Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project
Project Plan

Deep Learning Algorithms Applied to Surface Mapping of Mars

by

Niranjana Sundararajan

Email: niranjana.sundararajan21@imperial.ac.uk
GitHub username: acse-ns1321
Repository: https://github.com/ese-msc-2021/irp-acse-ns1321

Supervisors:

Dr Cédric M. John
Dr Philippa J. Mason

August 2022

Abstract

An accurate understanding of Martian Terrain and its features is crucial not only to further scientific research in the fields of planetary science but also essential to engineers responsible for planning rover landings and building models for landing site traversability analysis. While current tools and research provide for supervised classification of landforms and features based on known/suspected terrain classifiactions,these often fail to acknowdlege the possiblilty that Martian terrains may exhibit features that are significantly different from our present understanding of terrains. Using the Mars Reconissance Orbiter's HIRISE images that have a resolution of 25cm/pixel, this project focuses on first create a publically available API package to download the images and extract hidden patterns and features in the images using a deep learning based encoding algorithm such as a convolutional neural network or an autoencoder.Consequently the images are categorised using a clustering algorithms such as K Means and DBSCAN based on similarity of their derived features. Therefore this model will provide researchers and engineers alternative ways to study unique features in Martian Terrains.

1 Introduction

The Martian Terrain, much like earth, possesses a multitude of landscape features. The United States Geological Survey has divided the Martian land into thirty quadrangles, each defined by a specific latitude and longitude. NASA/JPL/MSSS have created a map shown below:

