Hindawi
Discrete Dynamics in Nature and Society
Volume 2022, Article ID 4494411, 12 pages
https://doi.org/10.1155/2022/4494411

Research Article

Cerebral Arterial Stenosis Detection Based on a Retained Two-Stage Detection Algorithm

Hanqing Liu,[1] **Xiaojun Li**,[1,2] **Jin Wei**,[3] **and Xiaodong Kang**[1]

[1]*School of Medical Technology, Tianjin Medical University, Tianjin 300202, China*
[2]*Chongqing Qianjiang Center Hospital, Chongqing 409000, China*
[3]*Tianjin Third Central Hospital, Tianjin 300171, China*

Correspondence should be addressed to Jin Wei; wj9717@sina.com and Xiaodong Kang; kxd2004@126.com

Received 28 January 2022; Accepted 17 March 2022; Published 26 April 2022

Academic Editor: Jinliang Wang

Stroke is one of the fatal diseases worldwide, and its primary mechanism is produced by cerebrovascular stenosis, blockages, or embolisms. Computer-aided diagnosis can assist clinical practitioners in identifying cerebrovascular anomalies, elucidating the precise lesions' location in the patients, and providing guidance for clinical therapy. Due to different portions of the cerebrovascular possessing diverse morphological properties and the limited narrow area, the detection effect is unsatisfactory. A retrained two-stage algorithm for detecting cerebral arterial stenosis in CTA images is proposed to solve these problems by further fusing image features and improving the quality of regions of interest. In Faster R-CNN and Libra R-CNN, the backbone network was Resnet50, with deformable convolutional and nonlocal neural networks introduced in the third, fourth, and fifth stages of the backbone network. Deformable convolutional networks learned offsets to extract morphological features of blood vessels in different tomographic planes. Nonlocal neural networks fused global information and extracted global features from location information of feature maps. A cascade detector refined object classification and bounding box regression before prediction. The experimental results show that the retained algorithm increases mAP by 7.3% and 7.5%, respectively, compared with Faster R-CNN and Libra R-CNN. Deformable convolutional networks, nonlocal neural networks, and cascade detectors are incorporated into further feature fusion; thus, semantic information about the cerebrovascular structure is learned, demonstrating more accurate stenotic region detection and demonstrating generalizability across different two-stage algorithms.

1. Introduction

Cerebrovascular disease is currently the world's second leading cause of mortality [1], with ischemic stroke being the most common type. As the primary pathogenesis of ischemic stroke, the atherosclerotic plaques cling to the vessel wall over time, restricting the lumen. Plaques can easily account for embolism, and prolonged ischemia develops as ischemic stroke [2]. Due to its speed, low physical trauma, and cost, computed tomography angiography (CTA) is frequently the first-choice radiological assessment approach for cerebrovascular disorders. Therefore, computer-aided detection (CAD) as liaison with CTA images to identify cerebrovascular stenosis can automatically assist clinicians in diagnosing abnormalities and pinpointing the precise site of lesions.

Traditional detection approaches for vascular stenosis mainly employ machine learning algorithms [3–6], which heavily rely on handcrafted features. Designing and extracting features is time-consuming and prone to human mistakes. Furthermore, traditional machine learning approaches are not competent for local and global encoding information in which they lack the semantic content of vascular image characteristics.

The advent of the convolutional neural network (CNN) has substantially increased object identification accuracy and delivered significant advances to the field of computer vision in recent years. The real-time application of a deep neural network is achievable. It attributes to the rapid growth of hardware technologies, massive data, structured information, and image processors, in which accuracy exceeds that

of many other advanced methods. The two-stage detection algorithm, which combines the proposal detector and region classifier, has gradually become prevalent because of the success of R-CNN (Region-Convolution Neural Network). Previously, the idea of region feature extraction was proposed by SPP-Net (Spatial Pyramid Pooling-Net) [7] and Fast R-CNN [8] to decrease redundant calculations in R-CNN and enhance response speed. Faster R-CNN [9] followed, achieving further acceleration by introducing a regional proposal network (RPN). The shallow layers typically learn location information, while the deep layers are responsible for semantic information. Faster R-CNN only employs the last layer (i.e., high-dimensional features) for object detection and top layer for feature prediction. It neglects feature information from other layers, resulting in a clear lack of detection capabilities for small objects. Feature pyramid networks (FPN) [10] integrate high- and low-dimensional feature information to produce fused features extracted for prediction and then provide a more substantial detection effect.

Lesion detection methods combined with deep learning also perform well in medical image processing. For example, Joo et al. [11] employed a 3D residual network combined with magnetic resonance imaging (MRI) images of the brain to the detection of aneurysms; high sensitivity and positive predictive specificity were obtained through internal and external dataset verification; Fast R-CNN was used by Smistad and Løvstakken [12] to detect deep venous thrombosis in ultrasonography, cross validation was carried out in the femoral regional dataset, and the average accuracy was 94.5%, while the accuracy was 96% in the carotid data set. Stib et al. [13] located cerebral vascular embolism sites by using DenseNet in multiphase CTA images with an AUC of 0.89, sensitivity of 100%, and specificity of 77%. De et al. [14] employed a residual encoder-decoder convolutional neural network to determine the core position of the coronary artery and a fully connected neural network to estimate the lumen cross-sectional area and demonstrated the method's viability in CT image analysis. Dai et al. [15] extracted 2D neighborhood projection images from 3D CTA images and combined them with Faster R-CNN to complete the detection of cerebral aneurysms, which was better for detecting aneurysms larger than 3 mm with a sensitivity of 96.7%. Yang et al. [16] incorporated a convolutional block attention module to Resnet18, which uses dense, atrous convolution and residual multikernel pooling blocks between encoding and decoding stages to perform intracranial aneurysm risk assessment. The algorithm detected cerebral aneurysms on CTA images with a sensitivity of 97.5%. Shinohara et al. [17] employed deep convolutional neural networks to detect hyperdense middle cerebral artery sign (HMCAS) in CT for the identification of acute cerebral infarction in the supply region. The approach effectively detects acute ischemic stroke by identifying HMCAS on noncontrast CT, with cross-validation sensitivity, specificity, accuracy, and AUC area of 82.9%, 89.7%, 86.5%, and 0.947, respectively. Hong et al. [18] fed coronary CTA dataset into CNN to quantify stenosis, and it was no significant differences between expert practitioners and the deep learning (DL) method. Chen et al. [19] assessed coronary artery stenosis with the DL method; diagnostic performances from three aspects showed that DL could take a faster and more precise response.

