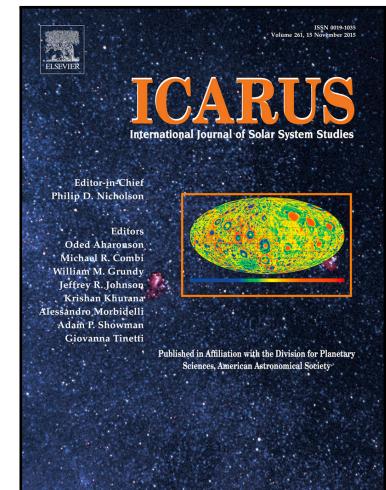


Accepted Manuscript



Lunar Crater Identification via Deep Learning

Ari Silburt, Mohamad Ali-Dib, Chenchong Zhu, Alan Jackson,
Diana Valencia, Yevgeni Kissin, Daniel Tamayo, Kristen Menou

PII: S0019-1035(18)30138-6
DOI: [10.1016/j.icarus.2018.06.022](https://doi.org/10.1016/j.icarus.2018.06.022)
Reference: YICAR 12943

To appear in: *Icarus*

Received date: 6 March 2018
Revised date: 29 May 2018
Accepted date: 22 June 2018

Please cite this article as: Ari Silburt, Mohamad Ali-Dib, Chenchong Zhu, Alan Jackson, Diana Valencia, Yevgeni Kissin, Daniel Tamayo, Kristen Menou, Lunar Crater Identification via Deep Learning, *Icarus* (2018), doi: [10.1016/j.icarus.2018.06.022](https://doi.org/10.1016/j.icarus.2018.06.022)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

¹ **Highlights**

- ² • Trained a Deep Learning model to automatically identify Lunar craters.
- ³ • Reproduced a human-classified crater distribution on a Lunar test
- ⁴ patch.
- ⁵ • Used Moon-trained model to successfully recognize craters on Mer-
- ⁶ cury.

Lunar Crater Identification via Deep Learning

Ari Silburt^{a,b,c,f}, Mohamad Ali-Dib^{a,d,f}, Chenchong Zhu^{b,d}, Alan Jackson^{a,b,e}, Diana Valencia^{a,b}, Yevgeni Kissin^b, Daniel Tamayo^{a,d}, Kristen Menou^{a,b}

^a*Centre for Planetary Sciences, Department of Physical & Environmental Sciences, University of Toronto Scarborough, Toronto, Ontario M1C 1A4, Canada*

^b*Department of Astronomy & Astrophysics, University of Toronto, Toronto, Ontario M5S 3H4, Canada*

^c*Department of Astronomy & Astrophysics, Penn State University, Eberly College of Science, State College, PA 16801, USA*

^d*Canadian Institute for Theoretical Astrophysics, 60 St. George St, University of Toronto, Toronto, ON M5S 3H8, Canada*

^e*School of Earth and Space Exploration, Arizona State University, 781 E Terrace Mall, Tempe, AZ 85287-6004, USA*

^f*These authors contributed equally to this work.*

Keywords: Moon; Crater Detection; Automation; Deep Learning

Corresponding authors:

- Ari Silburt, Department of Astronomy & Astrophysics, Penn State University, Eberly College of Science, State College, PA 16801, USA. e-mail: ajs725@psu.edu .

- Mohamad Ali-Dib, Centre for Planetary Sciences, Department of Physical & Environmental Sciences, University of Toronto Scarborough, Toronto, Ontario M1C 1A4, Canada. e-mail: m.alidib@utoronto.ca .

²⁹ **Abstract**

³⁰ Crater counting on the Moon and other bodies is crucial to constrain
³¹ the dynamical history of the Solar System. This has traditionally been done
³² by visual inspection of images, thus limiting the scope, efficiency, and/or
³³ accuracy of retrieval. In this paper we demonstrate the viability of using
³⁴ convolutional neural networks (CNNs) to determine the positions and sizes
³⁵ of craters from Lunar digital elevation maps (DEMs). We recover 92% of
³⁶ craters from the human-generated test set and almost double the total num-
³⁷ ber of crater detections. Of these new craters, 15% are smaller in diameter
³⁸ than the minimum crater size in the ground-truth dataset. Our median
³⁹ fractional longitude, latitude and radius errors are 11% or less, represent-
⁴⁰ ing good agreement with the human-generated datasets. From a manual
⁴¹ inspection of 361 new craters we estimate the false positive rate of new
⁴² craters to be 11%. Moreover, our Moon-trained CNN performs well when
⁴³ tested on DEM images of Mercury, detecting a large fraction of craters in
⁴⁴ each map. Our results suggest that deep learning will be a useful tool for
⁴⁵ rapidly and automatically extracting craters on various Solar System bod-
⁴⁶ ies. We make our code and data publicly available at <https://github.com/>
⁴⁷ [silburt/DeepMoon.git](https://github.com/silburt/DeepMoon.git) and <https://doi.org/10.5281/zenodo.1133969>.

⁴⁸ **1. Introduction**

⁴⁹ Craters formed by small impactors constitute an important surface prop-
⁵⁰ erty for many bodies in the Solar System. On airless bodies like the Moon,

51 Mercury, Ceres, and Vesta, weather based erosion, tectonics and volcanic
52 activity have been largely non-existent resulting in the accumulation of im-
53 pact craters over time. However, other eroding factors such as micromete-
54 orite bombardment can affect smaller craters.

55 Crater densities permit the geological history of a body to be examined,
56 and the relative chronology of a region to be assessed remotely. In addition,
57 when in-situ samples are recovered from a body, absolute chrohologies can
58 be determined too. Inferred temporal variation in cratering rates have been
59 used to make inferences about the dynamical history of the Solar System,
60 including the (debated) possibility of a Late Heavy Bombardment, (e.g.,
61 Hartmann 1970; Ryder 2002; Gomes et al. 2005; Chapman et al. 2007;
62 Bottke and Norman 2017). Crater records and chronology are thus cen-
63 tral to any formation theory about the Solar System. In addition, the size
64 distribution of craters directly probes the dynamics and size distribution of
65 the impactor population (Strom et al., 2005). For example from the size
66 distribution of craters on the Lunar highlands, Minton et al. (2015) argued
67 that the impactor population contained comparatively fewer large bodies
68 than the asteroid belt does today.

69 Traditionally, crater detection has been done manually via visual inspec-
70 tion of images. However this approach is not practical for the vast num-
71 bers of kilometre and sub-kilometre sized craters on the Moon (and other
72 Solar System bodies), resulting in human-generated databases that are ei-
73 ther spatially comprehensive but restricted to the largest craters, or size

74 comprehensive but limited to a very specific geographic region (Stepinski
75 et al., 2012; Bandeira et al., 2012). In addition, manual crater counting by
76 experts can yield disagreements as high as 40% (Greeley and Gault, 1970;
77 Kirchoff et al., 2011; Robbins et al., 2014).

78 As a result, scientists have developed crater detection algorithms (CDAs)
79 to automate the process of classifying craters. Such CDAs include edge de-
80 tection (Emami et al., 2015), Hough transforms (Salamunićcar and Lončarić,
81 2010), support-vector machines (Wetzler et al., 2005), decision trees (Stepin-
82 ski et al., 2012) and neural networks (Wetzler et al., 2005; Cohen et al.,
83 2016; Palafox et al., 2017). Multi-step approaches have also been tried.
84 For example, Di et al. (2014) used a boosting algorithm to box the crater-
85 containing region, then a Hough transform to delimit the edges of the
86 crater. Boukercha et al. (2014) used a similar approach where an initial
87 detection algorithm provided crater candidates which were subsequently
88 classified as true or false positives by a support-vector machine or polyno-
89 mial classifier.

90 These CDAs tend to perform well on the datasets upon which they were
91 trained, but not to generalize well on unseen patches or other bodies (see
92 Stepinski et al. (2012) for a review and Chung et al. (2014) for a compar-
93 ison between classical and machine learning based techniques.). The diffi-
94 culty in designing robust CDAs stems from the complex nature of craters,
95 having large variations in shape and illumination, orders of magnitude size
96 differences, overlap and degradation. An algorithm capable of universally

⁹⁷ identifying craters on Solar System bodies would be invaluable to the com-
⁹⁸ munity.

⁹⁹ In this work we train a deep learning architecture known as a convo-
¹⁰⁰ lutional neural network (CNN) to perform crater identification¹ on Lunar
¹⁰¹ digital elevation map (DEM) images, and transfer-learn our Moon-trained
¹⁰² CNN to identify craters on Mercury. There are numerous reasons for using
¹⁰³ CNNs to detect craters. First, CNNs have demonstrated impressive perfor-
¹⁰⁴ mance on a variety of computer vision problems and other datasets where
¹⁰⁵ features are correlated (e.g. Long et al., 2015), demonstrating their ver-
¹⁰⁶ satility. This includes, in addition to images, sounds and signals. Second,
¹⁰⁷ CNNs engineer their own representation features, alleviating the need for
¹⁰⁸ a human to develop sophisticated pre-processing algorithms and custom
¹⁰⁹ input features. Finally, CNNs have been able to successfully classify objects
¹¹⁰ that appear at multiple scales in a single image (Zhang et al., 2016; Zeng
¹¹¹ et al., 2017), a property very relevant to crater counting.

¹¹² 2. Methods

¹¹³ The code to generate the data set (Section 2.1), train our model (Sec-
¹¹⁴ tion 2.7), and extract the resulting crater distribution (Section 2.4 and Sec-
¹¹⁵ tion 2.5) is available at <https://github.com/silburt/DeepMoon.git>. The

¹By crater identification, we mean a) output the pixel locations of crater rims from DEM images via a CNN segmentation task, and b) extract the crater coordinates from these CNN outputs using a custom pipeline, explained in the Methods section.

116 data used to train, validate and test our model, the global DEM used to
117 make our input images, our best model and final test set crater distribu-
118 tion can be found at <https://doi.org/10.5281/zenodo.1133969>. We use
119 Keras (Chollet, 2015) version 1.2.2 with Tensorflow (Abadi et al., 2016)
120 version 0.10 to build and train our model, but our code is also compatible
121 with Keras 2.0.2 and Tensorflow 1.0.

122 *2.1. Data Preparation*

123 Our input data was generated by randomly cropping digital elevation
124 map (DEM) images from the Lunar Reconnaissance Orbiter (LRO) and Kaguya
125 merged digital elevation model, which spans ± 60 degrees in latitude (and
126 the full range in longitude) and has a resolution of 512 pixels/degree, or
127 59 m/pixel (Barker et al. 2016; available at LOLA Team and Kaguya Team
128 2015). This global grayscale map is a Plate Carree projection with a resolu-
129 tion of 184320×61440 pixels and a bit depth of 16 bits/pixel; we downsam-
130 pled it to 92160×30720 pixels and 8 bits/pixel. We use an elevation map,
131 rather than an optical one, because a crater's appearance in an elevation
132 map is not affected by the direction of incident sunlight. This reduces vari-
133 ation in appearance between craters, making it easier to train a CNN to
134 identify them.

