Skull Fracture Detection

Min-Han Shih, Geng-Ruei You, Yang-Che Tseng, Jr-Jen Chen

Min-Han Shih , Department of Electrical Engineering , National Taiwan University b08502141
Geng-Ruei You , Department of Electrical Engineering, National Taiwan University b08901211
Yang-Che Tseng , Department of Computer Science and Information Engineering, National Taiwan University b08902134
Jr-Jen Chen , Department of Mechanical Engineering, National Taiwan University b07502071

Goal

Train a neural network to detect if the patient has skull fractures and where they are by using CT images.

Introduction

Deep Learning on Skull Detection

Some studies have attempted the automatic detection of skull fractures from the CT scans using the classical computer vision methods. While these methods only **considered local features** for the prediction of skull fracture, some novel deep learning approaches have achieved remarkable progress in such task. Such methods include **extracting features of images** through a cascade of several layers of non-linear processing units in an attempt to explain the representations of the image data based on the learning of multiple levels of features.

Data Preprocessing

- Turn these CT images into gray-scale images.
- Normalize them into [0, 255].
- Split YOLO predictions into 3 classes: [1]fractures, [2]background with fractures, [3]background without fractures. The concept is inspired by contrastive learning where we utilize negative sample to enable the model to learn the difference between images with and without fractures.

Our methods

- Object Detection with YOLOv5
- Pipelined Detection with YOLOv5
- R-CNN
- Detection with Transformer

Implementation Detail

Object Detection with YOLOv5

In this approach, we adopt the object detection method [1] and replaced YOLOv3 [2] with its lightweight alternative YOLOv5 for a faster training procedure. We detect the fracture by identifying the tightly enclosing rectangular bounding boxes defined by a 4-tuple (x, y, h, w), indicating its position and size. We arbitrarily regard the center point of each bounding box as the fracture point. With this approach, we output -1 if no bounding box is presented and 1 otherwise. However, the ratio between positive and negative samples is extremely unbalanced, and we solved it by using all positive samples and only taking partial negative smaples, keeping their ratio at 9:1. We train the model with Adam optimizer of learning rate 10^{-2} and the 1cycle learning rate policy [3]. Binary cross-entropy loss (BCELoss) is introduced to predict whether each slice contains skull fracture. After that, we postprocess our prediction by eliminating cases with non-contiguous fracture predictions and see great improve in case-level prediction accuracy.

Pipelined Detection with YOLOv5

Following the previous method, we replace the post processing by a case-level based classification in belief that the deep learning method should prevail the heuristic post processing algorithm. We trained another model with 3 labels for object detection but failed, and it turned out that the model works well in classification. We first classify whether a case is positive, and we detect the fracture afterwards only for those positive case.

