Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project
Final Report

Crater Detection Algorithm and Degradation State Classifier using YOLOv3

by

Giles Matthews

giles.matthews20@imperial.ac.uk
GitHub login: acse-gcm20

Supervisors:

Prof. Gareth Collins, Navjot Kukreja
and Marijan Beg

August 2021

GitHub repository: https://github.com/acse-gcm20/YOLOv3-CDA

Abstract

Analyses of impact craters on planetary bodies offer effective techniques to determine relative ages of planetary surfaces and their geological histories. We implemented a successful automated solution to detect craters in images of Mars by using the 'You Only Look Once' v3 (YOLOv3) architecture. It is important to automate this process as the distribution of craters follows an inverse power-law, meaning for craters with diameters less than 1 km, it is not feasible to perform manual counts. This model was trained on a ground-truth set of labelled THEMIS images, thermal images from the 2001 Mars Odyssey mission. The detection model achieved an AP of 0.699 on its test set. A second model is developed which simultaneously detects and classifies craters according to their degradation state using a four-class system. This model achieves an mAP of 0.439 which we conclude is a valid proof-of-concept for the technique of using YOLOv3 to classify craters. With a larger and higher-quality training set it can be expected that this performance will be substantially improved upon.

1 Introduction

1.1 Background

The study of impact craters on planetary surfaces has been a fundamental aspect of planetary science for years. Accurate analyses of the frequency and density of crater distributions on planetary surfaces provides us with a mechanism to determine the relative ages of different surfaces and their geological histories (Urbach and Stepinski 2009). Until recently this process of counting craters has been performed manually in a laborious and time consuming manner. The difficulty of this task is compounded by the inverse power-law relationship between the size and frequency of craters which quickly renders the manual counting of sub 1 km craters impossible (Hartmann and Daubar 2017).

Since the first systematic crater counts were performed, the resolution of cameras has increased dramatically from km per pixel to cm per pixel scale resolutions. There is therefore an opportunity to automate the process of crater counting to bring it in line with current technology. This can be done using machine learning image detection techniques and convolutional neural networks (CNNs) (Albawi, Mohammed, and Al-Zawi 2017).

In addition to understanding the frequency and distribution of craters, it is also useful to determine the relative states of degradation that the craters are in. An understanding of the degradation states of craters in an area would further help to inform analyses of the geological history of a surface (Pina and Marques 2016). This kind of degradation data is very sparse in the literature due to the additional effort that goes into the classification of craters when performing counts. Some examples of this work are centered around the landing site of the InSight mission (Banerdt et al. 2020) but tend to focus on just rocky ejecta craters and do not provide a comprehensive database (Sweeney et al. 2018). Other attempts at automatically determining degradation states focus on determining the state of the crater rim and passing this to a learning algorithm to determine the crater degradation state (Pina and Marques 2016).

Some successful implementations of automated crater detection solutions have utilised, among other things, the 'You Only Look Once' (YOLO) architecture (Redmon, Divvala, et al. 2016), which can rapidly identify features in images, trained to detect craters. In this project we use YOLOv3 (Redmon and Farhadi 2018) - released in 2018, this version of the architecture offers improved accuracy at rapid speeds compared to previous iterations . The YOLOv3 feature extractor is built on the Darknet-53 architecture which has 52 convolutional layers (Redmon 2013–2016). As described by

Redmon (Redmon and Farhadi 2018), a key strength of this architecture is its ability to maintain strong performance over a range of image resolutions - this is particularly useful for the problem of crater detection. A key inspirational piece of work for this project was the work by G. Benedix et al. (Benedix et al. 2020) in applying the YOLOv3 object detection system to THEMIS images of Mars and training a successful crater detection algorithm from a carefully curated dataset.

One of the most notable contributions to the research of craters on Mars is the work of S. Robbins (Robbins and Hynek 2012) in collating a comprehensive database of all Martian craters greater than 1 km in diameter. This database is extremely useful but is also a good example of the need for automation as there are millions of craters on the surface of Mars with diameters less than 1 km which remain uncounted. Approximately 20% of the craters in the Robbins database have a degradation state classification, it is these craters that will be used to train a degradation state classifier.

1.2 Objectives

- Develop a crater detection algorithm (CDA) using the YOLOv3 object detection architecture (Redmon, Divvala, et al. 2016).
- Investigate the success of using the YOLOv3 architecture to simulateously detect and classify craters according to their degradation state.
- Investigate the effect of 'ambiguity' on the performance of models and identify methods to address this. The term 'ambiguity' is used to describe images which contain potentially confusing features alongside the desired subject. An example would be an image containing a number of unlabelled craters which could confuse the model during the training process.