135 Each input DEM image is generated by a) randomly cropping a square
136 area of the global map, b) downsampling the cropped image to $256 \times$
137 256 pixels, c) transforming the image to an orthographic projection using

¹³⁸ the Cartopy Python package (UK Met. Office, 2015) in order to minimize
¹³⁹ image distortion, and d) linearly rescaling image intensity to boost contrast.

¹⁴⁰ The position of the cropped region in a) is randomly selected with a uniform
¹⁴¹ distribution, and its length is randomly selected from a log-uniform
¹⁴² distribution with minimum and maximum bounds of 500 and 6500 pixels
¹⁴³ (59 km and 770 km), respectively. The transformation in step c) for our in-
¹⁴⁴ put data often produces non-square images that are padded with zeros;
¹⁴⁵ these are the black bars on the sides of the Moon DEM in Figure 1.²

¹⁴⁶ For each input image, we generate a corresponding ground-truth “out-
¹⁴⁷ put target” that is also 256×256 pixels. Craters are encoded in the tar-
¹⁴⁸ gets as rings with thicknesses of 1 pixel, and with radii and centers derived
¹⁴⁹ from craters’ physical locations and diameters in the catalog, described be-
¹⁵⁰ low. All target pixel values are binary, including at ring intersections. Any
¹⁵¹ craters with diameter $D_{\text{pix}} < 1$ pix are excluded. We experimented with
¹⁵² other target formats, including density maps, and binary and non-binary
¹⁵³ filled circles. We found, however, that out of all of these formats binary
¹⁵⁴ ring targets were best reproduced by the CNN, particularly for situations
¹⁵⁵ with many overlapping craters. While the rings highlight crater edges, the
¹⁵⁶ CNN still uses information about the interior of craters to generate the ring
¹⁵⁷ masks. We tested this on a limited sample following the method of Zeiler

²To check that this padding does not affect our results, we experimented with training our CNN on DEM images with all padding cropped out, and found our performance metrics (Table 3.1) differed by only a few percent. We thus conclude that it has a negligible effect.

¹⁵⁸ and Fergus (2013); Shallue and Vanderburg (2018) by perturbing the pixels of input DEMs, and found that we could prevent a crater detection by ¹⁵⁹ only increasing the pixels within its interior, without modifying its rim. This ¹⁶⁰ demonstrates that the interiors of craters are being used by our network. ¹⁶¹

¹⁶² The data used to construct the targets was obtained by merging two ¹⁶³ human-generated crater catalogs. For 5 – 20 km craters we used the global ¹⁶⁴ crater dataset assembled by Povilaitis et al. (2017) using the LRO Wide ¹⁶⁵ Angle Camera (WAC) Global Lunar DEM at 100 m/pixel (303 pixels/degree) ¹⁶⁶ resolution (GLD100; Scholten et al. 2012), and for > 20 km craters we used ¹⁶⁷ the global crater dataset assembled by Head et al. (2010) using the LOLA ¹⁶⁸ DEM with a resolution of 64 pixels/degree (472 m/pixel). Merging these two ¹⁶⁹ datasets in this way was intended by Povilaitis et al. (2017), who explicitly ¹⁷⁰ designed their dataset as a continuation of that of Head et al. (2010) to ¹⁷¹ smaller sizes. The methods used to assemble these datasets are described ¹⁷² in Head et al. (2010) and Povilaitis et al. (2017) and are typical for human ¹⁷³ generated datasets.³

¹⁷⁴ We split our input image/output target pairs into three separate datasets ¹⁷⁵ to be used for training, validating and testing our CNN (see Section 2.7 for ¹⁷⁶ their uses). The three datasets are sampled from equal sized and mutu- ¹⁷⁷ ally exclusive portions of the Moon, spanning the full range in latitude and

³During initial testing we also used the LU78287GT Lunar Crater Catalog (Sala-munićar et al., 2014), which was generated by a Hough Transform-based CDA. We transitioned to solely using human-generated catalogs to prevent the CNN from inadvertently learning the biases of another CDA.

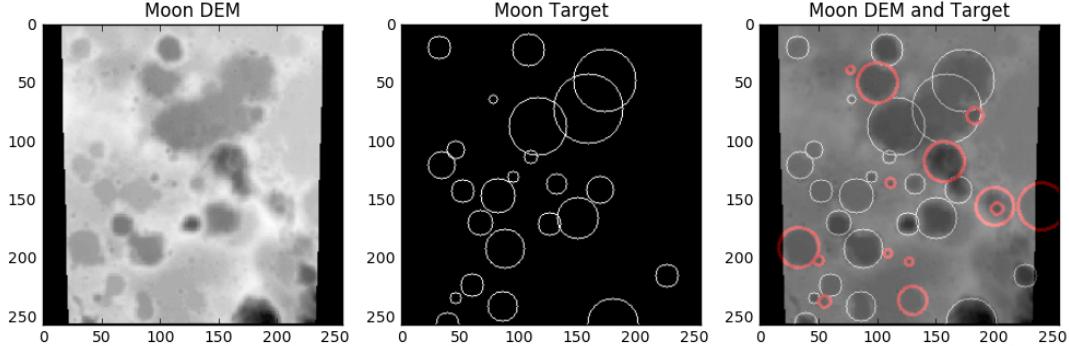


Figure 1: Sample Moon DEM image (left) and target (middle) from our dataset, with the two overlaid (right). Red circles (right panel) show [features that appear to be craters but are](#) absent from the Head et al. (2010) and Povilaitis et al. (2017) datasets, representing apparently missed classifications.

178 -180° to -60° , -60° to 60° and 60° to 180° in longitude for the training,
 179 validation and test sets, respectively. Each dataset contains 30000 DEM im-
 180 ages, and the median number of craters per DEM image is 21. Because we
 181 use a log-uniform sampling of crop lengths, all scales are equally repre-
 182 sented in area, but the datasets contain far more DEM images with small
 183 crop lengths. An example input DEM image and target pair is shown in the
 184 left and middle panels of Figure 1.

185 Our human-generated crater dataset is incomplete and contains many
 186 apparent missed identifications. Fassett et al. (2012) estimated an incom-
 187 pleteness of 12% for the Head et al. (2010) dataset. The incompleteness
 188 of the Povilaitis et al. (2017) dataset is unknown at this time, but appears

189 to be comparable or higher. For the sample DEM image/target pair shown
 190 in Figure 1, we highlight features that appear to be craters but are miss-
 191 ing from the Head et al. (2010) and Povilaitis et al. (2017) datasets as red
 192 circles in the right panel. It is unclear at this time why these craters were
 193 missed. In addition, the right panel of Figure 1 shows how our binary ring
 194 targets do not match the rims of non-circular craters. Together, these hin-
 195 der the training of our CNN since genuine crater rims will be present in our
 196 dataset with no corresponding target rings, potentially confusing the CNN
 197 (see Section 4 for a discussion).

198 2.2. Convolutional Neural Networks (CNNs)

199 In this section we provide a brief, heuristic background to convolu-
 200 tional neural networks (CNNs) before introducing our network architec-
 201 ture. More in depth descriptions of the theory and mathematics of CNNs
 202 can be found in references such as Goodfellow et al. (2016) Chapter 9.

203 Machine learning algorithms in general can be thought of as universal
 204 function approximators, and neural networks (NNs) are a type of machine
 205 learning algorithm that uses “neurons” (loosely modelled after the human
 206 brain) to make such approximations. A neuron can be represented mathe-
 207 matically by:

$$y = f \left(\sum_j w_j x_j + b \right) \quad (1)$$

208 where x_j are a set of input values that are linearly combined with a set of

209 weights w_j , and added to a bias offset b . This linear combination is **then**
 210 **fed through a (typically non-linear)** “activation function” f , which returns
 211 output y . Depending on the choice of $f(z)$, where $z = \sum_j w_j x_j + b$, a neuron
 212 is able to represent a number of simple functions, eg. $f(z) = z$ for a line, or
 213 $f(z) = 1 / (1 + \exp(-z))$ for a sigmoid function. Varying the weights w_j and
 214 bias b to best approximate a set of known y values given a corresponding
 215 set of x_j values is thus equivalent to linear regression when $f(z)$ is linear,
 216 and logistic regression when $f(z)$ is sigmoidal.

217 A neural network contains sets, or layers, of neurons $\{y_i\}$, where y_i rep-
 218 resents the output of the i -th neuron in the layer. Neurons within each
 219 layer are independent of one another, but different layers are stacked on
 220 top of one another, with the output of one layer serving as the input to the
 221 next.⁴ Input data is fed into the first layer, while the network’s **“prediction”**,
 222 **or predicted output target**, comes from the last. A network is “trained” by
 223 tuning the weights of all neurons in all layers so that the network best ap-
 224 proximates a set of **known, or ground-truth**, targets given corresponding
 225 input. This tuning is typically done via backpropagation (Rumelhart et al.,
 226 1986), and goodness of fit is defined through a loss function, e.g. mean
 227 squared error between the known targets and network predictions. Much
 228 like adding terms to a Taylor series, increasing the number of layers and/or
 229 the number of neurons per layer allows the network to approximate func-

⁴This is true of standard “feed-forward” neural networks. Other types of networks exist, e.g. recurrent neural networks, but are beyond the scope of this paper.

230 tions of increasing complexity. However, this added complexity comes at
 231 the cost of more tunable parameters and the potential for errors to be am-
 232 plified from one layer to the next; both make network optimization more
 233 difficult.

234 In computer vision problems, the input is typically pixel intensities from
 235 images, while the target is typically either a single class label for the image
 236 or another array of pixel intensities. In the latter case (which used in this
 237 work), the approximated function is a mapping from one type of image to
 238 another. Since the input and output are two-dimensional, we may represent
 239 the neuron in the i-th row and j-th column of one layer as

$$y_{ij} = f \left(\sum_k \sum_l w_{ij,kl} x_{kl} + b_{ij} \right) \quad (2)$$

240 where, for the NN's first layer, x_{kl} represents input image pixel intensities.
 241 Classical NNs, however, place no restrictions on $w_{ij,kl}$, and allow weights of
 242 different neurons in a layer to vary independently of one another. For im-
 243 ages, this means any number of pixels in any orientation can be connected,
 244 and there is no guarantee the spatial information of the image is preserved
 245 from one layer to the next.