Although CTA plays a role as the convenient radiologic modality for cerebral arterial stenosis, the lesion location is usually not visible, and clinicians must still be strongly reliant on their discipline. Meanwhile, prolonged diagnosis raises the risk of misdiagnosis or missed diagnosis. Cerebral arteries can exhibit different shapes depending on the tomographic level, such as round, oval, shuttle, or irregular shapes. Due to individual differences among patients, feature maps are required to obtain more semantic information to detect the condition of the arteries to the greatest extent possible. The small stenosis area will lead to problems, such as sample imbalance when sampling positive and negative samples.

This paper proposes a retrained two-stage detection algorithm to identify the qualities of cerebrovascular stenosis in CTA images, improving the lesion detection performance, especially for small vessels. Faster R-CNN, classical two-stage detection neural network, and Libra R-CNN, which can effectively solve sampling imbalance, are utilized in lesion detection. Deformable convolutional networks and nonlocal neural networks were incorporated into the backbone in turn, and a cascade detector refines object classification and bounding box regression to solve the following three problems:

(1) Deformational characteristics of cerebral arteries and lesions with different tomographic levels

(2) Integrating semantic features on the lesions' global location information

(3) Optimizing the detection performance by increasing the threshold value of (intersection over union) IoU in stages

2. Libra R-CNN

An object detection network is trained in the following three stages: candidate region generation and selection, feature extraction, and category classification and bounding box regression, among others. In the detection task, the networks' performance is frequently limited by the imbalance of sample, feature, and object levels. Therefore, Libra R-CNN recommends IoU balanced sampling, a balance feature pyramid (BFP), and a balanced L1 loss function [21].

2.1. IoU Balance Sample. Difficult samples have larger loss functions, while easy samples have smaller ones. Difficult samples are essential during sampling because they are more effective at improving detection performance. A random method of selecting positive and negative samples after training the detector and generating certain boxes may result in most candidate boxes with negative samples lying in a smaller region with the IoU of the ground truth. Assume that N negative samples are drawn from a sample of M matching

candidates and that the probability of each sample being chosen at random is

$$p = \frac{N}{M}. \tag{1}$$

The IoU threshold interval is divided into K copies to increase the probability of difficult negative samples being selected. The same quantity of negative samples are sampled in each subinterval (if the average number is not reached, all samples in that subinterval are obtained), ensuring that the sampled negative samples reach as balanced a state as possible in the different IoU subintervals, and the IoU balanced sampling probability is

$$p_k = \frac{N}{K} * \frac{1}{M_k},\ k \in [0, K). \tag{2}$$

The quantity of sampling candidates in the corresponding interval K is M_k in equation (2). The default value in this study is 2, which means that the negative samples are divided into two parts according to IoU. The samples larger than the threshold are bucketed according to IoU to calculate the quantity of samples that should fall in each bucket. Finally, the negative samples with uniform IoU distribution are obtained, and the samples below the threshold are randomly sampled.

2.2. Balanced Feature Pyramid. Features' output from the backbone is fused in the FPN with the lateral connections. The feature maps of the enhanced FPN structure's output are completed in the BFP for the four-ordered steps: rescale, integrate, refine, and strengthen. The BFP is shown in Figure 1.

(1) Rescale: network proposes to obtain balanced semantic features, the semantic features of each layer must first be rescaled, and the feature maps output from the $\{C2, C3, C4, C5\}$ layers must then be unified in the $C4$ layer via interpolation and downsampling.

(2) Integrate: after unification, the feature maps are fused, and different levels of features are integrated with expressions such as

$$C = \frac{1}{L} \sum_{l=l_{\min}}^{l_{\max}} C_l, \tag{3}$$

where C_l denotes the feature at resolution level l, L the number of multilevel features, and $l_{\max}$ and $l_{\min}$ the highest and lowest level indices, respectively.

(3) Refine: it provided that convolutional kernels or nonlocal neural networks refine balanced semantic features, making more discriminators. The convolutional kernels typically have a smaller receptive field and learn local features, whereas the nonlocal neural network can incorporate more spatial location information and use the difference between local and global features to find more salient parts of the image, acquiring richer semantic features.

(4) Strengthen: the feature maps after being strengthened are added to the original feature maps of different layers to produce the enhanced FPN output $\{P2, P3, P4, P5\}$.

2.3. Balanced L1 Loss. The loss function in the object detection tasks is the sum of the classification and regression. Provided that the classification score is high, the final prediction result will have higher accuracy even if the regression is poor, so the weight of the regression loss function should be increased. In PRN, the smooth L1 loss function is frequently used to calculate the regression, the gradient corresponding to the difficult samples. In smooth L1, the gradient of the difficult samples is greater than that of the easy samples, resulting in an imbalance in the learning ability of the different samples. By smoothing the gradient at the boundary between the difficult and easy samples in the balanced L1 loss function, the balanced L1 loss function improves smooth L1:

$$\begin{aligned} &\frac{a}{b}\left(b|x| + 1\right)\ln\left(b|x| + 1\right) - \alpha|x|, \text{if } |x| < 1, \\ &\gamma|x| + C, \quad \text{otherwise}, \end{aligned} \tag{4}$$

where $\gamma = \alpha\ln(b+1)$ is defined, and the balance of loss functions between classification and regression is achieved by adjusting values of α and γ.

3. Related Work

A retrained two-stage object detection algorithm is presented in this section. Figure 2 depicts the flowchart of this study, which includes deformable convolutional networks and nonlocal networks in the backbone, as well as a cascade detector in the final prediction.

3.1. Deformable Convolutional Network. Stacking multiple convolutional layers, CNN can learn high-dimension semantic features automatically. Nevertheless, the convolutional kernels and pooling layers cannot adapt to spatial features. In a standard convolutional kernel, convolutional units sample the input feature map at a fixed position, and typical pooling layers (e.g., the average or maximum pooling layer) are fixed as well. They cannot be adaptively learned for feature downsampling, making it difficult to adapt to objects of various scales or shapes.

Standard convolutional operations, in which the activation units of the same convolutional layer all have the same receptive field, are not desirable for shallow networks encoding location information because of the complexity of the vascular structure. Different locations may correspond to objects of different geometries, and these layers require methods to adjust the scale or receptive field automatically. Deformable convolutional networks [22] learn offsets in the receptive fields to approximate the vessel shapes.