1.3 Code Metadata

The code used in this project can be found in the GitHub repository [1]. The initial code base was forked from a PyTorch (Paszke et al. 2019) implementation of the YOLOv3 architecture by Erik Lindernoren [2]. This code base was extensively edited and added to in order to make it more amenable to a Google Colab based workflow and so that it could perform the tasks required for this project in a convenient and efficient way. These changes included:

- Reworking the main functions so that they could be imported and called in a Python environment, previously these used command line arguments.
- Calculating and Exposing statistics such as Precision, Recall and mAP.
- Printing relevant statistics during training and saving all statistics to a text file for plotting.
- Editing the loss function to simply sum the three loss metrics and remove their relative weightings.
- Outputting the predicted label file when performing a detection as well as the output image.
- Writing a number of utility functions which produced plots and performed data processing tasks.

The vast majority of the computation for this project was carried out using the Google Colab platform. This is an excellent platform which provides free access to CPU and GPU resources, making it an ideal tool for working with CNNs. It is recommended that when exploring the code base for this project, Google Colab is utilised. Jupyter notebook based workflows are available in the repository with further instructions on how to use them.

[1] https://github.com/acse-gcm20/YOLOv3-CDA

[2] https://github.com/eriklindernoren/PyTorch-YOLOv3

In addition to the Google Colab platform, Google Drive was also used extensively to store the large datasets and weights files which cannot be stored easily by the GitHub repository. Instructions on how to access and use the files on Google Drive are in the README and the example workflows.

2 Methodology

2.1 Mars Images

THEMIS (Christensen et al. 2004) - The Thermal Emission Imaging System (THEMIS) on board the 2001 Mars Odyssey spacecraft provides $100\,\mathrm{m}$ per pixel resolution images of the entirety of Mars.

HiRISE (McEwen et al. 2007) - The High Resolution Imaging Science Experiment (HiRISE) on board the 2007 Mars Reconnaissance Orbiter (MRO) produces high-resolution images of the Martian surface at $30\,\mathrm{cm}$ resolution.

2.2 Datasets

Benedix (Benedix et al. 2020) - A set of 1,762 unique craters across 889 images. The images are 416x416 pixel tiles of THEMIS (Christensen et al. 2004) images. The size of the dataset was increased four-fold by performing horizontal and vertical reflections to give a final size of 7,048 craters across 3,556 images. This dataset has previously undergone extensive manual checking to remove any ambiguous images meaning it is a very high-quality, reliable dataset. The dataset does not contain any degradation state information and so was only used to train the detection algorithm .

Robbins (Robbins and Hynek 2012) - A database of 384,383 Martian craters with statistical completeness for craters of diameter greater than 1 km. The revised 2014 version of this database (Robbins, Antonenko, et al. 2014) also provides classifications of degradation state for 20% of the craters. Four classes of degradation state were used by Robbins to classify the craters, Class 1 contained the most degraded craters and Class 4 contained the most pristine craters. Three key criteria were used to determine the degradation state of a crater, these were: the crater rim height, the surrounding ejecta and the interior infill. As craters degrade their rims become eroded, the impact ejecta is distributed away from the crater and erosional and aeolian infill occurs in the interior.

In order to utilise the Robbins database to train the detection algorithms, a crater labelling GUI was developed by Yinze Li [3]. This tool divided the THEMIS images [4] into 1 degree x 1 degree tiles. For each tile, the location and bounding box of each crater in the image was extracted from the Robbins database and written to a label file suitable for training. Processing the database in this way required substantial manual input to accurately align the bounding boxes of the craters and remove particularly ambiguous images, but was a useful method to generate more high-quality training images. Due to time constraints only the Aeolis, Amazonis, Elysium, Memnonia, Phoemicis Lacus and Tharsis quadrangles were processed - this represents six of the thirty quadrangles which cover the surface of Mars. These six quadrangles are all located within ± 30° latitude of the Equator.

Sweeney (Sweeney et al. 2018) - This dataset consists of rocky ejecta craters (RECs) from the InSight (Banerdt et al. 2020) landing site on Elysium Planitia. The RECs have diameters between 10 m and 1.2 km and are classified into five different levels of degradation. Unlike the Benedix data, this dataset was not collected for the purpose of training a CNN and so in order to use it for this purpose a substantial amount of pre-processing was required. The Sweeney data relates to craters in the high-resolution HiRISE images (McEwen et al. 2007) and so are orders of magnitude smaller than craters in the Benedix or Robbins datasets.

[3] https://github.com/acse-gcm20/Robbins-to-THEMIS

[4] https://planetarymaps.usgs.gov/mosaic/Mars/THEMIS_controlled_mosaics

2.3 Metrics

2.3.1 Training Loss

Three different loss metrics were combined to calculate the training loss value used for back propagation during training. These were: IoU box loss, object loss and classification loss.

