symmetry

MDPI

Article

Symmetry GAN Detection Network: An Automatic One-Stage High-Accuracy Detection Network for Various Types of Lesions on CT Images

Yan Zhang †, Shupeng He †, Shiyun Wa †, Zhiqi Zong †, Jingxian Lin †, Dongchen Fan †, Junqi Fu † and Chunli Lv *

College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China; 2019308250102@cau.edu.cn (Y.Z.); 2019505430320@cau.edu.cn (S.H.); 2019308250126@cau.edu.cn (S.W.); 2021505030412@cau.edu.cn (Z.Z.); 2018314860117@cau.edu.cn (J.L.); 2018314860110@cau.edu.cn (D.F.); 2018314130321@cau.edu.cn (J.F.)

* Correspondence: lvcl@cau.edu.cn

† These authors contributed equally to this work.

Abstract: Computed tomography (CT) is the first modern slice-imaging modality. Recent years have witnessed its widespread application and improvement in detecting and diagnosing related lesions. Nonetheless, there are several difficulties in detecting lesions in CT images: (1) image quality degrades as the radiation dose is reduced to decrease radiational injury to the human body; (2) image quality is frequently hampered by noise interference; (3) because of the complicated circumstances of diseased tissue, lesion pictures typically show complex shapes; (4) the difference between the orientated object and the background is not discernible. This paper proposes a symmetry GAN detection network based on a one-stage detection network to tackle the challenges mentioned above. This paper employs the DeepLesion dataset, containing 10,594 CT scans (studies) of 4427 unique patients. The symmetry GANs proposed in this research consist of two distinct GAN models that serve different functions. A generative model is introduced ahead of the backbone to increase the input CT image series to address the typical problem of small sample size in medical datasets. Afterward, GAN models are added to the attention extraction module to generate attention masks. Furthermore, experimental data indicate that this strategy has significantly improved the model's robustness. Eventually, the proposed method reaches 0.9720, 0.9858, and 0.9833 on P, R, and mAP, on the validation set. The experimental outcome shows that the suggested model outperforms other comparison models. In addition to this innovation, we are inspired by the innovation of the ResNet model in terms of network depth. Thus, we propose parallel multi-activation functions, an optimization method in the network width. It is theoretically proven that by adding coefficients to each base activation function and performing a softmax function on all coefficients, parallel multi-activation functions can express a single activation function, which is a unique ability compared to others. Ultimately, our model outperforms all comparison models in terms of P, R, and mAP, achieving 0.9737, 0.9845, and 0.9841. In addition, we encapsulate the model and build a related iOS application to make the model more applicable. The suggested model also won the second prize in the 2021 Chinese Collegiate Computing Competition.

Keywords: CT images; object detection; symmetry GANs; one-stage network

Citation: Zhang, Y.; He, S.; Wa, S.; Zong, Z.; Lin, J.; Fan, D.; Fu, J.; Lv, C. Symmetry GAN Detection Network: An Automatic One-Stage High-Accuracy Detection Network for Various Types of Lesions on CT Images. *Symmetry* **2022**, *14*, 234. https://doi.org/10.3390/sym14020234

Academic Editor: Dumitru Baleanu

Received: 31 December 2021
Accepted: 18 January 2022
Published: 25 January 2022

1. Introduction

Lesions occur in body tissue due to various factors, including trauma, infection, or cancer [1]. Take, for example, a brain tumor; this type of neoplasm arises in the brain and has a considerable fatality probability. Brain tumors occupy the most intracranial space, impacting brain function, severely impairing the patients' central nerves, and overwhelming brain cells. Meanwhile, brain tumor varieties are numerous and distinct. Some tumors are problematic to scrutinize, such as schwannoma [2]; others are challenging to locate,

such as glioma and glioblastoma [3]. In addition, lung nodules, i.e., sarcoidosis, are a multi-system and multi-organ granulomatous disease of unknown etiology. Lung nodules often invade the eyes, lungs, skin, bilateral hilar lymph nodes, and other organs, with a chest invasion rate of 80% to 90%. Furthermore, lymph nodes are distributed throughout the body and are essential immune organs. The enlarged lymph node is caused by acute and chronic inflammation caused by pathogenic microorganisms, such as acute cellulitis, purulent tonsillitis, gingivitis, tuberculosis, etc. These types of lesions severely impair the mechanisms and functionalities of the human body.

X-ray computed tomography (CT) is widely applied to visualize cross-sections in clinical and industrial fields [4], especially providing explicit information for diagnosing the lesions mentioned above. The wide application of CT is because of its ability to examine the body's interior structures without destroying organs' or subjects' surfaces [5]. Currently, medical CT images are typically used in anatomical structure research, treatment planning, tissue recognition, and gland volume measurement [6].

In clinical medicine, doctors generally need to determine the location of the lesion before the diagnosis and operation [7]. Following confirmation of the location, clinicians can collect basic information about the lesion, including the shape, position, and size. Subsequently, they develop a proper and precise therapeutic scheme or surgery. Medical imaging techniques are used to identify and depict lesions for further treatment, as mentioned above. Currently, these types of techniques for lesions include MRI [8], PET/CT [9], and CT [4]. CT images can represent the anatomical structure of organs and tissues in the human body, making them the exemplary medical imaging modality for numerous lesions. Radiology doctors can manually observe and identify the situation of lesions based on CT images. Size, shape, location, physiological trait, and metabolic status are the most common factors they evaluate.

However, considerable impediments and challenges exist regarding lesion detection in CT images.

1. The drastic growth in CT utilization leads to an upward trend in the total quantity of radiation applied to patients [10]. Radiation damage to the body accumulates with the number of times it is exposed to radiation. Therefore, each CT examination raises the risk, which will eventually lead to a significant radiation dose after a while.
2. Assuming a reduction in the radiation dose to address the above issue, the image quality drops if the scan and reconstruction variables remain unchanged. Such a dose decrease and image quality degradation might jeopardize the assessment of specific anatomic regions [11]. Moreover, it will also impact the diagnostic information in particular body regions.
3. Most lesion images display complex tissue structures since diseased tissue usually results from complex conditions, such as rupture, unclear boundaries, and external factors such as noise [12]. Moreover, various structures and blood vessels are distributed in diseased organs [13]. These features make determining the extent of the lesion difficult.
4. The image quality is frequently hampered by noise interference [14]. However, image information details would be reduced while noise would be eliminated.
5. Lesion structure between individuals exhibits a vast difference. Furthermore, even within the same human body, there is a considerable degree of variability in the morphology of tissues, and similarities between lesion tissues and normal tissues can be observed, easily leading to misdiagnosis and missed diagnosis [15].
6. Because of pathological variables and external noise interference, the contrast between the targeted object and the background is minimal. Nonetheless, traditional detection necessitates a visible distinction of the object's illumination in comparison to the backdrop.

At present, a few clinicians still rely on manual and subjective lesion detection and lesions' and organ contours' delineation in CT [16], which requires the doctors to have extensive prior knowledge. Despite the fact that computer-aided detection/diagnosis

(CADe/CADx) has been a thriving research topic and occupies a prominent position in medical image processing [17], numerous lesion detection approaches are challenging to undertake and difficult to implement in clinical diagnosis. In addition, at present, the majority of research in lesion detection merely adopts a dataset that includes only one or a few types of lesion. Given the clinical need and the medical value, establishing an accurate, dependable, and fully automatic detection system for various lesion types is critical.

Inspired by the medical requirements and preceding research, this paper chooses a dataset containing various lesion types, such as renal lesions, bone lesions, lung nodules, and enlarged lymph nodes. This paper suggests a symmetry GAN detection network based on the dataset, intending to address difficulties, promote technological development in CADe/CADx, and contribute to clinical medical research. Additionally, the symmetry GAN detection network proposed in this paper won the first prize in Beijing's 2021 Chinese Collegiate Computing Competition, while ranking second nationwide. The following are the primary contributions of this paper:

1. The symmetry GAN models: Firstly, we add a generative model ahead of the backbone to expand the input CT image series, aiming to address the typical challenge of small sample size in medical datasets. Subsequently, we also add GAN models to the attention extraction module to generate attention masks. By adding GAN models on feature maps, this strategy can effectively make the model robust enough.
2. Effectiveness verification of multiple implementations of symmetry GANs: We use DCGAN and CVAE-GAN to test the performance of GAN model A, and we adopt Self-Attention Generative Adversarial Networks (SAGAN) and Spatial Attention GAN (SPA-GAN) to test that of GAN model B. Experimental results show that the combination of DCGAN + SPA-GAN performs the best, further improving the model's detection accuracy.
3. Parallel multi-activation functions: We utilize parallel multi-activation functions to replace single activation functions. Theoretically, this optimization proves that the performance of parallel multi-activation functions is superior to that of single activation functions. Furthermore, we replace the IoU loss with a more reasonable $CIoU$ loss to enhance the detection task's loss function.
4. An IOS application: Additionally, we encapsulate the symmetry GAN model and establish an application based on the iOS platform, realizing the practical value of the model.

