# Automated Planetary Terrain Mapping of Mars Using Image Pattern Recognition

**Requirements Document**
**Version 3.0**


**Team Strata:**

Sean Baquiro
Jorge Felix
Tsosie Schneider

# Table of Contents

## 1 Introduction

The goal for this capstone project is to create a computer program that allows human users to identify characteristic terrain types, train a neural network to recognize said terrain types by feeding examples to the algorithm, and map the characteristics across an entire set of images. This project will primarily focus on producing the automation of terrain types by using hi-resolution images of Mars. Some examples of the different terrain types and features that we are trying to identify include sand dunes, sinuous ridges, valleys, and canyons. The program should use multiple co-registered orbital data sets acquired from the instruments equipped on the Mars Reconnaissance Orbiter, the HiRISE and CTX camera. The images produced from HiRISE are in the JPEG2000 format and the CTX camera are in IMG format but can be easily converted into JPEG2000.

## 2 Problem and Solution Statement

The capstone project, Automated Terrain Mapping of Mars using Image Pattern Recognition, was proposed and is sponsored by Dr. Ryan Anderson. Dr. Anderson is a planetary scientist at the United States Geological Survey (USGS) Astrogeology Science Center. The Flagstaff Science Campus of USGS provide the ability to address science issues related to water, ecosystems, climate, energy and minerals, environmental health, and planetary exploration and study. More specifically, the Astrogeology Science center is involved in serving the nation, general public, and the science community by pursuing new knowledge about our solar system. The USGS Astrogeology Science Center has worked with space agencies like NASA to lead scientific investigations for numerous spacecraft missions. The goal of USGS is to strive for research that will allow us to better understand the origins, evolutions, and geologic processes operating on the celestial bodies from our solar system.

Dr. Anderson’s research involves geologic mapping and characterization of Mars’ locations. Specifically, his research has focused on Gale Crater, which is a crater located on Mars with various terrain types around the area. The main problem our sponsor faces is being able to efficiently map different characteristic terrains or features on an image of Mars. Some examples of the different terrains and features located on Mars include sand dunes, sinuous ridges, valleys, and canyons. The two main factors that make this a challenge for scientists are efficiency and time. It is possible to find similar patterns in a set of data by hand, but it will take the scientist a substantial amount of time. For example in one of Dr. Anderson’s research papers of Gale Crater, a couple of images with different identified terrain types were included and were manually mapped by Dr. Anderson himself. This took him a couple of months to properly map the different terrain types on an image and would have taken a skilled mapper a couple of weeks. Even when the processing of the data set is completed by hand, how efficient and effective was the scientist in detecting all the similar data in an adequate amount of time? Our task is to automate this processing by taking in an orbital data set with a terrain type of interest and running it through an algorithm that will detect similar data in an efficient amount of time. The sponsor will be providing an image with fully labelled terrain types that will serve as the basis for training datasets and terrain types to be mapped. Terrain mapping conventions used by sponsor will be maintained. While information produced by the algorithm will be written into the transparency layer of the provided image.

The key pieces of functionality for the planned system involve pre and post processing of a data set and using a Machine Learning algorithm to recognize patterns in the input orbital data. The system will use the C++ programming language to pre-process a Hi-res image of Mars in order to convert it into a more manageable data set.

The Hi-res images will be obtained from HiRISE (High Resolution Imaging Science Experiment). HiRISE is one Nasa's science instruments on the Mars Reconnaissance Orbiter. HiRISE is a camera that operates with telescopic lens that produce images at resolutions never before seen in planetary exploration missions.

Another place the input data will be collected from is the Mars Reconnaissance Orbiter Context Camera. Currently orbiting Mars, the Context Camera is acquiring grayscale images at 6 meters per pixel scale and over 30 kilometers wide. Currently images can be accessed through image catalogs on their respective websites and others sites such as NASA and USGS. In this case, the image will be converted into a byte array. The byte array will allow the Machine Learning algorithm to handle the data more efficiently. See Figure 1.

