Cervical Spine Fracture Detection

W210 Capstone Presentation II
Fall 2022 - Section 3

Weijia Li, Jane Hung, Minjie Xu, Fengyao Luo

Agenda

Project Recap

Pipeline

Image Preprocess & Augmentation

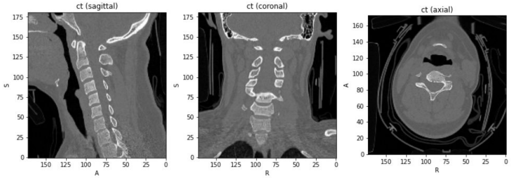
Models Comparison

Model Results

Error Analysis

Next Steps

Project Recap

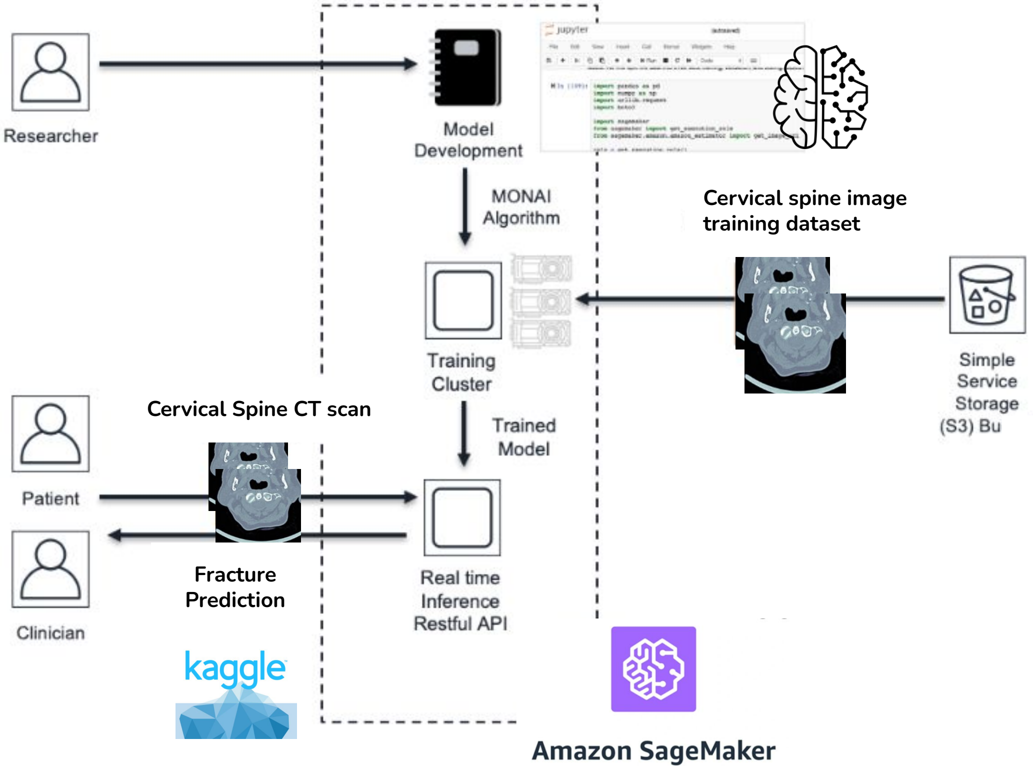


NeckFrac

CERVICAL SPINE FRACTURE DETECTOR

- Background
- Market size
- RSNA Cervical Spine Fracture AI Challenge
- MVP
- Product vision
- Mission statement: Quicker, better, more accurate diagnosis to save lives.
- Customers
- Competitors

Pipeline



Models

- **EfficientNet** (use .dicom)
- **DenseNet** - baseline (use JPEG)
- **Custom CNN** - Preprocess + Augmentation (use .dicom)

Train vs Validation Split

(80% vs 20%)

Evaluation Metrics

EVALUATION

- Competition Weighted Loss
- Train loss vs Validation loss
- Accuracy, Precision, Recall, F1 Score
- FP Rate, FN Rate

Image Preprocess

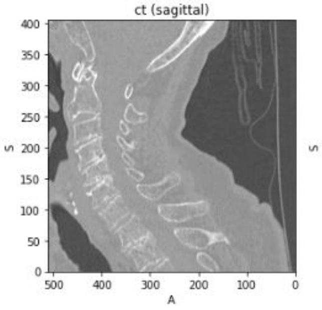
Reduce the memory per patient from 99.5M to 21.4M

