UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

UNDERGRADUATE SCHOOL

Machine Learning in Medicine Final Report

By

Nguyen Phan Gia Bao BI12-048

Can Trung Hieu BI12-160

Chu Bao Minh BI12-281

Doan Hoai Nhi BI12-338

Academic years: 2023 - 2024

Tilte:

Abdominal Trauma Recognition

Contents

1 Introduction

Abdominal CT analysis is pivotal in data-driven scientific image processing, providing valuable insights into the human abdominal region's intricate structures and physiological characteristics. The field encompasses a diverse range of applications, from medical diagnostics to biomechanical studies, where the focus is primarily on extracting and analyzing relevant features from abdominal images. In this context, abdominal images refer to visual representations obtained through various imaging modalities such as computed tomography (CT) scans, magnetic resonance imaging (MRI), and ultrasound, capturing the internal complexities of organs, tissues, and surrounding anatomical structures.

Analyzing abdominal images involves developing and applying advanced algorithms and techniques within data science and image processing. These methods aim to enhance imaging data quality, facilitate accurate anatomical structure segmentation, and extract quantitative information for diagnostic and research purposes. Machine learning and deep learning algorithms have become increasingly prevalent, offering promising avenues for automated image interpretation and pattern recognition in abdominal imaging. This intersection of abdominal imaging and data science not only holds immense potential for medical advancements but also opens new avenues for understanding the physiological intricacies of the abdominal region in diverse scientific disciplines.

According to the "Global, regional, and national burden of chronic kidney disease, 1990-2017: a systematic analysis for the Global Burden of Disease Study 2017"[1], the number of people with kidney disease exceeds 2.5 million. It is projected to double to 5.4 million in 2030. Moreover, "Liver and Intrahepatic Bile Duct Cancer - Cancer Stat Facts"[2] shows that the estimated number of new cases of liver cancer in 2023 is 41,210, accounting for 2.1% of all new cancer cases. The number of deaths due to liver cancer in 2023 is estimated to be 29,380, accounting for 4.8% of all cancer deaths. In addition, "Epidemiology and management of splenic injury: An analysis of a Chinese military registry"[3] describes that from July 2000 to March 2009, 7,807 patients with splenic injuries were admitted to Chinese military hospitals. Of these patients, 84.3% were male, and 15.7% were female. The mortality rate was 0.9%.

The statistics above show that kidney, liver, and spleens-related diseases are hazardous and prevalent. Detection and classification of injuries are essential to effective treatment and favorable outcomes. A large proportion of patients with abdominal trauma require urgent surgery. Abdominal trauma often cannot be diagnosed clinically by physical exam, patient symptoms, or laboratory tests. Therefore, our group has developed a model to address this issue. This model will identify and predict the affected organs so that experts can quickly treat and diagnose patients.

2 Objectives

In the wake of abdominal trauma, swift and accurate diagnosis is paramount for saving lives and minimizing long-term complications. Traditionally, this process relies on the expertise of radiologists to analyze CT scans, but time constraints and subjective factors can introduce uncertainties. This research strives to address these limitations by exploring the potential of machine learning to predict the severity of injuries in three vital organs – kidneys, liver, and spleen – with enhanced speed and precision.

Before designing the algorithm to handle this problem, we want to clarify our challenges:

- Challenge 1: Adjacency slices are essential. Diagnosing CT images based on a single image is unscientific because many situations must be judged by considering adjacent slices [4].
- Challenge 2: Limitation of data labels. Labeling at the image level for sequential medical images like CT scans is time-consuming and unrealistic since each series contains from 41 to more than 1000 slices. The data has only annotation for each series, so we must train a model using a few simple labels.
- Challenge 3: Redundant noisy data. Not all slices in a series reflect the trauma. Processing these will affect our algorithm's time complexity. To make a swift and accurate diagnosis, we remove a part of the series to achieve an ideal inference time.

We propose a model architecture built upon the foundation of readily available CT scans. It commences with the powerful object detection capabilities of YOLOv8[6], pinpointing the targeted organs amidst the complex anatomy. The model extracts rich and nuanced features specific to each organ from there by employing convolutional neural networks (CNNs). These features capture the telltale hallmarks of injury severity, ranging from subtle structural changes to extensive tissue damage.

The model employs slice attention. As in challenges 2 and 3, we must consider all results from each slice, but not all contain important information. It is reasonable to assign a weight for each slice based on the prediction of CNNs and confidence of the YOLOv8(whether or not it contains the organ in that slice).

We aim to meticulously evaluate the performance of our model, scrutinizing its accuracy, sensitivity, specificity, and, most importantly, processing speed. The main goal is to identify the most optimal approach for rapid and reliable organ injury prediction by comparing their effectiveness. Beyond achieving superior diagnostic accuracy, the ultimate goal is to significantly shorten the diagnostic window, enabling critical interventions to be initiated with unparalleled speed. Such advancements can potentially translate into better patient outcomes, reducing long-term morbidity and mortality associated with severe abdominal trauma.

This project represents a significant leap toward realizing the transformative potential of artificial intelligence in enhancing medical diagnosis. By harnessing the power of machine learning to interpret CT scans with remarkable speed and precision, we hope to empower clinicians in their fight against the time-sensitive challenges of abdominal trauma, ultimately saving lives and improving the quality of care for countless patients.

3 Competition Overview

3.1 Introduction to competition

The RSNA 2023 Abdominal Trauma Detection competition on Kaggle emerges as a pivotal event within the expansive landscape of medical imaging. This influential challenge, orchestrated by the Radiological Society of North America (RSNA), is intricately designed to address the intricate task of detecting injuries within critical abdominal organs—specifically, the kidney, spleen, and liver—utilizing advanced techniques in machine learning.

The evaluation of competition results goes beyond mere quantitative metrics. While accuracy and sensitivity remain key considerations, the competition places a unique emphasis on the model's ability to self-assess and elucidate its decision-making process. This intentional focus on transparency underscores the importance of understanding and interpreting the inner workings of the machine learning models, paving the way for responsible and ethically sound applications in the medical field.

In conclusion, the RSNA 2023 Abdominal Trauma Detection competition transcends the boundaries of a conventional data science challenge. It stands as an invaluable opportunity for participants to contribute meaningfully to the advancement of healthcare by enhancing diagnostic capabilities for abdominal injuries through cutting-edge technology. Beyond the competitive landscape, the event serves as a dynamic platform for networking, collaborative learning, and continuous self-improvement in a community united by a shared passion for pushing the boundaries of innovation in medical diagnostics.

3.2 Competition Dataset and Analysis

We use the Abdominal Trauma Detection dataset from Kaggle as part of The RSNA Abdominal Trauma Detection AI Challenge. The CT scan data is in DICOM format. DICOM format provides several advantages for data science. It ensures standardization, consistency, and interoperability across various imaging devices and systems. Key terms included "patient_id", "series id", "instance_number", "kidney", "spleen", and "liver". This dataset has 3147 patients with about 4000 scans. There will be 3 levels of organ disease which are denoted as 0 is healthy, 1 is low, and 2 is high.

- **patient_id** - A unique ID code for each patient.
- **series_id** - A unique ID code for each scan.
- [**kidney/liver/spleen**] [**healthy/low/high**] - The three injury types with three target levels.
- **instance_number** - The image number within the scan. The lowest instance number for many series is above zero, as the original scans were cropped to the abdomen.

The number of healthy patients is much larger than that of critical patients. Healthy patients are those with all components in a healthy condition, while dangerous patients are those with at least one component at level 3.

The percentage of people injured by active extravasation is 53%, while the percentage of people injured by bowling is 47%. This chart shows an equal distribution of injuries between active extravasation and bowling.

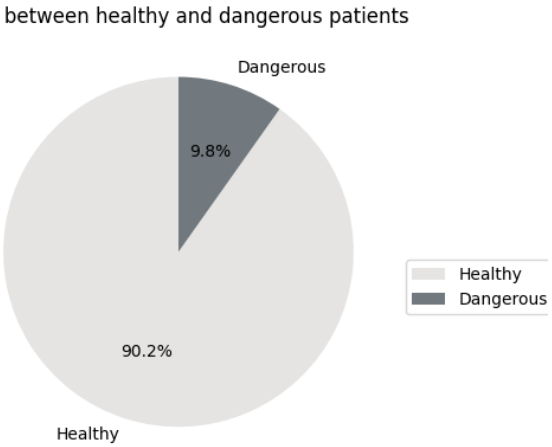


Figure 1: Ratio between healthy and dangerous patients

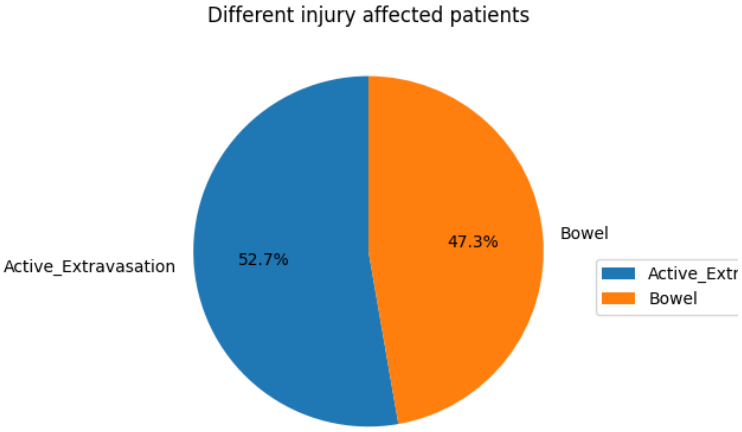


Figure 2: Ratio between Active_Extravasation and Bowel

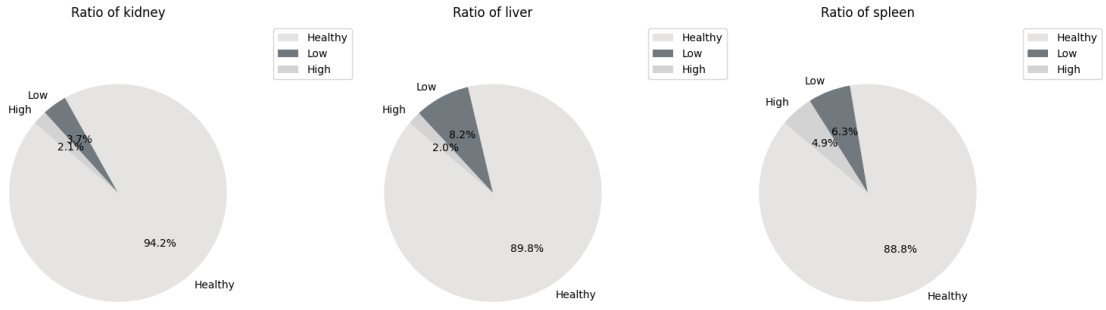


Figure 3: Ratio of the class of the organ

We can see that the 'healthy' level predominates within each organ. Additionally, the number of organs at the 'high' level is relatively low and accounts for less than 5% of each organ, which is highly positive and necessary.