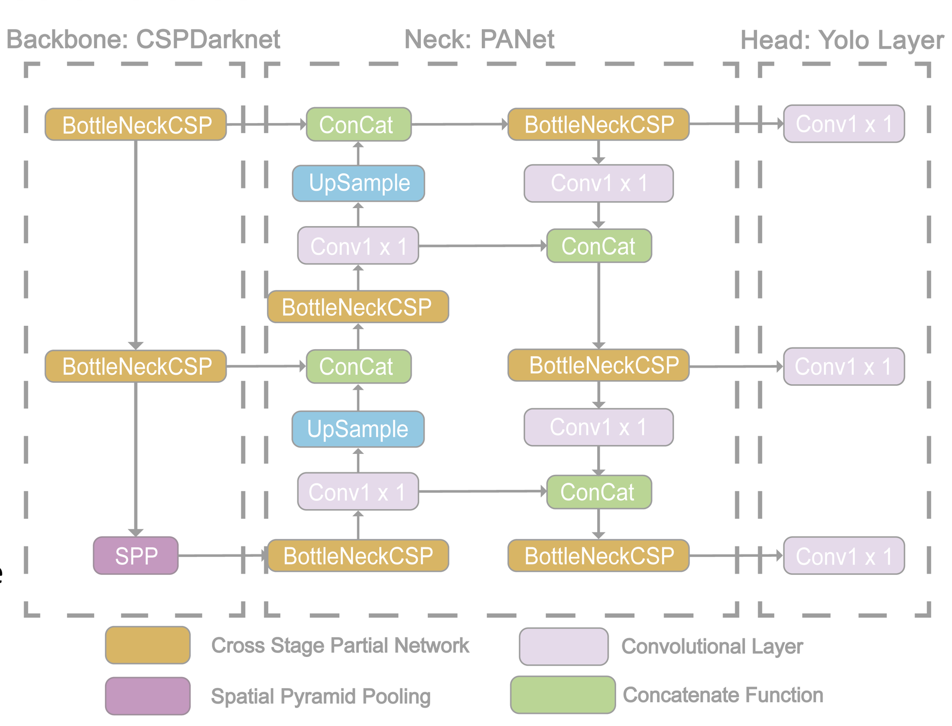


Fig. 1
Yolov5 structure

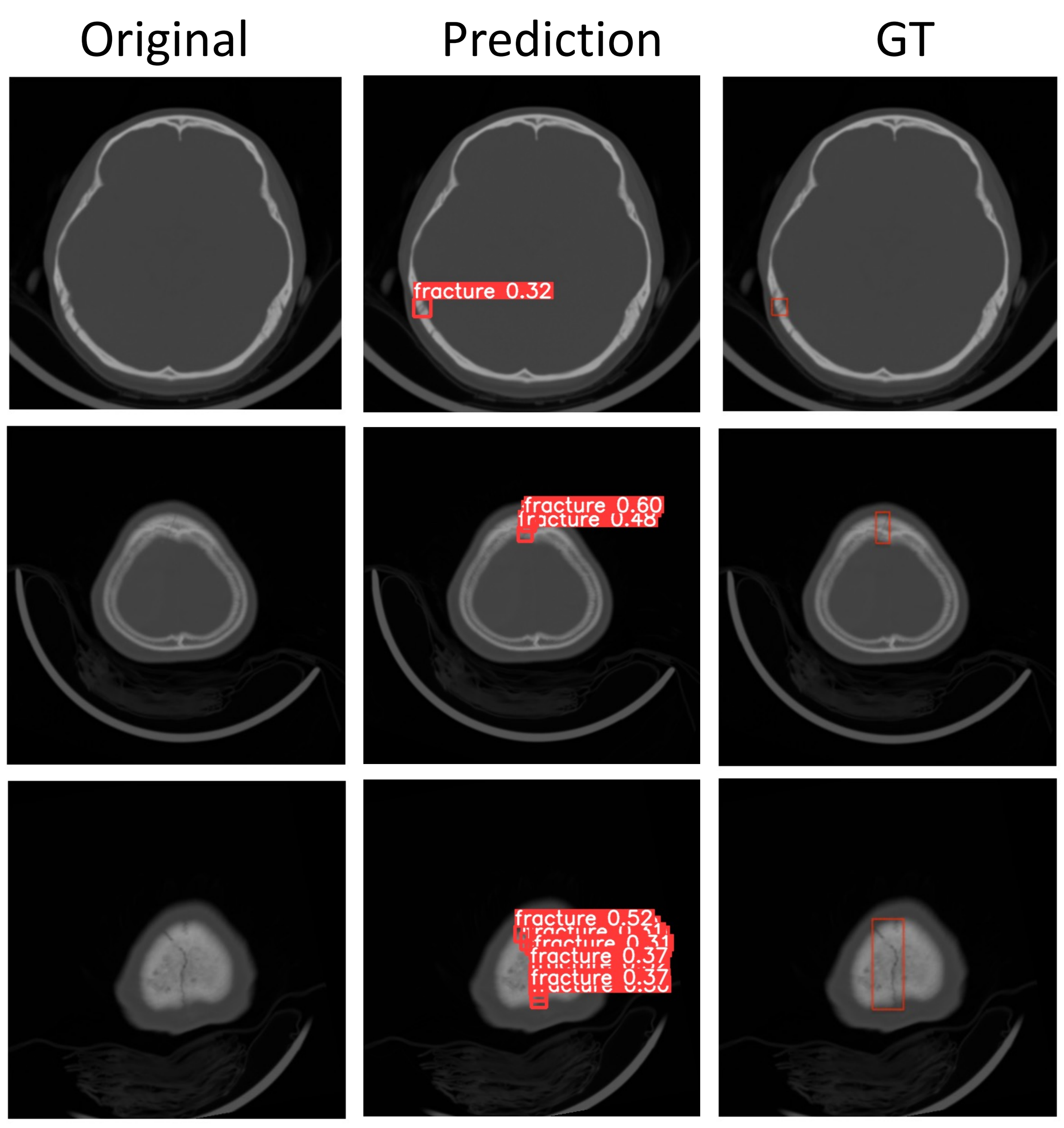
Skull R-CNN

Other than region-free network, YOLOv3, we also tried a novel regional-proposal CNN dedicated to this task, Skull R-CNN [4]. The modified Faster R-CNN [5] integrates the skull morphological features with CNN in an attempt to accurately detect the skull fractures; here, however, we apply Mask R-CNN [6], which is based on top of Faster R-CNN. The pipeline includes skeletonization and multi-scale feature extraction, Roi-Align [6] and classification. BCELoss is also used here for binary classification.

Detection with Transformer

While convolution neural network has seen promising performance in various computer vision tasks, the arise of attention mechanism [7] has render great