The IoU loss (intersection-over-union) is a measure of how well a predicted bounding box matches up with the corresponding ground-truth label. An IoU of zero indicates the boxes have no overlap whereas a value of 1 indicates the boxes are perfectly aligned. During training and inference, a value greater than 0.5 was considered a correct prediction.

The object loss, or confidence loss, is a measure of how confident that model is that its detection is true. It is hypothesised that this value is particularly dependant on the ambiguity of the training set.

Finally, the classification loss relates to whether the classification of the prediction is correct or not. This value is not used in the single-class crater detection problem, but is very important in the multi-class classification problem.

For each batch, the three loss metrics were linearly combined by summing them with no relative weightings applied, and passed to the optimizer for back-propagation.

2.3.2 Precision and Recall

Precision, P, and Recall, R, are two very commonly used metrics to measure the ability of a model against a ground-truth test set. Precision, sometimes known as the positive predictive value (PPV), is a measure of the likelihood that a positive result is truly positive or not. Whereas, recall, or sensitivity, is a measure of how likely the model is to identify the true positives. Precision and Recall are calculated with the following formulae:

$$P = \frac{TP}{TP\ +\ FP} \quad R = \frac{TP}{TP\ +\ FN},$$

where, TP is the number of true-positive detections, FP is the number of false-positive detections and FN is the number of false-negative detections.

The mean average precision (mAP) of a model was used throughout the project as a metric to compare the success of different models. Average precision, AP, is the average of the precision score across all of the images in the test set for each class. The mean of the AP of the classes then gives a single value for the model which can be used for comparison. In the case of the single-class CDA problem, the mAP is equal to the AP because there is only one class to average over. In the case of the multi-class classifier, however, the AP score for each class is a useful indicator of in which classes the model is successful and in which it falls behind.

Precision-Recall curves were plotted to assess how the models performed at a range of confidence thresholds. The area under this curve is another useful metric for the success of the model, and a good indicator of the optimal confidence threshold to use.

2.4 Crater Detection Algorithm

The first stage of this project was to attempt to recreate a workflow similar to that of G. Benedix (Benedix et al. 2020) and develop a relatively simple crater detection algorithm trained on THEMIS images . Having been provided with the Benedix dataset, the YOLOv3 model was trained for 100 epochs. The first 50 epochs used a learning rate of 0.001 which stepped down to 0.0001 for the second 50 epochs. Improved performance was achieved by using initial weights pretrained on the COCO dataset (Lin et al. 2014).

Hyperparameters used for the training process include:

- Batch Size - 16
- Momentum - 0.9
- Weight Decay - 0.0005
- Optimizer - Adam

These hyperparameters were default to the YOLOv3 implementation used and were kept constant throughout the project.

A second model was trained once the Robbins database had been processed to see if training on this larger dataset of THEMIS images yielded a better performing model. The entire set of processed images was used in this case with all craters labelled as the '0' class for this single-class detection problem.

2.5 Degradation State Classifier

The second aspect of the project is to develop a model which can simultaneously detect and classify craters into one of four degradation state classes. These classes are based on those in the Robbins database where Class 1 represents the most degraded craters and Class 4 represents the most pristine classes (Robbins and Hynek 2012).

Before training the classifier, the Robbins dataset needed to be extensively pre-processed to prepare it for use. Firstly, the dataset of processed Robbins images was cut down to only include the images which contained craters which had been classified by Robbins - images containing only unclassified craters would only confuse the model. Then a new set of labels was generated by linking the ID of each crater to its degradation state in the Robbins database, previously all craters had the label '0' indicating the crater class for the CDA. All labels of unclassified craters were also removed. This dataset was split into training, validation and testing sets (81%, 9% and 10% respectively) using a random shuffle. The training and validation sets were used in three different ways to train three sets of weights in order to compare which method of training was most successful. The three models trained on each of the three training sets are summarised below:

1. **Full Robbins** - Uses the entire set of classified craters from the processed Robbins database with no further processing.
2. **Half Robbins** - The images in the Full Robbins dataset are ranked according to their object loss as calculated by the CDA. The top 50% of these images with the lowest object loss are used as a smaller, but less ambiguous dataset.
3. **Clean Robbins** - The Full Robbins dataset is cut down to include only the images which exclusively contain classified craters.

With no further processing, the database of classified Robbins craters contained 3661 craters across 1845 images. The distribution of craters across the four classes is shown in Table 1, where class 1 is the most degraded craters and class 4 is the most pristine craters. Many of the images in this dataset also contain craters which have not been classified - these are therefore considered to be unlabelled craters with respect to the classifier and will contribute to the ambiguity of the dataset. Figure 1 shows a comparison of an unambiguous image with a low object loss and an ambiguous image with a high object loss to demonstrate the features and unlabelled craters which could cause confusion when training a model.