4 Methodology and Material

To simplify the matter, we assume that the trauma of each organ does not correlate. Since the series of each patient contains multiple images that do not fully cover the solid organ, we will go through every CT scan within a series to localize kidneys, livers, and spleens. After extracting each organ from an image, we stack all the cuts belonging to each organ within a series. These stacks represent a 2.5D representation of each organ in a series. Finally, we feed these stacks through feature extraction layers and LSTM to obtain the injured level of the organ.

4.1 Training pipeline

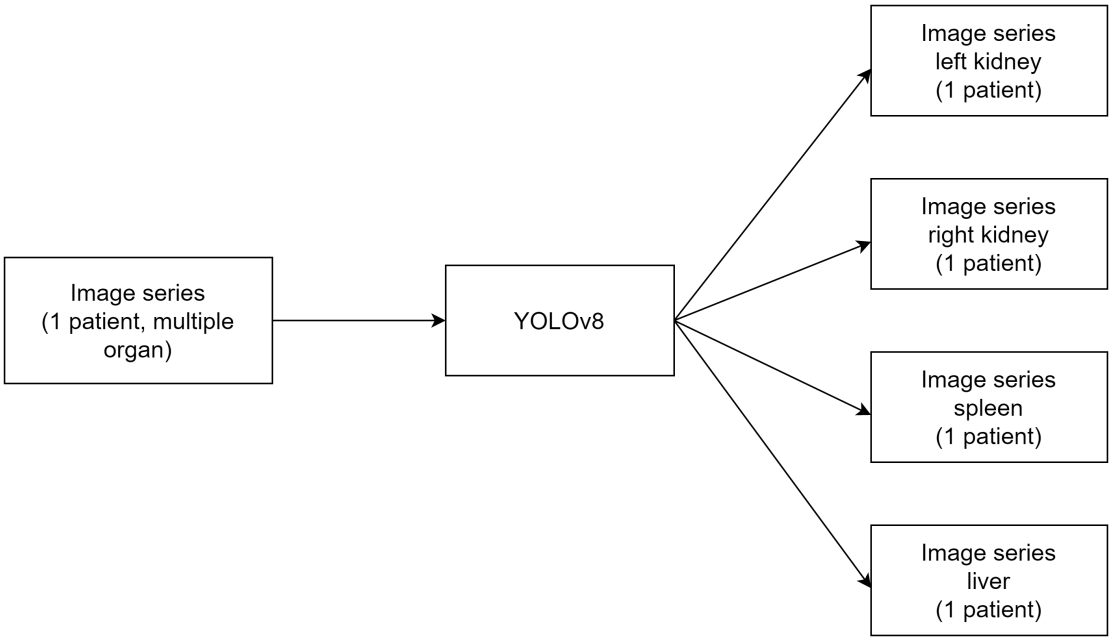


Figure 4: Pipeline for detection phase

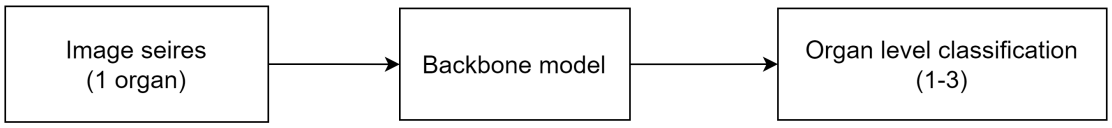


Figure 5: Pipeline for classification phase

4.2 Detection Phase

4.2.1 YOLOv8 Overview

We utilize YOLOv8 nano version for the detection phase, as it is the new state-of-the-art computer vision model. This latest version has the same architecture as YOLOv5 with numerous improvements, such as a new

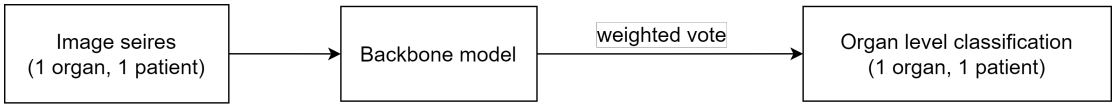


Figure 6: Pipeline for inference phase

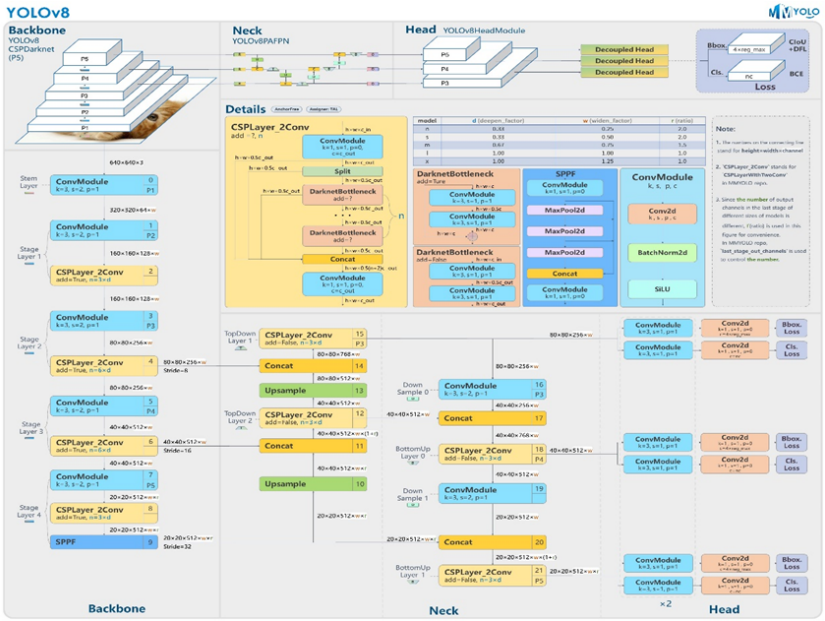


Figure 7: YOLOv8 architecture [5]

neural network architecture that utilizes both Feature Pyramid Network (FPN) and Path Aggregation Network (PAN). These features make it easier to annotate images for training the model. The FPN works by gradually reducing the spatial resolution of the input image while increasing the number of feature channels. This results in feature maps capable of detecting objects at different scales and resolutions. The PAN architecture, on the other hand, aggregates features from different levels of the network through skip connections. By doing so, the network can better capture features at multiple scales and resolutions, which is crucial for accurately detecting objects of different sizes and shapes[8].

YOLOv8 uses CSPDarknet53[7] as its backbone, a deep neural network that extracts features at multiple resolutions (scales) by progressively down-sampling the input image. The feature maps produced at different resolutions contain information about objects at different scales in the image and different levels of detail and abstraction. YOLOv8 can incorporate different feature maps at different scales to learn about object shapes and textures, which helps it achieve high accuracy in most object detection tasks. YOLOv8 backbone consists of four sections, each with a single convolution followed by a c2f module[6]. The c2f module is a new introduction to CSPDarknet53. The module comprises splits where one end goes through a bottleneck module (Two 3x3 convolutions with residual connections). The bottleneck module output is further split N times where N corresponds to the YOLOv8 model size. These splits are all finally concatenated and passed through one final convolution layer. This final layer is the layer where we will get the activations.

4.2.2 Loss Function and Update Rule of YOLOv8

The generalized loss function and weight update procedure can be defined as follows:

$$\mathcal{L}(\theta) = \frac{\lambda_{box}}{N_{pos}}\mathcal{L}(\theta) + \frac{\lambda_{cls}}{N_{pos}}\mathcal{L}_{cls}(\theta) + \frac{\lambda_{dfl}}{N_{pos}}\mathcal{L}_{dfl}(\theta) + \phi\left\|\theta_2\right\|^2 \tag{1}$$

$$V^t = \beta V^(t-1) \tag{2}$$

$$\theta^t = \theta t - 1 - \eta V^t \tag{3}$$

Where 1 is the generalized loss function incorporating the individual loss weights and a regularization term with weight decay φ, 2 is the velocity term with momentum β, and 3 is the weight update rule and η is the

learning rate. The specific YOLOv8 loss function can be defined as:

$$\begin{aligned}\mathcal{L} = & \frac{\lambda_{box}}{N_{box}} \sum_{x,y} \mathbb{1}^*_{c_{x,y}} [1 - q_{x,y} + \frac{\left\| b_{x,y} - \hat{b}_{x,y} \right\|_2^2}{\rho} + \alpha_{x,y} \nu_{x,y}] \\ & + \frac{\lambda_{cls}}{N_{cls}} \sum_{x,y} \sum_{c \in classes} y_c \log \hat{y}_c + (1 - y_c) \log(1 - \hat{y}_c) \\ & + \frac{\lambda_{dfl}}{N_{dfl}} \sum_{x,y} \mathbb{1}^*_{c_{x,y}} \left[-(q_{(x,y)+1} - q_{x,y}) \log \hat{q}_{x,y} \right. \\ & \left. + (q_{x,y} - q_{(x,y)-1}) \log \left(\hat{q}_{(x,y)+1} \right) \right]\end{aligned}$$

where:

