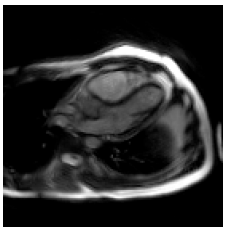
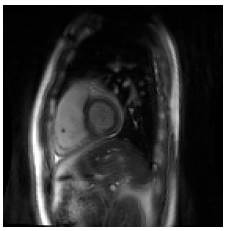
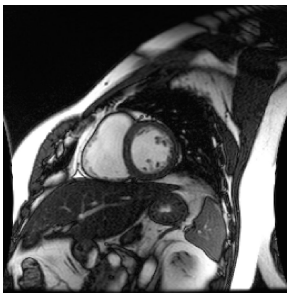
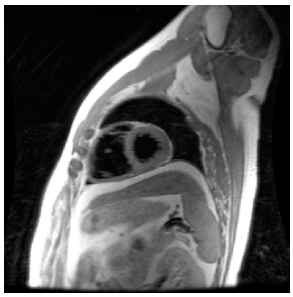
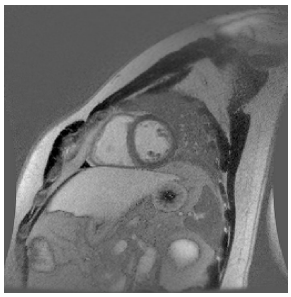
Problem 1: Classification

The data provided contains the following categories: Cardiac View, Sequence Type, and Acquisition Trajectory.

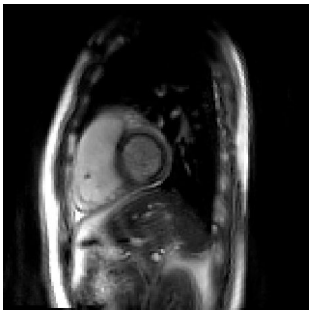
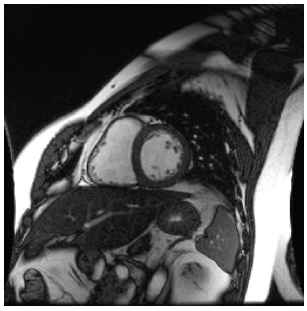
Cardiac View

Axial	2-Chamber	3-Chamber	4-Chamber	Short Axis
				

Sequence Type

SSFP	GRE	Black Blood	DE
			

Acquisition Trajectory

Spiral	Cartesian
	

Loading data

The script *load_data.py* provides functions for loading the data. It does not need to be modified. The data will be loaded when the script is imported.

Example:

```
>>> import load_data
>>> data = load_data.data
```

Each image includes the classification tags. The tags can be accessed with the following commands

```
>>> data = load_data.data
>>> image = data[0] # First image
>>> image['view'] # cardiac view
>>> image['sequence']   # sequence type
>>> image['trajectory'] # acquisition trajectory
```

NOTE: For this question, do not use information from the dicom metadata.

1a) Build a single neural net (preferably in TensorFlow) to classify the cardiac images in the three categories:

- Cardiac View
- Sequence Type
- Acquisition Trajectory

For these questions, the category refers to Cardiac View, Sequence Type, or Acquisition Trajectory.
The classes refer to the options within each category: Axial, 2-Chamber, Black Blood, Spiral, etc.

1b) Divide the available dataset into training and testing datasets for training

i. To get a sense of how well the model performs overall, provide the accuracy for matching all 3 categories successfully.
   - *Example*:
     Overall accuracy: Training Accuracy: 83%, Testing Accuracy: 80%

ii. Are some categories predicted better than others? Provide the accuracy individually for each category.
   - *Example:*
     Training accuracy - Cardiac View: 90%, Sequence Type: 97%, Acquisition Trajectory: 99%
     Testing accuracy - Cardiac View: 88%, Sequence Type: 94%, Acquisition Trajectory: 93%

iii. How about the classes within the categories? Provide the accuracy for each class within each category.
   - *Example:*
     Training accuracy
     View: Axial: 98%. 2-Chamber: 92%, 3-Chamber: 94%...
     Trajectory: Spiral: 99%, Cartesian: 99%...

Problem 2: Field of View Estimation

In MRI, the field of view is defined as the spatial extent of an image. The field of view can be determined based on the size of the pixel and the number of pixels in each dimension of an image.
For example, a 160x160 image with a pixel size of 2 mm has a field of view of 320 mm x 320 mm.

2a) Using *only data from* **load_data.data** for training, build a neural network model to predict the fields of views of images in the test dataset **load_data.fov_test_data**.
Do not include images from **load_data.fov_test_data** in the training data.

```
>>> import load_data
>>> training_data = load_data.data
>>> test_data = load_data.fov_test_data
```

Each image includes the field of view information.

```
>>> image = training_data[0]['image']
>>> field_of_view = training_data[0]['field_of_view']
```

2b) What is the average accuracy in millimeters that you are able to achieve on the test dataset?

Extra Credit

If you have some extra time, you now have some cardiac image data to play with. See if there are any interesting deep learning tasks you could apply to the data. Possible tasks could be:

- Unsupervised image classification
- Generating synthetic cardiac images
- Predicting metadata tags associated with the images
    - For this task you would need to access the metadata by modifying the load_data.py script