results [8][9] in computer vision. Despite the claim [10] that Transformer demonstrates lower performance on small objects which is our case, we tried to apply Transformer to slice of images and let the decoder predict the position of the fracture. As the number of parameters in the Transformer is nearly double of the other methods', the training requires prolonged time.

Model Prediction Visualization



Result

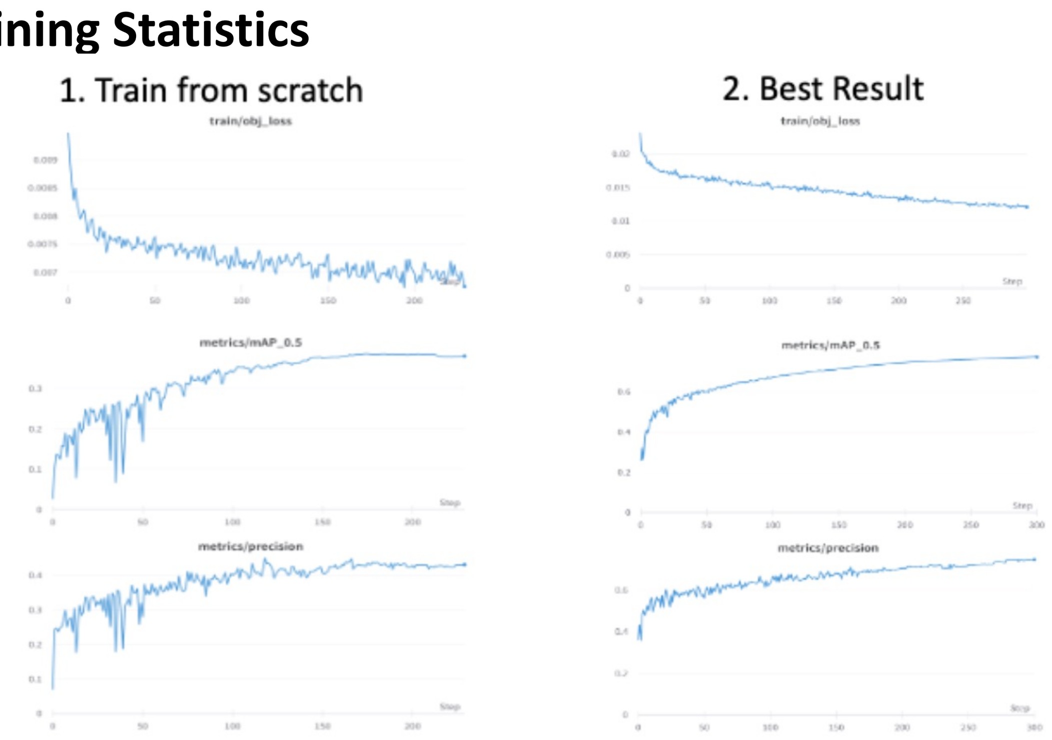
Evaluation

We evaluate our performance by four metrics, case-level accuracy, mAP and Precision & Recall and F1 score.

case	acc.	mAP	P&R	F1
YOLOv5				
vanilla	-	0.3805	0.43, 0.45	-
pre. s	0.52308	0.7736	0.74, 0.69	0.61
pre. l	0.8	-	-	0.66
YOLOv3	0.846	0.886	0.89, 0.82	0.68
Pipelined	0.8769	0.7736	0.74, 0.69	0.67
Skull R-CNN	-	-	-	-
DETR	-	-	-	-