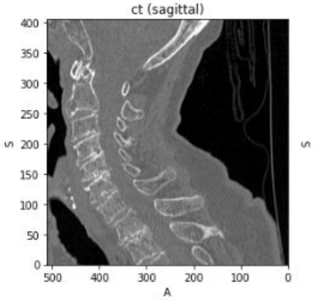
Intensity

Clamp intensity

Min: HU = -1000 (air)

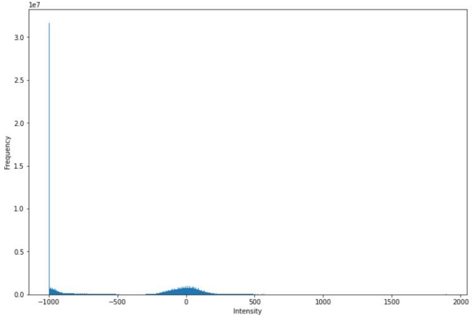
Max: HU = 1900 (bone)

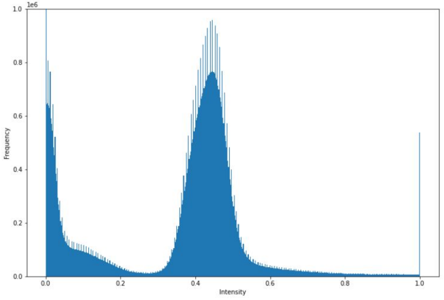




Rescale Intensity

normalize the values to [0, 1] for CNN

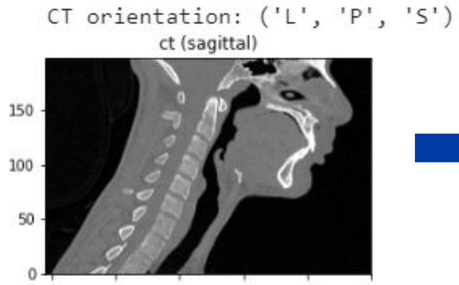


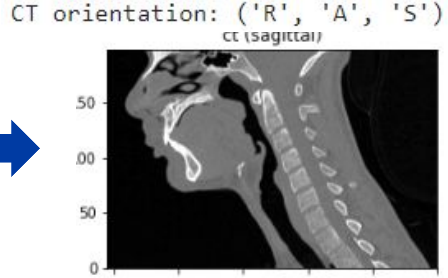


Spatial

ToCanonical

normalize the orientation (to RAS+)



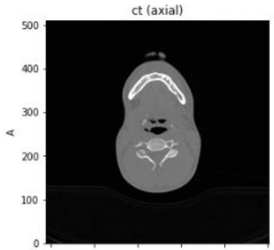


Resample

to a sensible value (1 mm isotropic) for faster computations

spacing: (0.58, 0.58, 1.00) → **spacing: (1.00, 1.00, 1.00)**

CropOrPad To (224,224,224)



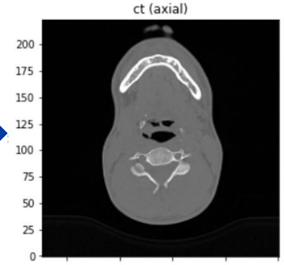


Image Augmentation

Random Anisotropy(p=0.25)

make images look anisotropic 25% of times

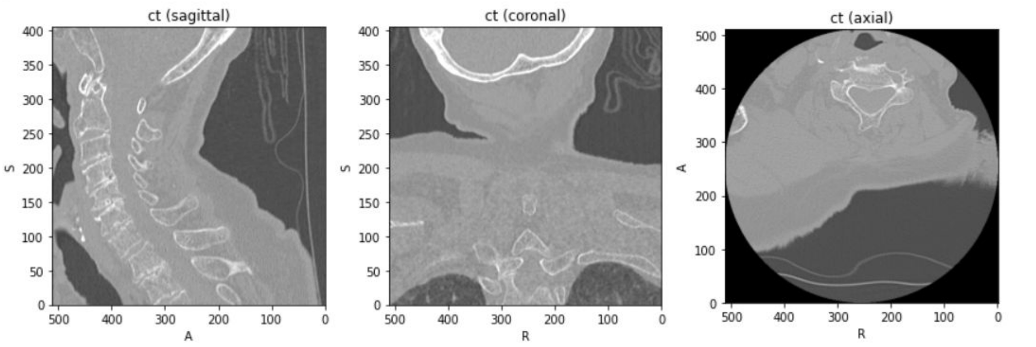
Random Affine

Apply a random affine transformation

Random Noise(p=0.25)