$$\boldsymbol{q_{x,y} = IoU_{x,y} = \frac{\hat{\beta}_{x,y} \cap \beta_{x,y}}{\hat{\beta}_{x,y} \cup \beta_{x,y}}}$$

$$\nu_{x,y} = \frac{4}{\pi^2} (arctan(\frac{w_{x,y}}{h_{x,y}}) - arctan(\frac{\hat{w}_{x,y}}{\hat{h}_{x,y}}))^2$$

$$\alpha_{x,y} = \frac{\nu}{1 - q_{x,y}}$$

$$\hat{y}_c = \sigma(\cdot)$$

$$\hat{q}_{x,y} = softmax(\cdot)$$

and:
− Npos is the total number of cells containing an object.
− is an indicator function for the cells containing an object.
− $\beta_{x,y}$ is a tuple that represents the ground truth bounding box consisting of (x_coord,y_coord, width, height).
− $\hat{\beta}_{x,y}$ is the respective cell's predicted box.
− $b_{x,y}$ is a tuple that represents the central point of the ground truth bounding box.
− y_c is the ground truth label for class c (not grid cell c) for each individual grid cell (x,y) in the input, regardless if an object is present.
− $q_{(x,y)} + / - 1$ are the nearest predicted boxes IoUs (left and right) $\in c^*_{x,y}$
− $w_{x,y}$ and $h_{x,y}$ are the respective boxes width and height.
− ρ is the diagonal length of the smallest enclosing box covering the predicted and ground truth boxes.
Each cell then determines its best candidate for predicting the bounding box of the object. This loss function includes the CIoU (complete IoU) loss proposed by Zheng et al. [9] as the box loss, the standard binary cross entropy for multi-label classification as the classification loss (allowing each cell to predict more than 1 class), and the distribution focal loss proposed by Li et al.[10] as the 3rd term.

4.2.3 Model diagnosis

Figure 2 shows the original CT scan image and the activation of the four c2f stages in the network, with each stage being more profound in the network from the second image right. The Activation Map corresponding to the shallowest c2f module shows the broadest activation. This module shows the abstraction of organs and determine what is these organs look like. The second activation map corresponds to the second c2f module in our backbone. It shows strong activation in the general shape of the organs. It appears that this layer is attempting to infer what type of each organ looks like in the image by highlighting these features. Finally, the model's final c2f module activates extremely fine-grained details and outlines in the respective images.

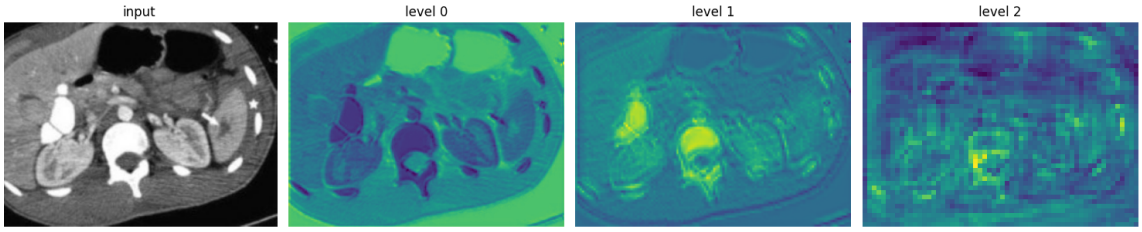


Figure 8: Feature activation maps for CT scan. From left to right, we have 3 features extracted from the model's CSPDarknet53 backbone

4.2.4 Detection result

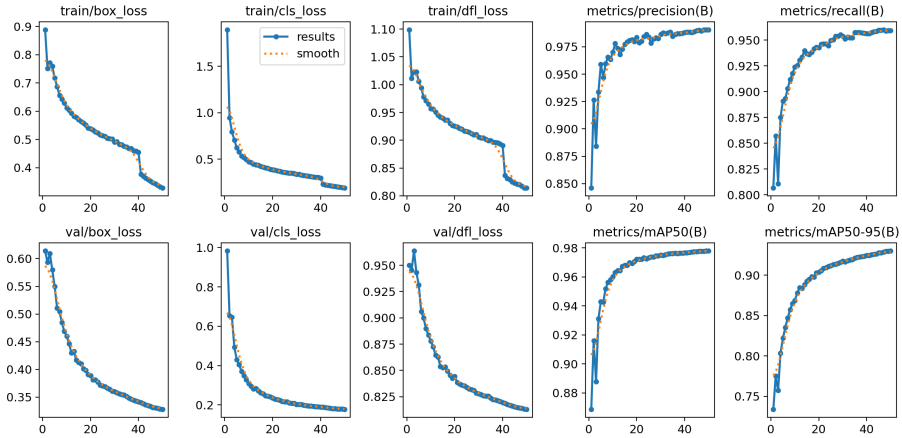
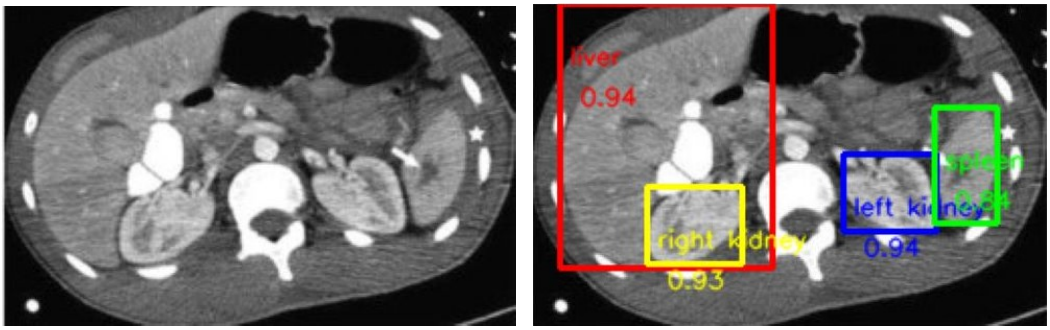
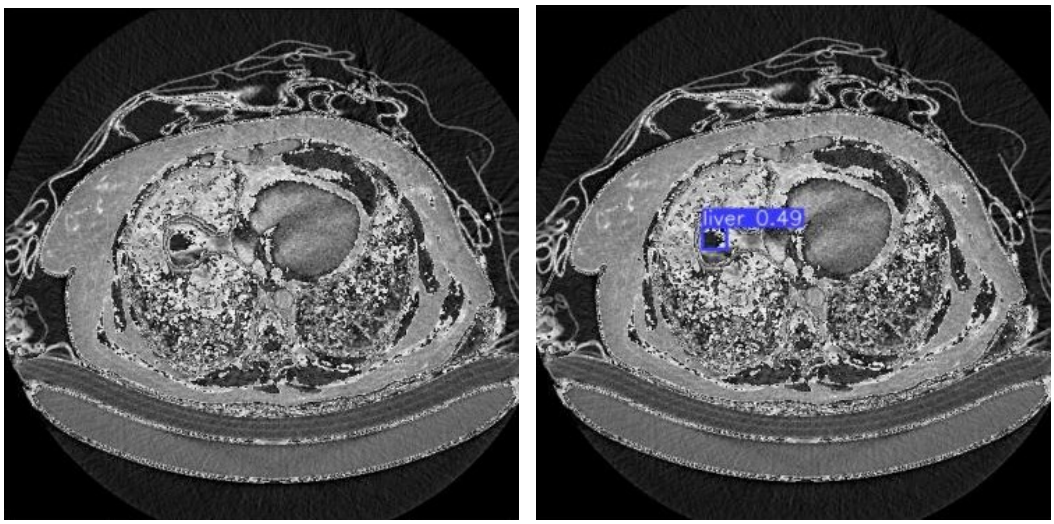


Figure 9: Training progress

We train yolov8 on 29148 images with a train/valid/test ratio of 64/16/20%. We choose the best hyperparameters based on validation mAP50-95 as batch size of 16, stochastic gradient descent (SGD) as the optimiser, momentum of 0.937, weight decay of 0.0005, classification loss weight λcls = 0.5, box loss weight λbox = 7.5, and distribution focal loss weight λdf l = 1.5. Because these organs' relative location is usually fixed (2 kidneys are always at the bottom, the spleen is lower than the liver), the location is highly correlated to its class, so we focus more on the area. From the 6 left figures of the training process, we can see that the model witnessed a slight overfit at around epoch 3, but it modified the learning rate instantly so that the loss could converge. The last 4 figures show the model's performance: precision, recall, mAP@50, and mAP@50-95. All these values are very high (over 95%), meaning the model is very good at localizing and classifying the organ.



(a) Original image with highly visible organs (b) Prediction

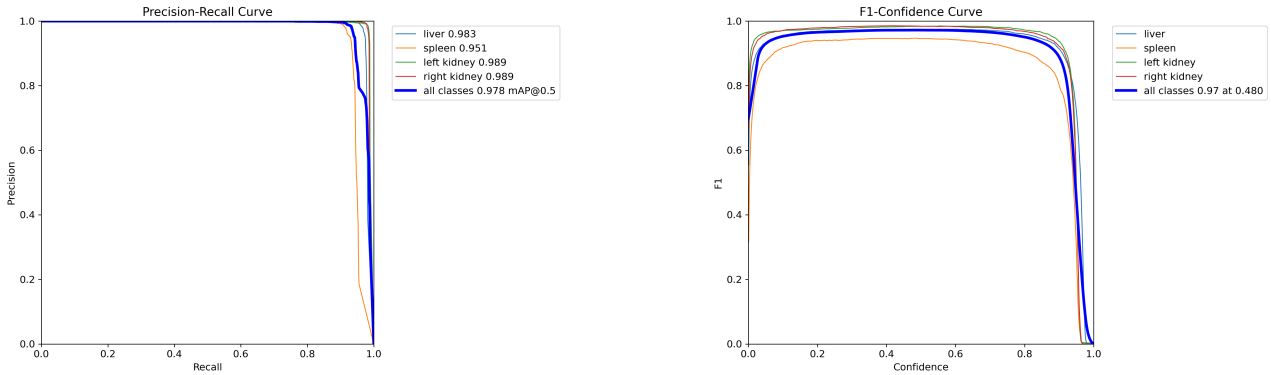


(c) Original image with highly occluded organs (d) Prediction

Figure 10: Output of detection

After training for 50 epochs, the model shows an accurate detection in Figure 9 with a high confidence score (over 80% confidence if the organ is visible to the human eye) and a low confidence score (under 50% confidence if the organ is highly occluded). This result shows that the model can give reliable detection for further work.

