Enhancing Martian Surface Evaluations by Applying Multi-Task Machine Learning Algorithms to Satellite Images

MSc Research Project
Data Analytics

Daniel Murphy
Student ID: x20138164

School of Computing
National College of Ireland

Supervisor: Jorge Basilio

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Daniel Murphy
Student ID:	x20138164
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Jorge Basilio
Submission Due Date:	15/08/2022
Project Title:	Enhancing Martian Surface Evaluations by Applying Multi-Task Machine Learning Algorithms to Satellite Images
Word Count:	XXX
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	14th August 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Martian Surface Evaluations by Applying Multi-Task Machine Learning Algorithms to Satellite Images

Daniel Murphy
x20138164

Abstract

Planning for manned missions to Mars in the near future is already well underway. However, the Martian surface topography is extremely complex and hazardous and requires accurate, detailed maps if these missions are to be successful. Various deep learning approaches are effective at mapping the surface for individual targets, but lack the ability to evaluate regions on multiple features. This research proposes a novel multi-task deep learning CNN to evaluate Martian regions based on two features: terrain classifications and crater detections. Three such multi-task model architectures, soft, firm and hard parameter sharing, are designed and compared to established single-task models. While the single-task model was found to outperform the multi-task for terrain classifications with recall and precision values of 41.95%, the multi-task model was found to have superior precision and F_1 scores (5.52% and 2.15%, respectively) in crater detection. Hence, the novel approach allows regions to be evaluated on multiple parameters instead of single. Future work to improve the presented models will add more classification tasks to eventually be able to evaluate a given region across all relevant characteristics.

1 Introduction

With space agencies and now private companies around the world investing in future manned-missions to the planet, Mars has become the next major milestone in space exploration. However, there remain persistent challenges which will need to be overcome before successful human missions can become a reality. Of primary concern, is the evaluation of the Martian planetary surface, which has been observed as extremely difficult to navigate or traverse, with numerous impact craters, gullies, deep valleys and large mountains present in abundance. The destination of any manned missions require the most detailed maps containing evaluations of surface regions considering all of these features.

Fortunately, since Mars has been a subject of scientific study for decades, there is an enormous catalogue of data collected by satellites readily available to the public and to researchers. Machine learning, as an efficient means of analysing vast quantities of data, has become commonly used in the evaluation of surface regions on Mars. Particularly, deep learning algorithms are especially effective in identifying and classifying these surface features and terrain types. However, considering the requirements for future manned missions, regions need to be evaluated across a number of factors such as terrain type and also surface feature classification. The current deep learning approaches are unable to evaluate given regions based on multiple factors and this limits their application to the mapping and analysis of Martian surface regions.

1.1 Research Question & Project Objectives

This research aims to advance the current set of deep learning techniques being applied to Martian surface mapping efforts by expanding their classification abilities to facilitate the evaluation of surface regions on multiple classification tasks. In particular, the classification tasks to be addressed are crater detection and terrain classification in images. As such, the research work aims to address the following research question:

- ***Research Question:*** *How can multiple rather than single classification techniques be used in machine learning models to better evaluate Martian surface regions?*

Based on the stated research question, the following key objectives have been identified:

- Design and implement a multi-task machine learning model capable of producing multiple classifications (craters and terrain types) based on a single input (Martian surface satellite images).
- Compare the multi-task machine learning model performance against single-task models based on relevant performance metrics.

2 Related Work

2.1 Introduction

In this section, a review of the related work to date is presented, focusing on the application of machine learning to planetary surface mapping. Each of the following subsections concentrate on this application in the contexts of planetary landform classification, terrain analysis and finally multi-task classifications. The review aims to identify and critically analyse the most relevant related research, with an emphasis on the identification of novel improvements in order to determine the most suitable methods to best address the research question. The literature survey scope is restricted to well-cited, peer reviewed articles published within a ten-year period from 2012-present.

2.2 Landform Classification

Traditionally, the identification of landforms on the surfaces of extra-terrestrial bodies such as the Moon or Mars was a labour-intensive and time exhaustive task which required much attention from domain experts. However, in recent years researchers have demonstrated the potential of machine learning to automate and enhance this task.

In a recent article, Rajaneesh et al. (2022) classified extra-terrestrial landslides, comparing a number of relevant approaches in the process. While the authors found the simple logistic model outperformed the others with a classification accuracy of 81%, they also reported strong performance in a number of other models including sequential minimal optimisation, meta classifier and the multi-layer perceptron which all reported accuracies above 77%. This appears to suggest there are a variety of models capable of accurately classifying this landform. Notably, Wang et al. (2017) classify the same landform but use the Adaboost model and achieve a superior detection accuracy of 92%. Furthermore, Wang et al. (2021) goes on to further research the classification of this landform in a more recent study where a number of models are compared, similarly noting strong performance of the logistic model as well as the boosting models, support vector machines (SVM) and random forests, but most significantly finds the convolution neural network (CNN) algorithm significantly outperforms all other considerations with an impressive classification accuracy of 92.5%.

Another of the most frequently classified landforms is the surface crater. Arguably one of the earliest successful implementations of a CNN model for such landform classifications was reported by Emami et al. (2015) in 2015, which achieved a detection rate of 92% with precision rate of 85%. Since that time the majority of researchers in this field have followed suit, focusing primarily on implementations of variations of the CNN model architecture. For example, more recently Emami et al. (2018) continued their research using a Fast R-CNN structure, while both Silburt et al. (2019) and Lee (2019) use a UNET CNN architecture for the same task. Lee and Hogan (2021) advanced on this work by using the ResUNET CNN architecture and achieved a human level performance in the model. However, while researchers cite these levels of performance, it is important to note that the performances of these models appear strongly dependent on the sizes of craters under test. For instance, Di et al. (2014) achieves a much higher performance in crater detection (74% recall) on craters greater than 6km in diameter, while the same model only produces a recall of <10% when tested on craters of all sizes. This would appear to be a crucial consideration in the evaluation of these types of models. Interestingly, each of these researchers apply their CNN approaches to the task of crater detection, suggesting this task is of particular interest in the latest research. It is also evident that researchers do not typically consider multiple types of landforms in their classifications, instead generally focusing on the detection of single landform types.

Notably, there appears to be a shift towards the CNN model in recent years among researchers, which is particularly evident in the experienced researchers opting for the CNN in their most recent papers and in their citing of superior performance of this model for landform classification tasks Wang et al. (2021); Lee and Hogan (2021); DeLatte, Crites, Guttenberg and Yairi (2019); DeLatte, Crites, Guttenberg, Tasker and Yairi (2019).

2.3 Terrain Classification

The classification of terrain is another field in which machine learning has become widely used by researchers. This is generally considered a separate task to that of landform classification described in the previous section, and involves the application of machine learning algorithms to satellite data and images.

Ono et al. (2015) defined the primary function of a terrain classifier as "to take an image as input, and classify every pixel in the image". In their paper, they classify images into one of five defined terrain classes which range from sand to pointed rocks. To achieve this, the authors implemented a random forest model consisting of 50 binary decision trees. However, while a confusion matrix is provided with accuracies of 76.2% and 89.2% for two of the terrain classes, the authors fail to explore the model's performance across each of the 5 terrain classes. Notably, this solution is applied to the task of terrain classification on Mars, which is particularly relevant to the research question of this work, and the authors do note the generalisability of the solution to other sources of satellite imagery from Mars. Furthermore, in 2013 Shang and Barnes (2013) also proposed a novel approach specific to Martian terrain classifications which was based on support vector machines (SVM). While the authors demonstrate the superiority of this approach over either decision trees and K-Nearest-Neighbours (KNN), their solution and discussion is primarily aimed at future Mars rovers missions, utilising data from onboard cameras and neglects the possibility of manned missions or the use of satellite data for global mapping applications.

