Investigating Convolutional Neural Network Applications for NASA HiRISE Mars Landmark Classification

Thomas Lu
Science Academy
University of Maryland, College Park
College Park, MD, United States of America
tlu37@umd.edu

***Abstract*—In this paper, the author evaluated the application of deep learning methods such as the Convolutional Neural Networks (CNN) to Mars landmark identification. Utilizing the Mars Orbital Image (HiRISE) Labeled Dataset Version 3.2, a total of 64,947 images of the Martian surface were preprocessed utilizing Apache Spark. Following preprocessing, this data was then trained and classified by a modified AlexNet architecture, evaluated by accuracy, recall, precision, and F1-score. After training, the trained AlexNet model was able to attain an F1 of 89%. The outcome of this experiment demonstrates the potential efficacy of deep learning on Mars landmark identification. Future work can focus on finding solutions to the severe class imbalance of the original dataset, as well as utilizing transfer learning on Earth-based landmarks to potentially achieve better results in identifying aspects of the Martian topography.**

Keywords—Computer Vision, Convolutional Neural Network, AlexNet, Mars, NASA, HiRISE, Mars Reconnaissance Orbiter, PyTorch, Spark

I. Introduction

As humanity comes closer every year towards exoplanetary exploration, it's become a crucial step to study the geology and topography of said planets. Significant amounts of work have been done from our observations of one of Earth's nearest neighbors, such as determining that the iconic red color of the soil is likely the result of various iron-bearing isotopes and materials within the soil in the form of iron rust. However, limitations arose when it came to examining the surface for specific landmarks and features, due to a variety of issues. Things such as the Earth's atmosphere and the Martian atmosphere distorting images from Earth-based telescopes, as potential disputed issues of color contrast and the limitations of the human eye, have made it difficult to study Mars, ever since humans began studying and writing about the red planet in the 1600s.

With the launch of satellite-based telescopes and space missions to Mars, we now know significantly more about the surface, both observationally and experimentally, in the case of rovers such as NASA's Spirit and Opportunity. Some of the highest fidelity imagery that we currently have of the Martian surface comes from the Mars Reconnaissance Orbiter (MRO), which was launched by NASA on August 12, 2005. Reaching Mars orbit on March 10, 2006, the orbiter contains a total of six different imaging tools, three engineering instruments, and two science facility experiments designed. These capabilities onboard are designed to not only acquire valuable information regarding the topography, atmosphere, and mineral composition of the surface, but also provide data for future development of spacecraft to be operating near Mars or further beyond. Despite its primary science phase only intending to last 2 years, the orbiter has maintained its relevancy and importance well into the modern day and is expected to be retired some time in the late 2020's, a far cry from its original end date.

In terms of identifying landmarks on the surface of Mars, the High-Resolution Imaging Science Experiment (HiRISE) camera onboard the MRO is one of the critical components in collecting new data on geological landforms. The data that's captured by this camera system comprises 28 gigabits (Gb) images, which is about 3.5 gigabytes (GB). This limitation is largely constrained by onboard memory capacity limitations. At a resolution of about 0.3m/pixel at 300km [2], the amount of information of the ground below that can be captured is far beyond consumer grade telescopes and imaging devices, and thus gives us our clearest images of the surface yet. Each image taken by the HiRISE camera can vary in resolution, however the maximum resolution on the red wavelength can be up to 20,000 pixels x 126,000 pixels [2], resulting in long, rectangular images. On the blue, green, and near-infrared wavelength bands, the images are even narrower, at only 4000 pixels x 126,000 pixels [2], due to the architecture of the camera itself. This camera is also able to use stereo pairs of images to map depth, as well as in conjunction with other camera systems onboard such as the Context Camera to provide 3D depth mapping of the Martian surface [2].

While current successful missions to Mars have primarily revolved around sending unmanned satellites and rovers to the planet, future missions will require more information regarding the surface to identify new locations of interest. This motivation is not exclusive to sending new unmanned rovers to the planet, but also potentially manned missions to the red planet. Despite the terabytes of data that this camera system has created, it is unfeasible to attempt to manually identify all of the various landmarks that could be present on the Martian surface. The advent of deep learning models and architectures offers and opportunity to utilize machine learning to assist in identifying and classifying these different landmarks.

