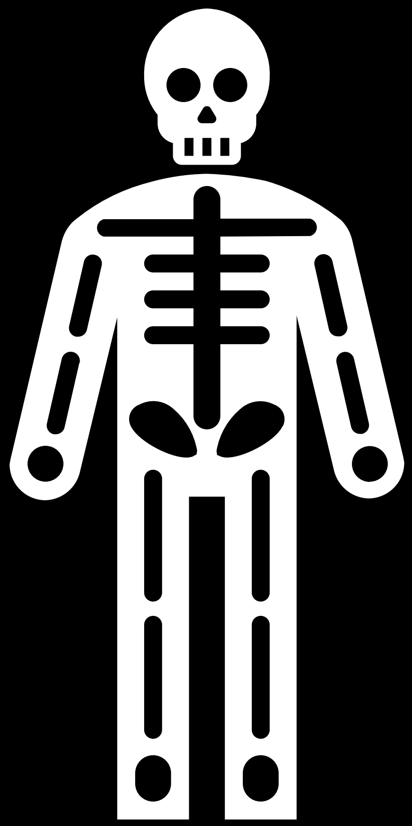
RSNA 2022 Cervical Spine Fracture Detection



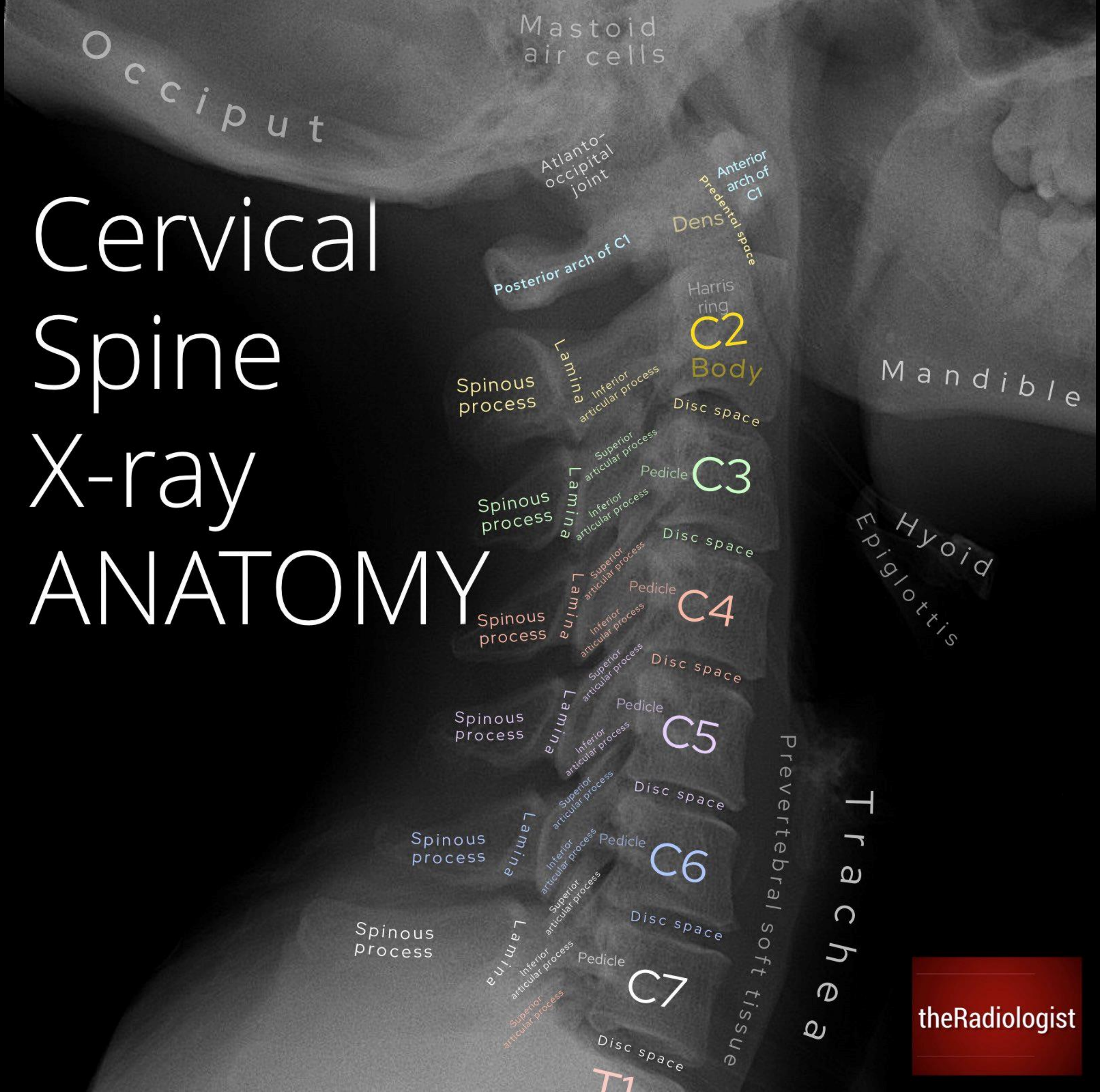
Problem Overview

Identify cervical fractures from scans

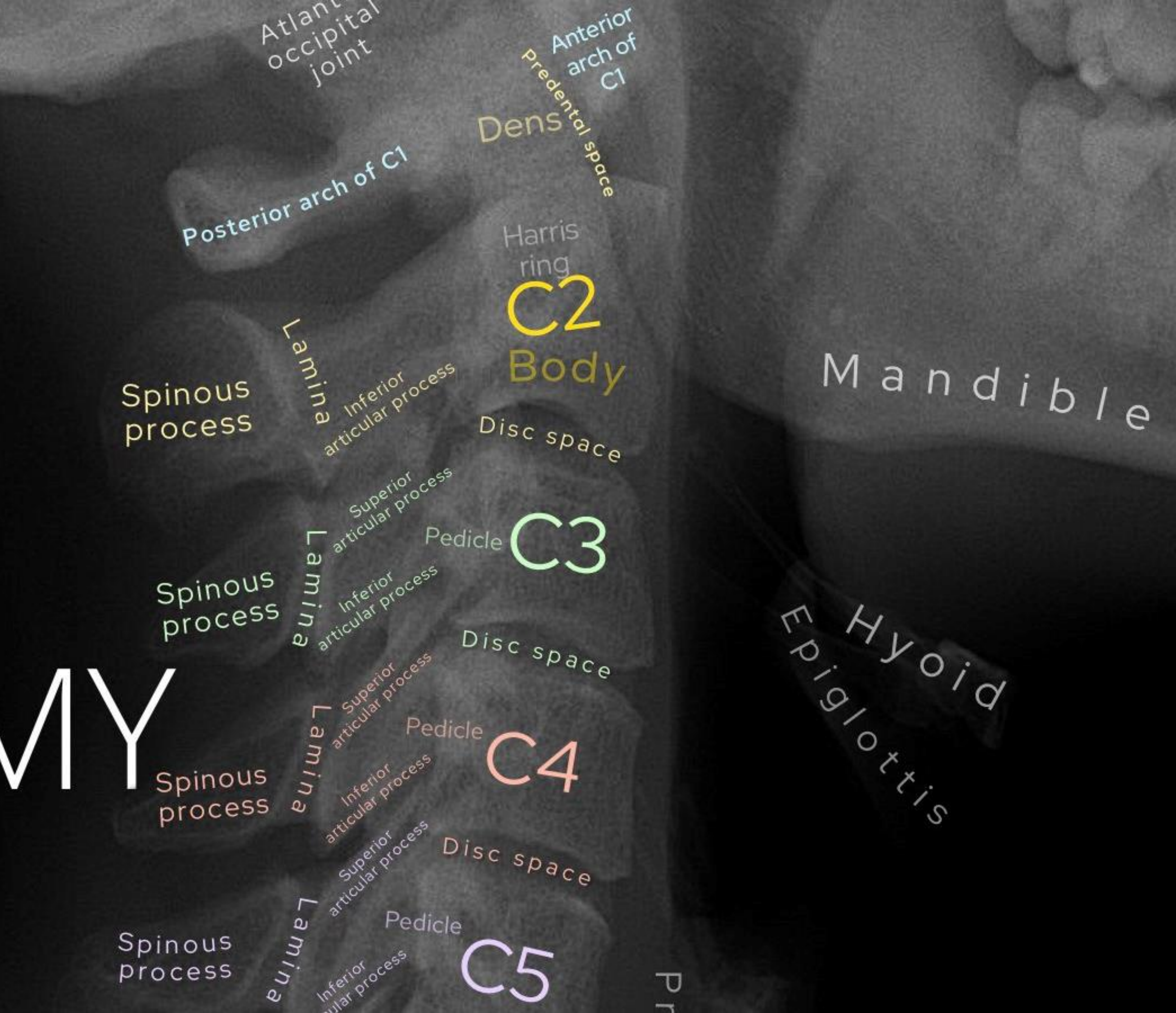
The goal of this competition is to identify fractures in CT scans of the cervical spine (neck) at both the level of a single vertebrae and the entire patient.

What is the “Cervical” region of our Vertebrae?

