CEAS-GNC-2022-059

CEAS EuroGNC 2022
"Conference on Guidance, Navigation and Control"
3-5 May 2022 @ Technische Universität Berlin, Germany

Attention-based DeepMoon for Crater Detection

Jianing Song Postdoctoral Research Fellow, City University of London, Department of Electrical and Electronic Engineering, EC1V 0HB, London, United Kingdom. Jianing.Song@city.ac.uk

Nabil Aouf Professor of Robotics and Autonomous Systems, City University of London, Department of Electrical and Electronic Engineering, EC1V 0HB, London, United Kingdom. Nabil.Aouf@city.ac.uk

Christophe Honvault European Space Agency, Keplerlaan 1, 2201AZ Noordwijk, The Netherlands christophe.honvault@esa.int

ABSTRACT

Aiming at the potential application of explainable artificial intelligence techniques in space tasks, we propose an attention-based deep network for crater detection during Lunar landing scenarios. A methodology combining a fully convolutional neural network and self-attention modules is developed to explore the explainability of automatically crater detection from images. By applying the transfer learning technique, the DeepMoon model is selected as the backbone of the proposed pipeline, and an encoder-decoder based architecture is therefore established and evaluated. The crater images generated using the Blender platform are trained and tested to estimate the performance of the mythology. The self-attention module and data augmentation techniques are applied to the dataset to enhance the segmentation results and improve the generalisation of the implementation. The experiment results on the synthetic greyscale dataset show that the precision, recall, and F1 scores of the crater detection results achieve 0.86, 0.84, and 0.83, respectively. The explainability of the proposed network is achieved by visualisation of each attention map of the attention modules, showing that the deeper attention module pays more attention to the craters.

Keywords: Attention Mechinasim; DeepMoon; Crater Detection; Deep Learning

Nomenclature

$\sigma(\cdot)$,	=	Sigmoid function
$C \times H \times W$	=	Dimension of feature maps, where C, H, and W represents the channels, height, and width.
$f^{7\times7}$	=	A 7×7 kernel of a convolutional block.
F1, P, R	=	Metrics of F1 score, precision, and recall respectively.
F_n, F_p, T_p	=	False negatives, false positives, true positives.
x_i, y_i	=	Groundtruth output target value and the network predicted one of pixel i.
$\otimes$	=	Element-wise multiplication.
$\boldsymbol{F}$	=	Feature map.
$\boldsymbol{M}_c$, $\boldsymbol{M}_s$	=	Channel attention map, spatial attention map
$\boldsymbol{W}_0$, $\boldsymbol{W}_1$	=	Network weights

List of Acronyms

2D	two-dimensional
3D	three-dimensional
BCE	Binary Cross-Entropy
BN	batch normalization
CBAM	convolution block attention module
CD	Crater Detection
CI	Crater Identification
CNN	Convolutional Neural Network
CRO	Candidate for a Regional Object
DEM	Digital Elevation Map
DL	Deep Learning
DNN	Deep Neural Network
FCL	Fully Connected Layer
IQR	interquartile range
LRO	Lunar Reconnaissance Orbiter
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
PyCDA	Python Crater Detection Algorithm
RPN	Region Proposal Network
SENet	Squeezeand-Excitation Network
TRN	Terrain Relative Navigation
WAC	Wide Angle Camera

1 Introduction

Craters are small depressions on the surface of a planet, Moon, or asteroid, created by the hypervelocity impact of smaller objects. The detection and study of craters have been attracting a lot of interest from both scientists and engineers. Scientific issues, such as relative age measurement of the planetary surface, nature of degradational processes, and planetary geomorphology, can be studied from craters. Moreover, craters are ideal landmarks for relative navigation on or around the Moon and asteroids [1, 2]. Specifically, the development of optical techniques leads to crater-based autonomous spacecraft navigation being a popular and reliable method.

Natural craters vary in shapes, including morphology, peak rings, central pits, wall terraces, or maybe overlap with others [3]. Conventional crater detection can be classified as the unsupervised detection method, which detects the edges of the crater by extracting the circular or elliptical features using image processing and target detection. For example, Ref. [4] use the Hough Transform and the Radial Consistency measure to detect craters on Earth and Mars from remote sensing images. Ref. [5] develops an improved morphological image processing and fast Fourier transform-based template matching methodology for detecting craters from planetary surface. Ref. [6] studies an automated counting and template generating method to detect very small craters in LROC NAC images of the Moon. Recently, Ref. [7] combines the Hough transform and unsupervised neural network to detect craters in the image. The unsupervised conventional crater detection methods do not need to label a large number of samples. However, they are not robust to complex terrain detection, as the imaged craters can be diverse in dimensions and appearance because of varying illumination conditions and different poses of onboard cameras [1].

Convolutional Neural Network (CNN)-based crater detection algorithms emerged due to the advances in computer vision and successful applications of CNNs in the object detection area. Most earlier methods utilise a CNN as a classifier to validate selected features for crater detection, such as Refs. [8–

10]. Recent research has given more attention to robust crater detection algorithms that utilise CNNs to fully process raw crater images. For instance, Ref. [11] develops an open-source crater detection library named Python Crater Detection Algorithm (PyCDA) to detect new craters using U-Net architecture from greyscale images. Ref. [1] proposes the CrateIDNet, an end-to-end fully CNN, to detect craters from also greyscale images. Additionally, Refs. [12, 13] study the CNN-based crater detection methods from Digital Elevation Map (DEM) images by presenting DeepMoon and LunaNet, respectively. Moreover, Ref. [14] uses multiple neural networks to process digital terrain model and thermal infrared imagery to identify and locate craters on Mars, where the ResUNETs are used to extract crater rims from images.

CNNs-based algorithms provide a lot of potential for improving the performance of crater detection. However, some of them are unable to explain their predictions, which poses a challenge to their usage in critical systems, such as space applications. Previous work has developed various CNNs for crater detection, while few studies have looked into the explainability of the proposed networks in terms of crater detection tasks. Thus, the motivation of this study is to investigate explainable crater detection CNN that plays an end-to-end manner for a more reliable crater detection for space applications, which attention-based DeepMoon network is proposed. The remaining part of the paper proceeds as follows: Section 2 presents the literature review of previous CNN-based crater detection works and an overview of the attention mechanism. Section 3 details the methodology of the proposed algorithm, including the attention-based DeepMoon model, data augmentation, and evaluation metrics. Section 4 gives experiments and discussion. In Section 5, concluding remarks end the work.

2 Related work

2.1 Review of CNN-based Crater Detection

The PyCDA is an open-source crater detection library composed of a detector, extractor, and classifier, which focuses on detecting new craters that have never been catalogued [11]. PyCDA uses a downsized U-Net architecture to compute the per-pixel likelihoods of a crater rim from inputs of greyscale intensity images. The pixel prediction map is then fed to the extractor to generate a list of crater candidates. A classifier CNN is finally applied to determine true craters. Thanks to PyCDA, numerous craters have been detected and categorised, thus helping to generate new labelled datasets for training and testing of Deep Learning (DL) algorithms.

