Skull Fractracture Detection

deepSkull

蔡哲民 孫欽鉉 林柏勳 宋家齊

I. Abstract

In this project, we proposed one and five architectures for case-level label and centroid-level fracture respectively. For the former, we use Resnet with 3D convolution. For the latter, each of them is Semantic Segmentation model, AutoEncoder, Faster-RCNN, RetinaNet, YOLOv5. The model performances are shown in VIII, in which YOLOv5 gives the best result

II. Dataset and Data Preprocessing

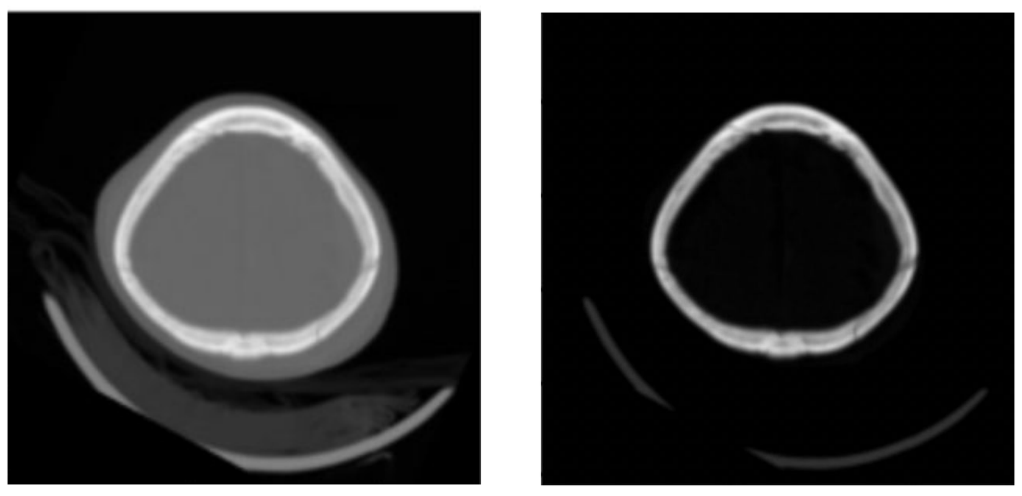
Dataset

- For all models training, we only pick images that contain skull fractures.

Data Preprocessing Method

- Clipping Hue value by the method from paper Skull-RCNN.

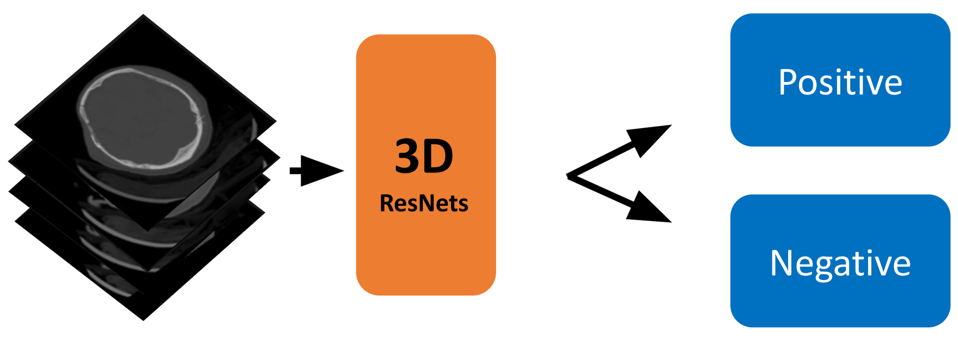
$$x = \begin{cases} 1, HU \geq 2550 \\ \frac{HU}{2550}, 0 < HU < 2550 \\ 0, HU < 0 \end{cases}$$



- Further nomarlize to [-1,1] or [0,1] by mean- max or min-max normalization.