Barrett et al. (2022) describe the application of a deep neural network to satellite images in order to classify regions of terrain as one of 14 defined terrain classes. The authors evaluated this novel solution on precision, recall and intersection over union (IoU) metrics, and achieved an IoU in excess of 80% for a number of terrain classes, clearly indicating the suitability of the neural network for this type of task. Furthermore, Schönfeldt et al. (2022) employ two models, both CNNs (AlexNet and U-Net architectures) to the classification of landslide terrains from elevation satellite data. Using this approach, the authors identified 12,000 square kilometres of landslide area. The authors again evaluate the models using precision and recall, but also note the high degree of variance in results, for instance with precision values ranging from 0.56 – 0.84 depending on interpolation method and data trained on (optical vs topographic), suggesting these are important considerations in such approaches.

In conclusion, while there are a number of approaches which have demonstrated successful terrain classifications, given the more recent research (Barrett et al.; 2022; Schönfeldt et al.; 2022) all opt for the CNN with well-documented advancements, which suggests the CNN is the most promising route for future research and to address the research question.

2.4 Multi-Task Classification

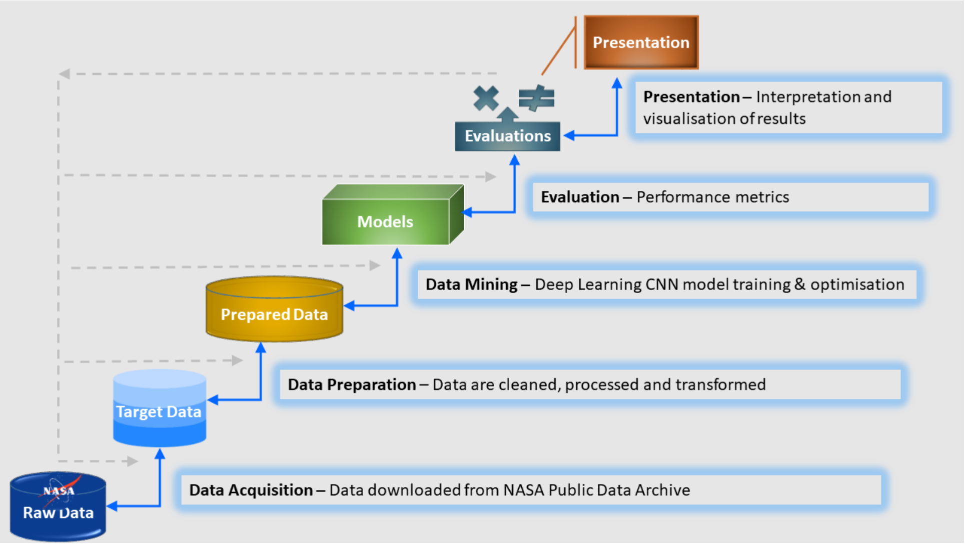
Unlike the single classification tasks discussed in the previous two sections, multi-task learning combines much of the same concepts to apply machine learning methods to more than one task. Researchers suggest that allowing a single model to learn from data related to multiple correlated tasks is both more computationally efficient and allows for a greater generalisability across related tasks.

Some emerging multi-task classification systems would appear to be a logical evolution of the CNN discussed previously. For instance, Murugesan et al. (2019) recently proposed a boundary and shape aware multi-task deep neural network for the segmentation of medical images, which they describe as a means to “improve and refine the performance of U-Net-like” CNNs. Dubbed “Psi-Net”, the model consists of an architecture designed for the accomplishment of 3 distinct classification tasks, namely mask predictions, contour predictions and distance map estimation. While it is worth considering that the authors report the Psi-Net model as outperforming the single-task U-Net CNN, it is also important to note that this application uses medical image data and may not be generalisable to satellite data. Perhaps a more relevant example of multi-task algorithms is Long et al. (2022) in which the BsiNet multi-task model delineates agricultural fields from high resolution satellite data. In particular, the authors note this approach achieved the lowest global total error of 0.291 in a comparison of models that included Psi-Net and the CNN, and also suggest the approach may be applied to different tasks. Furthermore, Tambe et al. (2021) also report advantages of a multi-task approach over a standard CNN in a reduction of trainable parameters allowing the model be trained on less data, a reduced computational cost and most significantly an improvement in performance as shown in an IoU score of 0.9434. These observations appear to be consistent with the findings of Khalel et al. (2019), who also demonstrate a superior performance in the multi-task CNN approach. Based on this consensus among researchers, the multi-task CNN algorithm is a particularly relevant approach to address the research question. While researchers have demonstrated its potential for high performance, they perhaps do not place enough emphasis on the particular tasks which can be combined within this approach, and the additional insight and domain context which can be garnered by combining classification tasks, and this therefore warrants further research.

Conclusively, there appears to be a gap in the research literature as researchers have focused on the application of these multi-task classification systems to particular tasks and where they make comparisons of performance they typically compare against the established CNN models such as the U-NET or ResUNET mentioned previously, overlooking considerations of variations in the multi-task algorithm architecture. Essentially, comparisons of different multi-task model structures or architectures is neglected. For instance, Khalel et al. (2019) mentions the ability to "hard-share" or "soft share" parameters between multi-task model layers, but fails along with the other researchers to explore the effect these model architectures could have on performance.

3 Methodology

This section outlines the methods followed as part of the research project. The general methodology is based upon the Knowledge Discovery in Databases (KDD) framework as the source data can essentially be considered akin to a database and is outlined below in Fig 1. This section details the acquisition, analysis and pre-processing of the dataset followed by an overview of the procedure carried out as part of the research work.



Presentation
Presentation – Interpretation and visualisation of results
Evaluations
Evaluation – Performance metrics
Models
Data Mining – Deep Learning CNN model training & optimisation
Prepared Data
Data Preparation – Data are cleaned, processed and transformed
Target Data
Data Acquisition – Data downloaded from NASA Public Data Archive
NASA
Raw Data

Figure 1: Research Methodology

3.1 Equipment & Materials

The research was carried out in a Google Colab Pro+ environment, with 2 GPUs and 52 GB of RAM. Data was stored in Google Drive which had 15 GB of storage capacity. The tensorflow software library was used in conjunction with the keras python interface, along with various packages available within python, including pandas, numpy and cartopy. The solution was developed using python version 3.

3.2 Raw Source Data Acquisition

The raw source data is the Mars MOLA DEM Global 200m v2 dataset which is made publicly available by the Astrogeology Science Center on their website (see footnote). [1] This raw data is a global digital terrain model (DTM) of the entire planetary surface of Mars. It consists of a mosaic of blended DTMs obtained from a number of satellite instruments including High-Resolution Stereo Camera (HRSC) and the Mars Orbiter Laser Altimeter (MOLA), aboard the Mars Express and Mars Global Surveyor satellites, respectively. The data contains measurements of altitude across the Martian surface. The resolution is approximately 200m/pixel.

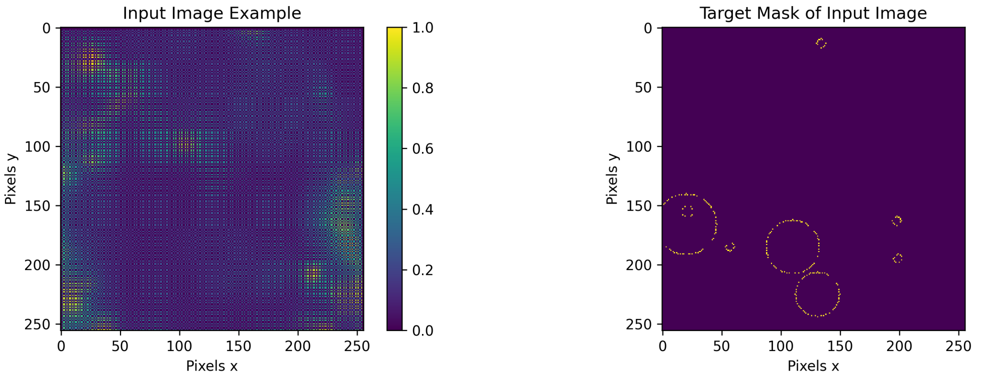
3.3 Data Pre-Processing

Due to the extremely large size of the source data (106694 x 53347 pixels), coupled with the hardware constraints of RAM, the dataset used as part of this research project was extracted as randomly sampled images from the source data. The random sampling process was carried out as follows:

1. A random pixel in the source DTM is chosen, and a square region of dimensions 256 x 256 pixels is selected by sliding 256 pixels horizontally and vertically.
2. The 16-bit resolution of the source image is converted to 8-bit.
3. The cylindrical Plate Carree projection of the source data is converted an orthographic projection using Python's Cartopy package. This gives the image a constant linear scale as opposed to constant angular resolution.
4. The image is padded to refill it to a square dimension where required after projection conversion.

