Automatic Spondylolisthesis Grading from MRIs across modalities using Faster Adversarial Recognition Network

Shen Zhao[a], Xi Wu[b,], Bo Chen[a], Shuo Li[a,]

[a] *University of Western Ontario, London ON, Canada*
[b] *Department of Computer Science, Chengdu University of Information Technology, Chengdu, 610225, China*

Abstract

Grading spondylolisthesis into several stages from MRI images is challenging because detecting critical vertebrae and locating landmarks in images of different characteristics is difficult. We propose Faster Adversarial Recognition (FAR) network to accurately perform spondylolisthesis grading by excellently detecting critical vertebrae without the need of locating the landmarks. The FAR network introduces the adversarial scheme by using a multi-task recognition network as the generator and an adversarial module as the discriminator. The multi-task recognition network (generator) is an integrated network that can reliably perform multi-scale hierarchical feature learning, critical vertebrae detection, detected vertebrae classification, bounding box regression, and spondylolisthesis grading in a hybrid supervised manner. The adversarial module (discriminator) takes the detection results as inputs to supervise the generative network by leveraging the high-order statistics of the distribution of the detected bounding box coordinates. The FAR network is evaluated to be accurate and robust in spondylolisthesis grading (training accuracy: 0.9883±0.0094, testing accuracy: 0.8933±0.0276) for MRI images of different modalities, which can be attributed to the excellent critical vertebrae detection (detection mAP_{75} for training: 1±0, for testing: 0.9636±0.0180, and IoU (Intersection-over-union) $\geqslant$0.9/0.8 for most detections with their corresponding ground truth in the training/testing dataset). This accuracy is comparable to that of the physicians and outperforms other state-of-the-art methods. These results indicate the potential of our framework to perform spondylolisthesis grading for clinical diagnosis.

Keywords: Spondylolisthesis Grading, object detection, GAN (generative adversarial network), MRIs across modalities

*Corresponding author
Email addresses: xi.wu@cuit.edu.cn (Xi Wu), slishuo@gmail.com (Shuo Li)

1. Introduction

Spondylolisthesis is defined as the forward displacement of vertebrae, which means the vertebral bones may progressively deform and press on corresponding nerves. Spondylolisthesis and its resulting symptoms cause low back pains (Hartvigsen et al., 2018), sciatica, neurologic compromise (Hresko et al., 2007), and even life-long functional disability for basic activities of daily life (such as dressing and outdoor walking) worldwide (Jamaludin et al., 2017; Möller et al., 2000). Early diagnosis and treatments are important for preventing spondylolisthesis progress and healing spondylolisthesis.

For clinical diagnosis of spondylolisthesis, there are 5 grades indicating the severity of spondylolisthesis (Wollowick and Sarwahi, 2015). The higher is the grading, the more severe is the disease. As shown in Fig.1, spondylolisthesis grading is measured by the ratio of the forward displacement (the length of the orange arrow in Fig.1(a)) to the total length of the vertebrae (the length of the yellow line with tick marks in Fig.1(a)). Spondylolisthesis grading 1∼5 respectively means the ratio falls into the interval 0%∼25%, 25%∼50%, 50%∼75%, 75%∼100%, higher than 100%; meanwhile, we denote grade 0 as patients who do not have spondylolisthesis (the ratio of the forward displacement is 0% or very close to 0%). For a visual demonstration, we label the critical points of adjacent gradings (the tick marks on the yellow line in Fig.1(c)) and show different gradings using different colors. Namely, if the red point (the end point of the orange arrow) falls into the blue (similarly, green/orange/red) area, the grading will be 1 (similarly, 2/3/4).

Automatically distinguishing spondylolisthesis patients with grading 0, 1, and 2 are the most important in clinical practice because: (1) As mentioned in (Passias et al., 2015), the overwhelming majority of patients (more than 99%) have spondylolisthesis grading less than 2; among them, a lot of patients do not have spondylolisthesis (i.e., they are "grade 0"). As a comparison, very few (∼1%) patients have spondylolisthesis grading higher than 3. (2) No treatments are needed for patients with grade 0, different treatments are applied to patients with grading 1∼2, while similar treatments are used for patients with higher grades. In more details, patients with grading 1 need only bed rest and avoiding activities that may further injury; patients with grading 2 need extra exercises and stretches for spondylolisthesis, while high-grade patients (higher than 3) typically need surgical treatments (Hresko et al., 2007). Thus, performing 0∼2 grading is the most important from the aspect of clinical diagnosis and making treatment plans.

Automatic spondylolisthesis grading is a crucial step for spondylolisthesis diagnosis because it eliminates the tedious and irreproducible manual grading procedure. Manual spondylolisthesis grading can be achieved by the Meyerding grading system (Niggemann et al., 2012), where the forward displacement and the total length of the vertebrae both need to be manually determined. This involves manually locating the landmarks, i.e., the superior surfaces of a vertebra (the yellow line with tick marks in Fig.1(c)) and the posterior endpoint of the inferior surface of its superior vertebra (the red point in Fig.1(c)), drawing

perpendicular lines (the orange dashed line), and measuring the distances (the orange arrow). These procedures are prone to subjective errors of the observers. Thus, it is crucial to develop an automatic computer-assisted spondylolisthesis grading system to obtain accurate and robust gradings (Liao et al., 2016).

However, automatic spondylolisthesis grading is challenging because (1) Spondylolisthesis grading requires critical vertebrae (L4, L5, and S1 vertebra, between which spondylolisthesis usually occurs (Wollowick and Sarwahi, 2015)) to be detected from raw medical images (Liao et al., 2016). This is error-prone because these critical vertebrae share similar appearance and anatomies with the other vertebrae of one patient; while the same vertebrae of different patients may look different because of additional spine diseases (Han et al., 2018), as shown in Fig.1(a) and (d). If the detection results are wrong, namely, there are false positive errors (for example, more than one L4 is detected, as shown in the middle figure in Fig.1(b)) or false negative errors (for example, no L4 is detected, as shown in the right figure in Fig.1(b)), it is very likely that the grading results would be wrong. (2) Spondylolisthesis grading requires accurate localization of the above-mentioned landmarks. However, the shapes of vertebrae are irregular (as shown in Fig.1(a)), and the locations of the landmarks tend to vary with different observers. According to (Liao et al., 2016), even slight deviations of landmark positions could result in large forward translation estimation error and wrong grading decision, as shown in Fig.1(c). (3) Spondylolisthesis grading based on MRI imaging is even more challenging because MRI images can be of different modalities. MRI images in each modality give more detailed views of the vertebrae and its surrounding soft tissues, discs, and foramen. This allows the physicians to observe more pathological features (and is recommended by (Tibrewal et al., 2012) as the most preferred clinical evaluation approach in bone-related medical imaging diagnosis), however, it adds to the difficulty for computational methods because of the image characteristics difference in different modalities. Unfortunately, the training dataset of a certain modality is often limited, which means that the shape, appearance, texture, resolution of the vertebrae, as well as the image intensity distribution, varies widely in the training dataset, as shown in Fig.1(d). Spondylolisthesis grading across MRI modalities is challenging because of these varieties in training data.

We propose a faster adversarial recognition (FAR) network for accurately perform spondylolisthesis grading basing on reliable critical vertebrae detection. As shown in Fig.2, our FAR network is an adversarial training network using the multi-task recognition network as the generator and an adversarial module as the discriminator. The multi-task recognition network is designed for accurately and robustly detecting the critical vertebrae and performing spondylolisthesis grading in MRI images of different modalities in a hybrid supervised manner. Inspired by the Faster RCNN scheme (Ren et al., 2015), the multi-task recognition network uses hierarchical Feature Extracting Network (FEN) to extract image features, Region Proposal Network (RPN) to efficiently find out positive proposals (regions that have high confidence of containing vertebrae), and Multi-task Detection-Grading Module (MDGM) to precisely classify the detected vertebrae, regress their bounding boxes, and decide spondylolisthesis gradings using

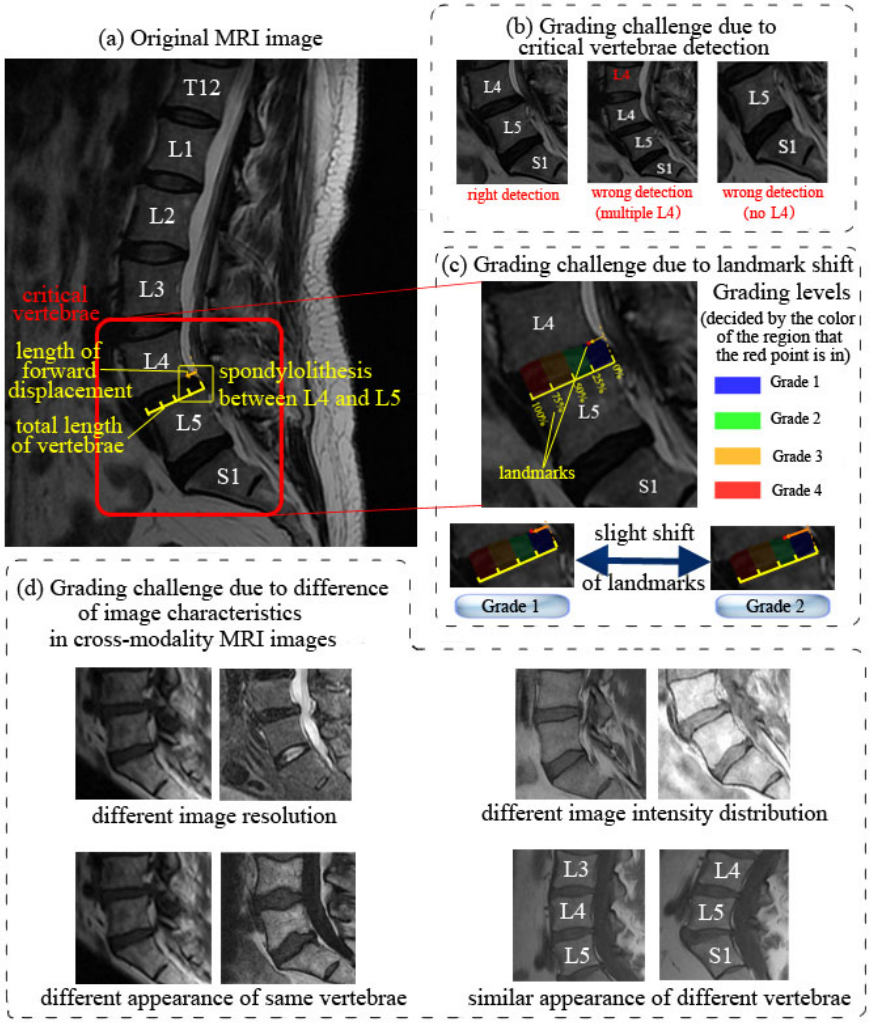

Figure 1: Challenge of automatic spondylolisthesis grading. Fig.1(a) shows the difficulty of finding critical vertebrae (the red box containing L4, L5, and S1) because of the similarity of the critical vertebrae and the other vertebrae. It also visually demonstrates the concepts mentioned in the Meyerding grading system (e.g., the length of the forward displacement is shown by the length of the orange arrow). Fig.1(b) shows failures of detecting critical vertebrae, which can hinder subsequent spondylolisthesis grading work. Fig.1(c) illustrates the difficulty of measuring the forward displacement due to the localization of the landmarks (the red point and the yellow line). An invisible slight deviation of landmarks (less than 5 pixels of red point translation or less than 10 degrees of the yellow line rotation) can lead to a large change of forward translation measurement (orange arrow length) and different grading results. Lastly, Fig.1(d) shows that image characteristics change (different image resolution, intensity distribution, and vertebrae appearance) caused by MRI images across different modalities makes the problem more challenging.

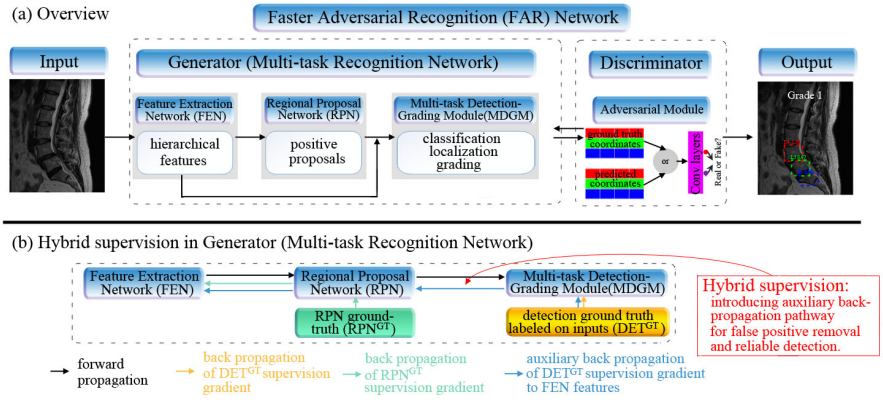

Figure 2: (a) Overview of FAR Network, which is an adversarial training network using the multi-task recognition network as the generator and an adversarial module as the discriminator. The multi-task recognition network successively uses feature extracting network (FEN) to extract hierarchical features, Region Proposal Network (RPN) for obtaining positive proposals (regions that have high confidence of containing vertebrae), and Multi-task Detection-Grading Module (MDGM) for accurate vertebrae classification, box regression, and spondylolisthesis grading. The adversarial module is used for enforcing the relative positions relationships of the detection box coordinates by leveraging their high-order statistics, which promotes the generator to yield logical and precise results. (b) The hybrid supervision strategy in the multi-task recognition network. Hybrid supervision introduces an auxiliary supervision pathway to allow gradients to smoothly back propagate from detection level to FEN features, which helps eliminate false positives and reliably detect critical vertebrae.

the shared features without the need of localizing the landmarks. Auxiliary supervision is introduced to multi-task recognition network to provide the loss gradient back-propagation pathway to FEN, which promotes the network to evolve towards producing reliable detections and prevents multiple detection or missing detection. The discriminative network is designed for improving the detection performance by using an adversarial model to access the high-order statistics of the detection box coordinates.

