Segmentation of Dune Crestlines Using Convolutional Neural Networks

Timothy Chee Cheng Lui*, Colin Marvin, Mathieu Lapôtre, Jef Caers
Department of Earth and Planetary Sciences, Stanford University
397 Panama St, Stanford, CA 94305
*timmylui@stanford.edu

Abstract

We use convolutional neural networks to segment dune crestlines across 5 Marian dune fields. Automating this tedious digitization process can save a lot of time and effort, and can be used to map whole planetary bodies. In extension, this can let us calculate other morphodynamic properties of interest that can inform on recent environmental changes. We show how our work is unique to the state of the art progress in the field of mapping dunes for planetary bodies, being the first to use Mars, gray scale images, and linear-like features all at the same time. We use a sigmoid, U-Net with an "EfficientNet 3B" backbone, and combine the Dice Loss and Focal Loss to make our loss function more robust. We exclude one full dune field as a testing set to more rigorously assess our models performance on unseen data. And we show how our model is able to qualitatively do a good job on identifying and segmenting dune crestlines.

1. Introduction

On planetary surfaces like Earth, Mars, or Saturn's moon Titan, we can find dune fields that form from the interaction between wind and sediment [3]. Properties from these dune fields can tell us about the conditions that shape them, such as wind speed and direction, sediment size and density, sediment availability, geometry of the area… [7]. The manual quantification of these dunes can require a lot of time and outlining them is a very redundant process [18]. The purpose of this paper is to explore the ability that convolutional neural networks (CNNs) can have of detecting and segmenting dune crestlines, which would then allow us to quantify environmental conditions of these planetary surfaces.

Being able to segment dune crestlines on planetary surfaces could help us automatically map dune fields without additional human effort. This would allow us to more easily quantify morphodynamic disequilibrium. Morphodynamic disequilibrium occurs when bedforms (such as ripples and dunes) are not in equilibrium with local environmental conditions. We can identify morphodynamic disequilibrium through interactions between dunes. If two dune crestlines are close to each other, the dunes are interacting. This could imply that there have been recent shifts in the environment (changing wind direction, or new wind speed, or amount of sediment supply…), and the higher the number of interactions, the more recent the environmental condition change. This is because the dune fields have not been able to mature and stabilize with the new environment [5, 15]. This would allow us to identify environmental change remotely on Earth and other planetary bodies.

The goal of this project is to develop CNNs that take in an image of a part of a dune field, and output an image that outlines the crestlines. In extension, the CNN can map dune crestlines across all sorts of planetary bodies, and be a part of the pipeline to quantify morphodynamic disequilibrium across the Solar System.

2. Related Work

The use of computer vision for the detection of dunes is not necessarily new, but it is a growing realm. Various data science, non-CNN methods have been explored for dunes, including the use of random forests and support vector machines to detect image patches with dunes on Mars [4], or Haar wavelets to create dune bounding boxes [1]. Their focuses were on dune detection, not crestline segmentation, but there is one non-CNN method for segmentation involving mapping crestlines using self organizing maps [8]. It was interesting to see the claimed good performance especially since they don't use convolutions, just fully connected neural networks. Unfortunately, I am not convinced of the results since the performance metrics fail to focus on rare crestlines. A dune field may have 99% non-crest pixels and 1% crest pixels. Failing to focus on and prioritize performance on the rarer crestlines means performance will likely be good, since it's easy to predict the majority class of no-crestline.

There have also been a good number of papers that use CNN methods for the remote sensing of dunes, and they have shown good performances for diverse problems.

Mostly they revolve around dune detection [22, 2, 13] and classification [22, 13]. Notably, there's a paper focused on dune outlining, which is a similar task to dune crestline segmentation [18]. We also find that Shumack et al. [19] try to segment dunes and have good success utilizing U-Nets. But fundamental differences exist that make our problem unique. Rubanenko et al. [18] work on barchan dunes, which are "blobbier/rounder", whereas our longitudinal dunes are linear. Shumack et al. [19] show success in longitudinal dunes, but they use RGB images from Earth, whereas our images are gray scale from Mars, bringing to question if maybe the lack of extra information from three channels is detrimental. Additionally, they focus on segmenting the whole dune (which can cover multiple pixels), whereas the focus of our project involves just the dune crestline which is 1 pixel wide. There are other domains in which CNNs have been applied to segment skinny, linear features [23, 10]. Our research involves the similar challenge of trying to predict skinny, linear features but using gray scale in a geological domain using remote sensed images from Mars.