246 CNNs primarily differ from traditional NNs in that their neurons are
 247 only locally connected to the previous input volume (LeCun et al., 1989),
 248 and layers are structured to perform a discrete convolution between their
 249 inputs and a kernel, or "filter", which is represented by the weights. In the

250 context of Equation 2, this means $w_{ij,kl}$ is zero other than a few adjacent
 251 values of k, l , and the weights for one neuron are just index-shifted versions
 252 of the weights for another. Convolutional layers hence embed the spatial
 253 continuity of images directly into the architecture of the network. Also,
 254 since there are only a few non-zero weights that are shared between all the
 255 neurons, only a small number of weights need to be stored in memory and
 256 adjusted during training, simplifying network optimization. The weight-
 257 sharing also exploits the property that weights useful to one section of an
 258 image might also be useful to another.

259 The main components of CNNs relevant to our work are convolutional
 260 layers, pooling layers, and merge layers. Convolutional layers, the primary
 261 component of a CNN, contain neurons that convolve an input with a filter
 262 in the manner described above. **The output of this convolution (i.e. y_{ij} in
 263 Equation 2) is called a “feature map”.** In practice, a single convolutional
 264 layer can contain a large number of filters, each acting on the layer’s input
 265 **to produce its own feature map.** We make use of these “filter banks” in our
 266 CNN architecture. Pooling layers perform downsampling operations along
 267 the spatial dimensions, reducing the dimensionality of the image and **thus
 268 the number of learnable weights needed for subsequent layers.** Merge lay-
 269 ers combine **feature maps from** convolutional layers of compatible spatial
 270 dimensions, facilitating complex connections within the CNN. Neither pool-
 271 ing nor merge layers have trainable parameters, and are not represented
 272 by Equation 2.

273 In recent years CNNs have demonstrated impressive performance on
 274 image-related tasks, including classifying individual pixels (aka “segmen-
 275 tation” in computer vision terminology) (Long et al., 2015). A particularly
 276 successful network for pixel-wise classification (image to image mapping) is
 277 the UNET architecture (Ronneberger et al., 2015), which was originally de-
 278 signed for biomedical segmentation problems. A novel aspect of the UNET
 279 architecture is the use of numerous “skip connections” which merge deep
 280 and shallow layers together, providing both spatial and semantic classifica-
 281 tion information for future convolutions.

282 *2.3. CNN Architecture*

283 In this work we implement a custom version of the UNET architecture
 284 (Ronneberger et al., 2015), shown in Figure 2.⁵ This architecture consists
 285 of a contracting path (left side) and expansive path (right side), joined
 286 through multi-level skip connections (middle). Lunar DEM images are in-
 287 put to the contracting path and predictions are output from a final layer
 288 following the expansive path. Unless otherwise stated, all convolutional
 289 layers have banks of filters that each apply 3x3 padded convolutions fol-
 290 lowed by a rectified linear activation unit (ReLU; e.g. Goodfellow et al.
 291 2016 Chapter 6.1), whose functional form is $f(z) = \max(0, z)$.

292 The contracting and expansive paths each contain 3 convolutional blocks.

⁵In the early stages of this work, we attempted to count the craters of an image, rather than localize them, using a traditional CNN regressor. However, that model’s resulting accuracy was low, motivating our shift to the UNET-based one.

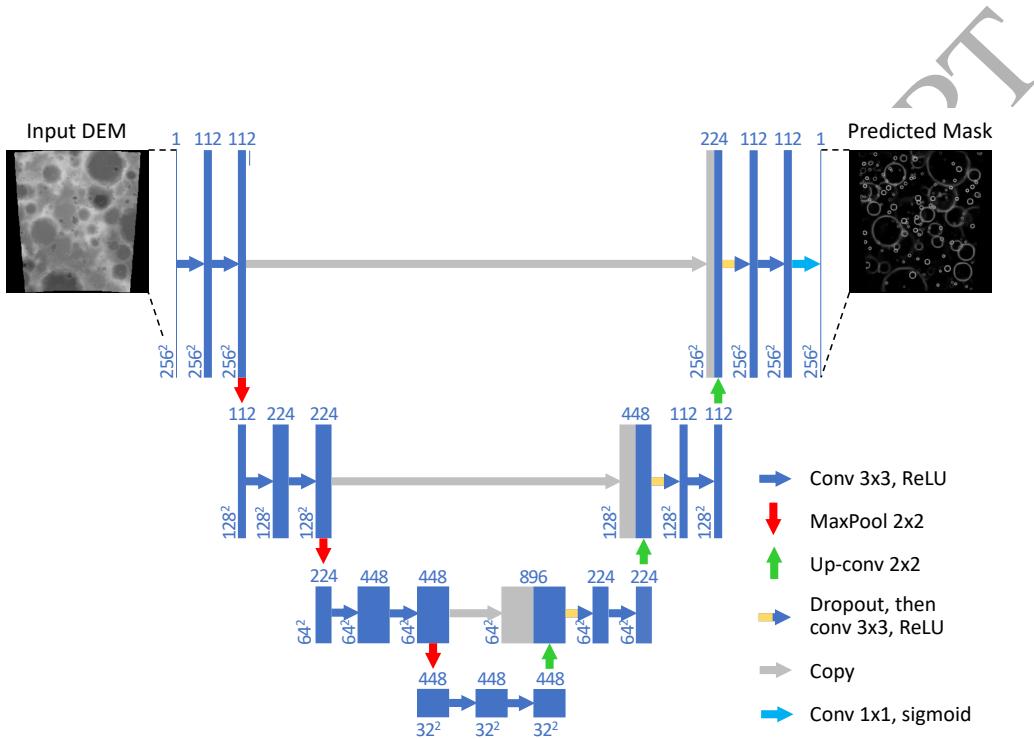


Figure 2: Convolutional neural network (CNN) architecture, based on UNET (Ronneberger et al., 2015). Boxes represent cross-sections of sets of square feature maps. For each set, its maps' dimensions are indicated on its lower left, and its number of maps are indicated above it. Half-grey boxes represent sets for which half of their maps are copied. The leftmost map is a 256×256 grayscale image sampled from the digital elevation map, and the rightmost the CNN's binary ring mask prediction. Arrows represent operations, specified by the legend - notably, blue arrows represent convolutions, while gray ones represent copying (skip connections).

293 A block in the contracting path consists of two convolutional layers fol-
 294 lowed by a max-pooling layer with a 2x2 pool size. A block in the expan-
 295 sive path consists of a 2x2 upsampling layer, a concatenation with the cor-
 296 responding block from the contracting path (i.e. a merge layer), a dropout
 297 layer (Srivastava et al., 2014), and two convolutional layers. The connect-
 298 ing path consists of two convolutional layers. Lastly, the final output layer
 299 is a 1x1 convolutional layer with a sigmoid activation and a single filter to
 300 output pixel-wise class scores. In the contracting path, each convolutional
 301 layer in blocks 1, 2 and 3 contain 112, 224 and 448 filters, respectively,
 302 while in the expansive path blocks 5, 6 and 7 contain 224, 122 and 122,
 303 respectively. Each convolutional layer in the connecting path contains 448
 304 filters. Our CNN differs from the original UNET by the number of **filters**
 305 **in each convolution layer** (which we selected for the model to fit into GPU
 306 memory) and the use of dropout in the expansive path.

307 2.4. Crater Extraction

308 A 256×256 DEM image passed through the CNN will output a 256×256
 309 target with activated pixels corresponding to the locations of the crater
 310 rims. However, the CNN does not explicitly extract crater position and
 311 size from these rims. Instead, this must be done separately using a cus-
 312 tom pipeline that relies heavily on the `match_template` algorithm from
 313 `scikit-image` (Van der Walt et al., 2014)⁶. This algorithm iteratively slides

⁶Although template matching is an expensive technique done in a brute force manner we found it far more accurate than others including the Hough transform (Duda and Hart,

³¹⁴ generated rings through the targets and calculates a match probability at
³¹⁵ each (x, y, r) coordinate where (x, y) is the centroid of the generated ring
³¹⁶ and r is the radius.

³¹⁷ Our custom crater extraction pipeline is as follows. For each CNN-
³¹⁸ predicted target we apply a binary threshold B such that pixel intensities
³¹⁹ greater than B are set to 1 and those otherwise are set to 0. We then
³²⁰ apply Scikit's `match_template` algorithm over a radius range r_{\min} to r_{\max}
³²¹ and classify any (x, y, r) ring with a match probability greater than P_m as a
³²² crater. Two craters i and j that fulfill the following criteria are flagged as
³²³ duplicates if they satisfy both of the following conditions:

$$\begin{aligned} & ((x_i - x_j)^2 + (y_i - y_j)^2) / \min(r_i, r_j)^2 < D_{x,y} \\ & \text{abs}(r_i - r_j) / \min(r_i, r_j) < D_r \end{aligned} \quad (3)$$

³²⁴ where $D_{x,y}$ and D_r are tunable hyperparameters. For duplicates we keep
³²⁵ only the crater with the highest match probability P_m . This process is re-
³²⁶ peated for all CNN-predicted targets.

³²⁷ As is standard practice in machine learning, our hyperparameters are
³²⁸ tuned by measuring the performance of various combinations on the vali-
³²⁹ dation data and picking the optimal set. After training our CNN (see Sec-
³³⁰ tion 2.7), we perform a randomly sampled grid search of size 210 on the

1972) and Canny edge detection (Canny, 1986) with enclosed circle fitting.

³³¹ validation data over the following hyperparameter ranges:

$$B = [0.05, 0.1, 0.15]$$

$$P_m = [0.3, 0.4, 0.5, 0.6, 0.7]$$

$$D_{x,y} = [0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2]$$

$$D_r = [0.2, 0.4, 0.6, 0.8, 1.0, 1.2]$$

³³² We find $B = 0.1$, $P_m = 0.5$, $D_{x,y} = 1.8$ and $D_r = 1.0$ yields the optimal
³³³ F_1 score (see Equation 9) of 0.74. We set $r_{\min} = 5$ to minimize errors (see
³³⁴ Section 2.8) and set $r_{\max} = 40$.

³³⁵ 2.5. Post-Processing

³³⁶ Since our dataset contains DEM images with a log-uniform distribution
³³⁷ of magnifications and uniform distribution of locations, a single crater will
³³⁸ appear on average in 120 ± 30 different DEM images. This increases the
³³⁹ likelihood of detection but also yields many duplicates across targets which
³⁴⁰ must be filtered. Therefore, in a final post-processing step we aggregate
³⁴¹ crater detections across all targets, convert from pixel coordinates to de-
³⁴² grees and kilometers, and filter out duplicates.

