inference speed (s/million pixels on CPU)
U-Net (Level-0)	185.6096	7.62
Deconv-Net (Level-0)	162.1826	7.23
U-Net (Level-4)	2.0652	0.053
MobU-Net (Level-4)	0.7133	0.0378
Deconv-Net (Level-4)	0.5952	0.058
MobDeconv-Net (Level-4)	0.5073	0.039

**Figure 8.** Mean and standard deviation of networks accuracy.

To further verify the performance of the four networks, the inference speed of the networks on test data and the peak memory cost is calculated and demonstrated in Table 7. The inference speed is evaluated when

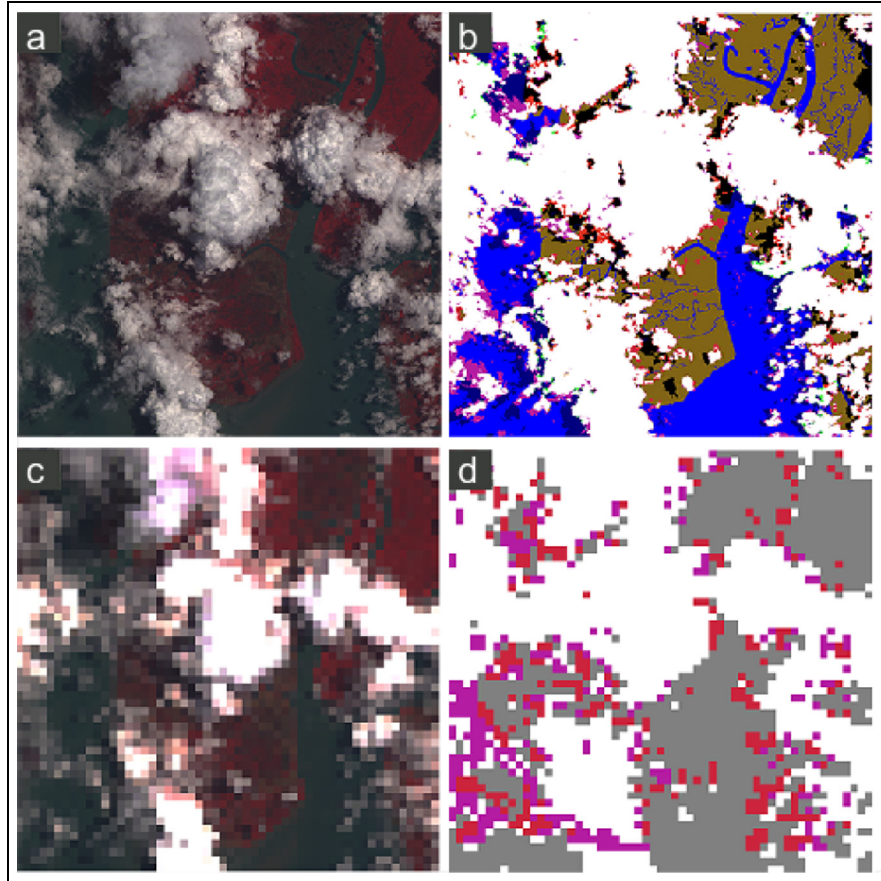


Figure 9. Cloud-detection results of level-0 and level-4 images. (a) raw image, (b) groundtruth of raw image and white class represents the cloud, (c) level-4 compressed image, and (d) classification result on level-4 image, red and pink represent the misclassification areas.

the trained model is implemented on a laptop computer with i7 CPU, 4 GB Memory, and integrated GPU.

The results in Table 7 demonstrate that the peak memory cost of the MobU-Net has been greatly reduced compared with the U-Net utilizing standard convolutional layers and the model with uncompressed data set. The two architectures have similar inference speed with stand convolution layers or depthwise convolution layers, while the architecture based on U-Net shows better potential of reducing memory cost by depthwise separable convolutional layers. In summary, the MobU-Net expresses better overall accuracy and recall than the MobDeconv-Net in Level-4 data set, and its memory cost is small enough for the ARM9 processor in ASRTU satellite. In addition, the inference speed of MobU-Net model based on the simulation satisfies the speed requirement of data processing. Thus, the MobU-Net would be selected for the cloud classification in this mission.

Figure 9 illustrates the classification results by MobU-Net combined with Haar wavelet compression algorithm. In Figure 9(d), white areas represents cloud areas that are classified correctly, red area means non-

cloud area misclassified as cloud area, and pink area represents cloud area misclassified as non-cloud areas. It is obvious that the major part of the cloud area are distinguished directly while some edge areas and corners with thin cloud is hard to classify.

Conclusion and discussion

The increase of the spatial resolution of RS missions with small size and lightweight proposes more requirements of downlink capability, the on-board detection and classification of the RS images are of great value to improve the downlink efficiency and enhance the performance of the optical imaging system, especially in small satellites with low cost.

A framework of the on-board cloud-detection system is investigated, and the experiment results have demonstrated that MobU-Net network combined with Haar wavelet compression algorithm shows the best performance in general on the SPARCS data set for the ASRTU mission. The experiment results illustrate that the overall accuracy of the MobU-Net can achieve 93.10%, while its maximum memory cost only requires

0.7133 Mb and the inference speed becomes 0.0378 s/million pixels.

There are still several aspects left for future improvement. Other data sets should be introduced to testify the algorithms and improve its performance. And careful system engineering is required to find the balance between the cost and the efficiency of the data transmission system. Moreover, more strategies that could reduce the model size should be testified and compared.


