Imperial College London

Department of Earth Science and Engineering

MSc in Applied Computational Science and Engineering

Independent Research Project
Project Plan

# Automated Crater Detection and Classification with Machine Learning

by

ZHAO, Sihang

Email: sihang.zhao21@imperial.ac.uk
GitHub username: acse-sz4521
Repository: https://github.com/ese-msc-2021/irp-acse-sz4521

Supervisors:

Prof. Gareth Collins
Dr. Marijan Beg

July 2022

# 1 Introduction

Deep neural network-based object detectors (ODs), like YOLO [1][2], can now rapidly process pictures and identify the entities using convolutional neural networks. In a vast scope of the scientific research fields, like planetary science, ODs can be applied to various object counting or classifying tasks. In particular, crater detection algorithms (CDAs) have helped scientists process thousands or even millions of pictures or real-time videos and detected millions of craters [3][4][5]. Several existing reviews have presented a comprehensive overview of Machine Learning-Based CDAs [6]. The number and sizes of those craters contain rich information like absolute age or relative age of the geographic unit [7]. Moreover, craters' shape (morphology) and their ejecta blankets can also provide clues to subsurface properties [8]. Thus it is a significant problem for the planetary scientists.

On the one hand, CDAs can automatically detect and classify the craters, with some features that human beings even cannot obvious. Computers will also never lose attention or feel tired, so their perceptual sensitivity to critical target events (in this case, craters) will not deteriorate [9]. On the other hand, With the development of aerospace technology and observation technology, we can observe more and more planets from more and more angles. So there are more craters in pictures that need to be processed and identified. The scope of craters that need to be detected is not merely vast; it is increasing and nearly infinite for all practical purposes. The above fact means that manually identifying all craters is an impossible task for human beings; that is another reason why CDAs are not only important but also necessary.

The state-of-the-art (SOTA) model YOLO V5 as a CDA has been proved successful in Themis data from Mars [3][10]. However, like humans, CDAs also make mistakes and produce incorrect results. For instance, CDAs may miscount some craters which exist and also count some similar patterns as craters. These kinds of mistakes can mislead scientists into getting the wrong conclusions, such as the ambiguous age of geological units and whether the great events occurred in a particular position of the planetary bodies. Especially, existing manually generated craters data sets like Robbins, [11] is not perfect, which cause difficulties for scientists training or testing their models.

Several different techniques were designed for testing deep learning systems, such as convolutional neural networks (CNN) models and recurrent neural networks (RNN) models [12], and there are also a number of techniques that were applied to test domain-specific applications, such as Q&A systems [13] and auto-driving systems [14][15]. However, there are no systematic methods for testing CDAs.

This paper aims to fill this critical gap. We design and implement a CDAs testing system, generate metamorphic images from the dataset for CDAs, make it a workflow, and then use the images which can cause the erroneous detection result to retrain the model; hopefully, we can get a better result.

# 2 Problem Description and Objectives

The domain problem is training a deep learning model to finish craters' detection and classification tasks. Section 2.1 will make a brief introduction to crater detection and classification tasks. This paper will use YOLO V5, a muti-classification object detection algorithm, as an example to show the progress of crater detection and classification tasks. Section 2.2 will explain how the CDAs testing system works and how to design it.

## 2.1 Machine Learning-based CDA

Machine learning-based Automated Crater Detection and Classification tasks take images or real-time images as input. The expected output is an annotated image with rectangles or circles surrounding

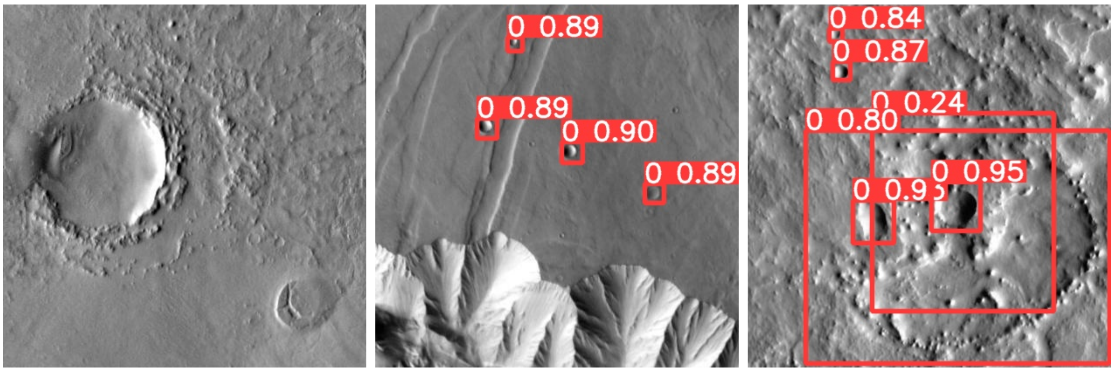


Figure 1: Example input of Thermal Emission Imaging System (THEMIS)[16] images and our output annotated by YOLO V5. The "0" in red labels means this object is detected as a "crater", and the numbers follows represent how confidence our model for identifying this object. The red rectangles are aim to cover the whole objects' bodies that have been detected.