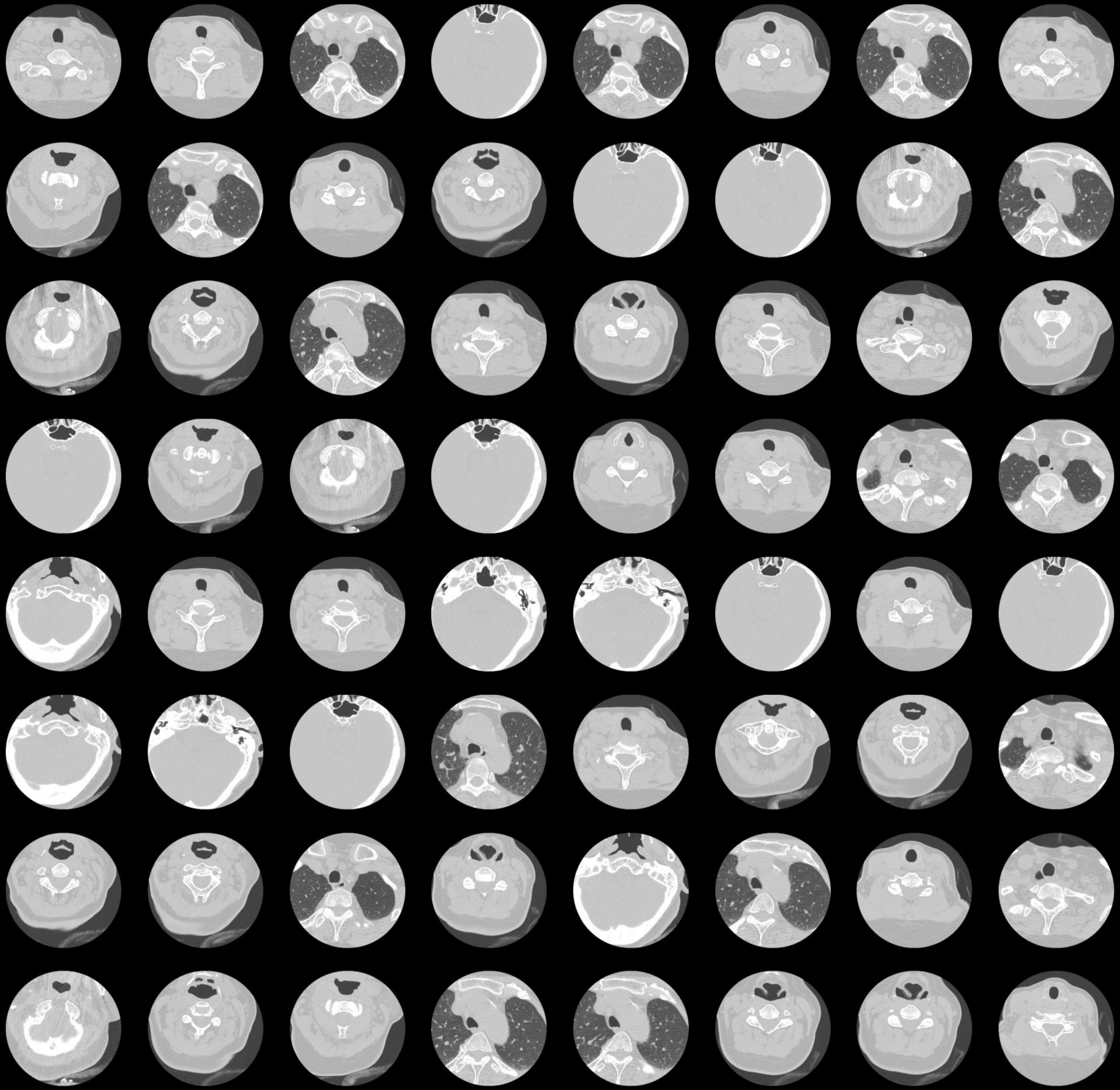
Cervical Spine X-ray ANATOMY



Cervical Spine X-ray ANATOMY



But what do the CT Scans look like?



