

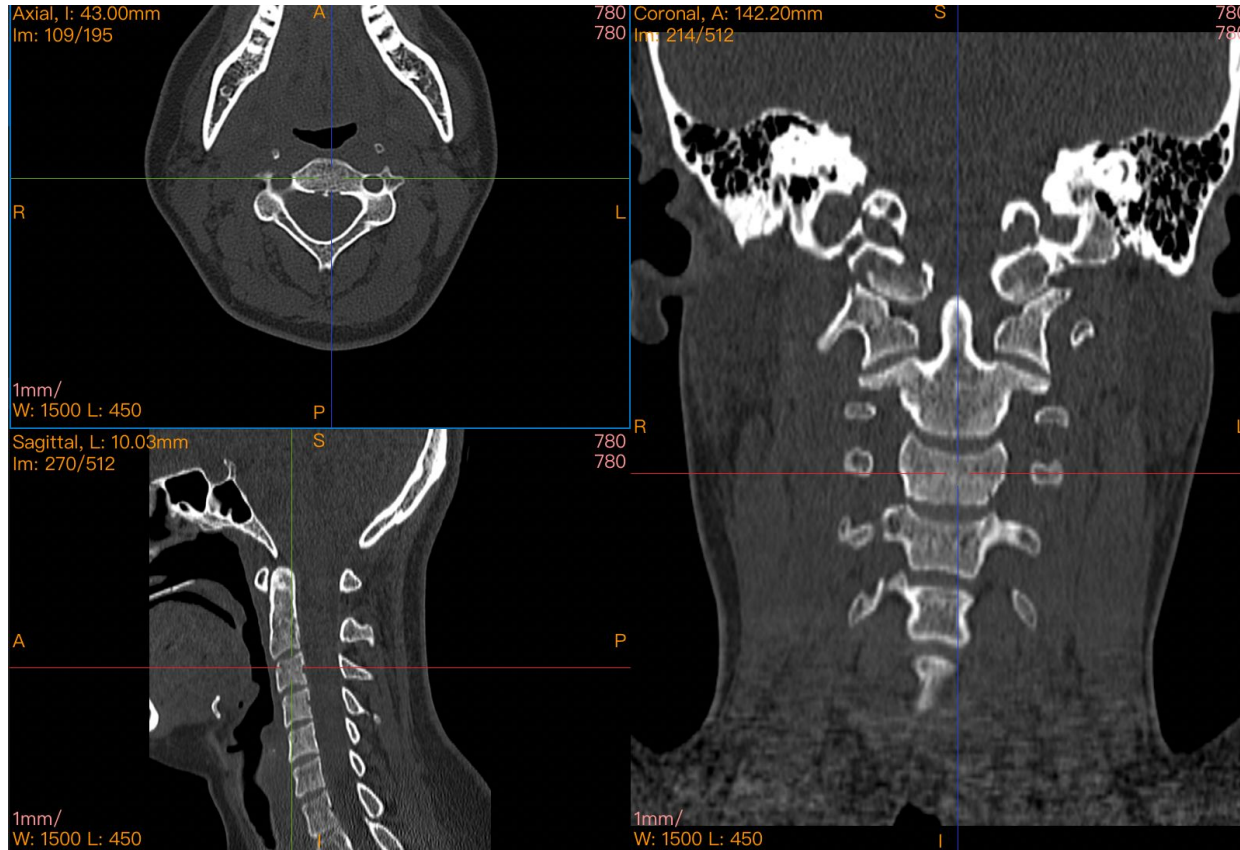
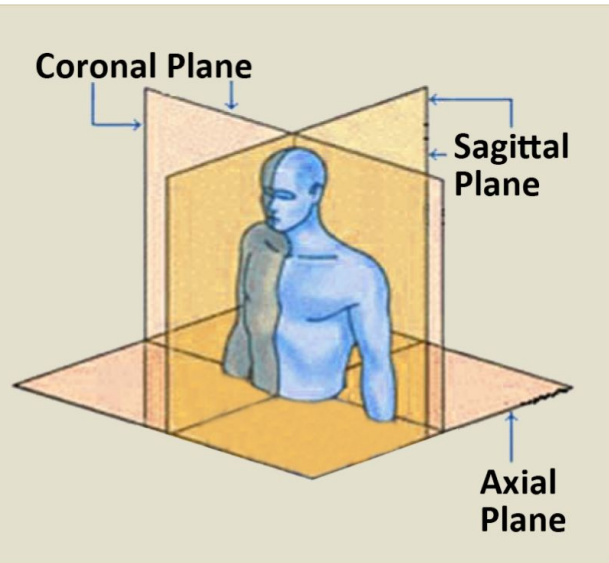
Sprint 2 : Cervical spine fracture detection