every crater that appear in that image, including the large craters that exceed the edge of the image and small ones that can hardly be identified individually with the naked eye.

| class | x | y | width | height |
|---|---|---|---|---|

Table 1: Example label format of training data. The first column of digits represents the class of target objects. The second and third column of digits represents the relative location of the image. The fourth and fifth column of digits is the height and width of the rectangles that contain the craters.

In our cases, we will implement and extend the existing CDA based on YOLO V5. The first step to developing the CDA is training the Neural Network. We use the 889 THEMIS images as the training source. They are flipped horizontally and vertically into 3556 images. Each of them is labeled in the format shown in Table 1. As we only have one class, which is craters, the first columns are all zero. The relative location of the images are the x and y coordinates of the craters' central location. With those labels, CDAs are able to locate and identify the craters while training.

YOLOv5 uses the BECLogits loss function to calculate the loss of the objectness score, the class probability score is the cross-entropy loss function (BCEclsloss), and the loss function for the bounding box takes the GIoU Loss [17].

In crater detection tasks, we only have one class, so we can drop the class probability score in our CDA.

$$GIoU = IoU - \frac{|C/(A+B)|}{|C|}, \tag{1}$$

The GIoU Loss is used to calculate the loss of the bounding box. For two arbitrary boxes, A and B, we find a minimum closed shape C so that C can include A and B, then calculate the ratio of the area of C that does not cover A and B to the total area of C, and then use A Subtract this ratio from the IoU of B. Similar to IoU, GIoU can also be used as a distance, and loss can be expressed as:

$$L_G IoU = 1 - GIoU, \tag{2}$$

## 2.2 CDAs testing system

Inspired by the wide range of success of metamorphic testing (MT), which can detect conventional software errors [18][19]. In this work, The CDAs testing system will generate a metamorphic crater pattern that can be detected and identified as a true crater. Even if the existing data in the training set is imperfect, the correctness of the actual output is hard to determine; we can still evaluate the output by making a comparison between our actual output and the expected output.

This testing method can help us find the recognition failures [20], which represent CDAs are fail to recognize an existing object or treat an arbitrary region on the image without any objects as an "object."

With this test system, we can easily evaluate whether a model is good or not, even if we don't have a perfect dataset.

## 3 Progress to Date and Future Plan

Currently, I have finished the literature review. After the group discussions, I used a completely independent test set and trained several basic models based on YOLO V5 with different hyperparameters. The models' loss converged very well. When the batch size was 16 and the epoch was set to 200, the mAP_0.5 exceed 0.98 and the recall rate can exceed 0.92.

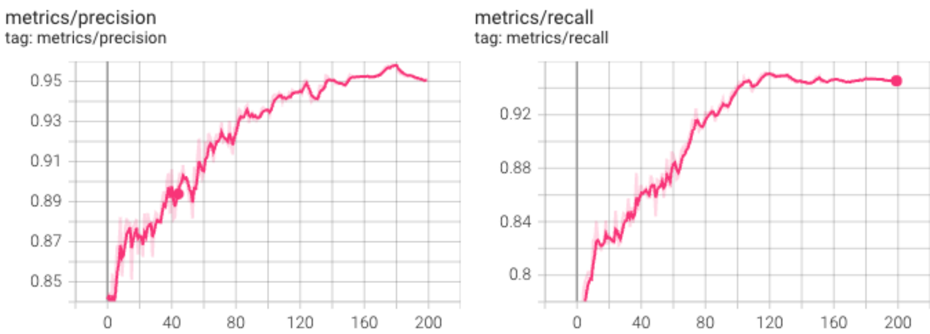


Figure 2: The precision and recall when epoch is 200 on the initial model.