The following shows the organization of the rest of this paper: Related Work demonstrates preceding studies in the selected research area; the Materials and Methods section introduces the dataset as well as design specifics; the Experiment section depicts the experimental operation and platform; the Results section displays the outcomes of the experiments and analyses; the Discussion section describes several ablation experiments to validate the improved approach's efficacy and our methodology's drawbacks; the Conclusions section summarizes the entire paper.

2. Related Work

Object detection is one of the most crucial research topics in the field of computer vision. Along with the machine learning [18–20] and deep learning [21,22] booms in image detection applications, several automated computer vision solutions have been introduced to assess image object detection. Before 2012, the traditional machine learning algorithm was generally adopted to conduct object detection. Afterward, CNN-based models for object detection could be divided into two branches: two-stage models and one-stage models. Two-stage models include Mask R-CNN [23] and Faster R-CNN [24]. The two-stage algorithm needs to generate proposals (a pre-selected box that could contain potential objects to be detected) and then conduct fine-grained object detection. One-stage models include the You Only Look Once (YOLO) [25–27] series, Single Shot multi-box Detector (SSD) [28] series, and EfficientDet [29] series; in comparison, the one-stage algorithm extracts features directly in the network to predict the object category and address.

Hence, the two-stage algorithm is relatively slow because it needs to run the detection and classification process several times. In contrast, the other one-stage object detection algorithm predicts all the bounding boxes by feeding them into the network only once, which is relatively fast and ideal for mobile applications. Due to the above, in this paper, we select the one-stage detection network.

Nowadays, thanks to some researchers incorporating new modules and improvements in related studies, such as agriculture, industry, and medicine, multiple new CNN methods are being developed. In the agricultural field, for example, an optimized CNN model was utilized to detect pear defects; more specifically, a deep convolutional adversarial generation network was adopted to expand the diseased images [30]. Experimental results showed that the detection accuracy of the presented method on the validation set reached 97.35%. Furthermore, the model worked satisfactorily on two untrained varieties of pears, which reflected its robustness and generalization potential. Taking maize leaf disease detection as another instance, Yan Zhang et al. [31] proposed a CNN enhanced by a multi-activation function (MAF) module. This study adopted image preprocessing to expand and augment the disease samples and adopted transfer learning and warm-up methods to increase the training speed. The suggested method could detect three categories of maize diseases efficiently and accurately, reaching an accuracy value of 97.41% in the validation set, surpassing that of the traditional AI methods. In addition, a CNN-based detection network model using a generative module and pruning inference was once proposed [32]. The presented pruning inference automatically deactivated part of the network structure in terms of diverse situations, decreased parameters and operations, and improved network speed. When detecting apple flowers, this model achieved values of 90.01%, 98.79%, and 97.43% in precision, recall, and mAP, respectively. The inference speed reached 29 FPS. In medicine, an automatic brain tumor segmentation algorithm—GenU-Net++—was suggested based on the BraTS 2018 dataset [7]. This study adopted the generative mask sub-network to develop feature maps, and it utilized the BiCubic interpolation method for upsampling to gain segmentation results. Meanwhile, this research applied an auto-pruning mechanism according to the structural features of U-Net++, which could deactivate the sub-network and automatically prune GenU-Net++ during the inference. This mechanism accelerates inference and improves the network performance. This algorithm's PA, $MIoU$, P, and R reached 0.9737, 0.9745, 0.9646, and 0.9527, respectively.

Moreover, among the fields mentioned above, the medical industry is one of the most vibrant research areas based on the application of CNNs, which has been developed, employed in computational biomedical domains, and significantly contributed [33]. Doctors can effectively analyze medical images for lesion detection and diagnosis decision-making using the CADe/CADx system. Automatic detection based on CNNs has lately gained popularity. Image features can be automatically learned via automatic detection [34]. B. Savelli et al. [35] proposed a new method for detecting small lesions in digital medical images. This method was built on the basis of a multi-context ensemble of CNNs. The innovative multiple-depth CNNs were trained on image patches of varying dimensions before combination. As a result, the final ensemble could detect and pinpoint anomalies on images by using the surrounding context and local features of a lesion. Statistically, the suggested ensemble showed notably sounder detection performance, displaying its efficacy in detecting minor abnormalities. Yang Liu et al. [36] proposed a novel privacy-preserving Faster R-CNN framework (SecRCNN) for detecting medical image objects. They created a set of interactive protocols to complete the three stages of Faster R-CNN: feature map extraction, region proposal, and regression and classification. SecRCNN's current secure computation sub-protocols, such as division, exponentiation, and logarithm, were upgraded to increase SecRCNN's efficiency. The provided sub-protocols can remarkably decrease the messages' numbers that have been swapped in the iterative approximation. Experimental results revealed that the communication overhead was reduced to 36.19%, 73.82%, and 43.37% in terms of computing division, logarithm, and exponentiation, respectively. Dimpy Varshni et al. [37] established an automatic system for detecting pneumonia without delay, especially in remote areas. Their study evaluated the pre-trained CNN

model's performance as feature extractors, followed by various classifiers to classify chest X-rays. For this purpose, the best CNN model was determined analytically. Statistically, the results of the experiments revealed that using pre-trained CNN models in conjunction with supervised classifier algorithms might be highly advantageous in evaluating chest X-ray images, particularly for detecting pneumonia.

Additionally, deep learning has wide application in the medical area due to its excellent accuracy and efficacy in image classification and biological applications [38]. The generative adversarial network (GAN) is prevalent and significant among all the deep learning architectures in the relative research topics. In the research on lesion detection, previous specialists continued presenting advanced algorithms to optimize the preprocessing, segmentation, and classification to enhance the detection accuracy and prepare for the subsequent processing for multiple types of lesion images. AviBen-Cohen et al. [39], for example, demonstrated a unique approach for generating virtual PET images from CT scans. They merged a fully convolutional network (FCN) with a conditional GAN to produce simulated PET data from supplied CT data. Encouragingly, the experimental results demonstrated a 28% drop in the average false positive per case from 2.9 to 2.1. The proposed solution can be extended to a variety of organs and modalities. Jin Zhu et al. [40] proposed a novel SISR method to enhance the spatial resolution for brain tumor MRI images while avoiding the introduction of unrealistic textures. In addition, they proposed an MOS that integrates experts' domain knowledge to evaluate the medical image SR results. According to the experimental results, the suggested method using MS-GAN accomplished efficient SISR for brain tumor MRI images. Such models can be successfully employed for a broader range of clinical applications. To detect brain abnormalities at diverse phases on multi-sequence structural MRI, Leonardo Rundo et al. [41] suggested an unsupervised medical anomaly detection generative adversarial network (MADGAN). The self-attention MADGAN could detect AD at an early stage, with an area under the curve of 0.727, and AD at a late stage with AUC 0.894, whereas it achieved AUC 0.921 for brain metastases detection on T1c scans. Moreover, Maryam Hammami et al. [42] designed a combined Cycle GAN and YOLO method for CT data augmentation. The experimental findings showed that detection was speedy and accurate, with an average distance of 7.95 ± 6.2 mm, which was particularly superior to detection without being augmented. The novel method outperformed state-of-the-art detection methods for medical images. Finck, Tom MD, et al. [43] adopted a deep-learning technique to generate computationally generated DIR images and compared their diagnostic performance to that of conventional sequences in patients with multiple sclerosis (MS). The use of synthDIR enabled the detection of much more lesions. This improvement primarily contributed to the better representation of juxtacortical lesions (12.3 10.8 vs. 7.2 5.6, P 0.001). Zhiwei Qin et al. [44] used a data augmentation technique based on GANs to classify skin lesions, allowing doctors to make more accurate diagnoses. Finally, the suggested skin-lesion-based GANs' synthetic images were incorporated into a training set, helping to train a classifier for superior classification performance. When the synthesized images were added to a training set, the primary classification indices, such as accuracy, specificity, average precision, sensitivity, and balanced multiclass accuracy, increased to 95.2%, 74.3%, 96.6%, 83.2%, and 83.1%, respectively.

3. Materials and Methods

The DeepLesion dataset [17] comprises 32,120 axial CT slices derived from 10,594 CT scans of 4427 individual patients. Each image contains one to three lesions, with its own bounding box and size information, for a total of 32,735 lesions. The lesion annotations were extracted from the NIH's picture archiving and communication system (PACS). There were also some meta-data supplied.