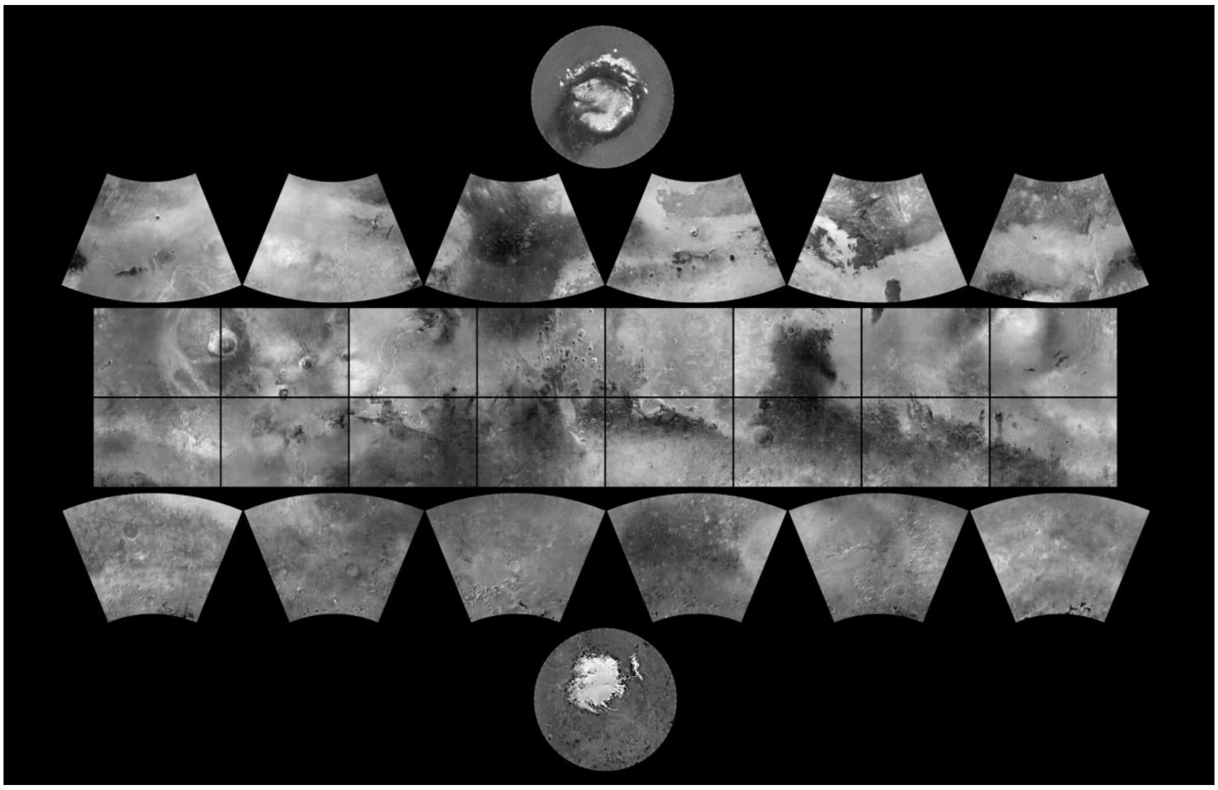
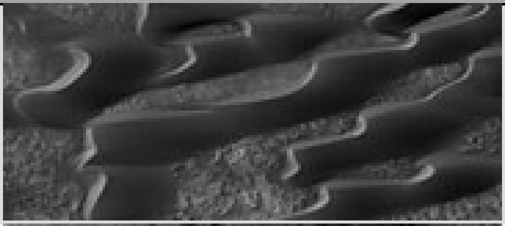
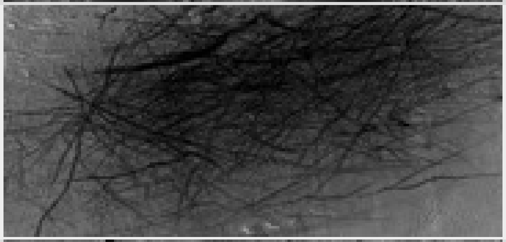
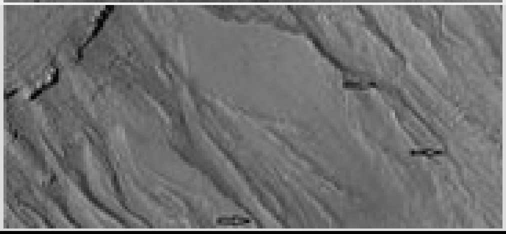
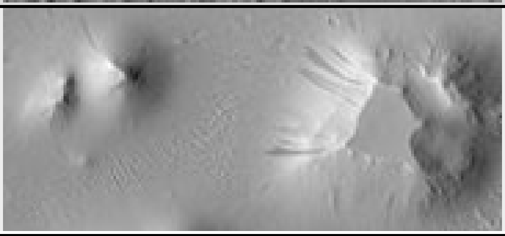
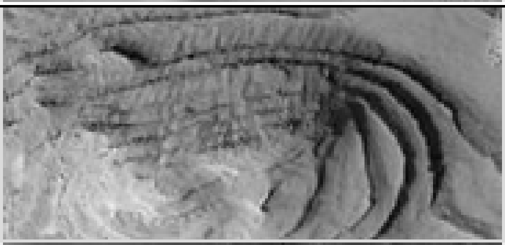
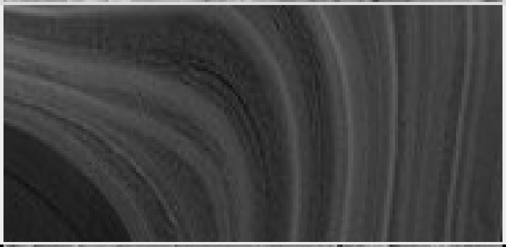
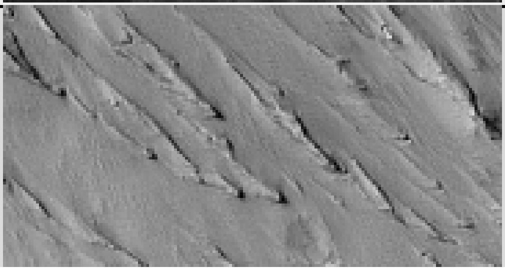