The final dataset was constructed by following the above steps to create 85,000 images. This is a larger dataset than those used in related literature (Lee; 2019). There were two classes of labels applied to the image dataset. Namely, these were the presence and location of craters, and the classification of the terrain in the images. Firstly, the coordinates and characteristics of the craters were taken from the Robbins and Hynek (2012) crater catalogue, which provides the ground-truth crater labels as classified by domain experts. However, since these coordinates are in the form of latitude and longitude values, their locations and diameters needed to be converted to pixel-based values corresponding to each image. This allowed binary target mask images to be created as the crater class. Secondly, the K-Means clustering algorithm was fit to the image dataset

[1] `https://astrogeology.usgs.gov/search/map/Mars/Topography/HRSC_MOLA_Blend/Mars_HRSC_MOLA_BlendDEM_Global_200mp_v2`



(a) Input Image

(b) Crater Target Mask

Figure 2: Visualisation of Input Image with Corresponding Crater Target Mask

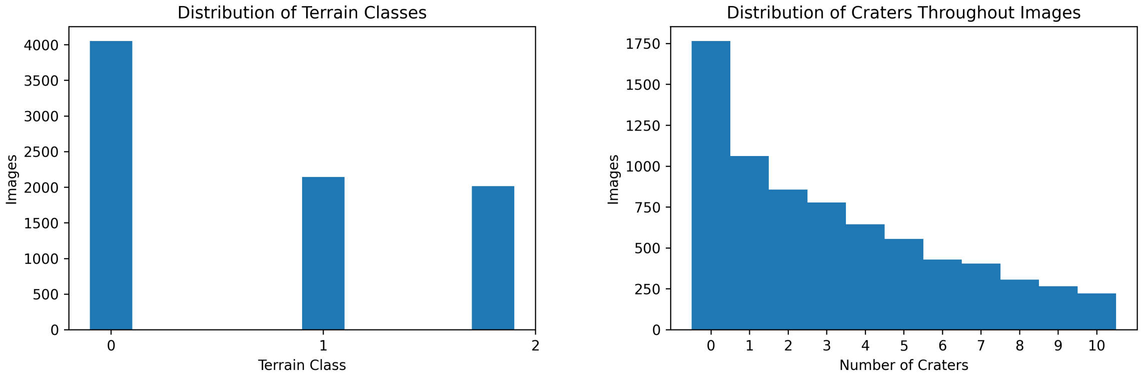
and terrain labels assigned to each image. Three clusters were used to represent flat, sloped and steeply sloped terrains.

3.4 Data Exploration

This section presents the exploratory analysis performed on the dataset constructed in the previous section.

Firstly, it was verified that there were 85,000 distinct images present as expected. The dataset was examined for null or blank corrupted images. Fig 2 shows an example of a typical input image along with the accompanying target mask label containing the crater locations. The terrain class label for this image was 0 which represents a flat terrain. In this figure, the pixel values of the image are presented on the left and there are 5 distinct craters which can be seen on the right.

It was also important to explore the distribution of the dataset by class labels. Since there were two class labels, i.e. craters and terrain, Fig 3 shows the distribution of terrain classes on the left and the number of craters contained in each image on the right. The terrain labels 0, 1 and 2 represent terrain classifications of flat, sloped and steeply sloped, respectively. It can be seen that there are roughly twice as many images classified



(a) Terrain Labels

(b) No. of Craters

Figure 3: Distribution of Classification Labels in Image Dataset

as being flat terrain (∼2000) compared to the other two classes (∼1000 each). Similarly, most images were shown to have 0 craters present (∼1750), while a large portion of images contained a small number of craters, for instance over 2000 images had between 1 and 3 craters, smaller numbers of images contained more craters. This figure focuses on images containing up to 10 craters, but it should be noted that very small numbers of images contained substantially more craters, the highest number of craters contained in a single image being 61. Images with this many craters may need to be considered as outliers as they may exert an undue influence on the rest of the dataset.

3.5 Data Preparation & Processing

There were a number of data processing steps taken in order to prepare the data for use in the data mining phase of the research. These steps are outlined as follows:

1. Blank, null or otherwise corrupted images were removed from the dataset.
2. Outliers were removed. Outliers included images which contained a disproportionate number of craters as these images exerted an undue influence over the dataset. Any image containing more than 50 craters was removed.
3. Craters that were too small to be seen in images were removed. Only craters that spanned at least 3 pixels in an image were included in the dataset.
4. The ground-truth list of craters was then used to create target mask images, which were binary and depicted the size and location of target craters.
5. The data in each image were normalised by dividing each pixel value by 255.
6. The distribution of the pixel values in each now-grayscale image is then standardised using a standard scaler.

The previous section identified a slight imbalance in the terrain labels of the dataset. In general, this would mean the data should be resampled, for instance by using smote to over-sample the minority classes, but this was not possible in this case as the data could not be resampled without negatively affecting the crater labels. This is because the crater labels were mapped from longitude/latitude coordinates to pixels locations, and resampling the images would interfere with this mapping. While a resampling of the data may be beneficial, there are still sufficient samples of all labels to proceed with the slight imbalance present.

3.6 Design Specification

This section presents the end-to-end design of the developed solution. The diagram shown in Fig 4 explains the stepwise design from data acquisition and preparation as described in the previous sections, as well as the designed model and the prediction tasks to be made.