Precisely, a two-dimensional offset is calculated for inputted image pixels to construct deformable sampling point locations. The sampled position of each pixel with the

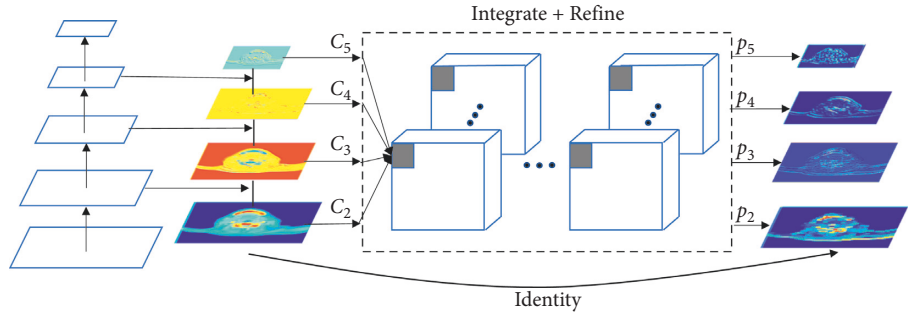


Figure 1: Visualization of balanced feature pyramid.

determined offset can override the locations of other surrounding pixels with similar features. Second, the identical structural information of neighboring pixels is compressed into a fixed grid using the deformable sampling points, and finally, the deformable feature image is formed. Therefore, a deformable convolutional network can describe the sophisticated structure, as indicated in Figure 3.

Assume that the regular convolution acts on a regular lattice R as

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n). \tag{5}$$

A deformable convolutional operation is performed on R, but each point is given a learnable offset Δp_n, and the operation is described as

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n). \tag{6}$$

A deformable convolutional network generates $2N$ feature maps corresponding to N 2D offsets Δp_n (each offset corresponding to having x- and y-directions).

3.2. Nonlocal Neural Network. CNNs are implemented as convolutional kernel windows that slide through the connections of neurons to perceive the local semantic information of an image and then integrate the local information at a higher dimension to obtain the global information. Wang et al. [23] and Shokri et al. [24] combined CNN with traditional nonlocal means to form a network structure of nonlocal blocks (shown in Figure 4), using the location information of feature maps to fuse global information and extract global features that traditional models cannot capture through repetitive convolutional operations. Furthermore, affluent global features help to find more significant parts of the image by exploiting the disparities between local features, bringing richer semantic representations to higher levels and improving the performance of existing methods. The representation of the nonlocal block is shown as

$$y_{i,j} = \frac{1}{C(x)} \sum_{\forall k,l} f\left(x_{i,j}, x_{k,l}\right) g\left(x_{k,l}\right), \tag{7}$$

where (i, j) are the position coordinates of the response to be computed and (k, l) are the coordinates of all possible positions in the input images. x denotes the input image or feature maps, and y denotes the output signal in the same dimension as x. The function $f(\cdot)$ calculates the scalar between (i, j) and (k, l), and the function $g(\cdot)$ represents the unary function of the input signal at position (k, l). $C(x)$ represents the response factor normalized to the output value. The expression for the one-dimensional function $g(\cdot)$ is

$$g\left(x_j\right) = W_g x_j, \tag{8}$$

where W_g is the weight matrix and the implementation of $g(\cdot)$ uses 2D convolutional kernels of size 1×1.

3.3. Cascade Detector. A positive sample in the detection network is larger than the IoU threshold; otherwise, it is a negative sample. A lower IoU threshold causes more background information in the extracted positive samples, making false detection more likely. A higher IoU threshold can reduce faults, but the quantity of selected positive samples will fall and then result in overfitting as the IoU threshold rises. A network predefines threshold u (assuming $u = 0.5$) to identify positive and negative samples. When the IoU of the input candidate regions is around this threshold, it frequently outperforms networks trained with alternative thresholds.

The core of Cascade R-CNN is the cascade detector [25] (shown in Figure 5), which is composed of a sequence of detection heads, each of which is trained with a dynamic threshold. The Cascade R-CNN improves on the Faster R-CNN by fine-tuning the region of interest (RoI) of the RPN output three times. The offset of each detection head's output and RoI decoding is fed into the next stage of RoI. A later detection head requires a higher IoU threshold that separates positive and negative samples. The cascade detector allows each stage of the detection head to focus on detecting the region proposals within a certain range of IoU and achieve the best results. Continuously improving the quality of the prediction frame to saturate the variation of the RoI inputs at different stages and improving the quality of the RoI ensures that each detection head has enough training samples to circumvent overfitting problems. The two components that make up the cascade detector are as follows.

(1) Multilevel bounding box regression: the regression branch calculates the offset from the candidate boxes to the ground truth. The vector of the candidate box