Here are 2 metrics of Yolov8 nano:



(a) Precision-Recall Curve (b) F1 Score Curve

Figure 11: Performance Curves

Figure 8a shows that the average AUC of the precision-recall curve for each class is very high (97.8% at IoU threshold=0.5), which means the model excels at localizing the organ and classifying it. Figure 8b shows a very high F1 score: 0.97 at IoU threshold 0.48; however, the F1 confidence curve is almost flat between the 0.2 and 0.8 thresholds. This means we could set a higher threshold for detection and still receive a good trade-off between precision and recall. In conclusion, even though the nano version of yolov8 contains only 3 million parameters and uses less computational cost (only 8.1 GFLOPs), it still handles the detection phase very well, making it easier for injury-level classification later.

4.3 Classification Phase

In this phase, we train models to classify the injury level on each image. We split the data into train, validation and test set with ratio (70, 20, 10). We choose hyperparameters: 20 epochs, optimizer SGD, initial learning rate 0.1, and ReduceOnPlateau scheduler, which reduces the learning rate 10 times when the validation accuracy doesn't increase after 2 epochs.

4.3.1 Classification model

4.3.1.1 ResNet50

We use Resnet50[11] to extract features from images. ResNet-50 is a deep convolutional neural network architecture introduced by Microsoft Research in 2015. It is a variant of the ResNet (Residual Network) model known for effectively training deep networks. The "50" in ResNet-50 refers to the number of layers in the network, including convolutional layers, pooling layers, fully connected layers, and shortcut connections. ResNet-50 comprises a series of blocks containing multiple convolutional layers followed by a shortcut connection. The shortcut connections allow for the propagation of gradients through the network, addressing the vanishing gradient problem commonly encountered in deep networks.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 12: Architectures for ResNet

Building blocks are shown in brackets, with the number of blocks stacked. Downsampling is performed by conv3 1, conv4 1, and conv5 1 with a stride of 2 ResNet-50 has been pre-trained on a large-scale image dataset, such as ImageNet, which enables it to learn valuable features from images. These known features can be leveraged for various computer vision tasks, including image classification, object detection, and image segmentation. The architecture of ResNet-50 has been widely adopted and serves as a benchmark for many computer vision tasks. It has achieved state-of-the-art performance on tasks such as achieving an impressive top-5 error rate (5.25%) on the ImageNet dataset, a large-scale dataset with millions of labelled images.

4.3.1.2 EfficientNetB2

EfficientNet-B2[12], an advanced convolutional neural network architecture, has demonstrated exceptional performance on the ImageNet dataset, a widely recognized benchmark for image classification. This achievement is attributed to the incorporation of powerful building blocks, including the MBConv (Mobile Inverted Residual Bottleneck Convolution) and Squeeze-and-Excitation (SE) blocks.

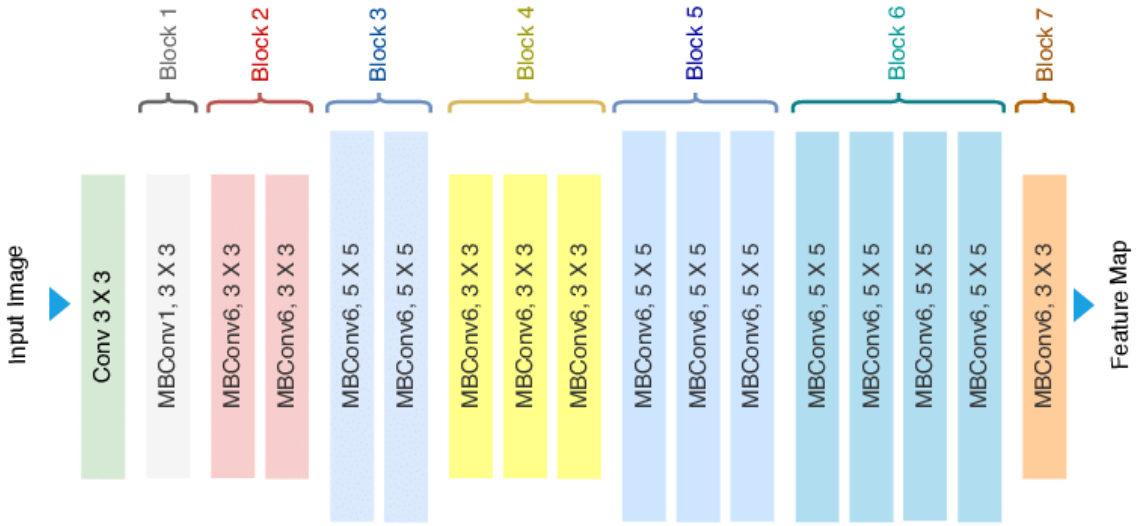


Figure 13: MBConv blocks

The MBConv blocks, inspered by the inverted residual structure from MobileNetV2, optimize the trade-off between computational efficiency and accuracy. By employing depthwise separable convolutions and expansion phases, MBConv effectively reduces computational complexity while capturing complex features.

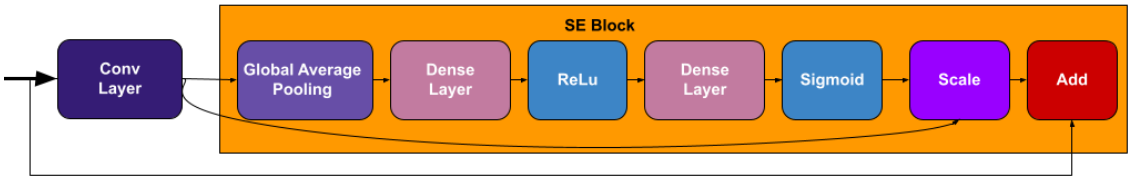


Figure 14: SE block

Additionally, EfficientNet-B2 leverages SE blocks to learn and adaptively recalibrate channel-wise feature dependencies. This selective amplification of informative features and suppression of less relevant ones enhances the network's representation power. The combined strength of MBConv and SE blocks enables EfficientNet-B2 to achieve top-tier performance on the ImageNet dataset, with top-1 accuracy around 80% and top-5 accuracy exceeding 95%. These impressive results showcase the effectiveness of EfficientNet-B2 in accurately classifying a diverse range of images and establish its superiority in handling complex visual tasks.

4.3.2 Classification head

After going through the backbone, we pass the result through 5 fully connected layers. The first 2 layers will reduce the feature from 1000 to 512 and 512 to 256 using SiLU activation. The next 2 layers keep reducing the feature by half and using ReLU activation. The final layer reduces the feature to 3 outputs corresponding to 3 classes(healthy, low, and high).

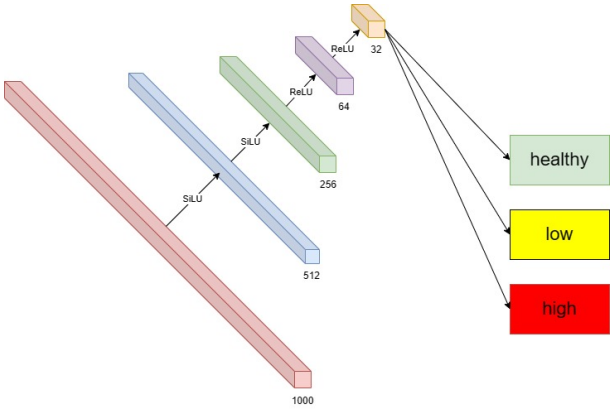


Figure 15: Classification head

4.3.3 Weighted vote

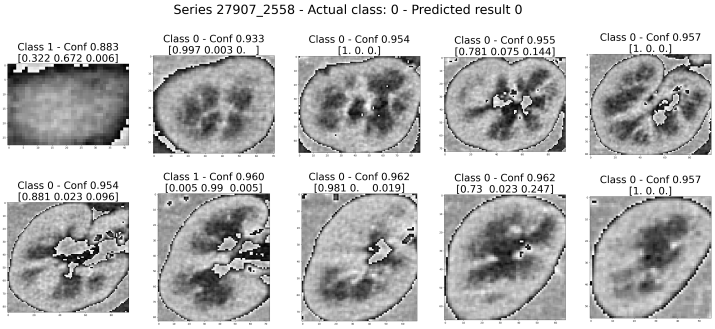


Figure 16: Weighted vote

In the given task, we aim to classify a series based on its level of injury. To accomplish this, we follow a step-by-step process. First, we obtain the probabilities of the image belonging to three categories: healthy, low injury level, and high injury level. These probabilities assess the severity of the injury present in the image.

Next, we multiply each probability with the detection confidence associated with that particular image. This step allows us to weigh the contribution of each image's probability based on the confidence level of the detection algorithm. Afterward, we sum up all the probabilities for each category: healthy, low injury level, and high injury level. This aggregation step provides a cumulative score for each class, considering the probabilities and their associated detection confidences across the entire series of images.

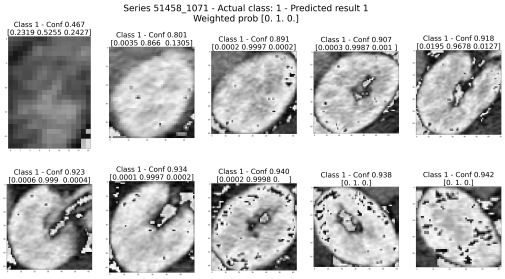
Finally, we determine the class with the highest score based on the summed probabilities. The class with the highest score represents the classification decision for the series of images, indicating the most probable level of injury present. In summary, by calculating and weighing the probabilities of each image and then aggregating the results, we can effectively classify the entire series of images and determine the predominant level of injury.