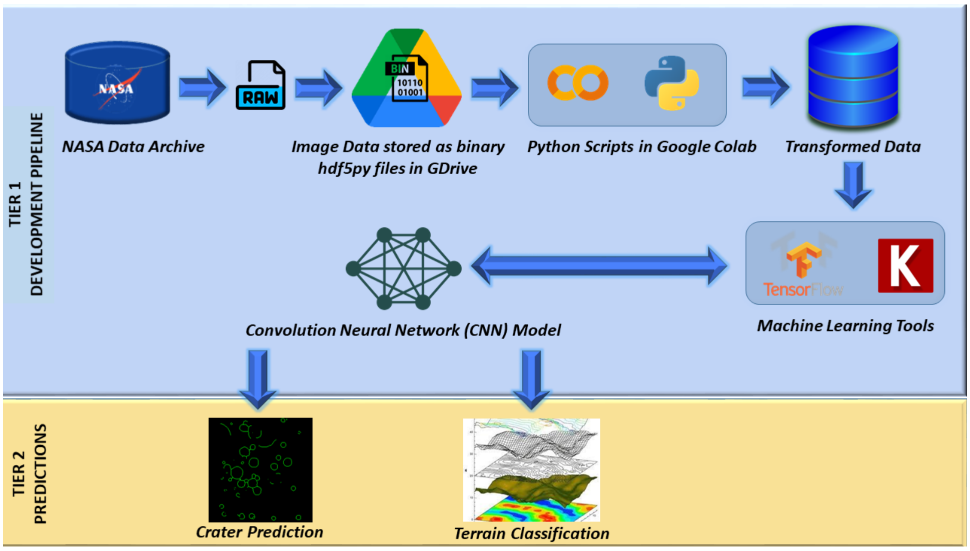


Figure 4: Solution Design

4 Implementation

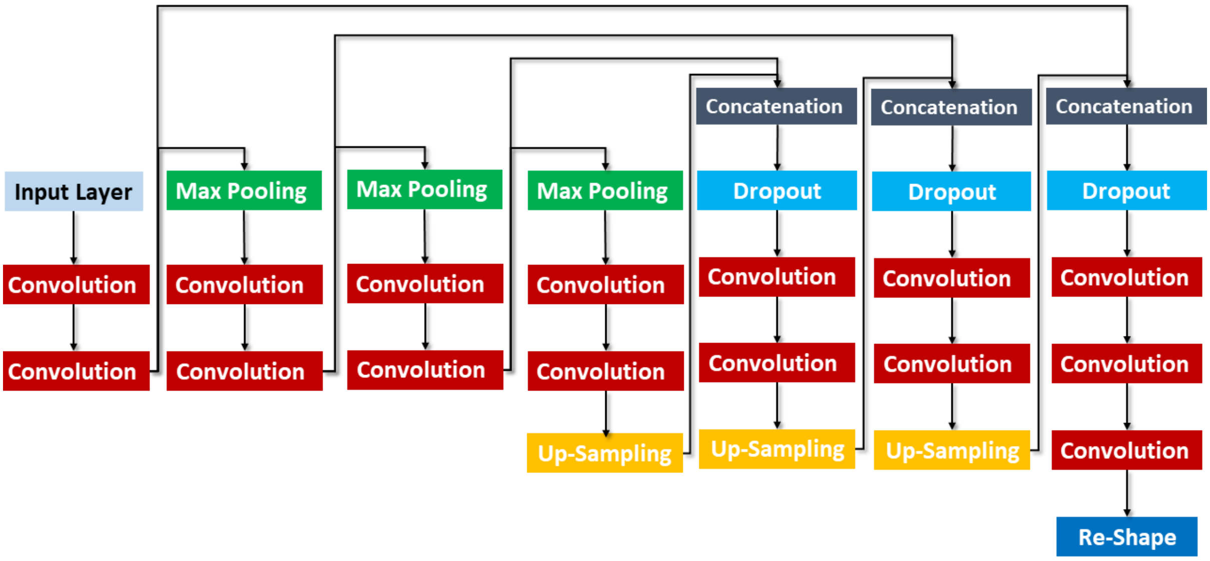
The multi-task convolution neural network (CNN) solution was designed for the purpose of performing two classification tasks: the detection of craters and the classification of terrain types. As a control, these tasks were firstly examined in isolation, with two single-task CNN models designed to perform both tasks separately. Then, three multi-task CNN algorithms were developed utilising three different model architectures in order to compare and investigate the optimal method of combining these two separate classification tasks into a multi-task CNN. A detailed description of each of these five total models is provided below.

4.1 Single-Task Crater Detection CNN

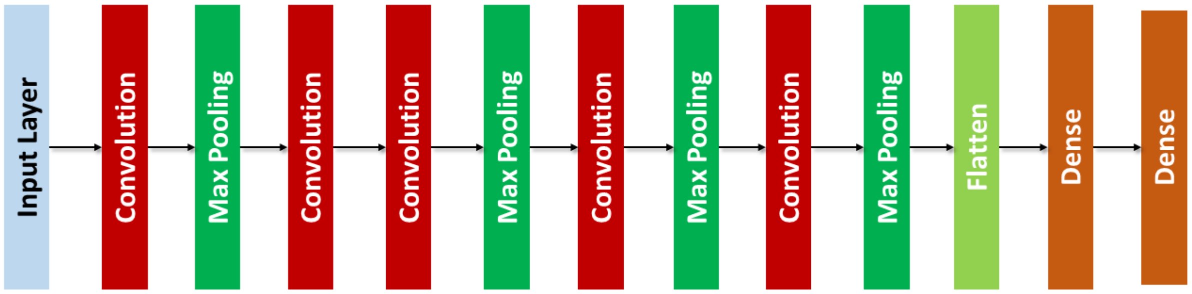
The single-task crater detection CNN was designed to highlight the presence of a crater in a given image. The design was based primarily on the approaches taken by both Lee (2019) and Silburt et al. (2019), which is depicted in Fig 5 part (a). The model takes a 256 x 256 pixel image as an input. It contains 3 sets of 2 convolution layers which are followed by max pooling layers. These are then merged with a dropout layer and this is followed by an additional set of 3 convolution layers and a final reshape.

4.2 Single-Task Terrain Classification CNN

The single-task terrain classification CNN was designed to classify the terrain captured in an image as one of 3 pre-defined terrain classes: flat, sloped and steeply sloped. The model architecture is shown in Fig 5 part (b) and was based on those described in Khalel et al. (2019); Tambe et al. (2021), and consists of a series 5 convolution layers, each one followed by a max pooling layer, with a single, fully connected hidden layer and finally an output layer.



(a) Single-Task Craters Detection CNN Model Architecture



(b) Single-Task Terrain Classification CNN Model Architecture

Figure 5: Model architecture of the single-task CNN models

4.3 Multi-Task Classification CNN: Independent Branches

The multi-task classification CNN essentially combines the two single-task models discussed above. This ’independent branches’ model is designed to take the same 256 x 256 pixel images as input, but provide two distinct output classifications: crater detections and terrain classifications. The model architecture is shown in Fig 6 and is designed to soft-share parameters between the layers, as described by Khalel et al. (2019). Each layer is specific to a single task, but the weights are updated as per the soft-parameter sharing which allows the model to update based on both tasks simultaneously. Each of the independent branches consist of the same layers as outlined in the two sing-task sections above.

4.4 Multi-Task Classification CNN: Mixed Layers

The ’mixed layers’ multi-task CNN is designed to build upon the independent branches model discussed previously. Instead of keeping all network layers separated by task, this model comprises a mixture of some layers shared between both classification tasks, while still retaining a number of layers that are task-specific. This can be seen in Fig 7, where the first 6 convolution layers and 2 max pooling layers are common to both tasks, before the model is split into 2 separated sets of layers, one utilising the single-task terrain type classification architecture, and the other utilising the single-task crater detection type architecture. In this way, this model can be considered to lie between a soft and a hard parameter sharing model, like a firm parameter sharing model. As shown in the diagram, the shared layers are made up of 3 sets of 2 convolution layers each followed by max pooling layers. The shared layers are then followed by the task-specific, independent layers. The craters classification set of layers is made up of 3 sets of 2 convolution layers, each again followed by max pooling layers and the output layer. The terrain classification