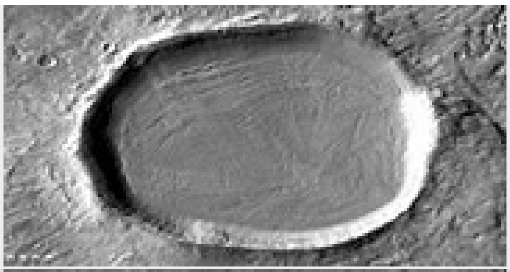
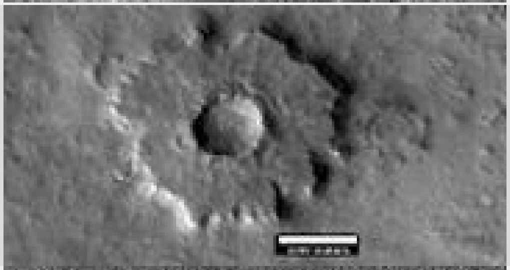
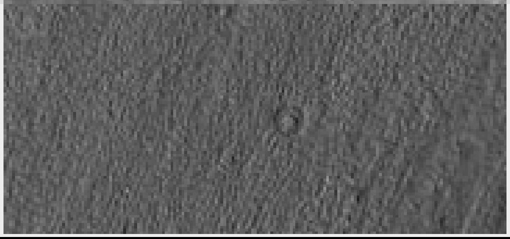
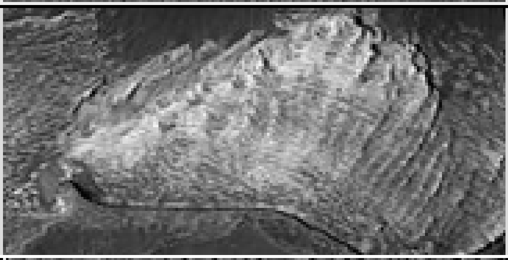
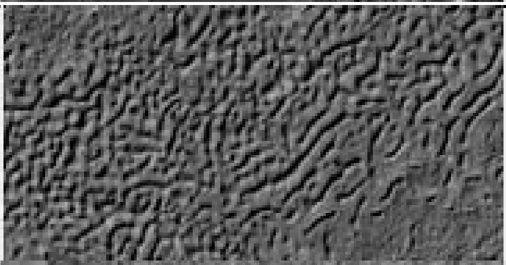
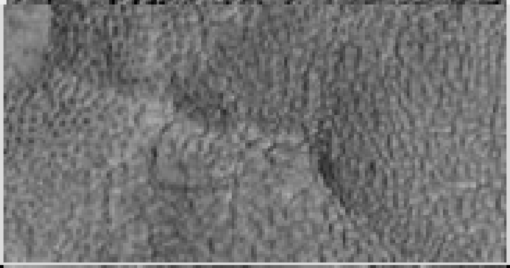
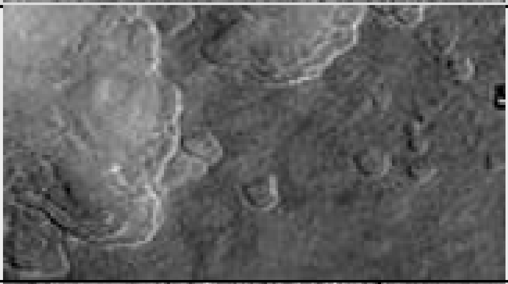
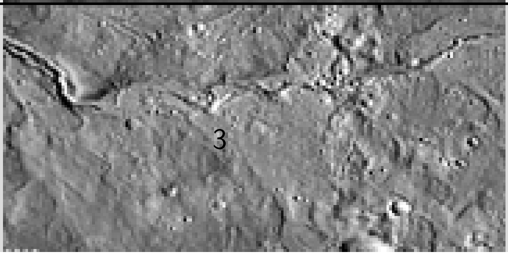