1 pre.: we loaded pretrained weights of YOLOv5, m and x indicates the size

Training Statistics



Ablation

case	acc.	mAP	P&R	F1
YOLOv5				
w/o norm.	data MIA			
w/ norm.	0.5230	0.7736	0.75, 0.70	0.61
bb = 24	0.5230	0.7736	0.75, 0.70	0.61
bb = 12	0.5308	0.9173	0.89, 0.85	0.62
2 labels	0.5230	0.7736	0.75, 0.70	0.61
3 labels	0.5153	-	-	0.016
w/o po.p.	0.5230	0.7736	0.75, 0.70	0.61
w/ po.p.	0.7308	0.77	0.75, 0.70	0.63
Pipelined				
w/o po.p.	0.8	0.7736	0.74, 0.69	0.663
w/ po.p.	0.88	0.7736	0.74, 0.69	0.668

2 norm.: normalization
3 bb: bounding box
4 po.p.: post processing

Future Work

Novel Attention-Based Detector

While we seen great performance using pipelined method, the mutli-stage training requires considerable training time. Therefore, we present the end-to-end 3D Transformer based Detector in an attempt to identify the skull fracture in a sequence to sequence manner. While similar case could be found in some related work [11], they usually focus on large object instead of subtle ones.

The model takes a sequential CT images with dimension $(512, 512, N)$ as input and predicts the position of the fracture defined by the 3-tuple (x, y, h), where N is the number of the slices of a patient and h is the index of the slice containing the fracture. The implementation detail is not yet completed as we still need to conduct more research and experiment. For instance, we are finding out whether to use 2D-sliced positional embedding like [8] or use 3D embedding. Moreover, we found it difficult to design the loss against the prediction set, and hence we temporarily adopt the bipartite matching loss [10][11] which we found viable. We are currently researching on keyword extraction and believe that we could integrate such task to ours.