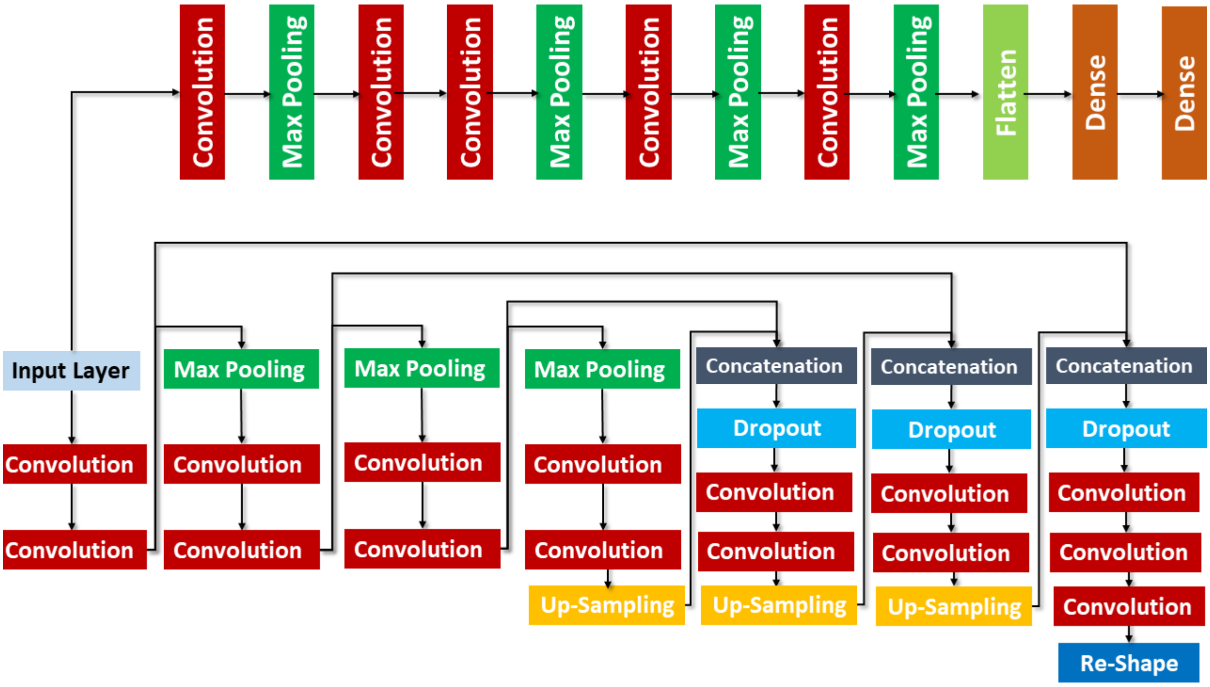


Figure 6: Multi-Task Independent Branches CNN Model Architecture

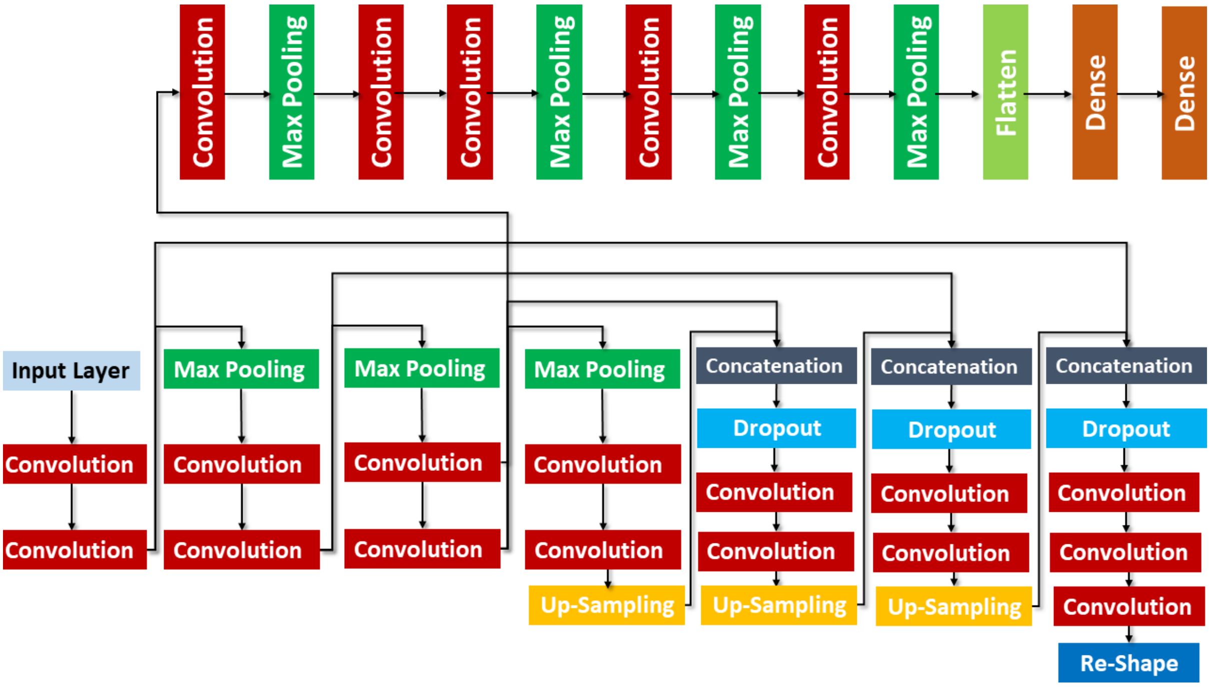


Figure 7: Multi-Task Mixed Layers CNN Model Architecture

set of layers contains 3 sets of single convolution layers, each followed by max pooling layers, a flattening layer, a hidden dense layer and finally the output layer.

4.5 Multi-Task Classification CNN: Shared Layers

The ’shared layers’ multi-task CNN is a hard parameter sharing model, as the architecture consists primarily of layers shared between both tasks. This can be seen in Fig 8. The model is designed with all layers, made up of a total of 14 convolution layers, 3 max pooling layers, 3 up-sampling layers, 3 dropout and 3 concatenation layers shared between both tasks. It is only at the final convolution layer that the model diverges to final output layers. This is to facilitate the output layers of the 2 given tasks, a flattening and a dense layer for the terrain class task, and a convolution and re-shape layer for the crater class task.

4.6 Model Training

The data was loaded from hdf5 files using python’s hfpy library and stored in numpy multi-dimensional arrays. Numpy and pandas were used to process the data as outlined in the previous section. Keras and tensorflow were used to create and train the models. As stated in the Methodology section, the total dataset contained 85,000 images. This was split into training, validation and testing sets, 57,375 images used for training, 19,125 for validation during the training phase and 8,500 reserved for testing. It should be noted that this is therefore a significantly larger dataset than that used by Lee (2019). However, due to this large volume of data coupled with the hardware constraints, it was not possible for the entire dataset to be fed into the model in a single instance. Instead, the following procedure was used to ensure the models were each trained on the full dataset for 10 epochs. For each epoch:

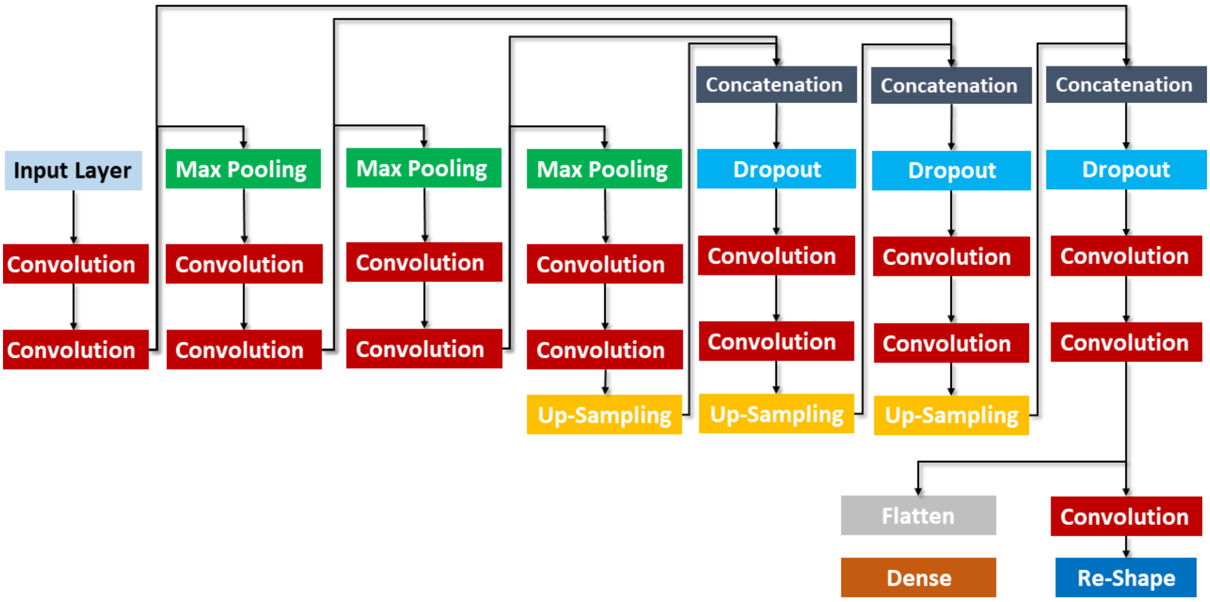


Figure 8: Multi-Task Shared Layers CNN Model Architecture

1. A saved version of the model was checked for. If the model already existed, it was loaded from it's last saved checkpoint. Otherwise, the model was created at this point.