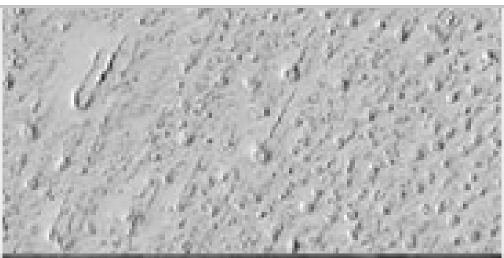
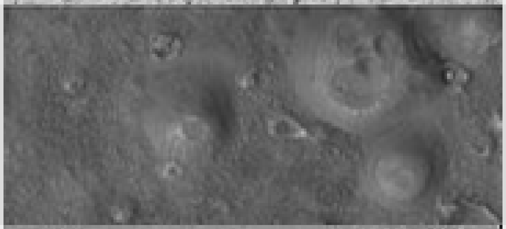
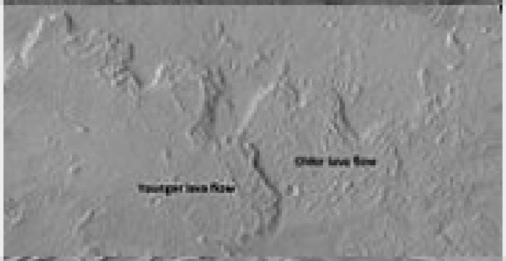
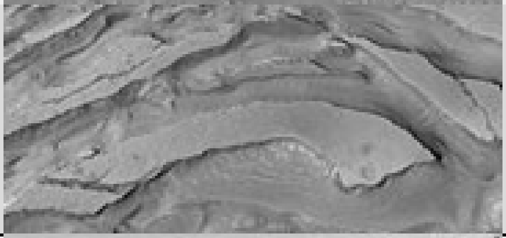
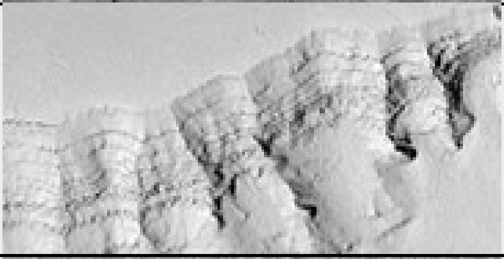
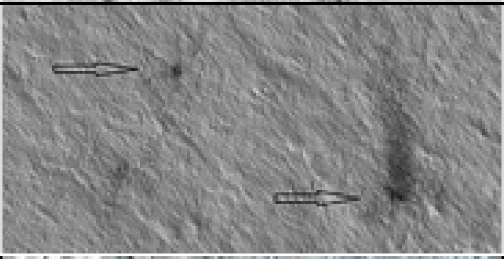
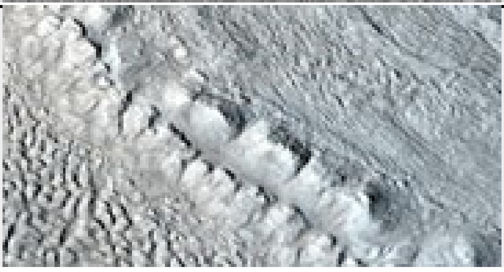
Figure 1: The MGS MOC Wide Angle Map of Mars. Source : NASA/JPL/MSSS

