# Generative Adversarial Neural Network for Organ Segmentation of CT Scans.

Ben Jenkins
Stephen Martinez
Dylan Ryden

## Abstract

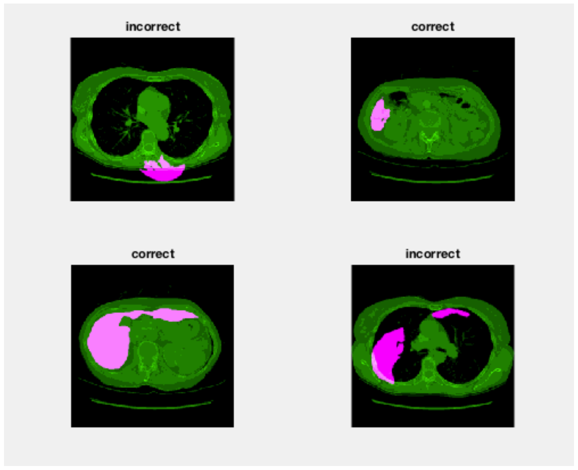
This project was designed to use the structure of a Generative Adversarial Neural Network (GANN) to design an automated computer algorithm that creates a bitmask of the liver when given a CT scan as an input. The way this technology works is by designing two separate neural networks and linking them together. These two networks are a Classifier network and a Generative network.

## Classifier Network

The purpose of this machine learning segmentation algorithm is to detect the validity of a bitmask to its corresponding CT image. Essentially, if given a CT scan with a bitmask that accurately represents a liver, the classifier would output logic 1, and logic 0 if the bitmask was an incorrect or even a poor representation of the liver in the CT scan.

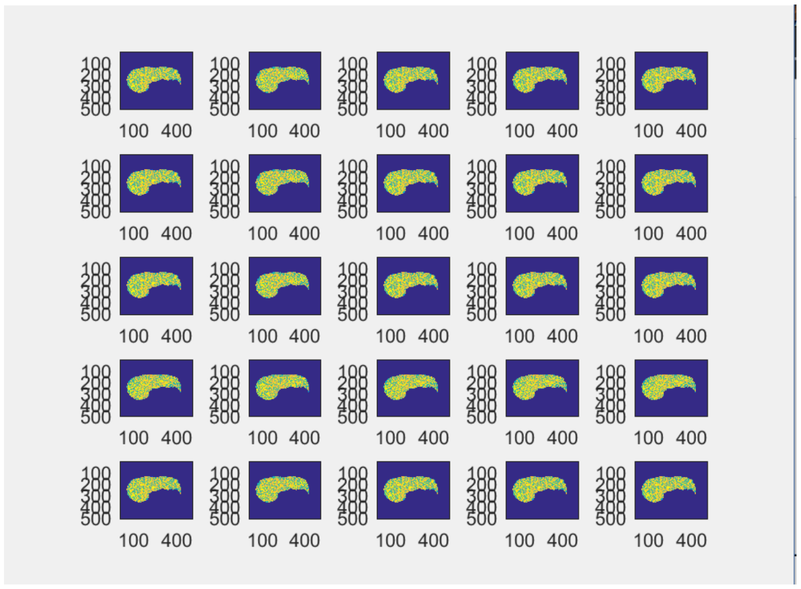
Two approaches were taken to design this classifier, both worked on in parallel to save time and increase the probability of achieving good results. The first approach, fully worked on by Stephen was to design the entire classifier from scratch and train the network on a pre-created set of data. This method involved creating each individual “layer” of the neural network and feeding the CT scans with their bitmasks through the network to train it. This method ended up needing significant computation time and power and hasn’t yielded the results we hoped for yet.

The second approach, primarily executed by Ben, was to use an existing Neural Network called AlexNet that was pre-trained on millions of images to recognize things like patterns, shapes, and shading. This involved replacing the last “layer” in the network with our own desired outcomes, which were either a correct or incorrect bitmask. Using the same training data we were able to construct a fairly good classifier that functioned as expected for all of our test cases. To clarify, our test cases were not extremely precise. They involved cases where the bitmasks were perfect fits and bitmasks that were different by a large enough factor to what the “correct” bitmask would be. These results do, however, prove the concept that a neural network can be retrained to do classification for what we hoped to.



## Generative Network

The final step in the project was to design a generative neural network and link it to the classifier network so the two could "compete" and train each other. This process was executed by Dylan, in which we once again created our own layers for the network and update those layers based on the training data of what a correct and incorrect bitmask looks like. This resulted in a network that generated bitmasks given a CT image, but the bitmasks were nowhere near correct for almost all cases. This is because they couldn't be retrained as a result of "linking" the two networks, which we couldn't figure out how to do in Matlab. The results are shown below. As you can see they are all the same output when they should be different.


100
200
300
400
500
100 400