3. Methods

For the segmentation task we utilize a U-net, first introduced by Ronneberger et al. [17]. The U-net, like many other CNNs, involves layers of convolutions, activations, and max-poolings, but what makes it unique involves the upsampling step. Our input is an image of 256x256 pixels. Since this is a segmentation problem, we are not outputting a single class like a classification problem; we are outputting an image of the same size that highlights what pixels are crestlines. Hence after the downsampling we use a series of upsampling that will return the image to its original size, and for each pixel it will return a class: "crestline" or "no crestline".

We also utilize the "EfficientNet B3" backbone, introduced by Tan and Le [21]. A backbone can be understood as a portion of a pre-trained network that has learned useful features (or structures) to extract (or look out for) from images. Then we can use this backbone to apply it to other datasets, which means we can expedite the learning process and focus only on the final layers of the CNN. Figures from their paper show how EfficientNet is able to have very high accuracies at lower required parameters, making it a very efficient backbone. What will be interesting to assess is how well this backbone will translate to our dataset. Common image problems involve "blobbier/rounder" things, say cats, cookies, flowers... Whereas our images are always going to be linear and skinny, around 1 pixel wide. This brings to question if maybe other backbones are optimized for linear/skinny features extraction.

We utilize the package "Segmentation Models" made by Yakubovskiy [9], a Python library that helps users set up segmentation problems. In order for our CNN to learn, we utilize a loss function which combines dice loss and focus loss, which is proposed in one of the examples of Yakubovskiy.

To start, "Dice Loss", inspired by the Sørensen–Dice coefficient developed by Sorensen [20] and Dice [6], and described by Li et al. [11], is a metric that considers both precision and recall, and is associated with the F1-score.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

Precision can be interpreted by: the things that your model predicts are crestline, what proportion are actually crestline?

$$Recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

Recall can be interpreted by: out of all the pixels that are truly crestlines, what proportion was your model able to correctly classify as crestline?

Once we have precision and recall, we can calculate the F1-score through

$$F1 - score = \frac{2 * precision * recall}{preciison + recall}$$

The F1-score balances precision and recall, ensuring that our model does not get comfortable with predicting too many pixels as crestline, but also giving it pressure to correctly encapture all crestlines. This F1-score, also known as the Sørensen–Dice coefficient, can be slightly altered to ensure no quirks related to multiplying or dividing by 0. We do this through adding a γ that can be set to something small like 1, to both numerator and denominator.

$$Sørensen–DiceCoefficient = \frac{2 * precision * recall + \gamma}{preciison + recall + \gamma}$$

To obtain the loss, for each class of the dataset we are going to calculate the Sørensen–Dice coefficient, subtract that from 1, and average it out across equally weighted classes, N. This will allow for all classes to be of equal weight, and not bias the model towards being able to better predict a specific class solely out of more abundance. As critiqued of [8], our dataset has a huge proportion of pixels labelled "no crestline", and very few "crestline". It is important to allow for different classes to be of equal weight to avoid this issue of imbalance. For final dice loss, we also add a square to the denominator which helps reach convergence faster [16].

$$DiceLoss = \frac{1}{N} \sum_{class} \left[1 - \frac{2 * precision * recall + \gamma}{precision^2 + recall^2 + \gamma} \right]$$

Next, we need to understand the "Focal Loss" [12]. The inspiration for focal loss surged from a desire to deal with class imbalance. It starts off as cross entropy loss, but through a parameter γ, it will relatively weigh the loss of certain classes more importantly. Classes that are easy to predict will have the loss reduced, whereas classes that are hard to predict will have their loss weighed more heavily. This will make the model emphasize predicting classes that are hard to predict. This is yet another way that we can focus on making the rarer class ”crestline” the important one to predict correctly.

$$CrossEntropyLoss = \begin{cases} log(p) & \text{if } y = 1 \\ log(1-p), & \text{otherwise.} \end{cases}$$

$$FocalLoss(p_t) = -(1-p_t)^{\gamma} log(p_t)$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise.} \end{cases}$$

As seen, parameter γ affects the first component, and the higher that γ is, the stronger the effect of reducing the loss of well predicted classes. Choosing a γ of 0 means normal cross entropy is used.

Hence, the combined loss function we described earlier simply involved adding the Dice Loss and the Focal Loss.

$$Loss = DiceLoss + FocalLoss$$

Benefits of our loss function include the fact there are various mechanisms involved that will help manage the imbalance that comes from images being mostly ”not crestline” with rare ”crestline” classification. In extension, by adding two losses, we hope we can benefit in similar ways to how ensembling can make decision trees more robust by making random forests. By having multiple ”opinions” on how bad, how much loss, a certain prediction has, we gain a more robust and diverse understanding of how truly bad the prediction is.

