TU-Net and TDeepLab: Deep Learning-based Terrain Classification Robust to Illumination Changes, Combining Visible and Thermal Imagery

Yumi Iwashita[1], Kazuto Nakashima[2], Adrian Stoica[1] and Ryo Kurazume
[1]Jet Propulsison Laboratory, California Institute of Technology,
4800 Oak Grove Dr., Pasadena, CA, USA
[2]Kyushu University, 744 Motooka Nishi-ku, Fukuoka, Japan
Yumi.Iwashita@jpl.nasa.gov

Abstract

In this paper we propose two novel deep learning-based terrain classification methods robust to illumination changes. The use of cameras is challenged by a variety of factors, of most importance being the changes in illumination. On the other hand, since the temperature of various types of terrains depends on the thermal characteristics of the terrain, the terrain classification can be aided by utilizing the thermal information in addition to visible information. Thus we propose 'TU-Net (Two U-Net)' based on the U-Net and 'TDeepLab (Two DeepLab)' based on DeepLab, which combine visible and thermal images and train the network robust to illumination changes implicitly. To improve the network's learning capability, we expand the proposed methods to the Siamese-based method, which explicitly trains the network to be robust to illumination changes. We also investigate multiple options to fuse the visible and thermal images at at the bottom layer, middle layer, or the top layer of the network. We evaluate the proposed methods with a challenging new dataset consisting of visible and thermal images, which were collected from 10 am till 5 pm (after sunset), and we show the effectiveness of the proposed methods.

1. Introduction

Terrain classification, as an essential component of a broader understanding of the terrain to be traversed, is paramount for autonomous path planning and navigation, both on Earth and on the surfaces of other solar bodies, such as the Moon or Mars. Terrain classification gives information on traversability, in terms of potential risks, maximum possible velocity, energy consumed during traverse etc. In the context of Moon or Mars the dangers of the terrain often come from obstacles and slopes that need to be avoided, rocky regions which potentially can damage the wheels as they perforated one of the wheels of the Curiosity rover, sandy regions, which can create slip, sinking, and even restrict ability to move further, also in case of the Spirit rover.

Most often the terrain is assessed via the on-board perception system, which uses a diversity of sensors such as cameras in the visual domain, infrared cameras, LIDAR, etc [11] [7] [6]. The use of cameras in the visual domain, although the simplest solution, is challenged by a variety of factors, of most importance being the changes in illumination. The RGB information returned is dependent on the incident light, reflections from the environment, and properties of the material, which in turn may depend on other factors. Thus, both the conventional, model based analytical approaches and the learning approaches to determine classifiers are influenced by the changes in the 'colors' of the terrain regions. Figure 1 shows examples of challeng-

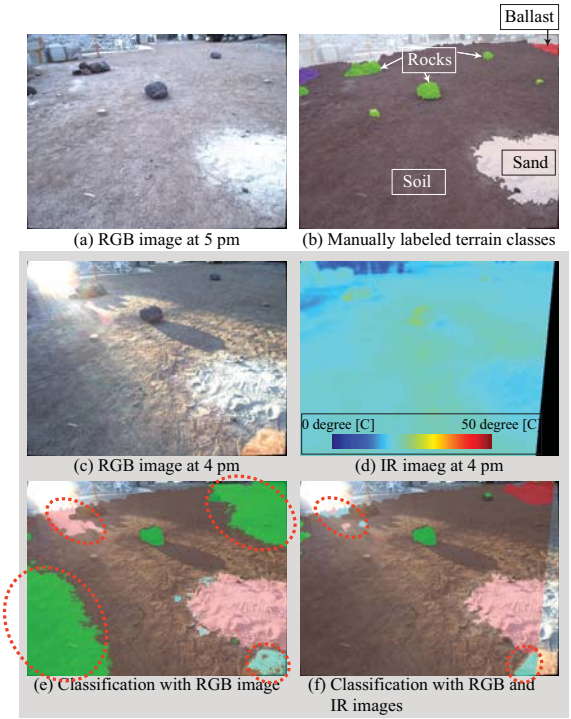


Figure 1. (a) An example RGB image at 5 pm (after sunset), (b) manually labeled terrain types of (a), (c) an example RGB image at 4 pm (same location with (a)), (d) IR image of (c), (e) terrain classification result with RGB image only by U-Net, and (f) terrain classification result with RGB and IR images by the proposed Siamese TU-Net. Red dotted circles in (e) and (f) show false positives.