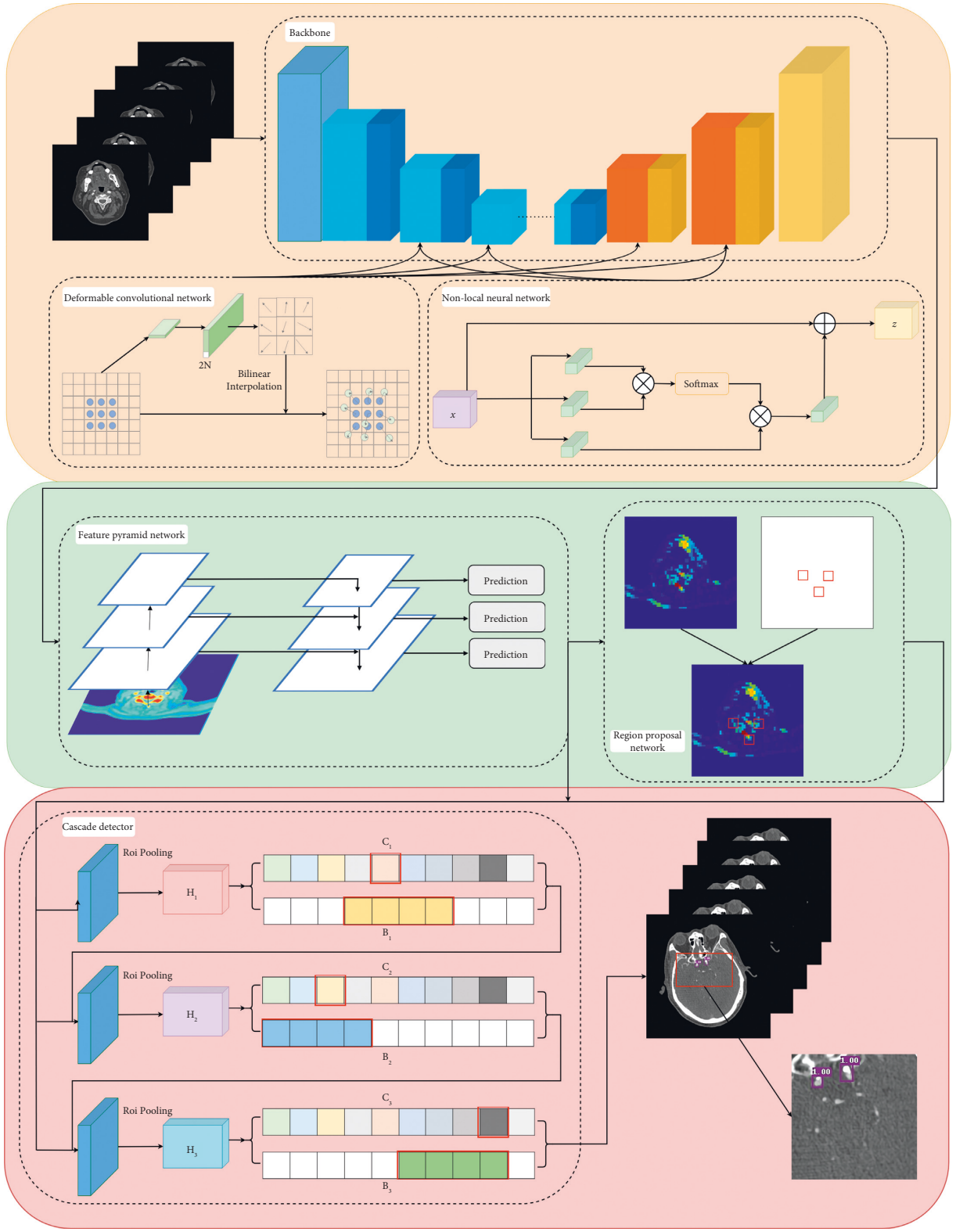


Figure 2: Flowchart of the retrained algorithm in this study. CTA images are fed into the backbone, consisting of Resnet50, deformable convolutional networks, and nonlocal neural networks incorporated in the third, fourth, and fifth stages of Resnet50. Feature maps from feature pyramid network and region proposal network with multiscale and proposal region information are integrated and then input to the cascade detector to refine the classification and regression by increasing the IoU thresholds in stages.

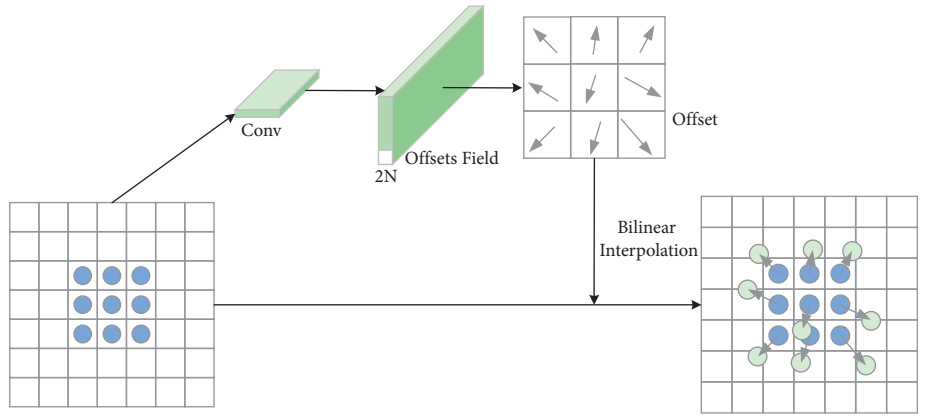


Figure 3: Structure of deformable convolutional network. The deformable convolutional layers were represented in this study by adding 18 output channels to a 2D convolution layer with a convolution kernel size of 3×3, a convolution step of 2×2, and a feature map padding of 1 as offsets before a 2D convolution layer with a convolution kernel size of 3×3, a convolution step of 2×2, and a feature map padding of 1.

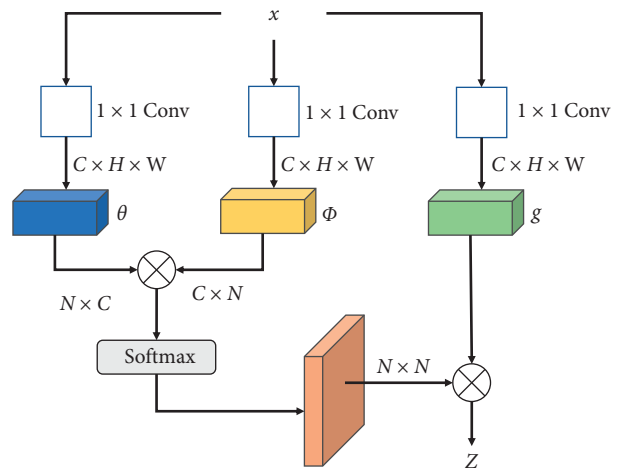


Figure 4: Structure of nonlocal neural network. C, H, and W correspond to dimensions, where $N = H \times W$.