4. Dataset and Features

Images are obtained from the Context Camera on the Mars Reconnaissance Orbiter, a satellite orbiting Mars that captures gray scale images with resolution of around 5m/px [14]. We have collected 5 images of barchanoidal dune fields from this dataset involving circles of diameters ranging from 700 to 2,000 pixels. Dune crestlines for these images were digitized by collaborators Marvin et al. [15]. Example can be seen in Figure 1.

To create our dataset we chose to take each gray scale image and digitized crestline pair, and slide a 256x256 pixel window across them. This created corresponding images of the real image data and the true mask of what the crestline is. The stride for our moving window was 128 pixels, meaning we allowed for the window to overlap 50%. This left us with various images for 5 different dune fields in Mars.

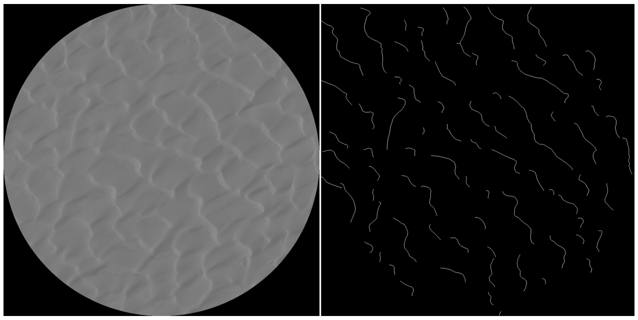

Figure 1. Pair of gray scale image from Mars dune field on left, with corresponding digitized dune crestlines on right.

Images within each of the 5 dune fields will be inherently similar, given they will have same wind directions and hence same angles of trajectory, they were taken at the same time and hence have the same solar angle and shadows... The goal of this model is to predict segmentation performance on a dune field the CNN was not trained on. For this reason, we will use an nontraditional system of train-test split that will more fairly represent how the model would do on data it has never seen before.

We will remove one of the dune fields, and use all the images from that dune-field as a testing set. With the remaining 4 dune fields, we will randomly select 80% of the images and use them as our training set, and the other 20% will be our validation set. This resulted in a training set of 316 images and validation set with 79 images which come from 4 different dune-fields. The testing set, the randomly selected image to be removed from the pipeline completely, was composed of 20 images. (In one of our later experiments we loop over each of the 5 dune fields, keeping one of them out for testing to see how sensitive our CNNs performance is to this random omission of a dune field).

We were able to artificially increase the training dataset's diversity through a series of image augmentations. Horizontal flipping and 360 rotations were introduced randomly which should help deal with issues regarding orientation, wind direction, and shadow direction. We also minimally augmented stretch and skew. We implemented contrast and brightness augmentations that could help diversify across images taken at different times of day, with different shadow and brightness impact. We also applied batch normalization to optimize the process.

5. Experiments/Results/Discussion

Early steps in our exploration involved testing out the various common key hyperparameters in computer vision problems. Notably, we ended up sticking with the sigmoid activation function for pragmatic reasons. We know that other activation functions should function better for various reasons (vanishing or exploding gradients for example), ReLU never worked despite countless attempts. Then, hoping that the problem was the 0 gradient on ReLU's negative side, we tried other alternatives like Leaky ReLU and GeLU, but all of them were resulting in poor performance. Given how long the models took to run, at some point, given that our early implementations of simple sigmoid worked, we decided to not try to fix what is not broken, and kept sigmoid.

Testing for hyperparameters like learning rate and optimizer was relatively simple. Adam was one of the more advanced optimizers (compared to stochastic gradient descent, or with momentum...), so given that it worked well we did not try to play around with it too much. For learning rate, we did what might be described as a one dimensional grid search by tweaking exponentially (eg, 1e-3, 1e-4, 1e-5, into 3e-4, 1e-4, 3e-5...) landing us at a comfortable 7e-5 selecting based on performance on validation set. Batch size was 8, chosen as a relatively random, comfortable number that did not make runtime too slow.

We did not think cross validation was necessary since we were already dealing with a bigger systematic problem of the grouping of the data (4 dune fields for training). A possibility is to further use one of the dune fields as validation and run a cross-validation loop with 3 dune fields as training, but given that some dune fields were smaller, we feared the training set might have become too small.