CrateIDNet is an end-to-end fully CNN for simultaneous crater detection and identification [1]. CraterIDNet takes remote sensing images in various sizes and outputs detected crater positions, apparent diameters, and indices of the identified craters. Instead of using large off-the-shelf Deep Neural Network (DNN) models, a small CNN architecture pre-trained on Martian crater samples [15] is first developed to extract feature maps. Next, two pipelines, namely Crater Detection (CD) and Crater Identification (CI) are proposed for simultaneous detecting and identifying craters. The CD process involves detecting the presence of craters and locating them within the image if they exist. The output of CD is then fed to the CI process to match the detected craters to surface landmarks in a known database, and matches of CI will provide position estimation. The CD modifies the Region Proposal Network (RPN) architecture [16] as the backbone, regressing objectness scores and crater diameters from feature maps. Experiments reveal that the light CraterIDNet with a size of 4 MB performs better than previous algorithms [9].

LunaNet framework is proposed in Ref. [13] to detect craters for lunar Terrain Relative Navigation (TRN) and take greyscale images as inputs. Thus, the method is more suitable for implementation aboard a spacecraft equipped with an optical camera without the need for a depth sensor. The output of the CNN is a crater rim prediction mask. Each feature extraction step of the LunaNet includes prediction mask, eroded and thresholded prediction, contour detection, and ellipse fitting. The data preparation is with the Lunar Reconnaissance Orbiter (LRO) Wide Angle Camera (WAC) Global Lunar DEM dataset

[17], followed by a histogram rescaling of the input greyscale images to match the intensity distribution of a DEM image. The LunaNet reduces the training effort and final detection results by applying the transfer learning from the DeepMoon model. Experimental results indicate that LunaNet's performance surpasses DeepMoon and PyCDA in terms of robustness to noisy images, location accuracy, and average crater detection time.

It has been observed that areas with low solar angles have heavy shadowing, resulting in reduced crater detection reliability. To identify high-value landmarks by optical navigation systems, Ref. [18] employs a CNN-based object detector to distinguish likely landmark candidates and predict detection probabilities along various lighting geometric flight paths. A massive dataset based on real lunar-surface data is collected. A Candidate for a Regional Object (CRO) is defined as an image object with specific latitudes and longitudes. The LunaNet architecture is then used and trained to identify CROs by maximising the discrimination between local areas of the Moon. Finally, the CRO performance map is formed based on the scored CROs arranged by considering the azimuth and elevation angles of the Sun during the year. Numerical experimental results demonstrate that the proposed landmark detection pipeline can provide usable navigation information even at Sun angle elevations of less than 1.8 deg in highland areas, which indicates a successful application for the worst dark highlands near the South Pole.

2.2 DeepMoon Network

DeepMoon network is proposed in Ref. [12], employing a CNN architecture for robust crater detection on the lunar surface using DEMs. The method relies on the developed DeepMoon network to identify the craters in terms of their centroid and radii and outputs pixel-wise confidence maps of crater rims on the surface of a rocky body. DeepMoon modifies U-Net [19] by changing the input image size, the number of filters in each convolution layer, and the use of dropout [20] in the expansive path for memory limitations and regularisation, respectively. Fig. 1 presents the architecture of the DeepMoon network.

The input of DeepMoon are 256×256 pixels Lunar DEM images, and correspondingly, 256×256 pixels target with activated pixels corresponding to the locations of the crater rims is the output from a final layer following the expansive path. DeepMoon differs from the original U-Net by the number of filters in each convolution layer and the use of dropout in the expansive path. In the contracting path, each convolutional layer in blocks 1, 2 and 3 contain 112, 224 and 448 filters, respectively. While filters of blocks 5, 6 and 7 in the expansive path are 224, 122 and 122, respectively. Each convolutional layer in the connecting path contains 448 filters. The total trainable parameters of DeepMoon is 10 278 017.

For training, the data used in DeepMoon is generated by merging two human-generated crater catalogues, which are the LRO WAC Global Lunar DEM [17] and the LRO Lunar Orbiter Laser Altimeter DEM [21]. The dataset is split into equal train-validation-test parts, yielding 30 000 DEM images per part. The minimised loss function is chosen as the pixel-wise Binary Cross-Entropy (BCE). DeepMoon produces a crater rim prediction mask that further feeds to a low-level image process and a template matching procedure to determine the actual craters. The median fractional longitude, latitude and radii errors are 11% or less, representing good agreement with the human-generated datasets. Additionally, transfer learning from training on lunar maps to testing on maps of Mercury is demonstrated successfully in qualitative.

2.3 Attention mechanism models

Attention mechanism is born to investigate the links between inputs and outputs of DL models in the field of natural language processing [22]. Because of the significant improvements made by attention mechanisms in machine translation, it soon expands to the computer vision field [23], and various other types of attention mechanisms are explored [24].

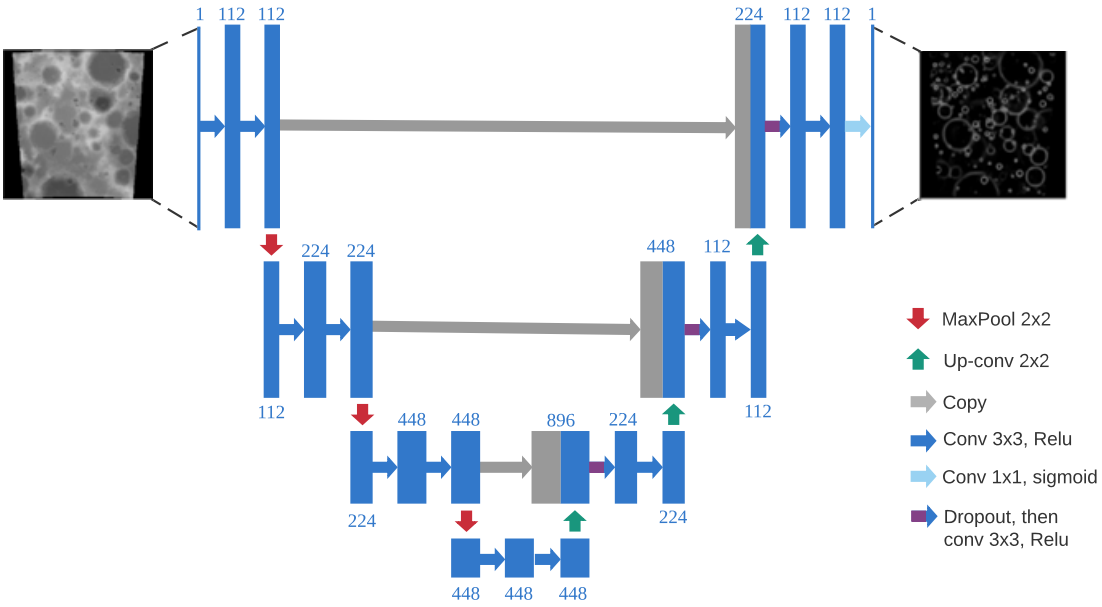


Fig. 1 Architecture of DeepMoon network (Reproduced from [12]).