ing images at the same location but different time (Fig. 1 (a) is just after sunset and Fig. 1 (c) is at 4 pm (1 hour before sunset)). The image at 4 pm (Fig. 1 (c)) is challenging due to different colors even on a same terrain type, and its terrain classification is strongly affected by the illumination changes as shown in Fig. 1 (e).

The classification in sand, ballast, rugged rocky terrain, smooth horizontal surface rocks etc can be aided by the fact that the temperature of various types of terrains depends on the thermal characteristics of the terrain. This infrared information can play an important role in helping terrain classification. Figure 1 (d) shows an example of a thermal image at 4 pm, which suggests additional information at each terrain type. Combining visible and thermal images can make the terrain classification robust to illumination changes as shown in Fig. 1 (f).

There are some existing works which utilize RGB and infrared images, such as [11] [12]. These methods focus on semantic segmentation and classify areas whose interclass variations are relatively huge, such as roads, sky, trees etc. Deep learning-based approaches brought huge improvements in semantic segmentation, and these methods can be separated into 3 categories. The first is a patch-based convolutional neural networks (CNN) [4], In this method each pixel was classified with a patch area around it, and it is relatively computational expensive due to the fully connected layers. Second is Fully Convolutional Networks (FCN) [8], which is much faster than the first one since FCN removed all fully connected layers. Brandon et al. [10] utilized "DeepLab" [2], which is one of implementations of FCN, for terrain classification and showed its feasibility in terrain classification. In general a patch-based CNN and FCN requires a huge dataset to train its parameters. Thus in case that users do not have enough dataset to train FCN from scratch, generally pre-trained parameters with public dataset such as ImageNet [5] are used. The third one is U-Net, which is popularly used in a medical image segmentation [3] and also was used by the winner of a satellite image segmentation competition (Kaggle competition [1]). U-Net has short-cut connections, which help a lot with parameter training even given a small number of dataset.

In this paper we propose two novel deep learning-based terrain classification methods based on U-Net and FCN (DeepLab). The first method is referred as 'TU-Net (Two U-Net)' based on the U-Net and the second method is as 'TDeepLab (Two Deeplab)', which combine RGB and infrared images. In TU-Net and TDeepLab, we fuse visible and thermal images at various fusion levels (i.e. bottom, middle, and top layers). Network architectures of MU-Net are shown in Fig. 2 and these three options mean the fusion of visible and thermal images over either local information, global information, or segmentation feature-level information. Moreover, to realize robustness to illumination changes, we extend TU-Net and TDeepLab to Siamese-based approaches.

To evaluate the proposed methods, we use a dataset consisting of visible and thermal images as shown in Fig. 1, which were collected every 1 hour from 10 am till 5 pm (sunset) on Nov. 17th 2017. The images in the dataset contain huge illumination variations. We annotated all images based on visible ones with terrain types (7 categories, unlabeled, sand, soil, rocks, bedrock, rocky terrain, and ballast). To the best of our knowledge, this is the first dataset consisting of visible and thermal images with terrain types.

The following sections are organized as follows. Section 2 introduces the proposed TU-Net and TDeepLab, and also we explain Siamese-based TU-Net and TDeepLab. Section 3 presents experimental evaluations of the proposed method with a new dataset, which consists of images with huge illumination variations. Section 4 shows the conclusions and future works.

