National Taiwan University

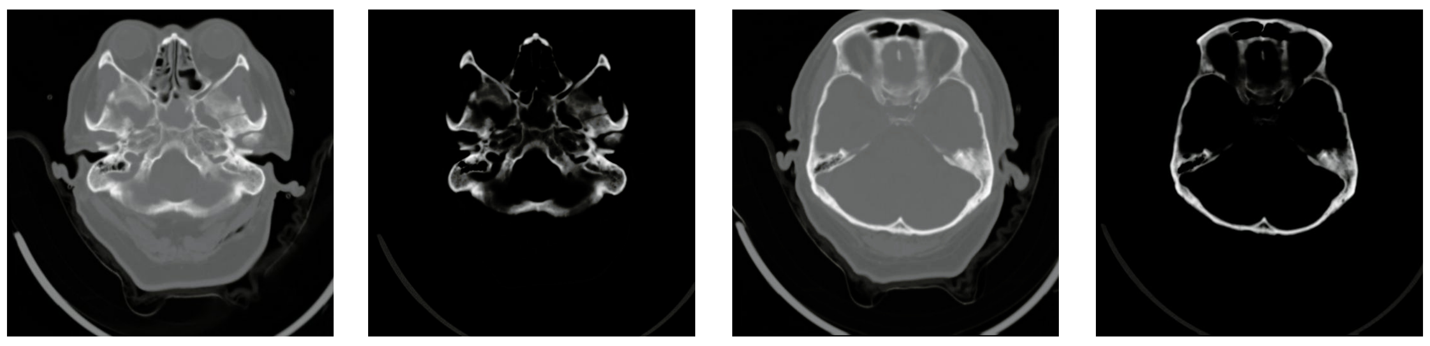
DLCV CHALLENGE I SKULL FRACTURE DETECTION

BY NAIVE

B07901019吳隆暉 B08901022吳彬世 B08901039傅譽 B08901054楊學翰 B08901065江浩瑋

CASE LEVEL CLASSIFICATION

Data Preprocess - Contrast Stretching



Vision Transformer [1]

Patch size	Class token dimention	Depth	Heads	Input size
16	128	12	8	512

Loss Function: Cross-Entropy Loss with imbalanced weight (1:10)
Optimizer: Adam(lr=3e-5), Scheduler: StepLR(step size=1, gamma=0.7)

Results & Findings

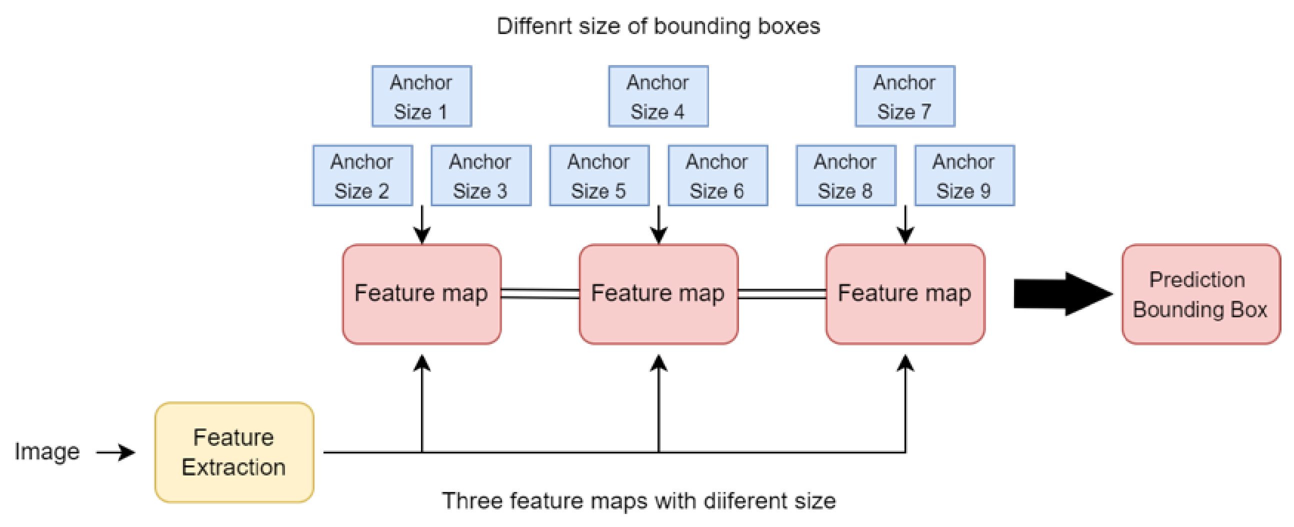
Case Level Accuracy: 0.83

1. The model overfits easily, validation loss reaches the lowest point fast
2. It's hard to observe the performance of classification from validation loss
3. Need more time to fine-tune the model using lower learning rate