In Ref. [23], the image is first encoded to extract features by a CNN. The Long Short-Term Memory (LSTM) decoder then consumes the convolution features to generate descriptive phrases, with the weights learned through attention. The attention weights are visualised to show the model pays attention to which portions of the image produce a specific word. The research also introduces the distinction between soft and hard attention, depending on whether the attention has access to the complete image or merely a patch of it. The alignment weights of soft attention are learned and softly assigned the whole patches in the source image, whereas hard attention only attends to one patch of the image at a time.

The soft attention-based model is smooth and differentiable, but the hard attention-based model is non-differentiable and generally trained by reinforcement learning. Therefore, we focus on soft attention methods in this report. In terms of the attention domain, soft attention can be classified into spatial domain[25], channel domain[26], mixed domain[27, 28] etc.

2.3.1 CBAM Attention Module

Ref. [28] studies the spatial and channel attention in convolution block attention module (CBAM), and the corresponding structure is given in Figure 2. Assuming an intermediate feature map $\boldsymbol{F} \in C \times H \times W$ as input, CBAM sequentially infers a 1D channel attention map $\boldsymbol{M_c} \in C \times 1 \times 1$ and a 2D spatial attention map $\boldsymbol{M_s} \in 1 \times H \times W$. The following is a summary of the overall attention process,

$$\boldsymbol{F'} = \boldsymbol{M_c}(\boldsymbol{F}) \otimes \boldsymbol{F} \tag{1}$$

$$\boldsymbol{F''} = \boldsymbol{M_s}(\boldsymbol{F'}) \otimes \boldsymbol{F'} \tag{2}$$

where $\otimes$ represents element-wise multiplication and $\boldsymbol{F''}$ is the refined feature output.

2.3.2 Channel Attention Module

Channel attention is developed to boost the representational power of a network by enhancing spatial encoding. Ref. [26] focuses on the channel relationship and proposes the Squeezeand-Excitation Net-

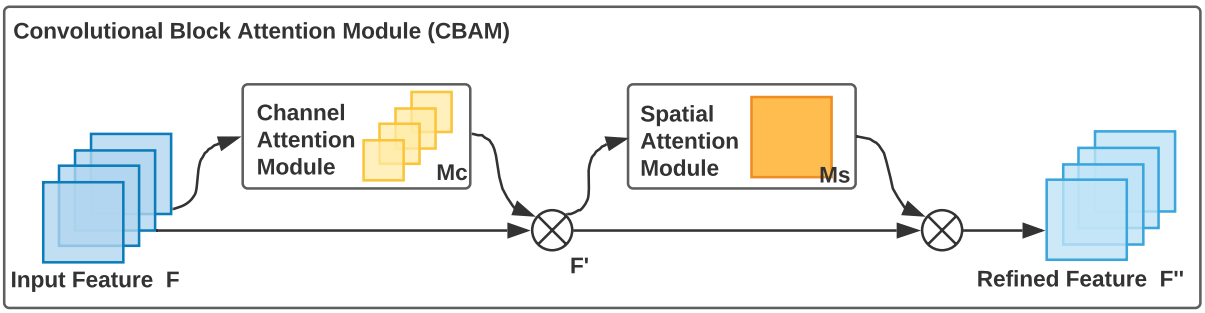


Fig. 2 The convolution block attention module (Reproduced from [28]).

work (SENet). The channel attention module of SENet consists of three parts: squeeze, excitation, and scaling(attention). First, the AdaptiveAvgPool is applied to the spatial dimension. For example, the input dimension of 7×7 is squeezed to the output of 1×1. Next, the channel attention is learnt through two Fully Connected Layers (FCL), followed by the Sigmoid function to normalise, which this step is called excitation. Finally, the learned matrix by excitation step is multiplied with the original matrix to get the feature weighted by the channel dimension (e.g. each 7×7 feature map is multiplied by 1×1 weight).

The channel attention module in [28] (see Fig. 3) is nearly identical to [26], except that MaxPool is added to get more unique channel characteristics. The methodology of channel attention in Ref. [28] is: (1) Performing both average-pooling and max-pooling on the input features, generating two different spatial descriptors denoted as $\boldsymbol{F}^c_{avg}$ and $\boldsymbol{F}^c_{max}$, respectively. (2) Feeding both descriptors to a shared Multilayer Perceptron (MLP), where the number of channels is first compressed and then reconstructed. (3) Element-wise summation of the output feature vectors from MLP. The Sigmoid function is then connected to generate attention weights of $0\sim1$ between channels. (4) Multiplying the scale back to the original input feature. Mathematically, the channel attention of CBAM is computed as,

$$\begin{aligned}\boldsymbol{M_c}(\boldsymbol{F}) &= \sigma(MLP(AvgPool(\boldsymbol{F})) + MLP(AvgPool(\boldsymbol{F}))) \\ &= \sigma(\boldsymbol{W}_1(\boldsymbol{W}_0(\boldsymbol{F}^c_{avg})) + \boldsymbol{W}_1(\boldsymbol{W}_0(\boldsymbol{F}^c_{max})))\end{aligned} \tag{3}$$

where σ denotes the Sigmoid function. The MLP weights are $\boldsymbol{W}_0 \in C/r \times C$, and $\boldsymbol{W}_1 \in C \times C/r$, followed by ReLU activation function.

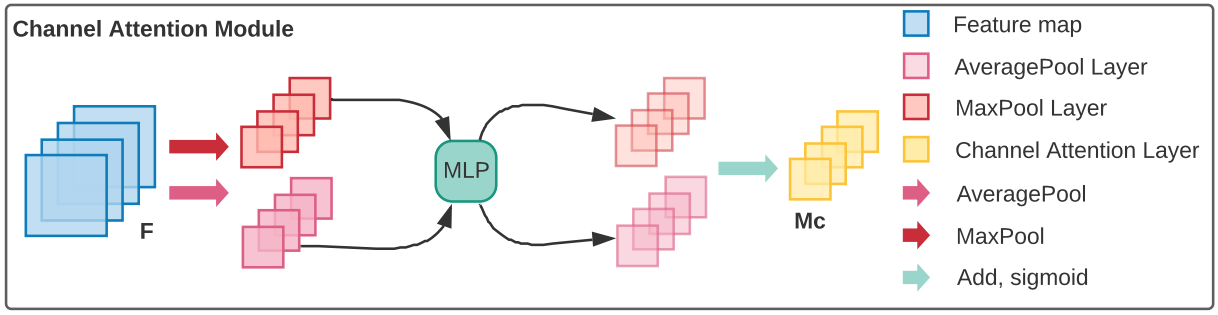


Fig. 3 Channel attention module of CBAM (Reproduced from [28]).