2. TU-Net (Two U-Net) and TDeepLab (Two DeepLab)

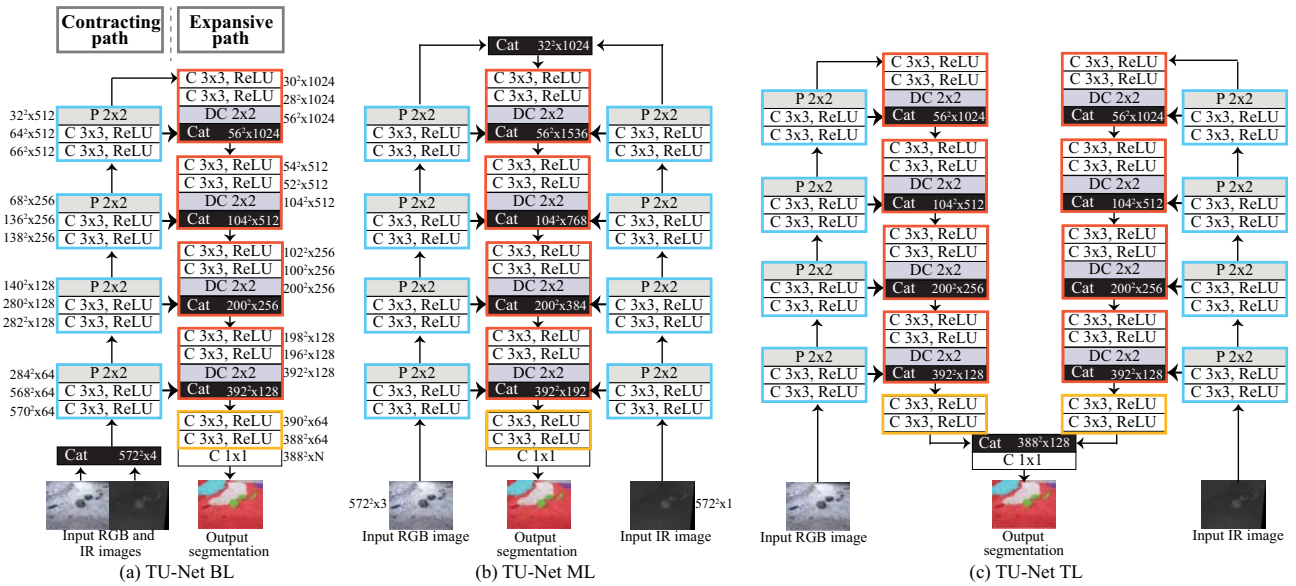
2.1. TU-Net

The three network architectures of TU-Net (TU-Net BL, ML, and TL) are shown in Fig. 2. The architecture of the TU-Net BL (Fig. 2 (a)) and the U-Net [9] are very similar, the only difference between the TU-Net BL and the U-Net is the first concatenate layer. Overall the architecture of the U-Net (and the TU-Net BL) consists of a contracting path (left) and an expansive path (right). Each path has repeated units. Output of the last unit on the contracting path contains global information after layers of convolution and pooling in the contracting path. In TU-Net BL, visible and thermal images are fused at the first layer, by simply concatenating channels at each pixel. This means input images into the network become 4 channel images (i.e. 3 channels of RGB image and 1 channel of IR image).

Figure 2 (b) shows the architecture of TU-Net ML, which has a contracting path for both visible and thermal images and fuses two different types of images at the middle layer. This fusion can be considered as a fusion of global information from visible and thermal images. Fused features are input to an expansive path. Here, the concatenate layer in the units of the expansive path receives three inputs from the deconvolution layer, convolution layer of the contracting paths of RGB image, and that of IR image.

Lastly, Fig. 2 (c) shows the architecture of TU-Net TL. After visible and thermal images pass their own contracting and expansive paths, outputs before the 1×1 convolution layer are fused, followed by mapping into the terrain classes. This fusion can be considered as a feature-level fusion.

The loss function $\mathcal{L}_{CE}$ of all architectures is defined as a pixel-wise soft-max over the final map, followed by the



(a) TU-Net BL (b) TU-Net ML (c) TU-Net TL

Figure 2. The TU-Net architectures. BL, ML, and TL shows data fusion at bottom layer, middle layer, and top layer, respectively. ”Cat”, ”C”, ”ReLU”, ”P”, and ”DC” mean ”concatenate”, ”convolution”, ”rectified linear unit”, ”pooling”, and ”deconvolution”, respectively. Relatively thick arrows between ”Cat” and ”C” include ”bilinear up-sample”. ”N” at the final layer show the number of classes. Light blue rectangles show units of the contracting path (contracting units). Red and orange rectangles show two different units of expansive paths (expansive unit 1 and 2).

cross-entropy loss function, as define as follows.