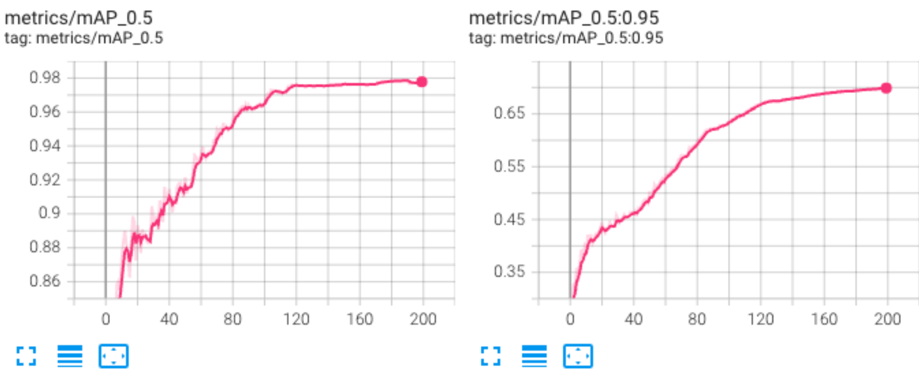


Figure 3: The mAP_0.5 and mAP_0.95 when epoch is 200 on the initial model.

As result samples showed in Figure 4, our model can successfully detect the overlapping craters like the situation in first image, and distinct the "none crater" circles like second image. On the right-bottom, the geographical patterns with different opposite shadow are not identified as craters.

In this state, there are still lots of work to do. First of all, I need to take a deeper look into YOLO V5 so that I can modify the model and make improvements to it. Secondly, because the training data is imperfect, I want to design and implement the CDAs testing system mentioned in the previous section and retrain the models. Last but not least, I need to prepare and leave enough time for my final report and presentation. Thus I made a GANTT chart (Figure 4) to show my schedule.

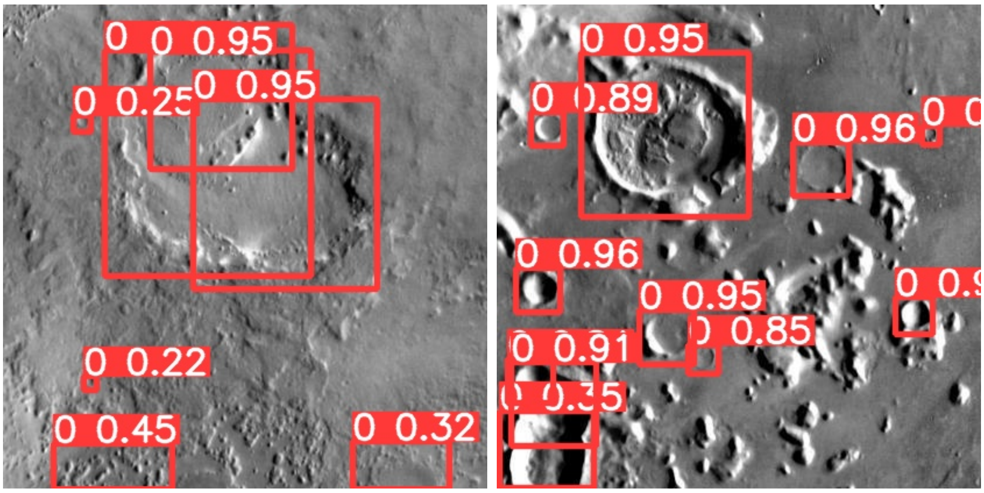


Figure 4: Example outputs annotated of my model.

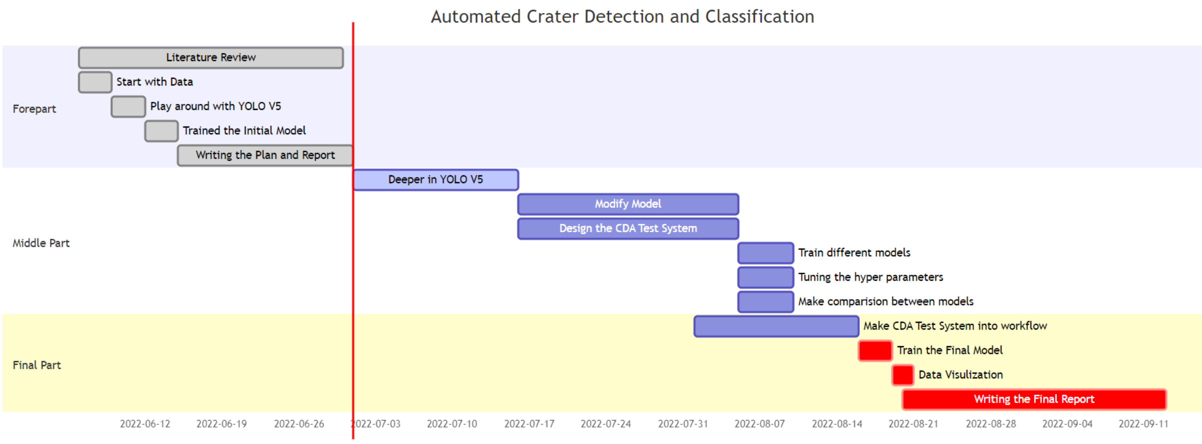


Figure 5: Time schedule for this project. The gray parts are finished tasks, blue parts are activating, and the red part is the important tasks.