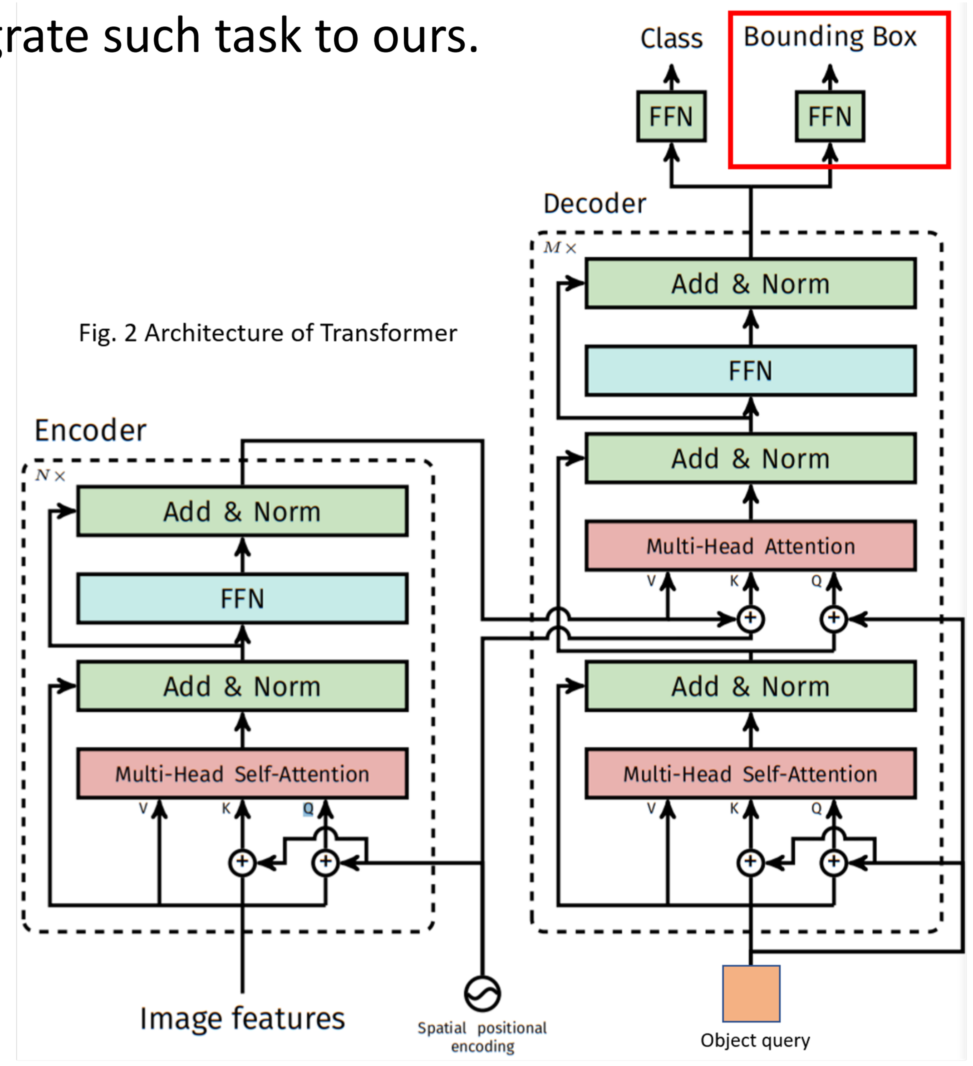


Fig. 2 Architecture of Transformer

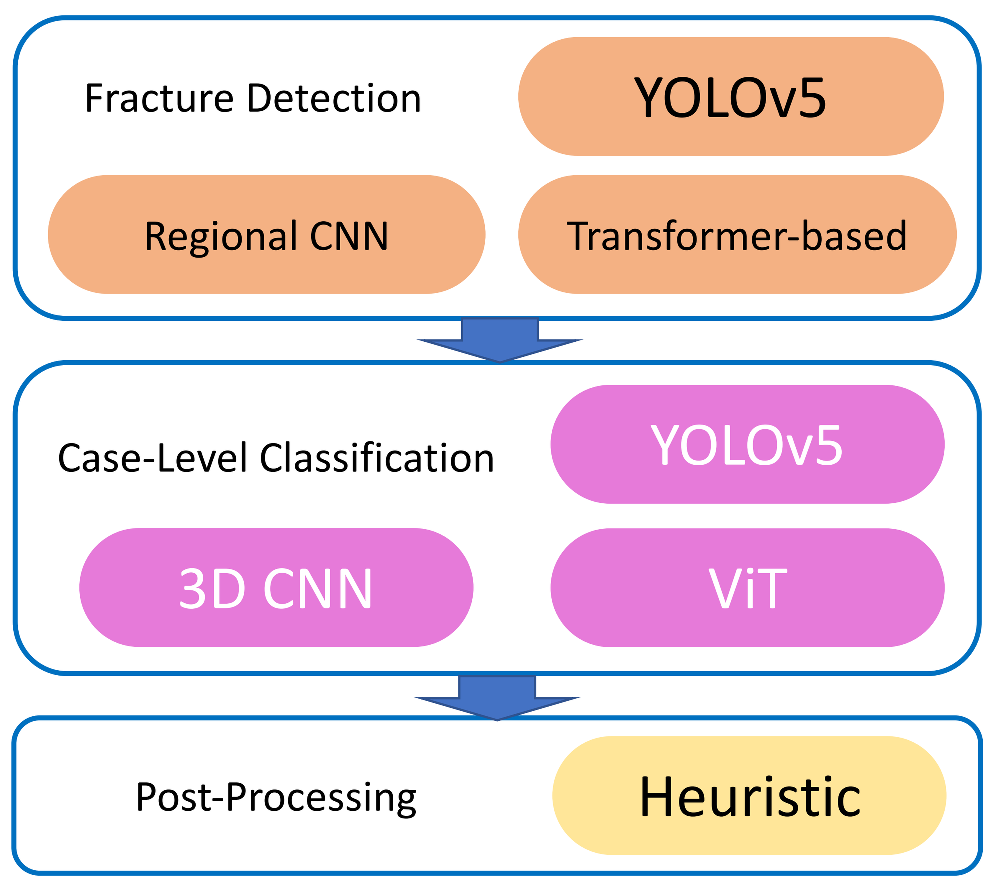


Fig. 3
Our Pipeline