In this study, I investigated the performance of the popular convolutional neural network AlexNet in this image

classification task. Originally designed to classify the ImageNet dataset, this model remains one of the most popular CNNs for its general performance. I hypothesized that due to its resilience and capacity to classify the numerous types of images that are present within the ImageNet dataset, AlexNet would likely also be able to handle classifying Mars landmarks taken from HiRISE images.

II. Methodology

A. Dataset

The dataset that was used for this dataset is version 3.2 of the Mars orbital image (HiRISE) labeled data set [1]. As noted from the dataset's publishers from the NASA's Jet Propulsion Lab, the dataset was formed from a set of 10,815 original landmarks, which were cropped from a total of 232 source HiRISE images and resized to 227x227 pixels. 9,022 have been augmented by the original team to generate 6 additional landmarks using the following methods:

1. 90 degrees clockwise rotation
2. 180 degrees clockwise rotation
3. 270
4. Horizontal flip
5. Vertical flip
6. Random brightness adjustment

Adding these augmented images to the original dataset left the dataset with a total of 64,947 JPG landmark images. These images vary somewhat in clarity, and some contain large angular sections of blacked out pixels, presumably due to the fact that they were cropped from the edges of their source HiRISE images. An example is shown below.

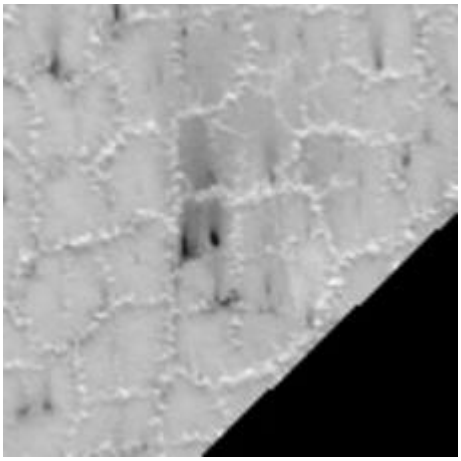

Fig. 1. Image "ESP_013049_0950_RED-0067.jpg" from the HiRISE dataset, representing a "Spider" in a polar cap. Note the bottom right corner being dark, due to where it was cropped from in the source image.

Each image is tagged 0 to 7 within a separate text file, denoting the type of landmark that image represents. A description of what each value is, as well as a short description of said landmark, is included in Table 1. These descriptions of each class were included in the publisher's description of the dataset and are included here for the reader's understanding.

TABLE I. HiRISE DATASET CLASSIFICATION DESCRIPTION

Value	***Classification***	***Description***
0	Other	Catch-all class for images of landmarks not of interest
1	Crater	Images of craters >= 1/5 the width of image, with at least half the crater's crater's circumference visible
2	Dark Dune	Sand dune, completely defrosted
3	Slope Streak	Slopes with dark flow-like features
4	Bright Dune	Sand dunes, with frost on surface
5	Impact Ejecta	Material blasted out by meteorite impact or volcanic eruption
6	Swiss Cheese	Pits in polar caps, likely due to CO_2 sublimation
7	Spider	Central pit with radial troughs in the south polar caps

Fig. 2. Description of class labels, as noted by dataset publisher [1].

B. Data Preprocessing

To preprocess the data, the folder of images was first ingested into PySpark as a Spark dataframe of binary files. Following ingestion, a user defined function was utilized to both resize the images to 224x224 pixel arrangements, to better accommodate the AlexNet implementation that will be used in this experiment. This UDF also normalized the pixel values from the original 0-255 to a range of 0 to 1 by dividing each pixel by 255. This leaves each image as a 224x224 greyscale image, with smaller values to speed up the model's learning.

As noted by the dataset publishers, a vast majority of the images within the dataset are classified as 0 or "Other". To visualize this, I utilized the matplotlib package to illustrate the overall distribution of each label within the dataset.

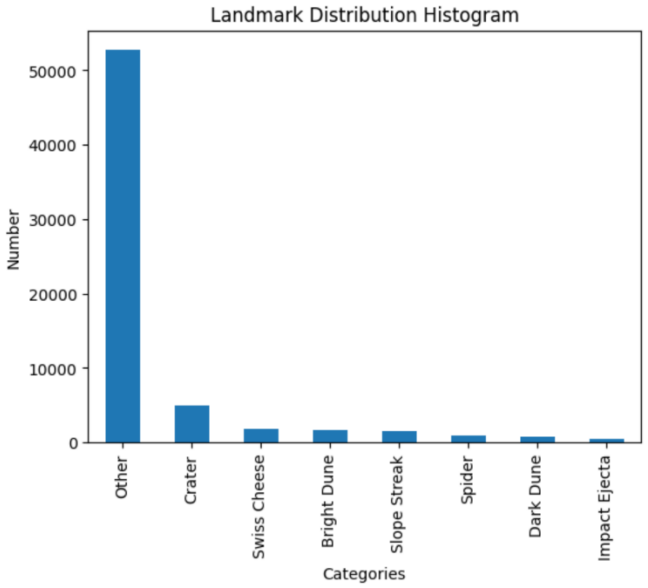

Fig. 3. Distribution of labels within the dataset.