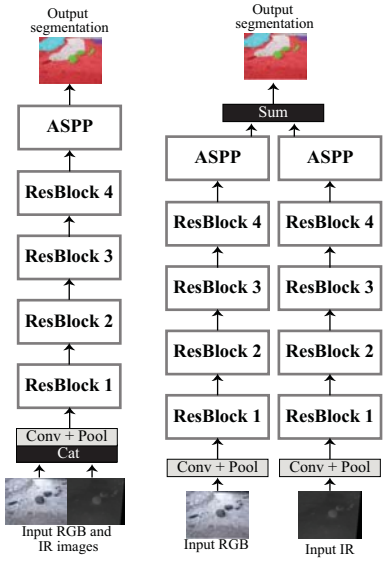
$$\mathcal{L}_{CE} = -\frac{1}{|S|}\sum_{i\in S}\sum_{j=1}^{N} y_{ij}\log p_{ij}, \qquad (1)$$

where N, $|S|$, y_{ij}, p_{ij} are the number of classes, the total number of pixels over images S, groud-truth distribution at each pixel, and outputted probability distribution at each pixel, respectively. The loss function is minimized by a stochastic gradient descent method.

To realize robustness to illumination changes, we can train the network with a training dataset with various illumination conditions. However, with this approach we expect the trained network to implicitly model illumination changes, and thus there is no guarantee that the network is efficiently robust to illumination changes.

2.2. TDeepLab

In this section we explain about the proposed TDeepLab. TDeepLab is based on DeepLab v2 from ResNet-101 [2], and it has totally 101 layers. Since this network is very deep, we use parameters trained with ImageNet as initial values, followed by fine-tuning with the images used in experiments. The main units in DeepLab are ’ResBlock’ which contains Residual Units and ASPP (Atrous Spatial Pyramid Pooling) as shown in Fig. 3. In the proposed TDeepLab, we have two architectures TDeepLab BL (Fig. 3 (a)) and TDeepLab TL (Fig. 3 (b)). TDeepLab BL integrates local features by concatenating input pair of RGB and IR images. On the other hand TDeepLab TL combines global features by getting summation of IR and RGB values



(a) TDeepLab BL (b) TDeepLab TL

Figure 3. (a) TDeepLab BL and (b) TDeepLab TL. ResBlock contains Residual Units and ASPP stands for Atrous Spatial Pyramid Pooling.

after a softmax process.

2.3. Siamese TU-Net and Siamese TDeepLab

Siamese networks, which are reported with improved network learning capabilities, enables to learn features of each terrain type explicitly. Since the TU-Net has three different types of architectures, Siamese TU-Net also has three