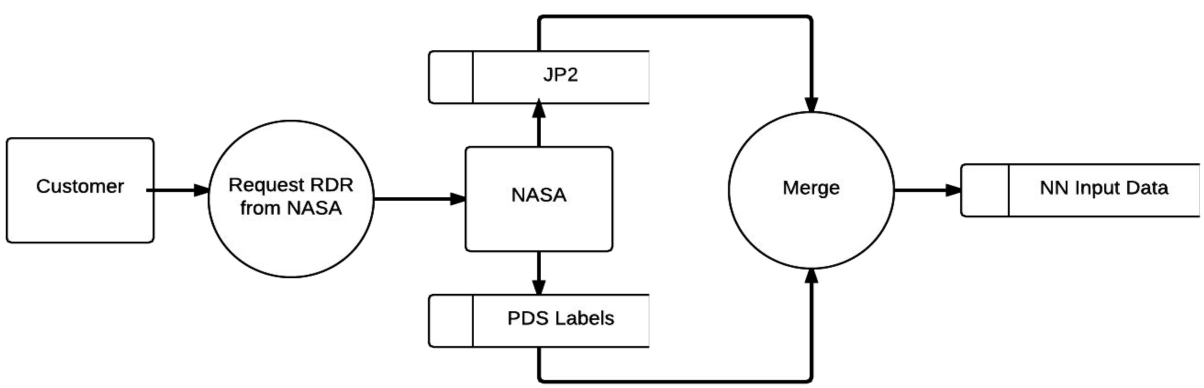


**Figure 1.** Reduced Data Record Acquisition

Once the image is pre-processed, a machine learning algorithm will handle the identification of all the similar patterns in the data. We will be using a Neural Network algorithm with supervised learning. By identifying an area of interest in the input data, the algorithm will be able to find similar sections throughout the whole image. When an area of interest is found, the algorithm will mark it or change the pixel value in order to identify it. After the data is processed in the algorithm, we will once again use C++ to process the byte array into a Hi-res image, Figure 2. The user will then be able to clearly identify the similar patterns they were interested in the image. This will allow scientists and researchers to quickly identify terrains or other areas of Mars they are interested in.

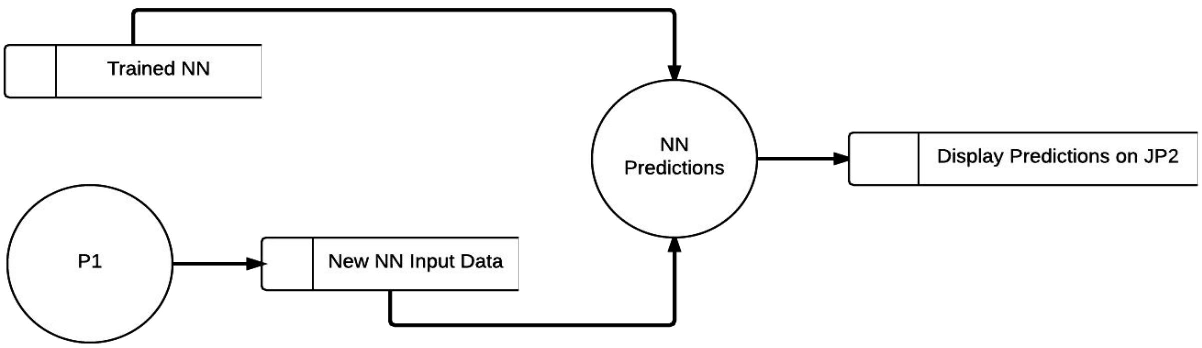


**Figure 2.** Neural Network Prediction Overview

The system design is intended to allow rapid identification of characteristic terrain types to speed up scientific research. It will not take the place of a human scientist when making geologic maps, but it could be used to rapidly assess how widespread certain features or terrain types may be. The sponsor for the project, Dr. Anderson and his organization USGS, will be able to use the automated system to research interesting features of Mars and even other planets and learn more about a planet's formation through geological processes. This system could also be used with terrestrial remote sensing data sets(e.g to identify water, vegetation, sand dunes, and other terrain types).

Alternatives to solve the problem were considered, however the decision to use a convolutional neural network  was made based on intensive research. Support Vector Machines were considered to handle the data manipulation of the Hi-res image. A Support Vector Machine can also be used to analyze data and recognize patterns. The main reason in choosing a Convolutional Neural Network algorithm over a Support Vector Machine was due to more available research. The field of Machine Learning is quite large and investigating all possible algorithms is beyond the scope of this project. Instead, our team will focus on Neural Network algorithms, which themselves represent a large sub-field of Machine Learning. Currently there's more available research showing the effectiveness of using a Neural Network in image pattern recognition problems and by using a Neural Network algorithm we will be able to solve our problem of identifying terrain types of Mars effectively and within time constraints.

# 3 Functional Requirement

## 3.1 HiRISE and CTX

The system shall use the data sets provided by the HiRISE and the CTX instruments as the primary input for the tool. These datasets are provided to the public as JPEG2000 (JP2) images. As a consequence, any pre-processing of the images will require adherence to JP2 specifications.