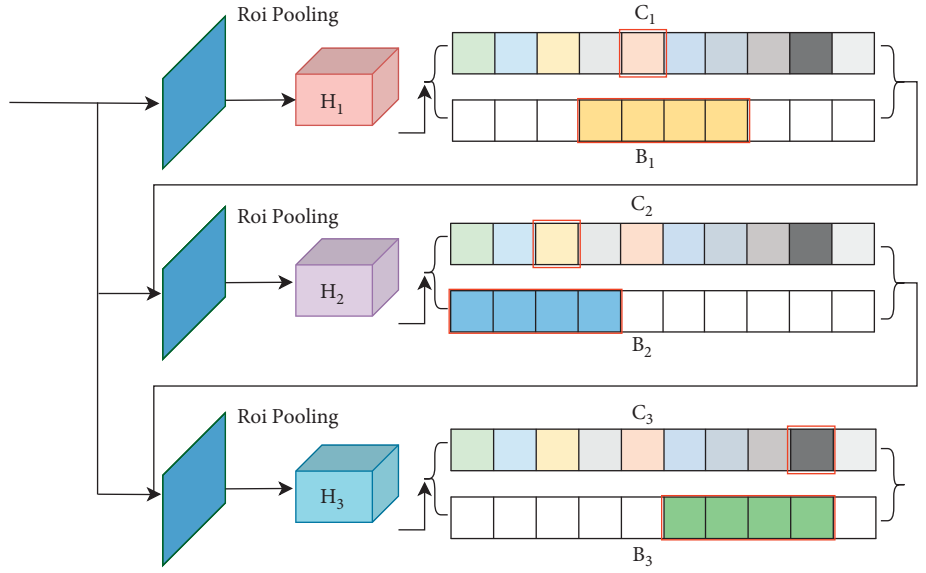


Figure 5: Structure of the cascade detector. Feature maps are fed into the cascade detector, and classification and bounding box regression are refined to optimize prediction, in which C_n and B_n represent classification and bounding box regression in n stage, respectively.

is represented by $b = (x_b, y_b, w_b, h_b)$, where x_b and y_b denote the central point position of the candidate box and w_b and h_b denote the width and height, respectively; the vector of the ground truth is represented by $b = (x^*, y^*, w^*, h^*)$, where x^* and y^* denote the central point position and w^* and h^* denote the width and height, respectively. The expression for the offset t^* from the candidate box to the ground truth is given as

$$\begin{cases} t_x^* = \dfrac{(x^* - x_b)}{w_b}, \\ t_y^* = \dfrac{h_b}{(y^* - y_b)}, \\ t_w^* = \log\left(\dfrac{w^*}{w_b}\right), \\ t_h^* = \log\left(\dfrac{h^*}{h_b}\right). \end{cases} \tag{9}$$

The candidate box is regressed towards the ground truth b^* using the regressor $f(x, b)$, where the expression for the loss function is

$$\begin{aligned} R_{\text{loc}}[f] &= \sum_{i=1}^{N} L_{\text{loc}}\left(f(x_i, b_i), b_i^*\right), \\ L_{\text{loc}}\left(y_i, b_i^*\right) &= \text{blance}_{L_1}\left(y_i - b_i^*\right). \end{aligned} \tag{10}$$

Cascade regression with a series of specific regression quantities is implemented with the following formula:

$$f(x, b) = f_T \circ f_{T-1} \circ \cdots \circ f_1(x, b), \tag{11}$$

where T denotes the total cascade stages and f_T denotes the regression corresponding to stage a.

(2) Classification: the classification function is defined as $h(x)$. The classifier divides the samples into $K+1$ classes, where class 0 contains the background information as well as the target to be detected. The given training sample set (x_i, y_i) is minimized by learning the classification risk as follows:

$$L_{cls}(h(x), y) = \sum_{i \in I} L_{cls}\left(h_i(x), y_i\right), \tag{12}$$

where L_{cls} is the cross-entropy loss function and y_i is the class label to which the corresponding image x_i belongs.

The cascade detector minimizes the multitasking loss function in each stage t by taking the output of the previous stage as the input of the next stage, the classifier h_t and the regressor f_t are used to optimize the threshold u_t $(u_t > u_t - 1)$ of the IoU, and the multitasking loss function expression is shown as.

$$L\left(x^t, g\right) = L_{\text{cls}}\left(h_t\left(x^t\right), y^t\right) + \lambda\left[y^t \geq 1\right] L_{\text{loc}}\left(f_t\left(x^t, b^t\right), g\right), \tag{13}$$

where $b_t = f_t - 1(x_t - 1, b_t - 1)$, g denotes the ground truth of the target x_t, y_t is the predicted label of x_t, and λ is the trade-off factor.

4. Materials and Parameters

The Qianjiang District Central Hospital in Chongqing provided the CTA dataset for cerebral arterial stenosis, including 109 patients. The data were desensitized and labelled in Pascal VOC2012 data format using Labelme by qualified radiologists with more than five years of experience. The data were then converted into COCO data format. The dataset was divided into a training set with 1645 images and a test set with 410 images.

The PaddlePaddle framework was used to create the experimental environment, which included an Ubuntu 18.04, a Tesla V100 graphics card, 32 GB of RAM, and a Resnet50 backbone network. With a total training epoch = 20, the initial learning rate was 0.00125, and with batch size = 2, the learning rate was reduced to 0.1 times the initial learning rate at epoch = 12 and epoch = 19, respectively. The size of the input image was 512×512 pixels.

5. Results

The experiment's metrics were conducted using mAPbest (mean average precision), mAP50, mAP75, and APS, where mAPbest represents the best mean average precision in the test set, AP50, AP75, and APS denote the average precision at IoU = 0.50, 0.75 and small objects (area $< 32 \times 32$). For objects in category C, the mAP is calculated as shown in equations (14) and (15):

$$\text{Precision}_C = \frac{\text{N}(\text{True Positives})_C}{\text{N}(\text{Total Objects})_C}, \tag{14}$$

$$\text{Average Precision}_C = \frac{\sum \text{Precision}_C}{N(\text{Total Images})_C}. \tag{15}$$

5.1. One-Stage and Two-Stage Algorithm Comparison. The results of one-stage detection Yolov3, as well as two-stage detection Faster R-CNN, Libra R-CNN, and Cascade R-CNN, are shown in Table 1. A training and testing strategy was employed in this experiment simultaneously. Figure 6 depicts the curve of mAPbest over time for each epoch, and Figure 7 depicts the visualization result of the above four basic algorithms.

5.2. Retrained Faster R-CNN Comparison. The experimental result, presented in Table 2, indicates that by adding deformable convolutional networks, nonlocal neural networks in the backbone network in Faster R-CNN, and the cascade detector added in the classification and regression branch

Table 1: Results of basic object detection algorithms.