In addition to using our loss function as previously described (a combination of Dice Loss and Focal Loss), we can assess performance through the metric of IOU, Intersection Over Union. IOU takes a predicted mask (what the CNN called crestline) and a true mask (what the geomorphologist expert called crestline) and assess how similar these are. The intersection, the overlapping area between the two masks, is then divided by the union, the total area covered between the two masks. If the two masks are very overlapping, the intersection will be very high and hence the IOU will be high. If the true crestline is small, and the CNN prediction covers a very high area, even if intersection is high, the union will be very high and IOU will be low. Our figures will show IOU as an additional metric that will help us intuitively assess performance on a scale from 0-1.

Even though many small experiments, tests, and results could be showed, due to a lack of space we will simply show our ultimate final model. This one combines the best of everything we had, and was run for 500 epochs to show the maximum potential of what we could achieve. IOU and loss scores can be seen in Figure 2.

Unfortunately the scaling of the model loss is hard to see, but the value of loss is constantly decreasing with more epochs for both curves. Regarding IOU score, we can see how training performance keeps going higher with more epochs, and so does our validation set (mislabelled as "test"). There is a constant gap which suggests overfitting, but also I do not believe our model has completely plateaued and reached its maximum train potential. There is a potential that with more epochs, the validation performance more approaches training performance, and there is less overfitting. Unfortunately, running these 500 epochs took 24 hours and a plateau was still not reached. More qualitative understanding of performance can be seen when looking at Figures 3 and 4.

Figure 3 shows the CNN performance on our training set. Each trio of images is composed of, on the left, the gray scale Mars image, in the middle, the geomorphologist identified dune crestline mask, and on the right, our CNN's predicted dune crestline mask. Off visual analysis, I would say the performance of the CNN is quite excellent. Most crestlines are mapped pretty well, with a few errors here and there. Every now and then, there is a crestline that is missed, or connected to the wrong series of crestlines.

We can see an interesting effect of our augmentations by looking at the true masks. Our true masks are skinny, linear features that represent a line in space that has to be projected onto a pixel grid. While the original line is continuous, we can see that after rotation, that reprojection of the line leads to artifacts such as small gaps within what should be a continuous crestline. It is pleasing for me to see that the predicted mask does not predict gappy crestlines, since that would mean it is overfitting to an artifact that should not be there.

In extension, we can see that the predicted masks are thicker, with a width of around 3 pixels. I interpret this as a way in which the machine was able to become more robust to some of the artifacts that arose from augmentations. Since some pixels were randomly being dropped to the sides, or in between the crestline, maybe the CNN optimized performance and loss score by predicting thicker lines, hence being able to encompass these artifacts. I feel the use of the focal score, that emphasized performance on harder to predict classes, helped make this happen.

Figure 4 shows the CNN performance on our testing set. All the images come from a dune field that the CNN has never seen, was never trained or validated on. This represents a harder testing set to perform well at, since this dune field might have quite unique properties our augmentation could not address. For example, certain shadow angles cannot be augmented for through just brightness and contrast alterations. Final loss for this testing set was 0.71, mean IOU score was 0.19, mean F1-score was 0.32. IOU score and F1-score scale from 0-1 with 1 meaning really good scores for both of them. Compared to other vision projects, these scores seem low, but again, when we qualitatively assess them, we see that this low performance might not be that big an issue.

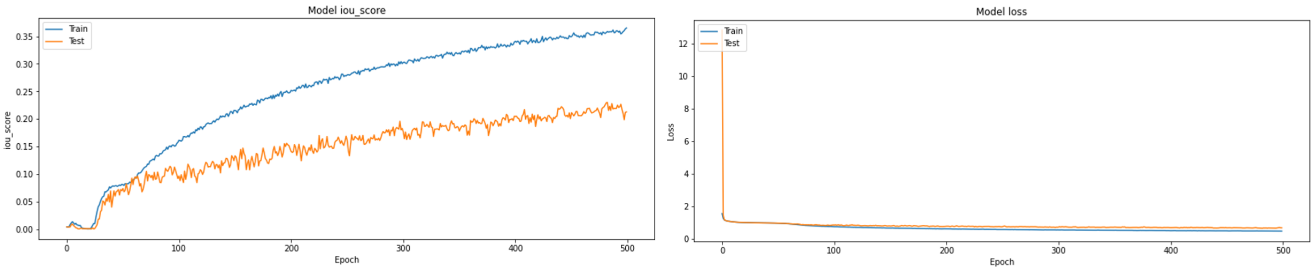

Figure 2. IOU Score and Loss Curve of 500 epoch experiment. Note there is an unfortunate mislabelling, the orange line is actually validation set performance and not test.