**a.** The HiRISE (High Resolution Imaging Science Experiment) provides high resolution (25 cm/pixel) images of Mars' surface.
**b.** The CTX (context camera) provides a more general overview of the surface at 6m/pixel.

## 3.2 Mars Orbital Datasets

The system shall allow users to be able to load and georeference multiple Mars orbital datasets, identify terrain types and features, and automatically map similar features across multiple input images.

## 3.3 Image Mapping

The system shall use HiRISE and CTX to present three Reduced Data Record (RDR) products conforming to Planetary Data System (PDS) standards.

**a.** HiRISE consists of two distinct RDRs. The first being a single-color RDR created from red-filtered cameras with a pixel width of 20,048. The second is a three-color product which is a composite of red, blue-green, and near-infrared cameras with a pixel width of 4,048. Image data shall be combined to maximize available dataset.
**b.** In contrast to HiRISE, the CTX camera has a comparably lower resolution.This necessitates a many-to-one and one-to-many mapping algorithm to associate pixel information between the HiRISE and CTX RDRs.

## 3.4 Artificial Neural Network

Due to the size and complexity of input RDRs, the system shall use an Artificial Neural Network (ANN) in order to minimize time required to map terrain types. A Convolutional Neural Network (CNN) which is the neural network often used for image recognition and that served as the inspiration for this project. [3]

CNNs are multilayer perceptrons that mimic visual perception by overlapping information segments. This also minimizes the need for image preprocessing. The ability to supervise training of the neural network layer by layer has made the CNN the prime candidate to implement neural network functionality required by the sponsor.

### 3.5 Training Datasets

CNNs are a type of feed-forward neural network, though powerful, they require relatively large datasets to adequately train the neural network on. To circumvent this issue, the system shall need tools to generate realistic and reliably labelled training datasets as the initial training dataset provided by the sponsor will be small. Mimicry of training datasets will shall utilize the following techniques stochastic downsampling, orientation inversion, and image reorganization.

### 3.6 Image Post-Processing

The system shall use side channel spatial information such as transparency and alpha planes, which are useful for transmitting information for processing the image for display, print, or editing. This will be utilized to maintain neural network predictions inside the output RDR, such as marking the area of interest in the transparency plane, without compromising the input RDR.

### 3.7 Algorithm Performance

The system shall be able to assess how well the algorithm performed by comparing the correctness of the identified terrain with the results of a human-generated map.

### 3.8 Robustness

The system shall be designed to handle error conditions easily, without failure. Such conditions include a tolerance of invalid datasets (non HiRISE/CTX produced data sets), unexpected operating conditions, and overfitting within machine learning algorithms.

## 4 Environmental Requirements

### 4.1 Operating System

The required operating systems the computer program shall run on are Windows and Linux.

### 4.3 C++

C++ is an object-oriented language and an extension of C. It is an intermediate language that contains features from low and high level languages. Both C and C++ are supported with many libraries that involve numerical and scientific analysis.

C++ will be used to preprocess the image and create a more manageable data set to use in a Neural Network algorithm. C++ will also handle the post processing of the image and convert the initial processed data back into a Hi-res image with the new data. C++ provides the extreme computational efficiency which is required for pattern understanding. Optimization can be nicely implemented with object-oriented techniques.

### 4.4 Python and Pycharm IDE

The Neural Network algorithm that will handle the Image Pattern Recognition will be implemented in Python.

Python is a high-level, interpreted, object-oriented programming language that is developed under an OSI-approved open source license. The open source license allows users to use and distribute it freely. Python is a powerful and fast language that can be incorporated well with other languages and systems. Python syntax is easy to learn and read, which in turn can reduce the cost of program maintenance.

One of the biggest advantages of Python is its support of packages and modules. The Python Package Index hosts thousands of third-party, open-source modules that involve web development, database access, desktop GUIs, scientific and numeric, education, network programming, and many more other packages. The use of packages will be decided during the coding and development phases of our project and python packages will be incorporated as needed.

The IDE our team will be using for Python during the development of this project will be Pycharm. Unlike other text editors that only do syntax highlighting, the Pycharm IDE is aware of all of Python’s language rules and code structure. Pycharm also comes with many useful features that include code navigation, autocomplete, integrated debugging, integrated unit tests, clickable stack traces, and many more. The built-in source level debugger allows programmers to inspect the code, stepping through it line by line to find the source of the error.

### 4.5 OpenCV

OpenCV is an open source machine learning software library. The library contains more than 2500 optimized algorithms that can be used to detect faces, identify objects, track camera movements, stitch images together, and many others. It contains interfaces for C++, C, Python, Java, and MATLAB and is supported on Windows, Mac OS, and Linux.