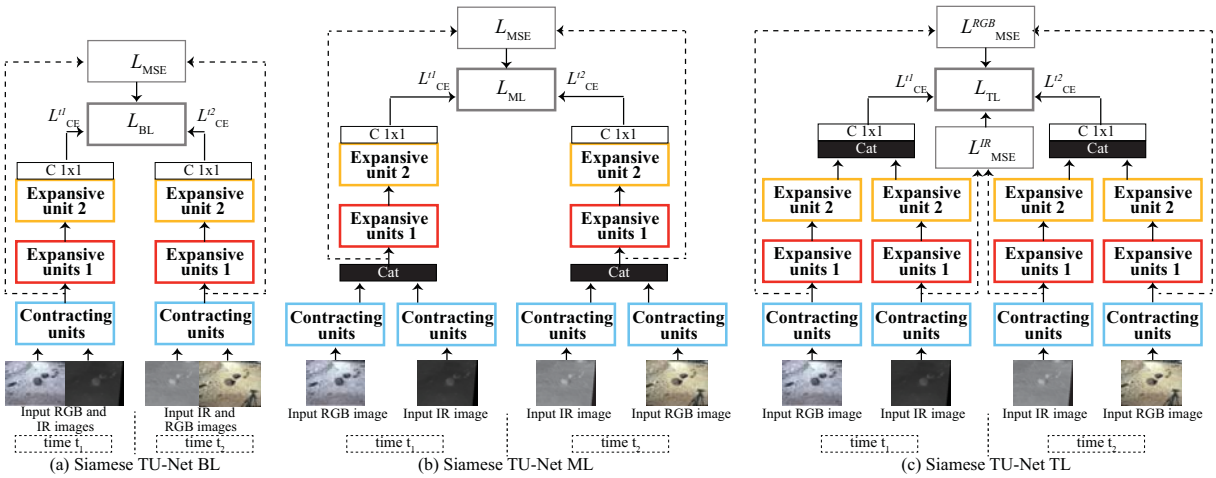


Figure 4. (a) Siamese TU-Net BL, (b) Siamese TU-Net ML, and (c) Siamese TU-Net TL.

types as shown in Fig. 4, Siamese TU-Net BL, ML, and TL. Each architecture consists of two branches of the same CNN, and these branches share parameters of the network. To train the network robust to illumination changes, pairs of visible and thermal images are prepared, which are images taken at the same location but different illuminations. In Fig. 4 the loss functions $\mathcal{L}_{CE}$ are shown with solid lines. We introduce a Mean Squared Error $\mathcal{L}_{MSE}$ based on global information, which is just right before the expansing units, to enforce the similarity measure between two images, as shown with dotted lines. $\mathcal{L}_{MSE}$ is defined as

$$\mathcal{L}_{MSE} = \frac{1}{|S| \times C} \sum_{i \in S} \sum_{j=1}^{C} (a_{ij}^{t_1} - a_{ij}^{t_2})^2, \qquad (2)$$

where $t1$ and $t2$ show different time, and C is the number of channels. For each of three Siamese TU-Nets, the total loss function $\mathcal{L}$ are defined as follows.

1. Siamese TU-Net BL:
   $\mathcal{L}_{BL} = \mathcal{L}_{CE}^{t_1} + \mathcal{L}_{CE}^{t_2} + \lambda \mathcal{L}_{MSE}$
2. Siamese TU-Net ML
   $\mathcal{L}_{ML} = \mathcal{L}_{CE}^{t_1} + \mathcal{L}_{CE}^{t_2} + \lambda \mathcal{L}_{MSE}$
3. Siamese TU-Net TL
   $\mathcal{L}_{TL} = \mathcal{L}_{CE}^{t_1} + \mathcal{L}_{CE}^{t_2} + \lambda \left(\frac{\mathcal{L}_{MSE}^{RGB} + \mathcal{L}_{MSE}^{IR}}{2} \right)$

Here, λ is a weight and in experiments we empirically set as 200.0. The loss function is minimized by a stochastic gradient descent method.

Siamese TDeepLab are also defined in a similar manner with Siamese TU-Net, and it has two types, Siamese TDeepLab BL and Siamese TDeepLab TL, as shown in Fig. 5. The loss function for both architectures is defined as $\mathcal{L}_{ML} = \mathcal{L}_{CE}^{t_1} + \mathcal{L}_{CE}^{t_2}$.

3. Experiments

In this section, we first introduce a dataset which includes visible and thermal images, followed by experimental results.