2. A single data file which contained 8,500 images was loaded into memory.

3. The model was trained on the loaded data with a batch size of 15 images for the

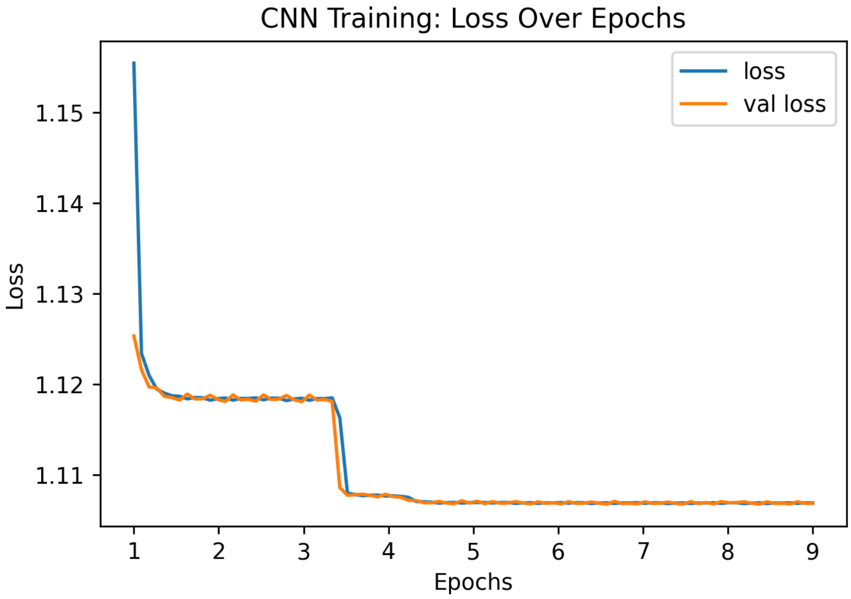


Figure 9: Model Training: Loss vs Epoch

single epoch.

4. The model was then saved to file.
5. Steps 1-4 were repeated until all the image files were loaded and used to train the model.

As Fig 9 shows, the models improved significantly in the early epochs, but stop improving after 3-4 epochs. Since this corresponds to a training set of 85,000 distinct images ($\sim$ 300,000 in total), this model training is consistent with that of Lee (2019). The similarity in loss between the training and validation sets also indicates a good model fit.

4.7 Evaluation

Given they are supervised models, the evaluation techniques for each of the 5 developed models were based on those as used by Lee (2019); Silburt et al. (2019), which are some of the most commonly used performance metrics for evaluating such models. These techniques are accuracy, recall, precision and F_1 Score and are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$F_1 Score = 2\frac{PR}{P + R} \tag{4}$$

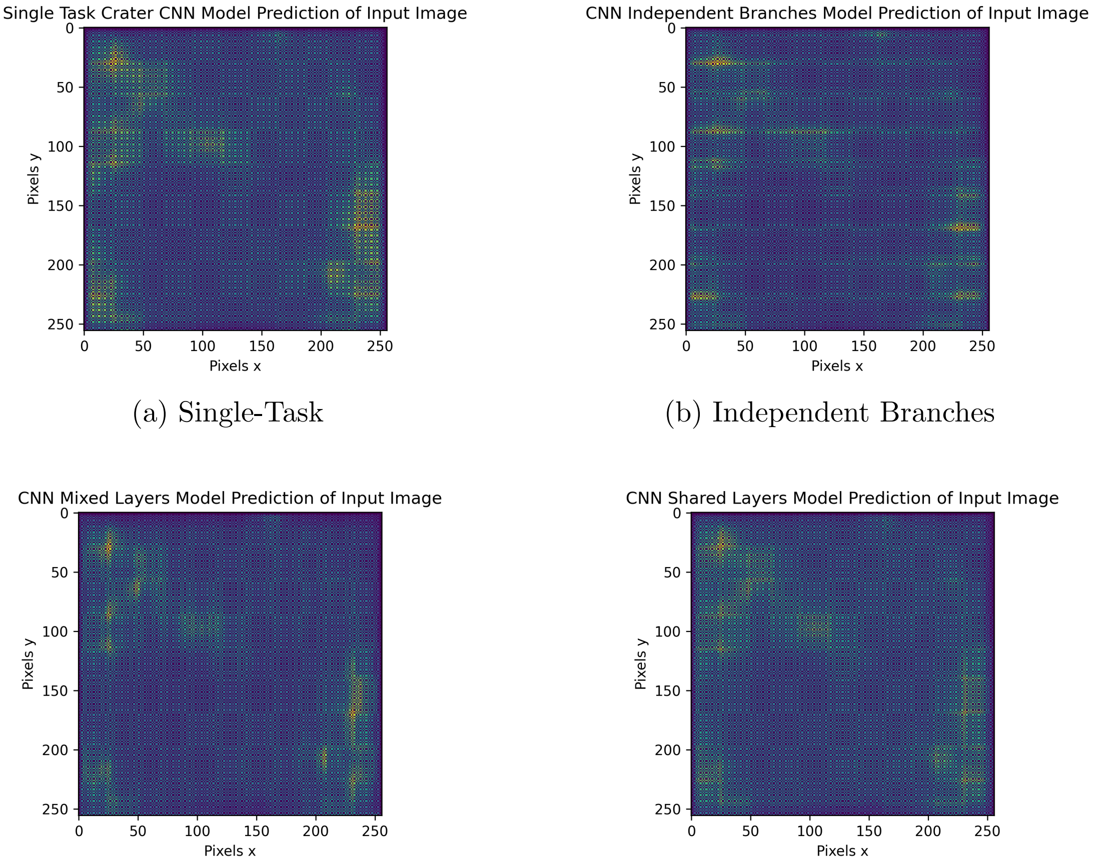
where TP denotes true positives, TN is true negative, FP is false positive and FN is false negative. In the crater detection task, craters are identified within model output predictions by converting the predicted images to binary and running python's scikit-learn template_match algorithm, while terrain class predictions are output as a series of class probabilities which are converted to the predicted class by using numpy's argmax function.

5 Results & Discussion

This section presents and discusses the results of the research project. Firstly, the outputs from the models described in the previous section are reported in the first subsection. The two classification tasks, crater detection and terrain classification are then examined separately in the subsequent subsections.

5.1 Model Outputs

The example input image shown in Fig 2 in the methodology section was used as an input to test each of the 5 models and the output predictions from each of the 4 cater-predictive models are presented in Fig 10 (as the single-task terrain classification model only predicts terrain types). As would be expected, the predictions from each of the



(a) Single-Task

(b) Independent Branches

(c) Mixed Layers

(d) Shared Layers

Figure 10: Crater Prediction Outputs From Each Model Based on Example Input image

models are similar, although there are subtle differences visible upon inspection of these subfigures. For instance, there appears to be a greater contrast in pixel values in the single-task model compared to the others. Furthermore, this difference in contrast can also be seen between the independent branches model versus the mixed and shared layers model. This may suggest that increasing the number of layers shared between the two tasks and reducing the number of task-specific layers in the model may diminish the contrast in pixel values in the resultant predicted images. Nevertheless, each of these models predict highest intensities in pixel values in generally the same areas, suggesting they may all perform similarly in making classifications.

5.2 Crater Detection

Table 1 presents the results of the crater detection task for each model. Interestingly, the single-task model appears to perform best with the highest accuracy and recall values, suggesting it is the most efficient model at detecting craters, although there is clearly merit in the multi-task models also, as the independent branches recorded the best precision by a considerable margin and the shared layers model had the highest F_1 score. With recall values in the range of 30-40% and precision values <1% for all 4 models, the performances of these models may appear poor at first glance, but these results are actually in line with the results reported by many researchers. In truth, while many researchers have been capable of identifying craters of particular sizes quite well, when generalised to include

Model Performances on Crater Detection Task					
Model	Craters Detected	Accuracy	Recall	Precision	F_1
Single-Task	301523	0.333	0.385	0.0125	0.0208
Independent Branches	476483	0.1	0.34	0.0552	0.0105
Mixed Layers	225993	0.1	0.343	0.0119	0.022
Shared Layers	257921	0.1	0.376	0.012	0.0215

Table 1: Model Performance Metrics For Crater Detection Task

craters of all sizes the results tend to degrade significantly. For instance, Di et al. (2014) reported recall values of 74% for craters larger than 6km in diameter, but their same model records a recall of <10% on craters of all sizes. Similarly, Lee (2019) reports a 'best' model capable of identifying 75% of craters in the same database as used here, but a 'worst' case model identifying only 46% of these craters. Therefore, while these results show that each of the 4 models are indeed capable of detecting craters, a limitation of the study was that these results only provide a view of the calculated metrics over the entire dataset. If the results could be broken down into bands of crater size, it may be found (as with other researchers findings) that the models perform significantly better when detecting craters of one size (larger craters with diameter >10km for instance) than on others. This would provide a greater insight into the model performances than the results available here and should form part of the future work to be discussed in the next section.