The images which contained classified craters were ranked according to their 'object loss' as calculated by YOLOv3 - it was hypothesised that the 'object loss' of an image could be used to estimate its ambiguity i.e. how many unlabelled craters it contained. Figure 1 shows a comparison of two images, one with low object loss and one with high object loss. A key concern for training the model was that a large proportion of the images would be too ambiguous to be effective. Having ranked the images, the top 50% images with the lowest object loss were selected to be used as a new training set. The threshold of 50% was chosen arbitrarily to improve the quality of the dataset without reducing its size too much. Further work could be done here to experiment with different thresholds. This dataset contained 753 craters across 561 images and is known as 'Half Robbins'.

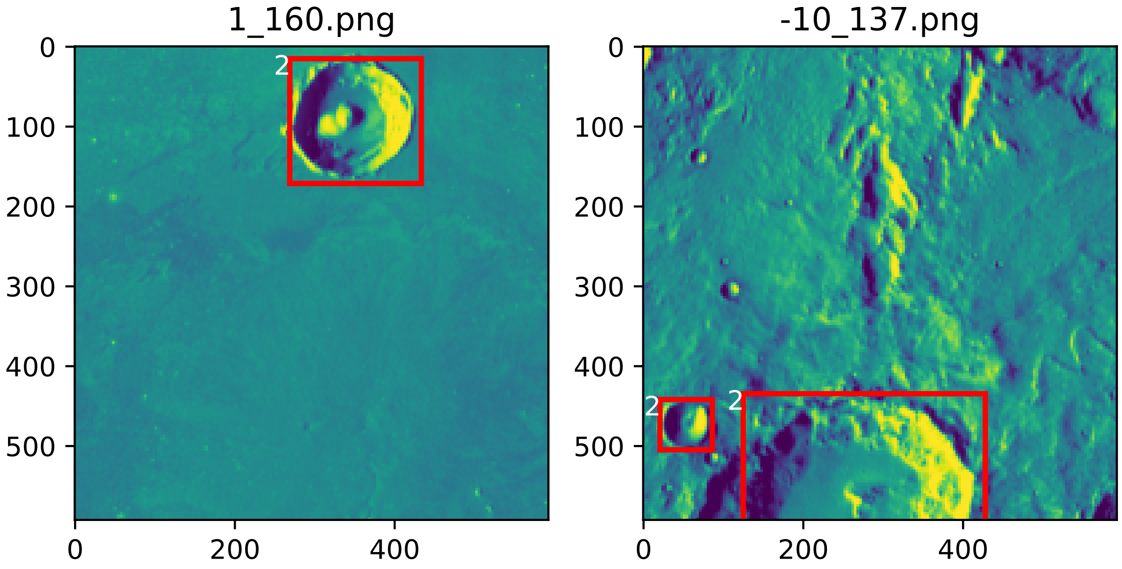

Figure 1: Comparison of two images with very low object loss (left) and very high object loss (right). The bounding box labels indicate the class of the crater - note that these are zero-indexed and so the label '2' refers to Class 3.

Cleaning the dataset down to just the images which exclusively contained classified craters drastically reduced the size of the dataset to just 163 craters across 139 images. The idea of this small subset is that it is completely unambiguous with respect to the classifier and so is the highest quality dataset available. This cleaned dataset and the model trained on it are known as 'Clean Robbins'.

	Crater Frequency		
Class	Full Robbins	Half Robbins	Clean Robbins
1	1086	337	61
2	580	156	45
3	1480	135	33
4	515	126	24
Total	3661	754	163

Table 1: Summary of the distribution of craters in each of the four classes for the three datasets used in developing the classifier.

A shapefile of the crater labels in the Sweeney dataset was provided and loaded into ArcGIS with the corresponding HiRISE images. Although this was an excellent dataset of craters classified according to their degradation state, the data only focused on rocky ejecta craters (RECs) and so there was a significant number of unlabelled craters in any image containing a classified crater. It was therefore decided not to proceed with this dataset as its ambiguity would be too high to train a model.

2.6 Data Augmentation

In the case of the Benedix dataset, the images had already undergone vertical and horizontal reflections to increase the size of the dataset by four - single horizontal flip, single vertical flip and both a horizontal and vertical flip (Benedix et al. 2020). This approach to augmentation was recreated on the Robbins images to also increase their frequency by four. Performing data augmentation in this way is an efficient way of artificially increasing the size of the dataset for training by generating more images which are just as realistic.