OpenCV will ease the interaction between C++ and Python and allow us to incorporate machine learning algorithms into our program. Specifically from the Machine Learning library in OpenCV we will be focusing on using Neural Network classes.

### 4.6 Boost.Python

Boost.Python is a C++ library that enables ease of interaction between C++ and Python. The library includes support for C++ to Python exception translation, manipulating Python objects in C++, and exporting C++ iterators as Python iterators. It allows developers to easily expose C++ class functions and objects to Python as well as the other way around. This library will help us incorporate both languages into the same project

## 5 Non-Functional Requirements

### 5.1 Open Source

The computer software shall be open source and available through a Github repository. The specific license is yet to be determined.

### 5.2 Availability

The system's availability, or "uptime", shall be constant and consistent while the user uses the program. There should be no activities in the system that causes expected downtime and the program should handle all error conditions.

### 5.3 Github Repository

Github is an online Git repository service offering all of the distributed version control and source code management. Github also provides access control and collaboration features which include bug tracking, feature requests, and task management.

The source code for our project shall be available through Github initially maintained by Team strata. Once the Image Pattern Recognition program is completed, the Github repository will be transferred over to the sponsor Dr. Ryan Anderson.

# 6 Potential Risks

## 6.1 Performance

The sheer size of an unoptimized neural network can represent a challenging computational burden, even on modern CPUs. Because of this, powerful GPUs are commonly used instead to train and run the networks and can speed up common operations such as large matrix computations. Desktop's with strong GPU's are available for testing during the development of the system. Using a convolutional neural network within the software can be a risk to the customer if not properly optimized. We aim to have the program produce results within days.

## 6.2 Training Datasets

In supervised machine learning, the dataset size needed to train a neural network will vary. With smaller training sets there will be tradeoff of bias and variance, while larger training sets require more processing time and has a greater risk of overfitting, but allow for lower bias and variance. Overfitting occurs when the system describes a random error instead of the underlying relationship. A overfit model will generally have poor predictive performance. Since the initial training set will be provided by the sponsor there is a potential that it will be too small to train any neural network without introducing unacceptable variance.

## 6.3 Team Expertise

There are key risks to consider when planning the development process of the system. Not all team members are fully familiarized with the required programming languages we will be using. The impact of not knowing the programming languages can cause major delays in our schedule. To prevent this impact, we will set some time to familiarize ourselves with the languages in order to be prepared when the coding portion of the project is started. Not being able to meet/communicate as a team can cause schedule delays and miscommunications among team members. In addition, miscommunications can also cause team members to make assumptions when developing the program. In order to prevent miscommunication problems, our team has set a time to meet weekly and discuss any design and programming decisions.

# 7 Project Plan

The project plan will be used to provide an overview of how and when the project's objectives will be achieved. Included in this section are team roles and the project schedule. In addition to the roles detailed below, each member will also take on the role of programmer. As developers, each of us will be responsible for writing the code of the project.

## 7.1 Team Roles

| | |
|---|---|
| **Role** | Team Leader, Software Architect, and Developer |
| **Member** | Tsosie Schneider |
| **Description** | The Team Leader role is intended to be administrative in nature, coordinating the effort of team members to maintain acceptable progress on the project. The Architect role is ensure that fundamental design decisions and functionality are maintained. |

| | |
|---|---|
| **Role** | Release Manager and Developer |
| **Member** | Jorge Felix |
| **Description** | The Review/Release Manager will maintain document standards and safeguard the github repository throughout the project. |

| | |
|---|---|
| **Role** | Recorder and Developer |
| **Member** | Sean |
| **Description** | The Meeting Recorder will take notes during meetings to maintain a record of project decisions and to be used as a primary source to mediate conflicts and misunderstandings. Records of meetings and project decisions will be kept by the recorder in a Google Drive folder and updated weekly. |

## 7.2 Project Schedule

Below is a project schedule table for the project that contains all the tasks we will complete during the full development cycle. Our tasks are divided into four current categories that include Research/Design, Coding, Testing, and Documentation.

The first semester will focus on the research and design of the project. During this task we will be required to write at least two versions of a requirements documents that will serve as a contract between Team Strata and the sponsor.

The coding task will involve three phases and will occur during the second semester of capstone. In the first phase our team will focus on implementing the pre-processing of the Hi-res input images. Once phase one is complete we will move onto phase two where we will implement the Neural Network handling the main data transformation. The Neural Network will be in charge of finding the similar patterns in the image by using the marked input with the area of interest. Phase 3 will involve the implementation of post-processing the data back into an Image. The third task involves completing all the testing on the program developed by our team. Task 3 will involve two phases where the first phase completes all of the preliminary testing on the first version of the program. In phase 2 of testing, the testing will test on the final program that is ready to be released.