Furthermore, it should be noted that the crater detections are extremely sensitive to the parameters used to define a 'distinct' detection and to determine when a detection is indeed a true positive crater matched to a ground-truth crater. These parameters include the following:

- **Minrad**: The minimum radius to search for within the predicted image.
- **Maxrad**: The maximum radius to search for within the predicted image.
- **Long_lat_thresh**: The squared minimum difference in latitude/longitude coordinates separating craters to consider them distinct detections.
- **Template_Thresh**: The minimum correlation coefficient of the match_template function to identify a detection.
- **Target_Thresh**: The threshold used to convert images to binary, where each pixel above the threshold is set to 1 and all pixels below the threshold are set to 0.

Table 2 shows the effect of changing these parameters on the calculated performance of the models. From this table, it is clear that reducing the template matching threshold produces significantly more crater detections, which in turn means more craters will be matched (true positives) and therefore the recall values tend to increase, but on the other hand the increased number of crater detections also increases the number of false positives, which leads to a significant reduction in the precision and F_1 metrics. Similarly, the Target_Thresh parameter also directly affects the number of craters detected. Lower values of this parameter allow more artefacts to be present when the image is converted to binary, thereby producing more overall crater detections. While other researchers (Lee

Crater Detection Parameter Tuning Effects on Model Performance Metrics									
Parameters					Metrics				
Minrad	Maxrad	Long-lat Thresh	Template Thresh	Target Thresh	Craters Detected	Matched Craters	Recall	Precision	F_1
2	30	5	0.2	0.0035	301523	9145	0.385	0.01245	0.0208
2	30	1	0.5	0.0035	5304	9145	0.164	0.193	0.103
2	50	1	0.35	0.0035	106931	9145	0.2483	0.01154	0.0175
2	50	1	0.2	0.005	951593	9145	0.290	0.002	0.005

Table 2: Crater Detection Parameters Effect on Model Performance

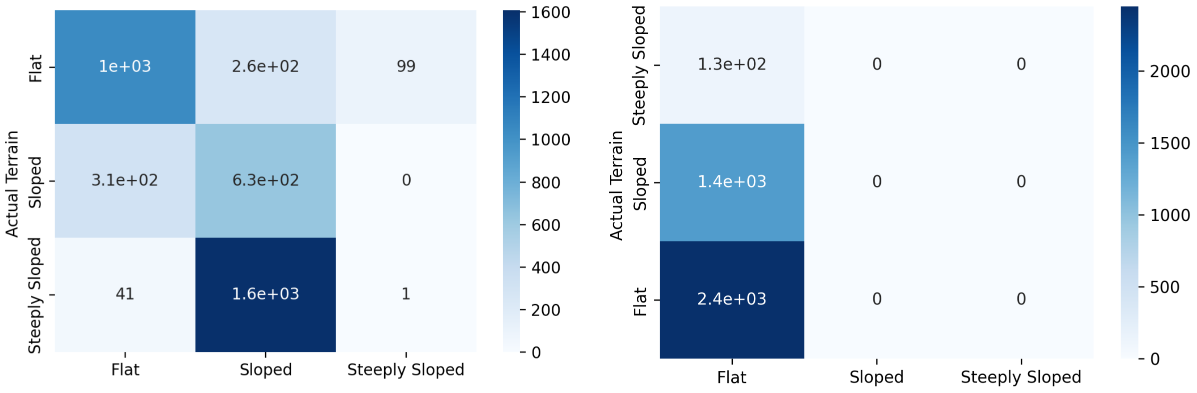
(2019); Silburt et al. (2019)) use values as high as 0.1, the mean pixel value of model-predicted images was just 0.0045, and the maximum pixel value was less than 0.007, and so lower values were used as part of this research. The long_lat_thresh parameter can also be seen to affect the number of crater detections as a larger threshold means craters must be located further apart in order to be considered two distinct detections, and so a higher threshold produces less detections which in turn results in lower recall but higher precision and F_1 values. In order to truly adjudge the potential and quantify the optimal performance of these models on crater detection, each of these parameters would need to be iterated over and the eventual results compared across all permutations of parameter values. However, given that some of these parameters can take unbounded values, the large number of possible increments in values, and the number of different parameters, the time required to perform this parameter tuning quickly exceeds the maximum runtimes allowed by the Google Colab environment used as part of this research. It is therefore possible that a tuned combination of parameter values exists that would produce a significantly enhanced performance of the models, although further research is required in order to determine this set of parameters.

5.3 Terrain Classification

Each of the 4 models (3 multi-task and here one single-task terrain classifier) were tested using the testing set of images and the results are presented in Table 3. Firstly, while these results appear to show some potential for classifying surface terrains, the performances of all 4 models across each of the accuracy, recall, precision and F_1 metrics are perhaps lower than has been reported by researchers using similar approaches (Barrett et al. (2022)) and there are a number of reasons which may account for this. In particular, unlike the task of crater detection above, there is no readily available database of terrain classifications for the Martian surface terrain created by domain experts to provide the "ground truth"

Model Performances on Terrain Classification Task				
Model	Accuracy	Recall	Precision	F-Score
Single-Task	0.4195	0.4195	0.4195	0.4195
Independent Branches	0.3515	0.3515	0.3515	0.3515
Mixed Layers	0.3515	0.3515	0.3515	0.3515
Shared Layers	0.3515	0.3515	0.3515	0.3515

Table 3: Model Performances on Terrain Classification Task



(a) Single-Task

(b) Multi-Task: Shared Layers

Figure 11: Terrain Classification Confusion Matrices

terrain class labels. Instead, a simple unsupervised K-Means algorithm was fit to the data and used to generate a set of 3 labels corresponding to flat, sloped and steeply sloped terrains. Clearly, ground truth labels validated by domain experts would be more desirable, and especially since these are supervised models, this likely has a direct negative impact on the classification results. Furthermore, as discussed in the previous subsection, given the RAM constraints in training the models, saving and reloading the models within each epoch may also have an effect on the performances. Access to improved hardware or using smaller training datasets may overcome this challenge.

Interestingly, the single-task model performed best with each of its metric results almost 42% while all of the multi-task models recorded metric values just above 35%. This may suggest that combining the two tasks into the multi-task models produced a degradation in performance. However, this did not seem to be the case in the crater classification task, and it is possible that alternative CNN architectures, such as the AlexNet or U-Net as proposed by Schönfeldt et al. (2022) may correct this.

It is also perhaps unexpected to see that the multi-task models all perform such that each of the accuracy, recall, precision and F_1 metrics result in the same values. This becomes clearer upon inspection of the confusion matrices of the models shown in Fig 11. While the single-task model makes predictions across each of the 3 terrain classes, the multi-task models appear to predict the same class regardless of the input. This suggests the single-task model has learned the task better, which accounts for its superior performance. However, it should be noted that since there are a significantly larger number of trainable parameters in the multi-task model (10 million in the multi-task compared to 3 million in the single-task model), it is possible that the multi-task model may need significantly more training data or require additional training epochs in order to adequately learn each task.