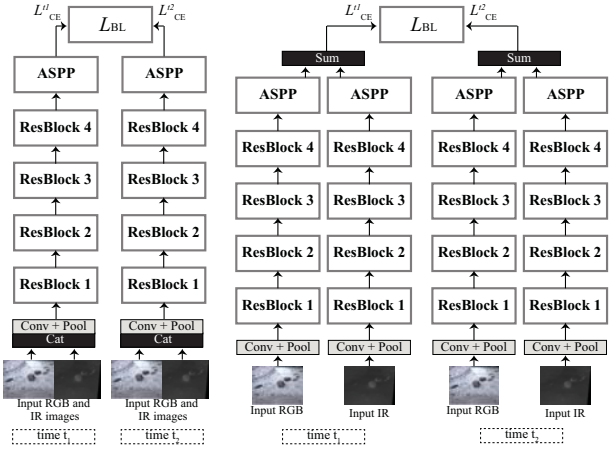


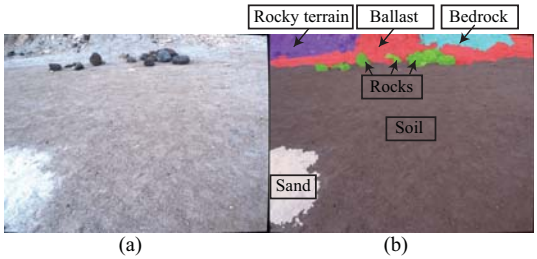
Figure 5. (a) Siamese TDeepLab BL and (b) Siamese TDeepLab TL.

3.1. Dataset of visible and thermal images

The dataset was collected at JPL on Nov. 17th 2017, with a RGB camera (FLIR Grasshopper 5M) and a thermal camera (FLIR AX65). We collected images every 1 hour, from 10 am till 5 pm (sunset), totally 8 times. Total number of images at each time is about 52 images, by changing the position of the cameras. The dataset includes challenging images such as shadows, reflection due to the Sun, and direct sunlight into the cameras (Fig. 1 (c)). The visible and thermal images are taken by different cameras, so a registration process between cameras is necessary. After we removed distortion with estimated camera inner parameters, we applied an affine transformation with estimated homography matrix.

We manually annotated all images into 7 categories, unlabeled, sand, soil, rocks, bedrock, rocky terrain, and ballast, as shown in Fig 6. To evaluate the proposed methods with the new dataset, we randomly separated images at each time into 3 datasets, 50 % for training, 25 % for evaluation,

and 25 % for test dataset. Training dataset and the evaluation dataset are used to train the network and to fix parameters, respectively. Test dataset is for estimating model properties, such as pixel accuracy and mean accuracy.



(a) (b)

Figure 6. (a) Example RGB images, and (b) manually labeled terrain classes of (a).

3.2. TU-Net

We conducted the following 3 different experiments: (Exp. 1) train, evaluate, and test with the dataset at 17:00, which has no influence of the Sun, (Exp. 2) train, evaluate, and test with all dataset from 10:00 to 17:00, and (Exp. 3) train and evaluate with dataset from 14:10 to 17:00, and two tests: (i) test with dataset from 10:00 to 13:00 and (ii) dataset from 14:00 to 17:00.

Exp. 1. The dataset at 17:00 is less challenging compared with other dataset, since this dataset is collected after the sunset. In this experiment we applied the U-Net with RGB images only, and the three TU-Net (BL, ML, and TL) with RGB and IR images. Table 1 shows 4 evaluation metrics: pixel accuracy, mean accuracy, mean Intersection over Union (IoU), and frequency weighted IoU (FW-IoU). Among these results, TU-Net (TL) shows better performance than other networks, so in the following experiments we use TU-Net (TL). From these results, we confirm that the proposed TU-Net combining both RGB and IR images outperforms the U-Net with RGB image only.

Table 1. Comparison of U-Net with RGB images and the proposed TU-Net (BL, ML, and TL) with RGB and IR images. Training, evaluation and test dataset at 17:00.

	Pixel accuracy	Mean accuracy	Mean IoU	FW-IoU
U-Net [9]	0.632	0.481	0.239	0.512
TU-Net (BL)	0.678	**0.529**	0.283	0.552
TU-Net (ML)	0.684	0.500	0.262	0.565
TU-Net (TL)	**0.727**	0.497	**0.304**	**0.592**

Exp. 2. In the next experiment we use all dataset from 10:00 to 17:00 and compared the performance of TU-Net (TL) and Siamese TU-Net (BL, ML, and TL). Table 2 shows the 4 evaluation metrics. This shows that the Siamese TU-Net outperforms the TU-Net, and this suggests that the Siamese TU-Net improved the network learning capabilities. Interestingly, Siamese TU-Net BL and ML perform better than Siamese TU-Net TL, possibly due to the following reasons. IR information in addition to RGB information can improve the performance of the terrain classification, but IR information without RGB information shows ambiguity, such as the same temperature among different terrain types due to several factors (e.g. rock in shade shows the same temperature with soil in Fig 1). Thus the new loss function $\mathcal{L}_{MSE}^{IR}$ of TU-Net TL in Fig. 4 (c), which is calculated independent from RGB information, may result in producing ambiguous information. Both Siamese TU-Net BL and ML shows almost the same results, and in the following experiments we use Siamese TU-Net (ML) due to higher pixel accuracy in ML.