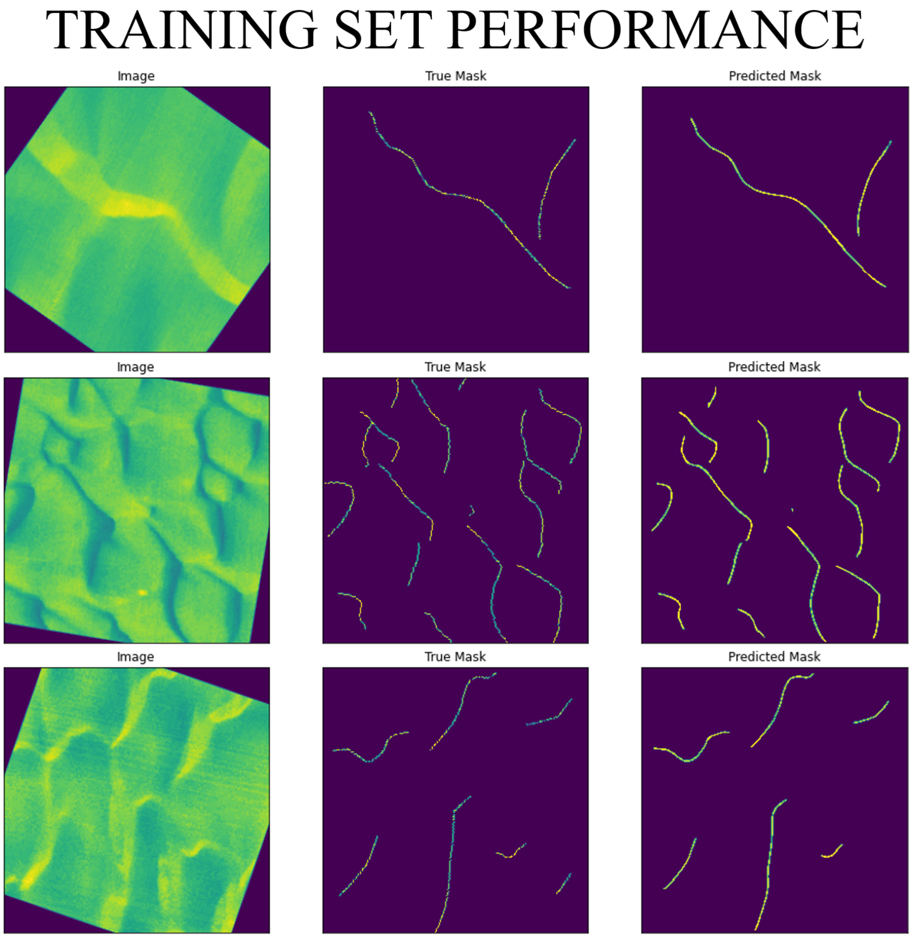

Figure 3. 3 examples of CNN performance on training set. Left image is the gray scale Mars image, middle image is the geomorphologist identified dune crestline mask, right image is the CNN predicted dune crestline mask.

Qualitatively, the performance seems good to me. Some crestlines have been split up into smaller ones, but maybe there are post-processing algorithms that can help transform these initial crestlines into connected ones, reaching better scores. As mentioned before, the model predicts thicker crestlines (3 pixel width) whereas the original crestline is 1 pixel wide, which is a source for the lower IOU and F1-score. Ultimately, this is ok. I would rather have a model that had lower IOU score but could accurately recall crestlines, than a model with high IOU score but misses some crestlines.

What I'm most focused on most recall of all crestlines present. As mentioned in the beginning, the purpose of mapping these crestlines is to derive further morphody-