This revealed that about 81% of the dataset is classified as "other". While this does reveal the dataset to be extremely imbalanced, such as with the Impact Ejecta class having very few entries, the previously generated augmentations that the dataset publishers contributed already encompassed many of the common augmentation methods. In addition, further augmentations would be likely to cause additional overfitting based on the limited original source images. Due to these factors, additional preprocessing was not performed.

Following this preprocessing the Spark dataframe was then converted into a Pandas dataframe, where the resized and normalized landmark images were then saved locally to a hard drive as a NumPy array. This choice was in large part due to memory constraints present on this device, which ended up causing significant issues when it came to designing in-memory solutions. Due to the intention to use PyTorch as the framework of choice for implementation of the deep learning model, the images were saved in a file structure that would be easily compatible with the base functionality of the PyTorch Dataset class, to load images from drive instead of memory.

After all processing was completed, the dataset was split into training and testing sets, with a total of 75% of the data going into training and 25% into testing. Due to the high volume of data that would be used to train this model, I felt confident in utilizing this split.

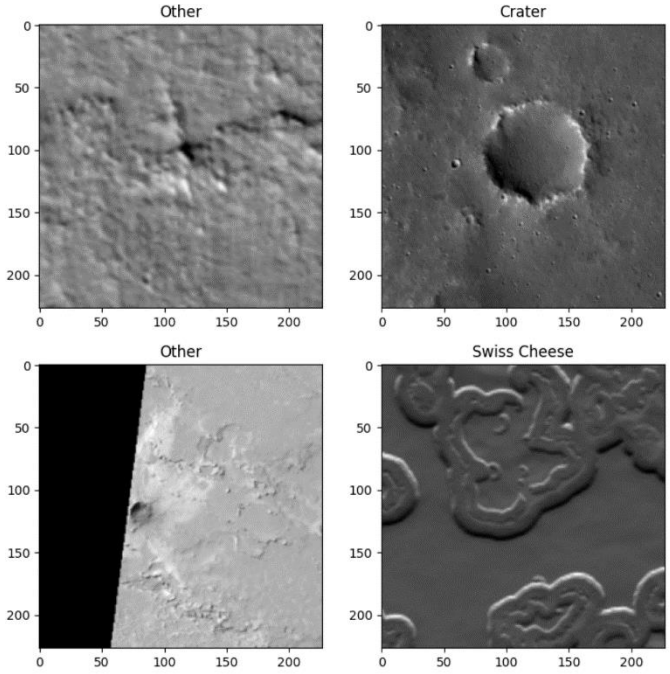

Fig. 4. Ingested images converted to binary, and visualized utilizing Matplotlib.

C. AlexNet

Deep learning has proven itself incredibly powerful in the pursuit of machine learning, with many state-of-the-art solutions to several different fields and areas having been developed in the past few years. In particular, convolutional neural networks have proven themselves exceptional in processing data such as images, utilizing convolutional layers to learn the features of input images to conduct computer vision tasks such as image classification.

AlexNet was one of the major breakthroughs and a significant advancement in deep learning, being state-of-the-art upon its release. Among the things that it remains known for includes it's utilization of the Rectified Linear Unit (ReLU) activation function as opposed to the tanh or sigmoid activation functions, as well as its utilization of GPUs in its training to facilitate high speed training, despite being deeper than many of its contemporaries at 5 convolutional layers and 3 fully connected layers.

While there are many newer models available, I decided to utilize and experiment with AlexNet, given that it is a CNN that is relatively well understood, while also having a good mix of training speed, performance, and robustness to image variation, which comprised most of the dataset since a majority of the dataset is augmented from the original landmark images. Likewise, I decided to utilize PyTorch as my deep learning framework for this experiment.