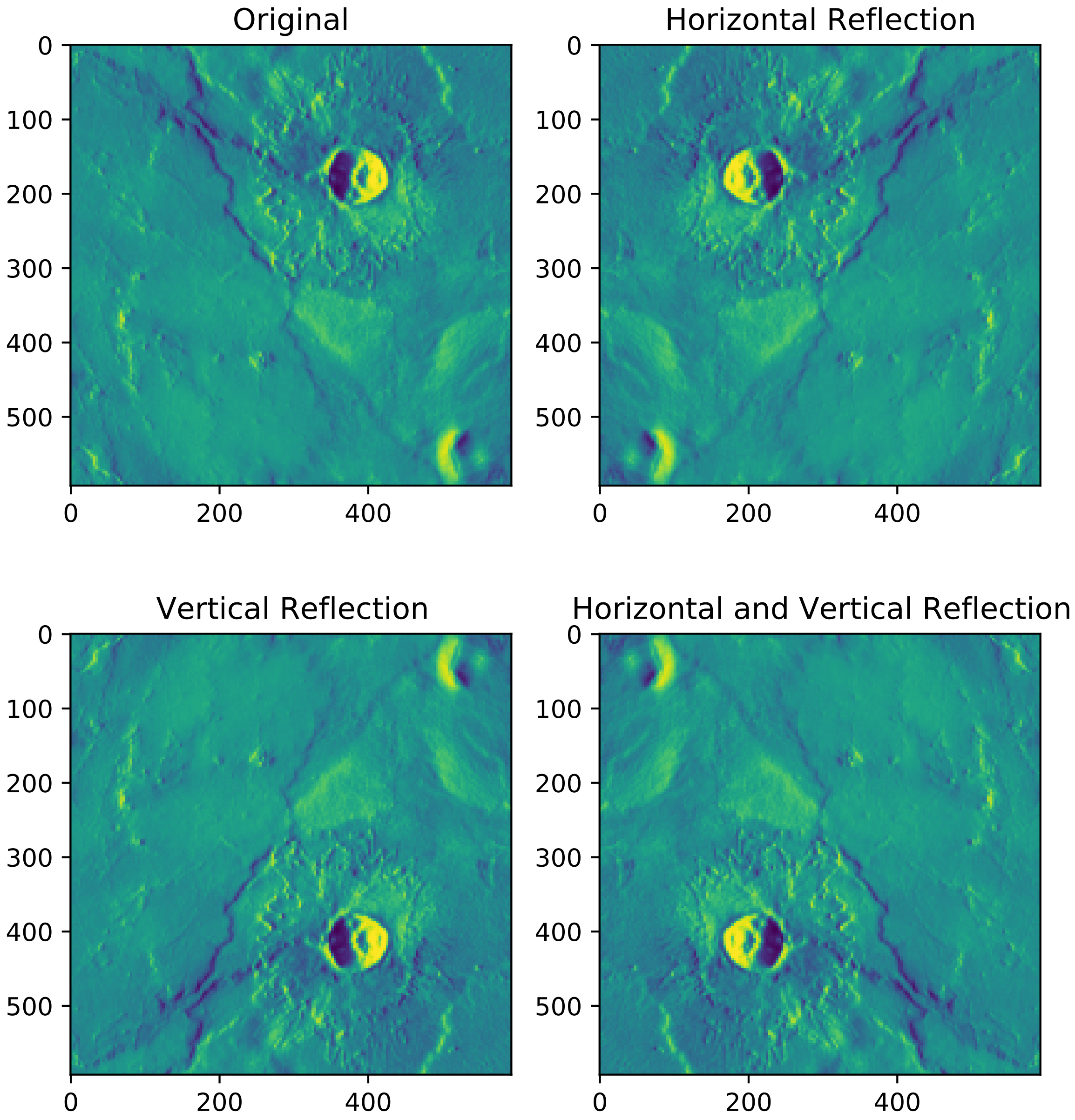

Figure 2: Demonstration of the data augmentation method use to increase the size of the dataset four-fold.

2.7 Testing

Test sets were extracted from each of the datasets and kept separate from the training and validation sets to ensure the models are not inadvertently trained on the images they will later be tested on. Three different test sets were generated in order to evaluate the single-class CDA. The first set was

10% of the Benedix dataset consisting of 356 images; the second was 10% of the processed Robbins dataset consisting of 1216 images; the third set was a union of the previous two and so consisted of 1572 images.

The primary test set used to quantitatively evaluate the different classifier models was a set of 56 images from the Clean Robbins set of images exclusively containing classified craters. Ideally a much larger test would have been used however there was a very limited supply of such images and 90% of them were used to train the model.

3 Results

3.1 Crater Detection Algorithm

The results of the three different crater detection models are summarised in Table 2. The key metric is the average precision on the 'Combined' test set. This ground-truth test set includes images from both the Benedix dataset and the Robbins dataset and so has the most relevance to potential applications of the model. Comparisons of performance on the 'Benedix' and 'Robbins' test exposes where performance is stronger or weaker.