CENTROID LEVEL F1 SCORE

Approach 1: YOLOv3 [2]

Prediction Flow



Model Architecture

1. Feature extractor : Darknet-53 (include 53 CNN layers)
2. Detection : include additional 53 CNN layers and upsampling
3. Prediction : Use IOU to choose anchors and bounding box

Loss Function

1. BCE loss between bounding box's coordinates and ground true
2. BCE loss between bounding box's (height and width) and ground true (set by ourselves)
3. BCE loss between predicted labels and ground true labels

Results

1. The model tends to predict all 0 because it can make the loss closed to 0.
2. The unbalanced data will affect the performance.
3. After trying, we can't use this model to detect the skull fracture correctly.

Possible Reasons

1. We don't have the ground true of bounding boxes
   --> Decided by random number from (6, 6) to (20, 20)
2. The BCE loss on the bounding box's h&w will lead to errors.
3. YOLOv3 use three different sizes of feature maps to detect different size of objects, which is not fit for this task.

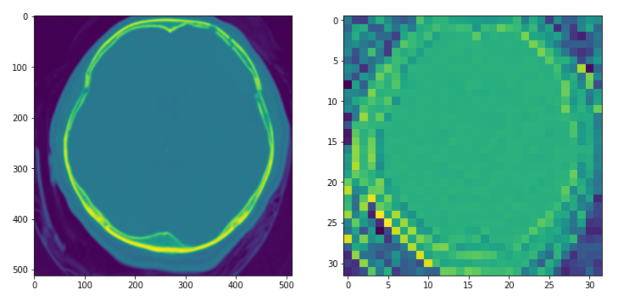
Approach 2: Attention Map

Model Architecture

1. We design two attention maps, one extract from the vision transformer in case-level classification, the other generated by our own self-attention layer.
2. Loss function: MSE Loss of the attention map and the ground-truth map
3. We also tried designing an algorithm for picking coordinates from attention map, but the performance is poor

Results

The locations of fractures didn't learnt well, so the intensity information in the map has little meanings.
Attention map from ViT:
F1 score on validation set = **0.2**
Attention map we designed:
F1 score on validation set = **0.35**



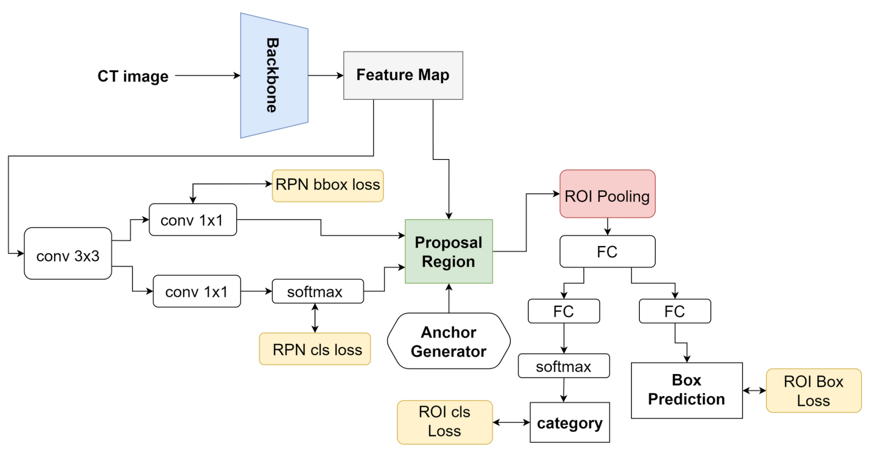
Approach 3: Faster RCNN

Data Preprocess

1. Transform the CT scans from npy format to gray-scale images
2. Convert gray-scale images into 3 channels
3. Normalize images using mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]

$$I = \begin{cases} 255 & HU > 2550 \\ \frac{HU}{10} & 0 < HU < 2550 \\ 0 & HU < 0 \end{cases}$$

Model



Loss function = RPN box loss + RPN class loss + ROI cls loss + ROI Box loss
Set 32x32 as the size of each ground truth coordinate
Only train on the images which have at least one fracture

Backbone	Optimizer	anchor sizes	aspect ratio	ROI output size	learning rate
inceptionv3	Adam	(16, 32, 64)	(0.5, 1.0, 2.0)	7	1e-3