2.3.3 Spatial Attention Module

A Spatial Attention Module generates spatial attention in convolutional neural networks by utilizing the inter-spatial relationship of features. Spatial attention focuses on where is an informative part, which is complementary to channel attention. To compute the spatial attention of CBAM, there are steps: (1) Both Global max-pooling and Global average-pooling operations are applied to input features along the

channel axis and (2) concatenated to generate an efficient feature descriptor. (3) Applying convolution operation, the demission of the channel decreases to 1. (4) The above results are fed to the Sigmoid function to generate spatial attention feature. The spatial attention feature furthermore can be multiplied with the input feature map to increase spatial attention. The mathematical expression of the spatial attention is,

$$\begin{aligned} \boldsymbol{M_s}(\boldsymbol{F}) &= \sigma(f^{7\times7}([AvgPool(\boldsymbol{F}); MaxPool(\boldsymbol{F})])) \\ &= \sigma(f^{7\times7}([\boldsymbol{F}^s_{avg}; \boldsymbol{F}^s_{avg}])) \end{aligned} \tag{4}$$

where $f^{7\times7}$ is a 7×7 kernel of a convolutional block, $\boldsymbol{F}^s_{avg} \in \mathbb{R}^{1\times H\times W}$, $\boldsymbol{F}^s_{max} \in \mathbb{R}^{1\times H\times W}$

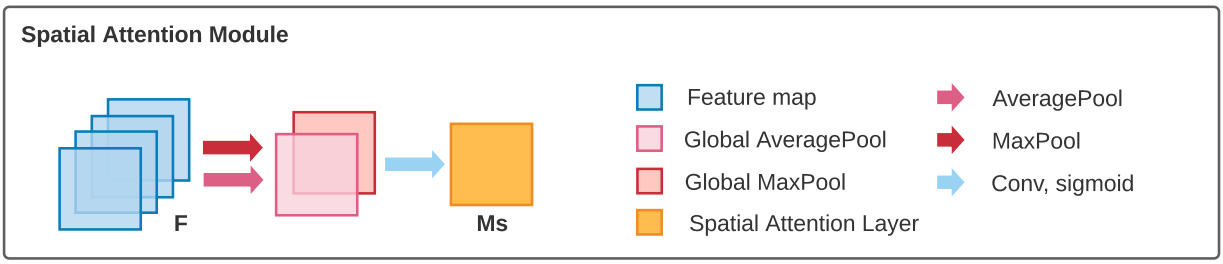


Fig. 4 Spatial attention module of CBAM (Reproduced from [28]).

3 Methodology

In this work, we propose an attention-based DeepMoon framework that exploits either DEM images or greyscale images. The proposed network takes advantage of attention mechanism techniques and DeepMoon structure, which performs a better segmentation for crater detection with the explainability.

3.1 Attention-based DeepMoon Using Transfer Learning

Fig. 5 shows a framework of attention mechanism-based DeepMoon for crater classification problems. We present a similar architecture from DeepMoon, which fully introduces the short-cut layers to combine the encoder information and decoder information. The existing architecture contains many feature channels in the upsampling part, which allows the network to propagate context information to higher resolution layers. Consequently, the expansive path is more or less symmetric to the contracting part and yields an encoder-decoder architecture. The network only uses the valid part of each convolution without any fully connected layers. The attention-based DeepMoon network can be first trained by using a transfer learning of DeepMoon, where the pre-trained parameters are given to achieve an efficient training process. Additionally, self-attention modules are motivated by how we pay visual attention to different regions of an image, which gives a way to show the explainability of the network. Thus, we introduce two self-attention modules for further enhancing the performance and explainability of crater detection. The proposed attention modules combine the spatial attention module and channel attention module, similarly to CBAM but with adding a batch normalization (BN) layer before the activation function. The spatial and channel attention modules are utilised in both encoders and decoders of the pixel-level detection models.

As discussed in Section 2, the size of the feature map introduces to the self-attention modules is $C\times H\times W$, where C is the number of channels, and $H\times W$ is the height and width of the feature map. The channel attention module is utilised to model and dig the inter-dependencies between feature channels. The proposed feature map is first reshaped and fed into fully connected layers and BN layers, which generates a new feature map with the size of $1\times1\times C$. After that, the feature map is fed to a

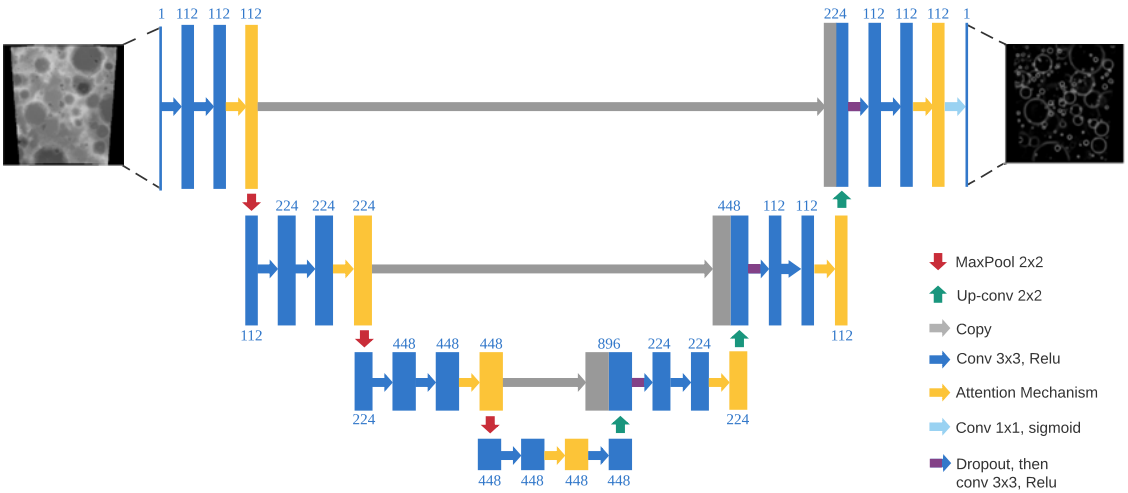


Fig. 5 Architecture of the proposed attention-based DeepMoon network.

sigmoid layer to calculate the channel attention map. Then the generated feature map and an element-wise multiply with the input feature map are used to obtain the output of the channel attention modules. In the spatial attention module, the feature map is input to the convolutional block with a kernel size of 3×3 and BN layers, which generates a new feature map. Then the new feature map is again fed into the convolutional block with activation layers to reshape its channels and generate a new feature map with the size of $1 \times H \times W$. The sigmoid function is widely employed to map the non-normalised output of a neural network to a probability distribution over predicted classes. Finally, an element-wise multiply operation is applied to the generated feature map to obtain the final output of attention modules.

3.2 Lunar Landing Scenario

To evaluate the proposed network, we first generate the dataset of crater images using the Blender platform. Blender is an open-source 3D creation that is suitable for creating planet and crater models. Figure 6 illustrates different phases of descent and landing on the lunar surface, which craters are ideal markers for relative navigation. First, crater detection is achieved by DNN frameworks and further used to estimate poses in the Lunar landing scenario.

In this study, we investigate the last landing phase that supposes the landing path is a projectile trajectory ranging from 1500 m to 200 m in altitude, as shown in Figure 6. The motivation for the choice of this scenario comes from two following aspects: 1) During the last landing phase, the craters in images captured by onboard camera change variety with the landing trajectory, which challenges conventional crater detection methods as most of them considering images that generated by an orbital camera. 2) Additionally, the detected craters in this landing scenario can be further applied in hazard detection and avoidance for Lunar landers. Therefore, the area around the Apollo 12 landing site is selected to build the scenario, in which the corresponding 3D model is loaded from NASA 3D resource[1]. The rendering output images are generated by 3D models built in Blender.