5 Results and Conclusion

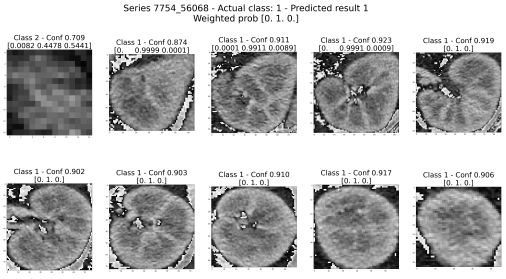
5.1 Result

5.1.1 Example output

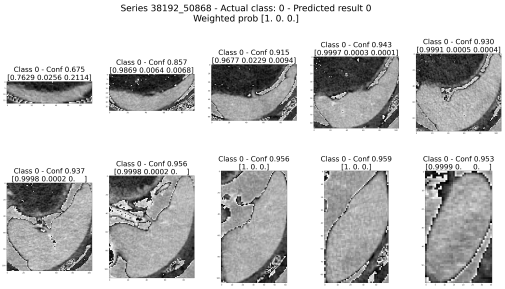
- True predictions:



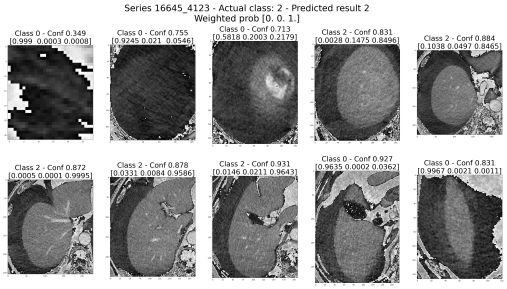
(a) Success right kidney prediction



(b) Success left kidney prediction



(c) Success spleen prediction



(d) Success liver prediction

Figure 17: True prediction

When organ abnormalities are clearly visible in a series of medical images, the predictive model frequently exhibits precise identification of the associated injury level. Moreover, the model assigns higher confidence to its detections when the abnormalities are more distinctive and easier to evaluate. This prioritization of high confidence detections can enhance the reliability of assessing the observed abnormalities. Importantly, the model provides its confidence level along with the results, which can serve as supplementary information to assist healthcare professionals in their decision-making and diagnosis. Ultimately, the final interpretation and diagnosis should be made by qualified doctors, taking into account both the model's predictions and their own expertise.

- False predictions:

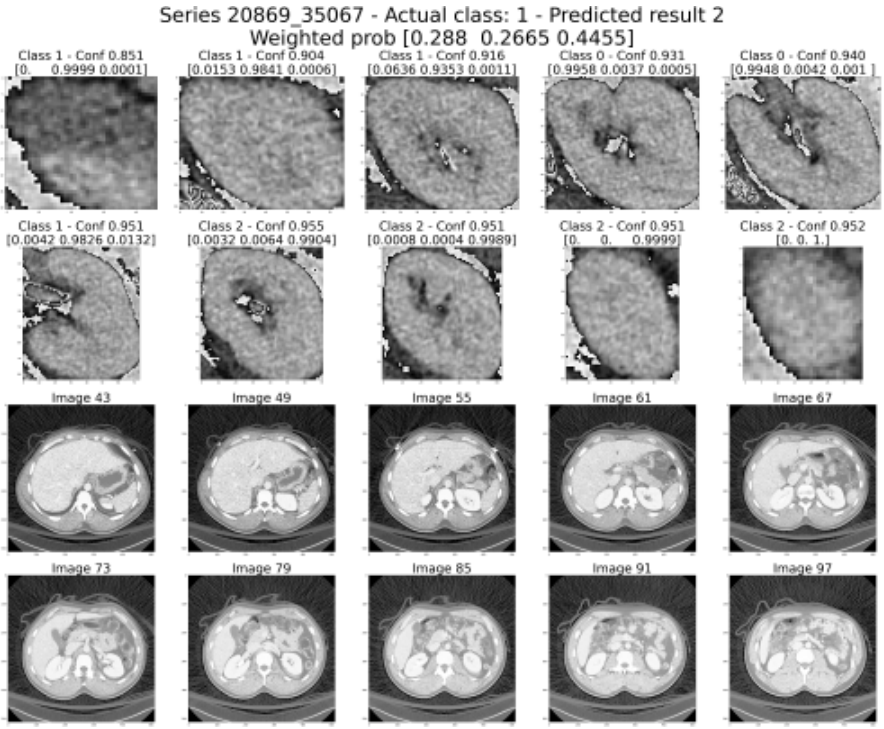


Figure 18: False left kidney prediction (First two rows are the detected organ, last two rows are the original images)

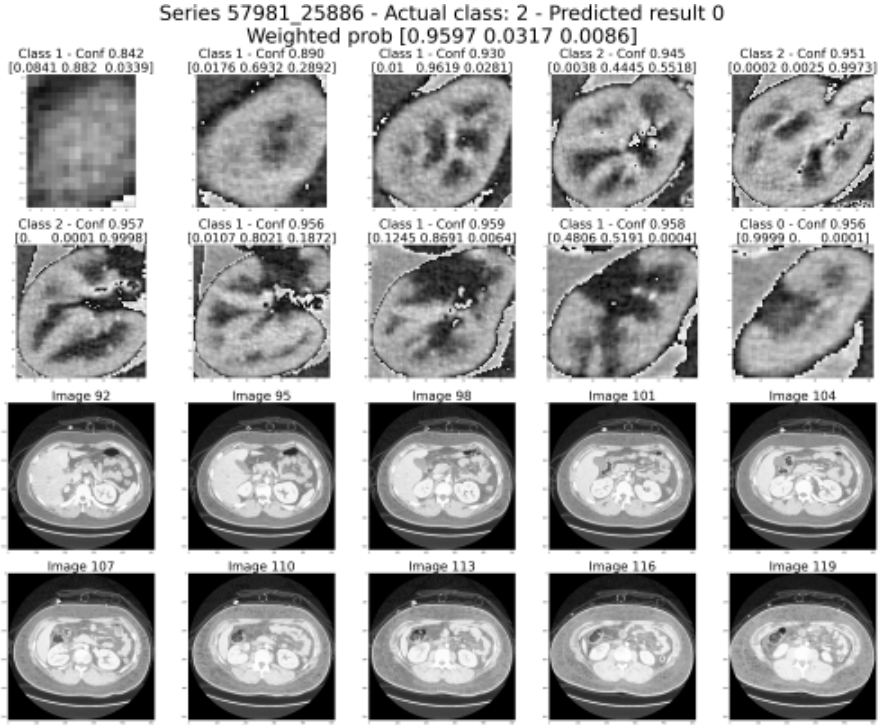


Figure 19: False right kidney prediction (First two rows are the detected organ, last two rows are the original images)

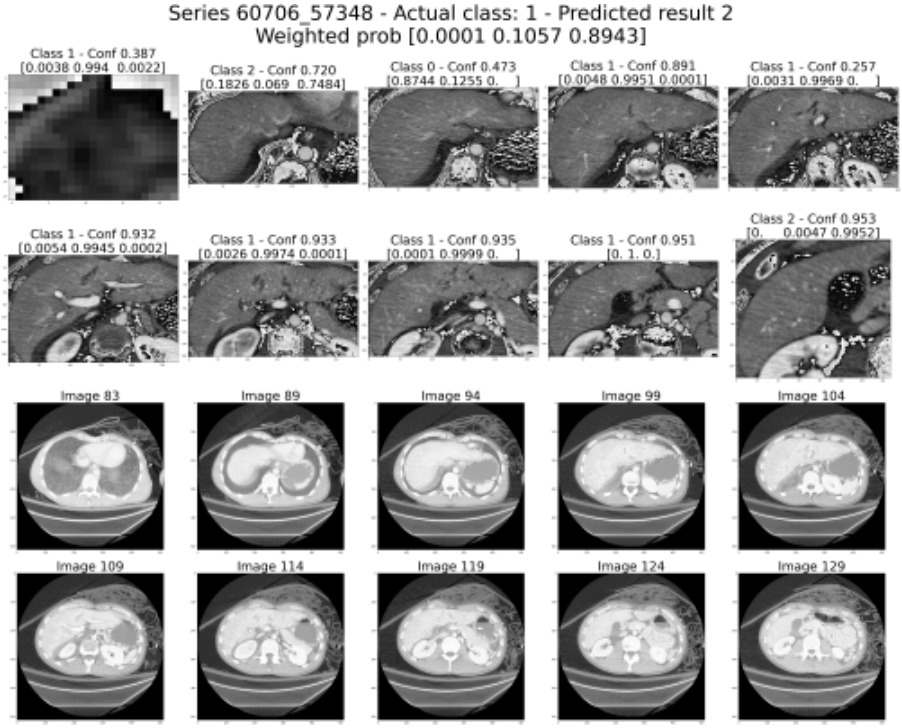


Figure 20: False liver prediction (First two rows are the detected organ, last two rows are the original images)