Once the program has been developed and completed, the documentation task will be the final step to completing the project. In the documentation task Team

Strata will write a full documentation describing the components of our program and provide a user guide for the sponsor.

| **Automated Terrain Mapping of Mars Project Schedule** | | | | |
|---|---|---|---|---|
| **#** | **Task** | **Start Date** | **Duration(Days)** | **End Date** |
| **1** | **Design/Research** | **9/17/15** | **84** | **12/10/15** |
| **1.1** | **Team Standards** | **9/17/15** | **7** | **9/24/15** |
| **1.2** | **Team Website** | **9/25/15** | **14** | **10/8/15** |
| **1.3** | **Tech Feasibility** | **10/15/15** | **14** | **10/29/15** |
| **1.4** | **Requirements Draft** | **10/29/15** | **7** | **11/5/15** |
| **1.5** | **Design Review I** | **11/5/15** | **14** | **11/19/15** |
| **1.6** | **Requirements Final Draft** | **11/19/15** | **21** | **12/10/15** |
| **1.7** | **Update Team Website** | **12/10/15** | **7** | **12/17/15** |
| **2** | **Prototyping** | **12/18/15** | **28** | **01/16/16** |
| **3** | **Coding/Development** | **01/19/16** | **60** | **03/18/16** |
| **4** | **Testing** | **03/18/16** | **21** | **04/08/16** |
| **5** | **Documentation/User Guide** | **04/08/16** | **21** | **04/24/16** |
| **6** | **UGRAD Presentation** | **4/29/16** | **1** | **4/29/16** |
| **Automated Terrain Mapping of Mars Project Schedule** | | | | |

## 8 Glossary

**CTX**
The Context Camera provides grayscale images with a pixel resolution up to 6m. The camera is designed to provide context maps for the targeted observations of HiRISE.

**Data Set**
An accumulation of data products.  A data set together with supporting documentation and ancillary files is an archive.

**Georeference**
To geo-reference is to define location in physical space and is crucial to making aerial and satellite imagery useful for mapping. Geo-referencing explains how position data (e.g., Global Positioning System (GPS) locations) relate to imagery and to a physical location. [4]

**HiRISE**
The High Resolution Imaging Science Experiment is a camera equipped on the Mars Reconnaissance Orbiter. The camera consists of a 0.5m aperture reflecting telescope which allows it to take pictures of Mars with resolutions of 0.3m/pixel.

**JPEG 2000 (JP2)**
A Image compression standard and coding system. Superseding the original discrete cosine transformed-based JPEG standard, JPEG 2000 offers a modest increase in compression performance compared to JPEG. The codestream of JPEG 2000 can be decoded in a number of ways, including, obtaining representation of images at a lower resolution, or signal-to-noise ratio.

**Mars Reconnaissance Orbiter**
The MRO is a spacecraft launched August 12, 2005, that acquires orbital observations during the mission's primary mapping phase. The spacecraft arrived at Mars orbit insertion in March 2006. It is equipped with a remote sensing instrument called the High Resolution Imaging Science Experiment (HiRISE).

**Overfitting**
In machine learning, overfitting occurs when a model is excessively complex, such as having too many parameters relative to the number of observations.

**Planetary Data System (PDS)**
Is an archive of data products produced by NASA planetary missions. The system utilizes its own set of standards designed to describe and store data for future scientist that are unfamiliar with the initial experiments. [2]

**Reduced Data Record (RDR)**
An image that has been radiometrically-corrected and resampled to a standard map projection, then stored as a JP2 image. [1]

**References**

[1] Eliason, E., Castalia, B., Becker, K., Anderson, J., Sides, S., Software Interface Specification for HiRISE Reduced Data Record Products, May 15, 2007, *University of Arizona and United States Geological Survey.*

[2] Knopf,W., About the PDS, *PDS: The Planetary Data System*, https://pds.nasa.gov/about/about.shtml.

[3] L. F. Palafox , A. M. Alvarez , C.W. Hamilton , Lunar and Planetary Laboratory, 2015, *University of Arizona,* http://www.hou.usra.edu/meetings/lpsc2015/pdf/2316.pdf

[4] Section 3.7: Geo-Referencing, 1-3, https://www.uscg.mil/hq/cg5/cg534/nsarc/CISAdd2-0-Geo-Ref.pdf , *United States Coast Guard.*