Segmentation Convolutional Neural Networks for Automatic Crater Detection on Mars

Youssef Khemiri
IEEE ISSATso SB
IEEE Tunisia Section
Nabeul, Tunisia
youssefkhemiri@ieee.org

Wassim Sghaier
IEEE ISSATso SB
IEEE Tunisia Section
Mahdia, Tunisia
wassimsghaierr@ieee.org

Taha Abidi
IEEE ISSATso SB
IEEE Tunisia Section
Sousse, Tunisia
tahaabidi@ieee.org

Mohamed Amine Chabbeh
IEEE ISSATso SB
IEEE Tunisia Section
Tunis, Tunisia
medaminechabbeh@ieee.org

***Abstract*—Research Reproducibility is a one of the main ways to confirm the validity of a new discovery. This paper task is reproducing the result of the scientific paper: Segmentation Convolutional Neural Networks for Automatic Crater Detection on Mars published in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, VOL. 12, NO. 8, AUGUST 2019. This research aims to automate crater counting using Unet architecture for image segmentation, The algorithm is trained using annotations of 2- to 32-km-radius Martian craters and THEMIS Daytime IR images.**

Keywords—feature extraction, image segmentation, geology, planets.

I. Introduction

The use of Machine Learning techniques in space science and engineering has grown in popularity and impact over the past several years, A research done by Danielle M. DeLatte, Member, IEEE, Sarah T. Crites, Nicholas Guttenberg, Elizabeth J. Tasker, and Takehisa Yairi, Member, IEEE, targeted automating crater counting on Mars. Planetary geologists use crater counting to age-date regions of planetary bodies. The surface age is determined by counting the number of craters of various sizes in a region and comparing those counts to expected accumulation from a known production function based on an expected meteorite impact rate. The crater counting task is done by citizen scientists, graduate students, and experts who label—by hand—the characteristics of hundreds of thousands of craters. The results enumerate how incorporating machine learning into the crater counting process is beneficial to planetary geologists. A segmentation network using convolutional neural networks is successfully implemented to find 65%–76% of craters in common with a human annotated dataset. Here comes our goal to reproduce the research results, with a potential for optimization. This research provided a data set containing images from mars and their annotated target images containing craters with size between 2 and 32 km radius, we'll be using a technique called segmentation, where an image map is created of all the detected objects (in this case, craters). The image map marks each pixel with a value related to whether the pixel in the original image belongs to the object category of interest

II. Methods

A. Image PreProcessing

The dataset of Martian craters we will be using is that of Robbins & Hynek (RH2012), The THEMIS image tiles are each 30° per side (7680 × 7680 px) with a resolution of 256 meters per pixel (MPP) and have excellent visual match with the RH2012 annotations without adjusting the projection. One challenge in using the THEMIS images is the absence of data in some areas, seen as the black streaks in Fig. 1, these missing pixels might interfere and have a bad impact on the training.

The original research replaced the black pixels with the average value of the tile, it was a successful approach but wasn't effective when the missing pixels crossed some craters, the filling was not accurate see Fig. 2.

Missing-pixel-filler[1] by SpaceML is a python package that targets removing missing pixels in order to allow CNNs to focus on the target. It proposes an augmentation technique that considerably removes the existence of swath gaps(missing pixels), This missing data filling algorithm is open source can be changed according to user preferences, thus we adjusted the code so it fits our Martian craters dataset, a method that uses "dynamic" system to fill missing pixels, with nearest pixels having higher probability of selection Fig. 2.

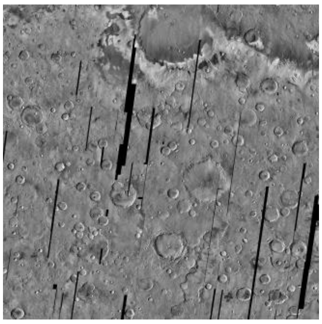

Fig. 1 Original Image showcasing missing pixels

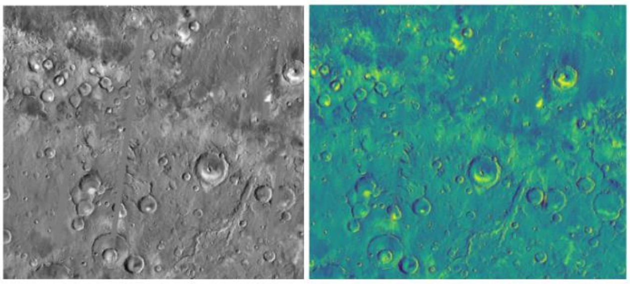
(a) (b)

Fig. 2 (a) is the old filling method, (b) is the missing pixel filler by SpaceML method

B. Data

This subset of the THEMIS dataset contains 24 “tiles” of 30° by 30°, shown in Fig. 2. Each is 7680 × 7680 px. These initial tests use a subset of the available (see Fig. 3) annotations: craters size 2–32 km in radius (see Fig. 4). Creating 15 × 15 sub-images size 512x512 from each image as an input to the network.