3.3 Database Collection

In this research, the DeepMoon dataset is first collected for the training and evaluation, which are generated by randomly cropping DEM images from the Lunar Reconnaissance Orbiter and Kaguya merged digital elevation model [29]. Furthermore, a synthetic dataset is generated for refined training, evalua-

[1] https://nasa3d.arc.nasa.gov/search/lunar

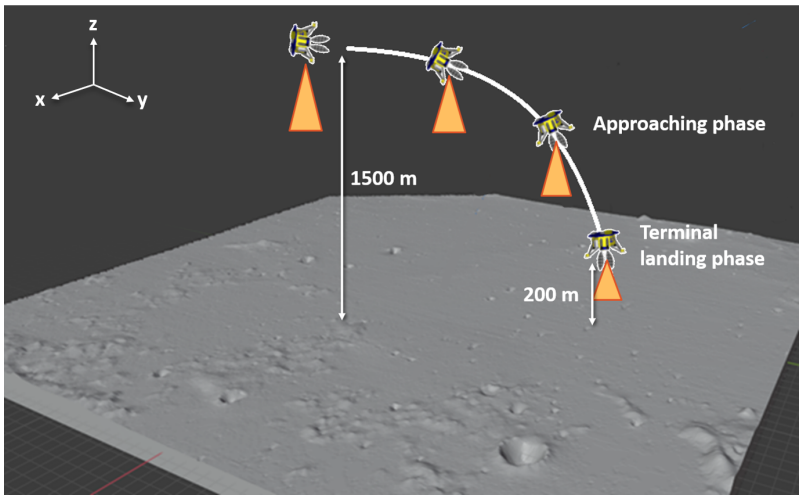


Fig. 6 Scenario built in Blender for Lunar crater detection.

tion, and testing. The data collection phase has been performed using the camera and Rendering function provided by Blender. Output images are generated as follows:

- **Parabola trajectory**: 8 trajectories along ±x, ±y, ±x±y, z range(from 1500 m to 200 m)
- **camera**: tracking the Lunar surface along the -z direction, which camera follows the aforementioned parabola trajectories.
- **Rendering**: Cycles and Workbench rendering methods, 100 frames
- **Output**: Depth images by Cycles rendering method, Visual RGB images by Workbench rendering method, size of 256 ×256 pixels

The groundtruth for DNN training dataset is labelled manually, shown as Fig. 7. First, black balls are used to cover the craters shown in the 3D model, of which the total number of craters is 94. Second, images, namely groundtruth images, are captured by a camera following the same trajectories. Next, We import the groundtruth images in MATLAB as a 2D matrix for each frame. Then a circle detection algorithm is applied to obtain the list of the position and radii of each crater in the image frame. Finally, the mask images are generated from the computed crater list, and greyscale images are used as an input for the network. By limiting the size of detecting craters using data preprocessing, 764 images are generated for developing and evaluating the proposed network.

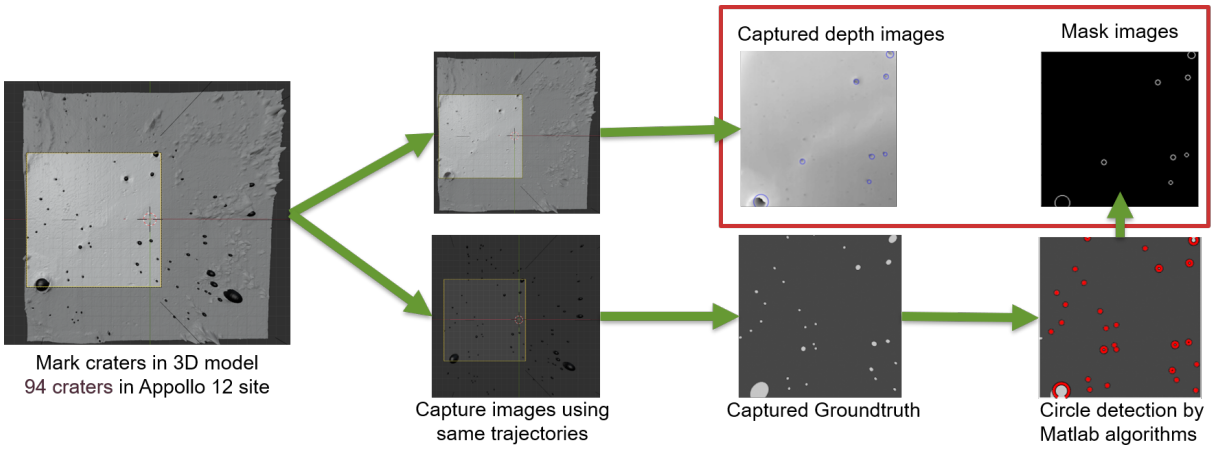


Fig. 7 Pipeline of groundtruth mask generation.

3.4 Accuracy Metrics

Several useful metrics are introduced to evaluate the performance of the proposed model. The pixel-wise BCE is used as the loss function for training the network, which is a standard loss function used for segmentation problems:

$$\text{BCE} = -\frac{1}{N}\sum_i y_i log(p(y_i)) + (1-y_i) log(1-p(y_i)) \tag{5}$$

where y is the label (1 for "crater" and 0 for "no crater"), $p(y_i)$ is the predicted probability of the pixel i being a crater, and N is the total number of pixels. The $p(y_i)$ values are the classification output of the CNN, which are subject to a sigmoid function. Since it is a binary classification problem, it takes the simplified form of a sigmoid function:

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{6}$$

Thus, by substituting the $p(y_i)$ with Eq. 7, Eq. 5 can be rewritten as,

$$\text{BCE} = \frac{1}{N}\sum_i x_i - x_i y_i + log(1+exp(-x_i)) \tag{7}$$

where x_i is the network predicted one of pixel i.

After generating the predicted the binary image from network, template matching is then applied to the output image to extract detected craters. The recall(R), precision(P), and F1 score are utilised to analyse the matched craters with the ground truth to further validate the performance of proposed network. Precision is the ratio of correctly predicted positive to the total predicted positive. Recall is the ratio of correctly predicted positive to the all observations. F1 Score is the weighted average of Precision and Recall, which takes both false positives and false negatives into account. Following gives the corresponding definition and formula:

$$\text{R} = \frac{T_p}{T_p + F_p} \tag{8}$$

$$\text{P} = \frac{T_p}{T_p + F_n} \tag{9}$$

$$\text{F1} = 2\frac{PR}{P+R} \tag{10}$$

where T_p are true positives, F_p are false positives, and F_n are false negatives.

Furthermore, considering the further application of detected craters in the navigation system, the detected craters in the image frame can be identified and matched to a local crater database in the landing site frame. In that case, by solving a two-dimensional (2D)-three-dimensional (3D) matching problem the pose of an onboard camera can be determined. By knowing the configuration matrix of the camera, we can compute the pose of the spacecraft with respect to the landing site frame. As the detection error on crater position and radii will affect the performance on crater identification, and then translate to the navigation error, we, therefore, analyse the median and interquartile range (IQR) of the position and radii of detected craters.