	Average Precision (AP) on Test Set		
Model/Training Dataset	Benedix	Robbins	Combined
(1) Benedix	0.737	0.508	0.525
(2) Robbins	0.653	0.668	0.663
(3) Combined	0.700	0.700	0.699

Table 2: Results of the Crater Detection models against the three sets.

A comparison of the results of the three CDAs clearly shows that the best performing model was the one trained on both the Benedix dataset and the Robbins dataset, the training and validation losses for the training of this model are shown in Figure 3. This result is as would be expected as the model was exposed to all of the images in both training sets and so had the largest opportunity to learn.

A noteworthy result is the disparity in performance for the Benedix model between the Benedix test set and the Robbins test set. It is likely that this is due to the extent to which the images in the Benedix test were curated to keep only the clearest images, whereas the Robbins dataset, although also carefully curated, contains more images with more complex features and larger numbers of craters. This suggests that by training on the Robbins dataset we have built a greater degree of robustness into the model so it can accurately handle a wider range of images.

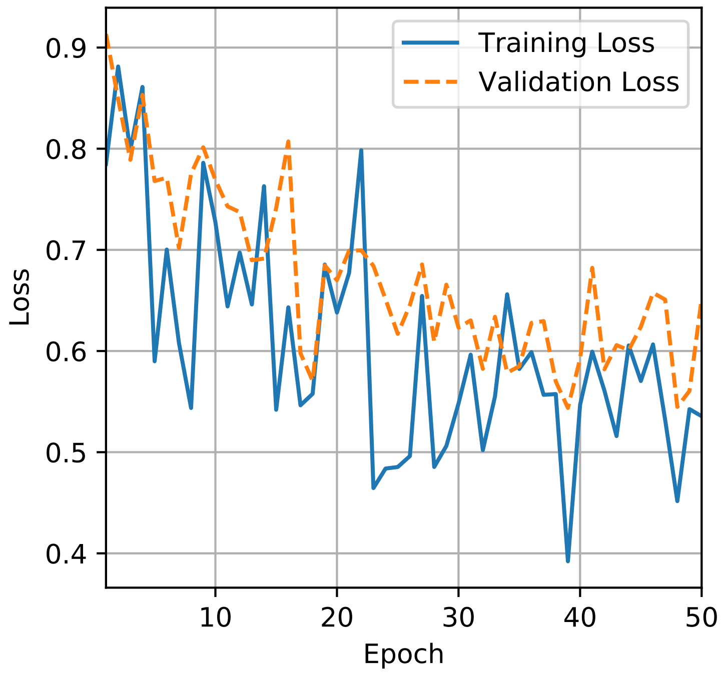

Figure 3: Training and Validation loss plot for the model trained on the combined dataset.

3.2 Classifier

The performances of the three different classifier models are summarised in Table 3. The results show the average precision per class as well as the mAP across all four classes.

	AP per Class				
Model/Training Dataset	1	2	3	4	mAP
(1) Full Robbins	0.357	0.115	0.434	0.848	0.439
(2) Half Robbins	0.103	0.083	0.647	0.589	0.356
(3) Clean Robbins	0.281	0.167	0.438	0.083	0.242

Table 3: Results of the classifier models against the test set.

These results show that the best performing classifier of craters by degradation state according to the mAP against the test set was the model trained on the entirety of the processed Robbins dataset, the training and validation losses for the training of this model are shown in Figure 4. The model trained on images exclusively containing classified craters was in fact the worst performing model; and the model trained on the top 50% images with the lowest object loss was in between. It is interesting to note however that the relative performance of each of the models is not consistent across each class - for example with respect to Class 3, the first model is actually the worst performing and the second model is the best performing.

Across all three models, class 2 is consistently very low scoring. This may be due to the fact that there are very fine differences in crater appearance between classes 1, 2 and 3 so there is lots of scope for confusion within the model.

It is surprising to note the disparity between the first and third models with respect to class 4. The model trained on the 'Full Robbins' dataset scored 0.848 whereas the 'Clean Robbins' model only scored 0.083. In both cases, the class 4 craters were the minority class, however the particularly poor performance of the 'Clean' model could be put down to it only having seen 24 examples of this class. Given the variety of appearances within a class, 24 craters is likely to be far short of an acceptable number of examples for the model to be able to confidently recognise it. In general, the performance of the 'Clean Robbins' version can be seen as a vindication of using a highly unambiguous dataset as although its mAP score was only slightly greater than half the score of the 'Full Robbins' version, it was exposed to less than a twentieth of the number of classified craters, as shown in Table 1.