A python library called patchify made in 2021 allows us to divide the original image with a new feature, choosing the step, allowing us to avoid the problems of dividing a crater between 2 sub-images Fig. 5.

C. Training

For training, the 24 tiles (see Fig. 5) are split randomly into one of three groups: training, validation, or test tiles. The training tiles are 19 out 24 tiles and corresponding annotations (target) are used to learn a model. Validation data (20% of the training data) and test data (5 out of 25 tiles) are used to compare architectures on data unseen in training. The data tiles and generated target images are split into 512 × 512 px sub-images before being fed through the network as mentioned in Fig. 5

D. Segmentation Network Design

Segmentation methods are similar, from a data processing perspective to what humans would do: take a whole image, identify the craters, and determine the number and locations of those craters inside the region of interest. Only that on the smallest scale (one pixel) and for a one-object detection, “segmentation” amounts to a binary classification of each pixel, giving a measure of whether the pixel is part of an example of the objects being identified.

We’ll be using U-net architecture with some modifications, just like the original research, first using

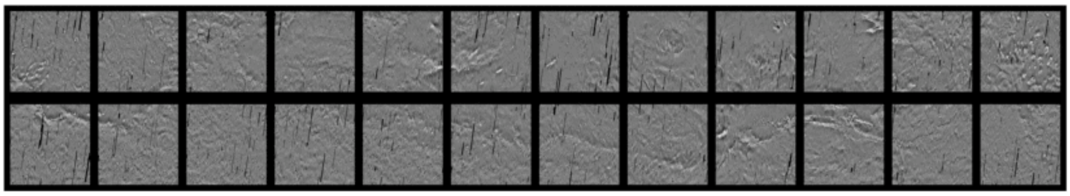

Fig. 3 Mars Craters Dataset, 24 tiles each is 7680 x 7680 px

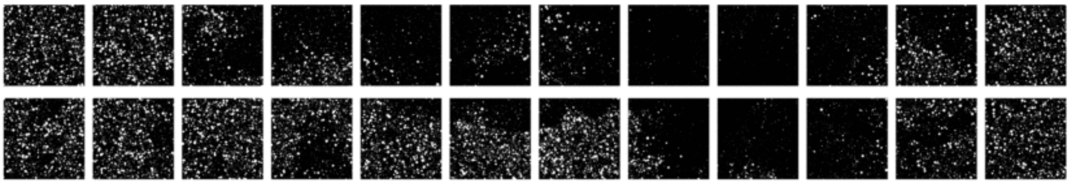

Fig. 4 Target Dataset, craters size 2-32 km in radius

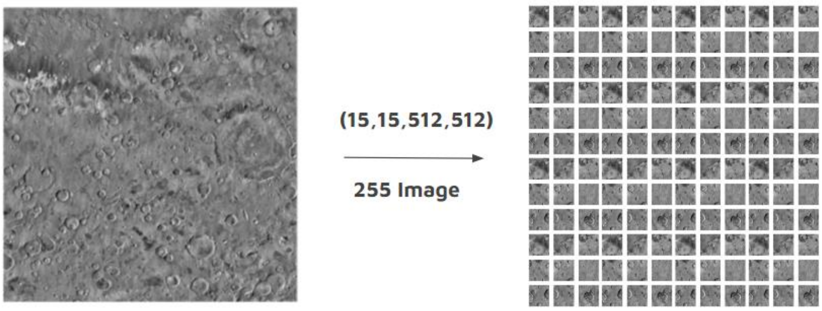

Fig. 5 sub-images 15 x 15 from the original tile [2]

