1. Data analysis before starting

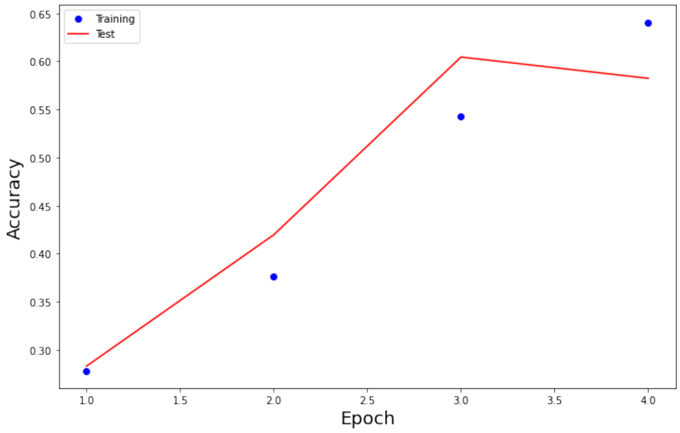
My data analysis shows that there are no picture with missing or invalid label. The dataset has a relatively balanced number of sandstones of each type. However there is a relatively low number of images, therefore requiring measures against overfitting.

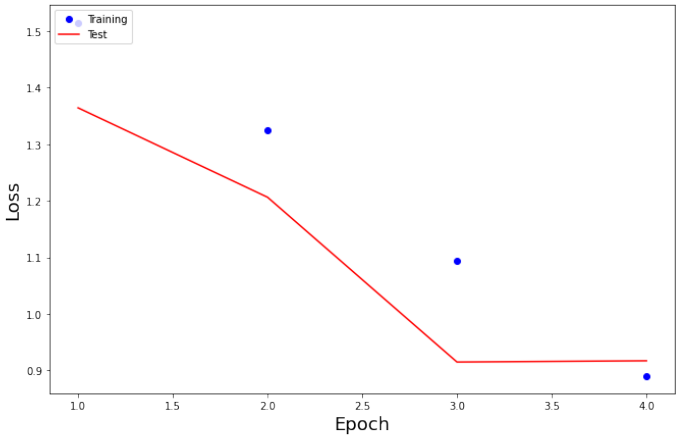
2. Pre-processing

Due to the low number of images, I plan to use image augmentation to increase the number of samples. Unlike cats and dog images, the sandstone images lack a irrelevant back ground to move against. While both horizontal and vertical flipping are safe, other operation might cause problems. While these problem can be alleviated by limiting the range, I consider flipping enough for my purpose.

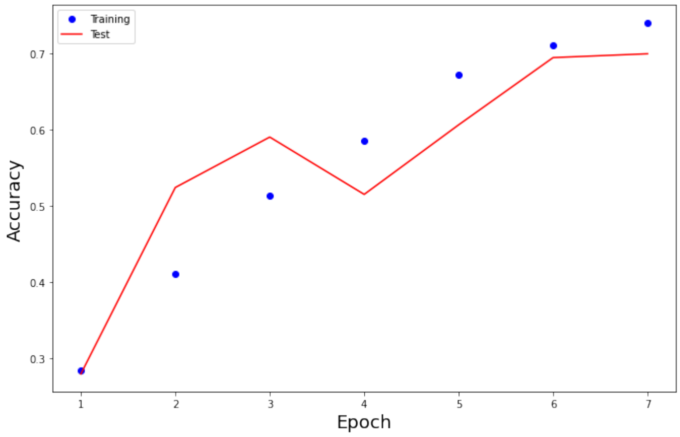
3. Models

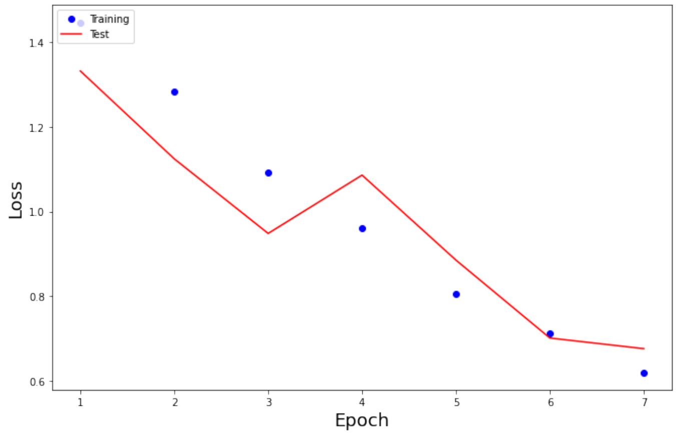
Initially I used a model and training process similar to the cats versus dogs classifier, because both problems suffer from a lack of samples. I used the similar model also because for the cats versus dogs classifier, all images are standardized to 150x150, the same dimensions as the sandstone images. Here are the learning curves.