³⁴³ Using the known properties of each DEM image, craters are converted
³⁴⁴ from pixel coordinates (x, y, r) to degrees and kilometer coordinates $(\mathcal{L},$
³⁴⁵ $L, R)$:

$$\begin{aligned}
 L - L_0 &= \frac{\Delta L}{\Delta H} (y - y_0) \\
 \mathcal{L} - \mathcal{L}_0 &= \frac{\Delta L}{\cos\left(\frac{\pi L}{180^\circ}\right) \Delta H} (x - x_0) \\
 R &= r \frac{C_{KD} \Delta L}{\Delta H},
 \end{aligned} \tag{4}$$

346 where \mathcal{L} and L are the crater's longitude and latitude centroid, subscript
 347 0 values are those for the center of the DEM image, ΔL and ΔH are the
 348 latitude and pixel extents of the DEM image along its central vertical axis
 349 (where $\mathcal{L} = \mathcal{L}_0$), excluding any padding, and

$$C_{KD} = \frac{180^\circ}{\pi R_{\text{Moon}}} \tag{5}$$

350 is a kilometer-to-degree conversion factor, where R_{Moon} is the radius of the
 351 Moon in km.

352 We then employ a similar filtering strategy as Section 2.4, classifying
 353 craters i and j as duplicates if they satisfy both of the following conditions:

$$\begin{aligned}
 \frac{((\mathcal{L}_i - \mathcal{L}_j)^2 \cos^2\left(\frac{\pi}{180^\circ} \langle L \rangle\right) + (L_i - L_j)^2)}{C_{KD}^2 \min(R_i, R_j)^2} &< D_{\mathcal{L},L} \\
 \frac{\text{abs}(R_i - R_j)}{\min(R_i, R_j)} &< D_R.
 \end{aligned} \tag{6}$$

354 where $\langle L \rangle = \frac{1}{2}(L_i + L_j)$. D_R and $D_{\mathcal{L},L}$ are hyperparameters which, like in
 355 Section 2.4, we tune by performing a grid search on the validation dataset

³⁵⁶ after training our CNN and tuning its hyperparameters, this time sampling
³⁵⁷ every combination from:

$$D_{\mathcal{L},L} = [0.6, 1.0, 1.4, 1.8, 2.2, 2.6, 3.0, 3.4, 3.8]$$

$$D_R = [0.2, 0.6, 1.0, 1.4, 1.8, 2.2, 2.6, 3.0, 3.4, 3.8]$$

³⁵⁸ We find $D_{\mathcal{L},L} = 2.6$ and $D_R = 1.8$ yields the optimal F_1 score (see Equa-
³⁵⁹ tion 9) of 0.67.

³⁶⁰ 2.6. Accuracy Metrics

³⁶¹ To train our network we use the pixel-wise binary cross-entropy, ℓ (Abadi
³⁶² et al., 2016; Chollet, 2015), a standard loss function used for segmentation
³⁶³ problems:

$$\ell_i = x_i - x_i z_i + \log(1 + \exp(-x_i)) \quad (7)$$

³⁶⁴ where z_i is the ground-truth output target value of pixel i and x_i is the
³⁶⁵ CNN-predicted one.

³⁶⁶ To optimize the hyperparameters in our crater extraction and post-
³⁶⁷ processing routines (Sections 2.4 and 2.5) we use the precision, P , and
³⁶⁸ recall, R , to measure accuracy, calculated according to:

$$\begin{aligned} P &= \frac{T_p}{T_p + F_p} \\ R &= \frac{T_p}{T_p + F_n} \end{aligned} \tag{8}$$

369 where T_p are true positives, F_p are false positives and F_n are false negatives.
 370 There is always a trade-off between precision and recall. For example,
 371 a machine that only classifies craters when extremely certain will have a
 372 high precision but low recall, while a machine that classifies craters when
 373 only moderately certain will have a higher recall but lower precision. A
 374 common single-parameter metric that balances precision and recall is the
 375 F_1 score:

$$F_1 = 2 \frac{PR}{P + R} \tag{9}$$

376 Implicitly encoded into our accuracy metrics is the assumption that our
 377 ground-truth datasets of Head et al. (2010) and Povilaitis et al. (2017) are
 378 complete. However, as mentioned in Section 2.1 this is incorrect. As a
 379 result, genuine new craters identified by our CNN will be interpreted as
 380 false positives, penalizing these loss functions vs. improving them. This
 381 is an unavoidable consequence of using an incomplete ground truth (see
 382 Section 4 for further discussion).

383 To measure the accuracy of identified craters (Sections 2.4 and 2.5) we
 384 calculate the fractional errors in longitude, \mathcal{L} , latitude, L , and radius, R ,
 385 according to:

$$\begin{aligned}
 d\mathcal{L}/R &= \text{abs}(\mathcal{L}_P - \mathcal{L}_G) \cos(\pi \langle L \rangle / 180^\circ) / (R_G C_{KD}) \\
 dL/R &= \text{abs}(L_P - L_G) / (R_G C_{KD}) \\
 dR/R &= \text{abs}(R_P - R_G) / R_G
 \end{aligned} \tag{10}$$

386 where subscript P corresponds to our CNN-predicted craters, subscript G
 387 corresponds to our ground-truth craters and $\langle L \rangle = \frac{1}{2}(L_P + L_G)$.

388 Finally, our pipeline discovers thousands of new craters (as will be
 389 shown in Section 3). We measure the new crater percentage, P , according
 390 to:

$$P = N/(N + G) \tag{11}$$

391 where N is the number of CNN-predicted craters without a corresponding
 392 match from the ground truth (i.e. they are either genuine new craters or
 393 false positives), and G is the number of ground-truth craters. A “match” be-
 394 tween a CNN-predicted and ground-truth crater is determined via Eq. 3 and
 395 Eq. 6 for post-CNN (Section 2.4) and post-processed (Section 2.5) craters,
 396 respectively.

397 2.7. Training

398 A recurring theme in machine learning is “overfitting”, which occurs
 399 when a model latches onto overly-complex and/or irrelevant features dur-
 400 ing training. Overfit models typically achieve high accuracy on the train-
 401 ing set but low accuracy (poor generalization) on new data. Many algo-

402 rithms control for overfitting by penalizing overly-complex models, retain-
403 ing only the essential characteristics from the training data that will gener-
404 alize to new examples. This penalization is generally mediated through the
405 model’s hyperparameters, which control the model’s complexity. For our
406 CNN, such hyperparameters include weight regularizations for each con-
407 volutional layer, the learning rate, dropout layers after each merge layer,
408 filter size, and depth of the network. These hyperparameters are tuned on
409 a separate validation dataset, forcing the model to achieve high accuracy
410 on two different datasets.

411 After training the model and tuning the hyperparameters a final evalua-
412 tion is conducted on the test set, another dataset distinct from both the
413 training and validation datasets. If the model achieves comparable ac-
414 curacy on the test set as the training and validation sets, it is likely that
415 minimal overfitting has occurred and the model should generalize well to
416 new examples. We also address overfitting through a custom image aug-
417mentation scheme that randomly flips, rotates and shifts DEM images (and
418 their corresponding targets) before they are used to train the CNN. This
419 augments the effective dataset size and minimizes the chance of the CNN
420 generating features related to image orientation.

421 We tune the hyperparameters of our model by training 60 models with
422 randomly chosen hyperparameters over standard ranges on the training
423 set and selecting the model with the best binary cross-entropy score (Equa-
424 tion 7) on the validation set. Defining an “epoch” as a single pass through

425 the entire training set and “batch size” as the number of examples seen
 426 per backpropagation gradient update, each model is trained for 4 epochs
 427 with a batch size of 8 using the ADAM optimizer (Kingma and Ba, 2014).
 428 The hyperparameters of our best model are weight regularization = 10^{-5} ,
 429 learning rate = 10^{-4} , dropout = 15%, 3 × 3 filter sizes, and a depth of 3.

430 *2.8. Errors*

431 A few sources of error affect the final extracted (\mathcal{L}, L, R) coordinates of
 432 each detected crater. First, craters can only be detected in pixel increments,
 433 and converting from pixels to degrees yields a quantization error, E_q , of:

$$E_q = C_{\text{offset}} \frac{\Delta L}{\Delta H}, \quad (12)$$

434 where $C_{\text{offset}} \leq 1$ is a constant of order unity representing typical sub-
 435 pixel offsets. Setting $C_{\text{offset}} = 1$, and considering our largest DEM images
 436 (6500 pixels) where $\Delta L \approx 25^\circ$, we find a maximum quantization error of
 437 $E_q \approx 0.1^\circ$, or ~ 3 km. In principle this error could be reduced by increas-
 438 ing the pixel resolution of each DEM image, though doing so would be
 439 memory-intensive.

440 Second, objects within an orthographic projection become more dis-
 441 torted further from the central longitude and latitude (\mathcal{L}_0, L_0), which changes
 442 the size of smaller craters, and introduces non-circular deformations in
 443 larger ones. Along the central vertical axis, the deviation of the distorted

⁴⁴⁴ radius from our estimated value using Equation 4 is

$$F_{rd} = \frac{R_{\text{distorted}}}{R} = \frac{1}{\cos\left(\frac{\pi}{180^\circ}(L - L_0)\right)}. \quad (13)$$

⁴⁴⁵ For our largest DEM images, $L - L_0 \approx 12^\circ$, $F_{rd} \approx 1.02$, so deviations are at
⁴⁴⁶ most 2% of a crater's radius.

⁴⁴⁷ Third, the longitude and latitude estimates in Equation 4 neglect higher
⁴⁴⁸ order terms (including the distortion described above) and cross-terms con-
⁴⁴⁹ taining both \mathcal{L} and L . To quantify this effect we passed the crater pixel po-
⁴⁵⁰ sitions from the ground-truth test dataset through Equation 4 to obtain \mathcal{L}_C
⁴⁵¹ and L_C . We then subtracted the ground truth's longitude and latitude val-
⁴⁵² ues from \mathcal{L}_C and L_C , respectively, and normalized by the longitude/latitude
⁴⁵³ extent of our DEM images. We found median relative offsets of 0.13% and
⁴⁵⁴ 0.28% in longitude and latitude, but, for the largest DEM images, maxi-
⁴⁵⁵ mum relative offsets can reach 1.0% in longitude and 1.9% in latitude. For
⁴⁵⁶ a 6500 pixel DEM image, this translates to 0.25° in longitude and 0.5° in lat-
⁴⁵⁷ itude. We also calculated the fractional error using Equation 10, replacing
⁴⁵⁸ \mathcal{L}_P and L_P with \mathcal{L}_C and L_C , and find median fractional errors of 3% in
⁴⁵⁹ longitude and 5% in latitude.

⁴⁶⁰ To help offset these errors we impose a minimum search radius, $r_{\min} =$
⁴⁶¹ 5, for our crater extraction pipeline (Section 2.4). This prevents quanti-
⁴⁶² zation and projection errors from ever being a significant fraction of the
⁴⁶³ crater's radius. This comes at a cost of not being able to probe the smallest

⁴⁶⁴ craters in each DEM image, yielding fewer new crater detections than we
⁴⁶⁵ otherwise would obtain.