Table 2. Comparison of TU-Net (TL) and Siamese TU-Net (BL, ML, and TL). Training, evaluation, and test dataset are images from 10:00 till 17:00.

	Pixel accuracy	Mean accuracy	Mean IoU	FW-IoU
TU-Net (TL)	0.581	0.435	0.211	0.447
Siamese TU-Net (BL)	0.612	**0.542**	**0.307**	0.491
Siamese TU-Net (ML)	**0.620**	0.534	0.303	**0.505**
Siamese TU-Net (TL)	0.563	0.517	0.257	0.463

Exp. 3. In the next experiment, we used the last half dataset (from 14:00 to 17:00) to train and evaluate the network, and tested with (i) the first half (from 10:00 to 13:00) and (ii) the rest (from 14:00 to 17:00). Table 3 shows the 4 evaluation metrics. From these results, test (ii) shows better results than test (i), since the training and evaluation were done with the same time range.

Table 3. Results of Siamese TU-Net ML. Train and evaluate with dataset from 14:00 to 17:00, and (i) test with dataset from 10:00 to 13:00 and (ii) dataset from 14:00 to 17:00.

Test data	Pixel accuracy	Mean accuracy	Mean IoU	FW-IoU
(i) 10:00 $\sim$ 13:00	0.633	0.461	0.279	0.531
(ii) 14:00 $\sim$ 17:00	0.687	0.529	0.374	0.559

3.3. TDeepLab

We also conducted experiments with TDeepLab in the same settings (Exp. 1 $\sim$ 3).

Exp. 1. We tested DeepLab and the proposed TDeepLab BL and TL with images at 17:00 and the results are shown in Table 4. These results show that TDeepLab performs much better than TU-Net (Table 1). Similar to the results of TU-Net, TDeepLab TL shows better performance than TDeepLab BL.

Exp. 2. In the next experiment we use all dataset from 10:00 to 17:00 and compared the performance of TDeepLab (TL) and Siamese-based TDeepLab. The results of the 4 evaluation metrics are shown in Table 5. These results show

Table 4. Comparison of TDeepLab BL and TDeepLab TL. Training, evaluation, and test dataset are images at 17:00.

	Pixel accuracy	Mean accuracy	Mean IoU	FW-IoU
DeepLab [2]	0.871	0.708	0.599	0.781
TDeepLab (BL)	0.859	0.772	0.610	0.792
TDeepLab (TL)	0.911	0.820	0.679	0.848

that the use of siamese technique with DeepLab is not effective.

Table 5. Comparison of TDeepLab (TL), Siamese TDeepLab (BL), and Siamese TDeepLab TL. Training, evaluation, and test dataset are images at 10:00 and 17:00.

	Pixel accuracy	Mean accuracy	Mean IoU	FW-IoU
TDeepLab (TL)	0.911	0.820	0.679	0.848
Siamese TDeepLab (BL)	0.803	0.665	0.482	0.696
Siamese TDeepLab (TL)	0.864	0.731	0.600	0.771

Exp. 3. In the last experiment we trained TDeepLab TL with images from 14:00 to 17:00, and tested images (i) from 10:00 to 13:00 and (ii) from 14:00 to 17:00. Table 6 shows the 4 evaluation metrics. Figure 7 shows visual comparison between TDeepLab TL and Siamese TU-Net ML. These results show that the evaluation metrics of TDeepLab are much better than those of TU-Net, though small rocks in Fig. 7 are well detected by Siamese TU-Net ML.

