Attempting to make a kNN classifier to identify nuclei

When thinking of the current beginner techniques we have learned in class to try and attempt to classify nuclei, kNN is one of the first ones that come to mind. It sounds simple, we have pixels in the images that identify nuclei as well as pixels in the test images that need identification. Nuclei should theoretically look similar in color to each other and therefore choosing a nearest neighbors (in this case, nearest pixel) classifier should let us obtain some decent results. This was our thought process for building a kNN classifier for the Kaggle Data Science Bowl 2018 but we quickly ran into many issues.

The kNN classifier works on a data set that one can plot on a graph. Typically this means one x value and one y value for each data point you insert (with further x,y,z,etc. values based on the dimensionality of the data set). For computer vision, each image is essentially one of these datasets that a kNN classifier would work on. In addition, since the images may be of different size due to the different microscopy techniques used to capture the images, it becomes very hard to compare across each image.

Machine generated alternative text:
IEncodedPixels 
6908 1 7161 8 7417 8 7672 g 7928 g 8184 g 8440 8696 g 8952 g 9209 8 9465 8 9722 7 9978 7 10235 6 10493 4 10752 1 
36269 7 36523 11 36778 13 37033 15 37288 17 37543 18 37799 18 38054 19 38310 19 38565 20 38821 20 39077 20 39333 19 39589 19 39845 18 40101 18 40357 17 40614 15 40870 15 41127 13 41384 10 41641 8 41899 4 
19919 6 20174 8 20429 10 20685 11 20941 12 21197 12 21453 13 21709 13 21965 13 22221 13 22477 13 22734 12 22990 12 23246 11 23503 g 23761 6 
18671 6 18926 8 19181 g 19436 10 19691 11 19947 11 20203 12 20459 12 20715 12 20971 13 21227 14 21483 14 21739 14 21995 14 22252 13 22510 10 22768 7 23025 5 23284 2 23541 1 
40158 3 40413 5 40669 5 40925 5 41182 3 

Figure 1: RLE Encoded Values of Cell Nuclei

Finally, due to the fact that each image is essentially a dataset we need to capture, kNN can build a decision boundary based on similarity in color, but it will not know which part of the decision boundary would correspond to the nucleus and which would correspond with the cytoplasm. Furthermore, utilizing one image to try to predict for another image would have very inaccurate results as the images can be very different from one another (even produced using different microscopy techniques).

Machine generated alternative text:
Microscope 
tv6croscope 
ptcroscope 

Figure 2: Differing types of microscope techniques capturing different types of images (random sample from the dataset)

Therefore, kNN is a poor choice for attempting to classify these nuclei. Neural networks seem the most popular way to attempt to solve these computer vision problems and therefore we will try to build a simple neural network next to attempt to solve the problem.

To get started with neural networks, we began by utilizing the U-Net Kernel Starter by Kjetil Amdal-Saevik (found at <https://www.kaggle.com/keegil/keras-u-net-starter-lb-0-277?scriptVersionId=2164855>). We tested the initial run with 10 epochs and the kernel gave us a score or 0.217 after testing. We will be further optimizing the algorithm for the U-Net starter by looking at the way the mean IoU is calculated and modifying that as a first step.