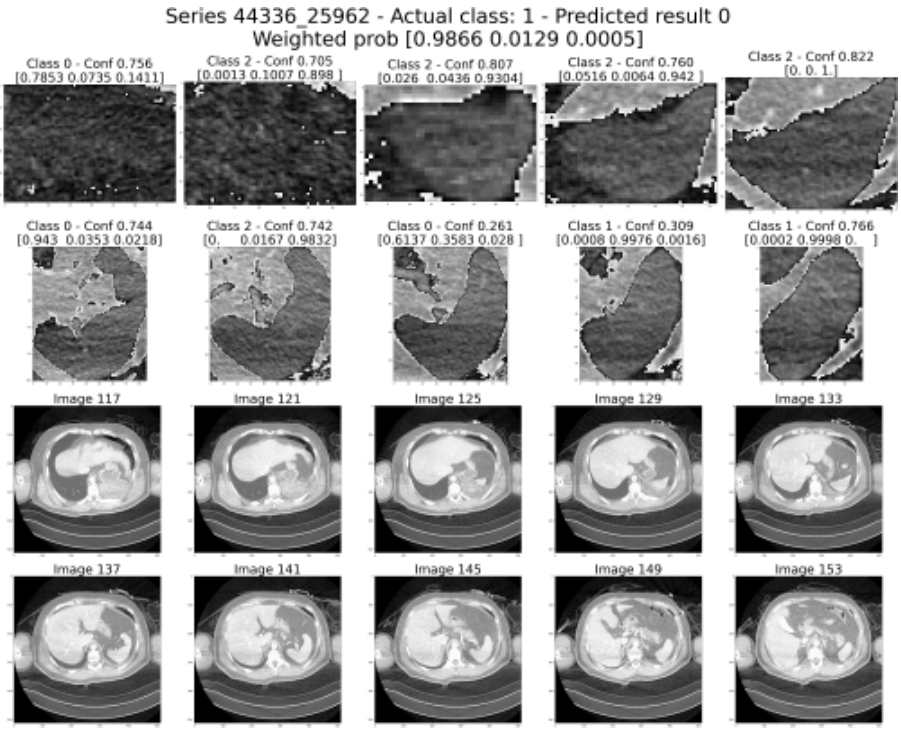


Figure 21: False spleen prediction (First two rows are the detected organ, last two rows are the original images)

False predictions may occur in cases where organs exhibit high confidence scores for detection, but their shapes deviate slightly from the normal shape or contain artifacts. These deviations can lead the model to misclassify the injury level. It is crucial to recognize that while the model's performance is generally reliable, it can be sensitive to variations in organ appearance and potential imaging artifacts. Therefore, it is important for healthcare professionals to carefully review and interpret the results, considering both the model's predictions and their own expertise, in order to make accurate diagnoses and treatment decisions.

- Miss detection:

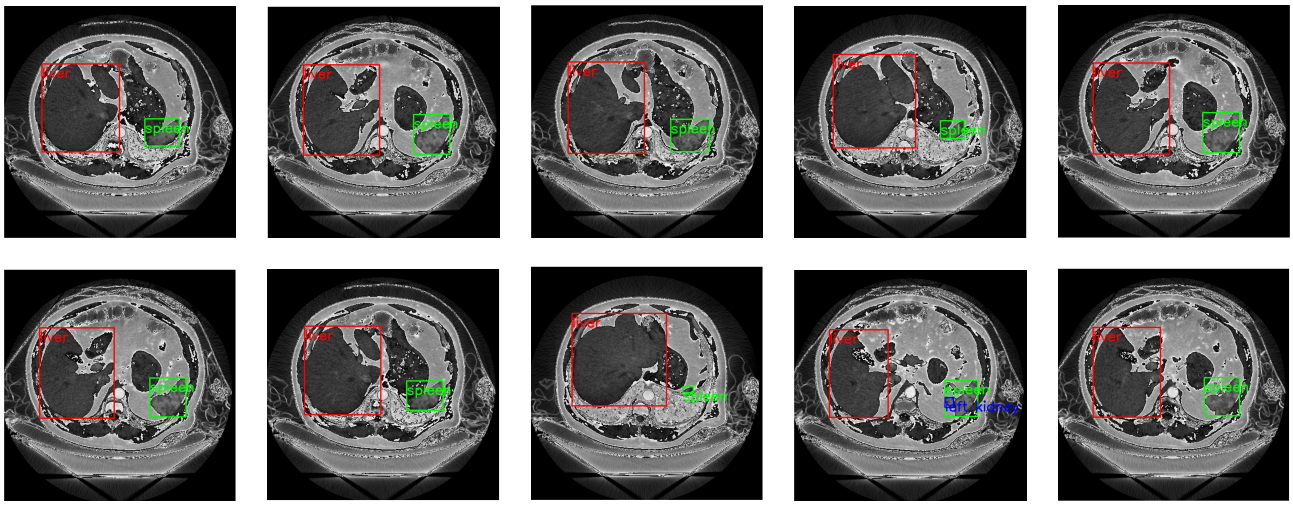


Figure 22: Miss detection (no right kidney)

Missed detections can occur when an organ is completely occluded in the entire series of medical images. In such cases, where the organ is not visible due to obstruction or other factors, the predictive model may fail to detect abnormalities or accurately classify the injury level. It is important to note that the model's performance relies on the availability and quality of the imaging data. In instances where an organ is entirely occluded, additional imaging modalities or techniques may be necessary to obtain a complete and accurate assessment. Healthcare professionals should exercise caution when interpreting results and consider alternative diagnostic approaches to ensure comprehensive evaluation of the patient's condition.

5.1.2 Model training time and complexity

Backbone Model	No. Parameters	Organ	Inference Time (s)
EfficientNetB2	9.1M	Left kidney	0.442
		Right kidney	0.442
		Liver	0.764
		Spleen	0.420
ResNet50	25.5M	Left kidney	0.302
		Right kidney	0.325
		Liver	0.597
		Spleen	0.321

Table 1: Average inference time for each organ over 1 series.

5.1.3 Performance metrics

- **Accuracy:** The proportion of correct predictions out of the total number of predictions.
- **Precision:** The proportion of true positives among the predicted positives.
- **Recall:** The proportion of true positives among the actual positives.
- **Specificity:** The proportion of true negatives among the actual negatives.
- **F1-score:** The harmonic mean of precision and recall, ranging from 0 to 1.

The results are summarized in the tables 2, 3, 4, 5 after training for 20 epochs.

Left kidney	EfficientNet			ResNet		
Injured level	Healthy	Low	High	Healthy	Low	High
Accuracy	84.62%	86.36%	87.50%	92.31%	90.91%	87.50%
Precision	91.67%	90.48%	70.00%	100%	90.91%	77.78%
Recall	84.62%	86.36%	87.50%	92.31%	90.91%	87.50%
Specificity	96.67%	90.48%	91.43%	100%	90.48%	94.29%
F1-score	88.00%	88.37%	77.78%	96.00%	90.91%	82.35%

Table 2: Performance comparison of EfficientNet and ResNet models for left kidney injury detection.

Right kidney	EfficientNet			ResNet		
Injured level	Healthy	Low	High	Healthy	Low	High
Accuracy	92.31%	100%	70.00%	100%	100%	70.00%
Precision	85.71%	90.91%	85.71%	81.25%	94.74%	77.78%
Recall	92.31%	100%	60.00%	100%	90.00%	70.00%
Specificity	93.33%	91.30%	96.97%	90.00%	95.65%	96.97%
F1-score	88.89%	95.24%	70.59%	89.66%	92.31%	77.78%

Table 3: Performance comparison of EfficientNet and ResNet models for right kidney injury detection.

Spleen	EfficientNet			ResNet		
Injured level	Healthy	Low	High	Healthy	Low	High
Accuracy	89.66%	89.66%	100%	89.66%	86.21%	91.30%
Precision	92.86%	89.66%	95.83%	92.86%	86.21%	91.30%
Recall	89.66%	89.66%	100%	89.66%	86.21%	91.30%
Specificity	96.15%	94.23%	98.28%	96.15%	90.38%	96.55%
F1-score	91.23%	89.66%	97.87%	91.23%	84.75%	91.30%

Table 4: Performance comparison of EfficientNet and ResNet models for spleen injury detection.

Liver	EfficientNet			ResNet		
Injured level	Healthy	Low	High	Healthy	Low	High
Accuracy	68.75%	95.56%	100%	81.25%	95.56%	72.73%
Precision	100%	89.58%	84.62%	86.67%	87.76%	100%
Recall	68.75%	95.56%	100%	81.25%	95.56%	72.73%
Specificity	100%	81.48%	96.72%	96.43%	77.78%	100%
F1-score	81.48%	92.47%	91.67%	83.87%	91.49%	84.21%

Table 5: Performance comparison of EfficientNet and ResNet models for liver injury detection.

In addition to the accuracy, precision, recall, and F1-score reported in the tables above, we also evaluated the performance of our model using:

- Confusion matrix
- Precision-recall curve: Shows the trade-off between precision and recall for the model, with higher values indicating better performance.
- ROC curve: Shows the trade-off between the true positive rate (TPR) and the false positive rate (FPR) for the model, with higher values indicating better performance.