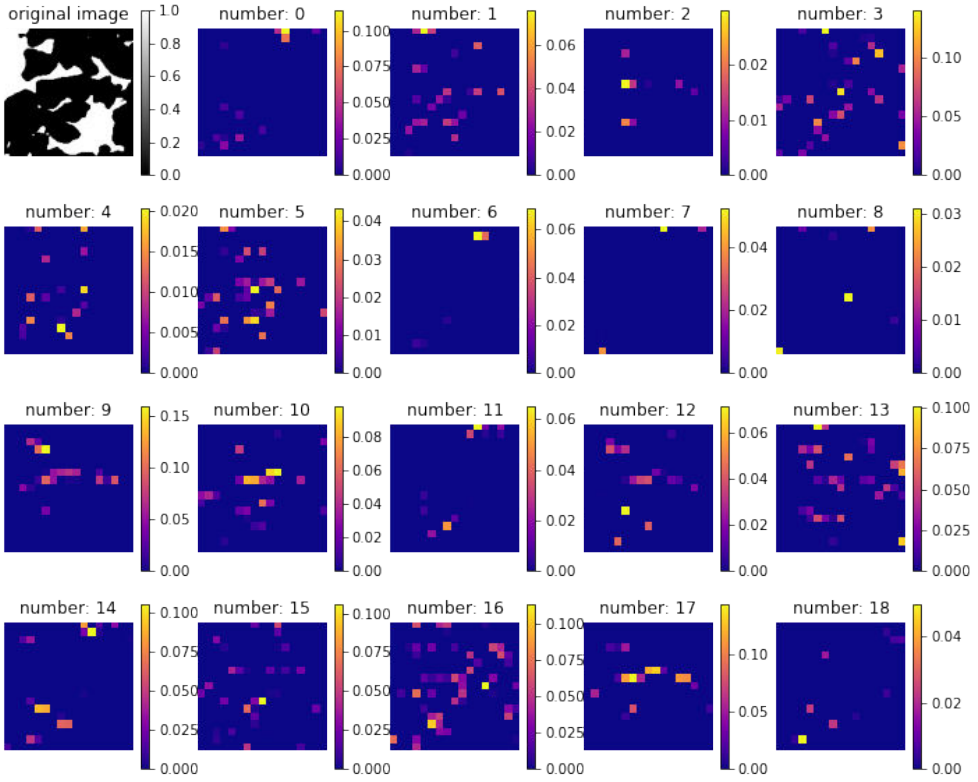
At this point it is hard to say wether the model is underfitting or overfitting, so I let the model run for a few additional epochs.

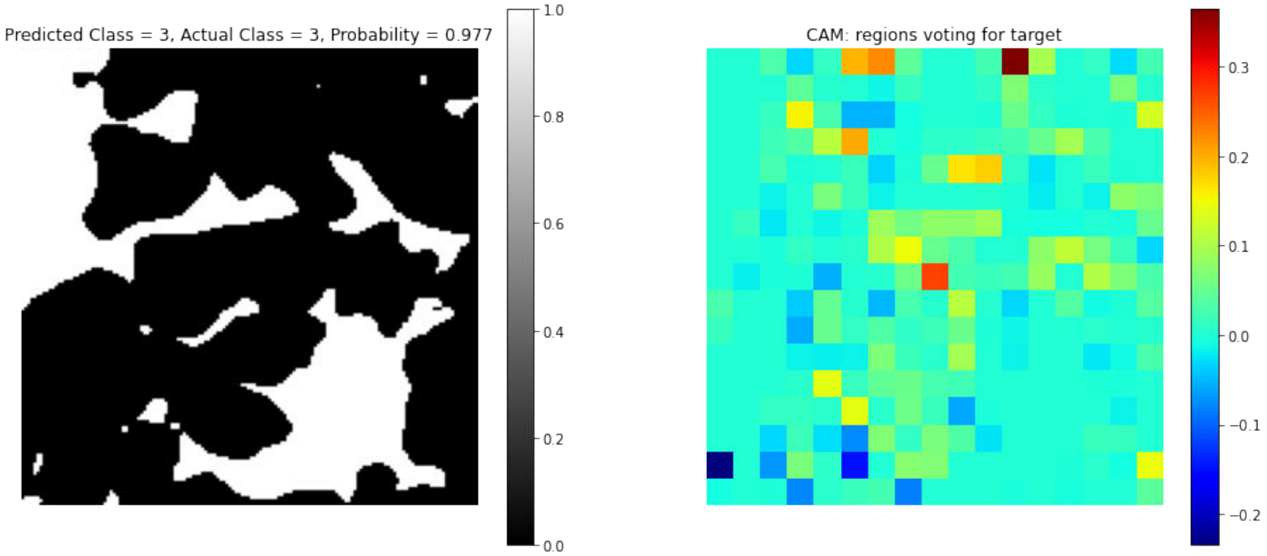




Now the model is starting to overfit but the accuracy is not satisfactory. Upon some tweaking, I have found that the model starts to overfit at around 7 epochs. This suggest either I need to improve the structure of the model so that it is less prone to overfit and better at capturing the pattern. One thing I am thinking is to try a few models and ensemble them in a voting classifier.