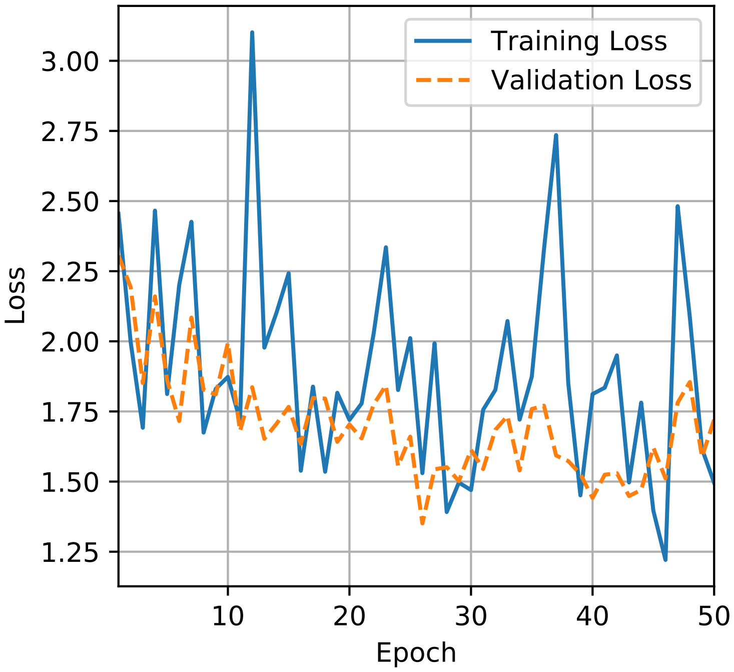

Figure 4: Training and Validation loss plot for the model trained on the Full Robbins dataset.

4 Discussion

4.1 The Ambiguity Problem

A key challenge throughout this project was dealing with 'ambiguous' data. In the context of this project we use the term ambiguous to describe images which contain a significant number of unlabelled craters. These kinds of images are problematic because during the training process the model may identify unlabelled craters but then be penalised as according to the ground-truth these detections are false-positive. This will therefore confuse the model as it is being penalised for what are in fact correct predictions. In most cases the unlabelled craters were craters with diameters less than 1 km which therefore were not included in the Robbins database. To address this problem and ensure the quality of the data, images containing more than five unlabelled craters were removed. In the case of the Benedix dataset this had already been performed, thus providing us with a very high-quality, unambiguous dataset. When processing the Robbins database we also made sure to remove ambiguous images to maintain a high-quality database.

The ambiguity problem was most significant when developing a dataset for training the classifier. This is because many large, well formed craters in the Robbins dataset did not have a degradation state recorded and so were in effect unlabelled with respect to the classification problem. There were a number of ways to address this.

The most direct method was to strip the processed Robbins dataset down to only the images which exclusively contained classified craters. This would mean that every crater seen by the model would be labelled and classified and the ambiguity would be negligible. Unfortunately, the classified craters were sparsely distributed so this reduced the dataset down to just 139 images.

Alternatively, because it was hypothesised that the 'object loss' of an image was a good indicator of ambiguity, the processed Robbins images were ranked according to their 'object loss' as calculated by the YOLOv3 CDA. A threshold of 50% was selected and only the least ambiguous half of the dataset was used as the training set. This does directly address the issue of the presence of labelled craters which are not classified but by inspection it was clear that these images generally contained fewer craters and were therefore less likely to contain unclassified craters.

A method that was not investigated in this project but which is a recommendation for further research is to apply a blur to the bounding boxes of labelled but unclassified craters. This could be an effective way to reduce the ambiguity of the dataset by eliminating the unclassified craters but without having to decrease the number of classified craters.

4.2 Two-Model Solution

A different approach to the classification problem which was not attempted in this project was to use a two-model solution in which a single-class CDA was used combined with a separate classification algorithm. One way in which this could work is for the CDA to identify the bounding boxes of craters in an image, these bounding boxes can then be cropped out and passed to the classifier which has no feature detection capabilities and simply classifies the whole image of the crater according to its degradation state. The advantage of this method would be that it would remove the issue of ambiguity in images and potentially make it far easier to develop a large training dataset.

By only passing the cropped bounding boxes of classified craters to the second model, this also allows for the easier application of other datasets such as the Sweeney dataset to train the classifier. Without cropping the classified craters, images from this dataset have far too many unlabelled craters in them to be useful as a training set.

4.3 Unequal Class Distribution

As shown in Table 1, the distribution of craters across the four classes was not even. The imbalance is significant but it was accepted that there were enough craters in each class for the purposes of this project. Recommendations for further work would include reducing the disparity between the majority and minority classes, perhaps through preferential augmentation methods in order to improve the accuracy of the model across the less represented classes. In addition to this, it is important to check that when splitting the training, validation and test sets, the distribution across the classes is consistent.