Declaration of conflicting interests


The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Zhaoxiang Zhang  <https://orcid.org/0000-0002-1469-1469>

Jianing Song  <https://orcid.org/0000-0003-0623-0395>

References

1. Penatti OAB, Nogueira K and Santos JAD. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, Boston, MA, 7–12 June 2015, pp.44–51. New York: IEEE.
2. Piattoni J, Ceruti A and Piergentili F. Automated image analysis for space debris identification and astrometric measurements. *Acta Astronaut* 2014; 103: 176–184.
3. Gu QW, Zhang SJ, Zheng ZK, et al. A convex relaxation optimization method of on-orbit servicing pose estimation using monocular vision. *J Astronaut* 2016; 37: 744–752.
4. Wu S, Wen C and Chao C. The STU-2 CubeSat mission and in-orbit test results. In: *Proceedings of the 30th annual AIAA/USU conference on small satellites*, Logan, UT, 6 August 2016. AIAA.
5. Wu Y, Gao Y, Lin JW, et al. Low-cost, high-performance monocular vision system for air bearing table attitude determination. *J Spacecraft Rockets* 2014; 51: 66–75.
6. Lin CH, Tsai PH, Lai KH, et al. Cloud removal from multitemporal satellite images using information cloning. *IEEE T Geosci Remote* 2013; 51: 232–241.
7. Zhong B, Chen W, Wu S, et al. Cloud detection method for Chinese moderate high resolution satellite imagery (Conference Presentation). In: *Remote Sensing of Clouds and the Atmosphere XXI*, International Society for Optics and Photonics, 2016, 10001: 100010R.
8. Chu Y, Pedro HTC, Nonnenmacher L, et al. A smart image-based cloud detection system for intrahour solar irradiance forecasts. *J Atmos Ocean Tech* 2014; 31: 1995–2007.
9. Li P, Dong L, Xiao H, et al. A cloud image detection method based on SVM vector machine. *Neurocomputing* 2015; 169: 34–42.
10. Chien S, Doubleday J, Thompson DR, et al. Onboard autonomy on the intelligent payload experiment CubeSat mission. *J Aerosp Inform Syst* 2016: 307–315.
11. Yokoya N and Iwasaki A. Object detection based on sparse representation and Hough voting for optical remote sensing imagery. *IEEE J Sel Top Appl* 2015; 8: 2053–2062.
12. Xia J, Bombrun L, Adal T, et al. Spectral–spatial classification of hyperspectral images using ICA and edge-preserving filter via an ensemble strategy. *IEEE T Geosci Remote* 2016; 54: 4971–4982.
13. Dev S, Lee YH and Winkler S. Categorization of cloud image patches using an improved texton-based approach. In: *Proceedings of the IEEE international conference on image processing*, Quebec, QC, Canada, 27–30 September 2015. New York: IEEE.
14. Yang J, Min Q, Lu W, et al. An automated cloud detection method based on green channel of total sky visible images. *Atmos Meas Tech* 2015; 8: 4581–4605.
15. Amita R and Pattnaik T. A hybrid JPEG and JPEG 2000 image compression scheme for gray images. *Int J Sci Eng Technol* 2014; 3: 105–111.
16. Sanchez-Hernandez JJ, Garcia-Ortiz JP, González-Ruiz V, et al. Interactive streaming of sequences of high resolution JPEG2000 images. *IEEE T Multimedia* 2015; 17: 1829–1838.
17. Christopoulos C, Skodras A and Ebrahimi T. The JPEG2000 still image coding system: an overview. *IEEE T Consum Electr* 2000; 46: 1103–1127.
18. Subudhiray S and Srivastav AK. Implementation of hybrid DWT-DCT algorithm for image compression: a review. *Int J Res Eng Appl Sci* 2012; 2: 2249–3905.
19. Kiely AB. *Progressive transmission and compression of images*. TDA Progress Report 42-124, 1996, https://tda.jpl.nasa.gov/progress_report/42-124/124E.pdf
20. Yadaiah N and Ravi N. Internal fault detection techniques for power transformers. *Appl Soft Comput* 2011; 11: 5259–5269.
21. O’Shea K and Nash R. An introduction to convolutional neural networks. arXiv preprint: arXiv:1511.08458, 2015.
22. Badrinarayanan V, Kendall A and Cipolla R. Segnet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE T Pattern Anal* 2017; 39: 2481–2495.
23. Long J, Shelhamer E and Darrell T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, Boston, MA, 7–12 June 2015, pp.3431–3440. New York: IEEE.
24. Ronneberger O, Fischer P and Brox T. U-Net: convolutional networks for biomedical image segmentation. In: *Proceedings of the international conference on medical image computing and computer-assisted intervention*, Munich, 5–9 October 2015, pp.234–241. New York: Springer.
25. Howard AG, Zhu M, Chen B, et al. Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint: arXiv:1704.04861, 2017.