Some of the characteristic features of the terrain include volcanos, canyons, impact craters and dry lake beds. A brief study of the main features studied in preparation for this project based on availabe information are summarized in the tables that follow:

Presently, there are mainly two types of research objectives that are considered while building deep

	Landform Name	HIRISE Image (from HiWish)	Location and Features
1	Sand Dunes		Many locations mostly in the northern polar caps. Including Eridania quadrangle, Mare Tyrrhenum quadrangle
	Dust Devil Tracks		Multiple areas – constantly changing including in the Noachis quadrangle
	Martian Gullies		Multiple Locations- along craters. Including in Phaethontis, Thaumasia, Diacria, Mare Acidalium, and Eridania quadrangle
2	Recurrent Slope Streaks		Multiple locations – along craters and gullies
3	Layers In the Sandy Terrain		Margaritifer Sinus quadrangle, Aeolis quadrangle, Memnonia quadrangle
	Layers In the Polar Ice-caps		The Northern Ice Cap
4	Linear Ridges (Yardangs)		Restricted regions mainly in the Amazonis quadrangle of the Medusae Fossae Formation

5	Craters : Concentric Craters		Phaethontis quadrangle, Casius quadrangle
	Craters: Pedestal Crater		Casius quadrangle, Amazonis quadrangle
	Craters : Ring Mould Craters		Mostly Ismenius Lacus quadrangle
6	Chaos Terrain		Multiple locations. (Left is Eos Chaos, others include Ister Chaos, Aureum Chaos, Arsinoes Chaos, Iani Chaos, Margaritifer Chaos etc)
7	Mantle : Brain Terrain		In the "Upper Plains Unit" in the Northern hemisphere the Ismenius Lacus quadrangle
	Mantle : Polygonal Patterns		Some regions including in the Casius quadrangle, Ismenius Lacus quadrangle
8	Scalloped Depressions		Multiple Locations mainly between 45° and 60° north and south
9	Dry Water Channels		Multiple loactions – mostly in craters and lake beds – including in Memnonia quadrangle, Cebrenia quadrangle

10	Volcanic Cones		Equatorial Regions
	Mud Volcanoes		Mostly the Northern Hemisphere
	Laval flows		Multiples Locations – including Tharsis quadrangle, Phoenicis Lacus quadrangle and Amazonis quadrangle
	Volcanos below Ice		Multiples Locations – mainly in the Ismenius Lacus quadrangle
11	Valleys : Noctis Labyrinthus		Phoenicis Lacus quadrangle
12	Spiders and Plumes: Dark channels formed after defrosting		Appear in winter across multiple locations
13	Glaciers		Restricted areas mainly in the Ismenius Lacus quadrangle

learning models involving martian terrains.
In the first kind, researchers have used automated image analysis and deep learning techniques for the detection of specific terrain features such as dunes fields [?], craters [??], volcanic cones [?] and aeolian ridges [?]. In the second, researches have used deep learning algorithms to build supervised algorithms that classify availabe images into defined classes. The Soil Property and Object Classification [?] software successfully classifies full resolution HIRISE images into 17 terrain classes using a specialized Fully Convolutional Neural Network.

A major shortcoming in these generalized classifiers using convolutional neural networks is that these models, that are supervised by nature, require images to be manually labelled before training. Therefore considering the total data volume and the abundance of new images, even the recent, more accurate classifiers such as those deployed within the Planetry Data System's Imaging Atlas [?] will likley fail to account for newly captured features.

2 Problem Description and Objectives

This project aims to build a classification model that takes in as input a set of preprocessed HIRISE images, extract and encode the features and image information and finally using clustering methods separate them into an appropriate number of unlabelled classes. The schematic of the proposed model stages can be seen below :