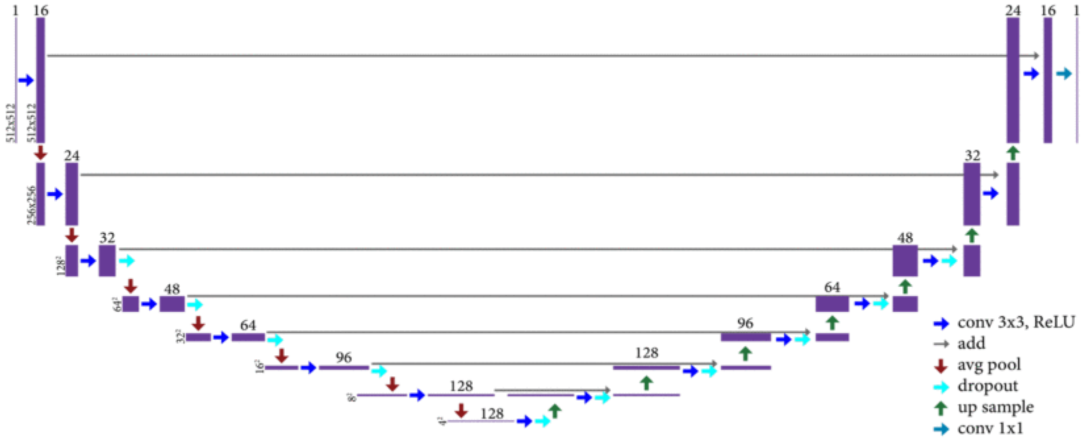

Fig. 6

dropout instead of copy and crop, secondly using average pooling instead of max pooling, Third, the pooling layer follows each convolution layer instead of having two convolution layers in a row. These changes lead to a deeper network and a dramatic reduction in training time (twice as fast as regular U-net).

We used the default network (see Fig. 7) has the following components: Rectified Linear Unit (ReLU) activation function [43], convolutional kernel size of 3 × 3, and filter values of the layers (starting at the upper left of Fig. 5): [16, 24, 32, 48, 64, 96, 128, 128, 128, 96, 64, 48, 32, 24, 16]. Convolution [46], dropout [45], average pooling [46], and upsampling [46] layers are used.
The network is coded in Python using Keras.
Data (grayscale 512 × 512 px image) enters the Crater U-Net on the upper left. As it "goes down" the left half of the network, downsampling occurs as features at larger scales are computed. After seven rounds of downsampling, upsampling through seven layers ("going up") produces an output array of the same size as the original input. Fig. 8. And kept the default kernel size and number of filters, The reported loss uses the binary cross-entropy function, and the accuracy uses the default Keras definition

E. *Evaluating Crater Detections*

Many pixel-based measurements are available for evaluating the results of a trained neural network. However, for the application of creating lists of craters in a region, these loss and accuracy measures are not the most suitable for evaluating which model will give the best scientific result. Scientists are less concerned by a pixel out of place and more concerned with whether the crater was detected. A metric derived from the actual crater counts is preferred. As a result, the F1 score, the harmonic mean of precision and recall, is used on a per identified crater basis.

III. Results

The model was trained on two types of target data, Filled and thin targets Fig. 7 Fig. 8

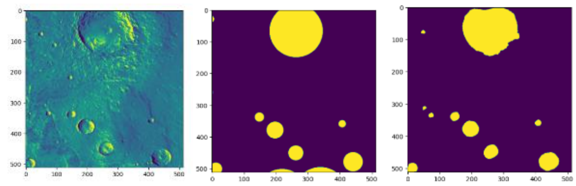

Fig. 7 Data, Filled circle target, Prediction

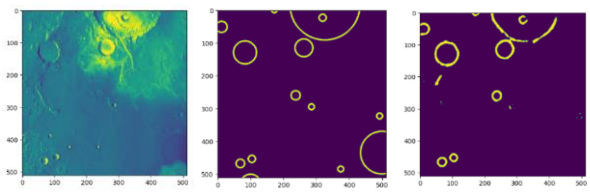

Fig. 8 Data, Thin target, Prediction

Acknowledgment *(Heading 5)*

Special Thanks to Professor Ahmed Ghazi Blaiech for his advice and guidance.

References

[1] Missing pixel filler by SpaceML, Work by Sarah Chen, Esther Cao, Anirudh Koul, Siddha Ganju, Satyarth Praveen, Meher Anand Kasam

[2] The right way to segment large images by applying a trained U-Net model on smaller patches by digital screeni

[3] Retinal-eye-vessel-segmentation by Maulana Akbar Dwijaya