⁴⁶⁶ **3. Results**

⁴⁶⁷ *3.1. Crater Identification on the Moon*

⁴⁶⁸ We apply our trained CNN and optimized crater identification pipeline
⁴⁶⁹ on the test set and list our various accuracy metrics in Table 3.1 for the
⁴⁷⁰ validation and test sets. “Post-CNN” statistics were generated on an image-
⁴⁷¹ by-image basis after Section 2.4 of the pipeline with an averaged mean
⁴⁷² and standard deviation taken across all predicted targets. “Post-processed”
⁴⁷³ statistics were generated after Section 2.5 of the pipeline, and hence rep-
⁴⁷⁴ resent our final crater distribution after combining extracted craters from
⁴⁷⁵ each target into one distribution and removing duplicates. Together, these
⁴⁷⁶ statistics convey how our pipeline is performing at various stages.

⁴⁷⁷ The similarity between our validation and test set statistics in Table 3.1
⁴⁷⁸ implies that little to no overfitting has occurred. Our post-processed test
⁴⁷⁹ recall is 92%, recovering almost all craters from the test set. By compar-
⁴⁸⁰ison, our post-CNN test recall is lower at $57\% \pm 20\%$, meaning that (on
⁴⁸¹average) our CNN detects only half of the craters per target. The drastic
⁴⁸²difference between post-processed and post-CNN recalls demonstrates the
⁴⁸³effectiveness of aggregating crater detections across multiple images and
⁴⁸⁴scales. A major reason for our low post-CNN recall is our CNN does not
⁴⁸⁵reliably detect craters with radii greater than ~ 15 pixels (see Section 4 for

Accuracy Metric	Post-CNN (Validation)	Post- Processed (Validation)	Post-CNN (Test)	Post- Processed (Test)
Recall	$56\% \pm 20\%$	92%	$57\% \pm 20\%$	92%
Recall ($r < 15$ pixels)	$83\% \pm 16\%$	—	$83\% \pm 13\%$	—
Precision	$81\% \pm 16\%$	53%	$80\% \pm 15\%$	56%
New Crater Percentage	$12\% \pm 11\%$	45%	$14\% \pm 13\%$	42%
False Positive Rate	—	—	—	$11\% \pm 7\%$
Frac. longitude error	$10\%^{+2\%}_{-2\%}$	$13\%^{+10\%}_{-7\%}$	$10\%^{+2\%}_{-2\%}$	$11\%^{+9\%}_{-6\%}$
Frac. latitude error	$10\%^{+3\%}_{-2\%}$	$10\%^{+8\%}_{-5\%}$	$10\%^{+2\%}_{-2\%}$	$9\%^{+7\%}_{-5\%}$
Frac. radius error	$8\%^{+2\%}_{-2\%}$	$6\%^{+5\%}_{-3\%}$	$8\%^{+1\%}_{-1\%}$	$7\%^{+5\%}_{-4\%}$

Table 1: Accuracy metrics on the validation and test sets. “Post-CNN” statistics were generated after Section 2.4 of the pipeline with a mean and standard deviation taken across targets, while “post-processed” statistics were generated after Section 2.5 of the pipeline, after combining extracted craters into a single global distribution. Precision and recall are calculated according to Eq. 8, new crater percentage according to Eq. 11, fractional longitude, latitude and radius errors according to Eq. 10 with a median and interquartile range (IQR) taken across all detections. The false positive rate of new craters is estimated by four different scientists classifying 361 new craters and averaging the results. We note that precision drops at the post-processed stage because many new craters (absent in the ground truth) are identified.

486 a discussion). Restricting to craters with a pixel radius r less than 15 pixels,
487 our post-CNN test recall improves to $83\% \pm 16\%$. Wetzler et al. (2005) es-
488 timated a human recall of 75% when re-classifying crater images, making
489 our post-CNN recall consistent with human performance for $r < 15$ pixels.

490 42% of post-processed test craters are new, almost doubling our catalog,
491 with 15% of them having diameters under 5 km (i.e. below the limits of our
492 ground-truth catalogs). Our estimated false positive rate of these new post-
493 processed craters is $11\% \pm 7\%$, which was estimated by four scientists from
494 our research group each classifying the same 361 new craters re-projected
495 onto their original DEM images and averaging the results. This procedure
496 allowed the scientists to classify the new craters under the same conditions
497 as our identification pipeline. These manually classified craters along with
498 their corresponding Moon DEMs, ground-truth targets and CNN predictions
499 are publicly available at <https://doi.org/10.5281/zenodo.1133969>. Al-
500 though individual false positive estimates differed between scientists, this is
501 in line with previous research (e.g. Robbins et al., 2014) that large disagree-
502 ments in human crater classification is common. For post-CNN, $14 \pm 13\%$
503 of test craters per DEM image are new. As a result of these new crater
504 detections, our post-CNN and post-processed precisions are low since new
505 craters are interpreted by the precision metric exclusively as false positives
506 (see Section 4 for a discussion).

507 Figure 3 compares our post-processed craters (top left) to the ground
508 truth (top right) for a large swath of the Moon (bottom left) from the test

509 set. Blue circles represent post-processed craters that were successfully
 510 matched to the ground truth (and vice versa), red circles represent new
 511 crater detections from our pipeline (without a corresponding ground-truth
 512 match), and purple circles represent ground-truth craters missed by our
 513 pipeline. As can be visually seen, our pipeline recovers many more craters
 514 than the ground truth, with overall few false positives and duplicates. Our
 515 median post-processed and post-CNN fractional errors in longitude, lati-
 516 tude and radius are 11% or less, representing overall good agreement with
 517 the ground truth despite the sources of error mentioned in Section 2.8.

518 3.2. Lunar Crater Size Distribution

519 In Figure 4 we show the cumulative size-frequency distribution (CSFD)
 520 of the craters recovered by our CNN as compared with the CSFD of craters
 521 in the human counted set. This is constructed following the recommen-
 522 dations in NASA Technical Memorandum (79730, 1978). In this case the
 523 surface area, A , used to normalise the number of craters is the area of
 524 the Moon between latitudes of 60° North and South. We note that an-
 525 other widely used method to visualize crater distributions is R plots (as
 526 for example used in Head et al. (2010)), however we refrain from using
 527 these due to their significant dependency on data binning that can be com-
 528 pletely avoided using CSFDs, where each crater is its own bin (Weaver
 529 et al., 2015).

530 As we can see, between roughly 5 and 20 km diameters the CNN de-

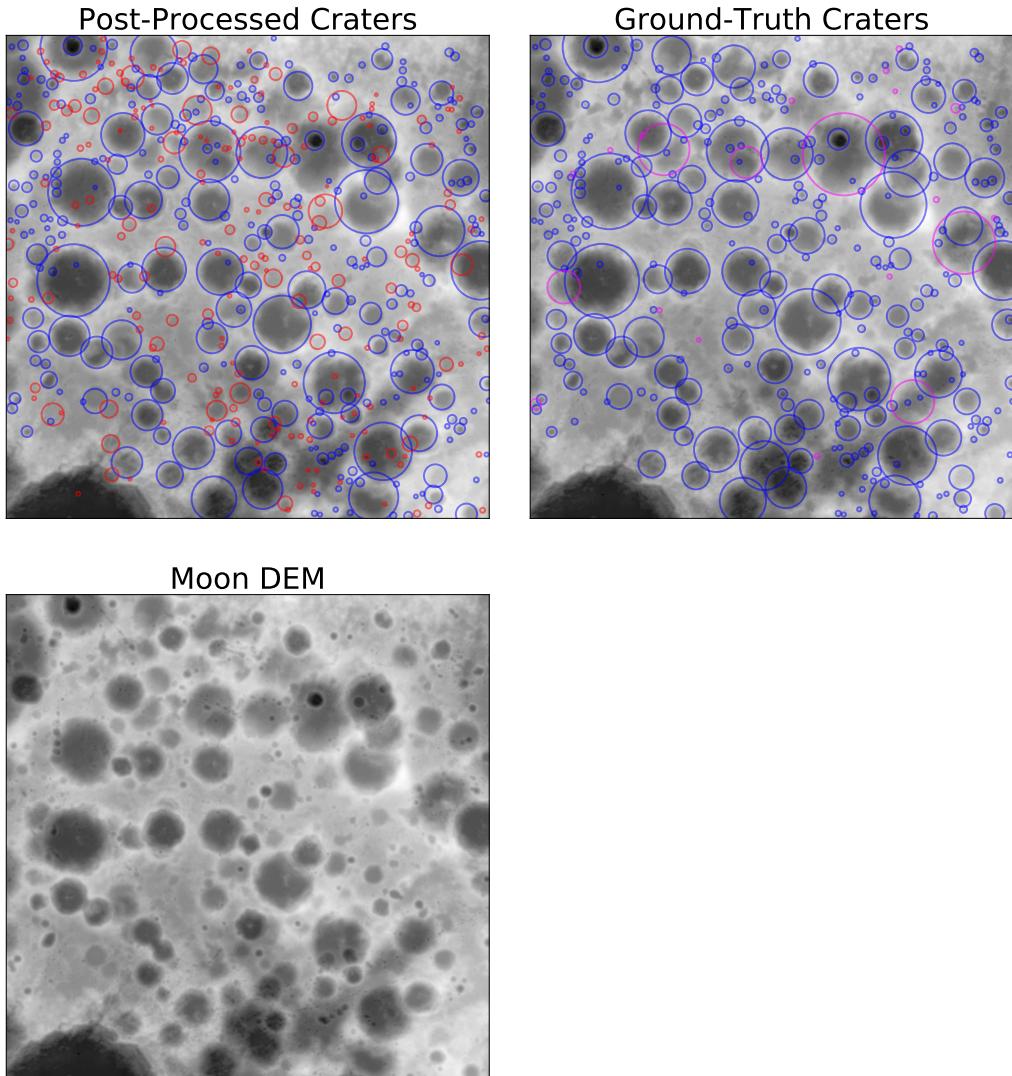


Figure 3: Sample patch of the Moon from the test set (lower left), with post-processed (top left) and ground-truth (top right) craters overlaid. Blue circles represent post-processed craters that were successfully matched to the ground truth (and vice versa), red circles represent new crater detections from our pipeline (without a corresponding ground-truth match), and purple circles represent ground-truth craters missed by our pipeline.

