COSC 4372
Final Project
Project Selected: The Significance of Signal Intensity from CT images
https://github.com/longle7/Stroke-Detection-Project

Team Members:
Andrew Castro 1809608, Long Le 1820820, Izhaan Akhtar 2041538, Cristopher Torres 1328148

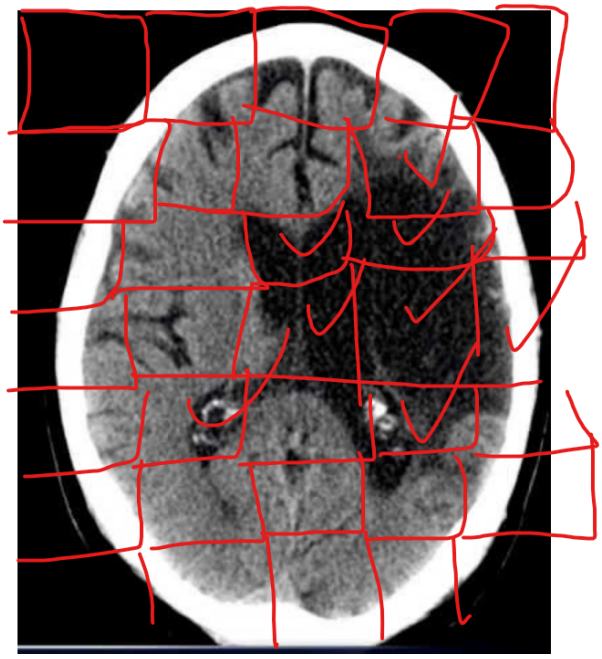
Instructions:
In this project, our goal is to implement a software to compare differences in a normal CT scan of a brain and CT scan of a patient with an ischemic stroke. Once the images are converted we will simulate a 2-d array, iterate and slice the array (5x5) of both images and flag only when the signal intensity value has significantly increased. To help us determine these values, each pixel will be from 0-255 and it will be on a greyscale. The library we will be using in python is “numpy” for this to help us generate a 2-D matrix. In result, helping us generate a 2-D flagged matrix of the damaged areas of the brain, leading us to clearly identify specific areas of stroke.

Our Goal:

1) First, we develop a function called “imagetoArray” and pass in the filename. Within this function it will contain our initialized values “image, column, row, data and pixels”.
2) Then, we will iterate through the array using a for loop and initialize our values in the matrix “data = pixels[i][j]”.
3) After testing our generated 2D arrays of each image, our function called “avgSquare” will take the average of signal intensity after we slice the 2D array into 5x5 of both images, and compare any differences in signal intensities. This will continue until we are finished looping through the entire image.
4) Furthermore, we begin by slicing the image into 25 equal areas, essentially splitting the image up into boxes. We accomplish this by using a slicing method, we sliced the image up while looping through the image.
5) Now we take the average of the normal and stroke image and output it. Luckily, numpy has a built-in function called “np.average” and subtract both and append as a list of values that we will compare.
6) Finally, we will use an if statement to determine if there are any abnormalities when it is above the threshold, which will be manually inputted by the user.

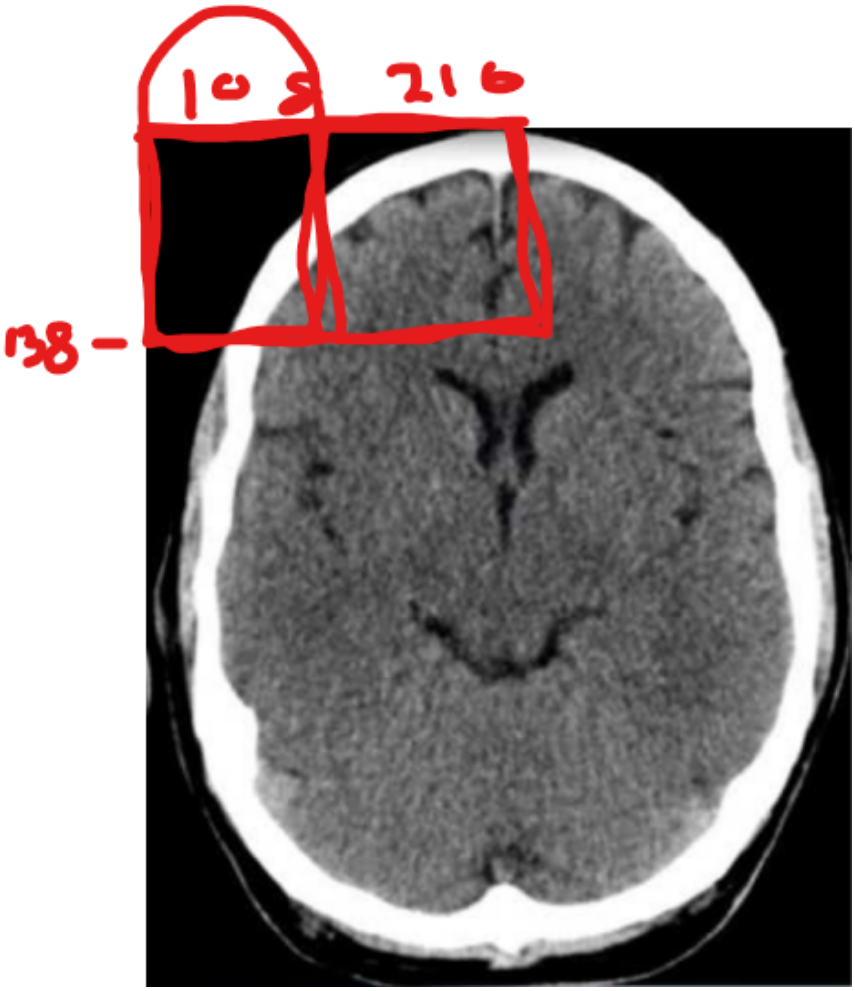
CT scan of normal brain:

CT scan of ischemic stroke:



Conclusions:
We have developed a simple way to help us identify areas of stroke by converting and comparing images of a 2D matrix of a normal brain and a patient with an ischemic stroke. When comparing the images of the arrays, we use the values of signal intensity to identify the areas of the brain with signs of strokes when compared to the CT image of a normal brain. For example, we slice the 2D arrays into 5x5s to help scalability and visualize our data to help identify any signs of abnormalities, in this case, strokes. However, we shouldn't assume all images will be the same size, so there possibly will be inaccurate data, but the concept still applies.

Slices of the CT scan:



CODE TESTING:

```
import sys

import cv2

import numpy as np



def imageToArray(fileName):
    #bring in image
    img_rgb = cv2.imread(fileName)
    #convert image to grayscale to get one value at each point
    img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
    return img_gray

#first cut into squares then find avg of each square
#assume both images are same size
def avgSquare(normal, stroke):
    row = len(normal)
    col = len(normal[0])
    rowSect = int(row/5) #138
    colSect = int(col/5) #108
    for x in range(0, row, rowSect):
        for y in range(0, col, colSect):
            areaNormal = normal[x:rowSect+x, y:colSect+y]
            areaStroke = stroke[x:rowSect+x, y:colSect+y]
            averageNormal = np.average(areaNormal)
            averageStroke = np.average(areaStroke)
            return averageNormal, averageStroke


if __name__ == '__main__':
    normalBrain = imageToArray("normalBrain.png")
    strokeBrain = imageToArray("strokeBrain.png")
    #if we want to see whole array without truncation
    #numpy.set_printoptions(threshold=sys.maxsize)
    print(avgSquare(normalBrain,strokeBrain))
```

1st Output:

(10.963116077772753, 22.02222078677088)

CODE FINAL DRAFT:

```
import math
import sys
import cv2
import numpy as np

def imageToArray(fileName):
    #bring in image
    img_rgb = cv2.imread(fileName)
    #convert image to grayscale to get one value at each point
    img_gray = cv2.cvtColor(img_rgb, cv2.COLOR_BGR2GRAY)
    return img_gray

#first cut into squares then find avg of each square
#assume both images are same size
def avgSquare(normal, stroke):
    diffList = []
    row = len(normal)
    col = len(normal[0])
    rowSect = int(row/5) #138
    colSect = int(col/5) #108
    for x in range(0, row, rowSect):
        for y in range(0, col, colSect):
            areaNormal = normal[x:rowSect+x, y:colSect+y]
            areaStroke = stroke[x:rowSect+x, y:colSect+y]
            averageNormal = np.average(areaNormal)
            averageStroke = np.average(areaStroke)
            #myList.append(str(round(averageNormal, 2)))
            if not(math.isnan(averageNormal) or math.isnan(averageStroke)):
                diffList.append(round(abs(averageNormal - averageStroke)))

    threshHold = int(input("What is the threshold? (avg is around 40): "))
    print(diffList)
    for x in range(len(diffList)):
        if diffList[x] > threshHold:
            return "There is a abnormality"
    return "There is no brain abnormality"

if __name__ == '__main__':
    normalBrain = imageToArray("normalBrain.png")
    strokeBrain = imageToArray("strokeBrain.png")
    #if we want to see whole array without truncation
```

```
    #numpy.set_printoptions(threshold=sys.maxsize)
    print(avgSquare(normalBrain,strokeBrain))
```

Output with user input as 20:
What is the threshold? (avg is around 40): 20
[11, 14, 9, 6, 10, 30, 13, 10, 48, 3, 37, 16, 52, 67, 27, 33, 6, 3, 31, 4, 22, 35, 23, 34, 4, 79, 95, 132, 56, 70]
There is a abnormality

Output with user input as 150:
What is the threshold? (avg is around 40): 150
[11, 14, 9, 6, 10, 30, 13, 10, 48, 3, 37, 16, 52, 67, 27, 33, 6, 3, 31, 4, 22, 35, 23, 34, 4, 79, 95, 132, 56, 70]
There is no brain abnormality