A1_10

Progress since sprint 1

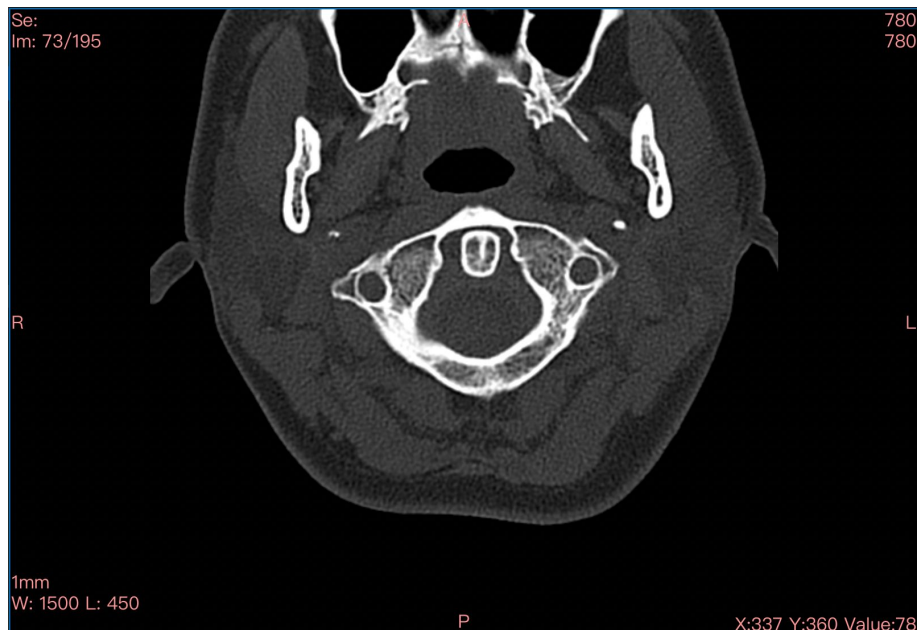
- Downloaded dataset
- Processing dicom file
- Constructing a simple model
- Training and testing model

Processing dicom file: Modify window width=1000, window level=600



- Axial:

Before:

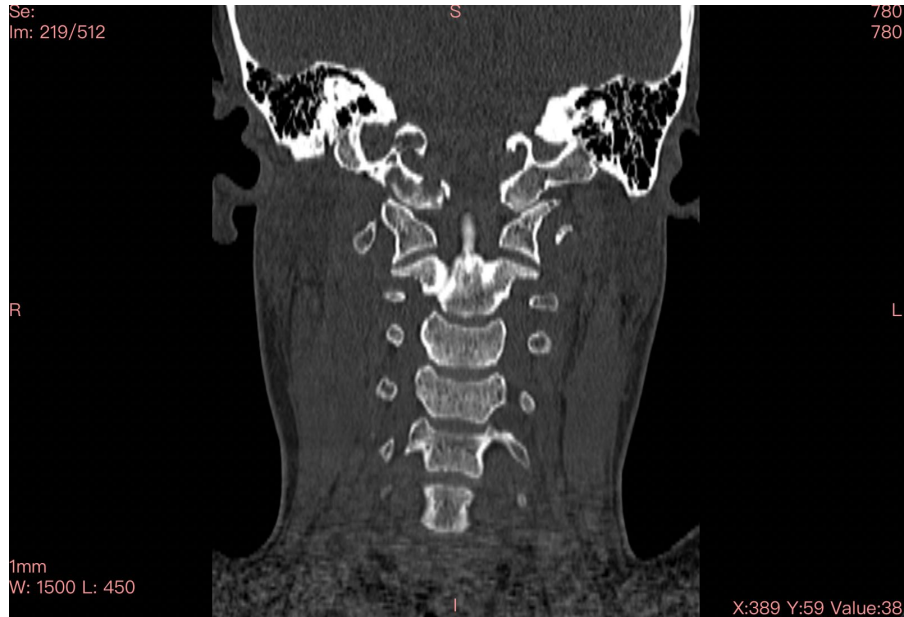


After:

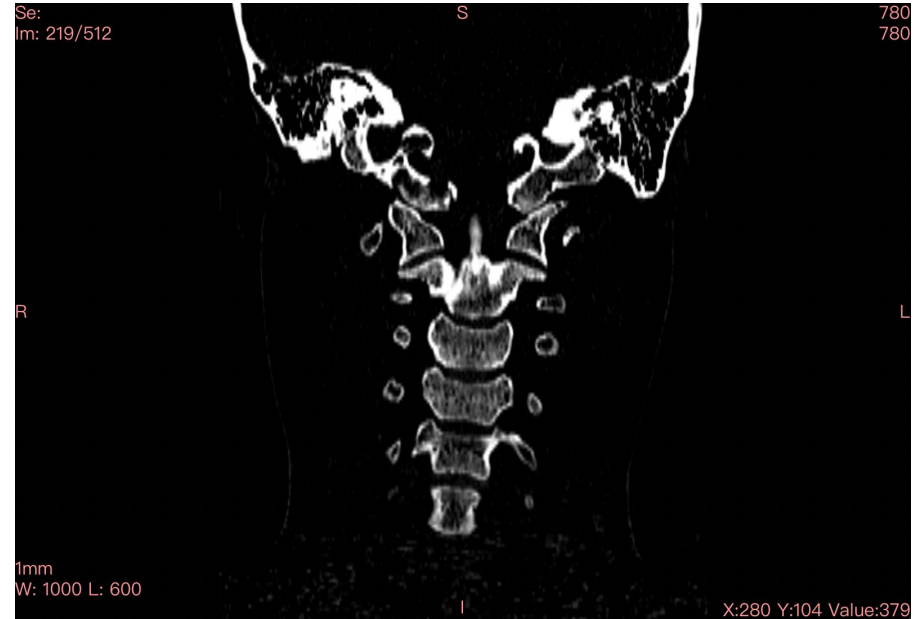


- Coronal:

Before:



After:



- Sagittal:

Before:



After:



3D CNN Model

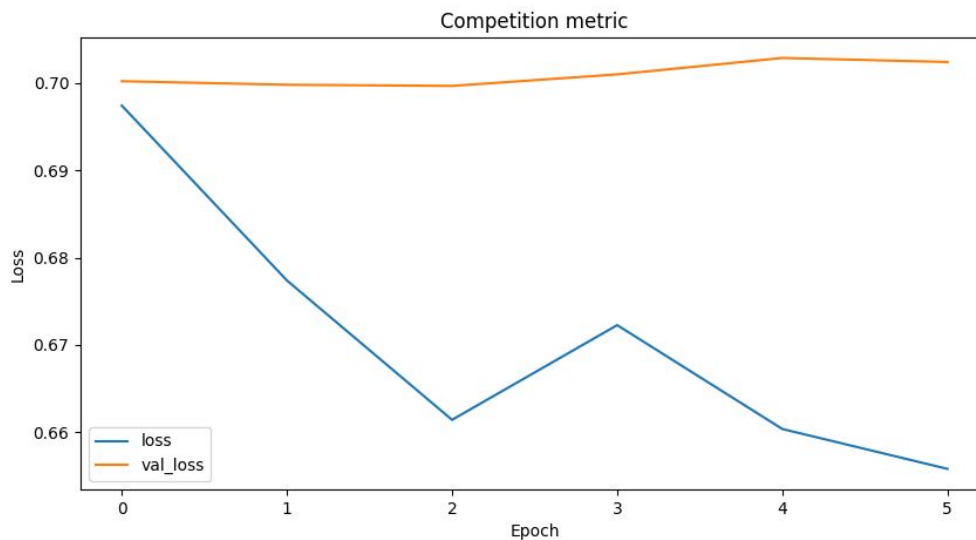
- 3 Convolutional layers
- Simple and fast enough to debug and test

```
# 3D convolutional neural network
class Conv3DNet(nn.Module):
    def __init__(self):
        super().__init__()

        # Layers
        self.conv1 = nn.Conv3d(in_channels=1, out_channels=16, kernel_size=7, stride=1, padding=0)
        self.pool = nn.MaxPool3d(kernel_size=2, stride=2, padding=0)
        self.norm1 = nn.BatchNorm3d(num_features=16)
        self.conv2 = nn.Conv3d(in_channels=16, out_channels=32, kernel_size=3, stride=1, padding=0)
        self.norm2 = nn.BatchNorm3d(num_features=32)
        self.conv3 = nn.Conv3d(in_channels=32, out_channels=64, kernel_size=3, stride=1, padding=0)
        self.norm3 = nn.BatchNorm3d(num_features=64)
        self.avg = nn.AdaptiveAvgPool3d((7, 1, 1))
        self.flat = nn.Flatten()
        self.relu = nn.ReLU()
        self.lin1 = nn.Linear(in_features=448, out_features=128)
        self.lin2 = nn.Linear(in_features=128, out_features=8)
```

Training Loss

- Trained on 7 cases for 6 epochs to test
- Loss value for training dataset decreased



Next goal for sprint 3

- Train our model using SCC
- Use GPU computing to accelerate training
- Implement a deeper open source model (like ResNet-50)
- Compare the results with former model