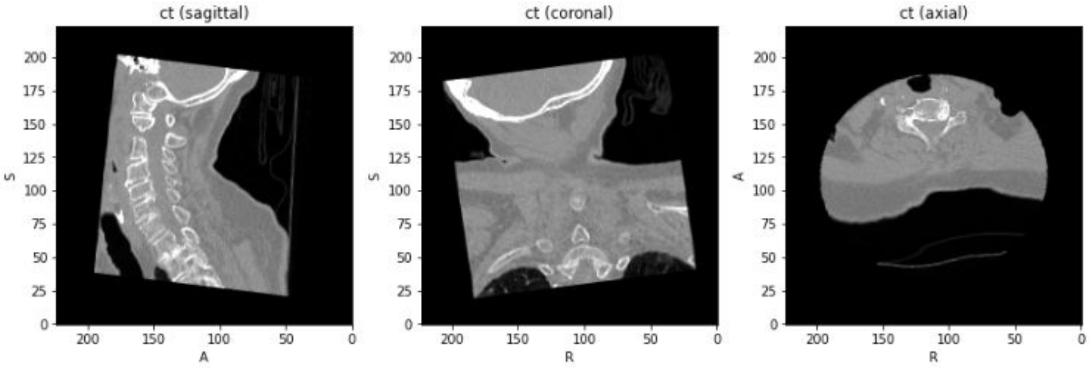
Gaussian noise 25% of times

Random Flip




```
ScalarImage(shape: (1, 512, 512, 406); spacing: (0.31, 0.31, 0.40); orientation: L
PS+; dtype: torch.ShortTensor; memory: 203.0 MiB)
```

Before

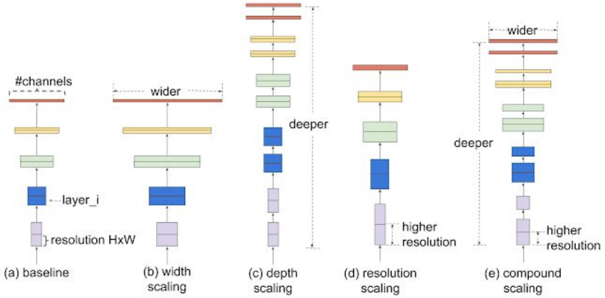


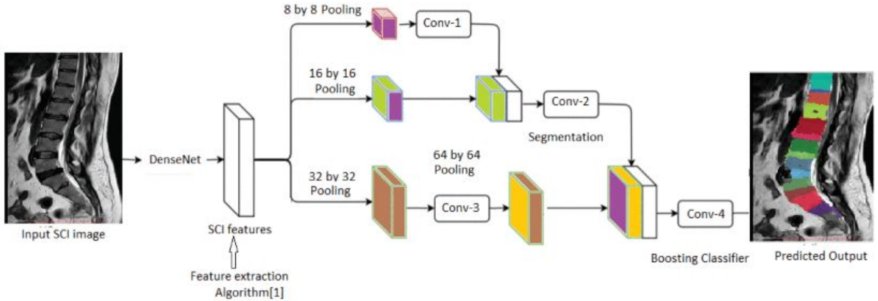

```
ScalarImage(shape: (1, 224, 224, 224); spacing: (1.00, 1.00, 1.00); orientation: R
AS+; dtype: torch.FloatTensor; memory: 42.9 MiB)
```

After

Models & Architectures

- EfficentNet vs. DenseNet vs. customized CNN (Torchio-CNN)
  - Keras vs PyTorch
  - Model scaling
  - Training size
  - Image resolution (resize)
  - Model running time





TorchIO

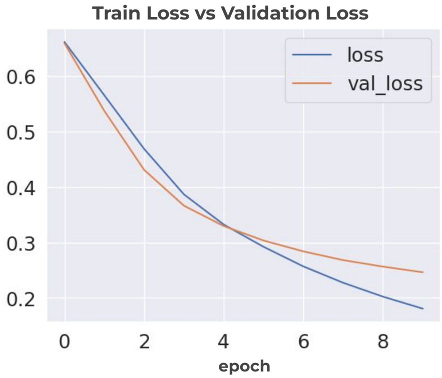
C1 to C7 Classification - EfficientNetV2

