**Cervical Spine Fracture Detection**

**Project overview (task description, evaluation metrics, etc.):**

<https://www.kaggle.com/competitions/rsna-2022-cervical-spine-fracture-detection>

**Dataset description:**

<https://www.kaggle.com/competitions/rsna-2022-cervical-spine-fracture-detection/data>

**train.csv** (metadata for the training set, 2019 patients)

* StudyInstanceUID: The study ID (unique study ID for each patient scan).
* patient\_overall: The patient level outcome (0 or 1: if any of the vertebrae are fractured).
* C1: Whether vertebrae C1 is fractured (0 or 1).
* …
* C7: Whether vertebrae C7 is fractured (0 or 1).

**test.csv** (metadata for the test set, size: 3 rows x 3 cols)

* row\_id: This will match the same column in the sample submission file.
* StudyInstanceUID: The study ID.
* prediction\_type: The target column to predict (patient\_overall, C1, C2, …, or C7).

**sample\_submission.csv** (a valid sample submission, size: 3 rows x 2 cols)

* row\_id: the row\_id in test.csv
* fractured: The target column (probability of fracture)

**train\_bounding\_boxes.csv** (bounding boxes for a subset of the training set, 239 patients, 7217 slices)

* StudyInstanceUID: The study ID (unique study ID for each patient scan).
* slice\_number: The slice index of this patient.
* x
* y
* width
* height

**train\_images/** (2019 folders, each for a patient)

**StudyInstanceUID/** (n slices for this patient, different n for each patient)

**1.dcm** (metadata and image pixels for the slice)

**2.dcm**

…

**n.dcm**

**test\_images/** (3 folders, each for a patient)

**StudyInstanceUID/** (n slices for this patient, different n for each patient)

**1.dcm** (metadata and image pixels for the slice)

**2.dcm**

…

**n.dcm**

**segmentations/** (87 folders, pixel level annotations for a subset of the training set)

**StudyInstanceUID.nii** (segmentations for each slice of this patient, 0 for pixels that are not vertebrae, 1~7 for pixels with vertebrae C1~C7)

**Project workflow:**

* Stage 1: EDA & Data Preprocessing
  + Tasks:
    - Image resizing
    - Data cleaning, column selection, etc.
  + Links:
    - <https://www.kaggle.com/code/andradaolteanu/rsna-fracture-detection-dicom-images-explore>
    - <https://www.kaggle.com/code/samuelcortinhas/rsna-fracture-detection-in-depth-eda>
* Stage 2: Vertebrae Detection
  + Tasks:
    - Assign each slice a predicted vertebrae index
      * Apply different CNN models
    - Localisation: pixel-level annotation
  + Links:
    - <https://www.kaggle.com/code/vslaykovsky/pytorch-effnetv2-vertebrae-detection-acc-0-95> (vertebrae detection by EfficientNetV2)
    - <https://www.kaggle.com/code/debarpanml/vertebrae-classification-accuracy-100> (vertebrae detection with 100% accuracy)
    - <https://www.kaggle.com/code/leventelippenszky/rsna-eda-dicom-segmentations-bboxes-3d-plot> (pixel-level annotation)
* Stage 3: Fracture Detection
  + Tasks:
    - Predict the existence of a fracture
      * Apply different CNN models
        + EfficientNetV2, DenseNet, etc.
      * Model optimization
        + Data augmentation, etc.
    - Fracture Localization
  + Links:
    - <https://www.kaggle.com/code/vslaykovsky/train-pytorch-effnetv2-baseline-cv-0-49> (fracture detection: training)
    - <https://www.kaggle.com/code/vslaykovsky/infer-pytorch-effnetv2-single-model-lb-0-49> (fracture detection: inference)
* Stage 4: Visualization
  + Tasks:
    - Vertebrae animation (pixel-level)
    - 3D image of cervical spine with fracture
  + Links:
    - <https://www.kaggle.com/code/shreydan/understanding-the-dataset-eda-animation> (animation)
    - <https://www.kaggle.com/code/leventelippenszky/rsna-eda-dicom-segmentations-bboxes-3d-plot> (pixel-level annotation)
    - <https://www.kaggle.com/code/fepegar/torchio-3d-loading-preprocessing-augmentation> (sagittal view)
    - <https://www.kaggle.com/code/andradaolteanu/rsna-fracture-detection-dicom-images-explore> (3D visualization)
    - <https://www.kaggle.com/code/songseungwon/break-time-just-enjoy-drawing-3d-cervical-spine> (3D visualization)
    - <https://www.kaggle.com/code/yiheng/loading-3d-image-and-mask-with-monai> (slice & mask animation)