To utilize AlexNet, however, some modifications had to be made to the original design. The original AlexNet was designed to take in 224x224x3 images from the ImageNet dataset. In addition, the final classifier involves a fully connected layer outputting to a total of 1000 classes. Given that this Martian landmark dataset comprises of 227x227x1 images in greyscale, I altered the first layer of the PyTorch implementation to take in the single channel images that we preprocessed earlier. Additionally, the final output layer was modified to output a total of 8 classes, one for each of the labels present in this dataset. The

TABLE II. MODIFIED ALEXNET STRUCTURE

Layer Type	***Parameters***
Input	224x224x1 greyscale image
Convolution 1	11x11 kernel, 4 stride, 2 padding, ReLU, 64 filter
Max Pooling1	3x3 kernel, 2 stride
Convolution 2	5x5 kernel, 2 padding, ReLU, 192 filter
Max Pooling 2	3x3 kernel, stride 2
Convolution 3	3x3 kernel, 1 padding, ReLU, 384 filter
Convolution 4	3x3 kernel, 1 padding, ReLU, 256 filter
Convolution 5	3x3 kernel, 1 padding, ReLU, 256 filter
Max Pooling 3	3x3 kernel, 2 stride
Linear	4096 length tensor, ReLU, 0.5 dropout
Linear	4096 length tensor, ReLU, 0.5 dropout
Linear	8 length tensor, ReLU
Output	1 class label

Fig. 5. Modified AlexNet layers and structure. This CNN is based on the PyTorch implementation of the original AlexNet.

D. Evaluation Metrics

For evaluation, four different metrics were utilized in determining the effectiveness of this CNN: accuracy, weighted recall, weighted precision, and weighted F1 score, utilizing the

scikit-learn implementation. These weighted metrics are based on the average weighted metrics determined by each label's support, or the number of true instances for each label. This variation was chosen due to the nature of this classification problem, being multi-class and imbalanced. This weighting should account for the imbalanced nature of the data when evaluating.

E. Training

Several hyperparameters were set as part of the training process. This test was performed utilizing 20 epochs of training, utilizing a batch size of 64 and a learning rate of 0.005, utilizing the modified AlexNet model described above. Additionally, the PyTorch implementation of the Cross Entropy Loss, primarily to handle the multi-class nature of this data, as well the Adam optimizer with a learning rate of 0.0001.

To avoid possible confounding variables with regards to transfer learning, I loaded the AlexNet model from PyTorch with random weights. While transfer learning has been shown to be incredibly powerful when it comes to progressing neural networks, there are significant differences when it comes to the ImageNet dataset and the HiRISE dataset. HiRISE is solely comprised of terrestrial objects and land masses, while ImageNet has a large variety of general terms comprising its dataset. It's unclear whether transfer learning would benefit or harm the training of this model, and may be explored in future work, however this study does not attempt to explore this variable.

Additional configurations were set up to accelerate training. PyTorch offers several libraries for performing calculations on tensors via CUDA-enabled NVIDIA GPU. This allows for much faster calculation and training compared to a CPU only library, and this was considered with the machine that this model was trained on. The author's laptop, which this code was run on, utilized an AMD Ryzen 9 6900HS, with 32GB of RAM and an NVIDIA GeForce RTX 3050 Laptop GPU. Utilizing this setup, training took around 34.2 minutes for 20 epochs, which was significantly faster than CPU training.

After each epoch, the model was evaluated on its weighted F1-score. If the model's score was better than the previous best model, the model's weights were saved to disk for retrieval later. The choice to base things on F1-score instead of accuracy was in large part due to the imbalanced dataset, as well as the desire to take false positives and false negatives into consideration when evaluating, which would not have occurred under just accuracy.

As a result of the 75/25 train/test split that was done on the full dataset during preprocessing, the PyTorch training dataset and dataloader was comprised of 48778 images, while the testing/validation dataset was the remaining 16169 images.

III. Results

At the end of each epoch, cross entropy loss values for both the training and the validation were saved, and plotted out. The validation loss of each epoch seems to swing wildly over the course of training, while the training loss seems to start low and continue low. This can be shown below in Fig. 6.