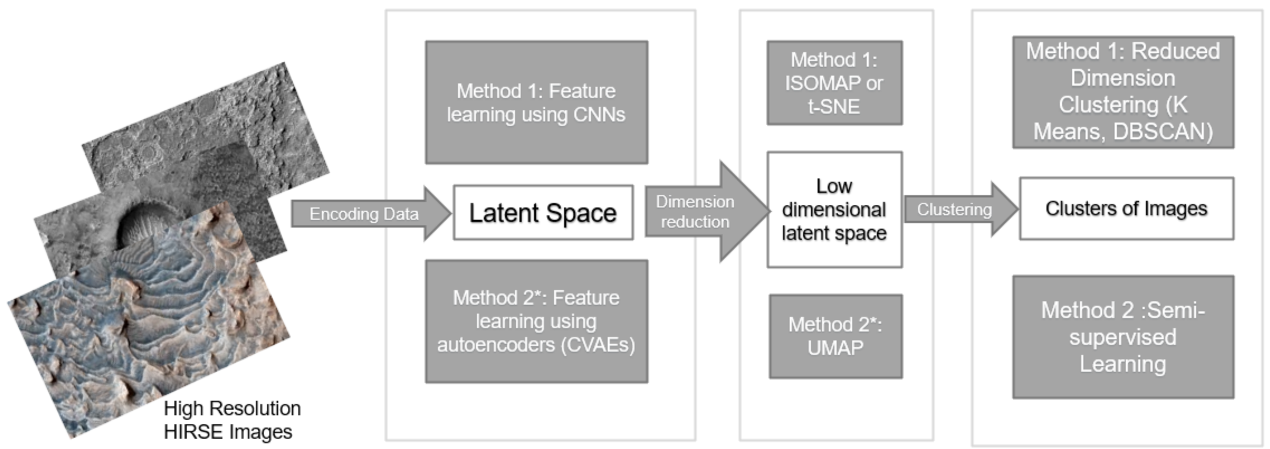

High Resolution
HIRSE Images
Encoding Data
Method 1: Feature learning using CNNs
Latent Space
Method 2*: Feature learning using autoencoders (CVAEs)
Dimension reduction
Method 1: ISOMAP or t-SNE
Low dimensional latent space
Method 2*: UMAP
Clustering
Method 1: Reduced Dimension Clustering (K Means, DBSCAN)
Clusters of Images
Method 2 :Semi-supervised Learning

Figure 2: The Proposed Model Development Schematic

The **first objective** in building this model is the injestion of the HIRISE images and the associated metadata. The proposed solution is the building of a python package designed based on Object Oriented Programming principles that allows for easy queying and downloads. The tool allows the user to webscrape the necessary file hierarchies from the websites as well as create an updatable database that holds all the currently available information. Building this tool has been the focus of the project so far.

The **second stage** and an important objective of this project is to find a deep learning algorithm that allows for appropriate encoding of the image data, such that the features are preserved and feature selection can be executed with accpetable results while decoding. Considering that the images in the HIRISE database range from 1Gb to 87Gbs, it is essential that this encoding not only captures features correctly but also allows for dimension reduction after latent space generation. Of the encoding algorithms available(including convolutional neural networks), autoencoders, specifically the convolutional variational autoencoder(CVAE), due to their stability and ability to retain spatial information show great promise for this purpose. While using a regular autoencoder has shown to produce good results when used for pre-training on Martian Terrains[**?**] as well as its proven suitability for dimension reduction purposes[**?**], it will be interesting to observe how much better a CVAE performs on these images.

Once the high dimensional latent space is generated, the **third stage** is to use projection based dimension reduction techniques such as Isomaps, t-distributed stochastic neighbour embedding(t-SNE) or the most approprite a Uniform Manifold Approximation and Projection(UMAP) that has demonstrated structure preservation on Mars images[**?**]. UPMAPs, in contrast to t-SNEs, compresses the high dimensional latent space instead of converting it point by point. This makes the process more computationally efficient.

The **fourth stage** in building the model is the selection of a clustering algorithm that will classsifiy the reduced and encoded latent space of images. The two possible operations include a reduced dimension clustering method such as K Means or DBSCAN that has previously shown good results for high resolution remote sensing images [**?**], or a semi-supervised learning algorithm that uses transfer learning, which is a pretrained network to group images into clusters.

In summary, the potential challenges that are likley to be encountered besides the preprocessing of the data are the selection of encoding methods, selection of a appropriate clustering algorithms and choosing the most appropriate performance metrics. More importantly, a special consideration is to be given for ease of use of the package and the reproducibility and scalabilty of the code, which will not only allow future researchers to use the existing tool for future classifications but also allow researchers to expand the tool's use to support their own research.

3 Progress to Date and Future Plan

HIRISE is a growing databse of images - presently populated with over 73,000 images and over 25 parameters that define each image. To be able to consume this data it was imperative to build a tool that allows for easy access to required images and their associated metadata. The Planetary Data System(PDS), maintained by NASA, while allowing the user to browse the catalogues of available information requires the user to manually navigate to specific folders and download single images. This is tedious and time-consuming.
This project implements a package that allows for these functions. It currently includes 32 querying functions, with additonal functions that allow for downloading the current database, updating the current database with new information from the PDS and downloading filtered/specified images directly. It also includes a package for preprocessing the image data that has functions such as tiling, resizing to zoom in/out on images (staring from 25cm/pixel to 250+ cm/pixel), loading the pytorch dataset, converting to grayscale etc.