DeepLesion, as stated by Ke Yan et al. [17], is a large-scale dataset comprising diverse lesion types. This dataset can be widely adopted for applications including lesion detection, classification, segmentation, retrieval, measurement, growth analysis, and relationship mining among distinct lesions. Because the utilized dataset only contained the lesion's bounding box information, we built a Symmetry GAN detection network based on it.

3.1. Dataset Analysis

This paper employs a dataset that includes eight types of CT images: abdomen (lesions in the abdominal cavity that are not in the kidney or liver), soft tissue (various lesions in the body wall, such as fat, head, muscle, limbs, neck, and skin), liver, lung, mediastinum, bone, pelvis, and kidney. In Figure 1, it is demonstrated that the dataset has the following characteristics.

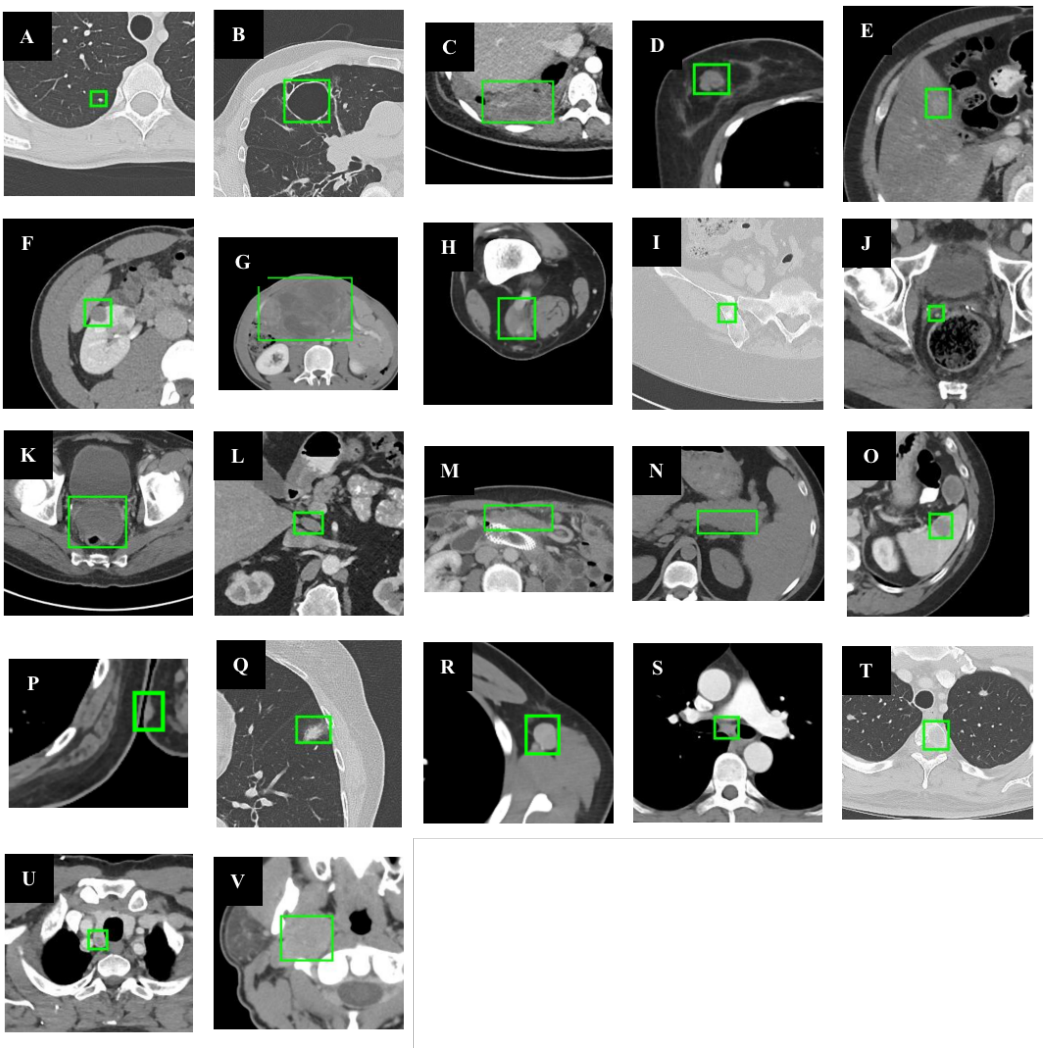

Figure 1. Dataset visualization (the green box marks the location of the lesion). The following sample lesions are displayed to demonstrate the tremendous diversity of the dataset: (**A**) is a lung nodule; (**B**) is a lung cyst; (**C**) is costophrenic sulcus (lung) mass/fluid; (**D**) is a breast mass; (**E**) is a liver lesion; (**F**) is a renal mass; (**G**) is a huge abdominal mass; (**H**) is a posterior thigh mass; (**I**) is an iliac sclerotic lesion; (**J**) is a perirectal lymph node (LN); (**K**) is a pelvic mass; (**L**) is a periportal LN; (**M**) is an omental mass; (**N**) is a peripancreatic lesion; (**O**) is a splenic lesion; (**P**) is a subcutaneous/skin nodule; (**Q**) is opacity of ground glass; (**R**) is an axillary LN; (**S**) is a subcarinal LN; (**T**) is vertebral body metastases; (**U**) is a thyroid nodule; (**V**) is a neck mass.

1. This dataset consists solely of 2D diameter measurements and lesion bounding boxes. It lacks lesion segmentation masks, 3D bounding boxes, and fine-grained lesion types. Hence, some processes—for example, lesion segmentation—might require additional manual annotations.
2. In the images, not all lesions are annotated. Merely representative lesions are generally marked in each study by radiologists. As a result, some lesions go unannotated.

3. In terms of manual examination, while the majority of bookmarks represent aberrant observations or lesions, a tiny bookmark fraction is a approach of classic structures—for example, standard-sized lymph nodes.

Due to the above characteristics of this dataset, various data augmentation methods are adopted in this paper to enhance the model's detection performance.

3.2. Data Augmentation

Fewer samples are involved in medical image datasets, necessitating data augmentation to raise the amount and complexity of training samples. This paper utilizes the following data augmentation strategies to address the insufficient network training. Typically, this kind of insufficient network training is driven by performance degradation induced by overfitting or due to an insufficient dataset.

3.2.1. Basic Augmentation

Conventional image geometry transformation, such as image cutting, rotation, translation, and other operations, can be used for simple data amplification. This research applied the method presented by Alex et al. [45]. In the beginning, each original image is cut into five subgraphs. Subsequently, we flip the five subgraphs horizontally and vertically. The aforementioned process requires a scale operation on the image, which this paper implements by an affine transformation. The target image's width and height are anticipated to be w_{target} and h_{target}, whereas those of the original image are w_{origin} and h_{origin}. Formula (1) illustrates that when images are enlarged and shrunk, the Ω, which represents the scaling factor, is first defined. At that moment, we split the width and height of the original image through Ω. Afterward, after the target frame's center point intersects with that of the processed image, we take a fragment inside the target frame.

$$\Omega = min\{\frac{h_{target}}{h_{origin}}, \frac{w_{target}}{w_{origin}}\} \tag{1}$$

The trimmed training set image was counted by outsourcing frames to avoid some of the outsourcing frames being cut out, and then HSV channel color change was carried out [46]. In this case, every original image generated 15 extended images.

3.2.2. Advanced Augmentation

We allude to a method demonstrated in the Mixup [47] and then present a series-Mixup data augmentation method with CT image series, tackling the large memory loss and the network's inadequate sensitivity to symmetry GANs. Formulas (2)–(4) show the method.

$$\lambda = Beta(\alpha, \beta) \tag{2}$$

$$mixed\_series_x = \lambda \times series_{x1} + (1 - \lambda) \times series_{x2} \tag{3}$$

$$mixed\_series_y = \lambda \times series_{y1} + (1 - \lambda) \times series_{y2} \tag{4}$$

$series_{x1}$ is a series sample, and $series_{y1}$ is the label matching the series sample. $series_{x2}$ is another series sample, $series_{y2}$ denotes the label corresponding to the series sample, and λ represents the mixing coefficient calculated through the *Beta* distribution of parameters α and β. When this study implements the method, there is no restriction on $series_{x1}$ and $series_{x2}$. When the series size is one, two images are mixed. When the series size is greater than one, it means that two series image samples are mixed consequently. Additionally, $series_{x1}$ and $series_{x2}$ can be either the identical series of samples or different series of samples. When implementing this method, $series_{x1}$ and $series_{x2}$ adopt the same series of samples. Among them, $series_{x1}$ is the original series image sample, and $series_{x2}$ is obtained after shuffle processing of $series_{x1}$ in the dimension of series size.