1.1. Existing work on spondylolisthesis grading.

Limited automatic spondylolisthesis grading work has been attempted in the existing literature. (Jamaludin et al., 2017) uses six main radiological features obtained from MRI images using a CNN model including spondylolisthesis. Spondylolisthesis is considered to be a binary measure of the vertebral slip, i.e., it is graded by 0 (no vertebral slip) and 1 (vertebral slip). Results comparable with those of an expert radiologist are reported. In (Liao et al., 2016), a hierarchical learning approach is used to detect and label vertebrae centers. Then, a critical anatomy region propagation method is used to roughly estimate the superior and inferior surfaces of vertebrae, then the endpoints of these surfaces are extracted using the domain-specific information. Lastly, spondylolisthesis grade is successfully and robustly determined using the Meyerding grading system (Niggemann et al., 2012) in CT images. However, as mentioned above, grading spondylolisthesis into several stages (which is more than simply judging the occurrence of vertical slip) from MRI images is more challenging because not all images in a dataset are collected using the same modality.

1.2. Methodology overview.

1.2.1. Instance detection

Recent instance detections methods are impressive and have achieved great success in many applications (Gao et al., 2017a,b; Zhao et al., 2019), however, they cannot be directly applied to detecting critical vertebrae for spondylolisthesis grading due to the complexity of the problem. Recent instance detections methods are mainly divided into two categories: two-step detection (such as Faster RCNN (Ren et al., 2015)) and one-step detection (such as YOLO (Redmon et al., 2016) and SSD (Liu et al., 2016)). In two-step detection, regions that have high confidence of containing object are first proposed, then the proposed regions are used for cropping the extracted features for object classification and refined bounding box regression; while in one-step detection, the images are directly separated into grids, these grids are used directly for object classification and bounding box regression. Generally, two-step detection has a higher detection accuracy but slower speed than one-step detection (Zhao et al., 2018). Methods of both categories have shown good results in object detection (Ren et al., 2015; He et al., 2017), architectural distortion detection (Ben-Ari et al., 2017), and small object detection (Li et al., 2017). In medical image analysis domain, although the postures of the objects do not change much, a great challenge is that different objects can also show similar appearances (for example, people in real-world images can be with different body postures, while

other non-people objects are significantly different from people; as a contrast, the vertebrae usually show similar appearances, while different vertebrae looks relatively the same). To make it more challenging, the successive grading task demands a perfect detection accuracy of discriminating similar-appearing vertebrae. Thus, even the more accurate the two-step detection method is chosen, multiple detection or missing detection can still happen in some images in the datasets(Fig.1(b)). Unfortunately, neither multiple detection nor missing detection is tolerable for the successive grading task. Fine tuning the hyper-parameters of existing Faster RCNN can adjust the number of detected objects in each image, however, this strategy is not robust enough to deal with the image characteristic variance of MRI images.

1.2.2. Adversarial Training

Adversarial Training, represented by the Generative Adversarial Networks (GANs) has the potential to be used in monitoring its generative network (G) by a discriminative network (D) to improve its performance. G and D are iteratively and sequentially trained to complete the following minimax optimization: (Goodfellow et al., 2014)

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_z}[\log(1 - D(G(\mathbf{z})))] \tag{1}$$

where $\mathbf{x}$ is the training data, and its distribution is p_{data}; $\mathbf{z}$ a random noise vector sampled from a certain distribution p_z, and the denotation $G(\mathbf{z})$ means G tries to generate synthetic data using $\mathbf{z}$. In solving the min-max optimization in equation (1), G is trained to generate data who have the same distribution as the real ones and minimize the probability of its generated data being recognized, and D is trained to discriminate the synthetic data from the real ones. GAN has been successfully used for image synthesis (Reed et al., 2016), image semantic segmentation (Luc et al., 2016), and super-resolved representations for small objects (Li et al., 2017). Since the input of D in equation (1) is the whole real image (or the whole output of G), the adversarial network is able to access large portions of its input (or the entire input), which implicitly leverages the high-order potential (for example the label consistency over super-pixels) to detect and correct mistakes in the synthesis/segmentation work. This is believed to be beneficial to the performance of G. Thus, we conjecture that adversarial training can similarly be used for vertebrae detection because the coordinates of these vertebrae also have some implicit internal higher-order potentials, which can be excavated for enforcing the relative positions of vertebrae coordinates for more precise detection. However, GAN has been known to be unstable to train (Radford et al., 2015), which leads to oscillation and mode collapse. Previous studies suggested several ways (such as virtual batch normalization (BN) in both G and D (Salimans et al., 2016), leaky ReLU activation in D (Radford et al., 2015), and gradient penalties in D (Mescheder et al., 2018)) to stabilize GAN training. These methods are easy to be embedded in the object detection framework because this framework contains CNN's, BN's and

ReLU's. Thus, combining object detection network and adversarial learning makes perfect complementation to each other.

1.3. Contributions.

- For the first time, we proposed FAR network to accomplish 3-level spondylolisthesis grading from MRI images accurately and robustly, which eliminates the tedious and irreproducible manual work.
- We provide a path for gradient back-propagation through the box coordinates by introducing hybrid supervision theme to two-stage detection network. This strategy provides supervision of ground truth object classes and box coordinates to the hierarchical feature extracting network, which results in excellent detection performance without false positive detections.
- We leverage the internal high-order relationships of the detected bounding box coordinates by combining adversarial module with the multi-task recognition network. This strategy enforces the relative positions relationships of the detected vertebrae, which refines the coordinates of the detected objects to make them more precise.
- We perform accurate and efficient spondylolisthesis grading by designing a novel multi-task detection grading module. This design leverages the shared hierarchical features to prompt the mutual benefit between the detection and grading tasks.

2. Methodology

As shown in Fig.2, the FAR network is composed of two parts: (1) The multi-task recognition network (generator, section *2.1*), which is a hybrid supervision network composed of a hierarchical feature extraction network (FEN), a regional proposal network (RPN), and a multi-task detection-grading module (MDGM). As shown in Fig. 3, FEN extracts abundant hierarchical features for the following network; RPN accurately and efficiently find out positive proposals; and MDGM accurately perform classification, bounding box regression, and spondylolisthesis grading after RoI pooling using proposals. RPN and MDGM are designed to share hierarchical features obtained by FEN. They both involve a loss term, but MDGM's loss term is dependent on the output of RPN. Under this configuration, auxiliary supervision that performs classification and bounding box regression similar to MDGM is introduced. Anchors are used for RoI pooling in the auxiliary branch, which helps the gradients to smoothly back-propagate from detection-level to FEN features. (2) The adversarial module (discriminator, section *2.2*), which refines the coordinates of the detected objects by using an adversarial model to assess the higher-order statistics of the bounding box coordinates.

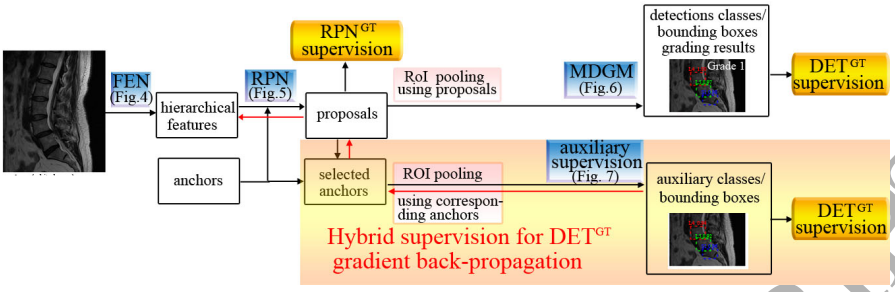

Figure 3: The overall procedure for training the multi-task recognition network. The abundant hierarchical features from FEN are fed into RPN for finding out the positive and negative proposals. Then, the proposals, as well as the shared hierarchical features from FEN, are fed into MDGM for classification, bounding box regression, and spondylolisthesis grading. Hybrid supervision (the shadow region) introduces an auxiliary supervision branch to provide back-propagation pathway for the ground truth of MDGM detection task (which are directly annotated on the input images, denoted as DET^{GT}) to FEN by using constant anchors to crop the features.

2.1. Multi-task recognition network.

2.1.1. Feature extraction network (FEN)

FEN uses bottom-up layers and top-down layers in sequence to extract abundant hierarchical features and leverage the semantics from low to high levels. Inspired by the Faster RCNN and Mask RCNN work, the FAR network used Resnet (He et al., 2016) with top-down pathway (Lin et al., 2017) as the backbone feature extracting CNN network to build up pyramid image features from the input 512×512 MRI image, as shown in the left part of Fig.4. The Resnet uses shortcut connections to improve training accuracy and solves the degradation problem in deeper networks. Image features are extracted after each stage of building blocks (conv2_x∼conv5_x in Table.1 of the original Resnet paper (He et al., 2016)) and denoted as C2∼C5. In our work, these features are of size 128×128×256, 64×64×512, 32×32×1024, and 16×16×2048. However, different from the original Resnet, we use group normalization (GN) (Wu and He, 2018) instead of batch normalization (BN) (Ioffe and Szegedy, 2015) after each convolution layer in the building blocks. This is because BN may not perform well with small batch size (for example, 2 or 4) as in medical image analysis domain. GN, however, divides the channels of all the input features in a batch into several groups (the group number is set to 32 by default), then calculate the mean and standard deviation of each group, and lastly perform the normalization within each group. This normalization is thus not dependent on batch size, which improves the accuracy when the batch size is small.

After obtaining the pyramid image features, the top-down layers perform up-sampling and merges the up-sampled feature maps with the lower level from the pyramid by lateral connections. As shown in the right part of Fig.4, the top-down layer starts from the top feature map (which is the coarsest and with the strongest semantics) and iteratively up-samples the feature maps. In each itera-

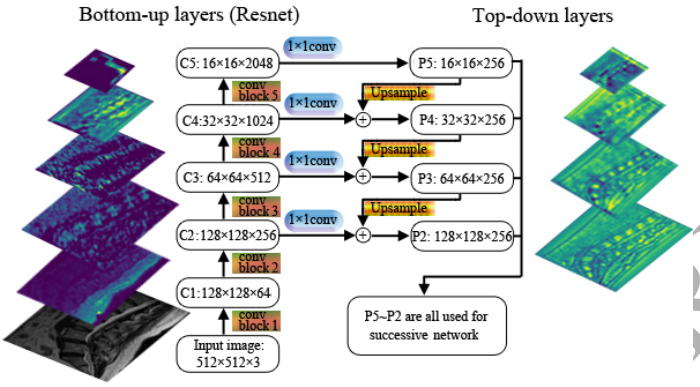

Figure 4: The feature extraction network (FEN) architecture. The bottom-up layers use Resnet to improve training accuracy and solved the degradation problem in deeper networks; the top-down layers build up a feature pyramid and share features of all levels to successive networks.

tion, the resolution is doubled, and the resulting feature map is merged with the corresponding lower level feature map by lateral connections (implemented by pixel-wise add). The merged feature map is then fed into the next up-sampling unit. The resultant feature maps (denoted as $P_5 \sim P_2$, whose resolutions are respectively 16×16×256, 32×32×256, 64×64×256, and 128×128×256) contain ablated information of all scales, which to the utmost extent retains spatial information and mitigates the semantic gaps of the bottom-up layers. These $P_5 \sim P_2$ are shared to the successive networks, which enhances the learning efficiency and grading accuracy for both detection and grading.

2.1.2. Regional proposal network (RPN)

The RPN is used for coarsely detecting object locations with high objectness scores in the form of proposals (bounding boxes). RPN first equidistantly samples grid points from the original input image and places boxes of different size and aspect ratio (namely, anchors, as shown in Fig.5) centered on the grid points at several hundred pre-defined locations in the input images (Ren et al., 2015).