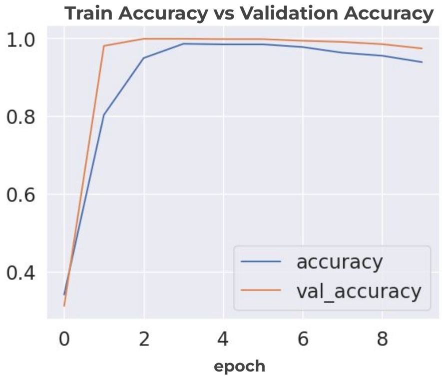
Model

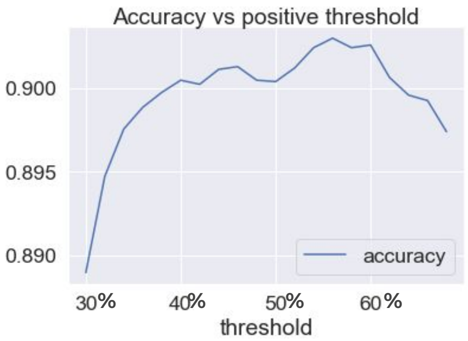
```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, None, None, 1)]   0

 conv2d_1 (Conv2D)           (None, None, None, 3)     30

 efficientnet-b5 (Functional  (None, None, None, 2048)  28513520
 )

 global_average_pooling2d (G  (None, 2048)             0
 lobalAveragePooling2D)

 dense (Dense)               (None, 9)                 18441

=================================================================
Total params: 28,531,991
Trainable params: 28,359,255
Non-trainable params: 172,736
_________________________________________________________________
```

Result

```
actual label:
[1. 1. 1. 0. 0. 0. 0. 0. 0.]

predicted output:
[0.92639005 0.88605517 0.84040582 0.05682087 0.03717361 0.0239842
 0.0283552  0.03207355 0.04325184]

predicted label:
[1. 1. 1. 0. 0. 0. 0. 0. 0.]
```

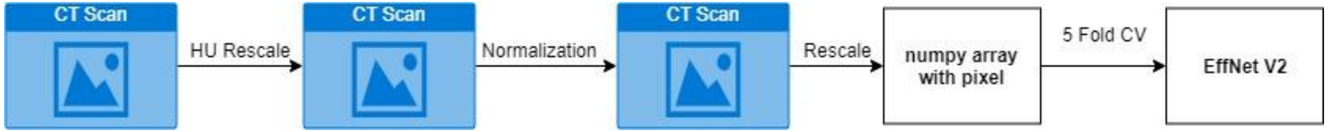


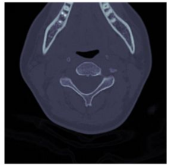


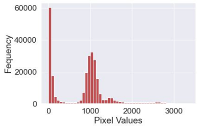


	StudyInstanceUID	Slice	C1	C2	C3	C4	C5	C6	C7
1362	1.2.826.0.1.3680043.12281	170	0	0	1	1	0	0	0
1363	1.2.826.0.1.3680043.12281	171	0	0	1	1	0	0	0
1364	1.2.826.0.1.3680043.12281	172	0	0	1	1	0	0	0
1365	1.2.826.0.1.3680043.12281	173	0	0	0	1	0	0	0
1366	1.2.826.0.1.3680043.12281	174	0	0	0	1	0	0	0
1367	1.2.826.0.1.3680043.12281	175	0	0	0	1	0	0	0
1368	1.2.826.0.1.3680043.12281	176	0	0	0	1	0	0	0
1369	1.2.826.0.1.3680043.12281	177	0	0	0	1	0	0	0
1370	1.2.826.0.1.3680043.12281	178	0	0	0	1	0	0	0
1371	1.2.826.0.1.3680043.12281	179	0	0	0	1	0	0	0

Fracture Prediction - EfficientNetV2

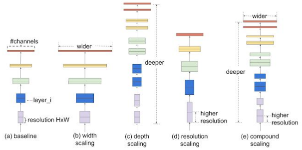






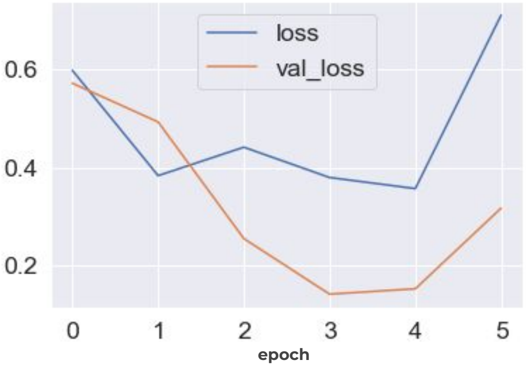
(512 , 512)