III. 3D-CNN (Case-Level)

1. Architecture



- We use 3D-ResNets to extract features and performs the binary classification task.
- By stacking the 2D slices into a 3D structure, we can train a 3D-ResNet to perform classification task with higher dimensional features

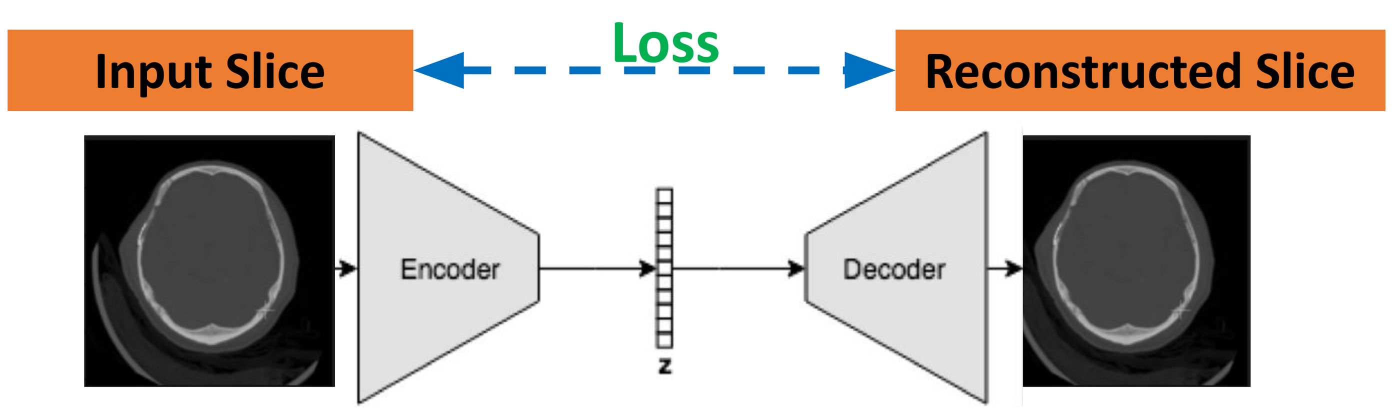
2. Loss function

$$L_{CE} = -\Sigma_{i=1}^{n} t_i \log(p_i)$$

- We choose cross entropy loss as loss function to train 3D-CNN
- The optimizer used is Adam, with learning rate 10^{-4}

IV. AutoEncoder (Centroid-Level Hit Rate)

1. AutoEncoder based method



- Anomaly structures which have never been seen during training cannot be properly reconstructed from the encoded latent space, such that the reconstruction errors will be high for anomalous structures

2. Loss function

AutoEncoder: Reconstruction Loss

- $L = \Sigma_{x \in X_N} |x - \hat{x}| + \lambda\left(-\Sigma_{x \in X_{P,P}} |x - \hat{x}| + \Sigma_{x \in X_{P,N}} |x - \hat{x}|\right)$
- $x - \hat{x}$: Pixel-wise residuals obtained from the difference of input sample x and their reconstruction $\hat{x}$
- In negative samples X_N, we want to minimize the reconstruction error, and in positive samples X_P, we want to maximize the reconstruction error in positive point $x \in X_{P,P}$ and minimize it in the negative point $x \in X_{P,N}$.
- We use ***torch.nn.L1Loss*** to implement.
- Adam optimizer, with learning rate 10^{-4}

V. Faster-RCNN (Centroid-Level Hit Rate)

1.Architecture

- We use Resnet34 without the last part as the backbone to extract the features, and other architectures are similar to the original work like RPN, classifier, etc.
- Combining three consecutive CT images didn't give better results, so we use single image and concatenate itself to three channel.

2.Method

- The number of output class is only one, which is the skull fracture.
- For training, we only pick images with fractures. For inference, we input the image and take the centers of the output bbox as the prediction result.

3.Loss function

- Adam optimizer, with learning rate $5*10^{-3}$, scaled by 0.95 every 5 epoches
- Since positive samples are less than negative ones, we modify the loss function as