As shown in Fig.5, the RPN is a fully-convolutional network that finds regions having high confidence of containing vertebrae. Following (Ren et al., 2015), RPN takes shared hierarchical features as input; it uses a 3×3 convolutional layer for dimension reduction, then uses two sibling 1×1 convolutional layers to predict **RPN class logits ($\mathbf{RPN^{pred}_{CL}}$)** and **RPN bounding box corrections ($\mathbf{RPN^{pred}_{BBC}}$)**. The RPN^{pred}_{CL}'s indicate the possibilities of the corresponding anchors containing vertebrae, while the RPN^{pred}_{BBC}'s are the modification values for transforming anchors to regional proposals. After obtaining proposals by applying RPN^{pred}_{BBC}'s to anchors, the predicted RPN^{pred}_{CL}'s and

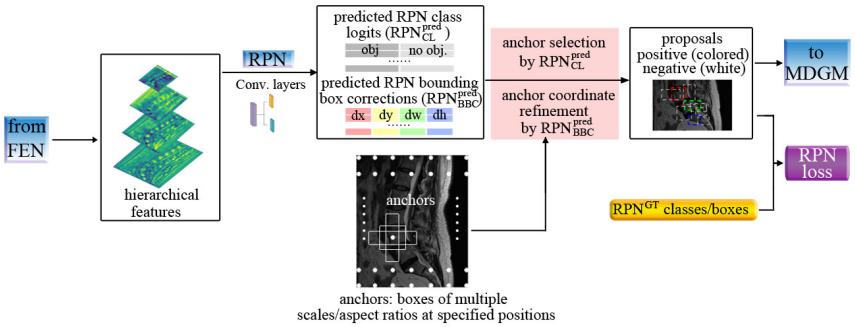

Figure 5: The detailed architecture of RPN, where two sibling networks are used to find out the possibilities of containing objects for each anchor (RPN_{CL}^{pred}) and their bounding boxes (proposal coordinates, which are refined anchors by RPN_{BBC}^{pred}). These procedures are supervised by RPN^{GT}, namely, RPN_{CL}^{GT} for supervising predicted RPN class logits (RPN_{CL}^{pred}) and RPN_{BOX}^{GT} for supervising the proposal coordinates.

proposals are used to calculate RPN loss with **RPN ground truth classes ($\mathbf{RPN_{CL}^{GT}}$)** and **RPN ground truth boxes ($\mathbf{RPN_{BOX}^{GT}}$)**. It should be noted that the **RPN ground truth ($\mathbf{RPN^{GT}}$)**'s are not the **manually labeled ground truth for the detection task ($\mathbf{DET^{GT}}$)** in MDGM on the input images because the proposals are class-agnostic, and each DET^{GT} box may assign more than one RPN^{GT} box to be positive (Ren et al., 2015).

The RPN loss function has a form of:

$$L_{RPN} = L_{RPN}(\boldsymbol{p}(\boldsymbol{\theta}), \boldsymbol{p}^*, \boldsymbol{t}(\boldsymbol{\theta}), \boldsymbol{t}^*) \tag{2}$$

where $\boldsymbol{\theta}$ represents all network parameters in FEN and RPN; $\boldsymbol{p}(\boldsymbol{\theta})$ and $\boldsymbol{t}(\boldsymbol{\theta})$ are lists of predicted RPN_{CL}^{pred}'s and proposal coordinates: $\boldsymbol{p}(\boldsymbol{\theta}) = \{\boldsymbol{p_i}(\boldsymbol{\theta})\}$ and $\boldsymbol{t}(\boldsymbol{\theta}) = \{\boldsymbol{t_i}(\boldsymbol{\theta})\}$, where i is the index of a proposal. $\boldsymbol{p}^*$ and $\boldsymbol{t}^*$ are correspondingly RPN_{CL}^{GT}'s and RPN_{BOX}^{GT}'s. The coordinates for each proposal is a list: $\boldsymbol{t_i}(\boldsymbol{\theta}) = \{t_{ix}(\boldsymbol{\theta}),\ t_{iy}(\boldsymbol{\theta}),\ t_{iw}(\boldsymbol{\theta}),\ t_{ih}(\boldsymbol{\theta})\}$, which respectively mean the x position of the proposal center, y position of the proposal center, proposal width, and proposal height. These notations in equation (2) show that the RPN loss is function of FEN and RPN parameters $\boldsymbol{\theta}$.

The RPN usually yields hundreds of proposals (each corresponds to an anchor). We use non-maximum suppression (NMS) and hard negative mining (HNM) to reduce redundancy. Only negative proposals with high objectness scores (which are difficult to recognize from positive ones) are preserved to satisfy a constraint (Negatives: Positives = 5:1). The positive proposals and preserved negative proposals are fed into the successive network (MDGM in our work). Usually, RPN is able to correctly find out regions having high confidence of containing vertebrae. However, due to the difference in RPN^{GT} and DET^{GT} (the DET^{GT} of one vertebra may assign more than one RPN^{GT} box to be positive), the training objective of FEN and RPN may be deviated (be-

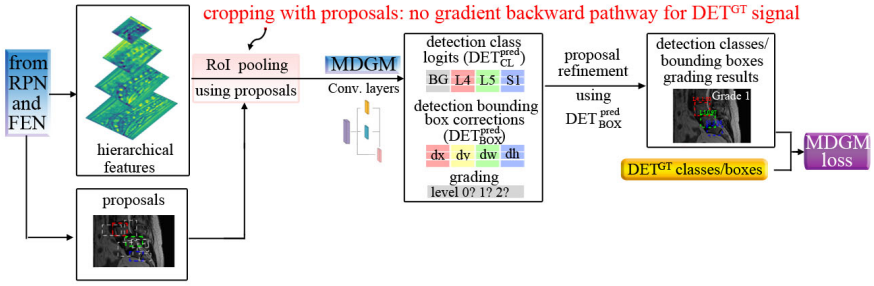

Figure 6: The detailed architecture of MDGM, where the final detection classes, box coordinates, and spondylolisthesis grading tasks are performed using the proposals and the shared hierarchical features. RoI pooling is performed by using proposal coordinates to crop the features. The cropping operation is not differentiable to the proposal coordinates, so the MDGM network are separately supervised with RPN (the MDGM results are supervised by DET^{GT}, which is the labeled ground truth on input images; the RPN results are supervised using RPN^{GT}, as shown in Fig.5).

cause an anchor that is actually negative may be assigned a positive label during training), which results in false positives in some images. Also, the RPN^{pred}_{BBC}'s may have large errors (namely, the proposals may have large deviations to the ground truth boxes) because it uses linear corrections to correct anchors which are relatively far to the target boxes. Thus, only using RPN^{GT} to supervise FEN and RPN might not be enough.

2.1.3. MDGM

Inspired by the RoI pooling layers in Faster RCNN, we introduce the shared features from FEN to perform critical vertebrae detection and spondylolisthesis grading in a multi-task network. As shown in Fig.6, the MDGM simultaneously performs the detection task and the grading task; and the detection task has two sub-tasks: classification and bounding box regression.

In the detection task, the selected proposals and the shared features are fed into MDGM to achieve these tasks while reinforcing the mutual benefit them by sharing parameters from the multi-output learning architecture. The shared features are of different spatial resolutions ($P_5 \sim P_2$), we choose one for each proposal using the feature level calculated by

$$k = \lfloor k_0 + \log_2(\sqrt{wh}/h_0) \rfloor \tag{3}$$

where w and h are the width and height of the proposal. This equation is similar to that in (Lin et al., 2017), except that we changed the canonical ImageNet pre-training size (224) to h_0, which is set to be 64 since the widths and heights of most vertebrae in our work is near 64. Then we set k_0 to be 4, which result in the 64×64 vertebrae corresponding to P_4. This configuration corresponds to Resnet (He et al., 2016) which uses C_4 as the single-scale feature map. For smaller proposals, a finer-resolution feature is used, which corresponds to a smaller k and a larger P_k resolution (for example, a proposal of size 32×32 corresponds to

P_2). This configuration also ensures most vertebrae has a feature level between 2∼5 because $h \times w$ of most vertebrae are between 16×16 and 128×128. For the very few exceptions, their feature levels are cut off to 2 (for vertebrae smaller than 16×16) or 5 (for those larger than 128×128). The feature level k is chosen specifically for each proposal, which ensures that most vertebrae of different scales can be assigned to features of appropriate scale and tackles the multi-scale problem without the need of re-calculating the features.

After feature map selection, the chosen features are cropped by the corresponding proposals, then each cropped feature is resized to 7×7. In this procedure, bilinear interpolation is used to compute the values of shared FEN features at the proposal coordinates used for the cropping operation (since the cropping coordinates may not be integer pixels) to align the features with the cropping proposals, which achieves better detection box coordinate accuracy (He et al., 2017). Next, the cropped and resized features are fed into 2 cascading convolutional layers (the first one has a kernel size of 7×7, and the second 1×1) to obtain shared features, and then the shared features are fed into two siblings 1×1 convolutional layers to predict the **detection object class logits ($\mathbf{DET^{pred}_{CL}}$)** and the **detection bounding box corrections ($\mathbf{DET^{pred}_{BBC}}$)**. DET^{pred}_{BBC}'s are applied to the positive proposals to calculate detection box coordinates. The loss of the detection task is formulated as

$$L_{Det} = L_{Det}(\boldsymbol{q}(\boldsymbol{\theta}), \boldsymbol{q}^*, \boldsymbol{u}(\boldsymbol{\theta}), \boldsymbol{u}^* \mid \boldsymbol{p}(\boldsymbol{\theta}), \boldsymbol{t}(\boldsymbol{\theta})) \tag{4}$$

where $\boldsymbol{q}(\boldsymbol{\theta})$ and $\boldsymbol{u}(\boldsymbol{\theta})$ are respectively lists of predicted DET^{pred}_{CL}'s and detection box coordinates: $\boldsymbol{q}(\boldsymbol{\theta}) = \{\boldsymbol{q_j}(\boldsymbol{\theta})\}$ and $\boldsymbol{u}(\boldsymbol{\theta}) = \{\boldsymbol{u_j}(\boldsymbol{\theta})\}$, where subscript j is the index of a detection. $\boldsymbol{q^*}$ and $\boldsymbol{u^*}$ are correspondingly DET^{GT}_{CL} and DET^{GT}_{BOX}. Similar to the proposal, the coordinates for each detection box is a list: $\boldsymbol{u_j}(\boldsymbol{\theta}) = \{u_{jx}(\boldsymbol{\theta}),\ u_{jy}(\boldsymbol{\theta}),\ u_{jw}(\boldsymbol{\theta}),\ u_{jh}(\boldsymbol{\theta})\}$ which means the x/y position of the detection box center, and the width/height of the detection box. These notations in equation (4) show that the loss of the detection task is dependent on both detections ($\boldsymbol{q}(\boldsymbol{\theta})$ and $\boldsymbol{u}(\boldsymbol{\theta})$) and proposals ($\boldsymbol{p}(\boldsymbol{\theta})$ and $\boldsymbol{t}(\boldsymbol{\theta})$) because the detection task is based on the proposal coordinates.

In the grading task, the shared features from FEN are chosen, cropped and resized by the detection boxes. This procedure is similar to that in the detection work, except that the cropped features are resized to 32×32 for a larger resolution. Then, the resized features are fed into a 4-layer convolutional network (the first two layers with 3×3 kernel size, the third with 8×8 kernel size, the last with 1×1 kernel size; all convolutional layers are with stride 1; all layers are followed by GN and ReLU; and the first two layers are followed by 2×2 max pooling after GN and ReLU) for spondylolisthesis grading features. Then, the grading features are flattened into a row vector. Afterward, the coordinates of the detection boxes are also flattened to a row vector and concatenated with the grading features. Lastly, a 1×1 convolutional layer (serving as fully connected layers) is used to predict the logits of spondylolisthesis grading using the concatenated vector, and cross-entropy loss is used to calculate the grading loss L_{gra}.

This multi-task training strategy not only reuses the shared hierarchical feature map to reduce computation cost, but also leads to higher grading accuracy because it eliminates the interference information of other parts of the image. Moreover, it eliminates the need to localize the landmarks and completely avoids the grading error resulting from the deviation of landmarks, which greatly helps to improve the robustness of grading.

2.1.4. The auxiliary supervision branch

Although the MDGM is able to simultaneously perform detection and grading task, the cropping operation hinders the back propagation of the gradient signal of DET^{GT} from MDGM to RPN/FEN. Even though the cropping operation has no parameters, it is implemented by using the proposal coordinates $\boldsymbol{t}$ to crop out a section from the shared features, which means that the cropped features are dependent on $\boldsymbol{t}$. Resultingly, the predicted detection object class logits (DET_{CL}^{pred}), the detection bounding box corrections (DET_{BBC}^{pred}), and the loss term L_{Det} are all dependent on $\boldsymbol{t}$ because they are deduced by the cropped features. In other words, different from traditional multi-task learning, the loss terms of the later networks are dependent on the output of its preceding network (equation (4) also shows this dependence). In optimizing the network, the total loss (which contains the term L_{Det}) and (4)) is minimized by evolving the network parameters $\boldsymbol{\theta}$. This minimization is implemented by the gradient descent method. When calculating the loss gradient, the derivatives of L_{Det} w.r.t. different parameters are calculated to compose the loss gradient. Based on the chain rule of back propagation, the derivative of L_{Det} w.r.t. an arbitrary parameter θ is:

$$\frac{\partial L_{Det}}{\partial \theta} = \frac{\partial L_{Det}}{\partial \boldsymbol{u_i}} \frac{\partial \boldsymbol{u_i}}{\partial \boldsymbol{F_i}} \frac{\partial \boldsymbol{F_i}}{\partial \boldsymbol{t_i}} \frac{\partial \boldsymbol{t_i}}{\partial \theta} \tag{5}$$

where $\boldsymbol{F_i}$ is the flattened features cropped by the i^{th} proposal (Dai et al., 2016). Equation (5) means that, the derivative of L_{Det} w.r.t. a network parameter should include a term concerning the derivative of the cropped features $\boldsymbol{F_i}$ to the proposal coordinates $\boldsymbol{t_i}$ (i.e., the term $\frac{\partial \boldsymbol{F_i}}{\partial \boldsymbol{t_i}}$). However, this term is undefined because the cropping operation does not have a derivative, which means that $\frac{\partial L_{Det}}{\partial \theta}$ can not be calculated. This hinders the back propagation of the detection loss signal (L_{Det}) from MDGM to the parameters ($\boldsymbol{\theta}$) of the preceding feature extraction network (FEN) and regional proposal network (RPN).