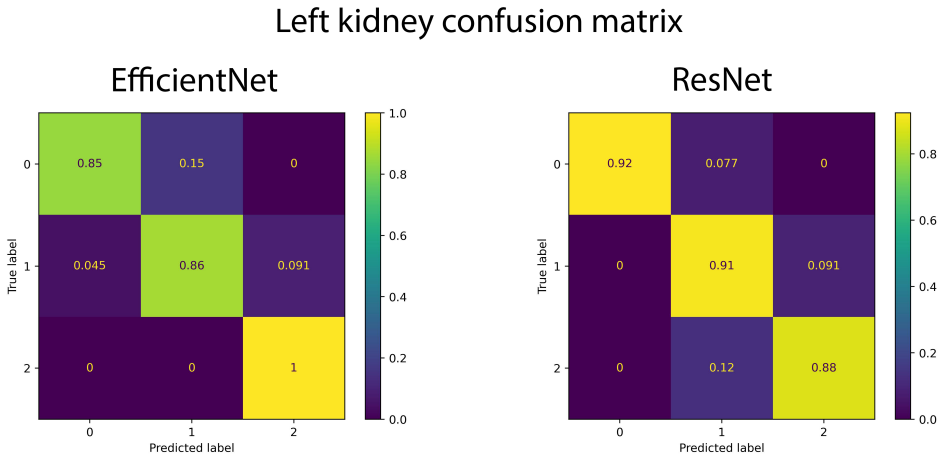


Figure 23: Left kidney confusion matrix

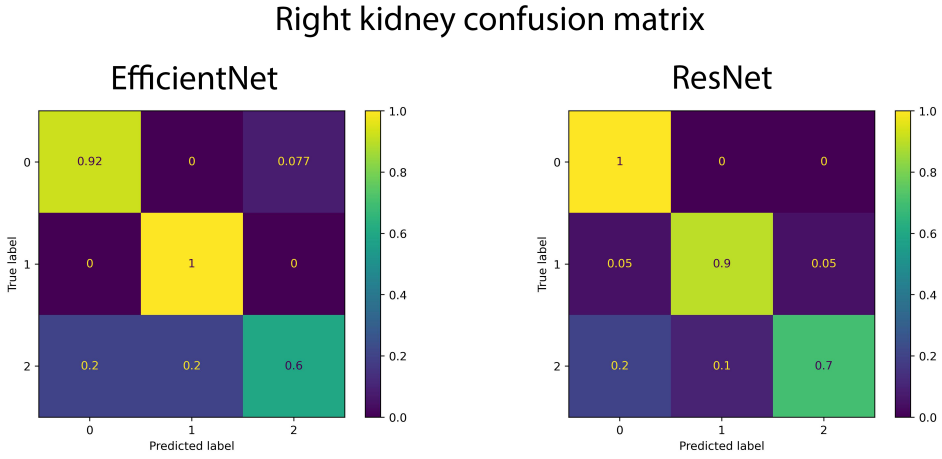


Figure 24: Right kidney confusion matrix

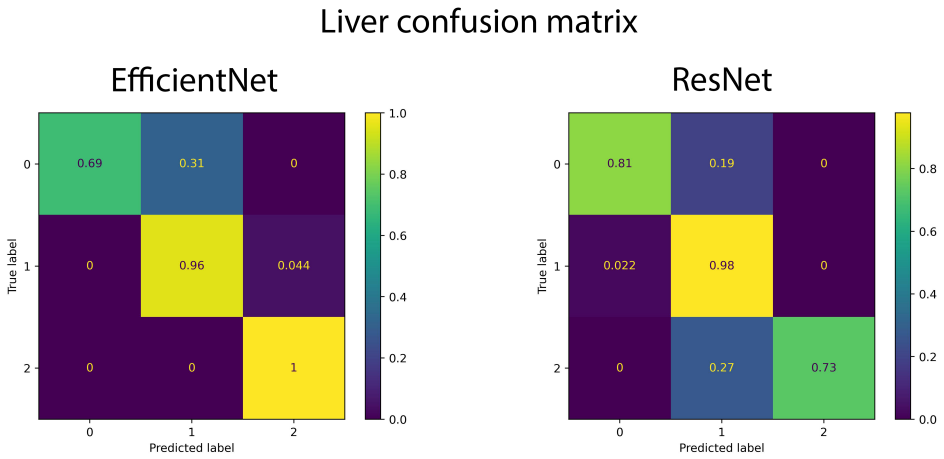


Figure 25: Liver confusion matrix

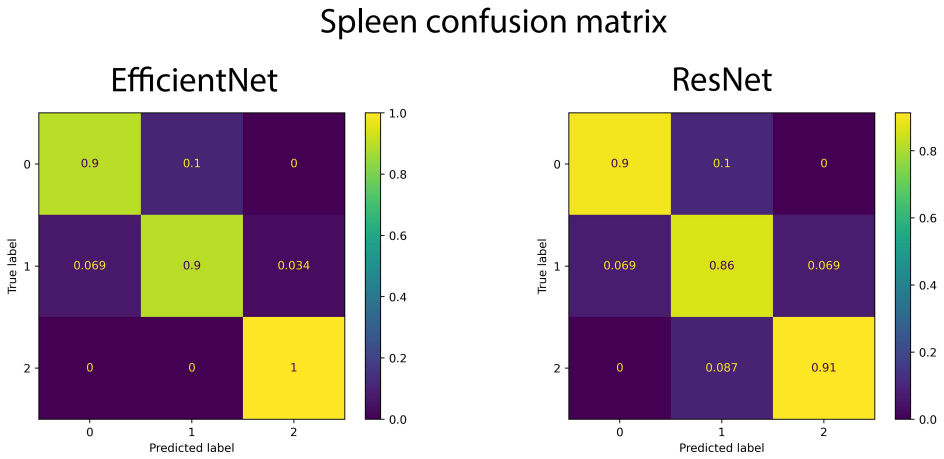


Figure 26: Spleen confusion matrix

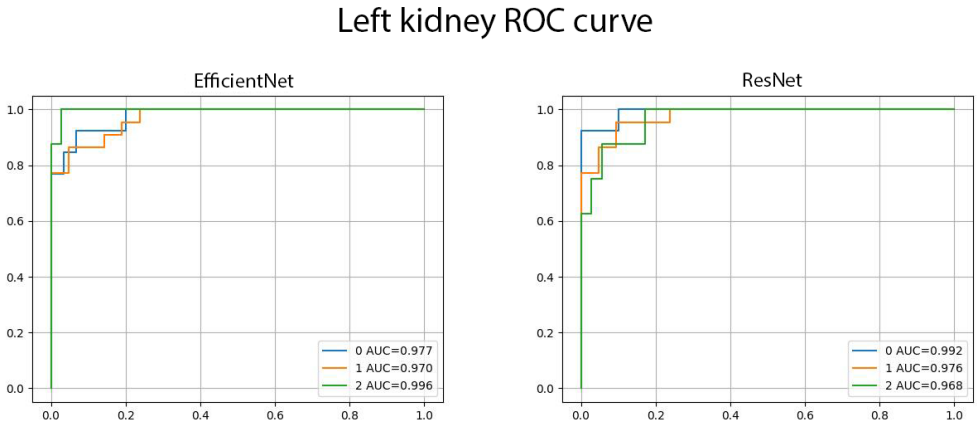


Figure 27: Left kidney ROC curve

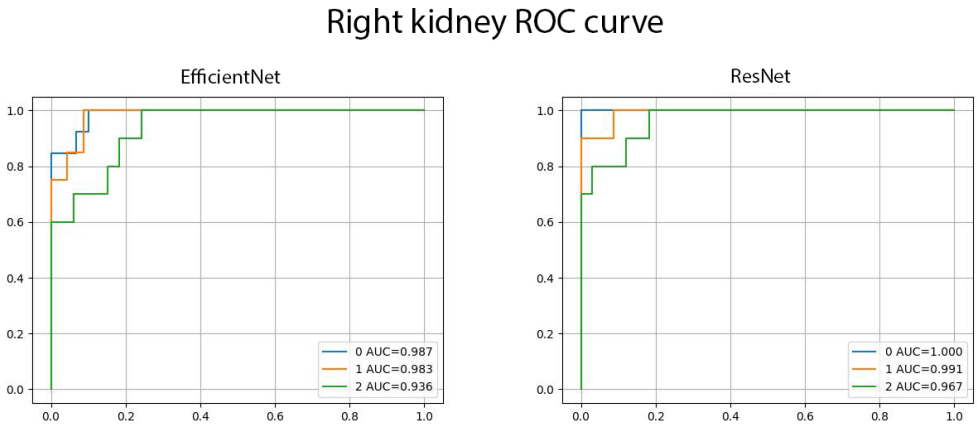


Figure 28: Right kidney ROC curve

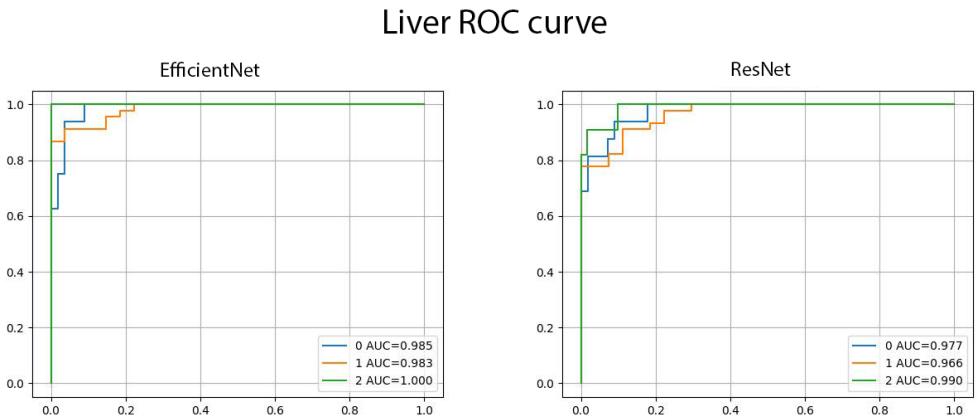


Figure 29: Right kidney ROC curve

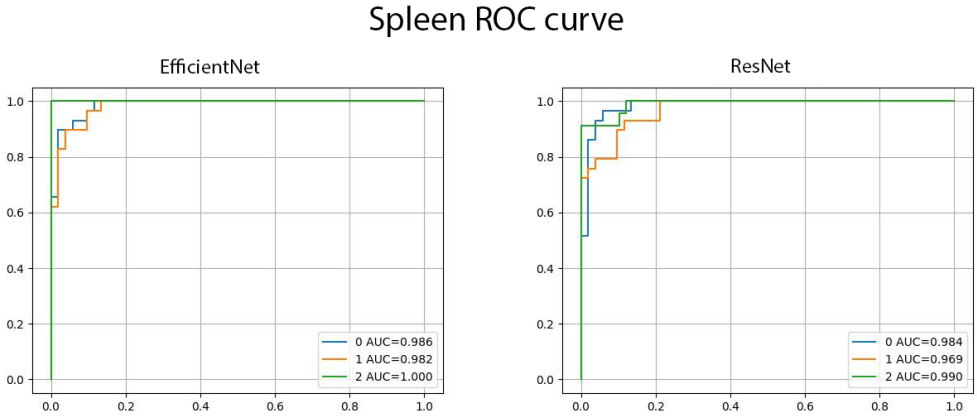


Figure 30: Right kidney ROC curve

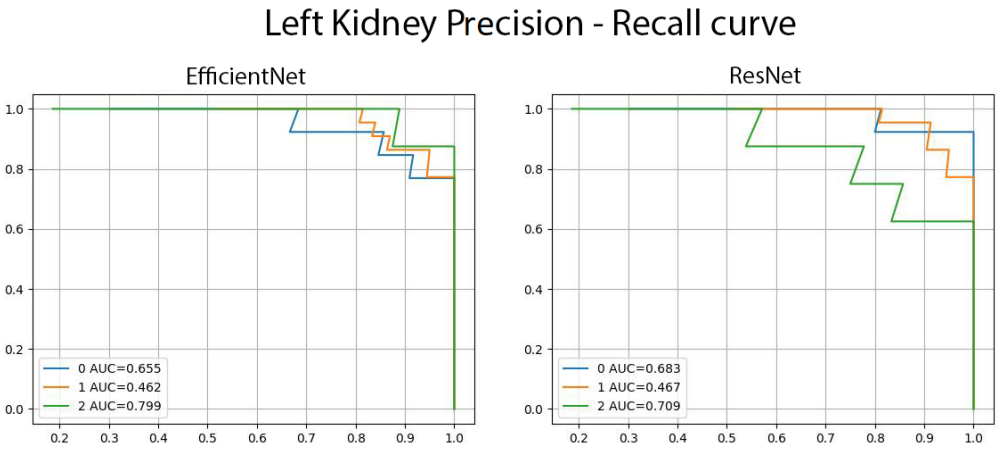


Figure 31: Left kidney PRC curve

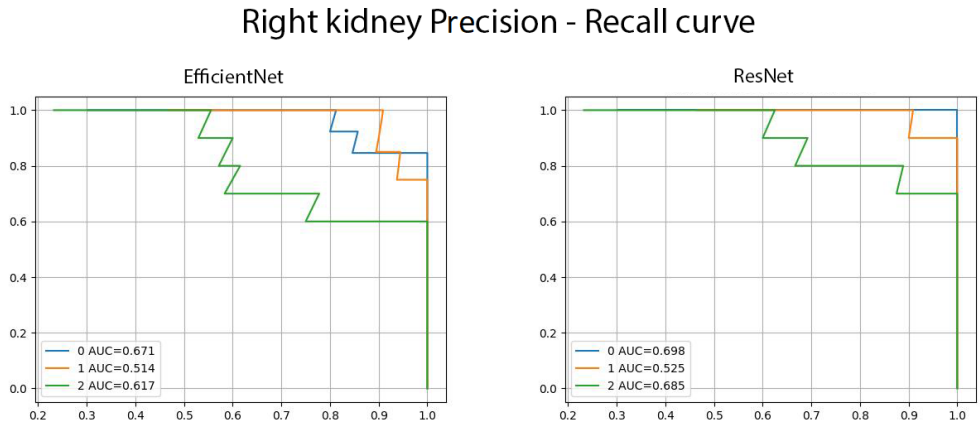


Figure 32: Right kidney PRC curve

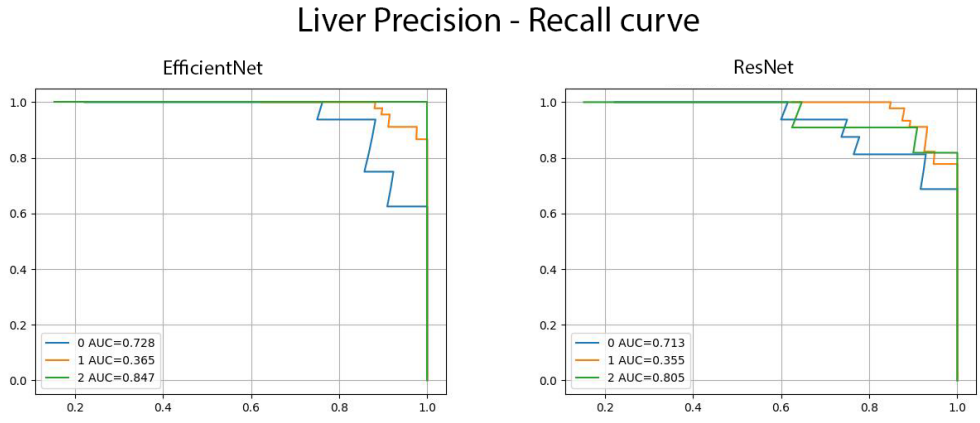


Figure 33: Right kidney PRC curve

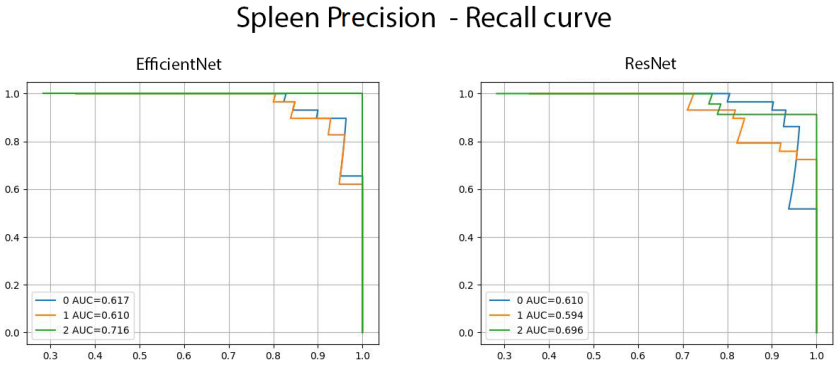


Figure 34: Right kidney PRC curve

5.2 Conclusion

In this study, a deep learning-based method was proposed to recognize the level of abdominal trauma using CT scans. The accuracy of the method was evaluated using a dataset of over 400 series of CT scans. The results showed that the proposed method could automatically determine the trauma level of each organ within approximately one second using just one series of scans.

The performance of the model was assessed using two evaluation metrics: the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve and the AUC of the Precision-Recall (PR) curve. The AUC of the ROC curve was found to be very high, exceeding 0.9, indicating that the model had a strong ability to distinguish between positive and negative cases across various thresholds. However, the AUC of the PR curve was acceptable but lower, around 0.6, suggesting that achieving high precision and recall simultaneously was challenging for the model. This indicates that the model may struggle with correctly identifying positive instances while maintaining a low false positive rate.

It is worth noting that in certain situations, a high false positive rate can still be acceptable or even beneficial, such as in early detection or risk stratification scenarios. However, it is crucial to carefully consider the specific context and potential consequences of false positives.

Compared to a previous slice-attention based approach [4], our method interprets the "key slice" and "adjacent slice" differently. We determine the "key slice" based on detection confidence, and the "adjacent slice" using weighted voting. Our method achieves comparable outcomes to the previous approach while offering a more streamlined pipeline and reduced training time.

Overall, the proposed method in this study provides a rapid and efficient pipeline for recognizing the level of abdominal trauma using CT scans.

6 Future work

The presented results for abdominal trauma recognition, including the left kidney, right kidney, liver, and spleen, show promising outcomes. Future work should focus on several key areas to propel the field of medical image analysis forward and enhance clinical applicability.