$$L = 5L_{cls} + 0.5L_{reg}$$

4.Ablation Study

Bbox size	64*64	32*32	16*16	8*8
F1 score	0.19	0.05	0	x

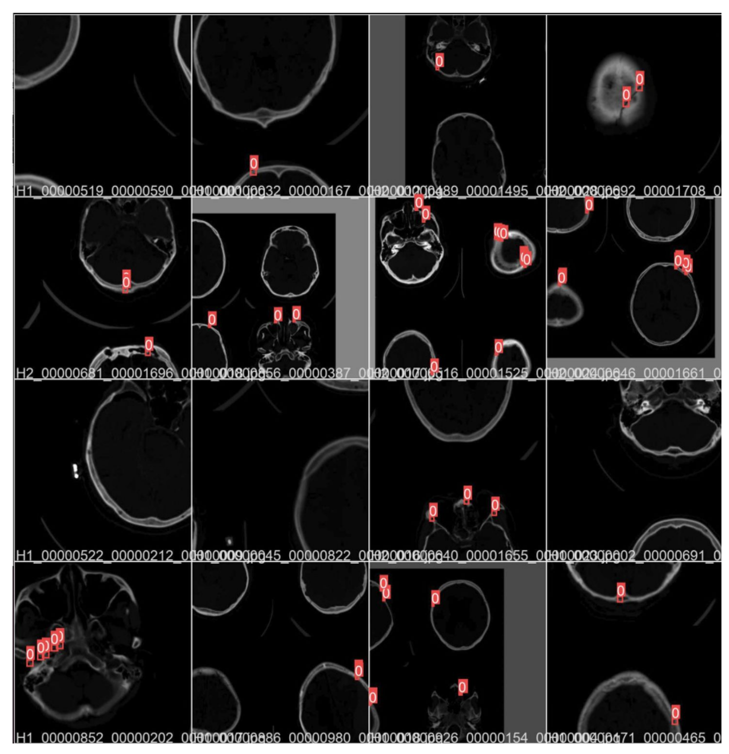
VI. YOLOv5 (Centroid-Level Hit Rate)

1. Architecture

- yolov5x. One Stage Detector

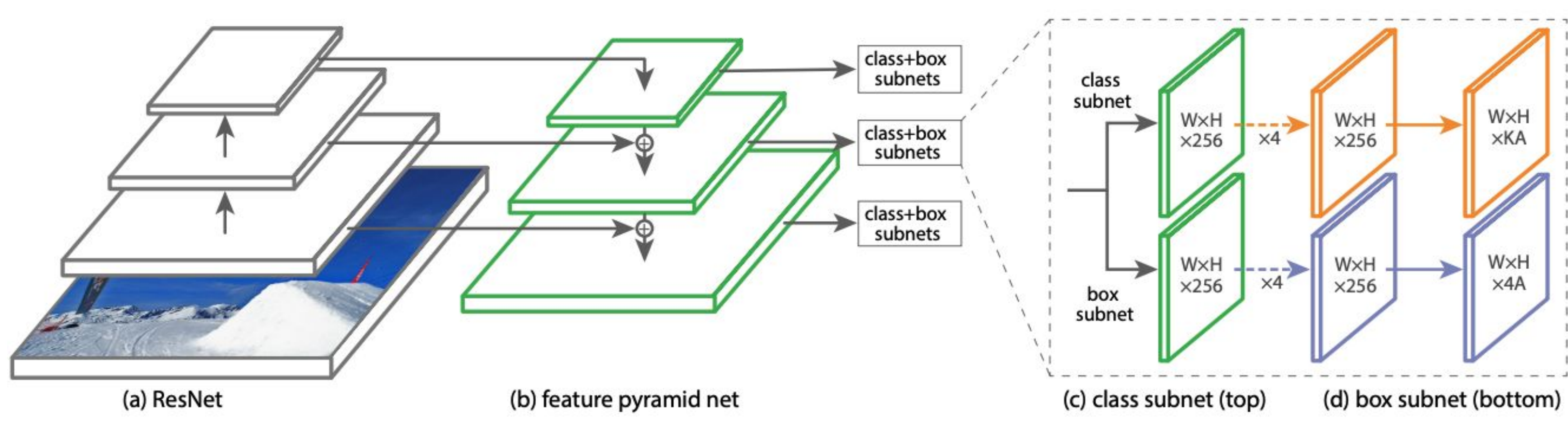
2. Data Augmentation

- Horizontal Flip
- Rotation
- Translation
- Test Time Augmentation



Mosaic Dataloader

VII. RetinaNet (Centroid-Level Hit Rate)



- FPN efficiently constructs a rich, multi-scale feature pyramid from a single resolution input image
- Focal loss down-weight easy samples, alleviate data imbalance
- Classification subnet is attached to each FPN level, and parameters of this subnet are shared across all pyramid levels
- Bounding box regressor with fewer parameters but equally effective

VIII. Model Results

	Resnet18	Resnet34	Resnet50	Resnet101
Case Level	0.859 (z_dim=48)	0.862 (z_dim=48)	0.871 (z_dim=48)	0.849 (z_dim=36)

	AutoEncoder	Faster-RCNN	3D-UNet	RetinaNet	YOLO
F1 score	0.17	0.51	0.1	0.35	0.71