Furthermore, to prevent overfitting of the network, we undertake a random erase operation on image data before they are sent to the backbone network. This method's function is similar to the dropout function [45]. Because the portion and location erased are

random for every round of training, the network's robustness can be improved, and the erased section can be considered as the blocked or distorted portion. Filling pixels with a predetermined color, such as black, or filling with the RGB channel mean of all pixels in the erased region are the two options for processing the erased section. The above-mentioned effects are depicted in Figure 2.

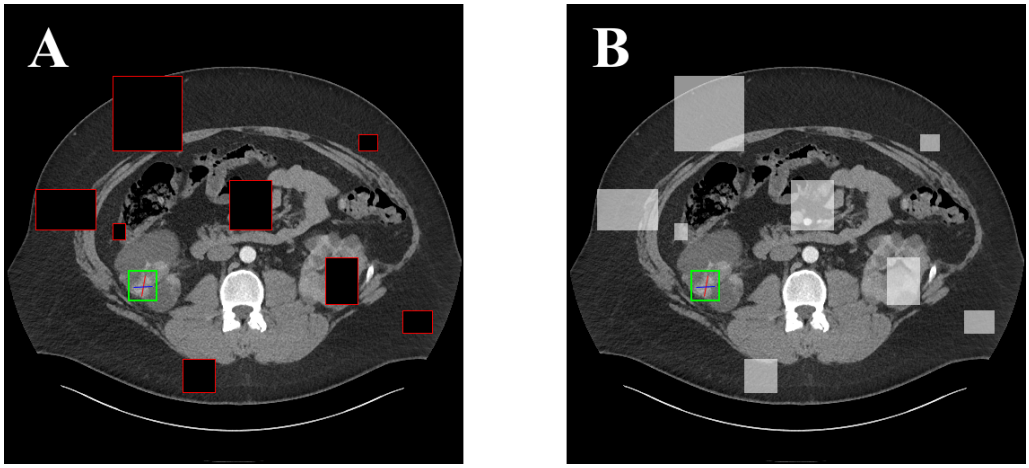

Figure 2. Examples of two types of CutOut fillings. (**A**) Filling with black pixels; (**B**) Filling with the average of surrounding pixels in the erased portion.

CT images are sparse, while the lesions in each image are insufficient. In order to maximize the backbone learning of lesion features, i.e., positive sample features, we borrowed the idea of CutMix [48]—we cut and pasted the lesion area to other background areas. Thus, the model learning of positive features in unbalanced samples can be enhanced, and the model's performance can also be improved.

In addition to the above, we also use the Mosaic [49] method. This method might employ numerous images at the same time. The most notable merit of this method is that it can embellish the discovered objects' backgrounds. The above data augmentation methods are used to maximize the robustness and detection performance of the model. Figure 3 shows the effect of applying these methods.

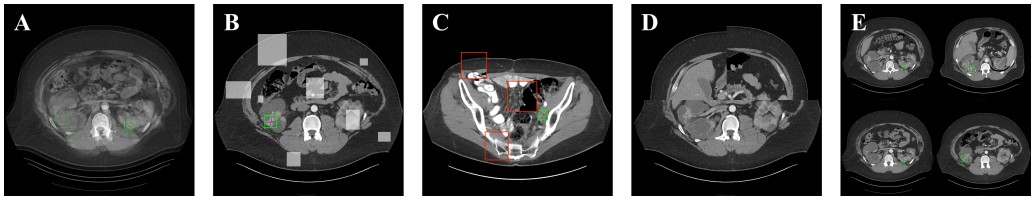

Figure 3. Illustration of different data augmentation methods. (**A**) series-Mixup; (**B**) Random-erase; (**C**) CutMix; (**D**) Mosaic; (**E**) source images.

3.3. Symmetry GAN Detection Network

Mainstream one-stage object detection models, such as YOLO [25,26,50,51] and SSD [52], have achieved excellent performance on the MS COCO [53] and Pascal VOC [54] datasets and are widely used in target detection tasks. However, since the anchor parameters of YOLO series do not match the actual CT images, the performance of the model obtained by directly training YOLO series is not good. The main reasons are as follows: the YOLO and SSD algorithms are mainly trained based on the MS COCO and Pascal VOC datasets, so the anchor points in the algorithm are not universal, especially the low target detection accuracy of small objects. Therefore, based on the idea of a one-stage network, a symmetry GAN detection network was proposed, which has a network structure based on one-stage detection networks and GAN and is mainly suitable for CT images.

Compared with mainstream one-stage detection networks, the main differences of the symmetry GAN detection network are as follows:

1. The GAN-based image generation network is added before the backbone network, and the GAN-based attention extraction module is added to the attention module, forming symmetric GANs.
2. The activation function is improved, and this paper replaces the single activation function with parallel multi-activation functions—for instance, LeakyReLU—improving the model's performance.
3. Using concepts from the feature fusion network (FPN) and the path aggregation network (PANet) [55], this paper adds multi-scale feature fusion modules to the backbone and improves the modules.
4. This paper optimizes the loss functions and develops specific loss functions for the lesion and background image recognition modules.
5. This paper additionally adds a label smoothing function at the backbone network's output, preventing classification overfitting.
6. To estimate the confidence threshold of detecting frame discarding, this paper adopted the out-of-fold (OOF) model cross-validation method [56].

Figure 4 illustrates the structure of the symmetry GAN detection network.

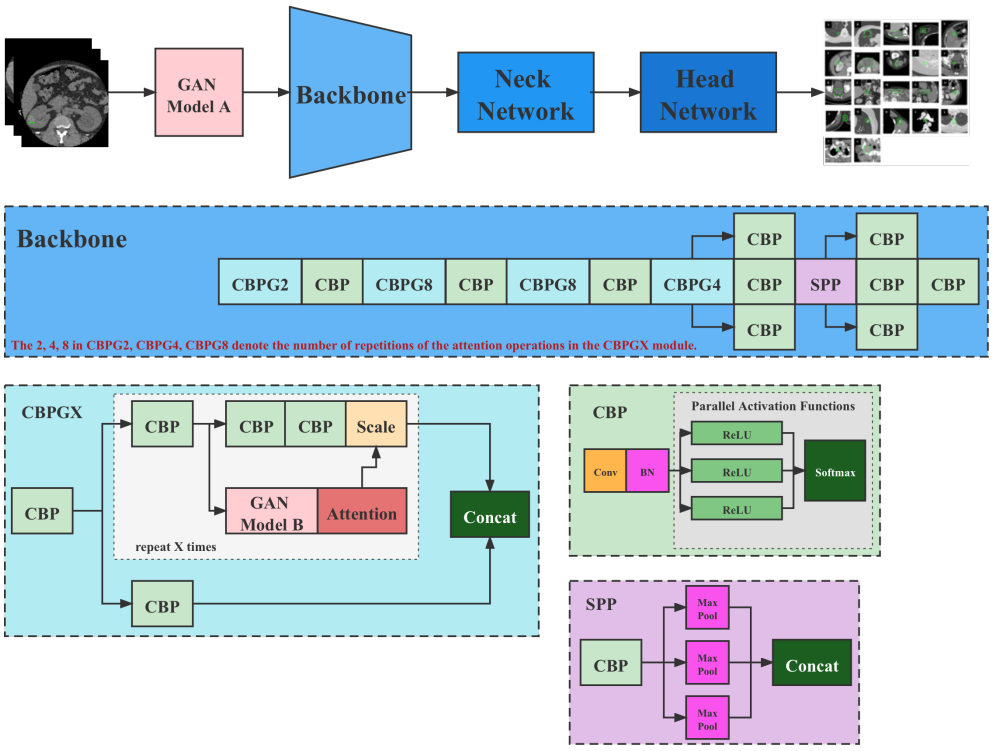

Figure 4. Flow chart of symmetry GAN detection network.

3.3.1. Symmetry GANs

Symmetry GANs comprise two GAN modules: the first one, the GAN model A in Figure 4, is located ahead of the backbone, which is used for expanding CT images. There are various ways to implement it. As an example, the algorithm flow of GAN model A is illustrated in Algorithm 1.

Algorithm 1 Algorithm flow of GAN Model A.

1: **Input**: dataset D
2: **Output**: dataset D'
3: Step 1: input the randomly generated data with Gaussian distribution to the Generator
4: Step 2: train Generator
5: Step 3a: input the data generated by Generator to Discriminator
6: Step 3b: input original data to Discriminator
7: Step 4: train Discriminator
8: Step 5: repeat the above steps until the discriminator cannot distinguish the generated data from the real data
9: Step 6: output real data and generated data

The generator is employed to generate more feasible eigenvectors matching with lesion images to improve the training. Consider DCGAN, which has two participants: the discriminator D and generator G. Let p_{data} be the retrieved eigenvectors' distribution. The target of generator model G is to construct a probability distribution p_g on the feature map x. This distribution is the estimated value of p_{data}. Two deep neural networks expound the discriminator and generator. Formula (5) expresses the DCGAN model's optimization purpose:

$$\min_G \max_D V(D,G) = \mathbb{E}_{(x\sim p_{data})}[logD(x)] + \mathbb{E}_{(z\sim p_{z(z)})}[log(1-D(G(z)))] \quad (5)$$

In Formula (5), x is a prior value of an input noise variable. During the training process, two deep neural network models are trained. The discriminative model is matched against the generative model G. In other words, these models will improve their objective functions by playing games. Nonetheless, in order to avoid the difficulties of identifying the exact Nash balance in real-world cases, we take the accuracy of the data generated in discriminator D as a stopping criterion. It specifies that if the misclassified probability of the data generated by G reaches a predetermined level, the training will be discontinued. Figure 5 displays the training process.