Without the auxiliary branch, the detection network can be trained anyway by approximate joint training method which ignores the above-mentioned problem, but this strategy may give rise to false positives in the detection task. As discussed in the original Faster RCNN paper (Ren et al., 2015), the approximate joint training strategy treats the proposals as fixed, pre-computed boxes, so the derivative of L_{Det} w.r.t. the proposal coordinates $\boldsymbol{t_i}$ is simply ignored. In this way, MDGM and FEN/RPN can anyway be trained together, and the back propagation takes place as usual. However, since the back propagation pathway of the loss gradient from MDGM to the preceding network (FEN/RPN) is

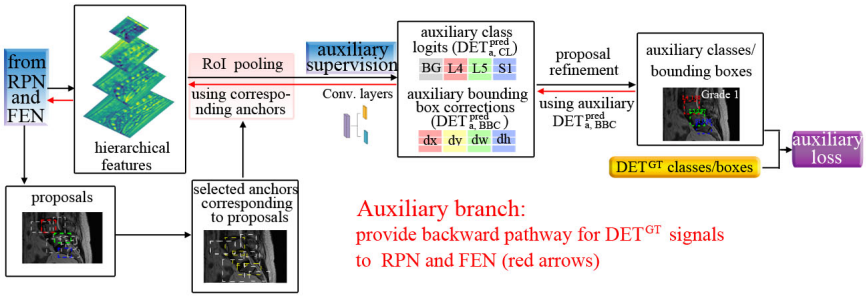

Figure 7: The auxiliary supervision branch in hybrid supervision. The auxiliary supervision works together with the supervisions in the main branch, which compose the hybrid supervision strategy. This hybrid supervision strategy introduces DET^{GT} signals (rather than simply using RPN^{GT} signals) to train FEN/RPN networks, and helps jointly training the whole multi-task recognition network, which promotes the network to yield reliable detections without false positives.

blocked, the detection ground truth (DET^{GT}) can only be used to supervise the MDGM network (but not the FEN/RPN network). In other words, although the two parts (MDGM and FEN/RPN) are jointly trained, they are actually supervised by different ground truths separately during the training. The FEN and RPN are only supervised by the RPN^{GT}, which may lead to false positives and large localization errors. In original faster RCNN, this does not seem a big problem because their experiments show that the RPN^{GT} supervision can achieve satisfactory detections in general. However, since a much more strict detection performance is demanded (no multiple/missing detection, as well as high IoU of detection with DET^{GT}_{BOX}) in our application, this strategy might not be enough because of false negatives and large errors of RPN^{pred}_{BBC}'s.

Thus, inspired by deep supervision (where an auxiliary branch is added to the main branch to provide gradient signals) (Lee et al., 2015) and Single Shot Detection (SSD, where fixed anchors of different scales are used to perform classification and bounding box regression for objects of different scale) (Liu et al., 2016), we add an auxiliary branch (the network in the red shadow area in Fig.3, and it is detailed in Fig.7) to provide smooth propagation pathway for DET^{GT} signals to FEN. Similar to SSD, anchors corresponding to the selected proposals by the NMS and HNM are used to crop the features and perform the detection task (classification and bounding box regression) in the auxiliary supervision branch. The importance of this auxiliary back propagation pathway is that it allows the DET^{GT} to be used to supervise the FEN/RPN and remove false negatives. Since the coordinates of anchors are now indeed constant (not "treated as" constant), the gradient of L_{aux} does not involve the gradient w.r.t. proposal coordinates $\boldsymbol{t_i}$. Namely, the derivative of L_{Det_aux} w.r.t. θ is:

$$\frac{\partial L_{Det\_aux}}{\partial \theta} = \frac{\partial L_{Det\_aux}}{\partial \boldsymbol{u_i}} \frac{\partial \boldsymbol{u_i}}{\partial \boldsymbol{F_i}} \frac{\partial \boldsymbol{F_i}}{\partial \theta} \tag{6}$$

Equation (6) is similar to (5), but the term $\frac{\partial \boldsymbol{F_i}}{\partial \boldsymbol{t_i}} \frac{\partial \boldsymbol{t_i}}{\partial \theta}$ is removed because the coordinates of anchors are no longer dependent on parameter θ. Thus, the auxiliary branch bridges the FEN/RPN and auxiliary detection network into a whole with a smooth back propagation pathway for the gradient of the DET^{GT} signal to the FEN/RPN. This auxiliary supervision branch is similar to deep supervision (Lee et al., 2015) in that the supervision is added after the gradient value vanishes, and the added supervision provides a stronger gradient signal to facilitate training. The auxiliary branch can be trained with any preceding/succeeding layers. Thus, network parameters $\boldsymbol{\theta}$ can be updated using gradient signals form DET^{GT}. Although the RPN^{GT} is unchanged, the introduction of DET^{GT} supervision to the FEN and RPN network can efficiently remove the false positives caused by RPN^{GT} supervision.

In more details, RPN^{GT} may have more than one positive boxes corresponding to one positive DET^{GT}, which misleads the FEN/RPN training to regard some proposals that are actually negative as positive. This resultantly gives rise to false positive anchors/proposals. In the main branch, all proposals (including false positives) are fed into MDGM through ROI pooling. When the DET^{GT} distinguishes the false positives, a large detection loss (L_{Det}) is produced. However, without the auxiliary branch, the large L_{Det} signal cannot be used to update the FEN/RPN because of the blocked back propagation pathway. As a contrast, in the auxiliary branch, a large auxiliary loss (L_{aux}) is similarly produced by the supervision of DET^{GT} when the auxiliary branch receives false positive anchors produced by FEN/RPN. With the help of the auxiliary branch, the loss signal from L_{aux} can now be back propagated to FEN/RPN through the auxiliary pathway. In this way, these networks are updated by DET^{GT} signal and thus promoted to yield reliable proposals (and thus final detections) without false positives in the forward pass. Also, the two bounding box corrections RPN^{pred}_{BBC}'s and DET^{pred}_{BBC}'s are trained in a unified network in the auxiliary branch. The mutual beneficial effect of RPN^{pred}_{BBC}'s and DET^{pred}_{BBC}'s prediction is enhanced to provide a more appropriate intermediate variable value (the proposal positions in our work). As such, the preceding RPN^{pred}_{BBC}'s and succeeding DET^{pred}_{BBC}'s can be more efficient in linearly correcting the bounding box positions, which at last results in more proper detections.

2.1.5. Objective function of Multi-task recognition network

The objective function of the detection framework includes four parts: the RPN loss L_{RPN}, the MDGM detection loss L_{Det}, the MDGM grading loss L_{Gra}, and the auxiliary loss L_{aux}. These losses respectively corresponds to the four terms in equation (7); L_{RPN}, L_{Det} and L_{aux} are further composed of

classification loss and bounding box loss.

$$
\begin{aligned}
L_G &= L_{RPN} + L_{Det} + L_{Gra} + L_{aux} \\
L_{RPN} &= \frac{\lambda_1}{N_1}\sum_{i_1=1}^{N_1} L_{cls}(\boldsymbol{p_{i_1}}, p^*_{i_1}) + \frac{\lambda_2}{N_2}\sum_{i_2=1}^{N_2} L_{loc}(\boldsymbol{t_{i_2}}, \boldsymbol{t^*_{i_2}}) \\
L_{Det} &= \frac{\lambda_3}{N_3}\sum_{i_3=1}^{N_3} L_{cls}(\boldsymbol{q_{i_3}}, q^*_{i_3}) + \frac{\lambda_4}{N_4}\sum_{i_4=1}^{N_4} L_{loc}(\boldsymbol{u}^{q^*_{i_4}}_{\boldsymbol{i_4}}, \boldsymbol{u^*_{i_4}}) \\
L_{Gra} &= \lambda_5 L_{cls}(\boldsymbol{G}, \boldsymbol{G^*}) \\
L_{aux} &= \frac{\lambda_6}{N_3}\sum_{i_3=1}^{N_3} L_{cls}(\boldsymbol{q_{a,i_3}}, q^*_{i_3}) + \frac{\lambda_7}{N_4}\sum_{i_4=1}^{N_4} L_{loc}(\boldsymbol{u}^{q^*_{i_4}}_{\boldsymbol{a,i_4}}, \boldsymbol{u^*_{i_4}})
\end{aligned}
\tag{7}
$$

All loss terms in equation (7) are functions of network parameters $\boldsymbol{\theta}$, but the notations $\boldsymbol{\theta}$ are omitted for simplifying presentation.

In L_{RPN}: (1) The first term is RPN class loss, where i_1 is the index of an anchor, N_1 is the total number of the positive and negative anchors in RPN, $\boldsymbol{p_{i_1}}$ is predicted RPN^{pred}_{CL}, $p^*_{i_1}$ is the corresponding ground truth label RPN^{GT}_{CL} for this anchor, L_{cls} means the cross-entropy loss of $\boldsymbol{p_{i_1}}$ and $p^*_{i_1}$. (2) The second term is the RPN box loss, where i_2 is the index of a RPN^{GT} positive anchor (an anchor whose RPN^{GT}_{CL} is true, i.e., it contains vertebrae), N_2 is the total number of the RPN^{GT} positive anchors (because only anchors whose RPN^{GT}_{CL} is positive account for bounding box regression loss), $\boldsymbol{t_{i_2}}$ is the list representing a predicted proposal coordinates; $\boldsymbol{t^*_{i_2}}$ is the RPN^{GT}_{BOX} coordinates corresponding to each prediction, and L_{loc} means the smooth L1 loss defined as:

$$
L_{loc}(\boldsymbol{t_i}, \boldsymbol{t^*_i}) = \sum_{c\in\{x,y,w,h\}} \begin{cases} 0.5(t_{ic} - t^*_{ic})^2 & if\ |t_{ic} - t^*_{ic}| < 1 \\ |t_{ic} - t^*_{ic}| - 0.5 & otherwise \end{cases} \tag{8}
$$

where t_{ic} means one element in the predicted proposal coordinates list $\boldsymbol{t_i} = \{t_{ix},\ t_{iy},\ t_{iw},\ t_{ih}\}$; and t^*_{ic} means the corresponding RPN^{GT}.

In L_{Det}: (1) The first term is detection class loss, where i_3 is the index of a selected proposal, N_3 is the total number of selected positive and negative proposals after HNM and NMS, $\boldsymbol{q_{i_3}}$ is predicted detection class logits DET^{pred}_{CL}, $\boldsymbol{q^*_{i_3}}$ is the corresponding detection class ground truth DET^{GT}_{CL}. The cross-entropy loss of $\boldsymbol{q_{i_3}}$ and $q^*_{i_3}$ are calculated as the detection class loss. (2) The second term is detection box loss, where i_4 is the index of a positive proposal, N_4 is the total number of the positive proposals after HNM and NMS, $\boldsymbol{u}^{q^*_{i_4}}_{\boldsymbol{i_4}}$ is a list representing the predicted detection box coordinates detailed in sub-section *2.1.3*, while the superscript $q^*_{i_4}$ means only the box of the ground truth class is used, $\boldsymbol{u^*_{i_4}}$ is the corresponding DET^{GT}_{BOX} coordinates. The term L_{loc} means the smooth L1 loss defined in equation (8).

In L_{aux}: The two terms (auxiliary class loss and auxiliary bounding box loss) are similar to those in L_{Det}, the only difference is that the predictions (class logits $\boldsymbol{q_{a,i_3}}$ and detection box coordinates $\boldsymbol{u}^{q^*_{i_4}}_{\boldsymbol{a,i_4}}$) are acquired from the auxiliary branch, and we thus use subscript a. The same cross-entropy loss and smooth L1 loss formulas as in MDGM are used in the auxiliary branch.

The weights ($\lambda_1 \sim \lambda_7$) in equation (7) are selected based on the previous experience of the original Faster RCNN paper (Ren et al., 2015), the original deep supervision paper (Lee et al., 2015) as well as our experiments. Based on the following considerations and experiments, we set all $\lambda_1 \sim \lambda_7$ to be 1, and add a constraint to force the losses of the auxiliary branch to be no larger than those of their corresponding main branch:

RPN and MDGM loss weights. The weights of RPN loss and MDGM loss ($\lambda_1 \sim \lambda_5$) are selected according to the results of the original Faster RCNN network. In (Ren et al., 2015), it is proved by experiment that change of the weights does not significantly affect the results within a scale of about two orders of magnitude. We infer that the weights between different parts of the multi-task recognition network (RPN and MDGM), and those among different tasks (classification, bounding box regression, and grading) in RPN and MDGM do not significantly affect the performance of the FAR network in a relatively wide scale. We thus simply set all these weights to be 1. We also changed λ_2 and λ_4 to 0.1 and 10 to alter the weight trade-off between classification and bounding box localization; the results show that the changes of the weights do not make significant in either grading or detection performance.

Auxiliary loss weights. The weights of the auxiliary branch ($\lambda_6 \sim \lambda_7$) are selected according to the results of the original Faster RCNN work (Ren et al., 2015), deep supervision work (Lee et al., 2015) as well as our own experiments. First, we set $\lambda_6 = \lambda_7$ as in (Ren et al., 2015). Then, we find in (Lee et al., 2015) that the weights of the auxiliary branches linearly decays as a function of epoch with the initial value 0.3. We infer that this configuration means: (1) The auxiliary supervision branch should provide gradient signals at the beginning of the training. (2) The auxiliary supervision loss should not account for a too large proportion in the total loss, especially when the training comes to stability. Based on these inferences, we set $\lambda_6 = \lambda_7 = 1$ and add a constraint to force the losses of the auxiliary branch to be no larger than those of their corresponding main branch. We also tried the same configuration of λ_6 and λ_7 as in (Lee et al., 2015), the grading or detection performance does not show significant changes.