(128 , 128)




```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, None, None, 1)]   0

 conv2d (Conv2D)             (None, None, None, 3)     30

 efficientnet-b5 (Functional  (None, None, None, 2048)  28513520
 )

 global_average_pooling2d (G  (None, 2048)             0
 lobalAveragePooling2D)

 dense (Dense)               (None, 7)                 14343

=================================================================
Total params: 28,527,893
Trainable params: 28,355,157
```

Minimum validation loss: 0.14071543514728546



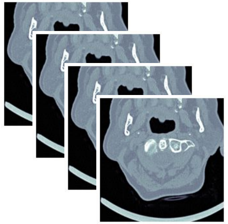
	row_id	fractured
0	1.2.826.0.1.3680043.22327_C1	0.596053
1	1.2.826.0.1.3680043.22327_C2	0.616532
2	1.2.826.0.1.3680043.22327_C3	0.329862
3	1.2.826.0.1.3680043.22327_C4	0.577175
4	1.2.826.0.1.3680043.22327_C5	0.434725
5	1.2.826.0.1.3680043.22327_C6	0.453186
6	1.2.826.0.1.3680043.22327_C7	0.507186
7	1.2.826.0.1.3680043.22327_patient_overall	1.000000

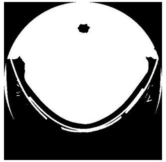
Model Baseline

Downsample dataset → Resize images → Process 3D data → Convert images to PNG → Define loss → DenseNet-121 Model

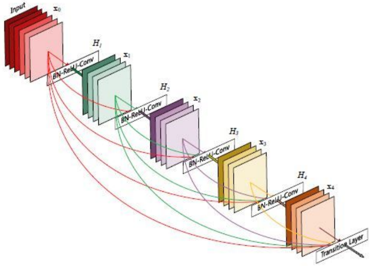
202 patients
(10%)

(512, 512)
↓
(150, 150)

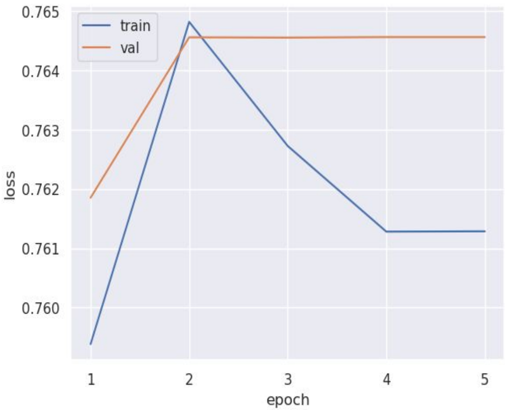


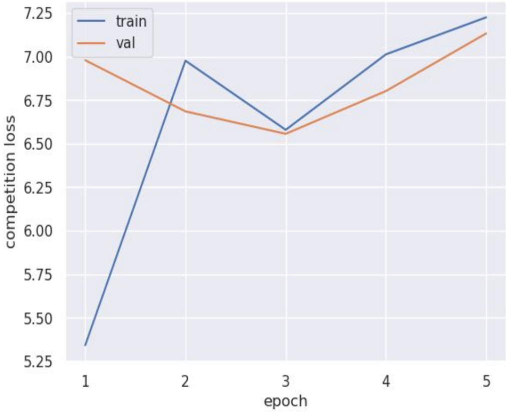


- BCEntropy
- Competition



tldr; Baseline overfits based on BCEntropy loss and would not perform well on Kaggle.





Error Analysis

Average inference time : 5.261 s per patient

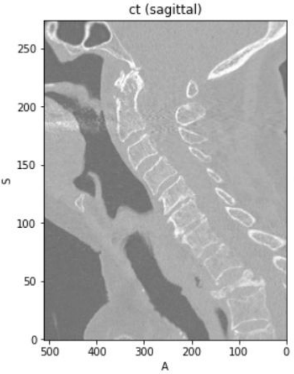
	eval_metric	patient_level	vertebrae_level	C1	C2	C3	C4	C5	C6	C7
0	recall (%)	0	8.6	0	0	33.3	100	0	0	0
1	precision (%)	0	7.3	0	0	3.7	14.3	0	0	0
2	tn	19	214	37	33	12	27	36	36	33
3	fp	0	38	0	0	26	12	0	0	0
4	fn	22	32	4	8	2	0	5	5	8
5	tp	0	3	0	0	1	2	0	0	0
6	fpr (%)	0	15.1	0	0	68.4	30.8	0	0	0
7	fnr (%)	100	91.4	100	100	66.7	0	100	100	100

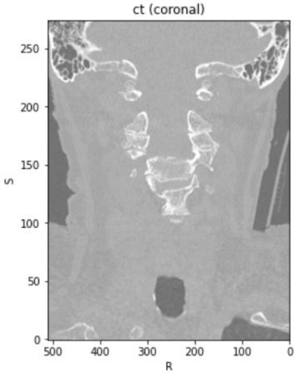
1) The baseline model can **not detect C1, C2, and C7 fractures**, which are the most common and medically problematic.