The aim is to build the software using agile methodologies and lean software development. Based on the agile manifesto the following are a few practices that were observed and implemented :

- **Continuous Delivery**: In order to obtain frequent feedback and facilitate continuous improvement multiple verions of the software were released during the development process. Presently there are 28 different verisons availabe on the pypi test wesbite with a progressive improvement and bug fixes noted with each version,

- **Problem Breakdown**: The software was developed first by planning the overarching architectture of features and functions required, then individual classes, their associated functions and utility modules were planned and executed in stages. The code undergoes refactoring and optimisations when the logic is complete
- **Measuring Progress** : Planning and progress is meaured through GIT's project board. This a Kanban Style board developed by git to aid lean software development by allowing for task breakdown, distribution, progress evaluation, commenting and feedback.

The next steps include expanding on the preprocessing package with functions such as cropping preprossed images, splitting the data into train and test sets taking into account spatial correlation, encoding using CVAEs, dimensionality reduction using UMAPS etc. These functions can then access the hirise image objects which will serve as inputs to these functions.
The step following the implementation of the preprocessing package will be another class associated with training that will perform the clusering based on the algorithm chosen. Finally based on the specifics of the implemented algorithm there will be a final class that calculates accuracy metrics and provies the user with information on how well the algorithm has performed.

The gantt charts below demonstrate the planned implementation of the different stages of the project for the coming months:

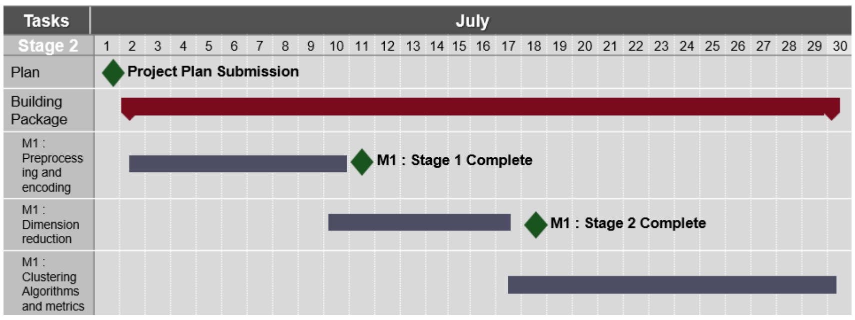


Figure 3
Proposed Gantt Chart for July

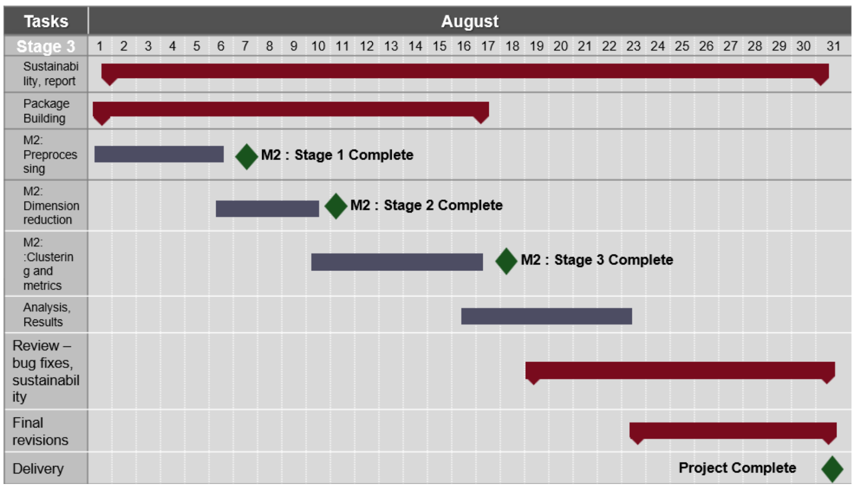


Figure 4
Proposed Gantt Chart for August