4 Experiments

4.1 Training Initialisation

The DeepMoon dataset contains 30 000 DEM images which the median number of craters per DEM image is 21. We apply transfer learning to train the proposed attention-based DeepMoon by taking advantage of the trained DeepMoon on its corresponding dataset for the DEM-based crater detection task. We then use the learned features, transfer them to our proposed attention-based network, training on the Blender-generated dataset for crater detection from greyscale images. To do so, the pretrained layers and parameters of the DeepMoon are first frozen in the proposed network and train the rest layers by using the DeepMoon dataset. Secondly, we unfroze the layers of DeepMoon and fine-tuning the whole proposed network with both the DeepMoon and Blender-generated datasets.

We divide the Blender dataset into three essential parts, training, validation and testing, showing in Figure 8. Additionally, data augmentation in data analysis is the technique utilised to enhance the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. Its effectiveness has been verified in many computer vision tasks. Thus, various mainstream data augmentation techniques are introduced and used in this task, such as random colour inversion, horizontal/vertical flips, random row or column pixel shifts, and random 90 deg rotations. The ground-truth is applied for the evaluation and test of the crater detection model. We implement all of our models using the Tensorflow framework. All methods are trained in Python environment with an Inter Xeon W-10855M CPU with 64-GB RAM and Quadro GTX 5000 GPU with 16GB RAM. The learning rate is initially 1E-4 for the training of frozen backbone model layers and the training of fine-tuning. The batch size of the training phase is 8 due to the computational efficiency. The filter size and dropout are set to be 3×3 and 0.15, respectively.

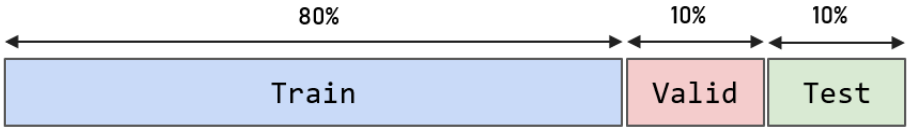


Fig. 8 Dataset splitting.

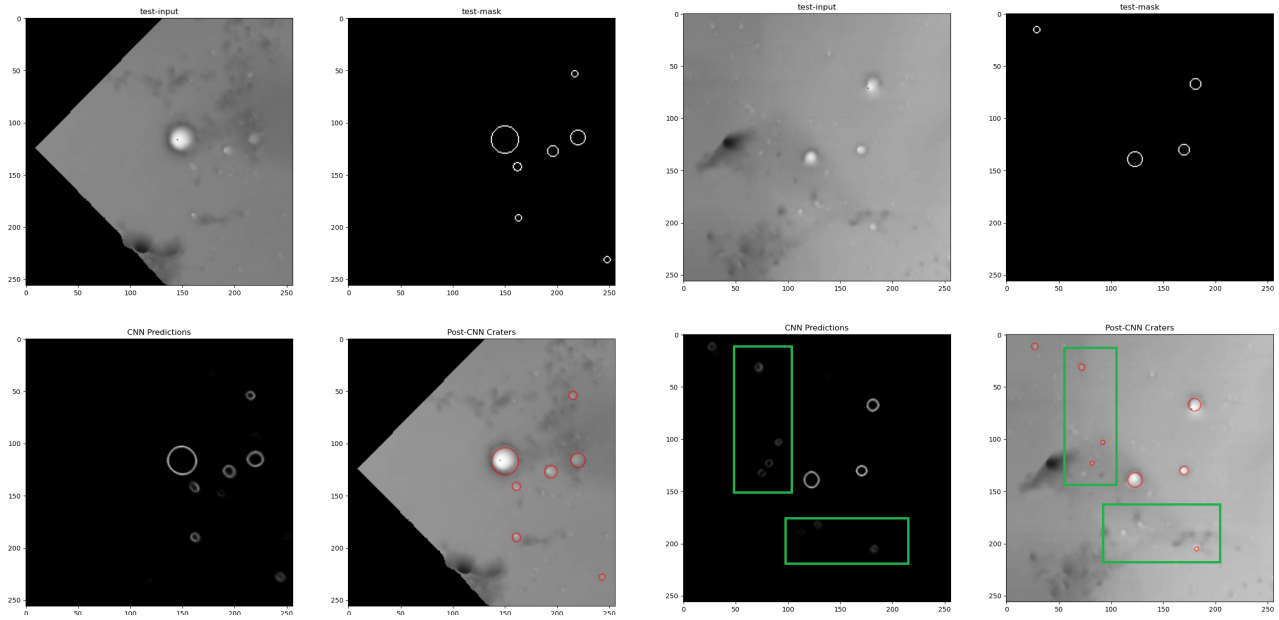
Table 1 Score comparison of the validation and test on the synthetic dataset for DeepMoon and Attention-based DeepMoon

Model	Image Type	Dataset	Accuracy Metric			
			Recall	Precision	F1 score	BCE
DeepMoon	DEM	Validation	78% ± 22%	90% ± 15%	82% ± 17%	0.011 029
		Test	74% ± 18%	89% ± 15%	80% ± 14%	0.015 370
	Greyscale	Validation	72% ± 21%	90% ± 16%	78% ± 18%	0.011 724
		Test	71% ± 22%	88% ± 16%	76% ± 18%	0.012 256
Attention-based DeepMoon	DEM	Validation	87% ± 18%	83% ± 17%	84% ± 15%	0.010 199
		Test	84% ± 15%	85% ± 16%	83% ± 13%	0.010 457
	Greyscale	Validation	84% ± 17%	87% ± 15%	84% ± 13%	0.010 528
		Test	86% ± 17%	84% ± 16%	83% ± 14%	0.011 045

4.2 Experimental results and discussion

Post-process is applied to the output of the crater detection network, aiming to extract crater coordinates (in pixels) from the CNN-predicted target and compare the results to the corresponding human-counted