2.2. Discriminator network.

The discriminative network increases detection accuracy by using an adversarial model to assess the joint distributions of the coordinates of the bounding boxes. This thought is inspired by the semantic segmentation scheme using adversarial networks, where higher-order terms are used to reveal and reinforce the spatial label relationship of adjacent points (for example, label consistency can be used as a higher-order term because the labels of adjacent points should probably be the same (Luc et al., 2016)). In our work, since the coordinates of critical vertebrae are also related to each other, we are inspired to leverage the higher-order terms of the critical vertebrae coordinates to improve detection performance. However, specifying one kind of higher-order terms for coordinates is difficult because the coordinates are continuous variables, and relative relationships of coordinates (for example the height, area, distance of different vertebrae, and even implicit joint distributions that are human-inconspicuous)

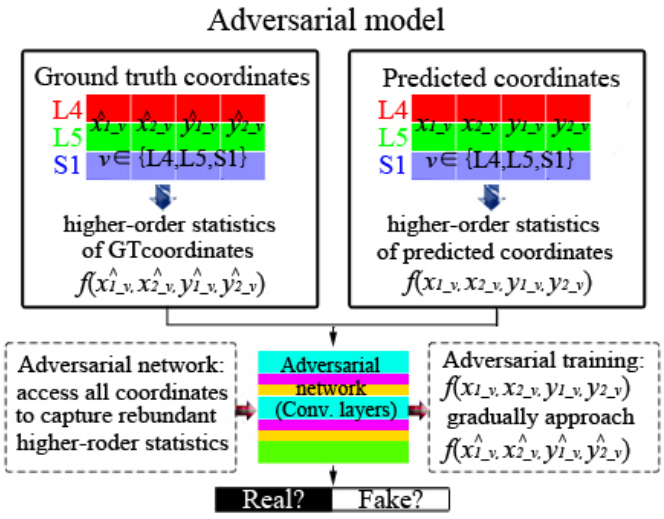

Figure 8: The adversarial model is able to simultaneously access all the coordinates of different vertebrae, which helps capture the implicit higher-order statistics in these coordinates. The adversarial training promotes the higher-order statistics of the predicted coordinates to gradually approach that of the ground truth coordinates. This enforces the internal relationships and regulates the distribution of the bounding box coordinates, which improves the performance of the generator.

are more often used rather than the values of the coordinates. Thus, we use an adversarial module to monitor a wide range of higher-order statistics of the bounding box coordinates without being limited to several manually defined kinds of higher-order statistics. Another advantage of introducing adversarial module to our detection work is that the positions of the critical vertebrae are dependent on each other, whereas the detection framework neglects this dependence and only performs independent detections of each vertebra. The adversarial element can be a sound choice to make full use of these internal higher-order statistics as it is able to access all the detection boxes. By leveraging these higher-order statistics as soft constraints (if these constraints are violated, D can discriminate the output of the detection network from ground truth detection boxes, and the D loss is high), the adversarial module serves as a supervisor to guide the higher-order statistics of detected vertebrae to gradually approach the ground truth higher-order statistics, as shown in Fig.8. In this way, the adversarial module helps the detection network to improve its performance.

In the adversarial module, we take the normalized coordinates of the ground truth boxes as the "real data" for all critical vertebrae in an image. This results in a 3×4 coordinate matrix (each row of the matrix corresponds to the four angular points of the bounding box of one critical vertebra). This ground truth coordinate matrix is used as $\mathbf{x}$ (real data) in equation (1) since the ground

truth box coordinates have certain spatial distributions and internal higher-order statistics, which is expected to be learned by G. Similarly, the predicted coordinate matrix composed of normalized coordinates of the predicted detection boxes serves as $G(\mathbf{z})$ in equation (1). However, since the input of G is the images $\mathbf{I}$ instead of the noise $\mathbf{z}$, we change the notation to $G(\mathbf{I})$ for clarity.

Since the coordinate matrix (input of D) is a tensor with a rather small size (much smaller than the images used in image semantic segmentation work), D can be implemented by a relatively simple network with fewer parameters to enhance training efficiency and avoid over-fitting. We use two convolution layers (3×3 kernel size, stride 1, with GN and leaky ReLU to stabilize the training (Radford et al., 2015)) followed by a fully connected layer to build the D network, as shown in Fig.8. This network, although simple, can access the coordinates of all the three detection boxes, so the mismatches in the high-order statistics of different vertebrae can be penalized by the adversarial loss term (Luc et al., 2016). For a comprehensible example, if wrong detection happens (say, for example, L3 is regarded as L4 by G), the coordinate matrix relationship would be significantly different from that of the correct coordinate matrix (the deviation of y coordinates of L4 to that of L5 will significantly increase because detected L4 is far above real L4); and this will be penalized by the adversarial loss, which urges the coordinates to gradually approach the internal distribution of the critical vertebrae coordinates.

2.3. FAR network training strategy.

The training strategy of FAR network empirically has a stable performance by combining the robust losses of the multi-task recognition network and the discriminative network. Since the multi-task recognition loss has been detailed in sub-section *2.1.5*, We mainly consider the adversarial loss here. In training, D network simultaneously receives the predicted coordinate matrix and the ground truth coordinate matrix and calculates $D(\mathbf{x})$ or $D(G(\mathbf{I}))$ ($D(\cdot)$ means the probability that D judges its input to be ground truth), then D loss is calculated as equation (1). FAR network uses a hybrid loss function that is a weighted sum of two terms comprising a detection loss and GAN loss:

$$\min_G \max_D \alpha L_G(G(\mathbf{I}), \mathbf{x}) - \beta[L_D(D(\mathbf{x}), 1) + L_D(D(\mathbf{G(I)}), 0)] \tag{9}$$

In equation (9), the term $L_G(G(\mathbf{I}), \mathbf{x})$ corresponds to the G loss, which is a concise denotation of equation (7) since it is a function of detection network parameters taking the image $\mathbf{I}$ and LGT detections $\mathbf{x}$ as inputs. The last two terms (with the negative sign) are D loss (the probability of the output results can be discriminated by D), which correspond to the terms in equation (1) and are calculated by the opposite number of the cross-entropy of the outputs of D and 1's for ground truth data (and 0's for predicted detections). The weights α and β are set to be 1.

When training, G and D should be trained respectively and simultaneously (Han et al., 2018; Goodfellow et al., 2014). The min-max optimization work equation (9) is split to two minimize work (equation (10) for training G and

(11) for training D). Since the last term in equation (9) is a function of both G and D parameters, it appears in both equations (10) and (11):

$$\min_G \alpha L_{detect}(G(\mathbf{I}), \mathbf{x}) - \beta L_{GAN}(D(\mathbf{G(I)}), 0) \tag{10}$$

$$\min_D L_{GAN}(D(\mathbf{x}), 1) + L_{GAN}(D(\mathbf{G(I)}), 0) \tag{11}$$

Following (Han et al., 2018), equation (10) is changed to equation (12) for a stronger gradient signal to speeds up training:

$$\min_G \alpha L_{detect}(G(\mathbf{I}), \mathbf{x}) + \beta L_{GAN}(D(\mathbf{G(I)}), 1) \tag{12}$$

To ensure that D is functioning, two momentum optimizers are respectively implemented on G and D. Considering the training of G is harder than D, the initial learning rate for G is set to be larger, as shown in Table. 1. Since the grading task is dependent on the detection task in G, the detection task in G and the adversarial module D are first respectively and simultaneously trained for 10000 steps (∼170 epochs) with the parameters of grading network fixed. Then, the grading network is trained together with the detection task and adversarial module for another 1000 steps (∼17 epochs).

3. Data and Experiments

3.1. Data acquisition.

The proposed FAR network has been intensively evaluated using a challenging dataset including 150 MRI images of different modalities (such as T1, T2, PD, and TSE) acquired from different medical centers. Besides the difference of modalities, the MRI images also have different image characteristics (such as vertebrae appearance, image resolution, intensity distribution) because they are examined by different models of vendors and scanned using different parameters. The raw MRI lumbar scans are 3D volumes containing multiple slices, however, since the universal clinical diagnosis method (Meyerding grading system) uses single slices that contain critical vertebrae to perform spondylolisthesis grading, we follow this method to perform spondylolisthesis grading using selected single slices. Among sequential MRI scans of each patient, the middle scan of the lumber is selected for a better presentation of the critical vertebrae in the sagittal direction. Thus, there are 150 lumbar scans from 150 patients in our dataset, and no patient is placed in both sets of training and testing. The numbers of vertebrae in the images of different patients vary widely from 7 to 15, which adds to the difficulty of precisely locating the critical vertebrae. The detection ground truth is labeled on each MRI image using our lab tool according to the clinical criterion. The spondylolisthesis ground truths are annotated by two experienced physicians using Meyerding grading system blinded to each other. The differences between the two manual grading results are used to assess the inter-operator variability.

Table 1: Training configurations of the FAR network

Network name	Multi-task recognition network	Discriminative network
Training method	Momentum Optimizer	
Learning rate decay type	Exponential	
Initial learning rate	1e-3	1e-4
End learning rate	1e-6	1e-5
Learning momentum	0.9	
Learning rate decay factor	0.96	
Number epochs per decay	10	

It should be noted that although we use the central sagittal slice of the 3D scan to better present the critical vertebrae and simplify the training, the FAR network is not limited to central slices, i.e., the central slice does not need to be designedly chosen from the 3D MRI scan. If the training dataset of FAR network contains sagittal slices of different indices (not only the central slice), the FAR network can learn to perform detection and grading for these slices. As long as the critical vertebrae for spondylolisthesis grading exists in the extracted slice, our FAR network can reliably perform detection and grading.

3.2. Implementation details.

For training, all the input selected slices to the FAR network are directly resized to size 512×512 without manual cropping, and the batch size is set to be 2 as mentioned in sub-section *2.1.1*. The loss weights are set as discussed in sub-section *2.1.5* and *2.3*. The codes of the FAR network are implemented in Python 3.6 on Tensorflow 1.2. The training configurations of the multi-task recognition framework and the discriminative network are listed in Table.1. The gradient is clipped so that its maximum L2-norm is 5 to avoid gradient exploding and accelerate the convergence of our FAR network. The parameters of FEN and RPN are initialized using the pretrained network on COCO dataset, while the other parameters are randomly initialized. Once all hyper-parameters are properly set, our FAR network does not need human intervention when processing MRI scans of different patients. The training is implemented on an NVIDIA GTX1080 GPU.

For evaluation and comparison, standard five-fold cross-validation is employed. Since the number of the dataset is limited, we did not divide it into training, validation and testing sets directly. Following the criterion of five-fold cross-validation, we randomly partitioned the original dataset into five equal size sub-datasets. Of the five sub-datasets, a single sub-dataset is retained as the testing data unseen to the network, and the remaining four sub-datasets are used as training data. During the cross-validation process, we choose the same training parameters to train the models, save the network training checkpoints, and evaluate the performance. The five results from the folds can then be averaged to produce a single result. This method can ensure that all images are used for both training and testing, and each image is used for testing exactly once.

3.3. Evaluation criteria.

Extensive experiments are conducted to validate the effectiveness of our FAR network from the following aspects. Since our FAR network is a multi-task detection grading network, we respectively evaluate FAR network from the aspect of grading and detection in all the following experiments.

3.3.1. Qualitative performance evaluation for FAR network

In order to visually demonstrate the accuracy and robustness to image characteristics, we choose images of different MRI modalities, vertebrae appearance, vertebrae numbers, image resolution, intensity distribution, and spondylolisthesis grading to visually evaluate the detection and grading results of our FAR network. The chosen images are from different folds in our five-fold cross-validation to verify the reproducibility of our FAR network when the training and testing data vary. The detection box and the ground truth box of the critical vertebrae are both demonstrated to prove the excellent vertebrae detection performance; the predicted gradings are also demonstrated to visualize the high grading performance based on the detected relative positions of the critical vertebrae.

3.3.2. Quantitative performance evaluation for the grading task

The grading accuracy is defined as the percentage of images whose grading and detection are correct. This is a rather strict metric because we find in some experiments that a small fraction of wrongly detected images could have a correct grading result, however, we treat these images as wrongly graded because the correct grading may be acquired by coincidence.

We also calculate the confusion matrix for a more detailed analysis of the spondylolisthesis grading performance. The confusion matrix is an N×N matrix, where N represents the total number of gradings (3 in our work). The element at the i^{th} row and j^{th} column of the confusion matrix means the number of images whose predicted grading is i and ground truth grading is j. It is thus easy to understand that the sum of the diagonal elements in the confusion matrix is the number of correctly graded images. Besides, we add a new row to demonstrate the detection results, each element in this row means the number of wrongly detected images for each grading.

3.3.3. Quantitative performance evaluation for the detection task

The detection accuracy of our FAR network is mainly evaluated by mAP_{75}, which is a widely used evaluation metric in the object detection domain. mAP_{75} is a comprehensive metric that considers the precision, recall as well as the IoU (Intersection-over-union) with the ground truth boxes of an object detector. Many state-of-the-art object detection networks such as Faster RCNN (Ren et al., 2015), Mask RCNN (He et al., 2017), and YOLO (Redmon et al., 2016) use this metric to evaluate the performance of their networks. Below, we give a brief introduction of mAP_{75} to demonstrate how this metric measures precision, recall as well as the IoU simultaneously in one single metric.

