Honors 3980 Research

By: Parker Brandt (parker.a.brandt-1@ou.edu)

In Conjunction With: Adrien Badre PhD (adrien.f.badre-1@ou.edu), Sinaro Ly Master's(sinaro.ly@ou.edu)

Advised by: Dr. Chongle Pan (cpan@ou.edu)

GitHub Link: https://github.com/parkerbrandt/hon3980-research

How to Run

To run the program, navigate into the '/code/' folder by doing:

`cd hon3980-research/code/`

From there, use Python to execute the code by executing on the command-line:

`python data_processing.py ../configs/config.json`

This will then run the code and tell the program that the necessary configurations are located in that config.json file. The user can edit config.json to change how they want the code to run.

How it Works

When calling 'data_processing.py', the script goes performs multiple tasks:

1. Read the configuration file.
2. Retrieve the paths/filenames of all of the input images.
3. Check the noise level of each image, then remove all 'noisy' images.
    - Noise detection is done in 'code/image_noise_detection.py'
4. For each 'clean' image, crop the image, then rotate n times according to configuration file.

Demo

Materials

```
- OU OSCER Supercomputer
- Jupyter Notebook / Python
- 900 3D Images provided by Dr. Qinggong Tang (qtang@ou.edu) and Chen Wang
(chen_wang_ou@ou.edu)
```

```
    - 10 different subjects
    - 3 classes each (cortex, medulla, pelvis) with each class having 30 3D images
    - Stored as .tiff files
```

The Problem

Part 1:
Given 3-dimensional images of kidney classes, can we create a machine learning model that can classify those 3-dimensional images to which class of kidney tissue the image belongs to accurately.

Part 2:
Can we use the existing image data set to create more images to train the machine learning model further?

Methods & Algorithms

Rotation & Cropping

For rotation, I used

`scipy.ndimage.rotate(image, angle, axes=(0,2), reshape=False)`

from the scipy Python library. This function allowed me to rotate 3-Dimensional images through an axis that I could specify. This function operates by using spline interpolation to alter the original image, and filling in empty spaces with a constant 0 value.

Extending the Dataset

Image Noise

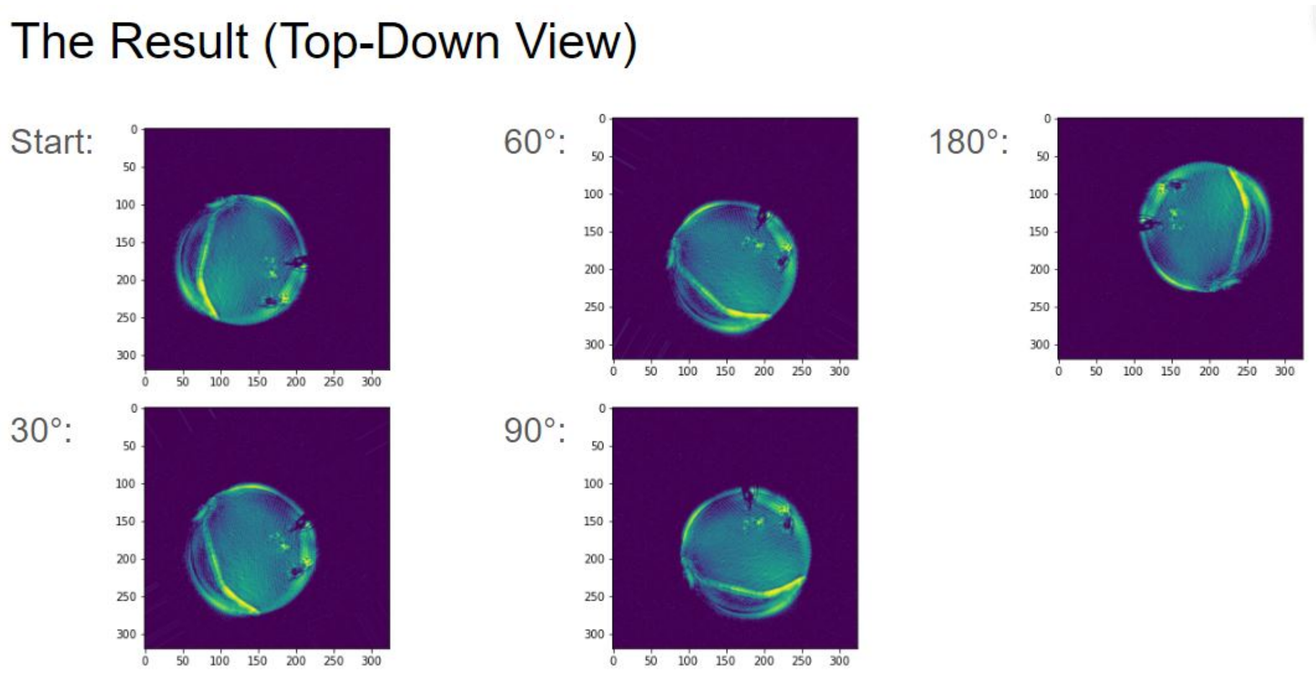
Results

Rotation & Cropping

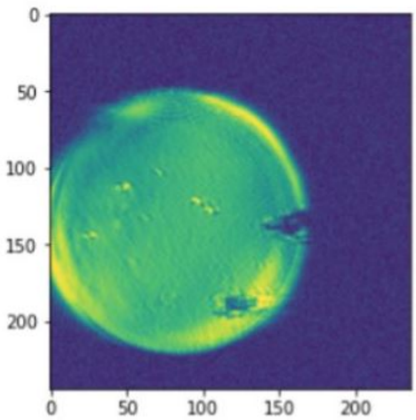
The first results I obtained were in creating the rotation function in

`code/data_processing.py/rotate_image()`

After creating this function, I tested by rotating an example image through the Z-Axis multiple times, and displaying it from a top-down view.