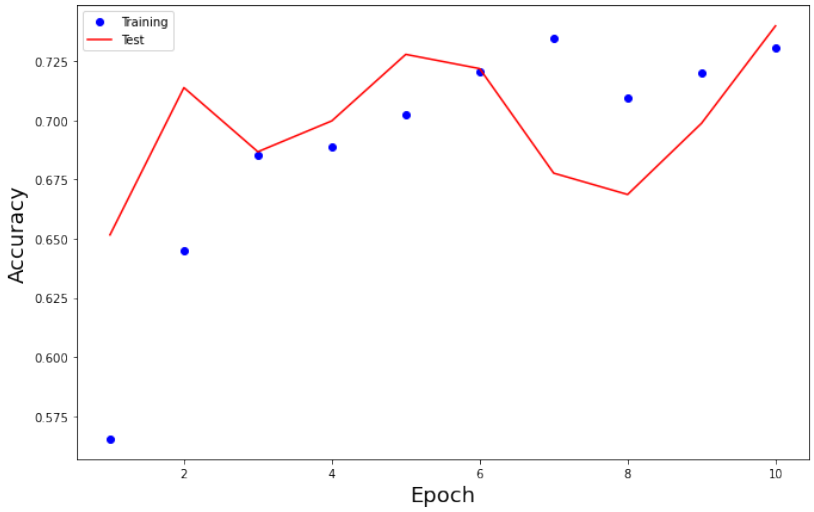
Since the sandstone classifier is a lot less intuitive than MINIST, I would like to see the CAM and feature maps before moving on.

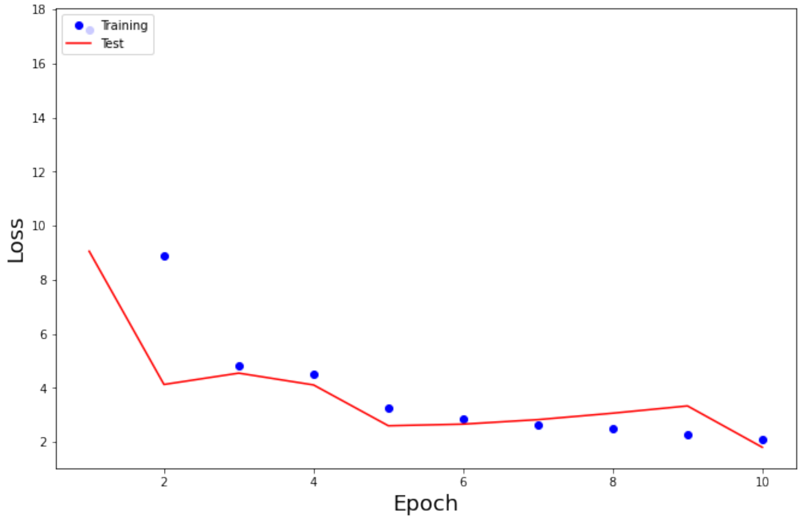




Sadly, the results are not very humanly readable and doesn’t help me with engineering features.

For the alternative models, the first thing I attempted is a one similar the transfer learning part of the cats versus dogs classifier. Again, it is due to the similarity between the two problems. However, there are two problems with this approach: 1. the sandstones are very different from the images on image net. 2. The images on image net are in RGB, therefore I need to convert the provided data set beforehand. I also need to redo the test train split to reduce correlation between the models Below are the learning curves.





As is shown above overfitting happen at 7 or so.

For the second alternative model I am using a random forest classifier because how different it is from the other models I have used. The problem is the random forest model we used in class is from sklearn and doesn’t work well with the ImageDataGenerator we used to augment data. Also, the performance of ramdom forest alone might not be as good as a CNN since it is dealing with way too many features. I can deal with both problem by using parts of my CNN model as a base model and extract features with it, but I am afraid that it might make my models too correlated for the voting to work. What I eventually choose to do is to adopt the ideas of transfer learning and use one of the pre-trained image processing models to extract features. Since I have used Xception before, I will used a different model this time.

For the voting, since I am using the random forest, I can only use a hard voting classifier. This might be the reason for the unsatisfactory results.