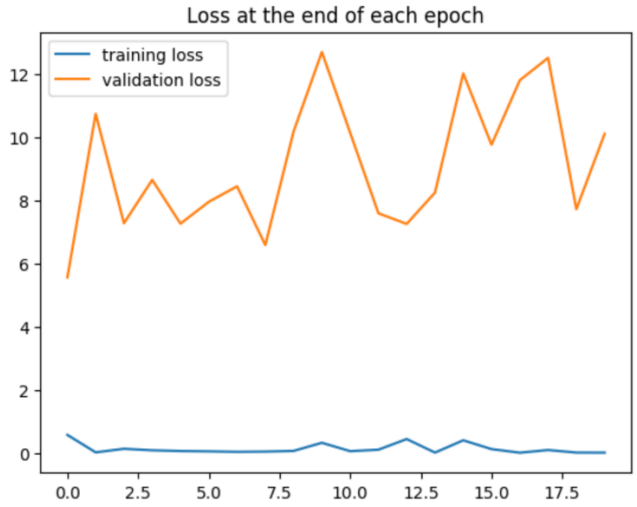

Fig. 6. Training loss and validation loss per epoch.

Some explanations for this behavior may be that the batch size or the learning rate may be too high, or the number of training epochs are too low. Alternatively, this may also be the result of the large class imbalance affecting loss and F1 in an adverse manner. This may require more specialized and powerful computer hardware to properly test and resolve.

A. Evaluation Metrics

Following the training, the best model in terms of F1-score was loaded from the PyTorch state dictionary containing the weights of that model. From there, we can run the dataset on the test set to find the model's best performance. From the loss graph, it appears that the best model within the 20 epochs happened around the 7th epoch. After setting the model to evaluation mode, predictions were made on the model and an evaluated based on the previous decided upon methodology.

TABLE III. Evaluation Metrics For Trained AlexNet Model

Metric	***Value***
Accuracy	0.8963
Precision	0.8894
Recall	0.8963
F1-Score	0.8899

Fig. 7. Accuracy, precision, recall, and F1 for the best trained model that was found.

From these results, we can see that the model performs quite well when it comes to our metrics. Achieving an 89% in this task seems to be quite impressive, especially given the age of AlexNet when compared to the benchmarks set by more modern and sophisticated models. Some initial concerns, such as the similarities between spiders and impact ejecta noted by the dataset publishers [1], caused some doubts, however the performance of the model overall can be considered high.

B. *Further Insights*

Some further insights can be acquired by analyzing a confusion matrix of the original dataset. Utilizing the Python plotting package seaborn, the predicted labels of the test set along with the true labels were packaged together into a heatmap for analysis. The resulting heatmap can be found below in Fig. 8.

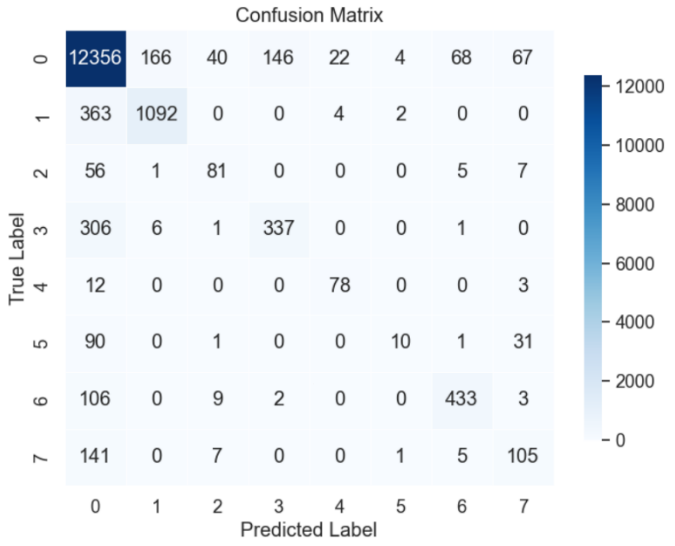

Fig. 8. Confusion matrix for the predicted results of the test set, by the best available model from this training session.

This led to some interesting findings regarding the performance of the model. Indeed, there were many correctly identified results, as indicated by the numbers going along the diagonal. However, the major issue of the dataset being mostly comprised of "Other" class images seems to present an issue with the model and its training. As noted by the high volumes of misclassified "Other" images, whether false positive or false negative, it appears that the severe class imbalance took its toll and the model may have issues generalizing its predictions. This may also explain the high values found by the weighted metrics, as the weight of the incorrectly identified labels was not enough to overcome the number of correctly identified "Other" labels. Interestingly, there were several images labelled "Other" that were mistakenly predicted as being members of the other labels, which may indicate that images in the "Other" category have some similarities to landmarks of interest.