Method	mAPbest	mAP0.5 : 0.95	mAP0.5	mAP0.75	mAPs
Yolov3 [26]	44.9	44.9	94.5	28.3	44.9
Faster R-CNN [9]	45.7	45.2	94.0	29.6	45.2
Libra R-CNN [21]	45.9	45.8	96.6	29.6	45.8
Cascade Libra R-CNN [25]	48.5	48.1	97.1	33.1	48.1

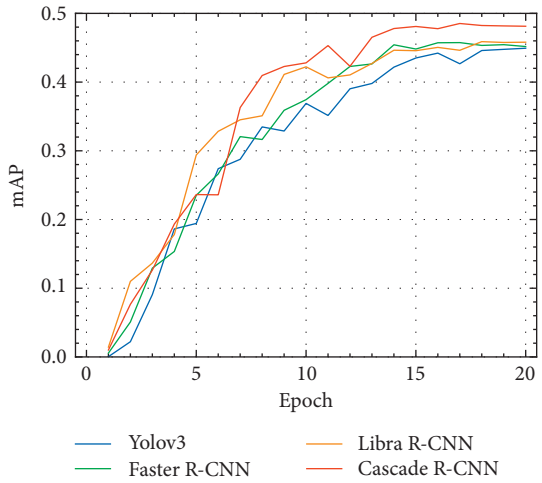
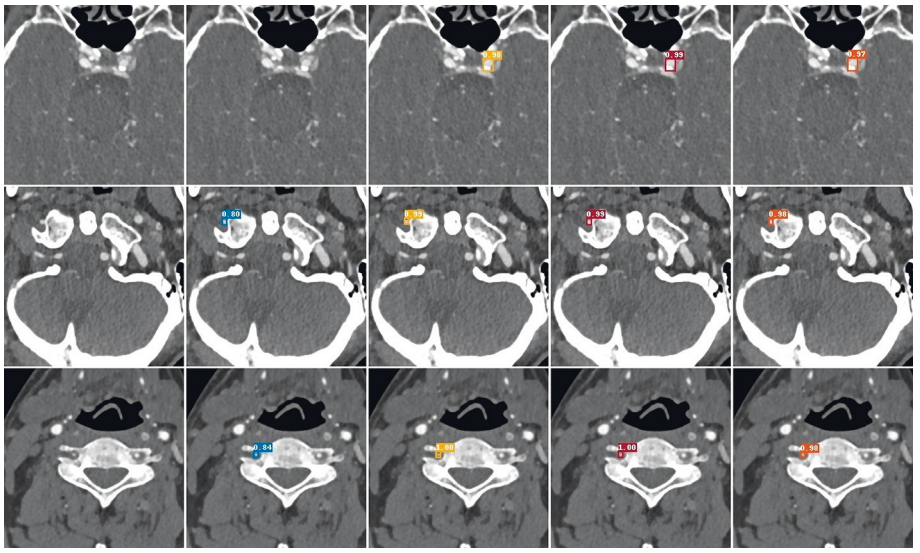


Figure 6: Visualization of the one-stage and two-stage algorithm comparison curve in mAPbest.



(a) (b) (c) (d) (e)

Figure 7: (a–e) Comparison between one-stage and two-stage algorithm prediction. (a) Raw image, (b) Yolov3, (c) Faster R-CNN, (d) Libra R-CNN, and (e) Cascade R-CNN.

refinement, metrics (in terms of mAP) are improved. The curve in mAPbest and the visualization are shown in Figures 8 and 9.

5.3. Retrained Libra R-CNN Comparison. The experimental result, presented in Table 3, indicates that, by gradually adding the above three modules to Libra R-CNN, all the

Table 2: Ablation comparison experiment of retrained Faster R-CNN.

Faster R-CNN	Dcn	Nonlocal	Cascade detector	mAPbest	mAP0.5 : 0.95	mAP0.5	mAP0.75	mAPs
✓				45.7	45.2	94.0	29.6	45.2
✓	✓			46.3	46.1	96.9	35.3	46.1
✓	✓	✓	✓	47.3	47.2	95.9	37.7	47.2
✓	✓	✓	✓	53.0	52.5	97.1	45.7	52.5

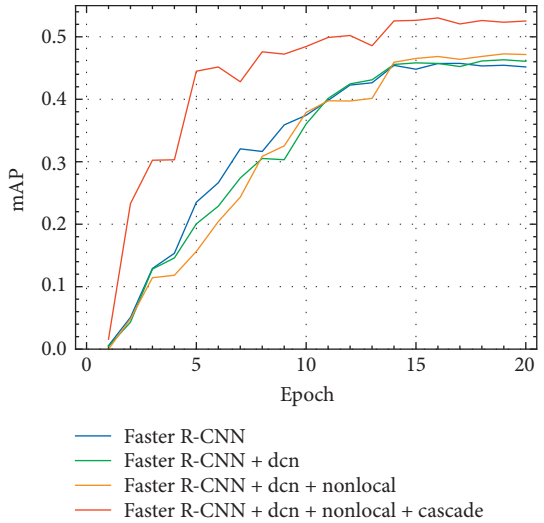


Figure 8: Visualization of the retrained Faster R-CNN curve in mAPbest.

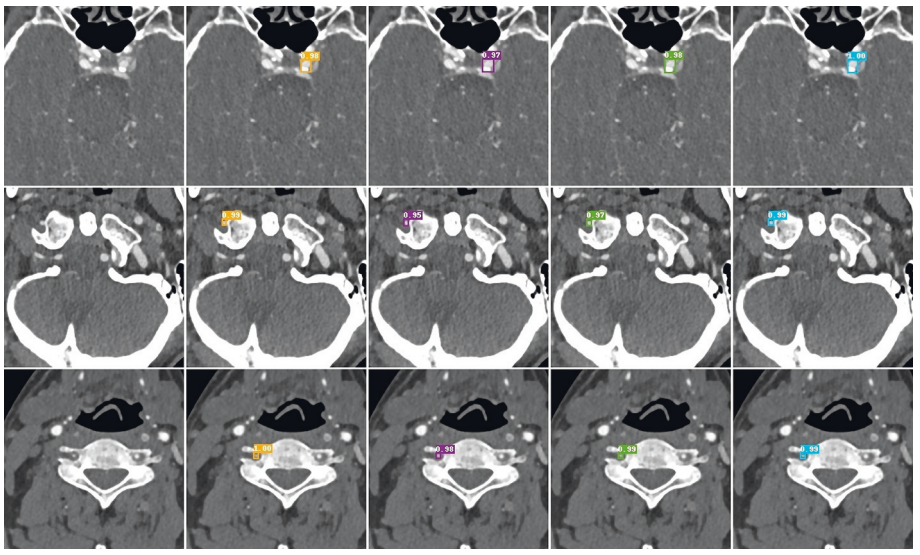


Figure 9: Prediction of retrained Faster R-CNN. (a) Raw image, (b) Faster R-CNN, (c) Faster R-CNN + dcn, (d) Faster R-CNN + dcn + nonlocal, and (e) Faster R-CNN + dcn + nonlocal + cascade detector.