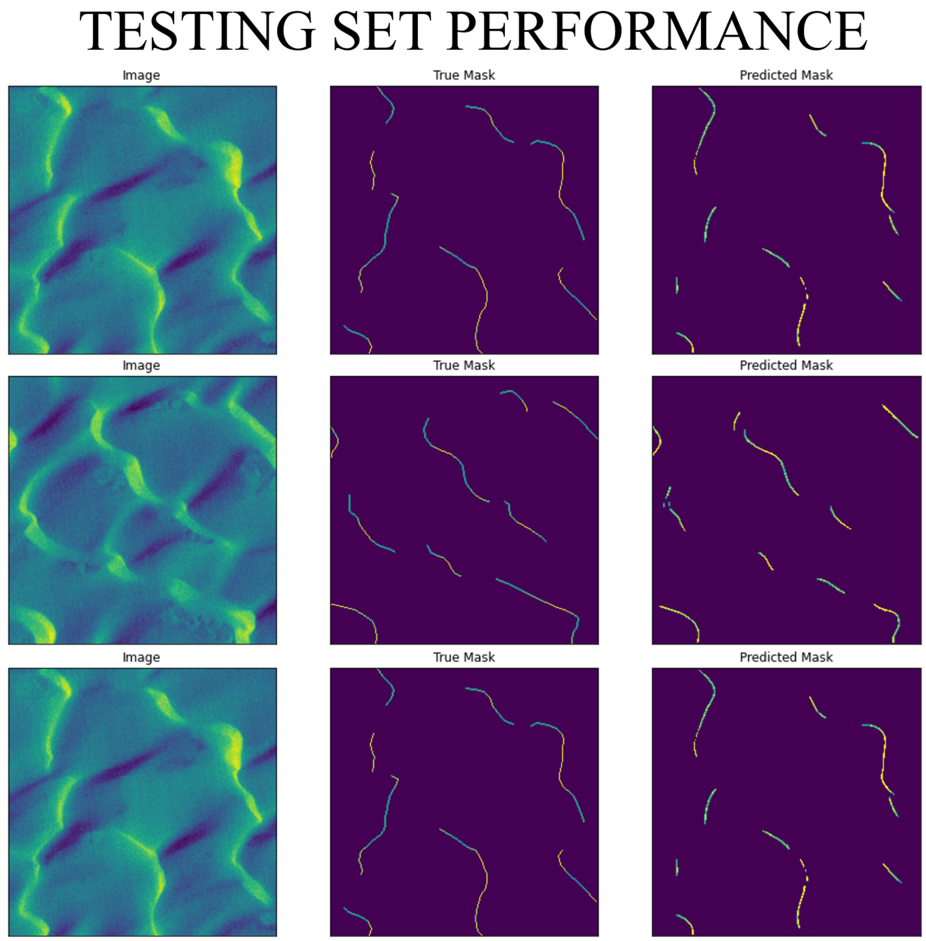

Figure 4. 3 examples of CNN performance on testing set. Left image is the gray scale Mars image, middle image is the geomorphologist identified dune crestline mask, right image is the CNN predicted dune crestline mask.

namic properties, such as dune spacing or interactions per area. Calculating dune spacing, the distance between dune crestlines, with 1 extra pixel on either side is not really going to affect the calculation. Whereas if the model were to skip a whole crestline and not detect it, that would majorly affect these further geomorphological calculations we want to perform.

When a crestline is split into 2 smaller parts, that does not really affect dune spacing calculations since they are part of the same ”line”. We care about recalling every dune, because say we cannot detect a dune crestline that is sandwiched between two dune crestlines. Our dune spacing calculation would be doubled from this error, whereas if we recall all dunes, but they are cut at random pieces, our dune spacing calculation would still be pretty accurate.

Potentially what this could mean is that the current loss function could be improved. Finding a way to make the loss say, ”it is ok to predict one pixel to the left or right of the true crestline” could help it not prioritize making the crestline skinnier and more accurate in that sence. Additionally, finding a way to make loss prioritize ”recall” of dunes, making sure every dune crestline in the image is somewhat detected, would be good for the ultimate purpose of the CNN (helping quantify morphodynamic properties).

Figure 5 shows a bonus figure including the previously hinted at experiment that looked at assessing robustness of using 4 training dune fields for 1 testing dune field. We can see how dune fields 1 and 2 tend to perform pretty well. Interestingly, dune field 5 does pretty bad, but every now and then a dune can be detected. After analysis, we can see that dune field 5 dunes happen at a very different scale, almost as if the images were zoomed in. This suggests to me that one of the augmentations that could increase performance on dune field 5 is zooming in and out.

Dune field 11 does incredibly bad, and we can see that the dune field looks completely different. This is one of the cases where we can see how shadows can be particularly complex based on the angle of the sun when the image was