Table 6. Results of TDeepLab TL. Train and evaluate with dataset from 14:00 to 17:00, and (i) test with dataset from 10:00 to 13:00 and (ii) dataset from 14:00 to 17:00.

Test data	Pixel accuracy	Mean accuracy	Mean IoU	FW-IoU
(i) 10:00 $\sim$ 13:00	0.861	0.816	0.625	0.772
(ii) 14:00 $\sim$ 17:00	0.886	0.786	0.653	0.806

4. Conclusions

In this paper we proposed novel deep learning-based terrain classification methods robust to illumination variations, called TU-Net and TDeepLab and those Siamese-based methods, can efficiently fuse visible and thermal images. Experiments with challenging dataset proved the effectiveness of the proposed methods, and TDeepLab outperformed Siamese TU-Net.

5. Acknowledgment

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

[1] https://www.kaggle.com/c/dstl-satellite-imagery-feature-detection.

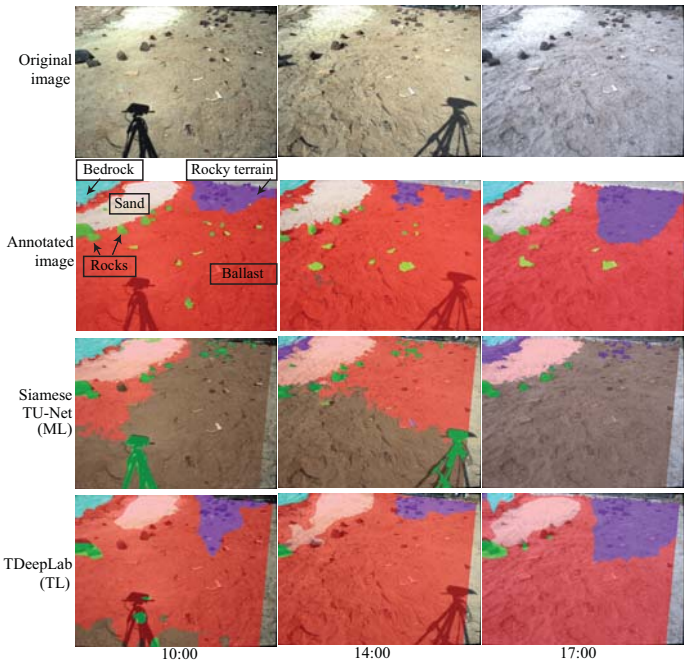


Figure 7. Results of terrain classification at 10:00, 14:00, and 17:00. First, second, third and the last rows show actual image, manually annotated images (ground truth), results of Siamese TU-Net (ML), and results of TDeepLab (TL).

[2] L. Chen et al. Semantic image segmentation with deep convolutional nets and fully connected crfs. *PAMI*, 2017.

[3] O. Cicek et al. 3d u-net: Learning dense volumetric segmentation from sparse annotation. *MICCAI*, 2016.

[4] D. Ciresan et al. Deep neural networks segment neuronal membranes in electron microscopy images. *NIPS*, 2012.

[5] J. Den et al. Imagenet: a large-scale hierarchical image database. *CVPR*, 2009.

[6] P. Filitchkin et al. Feature-based terrain classification for littledog. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[7] I. Halatci et al. Terrain classification and classifier fusion for planetary exploration rovers. *IEEE Aerospace Conference*, 2007.

[8] J. Long et al. Fully convolutional networks for semantic segmentation. *CVPR*, 2015.

[9] O. Ronneberger et al. U-net: Convolutional networks for biomedical image segmentation. *MICCAI*, 2015.

[10] B. Rothrock et al. Spoc: Deep learning-based terrain classification for mars rover missions. *AIAA SPACE*, pages 5539–5551, 2016.

[11] N. Salamati et al. Semantic image segmentation using visible and near-infrared channels. *ECCV 2012. Workshops and Demonstrations. Lecture Notes in Computer Science, Springer*, 7584, 2012.

[12] A. Valada et al. Robust semantic segmentation using deep fusion. *Robotics: Science and Systems (RSS 2016) Workshop, Are the Sceptics Right? Limits and Potentials of Deep Learning in Robotics*, 2016.