Besides the misclassification issue of the "Others" category, a notable issue that can be identified is the number of misclassified impact ejecta sites that were mistakenly classified as spiders by the model, as noted by the box indicated by the true label of 5 and the predicted label of 7. This appears to justify the concerns of the original dataset publishers, as they noted that the radial trough design of spiders can resemble the impact ejecta. Interestingly, the model does not seem to have the inverse problem, as a higher proportion of spiders were correctly identified compared to the impact ejecta. This may be the result of there being marginally more spider examples than impact ejecta examples present in the data, resulting in the model having more exposure spiders. This may have caused the model to associate most radial designs with spiders instead of impact ejecta. More training epochs and examples may rectify this issue, as it appears that even human identification may also be challenged given that the publishers decided to include a mention of this information within the dataset description.

IV. Future Work

While this experiment demonstrates the potential for deep learning to be applied to investigating the Martian surface, there is plenty of room for improvement. Perhaps the most glaring issue is that of the "Other" class's overrepresentation within the dataset. Many issues in this study can be linked to this, such as the misclassification and prevalence of false positives and false negatives when it comes to the model classifying this data.

There are several options to solving this problem that can be explored in the future. Given that the original data sources only stems from 232 unique HiRISE source images and 10,815 cropped landmarks, this problem may simply require more data to properly be handled. Without simply increasing the amount of data, however, there are some options. Augmenting the minority classes, particular classes such as impact ejecta and spiders, would likely help balance the dataset and allow for better discriminatory power by the model. Other methods may include oversampling minority classes or under-sampling the "Other" class, however due to the constraints of the project the author was assigned, a larger dataset size to make use of big data techniques such as Spark was generally preferred.

More future work could be performed by experimenting with different model hyperparameters or architectures. AlexNet, while state-of-the-art upon its release, has many superior options when it comes to image classification nowadays, with new benchmark models coming out almost yearly. This experiment is intended as a proof of concept for future endeavors into applying deep learning to the data.

V. Conclusion

As we approach the dawn of a new space age for humanity, the necessity to understand the planets near us before we attempt any human exoplanetary exploration presents a unique opportunity and challenge for the scientific community to develop new methods for investigating the celestial bodies near us. With new technologies available to us through the work of scientific organizations such as NASA, we've been able to acquire a plethora of information regarding Mars using platforms such as the Mars Reconnaissance Orbiter. However, with new data comes the need to properly analyze it.

The dataset, created and published by researchers at the Jet Propulsion Laboratory at NASA, in collaboration with many researchers from numerous universities, contained both original and synthetic images of landmarks generated from source HiRISE pictures taken aboard the Mars Reconnaissance Orbiter, corresponding to a number of different classes.

Utilizing a combination of technologies such as Pandas, Torch, and Spark, image data from the HiRISE camera system about the Mars Reconnaissance Orbiter was able to be ingested, transformed, and analyzed with the use of deep learning techniques and software. This process began with Spark, where the landmarks dataset was ingested and preprocessed. This

involved resizing and normalizing the images, then saved into a pixel NumPy array for use with PyTorch.

The AlexNet model, chosen for its balance of training speed, performance, and robustness to image variation, was trained on the resulting landmark image data via CUDA-enabled GPU. Following the training procedure outlined earlier, this model was able to achieve high values in all four metrics chosen to evaluate this model: accuracy, weighted precision, weighted recall, and weighted F1-score, with the F1-score in particular scoring a total of 88.99%.

While the overall weighted metrics for this model are high, there are some concerns over the validity of the results, particularly in relation to the severe class imbalance of the dataset. Future work may focus on resolving some of these issues, such as further augmenting the dataset to generate more minority class examples or utilizing sampling methods to properly teach the model how to discriminate between various classes. While this attempt may be flawed, this experiment demonstrates that there is high potential for deep learning to play a crucial role in assisting us in helping us analyze the Martian surface, and perhaps even be pivotal in planning our next voyages out to Mars, or to the vastness of space beyond.

REFERENCES

[1] Gary Doran, Emily Dunkel, Steven Luand Kiri Wagstaff, "Mars orbital image (HiRISE) labeled data set version 3.2". Zenodo, Sep. 16, 2020. doi: 10.5281/zenodo.4002935

[2] "HiRISE | High Resolution Imaging Science Experiment," www.uahirise.org. https://www.uahirise.org/

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, May 2012, doi: https://doi.org/10.1145/3065386.