

Dhruvin Patel

Dr. Chen

CSCI 547

4 May 2018

Final Project

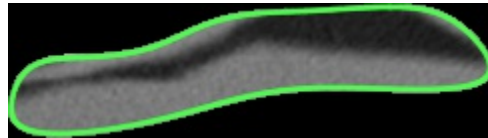
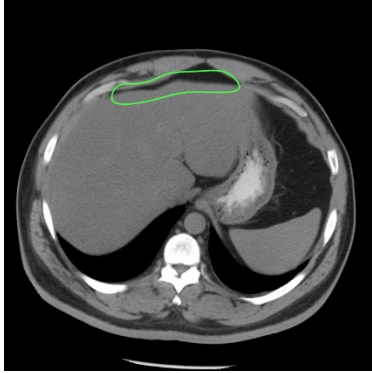
In this project, we were supposed to take a set of medical images of livers and determine if the liver is normal or cirrhotic. The dataset included 120 images for normal livers and 120 images of cirrhotic livers. My approach to the project was first to preprocess all the images, then split to train, test, and validation sets, and last use a convolution neural network to classify the images. My approach did produce accurate results; however, it has downfalls.

Preprocessing Images

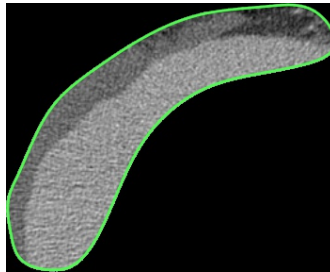
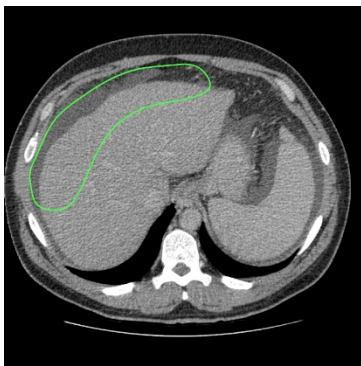
In order to load the images in python, I had to install pydicom. Pydicom allows python to read in the dicom medical images and get the pixel matrix. After getting the pixel matrix, I subtracted the green pixels to the red pixels to get the outline of the picture. After getting the outline from the image, I created a mask of the area in the green outline. The mask is used to black out all the pixels in the background. Then for every image, it finds the boundary box of the image based on the green outline. Using the boundary box, the program crops the images. I found that removing the green line before doing denoising and edge detection was better because it will detect the outline to be an edge. After removing the green outline, I applied a gaussian blur and denoising the image to get rid of noise in the image. Using OpenCV's Canny edge detection I was able to find the outline of the liver. For the cirrhotic images the threshold of Canny was much lower. After preprocessing the images, I resized them to 128x128 and save

them to a new folder. Below I have a normal and cirrhotic image as an example of my preprocessing.

Normal



Cirrhotic



Splitting Preprocessed Images

In order to train a model off the images I need to split into a training, test, and validation. To do so, I load the images from the folder and shuffled the list of files. Using a 60-20-20 split, 60 percent of the images went to training, 20 percent went to the test, and 20 percent went validation.

Classifier

I used Keras to write my CNN in python. Using a convolution with 32 3x3 filter to extract the features in the image, then used max pooling to reduce the spatial size of the representation. Repeat layers with a convolution and pooling. Then the network is flatten and

dense. Using a fully connected layer with sigmoid activation to get probability, determines if the image is normal or cirrhotic. The network is optimized with Adam and binary cross entropy for the loss.

Results

During training of the network, using the validation set it resulted in a 97% accurate. After the network was doing training, the test set resulted in 99% accurate.

Limitation

Although, the results from the network were great. My project has limitation. Since every 12 images were from the same client, the train, test, and validation set could probably contain similar images, the model is overfit. This leads to the network to have already seen the image and knowing the result. Also, since the CNN need to have a set image size, I reduced my picture to 128x128 since can cause the line edges to stretch and many create artifacts in the image.