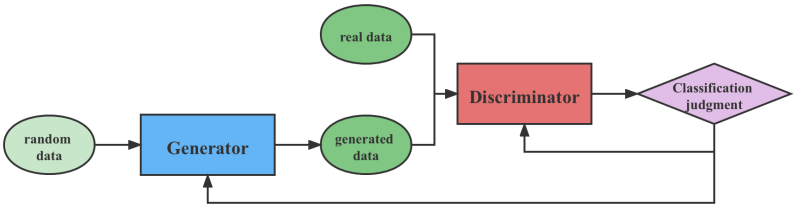

Figure 5. Flow chart of GAN model A.

The second GAN module, GAN model B in Figure 4, is located in the attention mechanism module. Its primary role is to add a noise mask to the feature maps obtained from the backbone to improve the model's robustness. From the subsequent results—shown in Section 5—it is observed that adding noise can significantly improve the model's performance. The GAN module of this section can also be implemented in various ways. For example, SAGAN can be applied as shown in Figure 6.

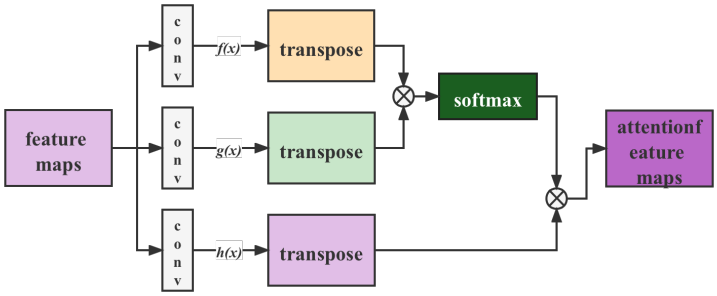

Figure 6. Flow chart of GAN Model B.

Figure 7 shows the visualization of the feature maps and attention feature maps.

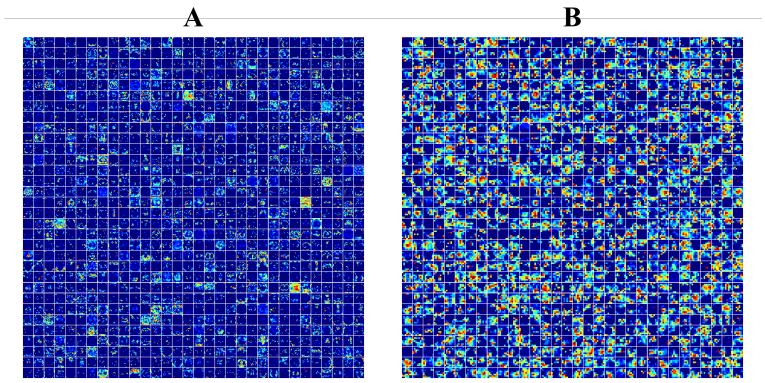

Figure 7. Comparison of feature maps with attention feature maps. (**A**) Feature maps; (**B**) attention feature maps.

3.3.2. Parallel Activation Functions

In the existing backbone networks, only the activation function layers are connected in series between the layers in the network, dominated by ReLU and LeakReLU. The parallel activation functions module proposed in this paper transforms the series activation function layers into parallel multiple-activation function layers, in which each base activation is preceded by a coefficient k_n. We guarantee $\sum_{i=1}^{n} k_i = 1$ so that the effect of ensembling multiple CNN models can be simulated by this parallel structure.

This paper selects the following types of base activation functions to implement the parallel activation functions module:

1. ReLU. The activation function employed in the above numerous backbone networks uses the ReLU function by default, first applied in the AlexNet network.
2. Mish [57]. Mish, proposed by Diganta Misra, is an activation function built to take the place of ReLU. It was reported that it surpassed a portion of the previous FastAI global leaderboard accuracy score record.
3. Sigmoid. Sigmoid is a smooth step function; the function can be derived. Sigmoid can change any value into $[0, 1]$ probability, primarily adopted for binary classification tasks.

CNNs have been developed for many years and produced numerous model structures, which can be classified into three kinds: the network structure formed by repeatedly stacking the convolutional layer–activation function layer–pooling layer represented by AlexNet [45] and VGG series [58]; the residual network structure model represented by ResNet series [59] and DenseNet series [60], and the multi-branch parallel network structure represented by GoogLeNet [61]. Figure 4 shows how to apply the parallel activation functions module to different kinds of backbones.

3.4. Loss Function

The symmetry GAN detection network's loss function is composed of three portions: box coordinate error, $CIoU$ error, and classification error, as shown in Formulas (6)–(9). The box coordinate error (x_i, y_i) denotes the predicted box's center position coordinate, and (w_i, h_i) is its width and height. $(\hat{x}_i, \hat{y}_i)$ and $(\hat{w}_i, \hat{h}_i)$ denote the coordinates and size of the labeled ground truth box, respectively. Furthermore, λ_{coord} and λ_{noobj} are constants. $K \times K$ represents the grids' amount. M expounds the predicted boxes' overall amount. Moreover, I_{ij}^{obj} is one when the ith grid detects a target and zero otherwise.

$$Loss = Loss_{bounding\_box} + Loss_{ciou} + Loss_{classification} \tag{6}$$

$$\begin{aligned} Loss_{bounding\_box} =& \lambda_{coord} \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} (2 - w_i \times h_i)[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\ & \lambda_{coord} \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} (2 - w_i \times h_i)[(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2] \end{aligned} \tag{7}$$

$$\begin{aligned} Loss_{ciou} =& \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} [\hat{C}_i log(C_i) + (1 - \hat{C}_i log(1 - C_i)] + \\ & \lambda_{noobj} \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{noobj} [\hat{C}_i log(C_i) + (1 - \hat{C}_i log(1 - C_i)] \end{aligned} \tag{8}$$

$$Loss_{classification} = \sum_{i=0}^{K\times K} I_{ij}^{obj} \sum_{c\in classes} [\hat{p}_i(c) log(p_i(c)) + (1 - \hat{p}_i(c) log(1 - p_i(c))] \tag{9}$$

Zheng [62] suggested a more high-efficiency IoU calculation method, $CIoU$. Formula (10) demonstrates $CIoU$'s calculation formula.

$$CIoU = 1 - IoU + \frac{\rho^2(\boldsymbol{A}, \boldsymbol{B})}{c^2} + \alpha v \tag{10}$$

The model's classification categories are divided into two types: positive and negative. The prediction box and its IoU are computed in every ground truth box. Specifically, the greatest IoU is a positive class, whereas the others are negative.

3.4.1. Label Smoothing

The backbone network of the symmetry GAN detection network outputs a confidence score for the current data corresponding to the foreground, i.e., wheat. The $softmax$ function normalizes these scores, and, ultimately, the probability of each category that the current data belongs to is obtained. The calculation formula is shown in Formula (11).

$$q_i = \frac{exp(z_i)}{\sum_{j=1}^{K} exp(z_j)} \tag{11}$$

Then, the cross-entropy cost is calculated:

$$Loss = -\sum_{i=1}^{K} p_i log q_i \tag{12}$$

where

$$p_i = \begin{cases} 1, \ if(i = y) \\ 0, \ if(i \neq y) \end{cases} \tag{13}$$

The predicted probability should be adopted for the loss function to fit the true probability. However, fitting the one-hot true probability function would bring problems as follows:

1. The generalization ability of the model cannot be ascertained, and it is likely to lead to overfitting.
2. The gap of categories that encouragements of full probability and zero probability belong to and other categories are as large as possible. Furthermore, the bounded gradient indicates that this situation is challenging to fit. It would cause the model to trust the predicted category heavily. In particular, it would contribute to the network model's overfitting so that the training data are not sufficient to represent all of the sample features.

The regularization strategy of label smoothing is adopted to tackle the aforementioned barriers. This strategy includes adding noise via soft one-hot and decreasing the weight of the real sample label category in the loss function's computation. It plays a role in suppressing overfitting.