## References


[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," pp. 779–788, 2016.

[2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[3] G. Benedix, A. Lagain, K. Chai, S. Meka, S. Anderson, C. Norman, P. Bland, J. Paxman, M. Towner, and T. Tan, "Deriving surface ages on mars using automated crater counting," *Earth and Space Science*, vol. 7, no. 3, p. e2019EA001005, 2020.

[4] C. Norman, J. Paxman, G. Benedix, T. Tan, P. Bland, and M. Towner, "Automated detection of craters in martian satellite imagery using convolutional neural networks," in *Planetary Science Informatics and Data Analytics Conference*, vol. 2082, no. 2081, 2018, p. 6004.

[5] C. Lee, "Automated crater detection on mars using deep learning," *Planetary and Space Science*, vol. 170, pp. 16–28, 2019.

[6] D. DeLatte, S. T. Crites, N. Guttenberg, and T. Yairi, "Automated crater detection algorithms from a machine learning perspective in the convolutional neural network era," *Advances in Space Research*, vol. 64, no. 8, pp. 1615–1628, 2019.

[7] C. A. T. W. Group *et al.*, "Standard techniques for presentation and analysis of crater size-frequency data," *Icarus*, vol. 37, no. 2, pp. 467–474, 1979.

[8] N. Warner, M. Golombek, J. Sweeney, R. Fergason, R. Kirk, and C. Schwartz, "Near surface stratigraphy and regolith production in southwestern elysium planitia, mars: implications for hesperian-amazonian terrains and the insight lander mission," *Space Science Reviews*, vol. 211, no. 1, pp. 147–190, 2017.

[9] J. E. See, S. R. Howe, J. S. Warm, and W. N. Dember, "Meta-analysis of the sensitivity decrement in vigilance." *Psychological bulletin*, vol. 117, no. 2, p. 230, 1995.

[10] A. Lagain, K. Servis, G. Benedix, C. Norman, S. Anderson, and P. Bland, "Model age derivation of large martian impact craters, using automatic crater counting methods," *Earth and Space Science*, vol. 8, no. 2, p. e2020EA001598, 2021.

[11] S. J. Robbins, I. Antonenko, M. R. Kirchoff, C. R. Chapman, C. I. Fassett, R. R. Herrick, K. Singer, M. Zanetti, C. Lehan, D. Huang *et al.*, "The variability of crater identification among expert and community crater analysts," *Icarus*, vol. 234, pp. 109–131, 2014.

[12] X. Du, X. Xie, Y. Li, L. Ma, Y. Liu, and J. Zhao, "Deepstellar: Model-based quantitative analysis of stateful deep learning systems," in *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2019, pp. 477–487.

[13] S. Chen, S. Jin, and X. Xie, "Testing your question answering software via asking recursively," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2021, pp. 104–116.

[14] R. B. Abdessalem, A. Panichella, S. Nejati, L. C. Briand, and T. Stifter, "Testing autonomous cars for feature interaction failures using many-objective search," in *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2018, pp. 143–154.

[15] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th international conference on software engineering*, 2018, pp. 303–314.

[16] P. R. Christensen, B. M. Jakosky, H. H. Kieffer, M. C. Malin, H. Y. McSween, K. Nealson, G. L. Mehall, S. H. Silverman, S. Ferry, M. Caplinger *et al.*, "The thermal emission imaging system (themis) for the mars 2001 odyssey mission," *Space Science Reviews*, vol. 110, no. 1, pp. 85–130, 2004.

[17] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 658–666.

[18] T. Y. Chen, F.-C. Kuo, H. Liu, P.-L. Poon, D. Towey, T. Tse, and Z. Q. Zhou, "Metamorphic testing: A review of challenges and opportunities," *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–27, 2018.

[19] T. Y. Chen, S. C. Cheung, and S. M. Yiu, "Metamorphic testing: a new approach for generating next test cases," *arXiv preprint arXiv:2002.12543*, 2020.

[20] S. Wang and Z. Su, "Metamorphic object insertion for testing object detection systems," in *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2020, pp. 1053–1065.