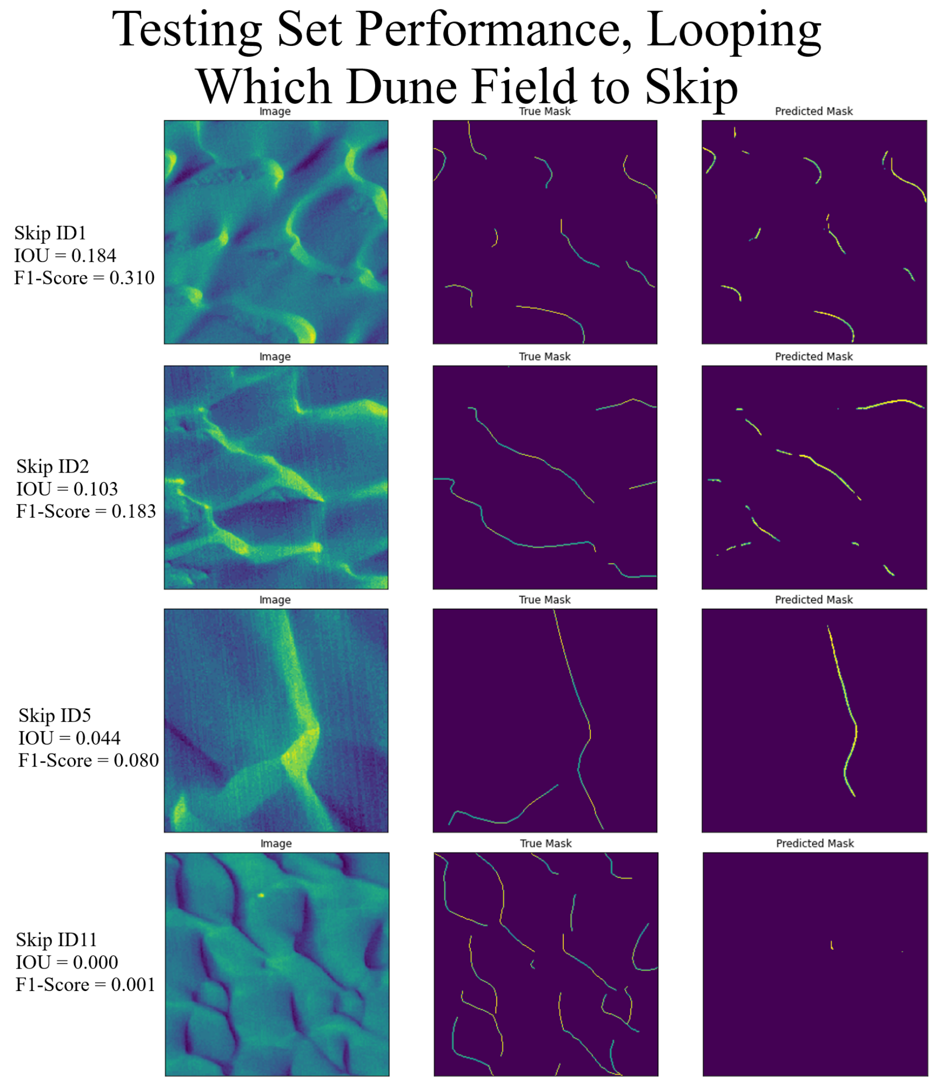

Figure 5. Loops of what dune field is being skipped and used as a testing set. For each iteration, all other 4 dune fields are being used as training set. Unfortunately, the code is about to finish running, in about 1 hour, for the final dune field, so we can only see performance of 4 dune fields. Run for 100 epochs each.

taken. Dune field 11 is unique in the dataset for these kinds of shadows and appearance, and given that similar things were not in the training set, it performed very poorly. We can see from Figure 3, the middle row example shows the potential for CNNs to understand these shadows when they are present in the training set.

6. Conclusion/Future Work

Our report shows how CNNs can be used to segment dune crestlines on planetary bodies like Mars. The most recent and relevant related studies have showed how barchan dunes (which are blobbier) can be detected [18], but we were uncertain if more linear objects can be segmented fairly. Others showed how longitudinal dunes were able to be detected [19], but they were on Earth, and also had three channels through RGB images. Our work shows the unique success of being able to use gray scale images to detect linear objects on Mars. We assess the performance of our CNN, and although scores tend to be low for a computer vision problem, qualitatively they were good predictions and ultimately can serve our main purposes of calculating morphodynamic properties.

Future work is strongly intended to be done, as this seems like a very publishable project. I intend to figure out more augmentations that could be done, hopefully specifically, a clever one that can deal with shadows or secondary dunes, which are a topic I had not brought up. Sometimes dunes can have other smaller, secondary dunes related to them, and we were focusing on predicting primary dunes. Sometimes our models would detect the secondary dunes, which makes sense cause visually they are still dunes, but the CNN could not learn the context that it is of a different scale.