Fine-tuning the existing models or considering more advanced architectures, possibly through transfer learning with pre-trained models on larger medical image datasets, presents an opportunity to achieve heightened accuracy and generalization. Ensemble methods, combining predictions from both EfficientNet and ResNet models, may also be explored to bolster overall predictive performance. Addressing class imbalance issues, especially in the presence of fewer samples for certain classes, should be a focus. Techniques like oversampling, undersampling, or the application of weighted loss functions during training could be employed to ensure fair representation of all classes.

Systematic hyperparameter tuning, encompassing adjustments to learning rates, batch sizes, and other relevant parameters, may lead to the identification of optimal configurations. Furthermore, incorporating model interpretability techniques will aid in understanding and visualizing decision-making processes, particularly important in critical medical applications. Optimizing models for reduced inference time ensures practical usability in clinical settings. Establishing continuous monitoring and updating mechanisms based on new data and evolving medical standards is vital for sustained accuracy and reliability. In summary, future research in medical image analysis should embrace advanced architectures, data augmentation, interpretability, and collaboration with healthcare professionals to create robust, efficient, and clinically applicable models, ultimately improving patient care in diverse medical scenarios.

References

[1] GBD Chronic Kidney Disease Collaboration. (2020). Global, regional, and national burden of chronic kidney disease, 1990-2017: a systematic analysis for the Global Burden of Disease Study 2017. *Lancet (London, England)*, 395(10225), 709-733. doi:10.1016/S0140-6736(20)30045-3

[2] SEER (Surveillance, Epidemiology, and End Results) database. (n.d.). *Liver and Intrahepatic Bile Duct Cancer — Cancer Stat Facts*. Retrieved [Insert Date], from `https://seer.cancer.gov/statfacts/html/livibd.html`

[3] Chen, Y., Qiu, J., Yang, A., Yuan, D., & Zhou, J. (2017). Epidemiology and management of splenic injury: An analysis of a Chinese military registry. *Experimental and therapeutic medicine*, 13(5), 2102-2108. doi:10.3892/etm.2017.4208

[4] Fu, G., Li, J., Wang, R., Ma, Y., & Chen, Y. (2021). Attention-based full slice brain CT image diagnosis with explanations. *Neurocomputing*, 452, 263-274. doi:10.1016/j.neucom.2021.04.044

[5] Signate Yamaguchi. Mmyolo visualization, 2022. `https://github.com/open-mmlab`

[6] Solawetz, J. (2023, Dec). What is Yolov8? the ultimate guide. *Roboflow Blog*. Retrieved from `https://blog.roboflow.com/whats-new-in-yolov8/`

[7] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *ArXiv*, abs/1804.02767. Retrieved from `https://api.semanticscholar.org/CorpusID:4714433`

[8] Terven, J. R., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*. doi:10.3390/make5010007

[9] Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 34, No. 07, pp. 12993-13000). Retrieved from `https://doi.org/10.1609/aaai.v34i07.6999`

[10] Li, X., Wang, W., Wu, L., Chen, S., Hu, X., Li, J., Tang, J., & Yang, J. (2020). Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. *ArXiv*, abs/2006.04388. Retrieved from `https://api.semanticscholar.org/CorpusID:219531292`

[11] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770-778). doi:10.1109/CVPR.2016.90

[12] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv*, abs/1905.11946. Retrieved from `https://api.semanticscholar.org/CorpusID:167217261`