Table 3: Ablation comparison experiment of retrained Libra R-CNN.

Libra R–CNN	DCN	Nonlocal	Cascade detector	mAP (best)	mAP0.5 : 0.95	mAP0.5	mAP0.75	mAPs
✓				45.9	45.8	96.6	29.6	45.8
✓	✓			48.6	47.6	96.9	34.3	47.6
✓	✓	✓	✓	48.1	48.1	95.7	40.1	48.1
✓	✓	✓	✓	53.4	53.1	97.0	50.7	53.1

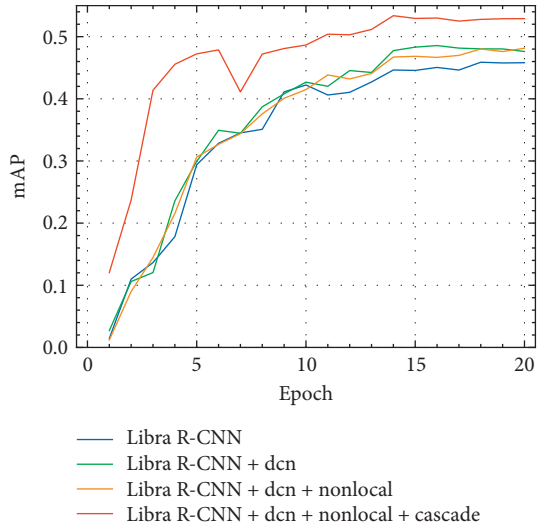
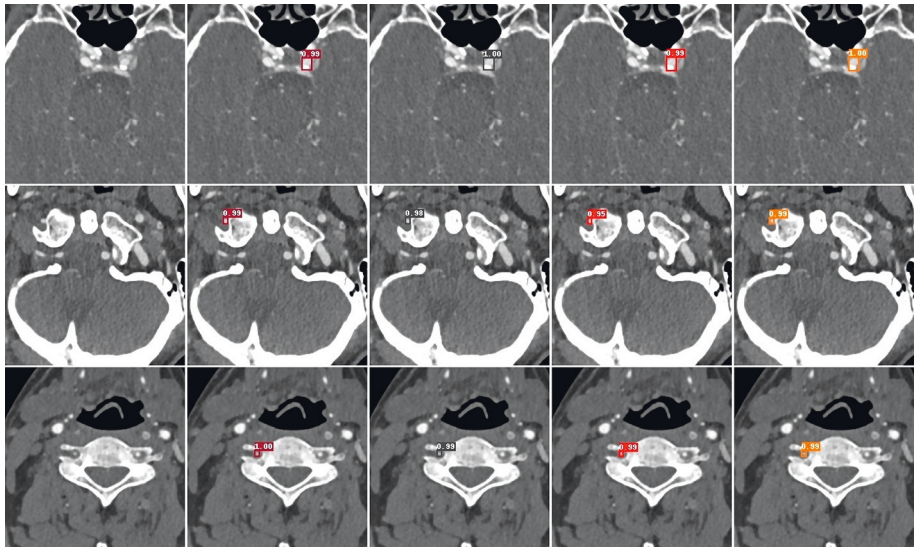


Figure 10: Visualization of the retrained Libra R-CNN curve in mAPbest.



(a) (b) (c) (d) (e)

Figure 11: Prediction of retrained Libra R-CNN. (a) Raw image, (b) Libra R-CNN, (c) Libra R-CNN + dcn, (d) Libra R-CNN + dcn + nonlocal, and (e) Libra R–CNN + dcn + nonlocal + cascade detector.

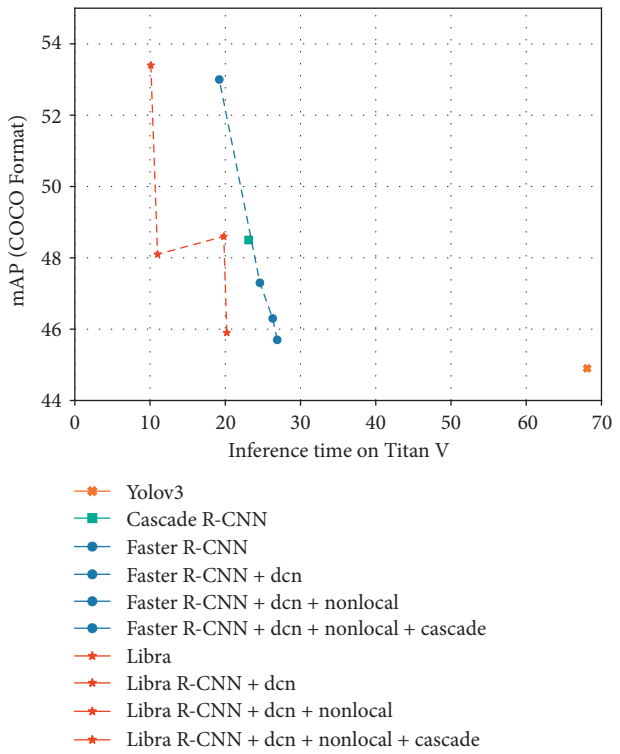


Figure 12: Visualization between inference time and mAPbest. It shows that the one-stage algorithm has its advantage in speed. The mAP of the two-stage algorithm rises as the above modules are added, but the computational complexity grows as well.

metrics of mAP are also improved. However, in Libra R-CNN + dcn + nonlocal, its mAPbest and mAP0.5 are decreased by 0.5% and 1.2%, respectively, compared to Libra R-CNN + dcn. The curve in mAPbest and the visualization are shown in Figures 10 and 11, respectively.

6. Discussion and Conclusions

This study proposed multiple modules as part of a two-stage algorithm for detecting cerebral arterial stenosis. The retrained networks employed deformable convolutional networks in backbone networks to learn offsets to extract morphological features of vessels in different tomographic planes, while nonlocal neural networks were incorporated into the backbone with the deformable convolutional networks at the same stages to learn deeper semantic representations by fusing global information with the location information of the features maps. Finally, a cascade detector optimizes the prediction performances by increasing the threshold value of IoU in stages.