The metric mAP_{75} is calculated by averaging the AP_{75} (Average Precision at IoU threshold 0.75) among different classes. AP_{75} is acquired by summarizing

the shape of the precision-recall curve (Everingham et al., 2010) for objects of each given class. As discussed in sub-section *2.1.4*, each detection result is represented by a set of detection box coordinates, a confidence score (which is obtained by feeding the DET_{CL}^{pred} to the softmax operation to obtain class probability vector, and then choosing the largest element of the class probability vector), and a predicted class label (the index where the class probability vector reaches its maximum) in our object detection task. Having this in mind, the AP_{75} for each class is calculated as follows:

First, all detection boxes having this class are sorted by their confidence scores. For the i^{th} point in the precision-recall curve, the first i detection boxes in the sorted box series (with i highest confidence scores) are chosen to be predicted positives. Then, for each detection box, if it has an IoU larger than 0.75 with the ground truth box and the predicted class label is correct, this detection is considered to be a true positive based on our selected IoU threshold (0.75). Otherwise, this detection is considered to be a false positive. Then, the precision and recall are calculated by:

$$\begin{aligned} precision &= \frac{true\ positive}{true\ positive\ +\ false\ positive} \\ recall &= \frac{true\ positive}{true\ positive\ +\ false\ negative} = \frac{true\ positive}{DET^{GT}\ positive\ objects} \end{aligned} \tag{13}$$

In this way, the i^{th} point in the precision-recall curve is determined. As i increases, more detection boxes are considered to be positive. If the newly chosen box is true positive, the recall and the precision will both increase (the precision will keep the same if it is 1); if the newly chosen box is false positive, the recall will remain the same and the precision will decrease. Thus, when i is gradually increased to reach some value, the recall will eventually reach 1. At this time, we can obtain the whole precision-recall curve. Then, the AP_{75} value is calculated by averaging the precision at all values of recall (which can be understood as the approximated area-under-curve of the precision-recall curve). This calculation has many variations such as "11-point interpolation" and "interpolating all points", however, this does not make much difference in our work. After calculating the AP_{75} for all classes, the AP_{75}'s of different classes are averaged to obtain mAP_{75}.

One advantage of mAP_{75} is that it penalizes methods which retrieve only a subset of true positive objects with high precision (i.e., the method omits some of the objects in the image) (Everingham et al., 2010).

The IoU threshold can be adjusted to obtain AP_M (and mAP_M), where M is the IoU threshold. The larger M, the more difficult it is for a predicted detection box to be true positive because it has to be highly overlapped with the ground truth box. In our work, besides mAP_{75}, we calculated mAP_M of different IoU thresholds from 0.5 to 0.95, which reveals the variation of mAP_M when the IoU threshold become becomes stricter and stricter. If the mAP_M is high at large M, we know that our method has very few false positives and false

negatives, and the detection boxes have high IoU's with their ground truths. This is required in our multi-task detection-grading work because the grading performance is significantly affected by the detection results.

3.3.4. Ablation experiments of FAR network

Ablation experiments following the same five-fold cross-validation protocol are carried out to respectively prove the necessity of the hybrid supervision and the adversarial module. First, the auxiliary branch in the hybrid supervision is removed (annotated as "without hybrid supervision") as a comparison experiment to demonstrate the importance of the auxiliary gradient back-propagation pathway. Second, the adversarial module is removed (annotated as "without adversarial module") to prove the strengths of the adversarial ability to leverage the higher-order statistics. Third, both the auxiliary branch and the adversarial module are removed (annotated as "only MDGM") for proving the abilities of the multi-task module, and also the necessity of the integration of proposed modules. Also, inner-comparison experiments are carried out to show that the FAR network is robust to changes of hyper-parameters such as anchor scales, anchor aspect ratio, and Resnet structure.

3.3.5. Inter-comparison experiments

Inter-comparison experiments concerning Resnet and four other popular networks (VGG-19, VGG-19-FCN, GoogLeNet-FCN, and Densenet) are conducted to demonstrate the strengths of the FAR network. These experiments are designed as two parts: (1) The first part is using these baseline networks to directly perform grading as a classification task. (2) The second part is using these baseline networks as feature extraction network (FEN) for the multi-task recognition network in Fig.2. This design on one hand demonstrates the effect of our multi-task detection-grading workflow, and on the other hand compares different FEN's for feature extraction.

Direct grading workflow. Five popular networks (VGG-19, VGG-19-FCN, GoogLeNet-FCN, Resnet, and Densenet) are used to directly perform spondylolisthesis grading without detection with the same five-fold cross-validation protocol. VGG-19 (Simonyan and Zisserman, 2014) is a network designed for large-scale image recognition using 16 convolution layers and 3 fully connected layers. VGG-19-FCN and GoogLeNet-FCN are both from the fully convolutional network (FCN) (Shelhamer et al., 2016) for natural image classification. VGG-19-FCN is composed by changing the last 3 fully connected layers of VGG-19 into 1×1 convolutional layers. GoogLeNet-FCN, as shown in Table 1 of (Szegedy et al., 2015), is a fully convolutional network of 5 stages, 22 layers with inception modules for performance enhancement. The Resnet backbone in FEN of our work is also used to directly perform spondylolisthesis grading without detection. Lastly, Densenet, which is one of the most state-of-the-art networks, is used for direct grading task. We use architecture DenseNet-121 in Table 1 in (Huang et al., 2017) for this task. All networks are implemented by ourselves on Tensorflow using the same training batch size. The grading accuracies are examined when the training steps reach 1000, 10000, and 11000

(which are the grading task training steps, the detection training steps, and the detection training steps plus the grading training steps in the FAR network) using the same metric as FAR network. The best one of the three accuracies is used to compare with the FAR network.

Multi-task detection-grading workflow using different FEN. This experiment series uses all above-mentioned networks as feature extraction network (FEN) in our multi-task detection-grading workflow. The subsequent procedures (top-down layers, regional proposal network, multi-task detection grading network, auxiliary supervision branch, and the adversarial module) are the same as those in the FAR network for a fair comparison, as shown in Fig.3. Similar to the FAR network, necessary modifications are applied to the baseline networks for feature extraction because the baseline networks are originally designed for classification. The shapes of the intermediate outputs of these baseline networks are modified to produce tensors with suitable shapes for the hierarchical features (that is, 128×128×256 for the first tensor, 64×64×512 for the second, 32×32×1024 for the third and 16×16×2048 for the forth), and these tensors are used as C2∼C5 in Fig.4. The modifications for different baseline networks are as follows:

- For VGG-19, we adjust the output widths, heights, and channels of the $2 \sim 5^{th}$ convolutional stacks (there are altogether 5 stacks in VGG-19, as shown in Table 1 in (Simonyan and Zisserman, 2014)) to acquire tensors of the above-mentioned shapes and use them as C2∼C5. This is achieved by adding/removing the pooling layers (to adjust the widths and heights) and adding 1×1 convolutional layer (to adjust the channels) after some convolutional stacks.

- For VGG-19-FCN, C2∼C4 are the same as those in VGG-19. Then, we convert the last three fully connected layers to 1×1 convolutional layers and adjust the output shape of the last 1×1 convolutional layer to be 16×16×2048. This output is used as C5.

- For GoogLeNet-FCN, the outputs of the second pooling layers and those of inception 3b∼5b in Table 1 of the original GoogLeNet paper (Szegedy et al., 2015) are used as C2∼C5. The widths, heights, and channels of these tensors are adjusted to fit our requirements in the same way (adding/removing the pooling layers and adding 1×1 convolutional layers) as in experiments with VGG-19.

- For Densenet, based on the DenseNet-121 architecture, the output of each transition layer succeeding one dense block is used as the hierarchical features C2∼C5. The hyperparameters "growth rate" and "layers in each block" are carefully adjusted to guarantee the features has the same shape with those of FAR network. We tried several combinations of these hyperparameters for the best performance.

3.3.6. Comparison of detecting all vertebrates separately and detecting all vertebrates in a single bounding box

In our FAR network, all critical vertebrae are detected separately. However, our ultimate objective is spondylolisthesis grading. Thus, we also carry out experiments to compare the grading performance of detecting all vertebrates separately and detecting all vertebrates in a single bounding box. In experiments detecting all vertebrates in a single bounding box, the upper-left corner point of the L4 bounding box, and the lower-right corner of the S1 bounding box are used to form a new bounding box. This new bounding box is used as the ground truth of the detection task. Intuitively, the predicted detection box for each image should be a single bounding box containing all three critical vertebrae. This bounding box is then used to perform the grading task instead of the three separate bounding boxes used in our FAR network. The grading results are recorded and compared with those of our FAR network.

4. Results and Discussion

4.1. Comprehensive analysis.

4.1.1. Qualitative evaluation results of FAR network

Fig.9 demonstrates that FAR network has simultaneously achieved accurate spondylolisthesis grading and excellent critical vertebrae detection. The images shown in Fig.9 are of different modalities, vertebrae appearance, vertebrae numbers, image resolution, and intensity distribution from different folds in our five-fold cross-validation. The high overlap of the detected boxes (dashed boxes) with their ground truth (solid boxes) in Fig.9 shows that our FAR network is robust to changes of image characteristics. Its performance is reproducible when the training and testing data varies. Also, the spondylolisthesis grades are different in these images, which means that the detection performance of our FAR network is not affected by spondylolisthesis grades. The grading results labeled in the figures are correct, which demonstrates high grading performance based on the detected relative positions of the critical vertebrae.

4.1.2. Quantitative detection and grading results of FAR network

General accuracies for detection and grading task. For quantitative analysis, our FAR network is able to correctly detect all critical vertebrae in all of the training images and 96% of the testing images, i.e., on average, ∼28.8 out of 30 testing images are correctly detected in each fold of the five-fold cross-validation. Based on these detection results, the overall spondylolisthesis grading accuracy is 0.9883±0.0094 for the training dataset and 0.8933±0.0276 for the testing dataset. This means that: (1) On average, the FAR network is able to correctly perform spondylolisthesis grading in ∼118.6 out of the 120 training images, while it is able to correctly grade ∼26.8 out of the 30 testing images in each fold of the five-fold cross-validation. (2) The grading network is properly trained based on perfect detection results in the training dataset, which ensures that all critical vertebrae are detected and used for the grading

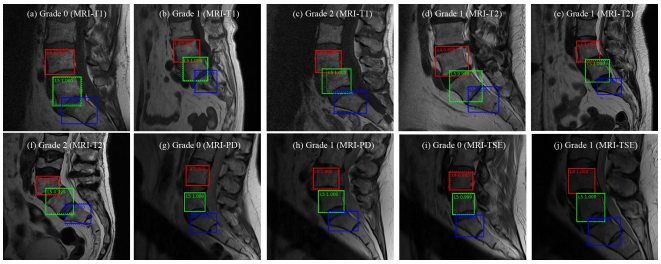

Figure 9: Qualitative visualization shows that our FAR network can accurately perform spondylolisthesis grading and critical vertebrae detection in different modality MRI images. Fig.9(a)∼(c) are MRI-T1 images with grading 0∼2; (d)∼(f) are MRI-T2 images with grading 0∼2; (g)∼(h) are MRI-PD images with grading 0∼1; (i)∼(j) are MRI-TSE images with grading 0∼1. The dotted boxes are the detection boxes with confidence scores, and the solid boxes are ground truth boxes. Although the area, appearance, resolution, images intensity distribution and relative positions of the critical vertebrae are different, our network achieves high accuracy.

task during training; neither multiple detection nor missing detection happens to perturb the training of the grading network in MDGM.

The confusion matrix for grading task. The confusion matrix (shown in Table.2) is calculated for a more detailed evaluation of the grading performance. As mentioned in Section *3.3.2*, each row in the confusion matrix indicates the predicted gradings while each column indicates the ground truth grades. For example, the first row of the training confusion matrix means that the FAR network predicts 291 instances to be grading 0, in which 287 of them are correct and 4 of them are actually grading 1. The confusion matrix counts up the detection results in the five-fold cross-validation: since we have respectively 72, 56, and 22 patients of grade 0, 1, and 2 in the dataset, and that each patient are used 4 times for training and 1 time for testing in the five-fold cross-validation, there should be respectively 288, 224, and 88 instances in training and 72, 56, and 22 instances in testing. It can be seen that 593 out of 600 training instances and 134 out of 150 testing instances are correctly detected and graded; all training instances and 144 out of 150 testing instances are correctly detected. For the wrongly processed instances in testing, 3 instances with grade 0 and 3 instances with grade 2 are wrongly detected, 5 instances with grade 1 are regarded as grade 0, 3 instances with grade 0 are regarded as grade 1, and 2 instances with grade 2 are regarded as grade 1. None of the predicted grades differs more than one from the ground truth.

It should be noted that although there are more wrong detections in images of grade 0 and 2 than grade 1 (the last row in Table.2), the detection result is actually not relevant to the spondylolisthesis grading. Instead, the difference in detection accuracy of different grades might be caused by the competition across classes when calculating DET_{CL}^{pred} (He et al., 2017). In the class competition, even if the probability of the correct class is just slightly smaller than that of the wrong class, the detection would be wrong. The class competition is a natural

Table 2: Confusion matrix spondylolisthesis grading estimated by our method compared to the ground truth for each patient. The horizontal "Grade 0∼2 means the ground truth grading, and the vertical "Grade 0∼2 means the predicted grading.

(training)	Grade 0	Grade 1	Grade 2
Grade 0	287	4	0
Grade 1	1	219	1
Grade 2	0	1	87
wrong detection	0	0	0

(testing)	Grade 0	Grade 1	Grade 2
Grade 0	66	5	0
Grade 1	3	51	2
Grade 2	0	0	17
wrong detection	3	0	3

property of the solution of classification tasks using CNN's, which is not affected by the spondylolisthesis grading.

