Darius Irani

I added three features from the skimage.feature package: (1) Harris corner detector, (2) Canny algorithm edge detector, and (3) Shi-Tomasi corner detector. Each of these features took in the image data array as a parameter, and a series of other option parameters that I tweaked the values for. The purpose of these features was to detect edges and corners within an image to find structural patterns that could be used for classification purposes.

1. The Harris corner detector defines a fixed size “filter” through which pixels of interest can be observed through. When this filter is shifted, one can observe changes in intensity which can help to classify an image (i.e. along an edge the intensity change will be small, while deviating will have substantial change). Mathematically, the corner response is the determinant of the image matrix minus the trace squared times a constant k, set to 0.1. The standard deviation of the noise reduction filter was set to 2.
2. The Canny edge detector was another feature used to distinguish edges in an image amidst noise. This edge detector uses an algorithm that removes noise from the image of interest and then computes an intensity gradient along the edges. Unwanted non-edge pixels are removed using non-maximum suppression, and finally edge features are classified by comparison with defined thresholds. For the Canny edge detector feature, I set the standard deviation for the noise reduction filter to be 1.5, and kept the default thresholds of 0.1 and 0.2 times the image data’s max intensity.
3. Lastly, the Shi-Tomasi corner detector functions similarly to the Harris corner detector, but its response is given by the minimum of the image matrix’s eigenvalues. If this minimum surpasses a set threshold, then the pixel in question is considered a corner. Again, I set the standard deviation of the noise reduction filter was set to 2.

These changes provided a validation f1-score of 0.86 at a 0.45 threshold. I also spent the majority of my time trying to implement tensorflow to solve the problem, and although I learned a lot in the process, was unable to get my code to work.

Sources:

<http://scikit-image.org/docs/dev/api/skimage.feature.html>

<http://www.cse.psu.edu/~rtc12/CSE486/lecture06.pdf>

<http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html>

<https://docs.opencv.org/3.3.1/da/d22/tutorial_py_canny.html>

<https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_shi_tomasi/py_shi_tomasi.html>