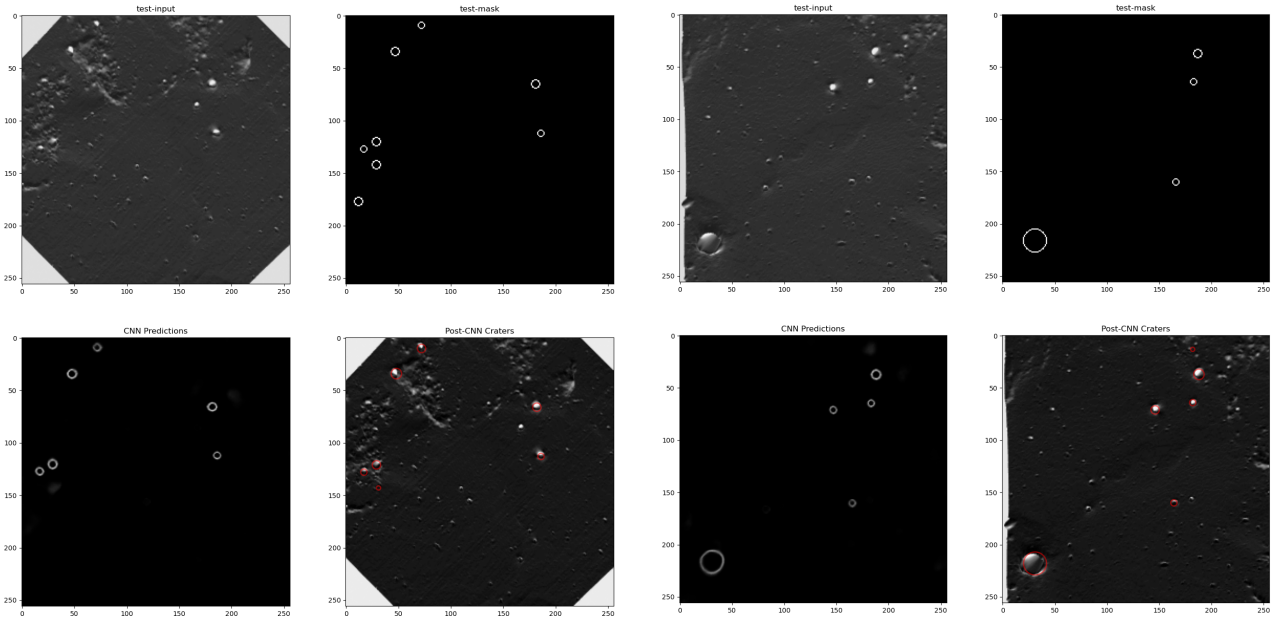
crater data. The minimum and maximum ring radius to search target over are set to be 3 pixels and 50 pixels, respectively. We apply the proposed attention-based DeepMoon pipeline for explainable crater detection on the Blender-based synthetic dataset and list the various accuracy metrics in Table 1 for both the validation and test datasets. To compare the performance with the DeepMoon, Table 1 also shows the score of DeepMoon on our synthetic dataset. As we can see, while the Recall has a marked growth after adding self-attention to DeepMoon, the Precision decreases slightly, which results in a bit better performance in terms of F1 score. Moreover, the BCE metric achieved by the proposed network supports the conclusion that the attention-based network outperforms the DeepMoon model. Additionally, the similarity between our validation and test set statistics achieved by the proposed method, as shown in Table 1, suggesting that overfitting is avoided almost or completely. Especially for the greyscale image dataset, a small decrease from the test dataset to the validation dataset indicates good robustness of the attention-based DeepMoon on greyscale images for further on-board application.



(a) Case of wrong-size detected craters **(b) Case with detected craters but not labelled**

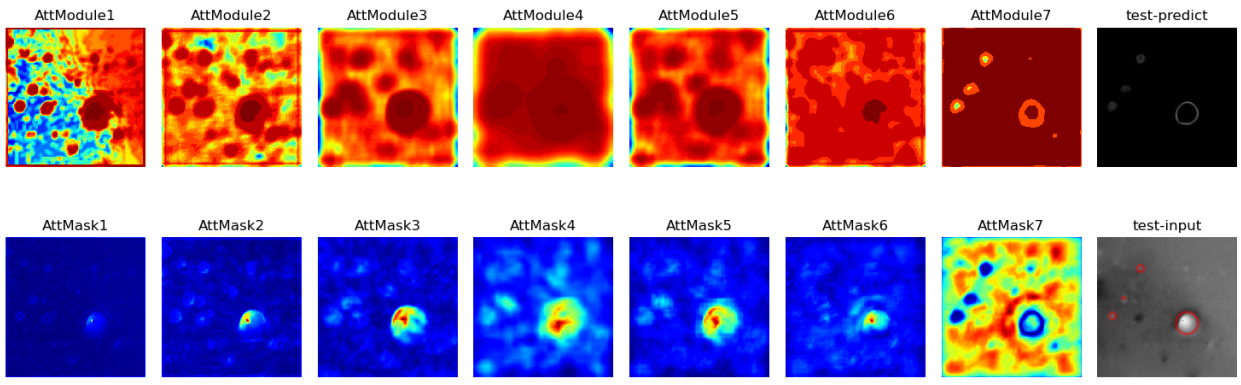
Fig. 9 Results on DEM images by proposed pipeline for crater detection.

Figures 9 and 10 show the results by the attention-based DeepMoon and post-process of the input images in different sampling trajectories, illustrating that the proposed crater detection method can extract craters from both DEM and greyscale images. However, the performance is affected by the illumination, as shown in Fig. 9a where detected craters are in a smaller size than the groundtruth caused by light. Furthermore, with a good light condition, the proposed network is able to detect the craters in small radii that are not labelled. As can be seen, the green rectangle in Figure Fig. 9b shows the craters that can be detected by the network but are not labelled because of small radii. A similar but slightly worse performance can be achieved with the visual images due to the lack of depth information, showing in Fig. 10. The explainability of the crater detection results is achieved by the visualisation of attention maps. To verify whether self-attention focuses on the craters, Fig. 11 shows the attention maps for each attention module layer on synthetic greyscale/DEM dataset. In Fig. 11, the AttModule represents for the output of sigmoid function of the spatial attention, and the AttMask represents for a refined feature map, which the output of sigmoid is multiplied by the input feature. Thus, the AttModule show where the attention focus to, while the AttMask is a visualisation of feature map, representing the value of feature map calculated by the mean of feature maps in different channels. When visualising the attention maps, the deeper attention map of the proposed network highlights craters more accurately. Moreover, this attention map can focus on multiple craters in one image.

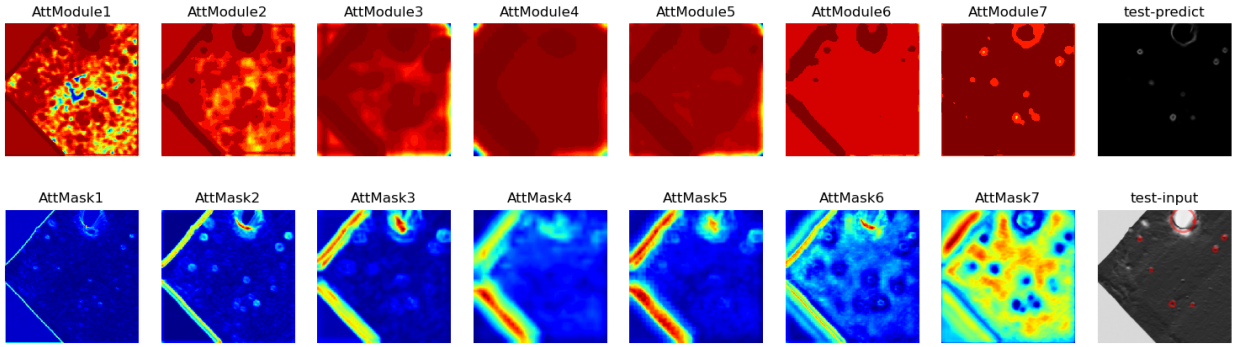


(a) Case with non-detected results **(b) Case with false detection results**

Fig. 10 Results on greyscale images by proposed pipeline for crater detection.