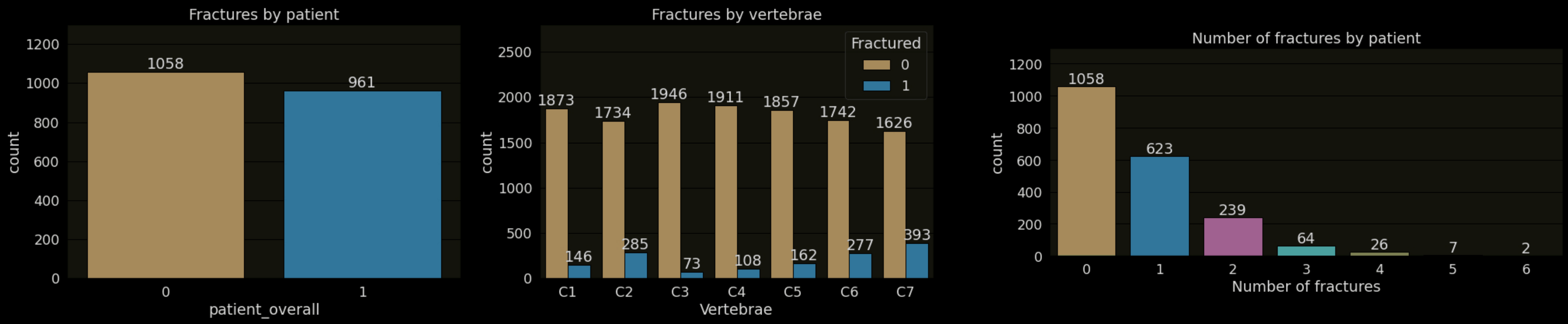
DICOM File Format

```
Dataset.file_meta -------------------------------
(0002, 0001) File Meta Information Version       OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID         UI: CT Image Storage
(0002, 0003) Media Storage SOP Instance UID      UI: 1.2.826.0.1.3680043.10001.1.101
(0002, 0010) Transfer Syntax UID                 UI: Implicit VR Little Endian
(0002, 0012) Implementation Class UID            UI: 1.2.40.0.13.1.1.1
(0002, 0013) Implementation Version Name         SH: 'PYDICOM 2.3.0'
-------------------------------------------------
(0008, 0018) SOP Instance UID                    UI: 1.2.826.0.1.3680043.10001.1.101
(0008, 0023) Content Date                        DA: '20220727'
(0008, 0033) Content Time                        TM: '175846.512627'
(0010, 0010) Patient's Name                      PN: '10001'
(0010, 0020) Patient ID                          LO: '10001'
(0018, 0050) Slice Thickness                     DS: '0.625'
(0020, 000d) Study Instance UID                  UI: 1.2.826.0.1.3680043.10001
(0020, 000e) Series Instance UID                 UI: 1.2.826.0.1.3680043.10001.1
(0020, 0013) Instance Number                     IS: '101'
(0020, 0032) Image Position (Patient)            DS: [-52.308, -27.712, -55.218]
(0020, 0037) Image Orientation (Patient)         DS: [1.000000, 0.000000, 0.000000, 0.00000
0, 1.000000, 0.000000]
(0028, 0002) Samples per Pixel                   US: 1
(0028, 0004) Photometric Interpretation          CS: 'MONOCHROME2'
(0028, 0010) Rows                                US: 512
(0028, 0011) Columns                             US: 512
(0028, 0030) Pixel Spacing                       DS: [0.253906, 0.253906]
(0028, 0100) Bits Allocated                      US: 16
(0028, 0101) Bits Stored                         US: 16
(0028, 0102) High Bit                            US: 15
(0028, 0103) Pixel Representation                US: 1
(0028, 1050) Window Center                       DS: '500.0'
(0028, 1051) Window Width                        DS: '2000.0'
(0028, 1052) Rescale Intercept                   DS: '-1024.0'
(0028, 1053) Rescale Slope                       DS: '1.0'
(7fe0, 0010) Pixel Data                          OW: Array of 524288 elements
```

HEADER

IMAGE

Baseline Solution


Fractures by patient
1200
1000
800
600
400
200
0
count
1058
961
0
1
patient_overall
Fractures by vertebrae
2500
2000
1500
1000
500
0
count
Fractured
0
1
1873
146
1734
285
1946
73
1911
108
1857
162
1742
277
1626
393
C1
C2
C3
C4
C5
C6
C7
Vertebrae
Number of fractures by patient
1200
1000
800
600
400
200
0
count
1058
623
239
64
26
7
2
0
1
2
3
4
5
6
Number of fractures

BASELINE = Only trying to predict whether a patient **has a fracture (1)** or **not (0)**

Transfer Learning

Treating each CT Scan as an IMAGE, but instead of 3 RGB Channels, the number of channels equals the number of slices/scans for that patient.

But here we run into a problem. The height of each scan varies from patient to patient, i.e. the number of slices for each patient varies.

We need a constant channel input for using CNN Architectures

Minimum number of scans for a patient = 300

Maximum number of scans for a patient = 1082

Sample <u>**N**</u> scans for each patient at equally spaced intervals. Thus making the input channel size CONSTANT

Now we can use any 2D-CNN architecture with in_channels = <u>**N**</u>

Using the mobilenetv3_large_100 architecture

Changing the input layers to take 69/300 channels.

Changing the output layer for binary classification.

Freezing the intermediate layers for training.

Setting the value of N as 69 and 300

Validation Accuracy (N=69) = 50%

Validation Accuracy (N=300) = 57 %

All values after 10 epochs

Further Improvements

01

Using 3D-CNN Architectures.

02

Using sequential models (prevent information loss).

03

Using segmentation data to isolate individual vertebrae for prediction.

Thank You

Amey Rambatla
190106011