Grading performance comparison with inter-operator variability. As mentioned in section *3.1*, we have two manual gradings performed by different physicians (denoted as P1 and P2). We count up the number of images where the two gradings performed by P1 and P2 are the same. The ratio of this summation to the total number of images is regarded as the accuracy for inter-operator variability, which is calculated to be 0.9200, i.e., 138 out of 150 manual gradings are the same with each other. It is thus shown that the accuracy of our method (0.8933, i.e., 134 out of 150 gradings are correct) is comparable with physician work. The wrong gradings of our FAR network most happen when the ground truth forward displacement measurement (the length of the orange arrow in Fig.1) falls at ranges close to critical points of adjacent gradings (forward translation percentage is less than 10% or in 25%±5%). In practice, this case accounts for 70% of the wrong grading (correctly detected but wrongly graded images) in testing.

The mAP_{75} and mAP_M at different threshold for detection performance. The high spondylolisthesis grading performance can be mainly attributed to the excellent detection accuracy and precise vertebrae bounding boxes of the multi-task recognition network. The mAP_{75} of our FAR network for the training/testing dataset are respectively 1±0 and 0.9636±0.0180, which means that the detection network has a good performance: the three critical vertebrae are detected and correctly classified in almost all images, and the detected vertebrae overlap very well with the ground truth (IoU$\geqslant$0.75). For a more generalized evaluation, we also calculate the mAP_M (mAP at different IoU thresholds) and plot the mAP-IoU curve. The calculated mAP-IoU curve as a function of IoU is shown in the left figure in Fig.10) for a more detailed evaluation of the detection performance. This curve shows that the mAP is still high ($\geqslant$0.9) when the IoU threshold is 0.9/0.8 in the training/testing dataset,

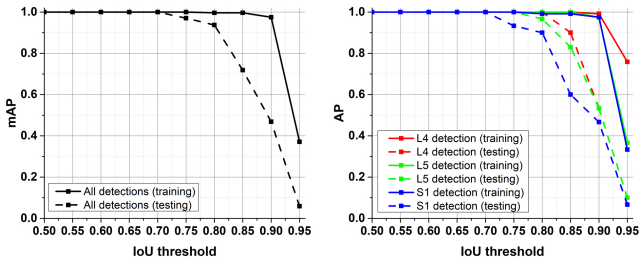

Figure 10: Average Precision (AP) for different classes (right) and mean Average Precision (mAP) for all classes (left) at different IoU thresholds calculated using the training (solid lines) and testing (dashed lines) dataset. The mAP value is averaged throughout each instance in the dataset. The mAP is still high when the IoU threshold is as high as 0.8, which means that our method not only detects all the critical vertebrae but also achieved high accuracy in the bounding box regression (IoU with ground truth boxes are high).

which means that almost all detections have an IoU higher than 0.9 with their ground truth in the training dataset, and above 90% of the detections have an IoU higher than 0.8 with their ground truth in the testing dataset. Moreover, as shown in the right figure in Fig.10), the mAP-IoU curves for all critical vertebrae are similar, which means that the performance of the detection task in our FAR network is insensitive to different classes.

Our FAR network can accurately and fully-automatically perform detection and grading, which is beneficial for clinical treatment processes such as therapeutic schedules and surgery plans. For example, the FAR network automatically presents that grade 2 spondylolisthesis happens between L4 and L5 vertebrae in Fig.9(c). In term of treatment planning, our FAR network indicates that physical therapy can be applied to this patient to reduce the amount of pressure the spine. However, if the patient feels that the condition is worsening, which implicates a risk of transition to higher grades, one should try bracing to help immobilize the spine. Moreover, the detection result also shows that the intervertebral disc between L4 and L5 may be compressed by the vertebrae, and there may be a risk of harming the disc and develop bulging disc or slipped disc in the future. In all, although the variability of image characteristics in multi-modality MRI images leads to unusual difficulties, the FAR network is able to acquire accurate detection and grading results.

4.2. Ablation experiments.

As shown in Table.3, the hybrid supervision and adversarial module contribute to a superior performance of detection and grading accuracy. As a baseline, FAR network on average achieves 0.8933±0.0276 testing grading accuracy and 0.9636±0.0180 testing mAP_{75} (first row in Table.3). (1) After only preserving the multi-task detection grading module (MDGM), the testing grading accuracy decreases to 0.8067±0.0398 and testing mAP_{75} decreases to 0.9364±0.0385 (fourth row in Table.3). This not only demonstrates the effectiveness of hybrid

Table 3: Ablation experiments demonstrating the effect of PRL and GAN. It can be seen that each component in FAR network efficiently improved the detection accuracy.

Row No.	Settings	test grading accuracy	test detection accuracy (mAP_{75})
1	our FAR network	0.8933±0.0276	0.9636±0.0180
2	without hybrid supervision	0.8200±0.0348	0.9378±0.0326
3	without adversarial module	0.8533±0.0315	0.9550±0.0278
4	only MDGM	0.8067±0.0398	0.9364±0.0385

supervision and adversarial module, but also proves that the multi-task detection grading module is capable of extracting correct features and finding out the fine-grained detailed difference between similar appearing vertebrae, which contributes to correctly classifying the critical vertebrae. (2)If the hybrid supervision is removed, the testing grading accuracy decreases to 0.8200±0.0348 and testing mAP_{75} decreases to 0.9378±0.0326 (second row in Table.3), which demonstrates the hybrid supervision can significantly correct detection errors by removing false positives and promote correct grading by yielding better bounding box positions. (3)If the adversarial module is removed (third row in Table. 3), the testing grading accuracy decreases to 0.8533±0.0315 and testing mAP_{75} decreases to 0.9550±0.0278, which demonstrates the adversarial plays a role in refining the predicted vertebrae coordinates probably by examining their internal higher-order relationships. Since FAR network achieves higher detection and grading performance than its ablated versions, the combination of the hybrid supervision and adversarial module makes the FAR network a more efficient and reliable resolution for critical vertebrae detection and spondylolisthesis grading.

4.3. Hyper-parameter changes

As shown in Table.4, the detection and grading performance is investigated with different settings of anchors. By default we use anchors of 5 scales and 3 aspect ratios; the grading accuracy and detection mAP_{75} are listed in the 1^{st} row (which is the same with the 1^{st} row in Table.3). Then, scales 8 and 128 are removed in sequence, as shown in the $2 \sim 3^{rd}$ rows. Compared with the default experiment, it is found that both grading accuracy and detection mAP_{75} drop only by an inconspicuous margin (∼0.66% drop in testing grading accuracy, which corresponds to ∼0.18 instance in each fold in the five-fold cross-validation; and ∼0.51% drop in testing mAP_{75}). Next, we use only 1 aspect ratio (1:1) to re-implement the above experiments, and the results are listed in the $4 \sim 6^{th}$ rows of Table.4. The results are almost as good as using 3 aspect ratios (the testing grading accuracies and mAP_{75}'s in the $4 \sim 6^{th}$ rows are similar to those in the $1 \sim 3^{rd}$ rows). These results show that the FAR network is robust to changes of anchor settings. This robustness may be because the size of the critical vertebrae is mostly 16∼64, and that different critical vertebrae have aspect ratios close to 1:1 (although they may appear differently). Moreover, the hybrid supervision manner may also contribute to this robustness by training

Table 4: Detection results using different settings of anchors. It is seen that when the anchor scales and ratios are reduced, the mAP of our proposed FAR network only drops by a inconspicuous margin, namely, our network is robust to changes of anchor settings.

Row No.	Settings	Anchor scales	Aspect ratios	test grading accuracy	test detection accuracy (mAP_{75})
1	5 scales, 3 ratios	8, 16, 32, 64, 128	2:1, 1:1, 1:2	0.8933±0.0276	0.9636±0.0180
2	4 scales, 3 ratios	16, 32, 64, 128	2:1, 1:1, 1:2	0.8933±0.0144	0.9602±0.0195
3	3 scales, 3 ratios	16, 32, 64	2:1, 1:1, 1:2	0.8867±0.0291	0.9585±0.0182
4	5 scales, 1 ratios	8, 16, 32, 64, 128	1:1	0.8667±0.0298	0.9590±0.0186
5	4 scales, 1 ratios	16, 32, 64, 128	1:1	0.8533±0.0292	0.9548±0.0207
6	3 scales, 1 ratios	16, 32, 64	1:1	0.8533±0.0325	0.9542±0.0191

Table 5: The effect of number of layers in Resnet. FAR network is robust to changes in the number of layers in the feature extracting network within a certain scope.

Row No.	Settings of (FEN backbone)	test grading accuracy	test detection accuracy (mAP_{75})
1	Resnet-101	0.8933±0.0276	0.9636±0.0180
2	Resnet-50	0.8600±0.0301	0.9645±0.0246

the RPN^{pred}_{BBC}'s and DET^{pred}_{BBC}'s in a unified network, which provides more appropriate proposal positions so that using linear correction is more appropriate. In this way, even if there are fewer anchors (which means that the positive anchors may be somewhat farther to the ground truth), a good performance could still be achieved.

We also change the number of layers in Resnet (which are used for feature extraction in FEN) following (He et al., 2017). The famous Resnet structure Resnet-50 and Resnet-101 are compared. The results listed in Table.5 shows that changing Resnet-101 to Resnet-50 does not significantly decrease testing grading accuracy or mAP_{75}. The FAR network is robust to changes in the number of layers in the feature extracting network within a relatively wide scope, which means that the hierarchical features are correctly extracted in FEN.

4.4. Inter-comparison.

As mentioned in Section *3.3.6*, five popular networks (i.e., VGG-19, VGG-19-FCN, GoogLeNet-FCN, Resnet, and Densenet) are used to perform spondylolisthesis grading in two workflows (i.e., multi-task detection and grading workflow and direct grading workflow without detection). The results in Table.6 show that: (1) our proposed multi-task detection and grading workflow is beneficial

Table 6: Comparison with the state-of-the-art. The annotation direct means direct grading, whereas the annotation multi-task means multi-task detection and grading.

Row No.	Settings	test grading accuracy	test detection accuracy (mAP_{75})
1	our method (Resnet-multi-task)	0.8933±0.0276	0.9636±0.0180
2	Resnet-direct	0.6933±0.0562	–
3	VGG-19-multi-task	0.8600±0.0356	0.9377±0.0384
4	VGG-19-direct	0.7600±0.0471	–
5	VGG-19-FCN-multi-task	0.8667±0.0392	0.9442±0.0367
6	VGG-19-FCN-direct	0.7800±0.0446	–
7	GoogLeNet-FCN-multi-task	0.5000±0.0898	0.7555±0.0786
8	GoogLeNet-FCN-direct	0.6733±0.0955	–
9	Densenet-multi-task	0.7933±0.0760	0.8981±0.0645
10	Densenet-direct	0.7000±0.0882	–

to the grading accuracy. (2) Our FAR network, i.e., multi-task detection and grading workflow with Resnet as FEN, performs the best among the baseline networks. Comparing with the state-of-the-art classification networks, FAR network shows significant advantages by an average of ∼15% grading accuracy.

4.4.1. Comparison with direct grading workflow

The $1, 3, 5, 7, 9^{th}$ rows in Table.6 are the experiments using the multi-task detection and grading workflow with different FEN's, while the $2, 4, 6, 8, 10^{th}$ rows are the experiments using the direct grading workflow with the corresponding FEN's. Generally, the multi-task workflow outperforms the direct grading workflow, as long as the detection accuracy is acceptable. Detailed discussions are as follows:

Our FAR network (multi-task workflow with Resnet as FEN) versus Resnet for direct grading. The first and second rows in Table.6 reveals the advantages of the multi-task workflow in our FAR network compared to the "Resnet-direct" method. Although using Resnet to directly perform spondylolisthesis grading as a classification work has a training accuracy as high as our FAR network (0.9917 compared with 0.9883, which are not shown in Table.6 for concision and readability), the testing accuracy is far lower (0.6933 compared with 0.8933). This indicates that the "Resnet-direct" model may be over-fitting when directly used to predict spondylolisthesis grading. The over-fitting may be due to the redundant information provided by the image, which is eliminated by the detection work by revealing the coordinates (and the relative locations) of the critical vertebrae. The most important decisive factors of spondylolisthesis are learned through the detection task, and the nonsignificant factors are wiped off to guarantee a higher grading performance. Actually, the RPN and MDGM have much fewer parameters than the FEN (RPN has 3 convolutional layers; MDGM has 9 convolutional layers, 4 for detection task and 5 for grading task), but they show a significant effect on grading performance. The multi-task

workflow in our FAR network reinforces the mutual benefit between detection and grading for superior performance.

Other networks for multi-task workflow versus direct grading. (1) For VGG-19 and VGG-19-FCN, the multi-task detection and grading experiments also show better grading performances than the direct grading experiments (the accuracy is 10% higher in the multi-task workflow). This again demonstrates the advantages of our designed MDGM, hybrid supervision, as well as the adversarial module. VGG-19 and VGG-19-FCN seem to be less over-fitted in the direct grading workflow (nevertheless, over-fitting still exists) probably because they contain fewer convolutional layers. (2) For GoogLeNet-FCN, the multi-task detection and grading experiments show even worse grading performance than direct grading. This may be due to the GoogLeNet network is not suitable for correctly detecting the critical vertebrae (as shown in the Table.6, the detection performance measured by mAP_{75} is not satisfactory; actually, in nearly 40% of the images, not all three vertebrae are correctly detected). The incorrect detections disturb the grading procedure and result in a worse grading accuracy. (3) For DenseNet, the multi-task detection and grading experiments show slightly better grading performance than direct grading, but the performances of both two workflows are lower than those of Resnet and VGG-19/VGG-19-FCN.