After label smoothing is added, the probability distribution changes from Formula (13) to Formula (14):

$$p_i = \begin{cases} 1-\epsilon, \ if(i=y) \\ \dfrac{\epsilon}{K-1}, \ if(i \neq y) \end{cases} \tag{14}$$

3.4.2. Out-of-Fold mAP Threshold Calculation

After the symmetry GAN detection network generates prediction boxes, it is necessary to discard the boxes where the mAP score is below the confidence threshold before the non-maximum suppression (NMS) algorithm. However, the setting of this threshold usually depends on manual experience. This paper uses the out-of-fold to determine the mAP threshold of the retention or discarding prediction box. The core idea of out-of-fold is to calculate the mAP of the verification set by traversing different thresholds and then obtain the optimal threshold value that maximizes the score of the mAP in the traversing process.

4. Experiment

4.1. Evaluation Metrics

To validate the model's performance, four metrics are used for the evaluation in this paper, namely mAP, precision (P), recall (R), and FPS. The Jaccard index, commonly known as the intersection over union (IoU), is specified as the intersection of predicted segmentation, which also divides the label. The value of this indicator ranges from 0 to 1: 0 indicates no overlap, and 1 represents complete overlap. It is a true situation when the $IoU \geq$ is 0.5; otherwise, it is a false positive situation. The binary classification calculation formula is:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \tag{15}$$

where A denotes ground truth and B is the predicted segmentation.

Pixel accuracy (PA) is the percentage of an image's accurately classified pixels, i.e., the proportion of correctly classified pixels to entire pixels. The formula is as follows:

$$PA = \frac{\sum_{i=0}^{n} p_{ii}}{\sum_{i=0}^{n} \sum_{j=0}^{n} p_{ij}} = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

n indicates the total amount of categories; $n+1$ represents the category amount, containing backdrops. p_{ii} indicates the overall amount of real pixels, in which the label is i and predicted to be class i, i.e., the entire amount of matched pixels for real pixels (class i). p_{ij} expounds the overall amount of real pixels (label i) that are predicted to be class j, which can be regarded as the amount of pixels (label i) that are classified into class j incorrectly. Moreover, TP denotes the amount of true positives (positive in both labels and predicted value). TN expounds the amount of true negatives (negative in both labels and predicted value). FP is the amount of false positives (negative in label and positive in predicted

value). FN describes the amount of false negatives (positive in label and negative in predicted value). In addition, $TP + TN + FP + FN$ specifies the overall amount of pixels, and $TP + TN$ specifies the amount of pixels that are correctly classified.

Mean pixel accuracy (*mPA*) is a straightforward improvement on *PA*. *mPA* computes the percentage of pixels precisely recognized in every class and averages the outcomes, as indicated in Formula (17).

$$mAP = \frac{\sum_{i=1}^{k}(AP_i)}{k} \tag{17}$$

Precision (P) is the percentage of samples categorized as positive samples among the accurately classified samples.

$$P = \frac{TP}{TP + FP} \tag{18}$$

Recall (R) demonstrates the percentage of correctly categorized positive samples among overall positive samples.

$$R = \frac{TP}{TP + FN} \tag{19}$$

4.2. Experiment Setting

A personal computer (CPU: Intel(R) i9-10900KF; GPU: NVIDIA RTX 3080 10 GB; Memory: 16 GB; OS: Ubuntu 18.04, 64 bits) was used to carry out the entire model training and validation process. We chose the Adam optimizer with an initial learning rate, $a0 = 1 \times 10^{-4}$. The learning rate increment was adjusted using the method specified in Section 4.3 and the training speed was optimized.

4.3. Learning Rate

Warm-up [59] is a training strategy. During the pre-training phase, one trains certain epochs or steps at a low learning rate, such as four epochs or 10,000 steps. Then, these epochs are changed into a predefined learning rate for training. We randomly assign the model weights when training starts, and the model's "level of understanding" of the data is set to zero. Assuming that a higher learning rate is utilized initially, the model may fluctuate. Warm-up adopts a comparatively reduced learning speed for training to supply the model with the data's prior knowledge. Afterward, during training, we utilize the predefined learning speed to enhance the model's convergence rate and efficacy. Ultimately, utilizing a low learning rate to continue with exploration avoids losing local best points. In the training procedure, for instance, we set the learning speed as 0.01 to train the model until the error was no more than 80%. Then, we set the learning speed to 0.1 for training.

The above warm-up is the constant warm-up. Its downside is that switching from a low learning speed to a comparatively high one might induce the training error to skyrocket. As a result, Facebook advocated for a gradual warm-up to address this issue in 2018. It begins with a very low learning rate and gradually increases until it reaches the relatively high, initially established learning rate, at which point it is used to conduct training.

The *exp* warm-up method is examined in this article, which involves linearly accelerating the learning from a minuscule value to the predefined learning speed and then fading in terms of the *exp* function law. Meanwhile, *sin* warm-up is explored, which increases the learning rate linearly from a low value. It decays according to the *sin* function rule once it reaches the predetermined value. Figure 8 depicts the changes between the two pre-training methods.

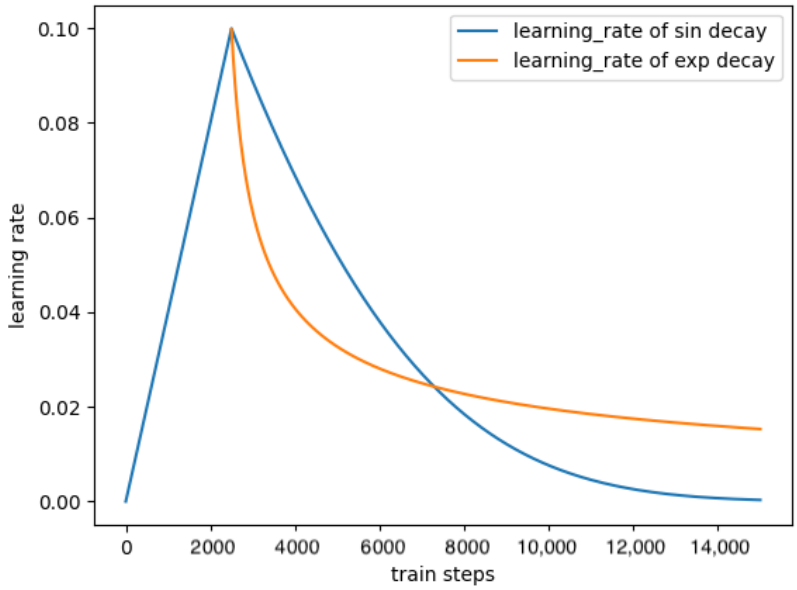

Figure 8. Warm-up learning rate schedule.

4.4. Pseudo-Label Training Enhancement

Because the size of the medical dataset is insufficient, the pseudo-label method is adopted to fully utilize the test set data to improve training. Three pseudo-label methods are tested, as shown in Figure 9. Among them, M represents a supervised model trained with labeled data, and M' denotes a model trained with labeled data and pseudo-labeled data. Pseudo-label model B uses M' to replace M and repeats until the model effect does not improve, as shown in Figure 9.

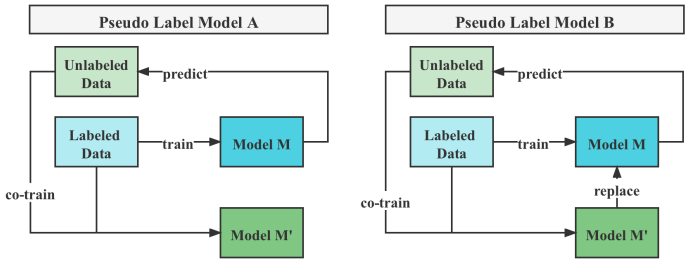

Figure 9. Flow chart of two pseudo-label models.

5. Results

5.1. Validation Results

The experimental results presented in this section refer to the test set after randomly segmenting the dataset into a training set and a test set with a ratio of 9:1. Table 1 contains the experimental results. The best results of the index are given in bold. In Table 1, YOLO v5 [27] demonstrates the best speed. The P, R, and mAP of Faster-RCNN [63] are 0.8022, 0.8519, and 0.8396, which show the worst performance of all models. These P, R, and mAP values of YOLO v5 are superior to those of Faster-RCNN, Mask-RCNN [23], and SDD series, whose values are 0.9446, 0.9718, and 0.9674, respectively. Although EfficientDet outperforms YOLO v5 in terms of precision, it does not perform as satisfactorily as YOLO v5 in terms of recall and mAP. This is probably due to the stronger performance of the attention extraction module in EfficientDet than in YOLO v5. Overall, EfficientDet and YOLO v5 are the two best models among the comparisons. We split the input into two transmission SGDNs of 300 $\times$ 300 and 500 $\times$ 500 for testing, and the results show that the latter has better performance, with the three parameters reaching 0.9720, 0.9858, and 0.9833, respectively, which are higher than those of YOLO v5 and EfficientDet. However, our model ranks third in terms of the FPS index. This is caused by the complexity of

the symmetry GAN module. As depicted above, SGDN 512 shows the best detection performance on the DeepLesion dataset, according to the outcomes.