Results & Findings

F1 Score: 0.51

1. False Negative is high in validation set.
2. Performance may depend on case level accuracy.
3. Faster RCNN is designed to detect objects with different size and different categories.
4. For a larger fracture, we need to output multiple coordinates, but RCNN will only predict a larger bounding box.
5. Does not consider CT images as sequential data

Experiments

Backbone	ResNet152	VGG16	Inceptionv3	MobileNetv2	ResNet50FPN*	MobileNetFPNv2*
F1 score	0.75827	0.7327	0.7072	0.6734		0.6887
TP	1421	1605	1464	1341		1516
FN	782	598	739	862		687
FP	124	573	473	439		683

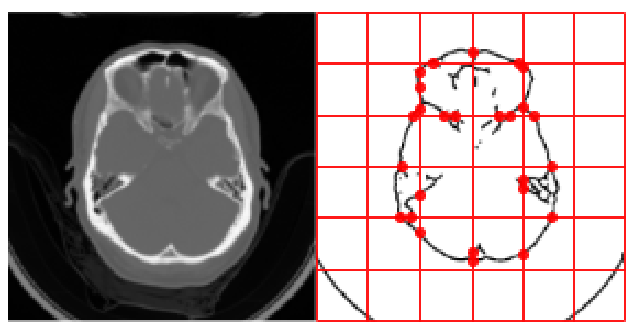
Approach 4: Patch-wise classification with inception v3 [3]

Model Overview

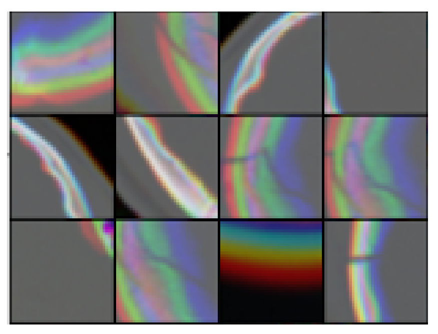
1. Inspired by two-stage object detection R-CNN
2. Replace region proposal network with skeletonization and patch sampling
3. Binary classification with each 64*64 patch using inception v3 network
4. Positional encoding: patch coordinate & slice sequence no.
5. Utilize sequential data by concatenating nearby patches into a 3-channel image

Data Preprocess

1. Skeletonize 2. Patch Sampling & Data Augmentation 3. Sequence Concatenation



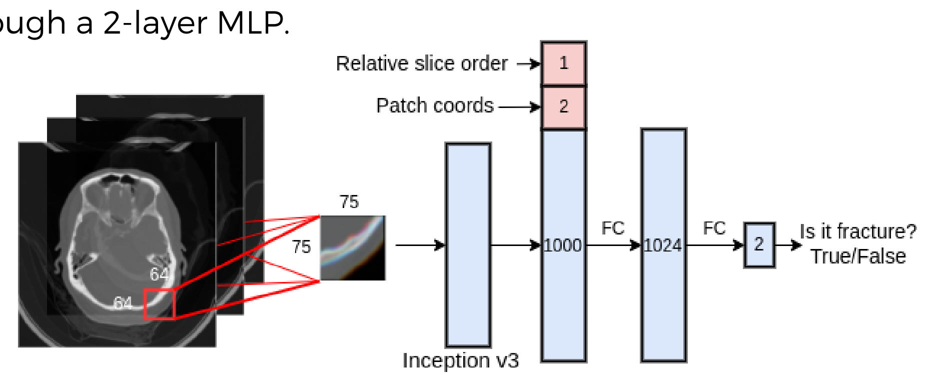
▲ Skeletonization & patch sampling



▲ Sequence concatenation

Model Architecture

We used inception v3 and concatenated the output with positional encoding, then passed the results through a 2-layer MLP.



Loss Function

Focal loss with gamma=1 and alpha=0.5

Result & Findings

F1 score: 0.47

1. Number of fractures detected depends on the number of sample points
2. Unbalanced data augmentation on positive and negative labels may lead to large FP

Experiments

1. Without positional encoding
2. Without channel concatenation
3. Different patch size

Reference:
[1] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.
[2] Improve YOLOv3 using dilated spatial pyramid module for multi-scale object detection - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/The-structure-of-YOLOv3-YOLOv3-predicts-objects-at-three-different-scales-and-finally_fig1_343170773 [accessed 16 Jan, 2022]
[3] http://proceedings.mlr.press/v121/kuang20a/kuang20a.pdf