4.4 Regression vs. Classification

The current approach does not take advantage of the ordinal relationship of the degradation state classes. By using a classification approach, the fact that the four classes represent a spectrum of degradation is not appropriately considered. A regression solution could therefore be a better way of estimating the relative degradation states of craters by predicting degradation as a quantity rather than a class. This would also help in the utilisation of other datasets which use different patterns of classes - for example, the Sweeney dataset uses five classes where Class 1 contains the most pristine craters and Class 5 contains the most degraded craters, the inverse to the Robbins database.

4.5 Loss Weighting

It was decided to use a linear combination of the three loss metrics to calculate the batch losses, this decision was made as it was unknown which loss components would be most critical in the training process. It is possible that more successful models could be developed by weighting these losses differently, investigating this is a recommendation for further research. The original implementation of YOLOv3 that provided a code base for this project used weighting coefficients of 0.05, 1 and 0.5

for the box loss, object loss and class loss respectively when combining the three loss components into a single loss value.

5 Conclusion

A successful crater detection algorithm has been developed using the YOLOv3 architecture, trained on THEMIS images, to identify and extract craters from images of the surface of Mars. As shown in Table 2, the best performing version of this model achieved an mAP of 0.699 on a ground-truth test set of THEMIS images.

The ability of the YOLOv3 architecture to simultaneously detect and classify craters according to their degradation state was also investigated. Shown in Table 3, the best performing classifier that was developed achieved an mAP of 0.439. This demonstrates a proof-of-concept for automatically classifying craters by their degradation states using YOLOv3. The classifier was trained on a subset of the Robbins database which had been manually processed to generate a usable dataset. The primary limitation of the performance of this model is the size and quality of the training set. There was limited access to datasets containing craters that had been classified by degradation state, and many of these craters were in 'ambiguous' images alongside unlabelled craters.

Recommendations for further work include:

- Continue to process the Mars quadrangles in the same way to continue to grow the training datasets for all crater detection algorithms.
- Investigate a two-model solution which separates the crater detection algorithm from the degradation state classifier and potentially implements a regression solution.
- Investigate using different weighting coefficients of the loss components in the training process.

References

Albawi, Saad, Tareq Abed Mohammed, and Saad Al-Zawi (2017). "Understanding of a convolutional neural network". In: *2017 International Conference on Engineering and Technology (ICET)*. Ieee, pp. 1–6.

Banerdt, W Bruce et al. (2020). "Initial results from the InSight mission on Mars". In: *Nature Geoscience* 13.3, pp. 183–189.

Benedix, GK et al. (2020). "Deriving Surface Ages on Mars Using Automated Crater Counting". In: *Earth and Space Science* 7.3, e2019EA001005.

Christensen, Philip R et al. (2004). "The thermal emission imaging system (THEMIS) for the Mars 2001 Odyssey Mission". In: *Space Science Reviews* 110.1, pp. 85–130.

Hartmann, WK and IJ Daubar (2017). "Martian cratering 11. Utilizing decameter scale crater populations to study Martian history". In: *Meteoritics & Planetary Science* 52.3, pp. 493–510.

Lin, Tsung-Yi et al. (2014). "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer, pp. 740–755.

McEwen, Alfred S et al. (2007). "Mars reconnaissance orbiter's high resolution imaging science experiment (HiRISE)". In: *Journal of Geophysical Research: Planets* 112.E5.

Paszke, Adam et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pina, Pedro and Jorge S. Marques (2016). "Automated prediction of crater degradation degree". In: *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 4364–4367. DOI: 10.1109/ICIP.2016.7533184.

Redmon, Joseph (2013–2016). *Darknet: Open Source Neural Networks in C*. `http://pjreddie.com/darknet/`.

Redmon, Joseph, Santosh Divvala, et al. (2016). "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Redmon, Joseph and Ali Farhadi (2018). "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767*.

Robbins, Stuart J, Irene Antonenko, et al. (2014). "The variability of crater identification among expert and community crater analysts". In: *Icarus* 234, pp. 109–131.

Robbins, Stuart J and Brian M Hynek (2012). "A new global database of Mars impact craters 1 km: 1. Database creation, properties, and parameters". In: *Journal of Geophysical Research: Planets* 117.E5.

Sweeney, J et al. (2018). "Degradation of 100-m-scale rocky ejecta craters at the InSight landing site on Mars and implications for surface processes and erosion rates in the Hesperian and Amazonian". In: *Journal of Geophysical Research: Planets* 123.10, pp. 2732–2759.

Urbach, Erik R and Tomasz F Stepinski (2009). "Automatic detection of sub-km craters in high resolution planetary images". In: *Planetary and Space Science* 57.7, pp. 880–887.