(a) DEM images



(b) Greyscale images

Fig. 11 Visualization example of mask-attention in different layers.

For the purpose of crater-based navigation, the median and IQR error of position and radius of the detected crater are calculated, with the result of 0.320142, 0.219679 (25%), 0.424127 (75%) in vertical axis, 0.223745, 0.175686 (25%), 0.307039 (75%) in horizontal axis, 0.147974, 0.095238 (25%), 0.197090 (75%) in radius. The effects of detected error on navigation performance are out of this paper and will be investigated in future. The proposed network contains a total parameter number of 10,651,775 with a size of 122 MB, and it uses ∼0.3853 s to predict the output image from the input greyscale image at each frame. To study the capability of the proposed network implementation on the onboard system, the codes need to be optimised and then translate to the embedded system for further analysis.

5 Conclusion

In this paper, an effective attention-based DeepMoon model is proposed to give the explainability of the DeepMoon network and improve the data processing efficiency. A new crater dataset is generated based on the Blender platform. The experiment results on the generated dataset demonstrate that the precision, recall, and F1 scores of the crater detection results achieve 0.86, 0.84, and 0.83, respectively, which shows competitive performance compared with the DeepMoon model. Additionally, the explainability of crater detection results is presented by the attention maps visualisation. There are still several aspects left for future improvement. To achieve better results, convolutional layers of the encoder-decoder framework could be changed with more complex modules, such as the Inception module, Residual module, and Dense module. To explore the possibility of onboard application for crater detection, details such as onboard image transformation and algorithm implementation will be researched further. Other datasets will also be used or generated to improve the performance of the proposed framework, such as training with PANGU[2] generated dataset and physical simulation dataset.

Appendix

This work is supported by the ESA, with the Contract No. 4000133292/20/NL/CRS/kk.

References

[1] Hao Wang, Jie Jiang, and Guangjun Zhang. CraterIDNet: An end-to-end fully convolutional neural network for crater detection and identification in remotely sensed planetary images. *Remote sensing*, 10(7):1067, 2018. DOI: 10.3390/rs10071067.

[2] David Vaniman, Robert Reedy, Grant Heiken, Gary Olhoeft, and Wendell Mendell. The lunar environment. *The lunar Sourcebook*, pages 27–60, 1991.

[3] J. D. O'Keefe and T. J. Ahrens. Complex craters: Relationship of stratigraphy and rings to impact conditions. *Journal of Geophysical Research Planets*, 104:27091–27104, 1999. DOI: 10.1029/1998je000596.

[4] Jon Earl, AF Chicarro, Christian Koeberl, P Giorgio Marchetti, and Martin Milnes. Automatic recognition of crater-like structures in terrestrial and planetary images. In *36th Annual Lunar and Planetary Science Conference*, page 1319, 2005.

[5] Miriam Maria Pedrosa, Samara Calçado de Azevedo, Erivaldo Antonio da Silva, and Mauricio Araujo Dias. Improved automatic impact crater detection on mars based on morphological image processing and template matching. *Geomatics, Natural Hazards and Risk*, 8(2):1306–1319, 2017.

[2] https://pangu.software/

[6] Peter H Cadogan. Automated precision counting of very small craters at lunar landing sites. *Icarus*, 348:113822, 2020.

[7] Ebrahim Emami, Touqeer Ahmad, George Bebis, Ara Nefian, and Terry Fong. Crater detection using unsupervised algorithms and convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 57(8):5373–5383, 2019. DOI: 10.1109/TGRS.2019.2899122.

[8] Ebrahim Emami, George Bebis, Ara Nefian, and Terry Fong. Automatic crater detection using convex grouping and convolutional neural networks. In *International symposium on visual computing*, pages 213–224, 2015.

[9] Joseph Paul Cohen, Henry Z Lo, Tingting Lu, and Wei Ding. Crater detection via convolutional neural networks. *arXiv preprint arXiv:1601.00978*, 2016.

[10] Leon F Palafox, Christopher W Hamilton, Stephen P Scheidt, and Alexander M Alvarez. Automated detection of geological landforms on Mars using convolutional neural networks. *Computers & Geosciences*, 101:48–56, 2017. DOI: 10.1016/j.cageo.2016.12.015.

[11] Michael R Klear. PyCDA: An open-source library for automated crater detection. *Proceedings of the 9th Planetary Crater Consort, Boulder, CO*, 2018.

[12] Ari Silburt, Mohamad Ali-Dib, Chenchong Zhu, Alan Jackson, Diana Valencia, Yevgeni Kissin, Daniel Tamayo, and Kristen Menou. Lunar crater identification via deep learning. *Icarus*, 317:27–38, 2019. DOI: 10.1016/j.icarus.2018.06.022.

[13] Lena Downes, Ted J. Steiner, and Jonathan P. How. Deep learning crater detection for lunar terrain relative navigation. In *AIAA Scitech 2020 Forum*. American Institute of Aeronautics and Astronautics, January 2020. DOI: 10.2514/6.2020-1838.

[14] Christopher Lee and James Hogan. Automated crater detection with human level performance. *Computers & Geosciences*, 147:104645, 2021.

[15] Stuart J. Robbins and Brian M. Hynek. A new global database of mars impact craters $\geq$1 km: 2. global crater properties and regional variations of the simple-to-complex transition diameter. *Journal of Geophysical Research: Planets*, 117(E6), June 2012. DOI: 10.1029/2011je003967.

[16] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017. DOI: 10.1109/tpami.2016.2577031.

[17] RZ Povilaitis, MS Robinson, CH Van der Bogert, Harald Hiesinger, HM Meyer, and LR Ostrach. Crater density differences: Exploring regional resurfacing, secondary crater populations, and crater saturation equilibrium on the moon. *Planetary and Space Science*, 162:41–51, 2018. DOI: 10.1016/j.pss.2017.05.006.

[18] Hoonhee Lee, Han-Lim Choi, Dawoon Jung, and Sujin Choi. Deep neural network-based landmark selection method for optical navigation on lunar highlands. *IEEE Access*, 8:99010–99023, 2020. DOI: 10.1109/ACCESS.2020.2996403.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, pages 234–241. Springer International Publishing, 2015.

[20] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[21] James W Head, Caleb I Fassett, Seth J Kadish, David E Smith, Maria T Zuber, Gregory A Neumann, and Erwan Mazarico. Global distribution of large lunar craters: Implications for resurfacing and impactor populations. *science*, 329(5998):1504–1507, 2010. DOI: 10.1126/science.1195050.

[22] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[23] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, volume 37, pages 2048–2057. PMLR, 2015.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[25] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 2017–2025, 2015.

[26] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. DOI: 10.1109/cvpr.2018.00745.

[27] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017. DOI: 10.1109/CVPR.2017.683.

[28] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. DOI: 10.1007/978-3-030-01234-2_1.

[29] Ari Silburt, Chenchong Zhu, Mohamad Ali-Dib, Kristen Menou, and Alan Jackson. Deepmoon: Convolutional neural network trainer to identify moon craters. *Astrophysics Source Code Library*, page 1805.029, 2018.