Table 1. Comparison of variant models and SGDN.

Method	*P*	*R*	*mAP*	FPS	Batch Size	Input Resolution
Faster-RCNN	0.8022	0.8519	0.8396	7	2	600 × 600
Mask-RCNN	0.9314	0.9499	0.9493	9	2	600 × 600
EfficientDet	0.9512	0.9688	0.9520	17	8	512 × 512
YOLO v3	0.9317	0.9513	0.9506	23	2	608 × 608
YOLO v4	0.9228	0.9497	0.9485	27	2	608 × 608
YOLO v5	0.9446	0.9718	0.9674	**51**	2	608 × 608
SSD 300	0.9281	0.9489	0.9460	32	8	300 × 300
SSD 512	0.9374	0.9672	0.9477	29	2	512 × 512
SGDN 300	**0.9771**	0.9839	0.9825	18	8	300 × 300
SGDN 512	0.9720	**0.9858**	**0.9833**	13	2	512 × 512
SGDN 512	0.9688	0.9840	0.9831	13	8	512 × 512

The model fusion method is then adopted to enhance the performance of our model. The model fusion method is simple because it calculates the intersection of the results of multiple models directly. In this paper, the model fusion method is adopted to incorporate the different SGDT models, as shown in Table 2.

Table 2. Results of model fusion.

Models	OOF	NMS Method	*P*	*R*	*mAP*
SGDT 300		soft NMS	**0.9771**	0.9839	0.9825
SGDT 512		NMS	0.9715	0.9763	0.9756
SGDT 512	+	soft NMS	0.9720	0.9858	0.9833
SGDT 300 + SGDT 512	+	WBF	0.9719	**0.9883**	**0.9871**

The experimental results show that the *mAP* obtained when fusing the SGDT 300 and SGDT 512 models is 0.9871, which is already higher than that of other detection models.

5.2. Detection Results

For further comparison, we extracted six images from the CT image series of DeepLesion. These images were taken from different sites of lesions and different areas of lesions, showing the detection results of the comparison model as comprehensively as possible. Figures 10–19 show the detection results. All green boxes represent ground truth; red boxes denote predicted bounding boxes. It can be seen that Faster-RCNN performs very poorly on small lesions and lesions that are not easy to identify, while YOLO v3, YOLO v4, and SSD series perform relatively well. However, the aspect regression of the bounding box at small lesion locations is still not accurate. On the other hand, EfficientDet, Mask-RCNN, and YOLO v5 perform relatively well and detect lesions accurately. This may be related to the attention extraction module in these networks.

Our model, especially SGDN 512, outperforms the previous models by detecting lesions with high accuracy for non-minimal lesions. Although there is still room for improvement, it has outperformed other models. On the one hand, we augment the image with the GAN model before it is fed into the backbone. On the other hand, we add the GAN model to the attention extraction module of the model, which can significantly improve the model's robustness.

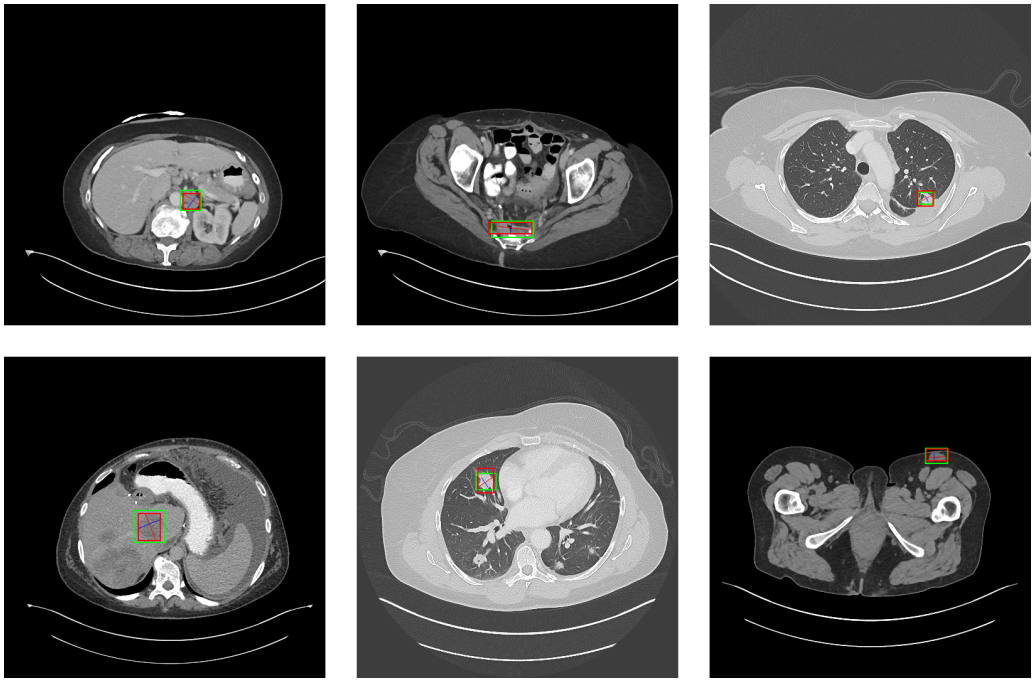

Figure 10. The detection results of YOLO v3 in the DeepLesion dataset. The green box marks the location of the lesion.

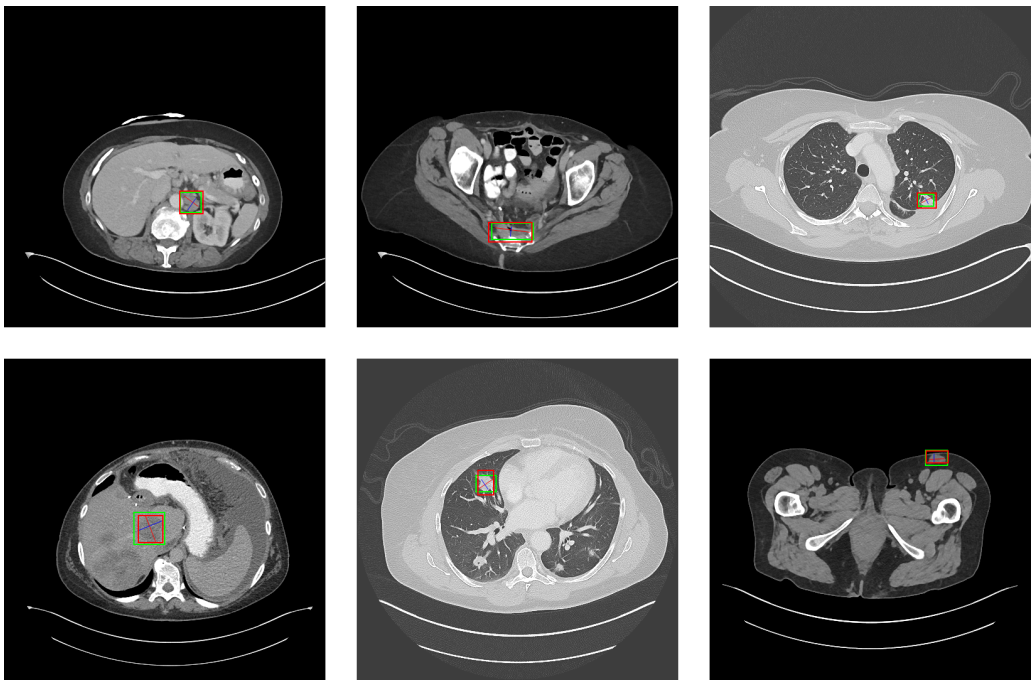

Figure 11. The detection results of YOLO v4 in the DeepLesion dataset. The green box marks the location of the lesion.