4.4.2. Comparison of different FEN's in the multi-task detection-grading workflow

Among the networks used in the multi-task detection-grading workflow, we find that the Resnet performs the best; VGG-19/VGG-19-FCN also show acceptable performance. The VGG-19/VGG-19-FCN have relatively large model capacity, so they produce generally good results; however, the deeper network and the shortcut connections in Resnet can extract more distinguishable features for different vertebrae and further improve the performance. VGG-19 and VGG-19-FCN show similar performance in the multi-task workflow, probably because the added three 1×1 convolutional layers do not make significant changes in the whole workflow containing FEN, RPN, MDGM, and the adversarial module.

To our surprise, we find that both detection performance and grading accuracy of GoogLeNet and Densenet are lower than those of Resnet and VGG-19/ VGG-19-FCN. Experiments in (Shelhamer et al., 2016) have also reported similar results, i.e., the performance of GoogLeNet does not match that of VGG-16. In order to explore the reasons for these results, we further analyze the structures of the convolutional networks. We find that: (1) VGG is a cascade network where the input of each layer is the output of the preceding layer; Resnet is a similar cascade structure except for some shortcut connections summing up the input and output of several convolutional layers. (2) Both GoogLeNet and Densenet contain parallel architectures, i.e., the same inputs are processed by different convolutional layers, and then the results are concatenated in the *channel* dimension. For example, in the inception unit in GoogLeNet, the same inputs are processed by four parallel branches (respectively 1×1 convolutional layer, 1×1 convolutional layer and 3×3 convolutional layer, 1×1 convolutional

layer and 5×5 convolutional layer, and Max Pooling layer and 1×1 convolutional layer), then the four results are concatenated in the *channel* dimension. Similarly, the input and output of the composite functions in the dense blocks are concatenated in Densenet (Huang et al., 2017). These concatenation operations make the network wider (Szegedy et al., 2015), but they may to some extent weaken the effect of the convolutional layers because only a proportion of feature maps after the convolutional layers are added to the "collective knowledge" of the network, while the remaining feature maps are kept unchanged (Huang et al., 2017). Since a proportion of the channels in the output feature maps are simply duplicated from the input (or only processed by a simple combination of Max Pooling layer and 1×1 convolutional layer), the effect of the convolutional layers in GoogLeNet and Densenet is weakened. Actually, for Densenet, we find that the performances are better when we use fewer "layers in each block" and larger "growth rate". This means that when we use fewer dense connections (and concatenation operations), the outputs of each layer are less affected by its input, and the results are better. Thus, we infer that the inappropriate concatenations in the parallel convolutional architecture are harmful to the detection/grading performance.

4.4.3. Comparison of detecting all vertebrates separately and detecting all vertebrates in a single bounding box

As shown in Table.7, compared to detecting all vertebrates separately, detecting all vertebrates together in a single bounding box for all vertebra results in lower grading performance and similar detection performance. This may be due to:

- For the grading task, detecting all vertebrae separately can acquire the bounding box coordinates of each critical vertebra. Since the grading is related to the relative positions of the vertebra, revealing their bounding box coordinates helps promote the grading performance. On the contrary, the single bounding box cannot reveal the relative positions of the critical vertebrae, which is unbeneficial to the grading task.
- For the detection task, the image characteristics are not affected by the detection target, and the difficulty of separate detection and single detection are almost the same. Thus, the detection performances of the two experiments are similar.

4.5. Using our FAR network to directly process 3D MRI scan.

Up to this point, the discussion of our FAR network is on processing 2D MRI slice. Now, we would like to demonstrate that our FAR network is able to be extended to directly process 3D lumbar scans. Since the grading task is clinically performed using 2D slice, processing spondylolisthesis grading using 3D MRI scan may lack clinical background support. Thus, we mainly discuss the detection task in this section. From the aspect of methodology, vertebrae

Table 7: Comparison of detecting all vertebrates separately and detecting in a single bounding box.

Row No.	Settings	test grading accuracy	test detection accuracy (mAP_{75})
1	Detecting each vertebra separately	0.8933±0.0276	0.9636±0.0180
2	Detecting all vertebra together	0.8333±0.0323	0.9582±0.0184

detection from 3D MRI scans can be broadly divided into two classes: (1) directly using 3D CNN's; (2) using 2D CNN's with some additional machine learning methods. We give a brief discussion of the two methods as follows:

Directly using 3D CNNs is a straightforward method of processing 3D MRI scans. 3D CNN is also a novel concept for 3D object detection that can leverage the spatial relevance of different sagittal slices in the 3D input. Also, they are not difficult to implement based on state-of-the-art deep learning architectures such as Tensorflow.

For our FAR network, replacing the 2D CNN's with 3D CNN's may be a method for directly using 3D CNN's for vertebrae detection. However, the computational cost may be a problem for directly training 3D CNN's. Although some methods have been proposed to deal with this problem, they may still face class imbalance problem and/or complicated pre-processing/post-processing procedures. For example, (Liao et al., 2018) uses image samples (cropped images) to train 3D CNN's to detect vertebrae. After the training completes, the trained 3D CNN network is converted to 3D FCN by reshaping the weight matrices into 1×1×1 convolutional layers. The entire 3D image is fed to the FCN during testing. This method can efficiently reduce computation, but it still faces the problems of losing the global information of the input 3D image. Even though samples appropriate for representing vertebrae are correctly cropped, it is hard for the network to distinguish the vertebrae without this global information because different vertebrae look similar. Sorting the positive samples by coordinates and feeding them into an LSTM can mitigate this problem, however, in the testing phase, once there exist false positives, the sorting procedure would be incorrect and the performance of LSTM is ruined. Also, the selection of samples also requires a lot of considerations and/or pre-processing. For example, if samples are randomly chosen, one should consider how to deal with the class imbalance problem (the negative samples would be much more than positive ones) and the portion cropping problem (whether to consider a cropped sample that contains only a section of a vertebra rather than the whole vertebra as a positive). Thus, we think that 3D CNN's are not necessarily needed in our work.

Using the extension of 2D CNN's to process 3D lumber scans is easy to implement. It does not require additional computational cost by introducing 3D CNN's. Also, since our FAR network is designed for 2D MRI images,

the detection and grading task for 3D MRI scans can be performed with the existing FAR network slice by slice. However, this scenario ignores the spatial relationship of different slices, which may affect detection performance.

One remedial measure of leveraging the spatial relationship is to introduce additional machine learning methods. After feeding the single slices to the FAR network, simple and effective machine learning methods such as voting, random forests, LSTMs can be used to integrate the detection results of single slices. In this way, the spatial relevance of different sagittal slices is leveraged without introducing too much computation.

Actually, this method is to some extent similar to that proposed by Liao. Liao's method (Liao et al., 2018) uses 3D CNN's for detecting single vertebra at some given locations (i.e., the cropped images) from different slices; and then integrate the information of different locations in one slice using LSTM (or other machine learning methods). Our method, as a contrast, uses 2D CNN's for detecting all vertebra in a given slice; and then integrate the information of different slices using LSTM (or other machine learning methods). Both methods first detect vertebrae from some dimensions using CNN's, then uses other methods to integrate the information from the other dimension. The difference is that the sequential order of the processing dimensions is different.

In summary, our FAR network is able to be directly extended to 3D lumbar scan without many modifications, and has strong application ability in computer-aided diagnosis and treatment plan of spondylolisthesis.

5. Conclusion

In this paper, we develop a faster adversarial recognition (FAR) network to detect critical vertebrae and perform spondylolisthesis grading from MRI images across multiple modalities. FAR network is trained in an adversarial scheme: the generator is a multi-task recognition network that performs high quality (large IoU with ground truth and no false-positives) detection and accurate grading with the help of an auxiliary gradient back-propagation pathway in hybrid supervision manner; and the discriminator implicitly leverages the high-order statistics of the detection coordinates to supervise the generative network and refine the detections. The experimental results demonstrate the effectiveness of FAR network in detecting critical vertebrae as well as performing spondylolisthesis grading from MRI images with different image characteristics.

Conflict of Interest

None

References

References

Ben-Ari, R., Akselrod-Ballin, A., Karlinsky, L., Hashoul, S., 2017. Domain specific convolutional neural nets for detection of architectural distortion in mammograms. In: Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium on. IEEE, pp. 552–556.

Dai, J., He, K., Sun, J., 2016. Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3150–3158.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. International journal of computer vision 88 (2), 303–338.

Gao, Z., Li, Y., Sun, Y., Yang, J., Xiong, H., Zhang, H., Liu, X., Wu, W., Liang, D., Li, S., 2017a. Motion tracking of the carotid artery wall from ultrasound image sequences: a nonlinear state-space approach. IEEE transactions on medical imaging 37 (1), 273–283.

Gao, Z., Xiong, H., Liu, X., Zhang, H., Ghista, D., Wu, W., Li, S., 2017b. Robust estimation of carotid artery wall motion using the elasticity-based state-space approach. Medical image analysis 37, 1–21.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680.

Han, Z., Wei, B., Leung, S., Nachum, I. B., Laidley, D., Li, S., 2018. Automated pathogenesis-based diagnosis of lumbar neural foraminal stenosis via deep multiscale multitask learning. Neuroinformatics, 1–13.

Hartvigsen, J., Hancock, M. J., Kongsted, A., Louw, Q., Ferreira, M. L., Genevay, S., Hoy, D., Karppinen, J., Pransky, G., Sieper, J., et al., 2018. What low back pain is and why we need to pay attention. The Lancet.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask r-cnn. In: Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, pp. 2980–2988.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778.

Hresko, M. T., Labelle, H., Roussouly, P., Berthonnaud, E., 2007. Classification of high-grade spondylolistheses based on pelvic version and spine balance: possible rationale for reduction. Spine 32 (20), 2208–2213.

Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q., 2017. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

Jamaludin, A., Lootus, M., Kadir, T., Zisserman, A., Urban, J., Battié, M. C., Fairbank, J., McCall, I., Consortium, G., et al., 2017. Issls prize in bioengineering science 2017: automation of reading of radiological features from magnetic resonance images (mris) of the lumbar spine without human intervention is comparable with an expert radiologist. European Spine Journal 26 (5), 1374–1383.

Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., Tu, Z., 2015. Deeply-supervised nets. In: Artificial Intelligence and Statistics. pp. 562–570.

Li, J., Liang, X., Wei, Y., Xu, T., Feng, J., Yan, S., 2017. Perceptual generative adversarial networks for small object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1222–1230.

Liao, H., Mesfin, A., Luo, J., 2018. Joint vertebrae identification and localization in spinal ct images by combining short-and long-range contextual information. IEEE transactions on medical imaging 37 (5), 1266–1275.

Liao, S., Zhan, Y., Dong, Z., Yan, R., Gong, L., Zhou, X. S., Salganicoff, M., Fei, J., 2016. Automatic lumbar spondylolisthesis measurement in ct images. IEEE transactions on medical imaging 35 (7), 1658–1669.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2117–2125.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A. C., 2016. Ssd: Single shot multibox detector. In: European conference on computer vision. Springer, pp. 21–37.

Luc, P., Couprie, C., Chintala, S., Verbeek, J., 2016. Semantic segmentation using adversarial networks. arXiv preprint arXiv:1611.08408.

Mescheder, L., Geiger, A., Nowozin, S., 2018. Which training methods for gans do actually converge? In: International Conference on Machine Learning. pp. 3478–3487.

Möller, H., Sundin, A., Hedlund, R., 2000. Symptoms, signs, and functional disability in adult spondylolisthesis. Spine 25 (6), 683–690.

Niggemann, P., Kuchta, J., Grosskurth, D., Beyer, H., Hoeffer, J., Delank, K., 2012. Spondylolysis and isthmic spondylolisthesis: impact of vertebral hypoplasia on the use of the meyerding classification. The British journal of radiology 85 (1012), 358–362.

Passias, P. G., Poorman, C. E., Yang, S., Boniello, A. J., Jalai, C. M., Worley, N., Lafage, V., 2015. Surgical treatment strategies for high-grade spondylolisthesis: a systematic review. International journal of spine surgery 9, 50.

Radford, A., Metz, L., Chintala, S., 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H., 2016. Generative adversarial text to image synthesis. arXiv preprint arXiv:1605.05396.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X., 2016. Improved techniques for training gans. In: Advances in Neural Information Processing Systems. pp. 2234–2242.

Shelhamer, E., Long, J., Darrell, T., 2016. Fully convolutional networks for semantic segmentation. arXiv preprint arXiv:1605.06211.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9.

Tibrewal, S., Jayakumar, P., Vaidya, S., Ang, S. C., 2012. Role of mri in the diagnosis and management of patients with clinical scaphoid fracture. International orthopaedics 36 (1), 107–110.

Wollowick, A. L., Sarwahi, V., 2015. Spondylolisthesis: Diagnosis, Non-Surgical Management, and Surgical Techniques. Springer.

Wu, Y., He, K., 2018. Group normalization. arXiv preprint arXiv:1803.08494.

Zhao, R., Liao, W., Zou, B., Chen, Z., Li, S., 2019. Weakly-supervised simultaneous evidence identification and segmentation for automated glaucoma diagnosis.

Zhao, Z.-Q., Zheng, P., Xu, S.-t., Wu, X., 2018. Object detection with deep learning: A review. arXiv preprint arXiv:1807.05511.