The proposed methods outperform the above mainstream object detection algorithms in the CTA dataset of cerebral arterial stenosis, with considerable improvements in both objective metrics (mAP, mAP50, and mAP75) and prediction visualization. The methods' accuracy for small objects is also increased by optimizing the network structure.

Although the proposed algorithm is superior to baseline approaches such as Faster R-CNN and Cascade R-CNN, multiple modules are layered on top of each other with redundant network topologies. They increased the parameter quantity, which may cause a slower detection speed (shown in Figure 12) and higher cost. Stenosis is not presented in only one tomographic plane for a patient whose images are regarded as independent samples in this study, and liaison between two continuous levels is not well established. Furthermore, the proposed method does not effectively solve the problems that outline the lesion area and classify the stenosis grading which can provide more precise guidance for subsequent clinical treatment. In our subsequent work, we plan to investigate ways to simplify the network structure, minimize the parameters, increase detection accuracy and speed, and calculate the lesion area and its grading.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Authors' Contributions

Hanqing Liu and Xiaojun Li contributed equally to this work.

Acknowledgments

This work was supported by the Beijing-Tianjin-Hebei Collaborative Innovation Project (17YEXTZC00020).

References

[1] Y.-Z. Hsieh, Y.-C. Luo, C. Pan, M.-C. Su, C.-J. Chen, and K. L.-C. Hsieh, "Cerebral small vessel disease biomarkers detection on MRI-sensor-based image and deep learning," *Sensors*, vol. 19, no. 11, p. 2573, 2019.

[2] S. Bash, J. P. Villablanca, R. Jahan et al., "Intracranial vascular stenosis and occlusive disease: evaluation with CT angiography, MR angiography, and digital subtraction angiography," *American Journal of Neuroradiology*, vol. 26, no. 5, pp. 1012–1021, 2005.

[3] K.-C. Hsu, C.-H. Lin, K. R. Johnson et al., "Autodetect extracranial and intracranial artery stenosis by machine learning using ultrasound," *Computers in Biology and Medicine*, vol. 116, Article ID 103569, 2020.

[4] T. Araki, P. K. Jain, H. S. Suri et al., "Stroke risk stratification and its validation using ultrasonic echolucent carotid wall plaque morphology: a machine learning paradigm," *Computers in Biology and Medicine*, vol. 80, pp. 77–96, 2017.

[5] L. Saba, P. K. Jain, H. S. Suri et al., "Plaque tissue morphology-based stroke risk stratification using carotid ultrasound: a polling-based pca learning paradigm," *Journal of Medical Systems*, vol. 41, no. 6, p. 98, 2017.

[6] S. L. Waddle, M. R. Juttukonda, S. K. Lants et al., "Classifying intracranial stenosis disease severity from functional MRI data using machine learning," *Journal of Cerebral Blood Flow and Metabolism*, vol. 40, no. 4, pp. 705–719, 2020.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[8] R. Girshick, "Fast R-CNN," in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pp. 1440–1448, IEEE, Santiago Chile, December 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, vol. 10949, pp. 2117–2125, IEEE, Honolulu, HI, USA, July 2017.

[11] B. Joo, S. S. Ahn, P. H. Yoon et al., "A deep learning algorithm may automate intracranial aneurysm detection on MR angiography with high diagnostic performance," *European Radiology*, vol. 30, no. 11, pp. 5785–5793, 2020.

[12] E. Smistad and L. Løvstakken, "Vessel detection in ultrasound images using deep convolutional neural networks," *Deep Learning and Data Labeling for Medical Applications*, Springer, Cham, pp. 30–38, 2016.

[13] M. T. Stib, J. Vasquez, M. P. Dong et al., "Detecting large vessel occlusion at multiphase CT angiography by using a deep convolutional neural network," *Radiology*, vol. 297, no. 3, pp. 640–649, 2020.

[14] Q. De Man, E. Haneda, B. Claus et al., "A two-dimensional feasibility study of deep learning-based feature detection and characterization directly from CT sinograms," *Medical Physics*, vol. 46, no. 12, pp. e790–e800, 2019.

[15] X. Dai, L. Huang, Y. Qian et al., "Deep learning for automated cerebral aneurysm detection on computed tomography images," *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, no. 4, pp. 715–723, 2020.

[16] J. Yang, M. Xie, C. Hu et al., "Deep learning for detecting cerebral aneurysms with CT angiography," *Radiology*, vol. 298, no. 1, pp. 155–163, 2021.

[17] Y. Shinohara, N. Takahashi, Y. Lee, T. Ohmura, and T. Kinoshita, "Development of a deep learning model to identify hyperdense MCA sign in patients with acute ischemic stroke," *Japanese Journal of Radiology*, vol. 38, no. 2, pp. 112–117, 2020.

[18] Y. Hong, F. Commandeur, and S. Cadet, "Deep learning-based stenosis quantification from coronary CT angiography," in *Proceedings of the Medical Imaging 2019: Image Processing. SPIE*, vol. 10949, pp. 643–651, CF, USA, March 2019.

[19] M. Chen, X. Wang, G. Hao et al., "Diagnostic performance of deep learning-based vascular extraction and stenosis detection technique for coronary artery disease," *British Journal of Radiology*, vol. 93, no. 1113, Article ID 20191028, 2020.

[20] K. Han, L. Chen, and D. B. Geleri, "Deep-learning based significant stenosis detection from multiplanar reformatted Images of traced Intracranial arteries," *In: American Society of Neuroradiology 58th Annual Meeting.*, 2020.

[21] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: towards balanced learning for object detection," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 821–830, IEEE, Long Beach, CA, USA, June 2019.

[22] J. Dai, H. Qi, and Y. Xiong, "Deformable Convolutional Networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision*, pp. 764–773, IEEE, Venice, Italy, October 2017.

[23] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local Neural Networks," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern recognition*, June 2018.

[24] M. Shokri, A. Harati, and K. Taba, "Salient object detection in video using deep non-local neural networks," *Journal of Visual Communication and Image Representation*, vol. 68, Article ID 102769, 2020.

[25] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, IEEE, Salt Lake City, UT, USA, June 2018.

[26] J. Redmon and A. Farhadi, "Yolov3: An Incremental Improvement [EB/OL]," 2021, https://arxiv.org/pdf/1804.02767.pdf.