The Cryosphere Discuss.,
https://doi.org/10.5194/tc-2020-115-RC1, 2020

Interactive comment on "Rock and snow differentiation from colour (RGB) images" *by* Alex Burton-Johnson and Nina Sofia Wyniawskyj

Anonymous Referee #1

Received and published: 2 September 2020

Review of Rock and snow differentiation from colour (RGB) images By Alex Burton-Johnson and Nina Sofia Wyniawskyj

Overall thoughts.

The main contribution is a new very manual PT method of rock/snow classification that can be applied to RGB images. I am not convinced that it really is better than MLC and other standard classifiers out of the ML toolbox. Nevertheless, I think the new PT method is a useful contribution, and a practical tool.

line 80: "the red/blue ratio differentiating rock and snow pixels increases along a second order polynomial curve". This is written as a statement of fact. I think it is an empirical observation that it appears to separate well.

Section 2.1.2: This section should describe how the coefficients of the polynomial determined. I can see from the github excel sheet that in practice it is done by hand tuning. This workflow cannot be fast. I do not think it is disqualifying that you need to manually tune the polynomial, but it has to be very clearly stated here. It is misleading to use the word "derived" for the hand-tuning.

Section 2.1.2: I am convinced that it must be possible to make an algorithm that determines the polynomial coefficients that objectively separates the two classes the best (by some metric). e.g. Using an approach similar to Support Vector Machines (SVM). I think most users would prefer that.

Figure 3: It would make sense to take the logarithm of the intensities before plotting the red vs red/blue. Perhaps the classes could then be separated by a straight line cinstead of a polynomial. The red/blue ratio can easily have outliers in shaded areas. So you can easily end up in a situation where your x-axis is spanning many orders of magnitude. In this plot there are no values with red<100. I.e. there are no pixels that really close to being dark.

Section 2.1.2: I think the PT approach will struggle when blue is near zero.

Section 2.1.1: I suspect that MLC would perform better if the inputs were log-transformed prior to running the algo. Speculation: the distributions will be better approximated by gaussians when you take the logarithm.

Section 3: Question: The performance of supervised methods depend on the relative size of the groups in the training set. Here you write that "training pixels were then selected in each image", but there are little details on how these pixels were selected. Were they selected randomly from the entire image? If so, then that would mean the training set is almost perfectly representative - This would be unlikely to be fulfilled in real applications. Please detail how the training set is selected. And if it is selected by random then please point out what that means. Also: Do you have any thoughts or recommendations for how large the snow vs rock classes should be in the training set.

section 2: There are so many standard supervised classification algorithms from machine learning. You only checked the performance of a single one (MLC) before inventing your own. This is a weakness. I would have liked to see the something like random forests or SVM applied to the problem. At the very least it would be nice to discuss why these ML methods were not considered. These would in principle be less labor intensive than PT as these methods would require zero hand tuning. (Maybe they hav stronger requirements on training data).

line 140: One potential issue with the manual hand tuning is that it can potentially be manipulated to "cheat" in the quality assesment. You can make tiny adjustments to the polynomial coefficients until it performs particularly well on the validation data set. It is concerning that there is this wiggle room, considering that the accuracy difference between PT and MLC are not huge. Let us, for the sake of argument, consider how this could affect the accuracy measures. So hypothetically: We apply the PT approach to a set of images. It generally performs well, but notice that it fails for one image. So we look into why that is. We look at plots like figure 3 and observe that there are two triangles very close to the line near x=0.9. Maybe we should shift the line slightly up so that there is better separation between the main body of points. We make this adjustment and are happy to observe that the performance for this image has improved. This adjustment is entirely sensible, but it will bias the accuracy measure. The validation dataset can no longer be said to have been kept separate from the training.

section 3.2: It is not really surprising that this does not work well. So, in my opinion this does not add much to the manuscript. It would ofcourse be nice to have an algorithm that did not need to be trained as that is labor intensive. However, i think realistically it would be more feasible to use a supervised method where the training is done once and for all or atleast not trained on every single image. E.g. your PT method but with a single global set of coefficients.

Line 157: The performance of supervised methods depend on the relative size of the

groups in the training set. Here you write that "training pixels were then selected in each image", but there are little details on how these pixels were selected. Were they selected randomly from the entire image? If so, then that would mean the training set is almost perfectly representative - This would be unlikely to be fulfilled in real applications. Please detail how the training set is selected. And if it is selected by random then please point out what that means. Also: Do you have any thoughts or recommendations for how large the snow vs rock classes should be in the training set.

I like your PT method even if it is manual. I can see it working with non-calibrated sensors. For traditional NDSI to make fully sense the input has to be corrected for the color of the atmosphere. The PT method has some flexibility. It helps to be able to deal with "fog" between sensor and surface (something like the distance effect in figure 11). It also allows it to deal with images that has been through some auto white balance. Auto white balance, and atmospheric effects would non linearly impact plots like figure 3 and NDSI. I would like to see some discussion of this. Atmospheric effects, white balance, ...

The main difference between RB-NDSI and PT is basically the order of the polynomial as far as i can see. RB-NDSI is just like PT except the line separating classes are a straight line through zero instead of a 2nd degree polynomial. This would suggest that PT would also work for traditional NDSI. Please discuss and speculate this point.

Figures: too many in my opinion.

Figure 4: Can this figure be compressed to a single page. (Suggestion: remove spacing between panels.)

Figure 7: x-axis label. It seems weird that when you exclude 100% then it performs best. Should it be "included" instead of "excluded"?

Figure 7: Here it would be nice if something like this could be done with PT. But because the coefficients are hand tuned, then i dont think that is feasible. (Unfortunately).

Figure 8: Interesting and good point.

figure 9: Not super interesting.

Line 187: I am confused by the citation. This is surely your own result, and so does not require a citation.

Line 183: The hand tuning part of the PT algorithm, and the wiggle room this leaves, means that I am not fully convinced that it is really better than MLC. (I also suspect other ML algos than MLC would perform even better.)

Line 219: I'm pretty sure this difference is not significant.

Line 225: Please include some example datasets in the github excel sheet. For example the points used in figure 3.

Interactive comment on The Cryosphere Discuss., https://doi.org/10.5194/tc-2020-115, 2020.