2) The model produces relatively **fewer false negatives for C3 and C4** fractures, but these are less likely to fracture.

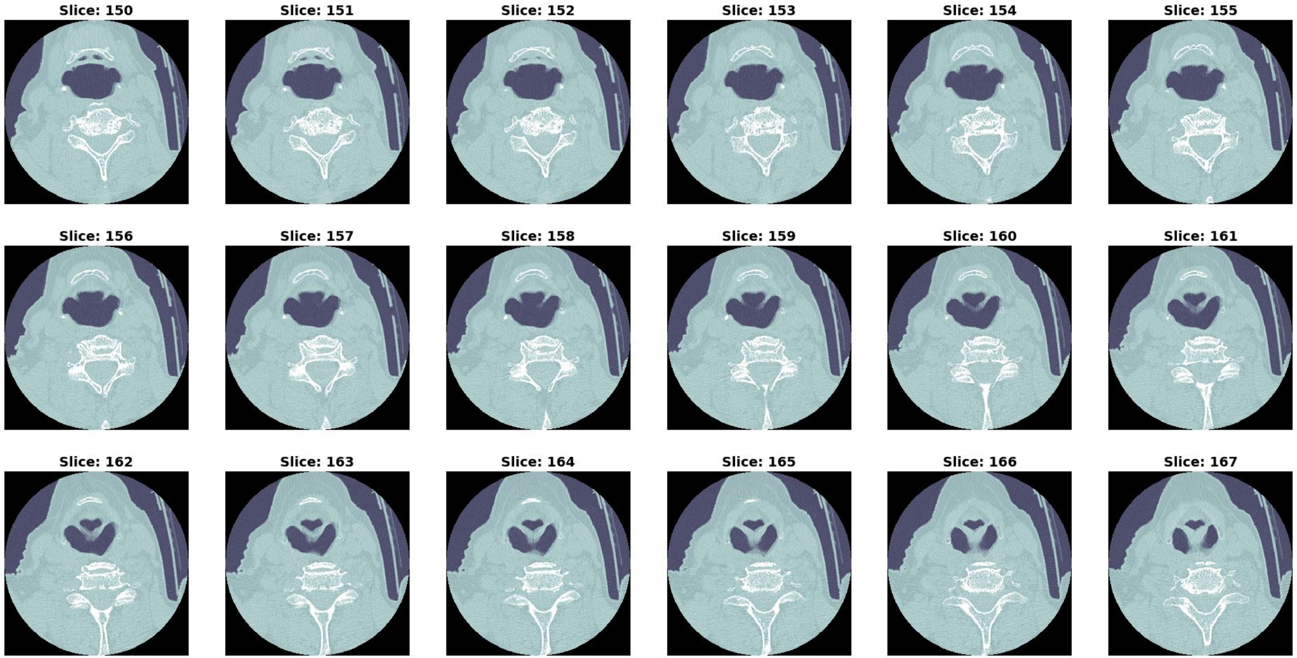
Error Analysis - Example

	actual	row_id	fractured
7	1	1.2.826.0.1.3680043.581_patient_overall	0.0
48	1	1.2.826.0.1.3680043.581_C1	0.0
89	1	1.2.826.0.1.3680043.581_C2	0.0
130	0	1.2.826.0.1.3680043.581_C3	1.0
171	0	1.2.826.0.1.3680043.581_C4	0.0
212	0	1.2.826.0.1.3680043.581_C5	0.0
253	0	1.2.826.0.1.3680043.581_C6	0.0
294	0	1.2.826.0.1.3680043.581_C7	0.0





ID: 1.2.826.0.1.3680043.581



Next step

- C1-C7 classification:
  - Run the model with full dataset
  - Implement the result to the fracture prediction model
- In depth error analysis to the existing models built
  - EfficientNet
  - DenseNet
  - CNN
- Implement additional feature engineering to the model
- Further optimize the infrastructure and model performance to train larger dataset
- Submit our model before the Final Submission Deadline on Thursday
- Sagittal view model
- Ensemble models