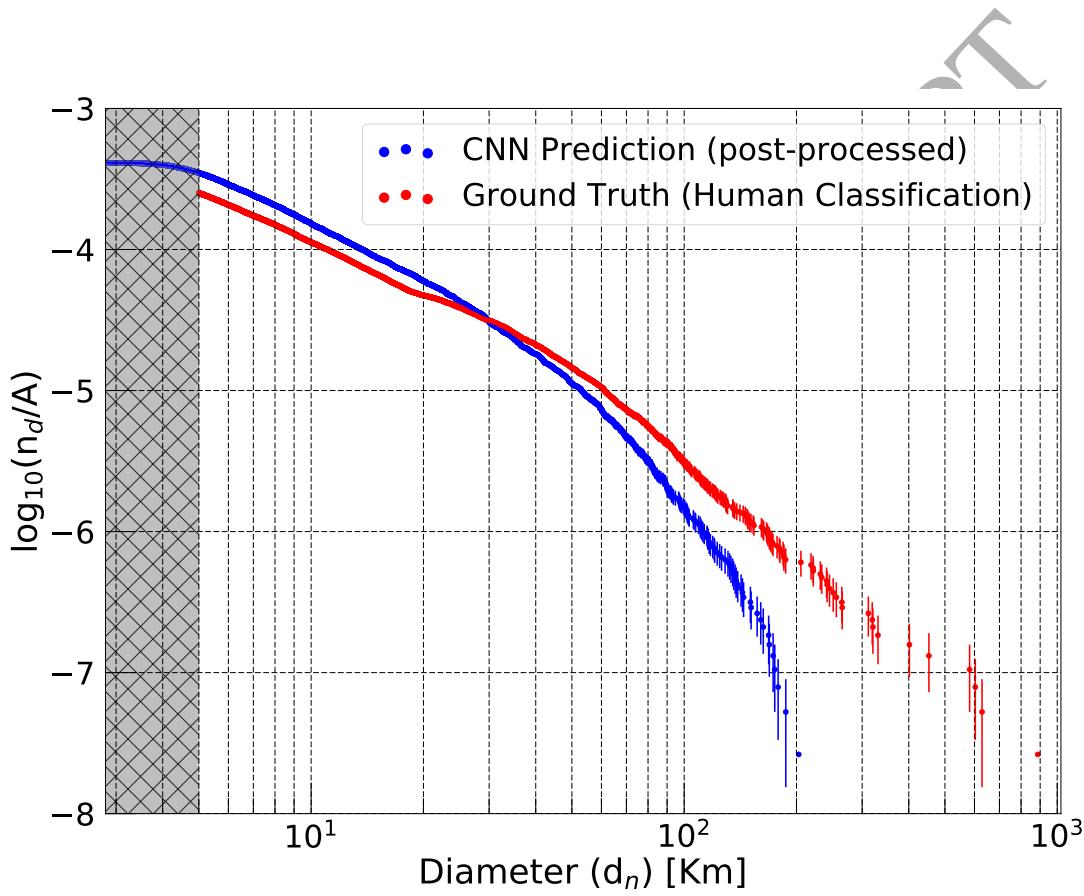


Figure 4: Lunar crater size-frequency distributions represented as CSFD plots. Red is the human-generated test set (the ground truth), compared to what our CNN predicts post-processing (blue). Our CNN recover the same slope as the GT for craters smaller than 20 km while detecting significantly more craters in this range. The shaded region inside 5 km however should not be physically interpreted due to data incompleteness.

531 rived CSFD is systematically higher than the human derived CSFD, but
532 runs essentially parallel to it. This indicates that while the CNN finds a
533 substantially larger number of craters than the human crater counters, the
534 craters newly identified by the CNN follow the same size distribution as the
535 human identified craters, which reassures us that they are likely to be real.
536 At 20 km however there is a slight kink and upturn in the human dataset
537 whereas the CNN prediction continues smoothly. The CNN prediction then
538 curves downward gradually before cutting off more sharply at diameters
539 of around 200 km, resulting in increasing divergence between the CNN
540 prediction and the human classifications for these larger craters.

541 There are several factors that might contribute to the divergence be-
542 tween the CSFDs of the CNN prediction and the human classified dataset
543 at large crater sizes. Firstly, as we describe in detail in Section 4, our CNN
544 can struggle with identifying craters that have a radius of more than around
545 15 pixels in the input images. While this is somewhat ameliorated by sam-
546 pling images at many different magnification levels (such that an $r > 15$ pix
547 crater in one image will have $r < 15$ pix in another) it can still slightly hin-
548 der the detection of craters with diameters larger than around 50 km. This
549 is likely responsible for some of the downward curvature of the CNN CSFD
550 at larger sizes and is probably the cause of the cut-off at ~ 200 km.

551 The location of the kink in the human classified CSFD at which the mis-
552 match between the CNN prediction and the human classification begins is
553 also notable. This kink appears at the boundary between the two compo-

554 nent datasets of our human classified ground truth, that of Povilaitis et al.
555 (2017) and Head et al. (2010). It is possible that there is a systematic dif-
556 ference between the crater classifications of the two groups. Indeed some
557 level of difference would be unsurprising given the findings of Greeley and
558 Gault (1970), Kirchoff et al. (2011) and Robbins et al. (2014). The sys-
559 tematic offset between the CSFDs of our CNN predictions and the human
560 classifications in the range where the two are parallel already indicates that
561 the human datasets are not complete, as was also found by Fassett et al.
562 (2012) for the Head et al. (2010) dataset. Since the Povilaitis et al. (2017)
563 dataset contains around four times as many craters it is inevitable that if
564 there are any systematic differences in the human classification that lead to
565 size-dependent completeness effects, the CNN will tend to follow Povilaitis
566 et al. (2017).

567 Another consideration is that 20 km is roughly the diameter at which
568 craters on the Moon transition from simple to complex (e.g. Pike, 1980;
569 Grieve, 1981; Stoffler, 2006). Above this size crater morphologies change,
570 with crater floors becoming flatter and the appearance of features like cen-
571 tral peaks. It is conceivable that this morphological transition could hinder
572 the detection of larger craters by the CNN. In particular, since smaller, sim-
573 ple craters dominate the human dataset, if the morphological differences
574 make it difficult to simultaneously optimise to detect both simple and com-
575 plex craters it is inevitable that the CNN will favour simple craters, just as
576 it would favour Povilaitis et al. (2017) over Head et al. (2010) in the case

577 of differences in their counting methods. In Section 4 we discuss how one
 578 could attempt to disentangle these possible effects.

579 Below around 5 km diameter the CNN prediction begins to roll over.
 580 This is due to incompleteness in the sampling of the lunar surface at the
 581 smallest scales/largest image magnifications. Despite incompleteness in
 582 the sampling and the lack of craters in the training set at these small sizes
 583 the CNN still finds many craters <5 km diameter.

584 *3.3. Transfer Learning on Mercury*

585 Domain shift is a common problem in machine learning that arises when
 586 a model is used to predict on data with different properties than its training
 587 set, and typically results in decreased performance. We briefly evaluate the
 588 sensitivity to domain shift for our network by taking our Moon-trained CNN
 589 and transfer learning to Mercury. Mercury has different properties than the
 590 Moon, including a different gravitational acceleration, surface composition,
 591 terrain, and impact history. In addition, we also use the Mercury MES-
 592 SENGER Global DEM with a resolution of 64 pixel/degree, or 665 m/pixel
 593 (Becker et al. 2016; available at USGS Astrogeology Science Center 2016),
 594 which has different image properties than our Moon DEM. All these affect
 595 the distribution and appearance of impact craters.

596 To evaluate our CNN on Mercury, we prepared DEM images from the
 597 Mercury MESSENGER Global DEM in a similar manner as described in
 598 Section 2.1 (except that we do not use a corresponding human-generated

PPT

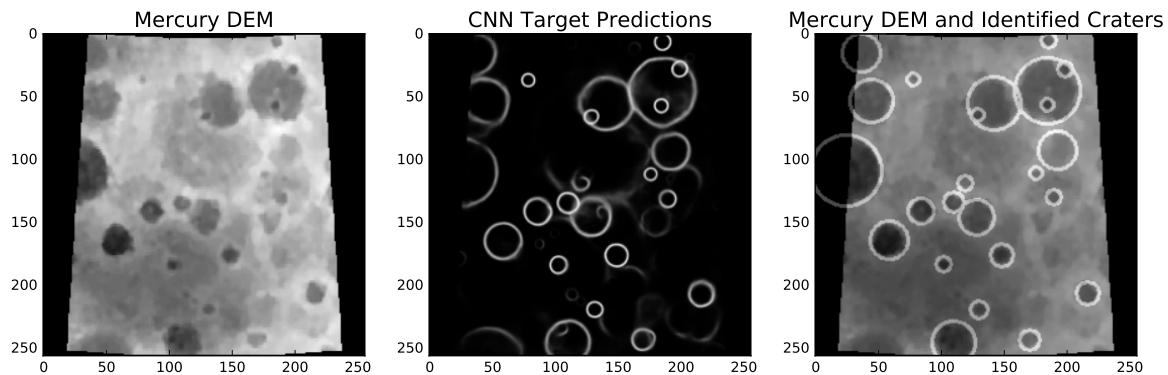


Figure 5: A sample Mercury DEM (left), CNN target predictions (middle), and post-processed identified craters overlaid on the original DEM image (right).

599 crater catalog). We then passed these DEM images through our Moon-
 600 trained CNN with no alterations to the architecture or weights. Figure 5
 601 shows a sample input DEM image of Mercury (left), CNN target predic-
 602 tions (middle), and post-processed identified craters overlaid on top of the
 603 original DEM image (right). Comparing the left panels of Figure 1 and Fig-
 604 ure 5, some differences are visible between Moon and Mercury craters, yet
 605 our CNN is able to detect both types. In addition, as shown in Figure 5, the
 606 CNN correctly identified a significant number of the crater-like features on
 607 Mercurys surface. Moreover, manual inspection on that patch of Mercury
 608 shows that almost all of the inferences made by the CNN do seem to be real
 609 craters. This demonstrates its efficiency in distinguishing craters from other
 610 terrain features. Simple edge detection techniques would not make such
 611 a distinction. While humans are often very good at transfer learning, and
 612 thus it may seem simple from a human perspective, transfer learning has
 613 commonly been challenging in machine learning. That our model seems
 614 to transfer well is thus greatly encouraging. While this demonstrates suc-
 615 cessful generalization, we leave a thorough analysis of transfer learning to
 616 future work.

617 4. Discussion

618 There are many reasons to believe our CNN has learned the complex
 619 features that define a crater. First, despite the Moon's large structure vari-
 620 ations across its surface our CNN was able to recover 92% of craters on a

621 face previously unseen by our CNN. Second, the similarity in accuracy met-
 622 rics between the validation and test sets in Table 3.1 implies that minimal
 623 overfitting has occurred, and the CNN has indeed learned useful, general-
 624 izable features. Third, 42% of extracted Lunar craters are new, and from
 625 human validation of a subset most appear to be genuine matches. Fourth,
 626 our Moon-trained CNN successfully detected craters on Mercury, a surface
 627 completely distinct from any specific region on the Moon. Finally, while
 628 simple edge detection techniques would activate non-crater features like
 629 mountains, ridges, etc., our CNN almost exclusively activates crater rims
 630 (e.g. see middle panel of Figure 5).