Issues



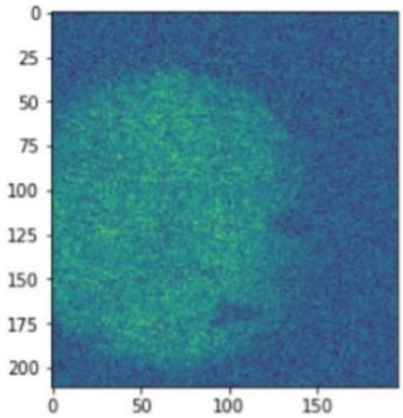
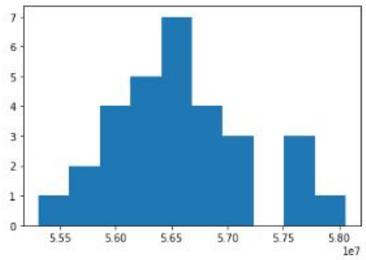


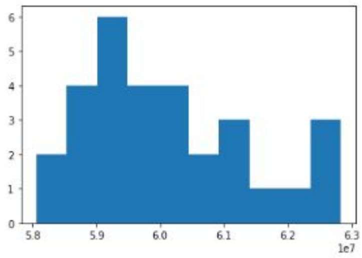
Image Noise

Using non-zero count vs. using sum of all intensities

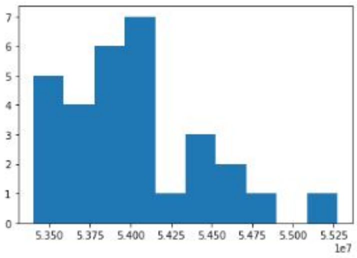
Kidney 01



Cortex (30 volumes)



Medulla (30 volumes)

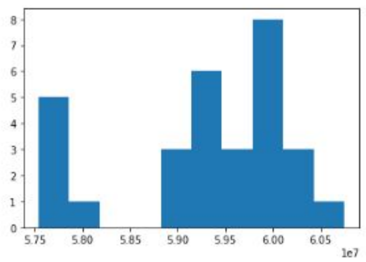


Pelvis (30 volumes)

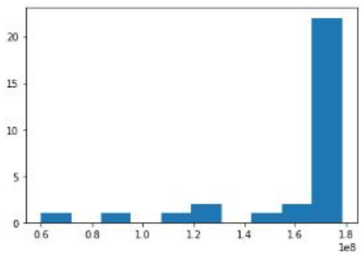
Legend:
X-axis: Non-zero pixel count
Y-axis: Frequency (number of volumes)

Noise Threshold: 1.2 * 10^8 (or 12.0 * 10^7)

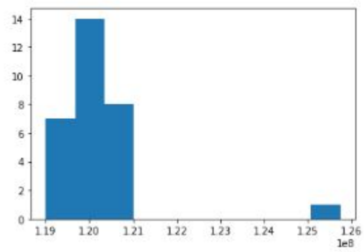
Kidney 02



Cortex (30 volumes)



Medulla (30 volumes)



Pelvis (30 volumes)

Legend:
X-axis: Non-zero pixel count
Y-axis: Frequency (number of volumes)

Noise Threshold: 1.2 * 10^8 (or 12.0 * 10^7)

After settling on using non-zero count as a valid way of

	Cortex	Medulla	Pelvis
Kidney_01	No Noise	No Noise	No Noise
Kidney_02	No Noise	27 Noisy Images	9 Noisy Images
Kidney_03	1 Noisy Image	All Noisy Images	14 Noisy Images
Kidney_04	5 Noisy Images	20 Noisy Images	10 Noisy Images
Kidney_05	All Noisy Images	All Noisy Images	All Noisy Images

	Cortex	Medulla	Pelvis
Kidney_06	All Noisy Images	All Noisy Images	All Noisy Images
Kidney_07	All Noisy Images	All Noisy Images	All Noisy Images
Kidney_08	All Noisy Images	All Noisy Images	All Noisy Images
Kidney_09	All Noisy Images	All Noisy Images	All Noisy Images
Kidney_10	All Noisy Images	All Noisy Images	All Noisy Images

** Using a cutoff of 1.2E8 nonzero pixels as a cutoff

Conclusion

Over the course of the Fall 2022 semester, I have learned about machine learning and image preprocessing techniques, and how data preprocessing and data generation is necessary for some machine learning models to work. I have learned practical uses for Python libraries such as numpy, scipy, and TensorFlow.

What's Next?

Machine Learning Training

As of 12/14/2022, we have gotten results for the first iterations of the machine learning model training and validation. The results