6 Conclusion & Future Work

This research work sought to address the research question of "How can multiple rather than single classification techniques be used in machine learning models to better evaluate Martian surface regions?" and did so by designing and implementing a novel multi-task deep learning approach to the tasks of crater detection and terrain classification. While the performance of this approach may appear below that of the established single-task models, the potential for using this type of model to evaluate regions of the Martian surface on multiple classification techniques has been demonstrated and this constitutes a significant contribution to the research literature.

The solution was developed by gathering publicly available data from the NASA archive, applying class labels derived from the Robbins and Hynek (2012) database and an unsupervised K-Means algorithm, designing a series of multi-task models and implementing them in python and finally training and testing the models on the data. While the single-task models were found to outperform the multi-task models in accuracy (37.6%) and recall (38.5%), the multi-task models recorded higher precision (5.52%) and F_1 scores (0.0215) in the crater detection task. The single-task model outperformed the multi-task in all metrics in the terrain classification task. Despite this, the multi-task models provide a viable means for evaluating surface regions on these two classification techniques instead of single techniques in isolation.

Given the potential additional insights which may be gained by using a multi-task model for this purpose, numerous identified points of future work are warranted. In particular, retraining the multi-task models on the entire training set in each epoch using upgraded hardware may improve these models' performance. Additionally, a further investigation and tuning of the parameters defining crater detections in predicted images may also contribute to future improvements. Moreover, examining the results broken down by crater sizes, and obtaining ground-truth labels for the terrain classification task would also be beneficial and would likely produce superior models. Furthermore, since the multi-task model architectures designed in this research work successfully produced multiple classification outputs, it would be interesting to incorporate different model architectures into the different branches and layers of the multi-task models. For instance, this work utilised a UNET CNN architecture, but future work may investigate the use of Alex-NET or Res-UNET architectures, as well as combining different architectures within the model to perform each separate task. Finally, this work concentrated on two classification tasks in the multi-task models, but the only upward limit on the number of tasks which can be performed by the multi-task model is the number of trainable parameters that can be handled. Therefore, future work should also seek to incorporate more than two tasks into the multi-task models, such as the detection of not only craters but perhaps valleys, dunes or mountains as well.

References

Barrett, A. M., Balme, M. R., Woods, M., Karachalios, S., Petrocelli, D., Joudrier, L. and Sefton-Nash, E. (2022). NOAH-H, a deep-learning, terrain classification system for Mars: Results for the ExoMars Rover candidate landing sites, *Icarus* **371**: 114701. **URL:** *https://www.sciencedirect.com/science/article/pii/S0019103521003560*

DeLatte, D. M., Crites, S. T., Guttenberg, N., Tasker, E. J. and Yairi, T. (2019). Segmentation Convolutional Neural Networks for Automatic Crater Detection on Mars, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **12**(8): 2944–2957. Conference Name: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

DeLatte, D. M., Crites, S. T., Guttenberg, N. and Yairi, T. (2019). Automated crater

detection algorithms from a machine learning perspective in the convolutional neural network era, *Advances in Space Research* **64**(8): 1615–1628.
URL: *https://www.sciencedirect.com/science/article/pii/S0273117719305071*

Di, K., Li, W., Yue, Z., Sun, Y. and Liu, Y. (2014). A machine learning approach to crater detection from topographic data, *Advances in Space Research* **54**(11): 2419–2429.
URL: *https://www.sciencedirect.com/science/article/pii/S0273117714005304*

Emami, E., Ahmad, T., Bebis, G., Nefian, A. and Fong, T. (2018). Lunar Crater Detection via Region-Based Convolutional Neural Networks, p. 2381. Conference Name: 49th Annual Lunar and Planetary Science Conference ADS Bibcode: 2018LPI....49.2381E.
URL: *https://ui.adsabs.harvard.edu/abs/2018LPI....49.2381E*

Emami, E., Bebis, G., Nefian, A. and Fong, T. (2015). Automatic Crater Detection Using Convex Grouping and Convolutional Neural Networks, *in* G. Bebis, R. Boyle, B. Parvin, D. Koracin, I. Pavlidis, R. Feris, T. McGraw, M. Elendt, R. Kopper, E. Ragan, Z. Ye and G. Weber (eds), *Advances in Visual Computing*, Lecture Notes in Computer Science, Springer International Publishing, Cham, pp. 213–224.

Khalel, A., Tasar, O., Charpiat, G. and Tarabalka, Y. (2019). Multi-Task Deep Learning for Satellite Image Pansharpening and Segmentation, pp. 4869–4872.

Lee, C. (2019). Automated crater detection on Mars using deep learning, *Planetary and Space Science* **170**: 16–28.
URL: *https://www.sciencedirect.com/science/article/pii/S0032063318303945*

Lee, C. and Hogan, J. (2021). Automated crater detection with human level performance, *Computers & Geosciences* **147**: 104645.
URL: *https://www.sciencedirect.com/science/article/pii/S009830042030621X*

Long, J., Li, M., Wang, X. and Stein, A. (2022). Delineation of agricultural fields using multi-task BsiNet from high-resolution satellite images, *International Journal of Applied Earth Observation and Geoinformation* **112**: 102871.
URL: *https://www.sciencedirect.com/science/article/pii/S1569843222000735*

Murugesan, B., Sarveswaran, K., Shankaranarayana, S. M., Ram, K., Joseph, J. and Sivaprakasam, M. (2019). Psi-Net: Shape and boundary aware joint multi-task deep network for medical image segmentation, *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference* **2019**: 7223–7226.

Ono, M., Fuchs, T. J., Steffy, A., Maimone, M. and Yen, J. (2015). Risk-aware planetary rover operation: Autonomous terrain classification and path planning, *2015 IEEE Aerospace Conference*, pp. 1–10. ISSN: 1095-323X.

Rajaneesh, A., Vishnu, C. L., Oommen, T., Rajesh, V. J. and Sajinkumar, K. S. (2022). Machine learning as a tool to classify extra-terrestrial landslides: A dossier from Valles Marineris, Mars, *Icarus* **376**: 114886. ADS Bibcode: 2022Icar..37614886R.
URL: *https://ui.adsabs.harvard.edu/abs/2022Icar..37614886R*

Robbins, S. J. and Hynek, B. M. (2012). A new global database of Mars impact craters 1 km: 1. Database creation, properties, and parameters, *Journal of Geophysical Research: Planets* **117**(E5). _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2011JE003966.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1029/2011JE003966*

Schönfeldt, E., Winocur, D., Pánek, T. and Korup, O. (2022). Deep learning reveals one of Earth's largest landslide terrain in Patagonia, *Earth and Planetary Science Letters* **593**: 117642.
URL: *https://www.sciencedirect.com/science/article/pii/S0012821X22002783*

Shang, C. and Barnes, D. (2013). Fuzzy-rough feature selection aided support vector machines for Mars image classification, *Computer Vision and Image Understanding* **117**(3): 202–213.
URL: *https://www.sciencedirect.com/science/article/pii/S1077314212001968*

Silburt, A., Ali-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., Tamayo, D. and Menou, K. (2019). Lunar crater identification via deep learning, *Icarus* **317**: 27–38.
URL: *https://www.sciencedirect.com/science/article/pii/S0019103518301386*

Tambe, R. G., Talbar, S. N. and Chavan, S. S. (2021). Deep multi-feature learning architecture for water body segmentation from satellite images, *Journal of Visual Communication and Image Representation* **77**: 103141.
URL: *https://www.sciencedirect.com/science/article/pii/S1047320321000870*

Wang, H., Zhang, L., Yin, K., Luo, H. and Li, J. (2021). Landslide identification using machine learning, *Geoscience Frontiers* **12**(1): 351–364.
URL: *https://www.sciencedirect.com/science/article/pii/S1674987120300542*

Wang, Y., Di, K., Xin, X. and Wan, W. (2017). Automatic detection of Martian dark slope streaks by machine learning using HiRISE images, *ISPRS Journal of Photogrammetry and Remote Sensing* **129**: 12–20. ADS Bibcode: 2017JPRS..129...12W.
URL: *https://ui.adsabs.harvard.edu/abs/2017JPRS..129...12W*