631 As mentioned in Section 2.1 and shown in Figure 1, our training data
 632 is incomplete, containing many missed craters as well as target rings that
 633 differ from true crater rims. Despite these shortcomings our CNN was still
 634 able to understand our training objective and correlate the binary ring tar-
 635 gets with the true rims of the craters. Proof of this can be seen in the middle
 636 panel of Figure 5, where some CNN-predictions are non-circular and bet-
 637 ter match the true crater rims than a circular ring could⁷. Together, these
 638 highlight the robustness and flexibility of deep learning solutions.

639 A fundamental difficulty when using an incomplete dataset is tuning
 640 the hyperparameters. Under this regime genuine new crater detections are

⁷To be clear, the 256×256 pixel CNN target predictions can produce non-circular ring boundaries, as shown in the middle panel of Figure 5. However, extracted post-processed craters (Section 2.4 and Section 2.5) do not retain non-circularity information, as shown in the right panel of Figure 5.

641 interpreted as false positives, penalizing the precision metric and artificially
642 lowering the F_1 score (which we are trying to maximize). Since thousands
643 of new craters were detected, the F_1 score, which favors hyperparameters
644 that yield the fewest new crater detections whilst still maintaining a high
645 recall, is reduced. As a result, our tuning procedure yields hyperparameters
646 that are likely conservative compared to if we had focused on finding new
647 craters or had a more complete ground truth. The same principle applies
648 when using the binary cross-entropy loss to train our CNN, yielding a final
649 model that likely generates more conservative crater predictions. Followup
650 work that uses a more complete ground-truth crater distribution would
651 presumably yield improved results.

652 Our CNN robustly detects craters from each DEM image with radii be-
653 low 15 pixels, but tends to miss larger craters. An example of this can be
654 seen in Figure 5, where a large crater with coordinates ($x = 130, y = 78, r =$
655 42 pix) is only partially recognized by the CNN and missed by our crater
656 extraction pipeline. We believe that this largely stems from the scale that is
657 imposed when using small 3x3 filters in our convolutional chain, yielding
658 a receptive field that is too small for large craters. Larger filter sizes were
659 attempted, but this dramatically increases both the number of trainable
660 parameters and network size, making model-optimization more difficult.
661 Dilated Convolutions (Yu and Koltun, 2015), larger convolution strides,
662 and/or deeper networks are possible avenues for improvement. However,
663 increasing the receptive field likely accomplishes the same effect as reduc-

664 ing the magnification of an image, which we have already shown to be a
 665 successful remedy, achieving a post-processed recall of 92% (see Table 3.1).

666 In addition, as we noted in Section 3.2, there are potential external ef-
 667 fects that may be feeding into the apparent size dependence of the ability of
 668 the CNN to recover larger craters. One possibility is that there is a sys-
 669 tematic difference in the two human counted datasets that we have stitched
 670 together to form our ground truth. Another is that the physical transition
 671 between simple and complex craters makes it difficult for the CNN to adapt
 672 to detect both with equal efficiency. In both cases the CNN would tend to
 673 adapt itself to better identify craters in the larger population, which is the
 674 Povilaitis et al. (2017) dataset that consists of craters below the ~ 20 km
 675 simple to complex transition. To disentangle these two possible effects re-
 676 quires a uniformly generated human dataset of both simple and complex
 677 craters. We would then be able to train the CNN on DEMs of solely sim-
 678 ple craters, solely complex craters, or both and then test it on DEMs of all
 679 physical scales/crater morphologies. This requires a substantial new hu-
 680 man counting effort and so we leave it for future work.

681 Our estimated post-processed false positive rate of new craters is $11\% \pm$
 682 7%, which, although generally low, is likely too high for our catalog to be
 683 used to produce high-precision crater catalogs. Our primary false positives
 684 are a) ambiguous circular looking depressions that may or may not be true
 685 craters (further analysis required), and b) overlapping craters that activate
 686 enough pixels in the region to breach the match probability threshold P_m ,

687 creating a “ghost” detection in our crater extraction pipeline (Section 2.4).
688 In addition, Table 3.1 shows that roughly 25% of post-processed craters
689 have coordinates that differ from the ground truth by 20% or more, and
690 examples of this can be seen in Figure 3. This higher-error tail is not present
691 in the post-CNN errors, so they arise from the post-processing methods,
692 whose sources of error are detailed in Section 2.8. These issues indicate the
693 need for further refinements to our overall crater identification pipeline in
694 order to produce precision crater catalogs, which we save for future work.

695 5. Conclusions and Future Work

696 In this work we have demonstrated the successful performance of a con-
697 volutional neural network (CNN) in recognizing Lunar craters from digital
698 elevation map images. In particular, we recovered 92% of craters from
699 our test set, and almost doubled the number of total crater identifications.
700 Furthermore, we have shown that our Moon-trained CNN can accurately
701 detect craters on the substantially different DEM images of Mercury. This
702 implies that the CNN has learned to robustly detect *craters*, and not features
703 particular to the planetary surface on which it was trained.

704 Two primary advantages of a deep learning solution over human crater
705 identification are consistency and speed. A CNN will classify an image
706 identically each time, but the same is not true for humans (Wetzler et al.,
707 2005). In addition, different humans will use slightly different criteria,
708 which adds to the error budget (Robbins et al., 2014). Once trained, our

709 CNN greatly increases the speed of crater identification, taking minutes to
710 generate predictions for tens of thousands of Lunar DEMs and a few hours
711 to extract a post-processed crater distribution from those DEMs. This is of
712 course all done passively, freeing the scientist to do other tasks. Our CNN
713 could also be used to assist human experts, generating initial suggestions
714 for the human expert to verify.

715 DEMs are available for many other Solar System bodies, including Mer-
716 cury (Becker et al., 2016), Venus (Magellan Science Team, 2014), Mars
717 (Ferguson et al., 2017), Vesta (Preusker et al., 2014) and Ceres (Preusker
718 et al., 2016). It will be interesting to study to what extent our CNN can
719 transfer-learn to other Solar System bodies with a DEM, possibly facili-
720 tating a systematic, consistent, and reproducible crater comparison across
721 Solar System bodies. While we have successfully shown transfer-learning
722 from our Moon-trained CNN to Mercury, a detailed analysis for Mercury
723 has been left to future work.

724 Our current work detected craters down to roughly 3 km diameter, but
725 since our CNN accepts images of arbitrary magnification we can transfer-
726 learn to kilometer and sub-kilometer scales on the Moon. We anticipate
727 that the uncharted territory of systematic small-size crater identification
728 will provide important new information about the size distribution of Lunar
729 impactors and the formation history of the Moon.

730 **Acknowledgments**

731 This work was supported by the Natural Sciences and Engineering Re-
732 search Council of Canada. Computations were performed on the P8 su-
733 percomputer at the SciNet HPC Consortium. SciNet is funded by: the
734 Canada Foundation for Innovation under the auspices of Compute Canada;
735 the Government of Ontario; Ontario Research Fund - Research Excellence;
736 and the University of Toronto. The authors are grateful to François Chol-
737 let for building Keras. The authors thank R. Povilaitis and the LROC team
738 for providing the 5-20 km dataset and useful discussions regarding the in-
739 put data, R. Malhotra and R. Strom for providing helpful comments on a
740 draft of the manuscript, and N. Hammond for useful discussions on craters
741 distributions. Finally, the authors thank David Minton and an anonymous
742 referee for their constructive comments during the review process that sig-
743 nificantly improved the quality of this manuscript.

744 **References**

- 745 Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Cor-
746 rrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I.,
747 Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur,
748 M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C.,
749 Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P.,
750 Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wat-
751 tenberg, M., Wicke, M., Yu, Y., Zheng, X., Mar. 2016. TensorFlow: Large-

- 752 Scale Machine Learning on Heterogeneous Distributed Systems. ArXiv
753 e-prints.
- 754 Bandeira, L., Ding, W., Stepinski, T. F., Jan. 2012. Detection of sub-
755 kilometer craters in high resolution planetary images using shape and
756 texture features. *Advances in Space Research* 49, 64–74.
- 757 Barker, M. K., Mazarico, E., Neumann, G. A., Zuber, M. T., Haruyama, J.,
758 Smith, D. E., Jul. 2016. A new lunar digital elevation model from the
759 Lunar Orbiter Laser Altimeter and SELENE Terrain Camera. *Icarus* 273,
760 346–355.
- 761 Becker, K. J., Robinson, M. S., Becker, T. L., Weller, L. A., Edmundson, K. L.,
762 Neumann, G. A., Perry, M. E., Solomon, S. C., Mar. 2016. First Global
763 Digital Elevation Model of Mercury. In: *Lunar and Planetary Science*
764 Conference. Vol. 47 of *Lunar and Planetary Inst. Technical Report*. p.
765 2959.
- 766 Bottke, W. F., Norman, M. D., Aug. 2017. The Late Heavy Bombardment.
767 *Annual Review of Earth and Planetary Sciences* 45, 619–647.
- 768 Boukercha, A., Al-Tameemi, A., Grumpe, A., Wöhler, C., Mar. 2014. Auto-
769 matic Crater Recognition Using Machine Learning with Different Features
770 and Their Combination. In: *Lunar and Planetary Science Conference*.
771 Vol. 45 of *Lunar and Planetary Inst. Technical Report*. p. 2842.