I think it would also be fun to explore mapping a complete field (rather than my current method of taking 256x256 pixel images). It would also be fun to explore how dunes get mapped by my CNN on other planetary bodies, say deserts on the Earth, or sediment in Titan which is made of ice and not sand. I also definitely want to play around with trying to figure out a loss function that would better express exactly what I consider to be a good prediction (ok with some nearby pixels being mislabelled, but want to emphasize recall of all dunes to some degree). Maybe this could be done through an augmentation of adding a buffer around the true mask with depleting scores? Pixels directly adjacent to the true crestline have a score of not 1, but 0.8. 2 pixels away, 0.4, 3 pixels away 0.1... This could help the model ”not punish” decently good guesses.

References

[1] M. Azzaoui, L. Masmoudi, H. El Belrhiti, and I. Chaouki. Segmentation of crescent sand dunes in high resolution satellite images using a support vector machine for allometry. *International Journal of Advanced Computer Science and Applications*, 10(11), 2019. 1

[2] M. Azzaoui, L. Masmoudi, H. El Belrhiti, and I. Chaouki. Barchan sand dunes collisions detection in high resolution satellite images based on image clustering and transfer learning. *International Journal of Advanced Computer Science and Applications*, 11(1), 2020. 2

[3] M. C. Bourke, N. Lancaster, L. K. Fenton, E. J. Parteli, J. R. Zimbelman, and J. Radebaugh. Extraterrestrial dunes: An introduction to the special issue on planetary dune systems. *Geomorphology*, 121(1-2):1–14, 2010. 1

[4] D. Carrera, L. Bandeira, R. Santana, and J. A. Lozano. Detection of sand dunes on mars using a regular vine-based classification approach. *Knowledge-Based Systems*, 163:858–874, 2019. 1

[5] M. Day and G. Kocurek. Pattern similarity across planetary dune fields. *Geology*, 46(11):999–1002, 2018. 1

[6] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945. 2

[7] R. C. Ewing and G. Kocurek. Aeolian dune-field pattern boundary conditions. *Geomorphology*, 114(3):175–187, 2010. 1

[8] M. Foroutan and J. R. Zimbelman. Semi-automatic mapping of linear-trending bedforms using ‘self-organizing maps’ algorithm. *Geomorphology*, 293:156–166, 2017. 1, 2

[9] P. Iakubovskii. Segmentation models. https://github.com/qubvel/segmentation_models, 2019. 2

[10] Q. Jin, Z. Meng, T. D. Pham, Q. Chen, L. Wei, and R. Su. Dunet: A deformable network for retinal vessel segmentation. *Knowledge-Based Systems*, 178:149–162, 2019. 2

[11] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li. Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*, 2019. 2

[12] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017. 2

[13] A. Lu, Z. Wu, Z. Jiang, W. Wang, E. Hasi, and Y. Wang. Dcv2i: A practical approach for supporting geographers’ visual interpretation in dune segmentation with deep vision models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22788–22796, 2024. 2

[14] M. C. Malin, J. F. Bell III, B. A. Cantor, M. A. Caplinger, W. M. Calvin, R. T. Clancy, K. S. Edgett, L. Edwards, R. M. Haberle, P. B. James, et al. Context camera investigation on board the mars reconnaissance orbiter. *Journal of Geophysical Research: Planets*, 112(E5), 2007. 3

[15] M. C. Marvin, M. G. Lapôtre, A. Gunn, M. Day, and A. Soto. Dune interactions record changes in boundary conditions. *Geology*, 51(10):947–951, 2023. 1, 3

[16] F. Milletari, N. Navab, and S.-A. Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pages 565–571. Ieee, 2016. 2

[17] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2

[18] L. Rubanenko, S. Pérez-López, J. Schull, and M. G. Lapôtre. Automatic detection and segmentation of barchan dunes on mars and earth using a convolutional neural network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:9364–9371, 2021. 1, 2, 7

[19] S. Shumack, P. Hesse, and W. Farebrother. Deep learning for dune pattern mapping with the aw3d30 global surface model. *Earth Surface Processes and Landforms*, 45(11):2417–2431, 2020. 2, 7

[20] T. Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biologiske skrifter*, 5:1–34, 1948. 2

[21] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 2

[22] Y. Tang, Z. Wang, Y. Jiang, T. Zhang, and W. Yang. An auto-detection and classification algorithm for identification of sand dunes based on remote sensing images. *International Journal of Applied Earth Observation and Geoinformation*, 125:103592, 2023. 2

[23] Z. Zhang, Q. Liu, and Y. Wang. Road extraction by deep residual u-net. *IEEE Geoscience and Remote Sensing Letters*, 15(5):749–753, 2018. 2