- 772 Canny, J., 1986. A computational approach to edge detection. IEEE Trans-
 773 actions on pattern analysis and machine intelligence (6), 679–698.
- 774 Chapman, C. R., Cohen, B. A., Grinspoon, D. H., Jul. 2007. What are the
 775 real constraints on the existence and magnitude of the late heavy bom-
 776 bardment? Icarus 189, 233–245.
- 777 Chollet, F., 2015. Keras. <https://github.com/fchollet/keras>.
- 778 Chung, C. P., Chung, O. C., Yam, C. H., 2014. Lunar Crater Identifica-tion
 779 from Machine Learning Perspective. Acta Futura 9, 41–45.
- 780 URL <http://www.esa.int/gsp/ACT/publications/ActaFutura/>
 781 issues.html
- 782 Cohen, J. P., Lo, H. Z., Lu, T., Ding, W., Mar. 2016. Crater Detection via
 783 Convolutional Neural Networks. In: Lunar and Planetary Science Con-
 784 ference. Vol. 47 of Lunar and Planetary Inst. Technical Report. p. 1143.
- 785 Di, K., Li, W., Yue, Z., Sun, Y., Liu, Y., 2014. A machine learning approach
 786 to crater detection from topographic data. Advances in Space Research
 787 54 (11), 2419 – 2429.
- 788 URL [http://www.sciencedirect.com/science/article/pii/
 789 S0273117714005304](http://www.sciencedirect.com/science/article/pii/S0273117714005304)
- 790 Duda, R. O., Hart, P. E., Jan. 1972. Use of the hough transformation to
 791 detect lines and curves in pictures. Commun. ACM 15 (1), 11–15.
- 792 URL <http://doi.acm.org/10.1145/361237.361242>

- 793 Emami, E., Bebis, G., Nefian, A., Fong, T., 2015. Automatic Crater Detection
 794 Using Convex Grouping and Convolutional Neural Networks. Springer,
 795 Cham, pp. 213–224.
- 796 URL https://doi.org/10.1007/978-3-319-27863-6_20
- 797 Fassett, C. I., Head, J. W., Kadish, S. J., Mazarico, E., Neumann, G. A.,
 798 Smith, D. E., Zuber, M. T., Feb. 2012. Lunar impact basins: Stratigraphy,
 799 sequence and ages from superposed impact crater populations measured
 800 from Lunar Orbiter Laser Altimeter (LOLA) data. Journal of Geophysical
 801 Research (Planets) 117, E00H06.
- 802 Fergason, R., Hare, T., Laura, J., 2017. HRSC and MOLA blended digital
 803 elevation model at 200m. http://bit.ly/HRSC_MOLA_Blend_v0.
- 804 Gomes, R., Levison, H. F., Tsiganis, K., Morbidelli, A., May 2005. Origin of
 805 the cataclysmic Late Heavy Bombardment period of the terrestrial plan-
 806 etes. Nat.435, 466–469.
- 807 Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press,
 808 <http://www.deeplearningbook.org>.
- 809 Greeley, R., Gault, D. E., Sep. 1970. Precision Size-Frequency Distributions
 810 of Craters for 12 Selected Areas of the Lunar Surface. Moon 2, 10.
- 811 Grieve, R. A. F., May 1981. Impact cratering. Nat.291, 16.
- 812 Hartmann, W. K., Sep. 1970. Note: Lunar Cratering Chronology. Icarus13,
 813 299–301.

- 814 Head, J. W., Fassett, C. I., Kadish, S. J., Smith, D. E., Zuber, M. T., Neu-
815 man, G. A., Mazarico, E., Sep. 2010. Global Distribution of Large Lunar
816 Craters: Implications for Resurfacing and Impactor Populations. *Science*
817 329, 1504.
- 818 Kingma, D. P., Ba, J., Dec. 2014. Adam: A Method for Stochastic Optimiza-
819 tion. ArXiv e-prints.
- 820 Kirchoff, M., Sherman, K., Chapman, C., Oct. 2011. Examining lunar im-
821 pactor population evolution: Additional results from crater distributions
822 on diverse terrains. In: EPSC-DPS Joint Meeting 2011. p. 1587.
- 823 LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard,
824 W., Jackel, L. D., 1989. Backpropagation applied to handwritten zip code
825 recognition. *Neural Computation* 1 (4), 541–551.
826 URL <https://doi.org/10.1162/neco.1989.1.4.541>
- 827 LOLA Team, Kaguya Team, 2015. LRO LOLA and Kaguya Ter-
828 rain Camera DEM merge 60N60S 512ppd (59m). https://astrogeology.usgs.gov/search/map/Moon/LRO/LOLA/Lunar_LRO_LrockKaguya_DEMmerge_60N60S_512ppd.
- 829
830
831 Long, J., Shelhamer, E., Darrell, T., June 2015. Fully convolutional net-
832 works for semantic segmentation. In: The IEEE Conference on Computer
833 Vision and Pattern Recognition (CVPR).

- 834 Magellan Science Team, 2014. Magellan global topography 4641m.
835 <https://astrogeology.usgs.gov/search/map/Venus/Magellan/>
836 RadarProperties/Venus_Magellan_Topography_Global_4641m_v02.
- 837 Minton, D. A., Richardson, J. E., Fassett, C. I., Feb. 2015. Re-examining
838 the main asteroid belt as the primary source of ancient lunar craters.
839 *Icarus*247, 172–190.
- 840 Palafox, L. F., Hamilton, C. W., Scheidt, S. P., Alvarez, A. M., 2017. Auto-
841 mated detection of geological landforms on mars using convolutional
842 neural networks. *Computers Geosciences* 101 (Supplement C), 48 – 56.
843 URL <http://www.sciencedirect.com/science/article/pii/S0098300416305532>
- 845 Pike, R. J., Jul. 1980. Formation of complex impact craters - Evidence from
846 Mars and other planets. *Icarus*43, 1–19.
- 847 Povilaitis, R., Robinson, M., van der Bogert, C., Hiesinger, H., Meyer, H.,
848 Ostrach, L., 2017. Crater density differences: Exploring regional resur-
849 facing, secondary crater populations, and crater saturation equilibrium
850 on the moon. *Planetary and Space Science*.
851 URL <http://www.sciencedirect.com/science/article/pii/S0032063316303798>
- 853 Preusker, F., Scholten, F., Matz, K.-D., Elgner, S., Jaumann, R., Roatsch, T.,
854 Joy, S. P., Polanskey, C. A., Raymond, C. A., Russell, C. T., Mar. 2016.

- 855 Dawn at Ceres Shape Model and Rotational State. In: Lunar and Planetary Science Conference. Vol. 47 of Lunar and Planetary Inst. Technical Report. p. 1954.
- 856
- 857
- 858 Preusker, F., Scholten, F., Matz, K.-D., Roatsch, T., Jaumann, R., Raymond, C. A., Russell, C. T., Feb. 2014. Global Shape of (4) Vesta from Dawn 859 FC Stereo Images. In: Vesta in the Light of Dawn: First Exploration of a 860 Protoplanet in the Asteroid Belt. Vol. 1773 of LPI Contributions. p. 2027.
- 861
- 862 Robbins, S. J., Antonenko, I., Kirchoff, M. R., Chapman, C. R., Fassett, C. I., Herrick, R. R., Singer, K., Zanetti, M., Lehan, C., Huang, D., Gay, P. L., May 2014. The variability of crater identification among expert and 863 community crater analysts. *Icarus* 234, 109–131.
- 864
- 865
- 866 Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional Networks 867 for Biomedical Image Segmentation. Springer International Publishing, Cham, pp. 234–241.
- 868
- 869 URL https://doi.org/10.1007/978-3-319-24574-4_28
- 870 Rumelhart, D. E., Hinton, G. E., Williams, R. J., Oct. 1986. Learning representations by back-propagating errors. *Nat.* 323, 533–536.
- 871
- 872 Ryder, G., Apr. 2002. Mass flux in the ancient Earth-Moon system and be- 873 nign implications for the origin of life on Earth. *Journal of Geophysical 874 Research (Planets)* 107, 6–1.

- 875 Salamunićcar, G., Lončarić, S., 2010. Method for crater detection from digi-
 876 tal topography data: interpolation based improvement and application to
 877 Lunar SELENE LALT data. In: 38th COSPAR Scientific Assembly. Vol. 38
 878 of COSPAR Meeting. p. 3.
- 879 Salamunićcar, G., Lončarić, S., Grumpe, A., Wöhler, C., Jun. 2014. Hybrid
 880 method for crater detection based on topography reconstruction from op-
 881 tical images and the new LU78287GT catalogue of Lunar impact craters.
 882 Advances in Space Research 53, 1783–1797.
- 883 Scholten, F., Oberst, J., Matz, K.-D., Roatsch, T., Wählisch, M., Speyerer,
 884 E. J., Robinson, M. S., Mar. 2012. GLD100: The near-global lunar 100 m
 885 raster DTM from LROC WAC stereo image data. Journal of Geophysical
 886 Research (Planets) 117, E00H17.
- 887 Shallue, C. J., Vanderburg, A., Feb. 2018. Identifying Exoplanets with Deep
 888 Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth
 889 Planet around Kepler-90. Astron. J.155, 94.
- 890 Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., Salakhutdinov,
 891 R., 2014. Dropout: a simple way to prevent neural networks from over-
 892 fitting. Journal of machine learning research 15 (1), 1929–1958.
- 893 Stepiniski, T., Ding, W., Vilalta, R., 2012. Detecting impact craters in plane-
 894 tary images using machine learning. Information Science Reference, Her-
 895 shey, PA, pp. 146–159.

- 896 Stoffler, D., Jan. 2006. Cratering History and Lunar Chronology. *Reviews*
 897 in Mineralogy and Geochemistry
- 898 Strom, R. G., Malhotra, R., Ito, T., Yoshida, F., Kring, D. A., Sep. 2005. The
 899 Origin of Planetary Impactors in the Inner Solar System. *Science* 309,
 900 1847–1850.
- 901 UK Met. Office, 2015. Cartopy: a cartographic python library with a mat-
 902plotlib interface. <http://scitools.org.uk/cartopy/index.html>.
- 903 USGS Astrogeology Science Center, 2016. Mercury MESSENGER global
 904 DEM 665m (64ppd) v2 oct. 2016. <https://astrogeology.usgs.gov/>
 905 search/map/Mercury/Topography/MESSENGER/Mercury_Messenger_
 906 USGS_DEM_Global_665m.
- 907 Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F.,
 908 Warner, J. D., Yager, N., Gouillart, E., Yu, T., 2014. scikit-image: image
 909 processing in python. *PeerJ* 2, e453.
- 910 Weaver, B. P., Robbins, S. J., Plesko, C. S., Riggs, J. D., May 2015. On the
 911 Binning and Associated Uncertainty of Crater Diameter Size-Frequency
 912 Distributions. In: Issues in Crater Studies and the Dating of Planetary
 913 Surfaces. Vol. 1841 of LPI Contributions. p. 9054.
- 914 Wetzler, P., Honda, R., Enke, B., Merline, W., Chapman, C., Burl, M., 2005.
 915 Learning to detect small impact craters. In: 7th IEEE Workshop on Appli-
 916 cation of Computer Vision. Vol. 1. pp. 178–184.

- 917 Yu, F., Koltun, V., Nov. 2015. Multi-Scale Context Aggregation by Dilated
918 Convolutions. ArXiv e-prints.
- 919 Zeiler, M. D., Fergus, R., Nov. 2013. Visualizing and Understanding Convolu-
920 tional Networks. ArXiv e-prints.
- 921 Zeng, L., Xu, X., Cai, B., Qiu, S., Zhang, T., 2017. Multi-scale convolutional
922 neural networks for crowd counting. CoRR abs/1702.02359.
923 URL <http://arxiv.org/abs/1702.02359>
- 924 Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y., June 2016. Single-image
925 crowd counting via multi-column convolutional neural network. In: The
926 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).