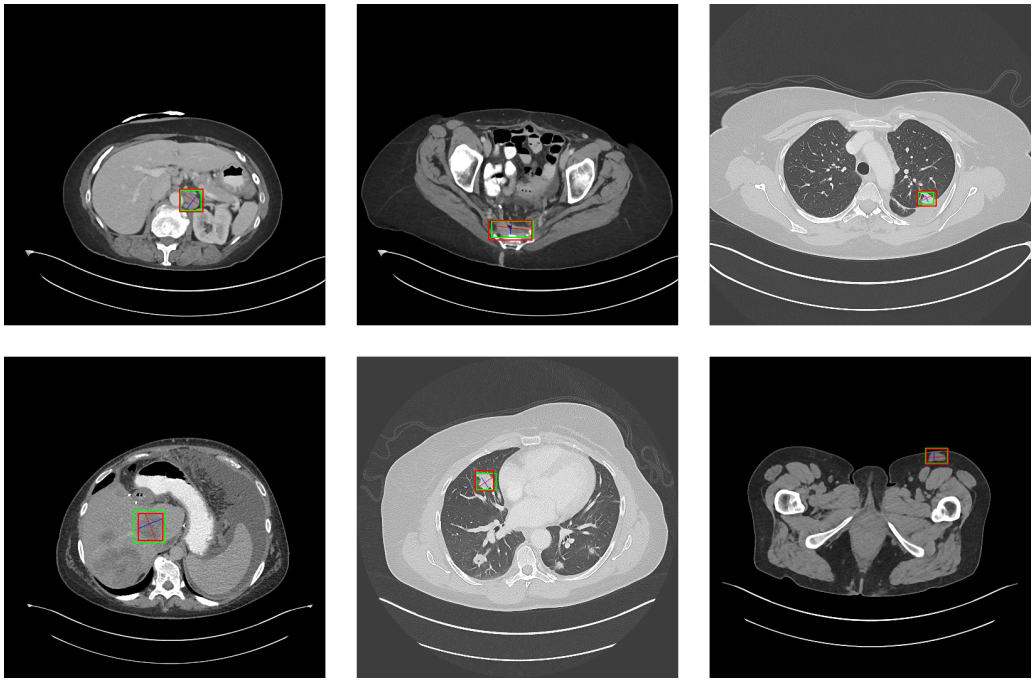

Figure 12. The detection results of YOLO v5 in the DeepLesion dataset. The green box marks the location of the lesion.

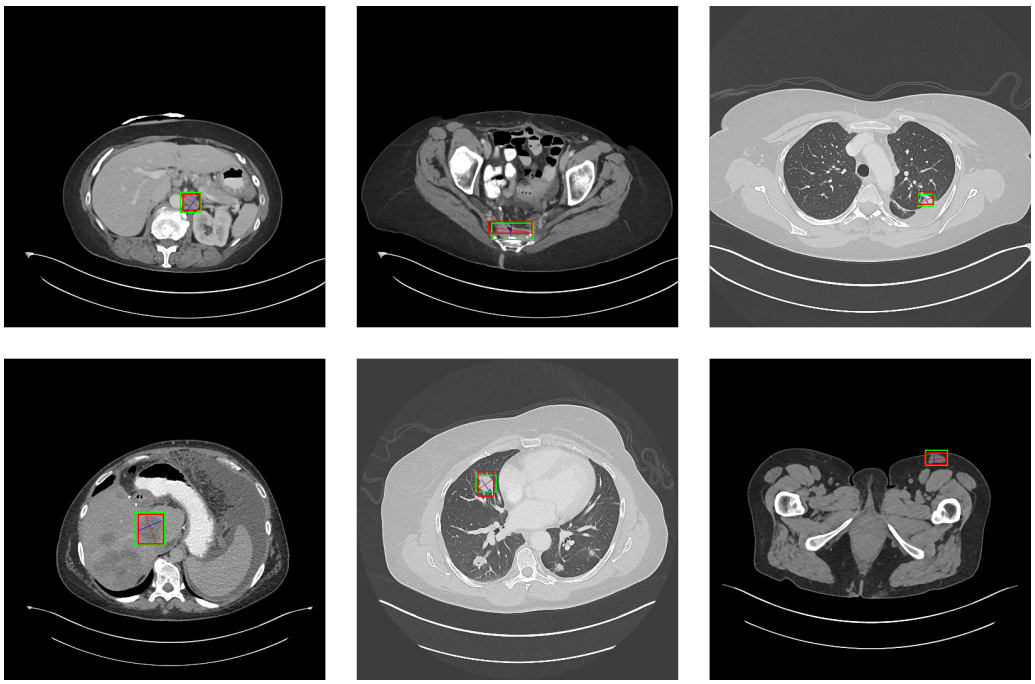

Figure 13. The detection results of Faster-RCNN in the DeepLesion dataset. The green box marks the location of the lesion.

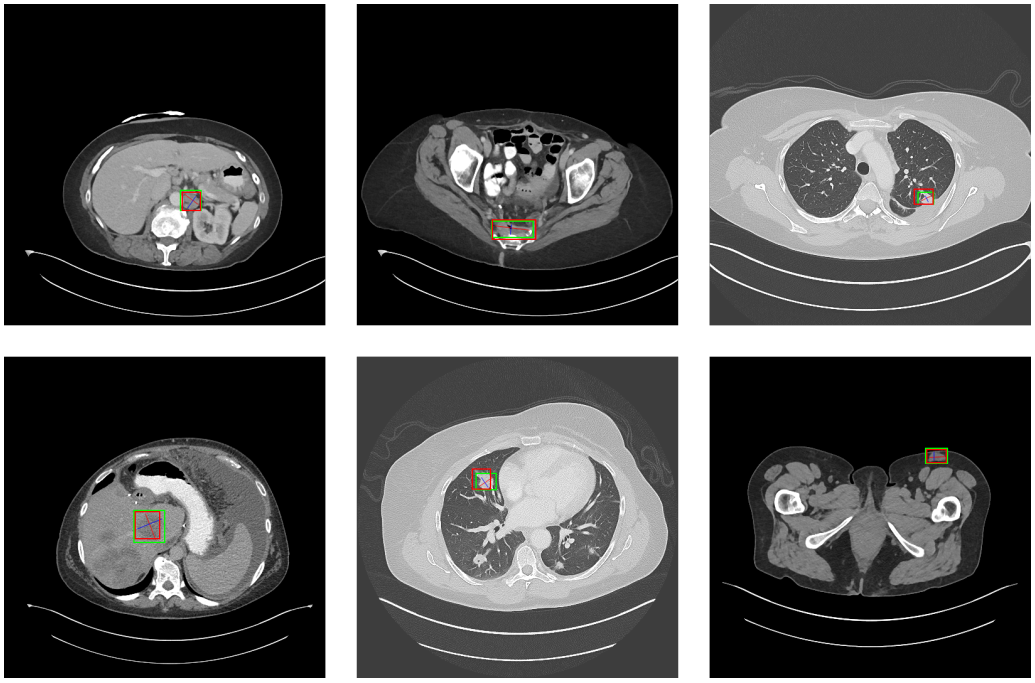

Figure 14. The detection results of Mask-RCNN in the DeepLesion dataset. The green box marks the location of the lesion.

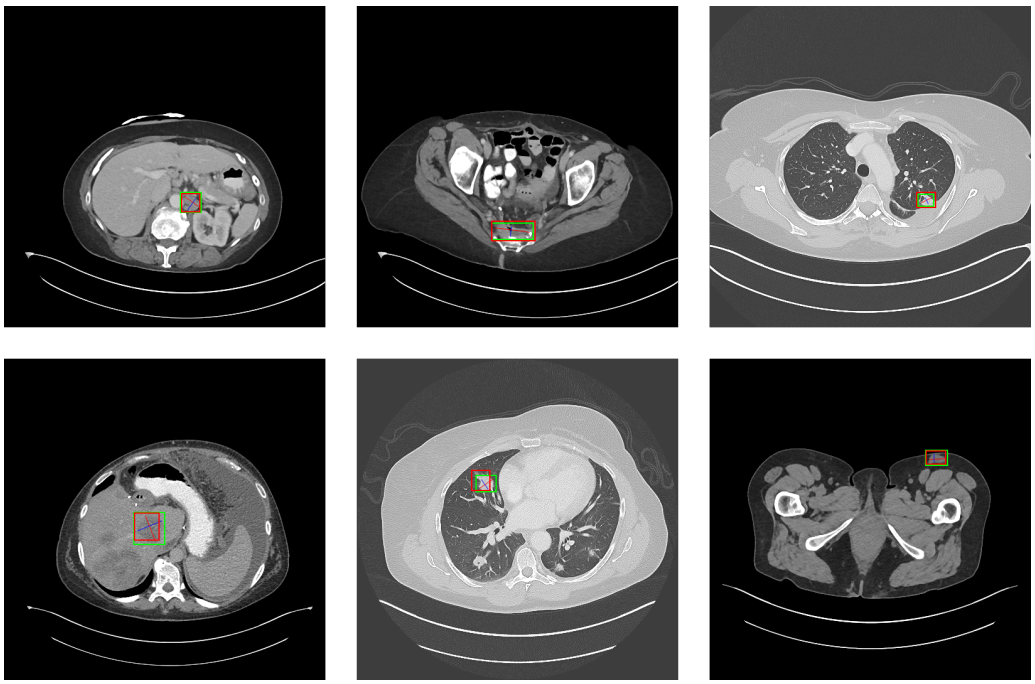

Figure 15. The detection results of EfficientDet in the DeepLesion dataset. The green box marks the location of the lesion.

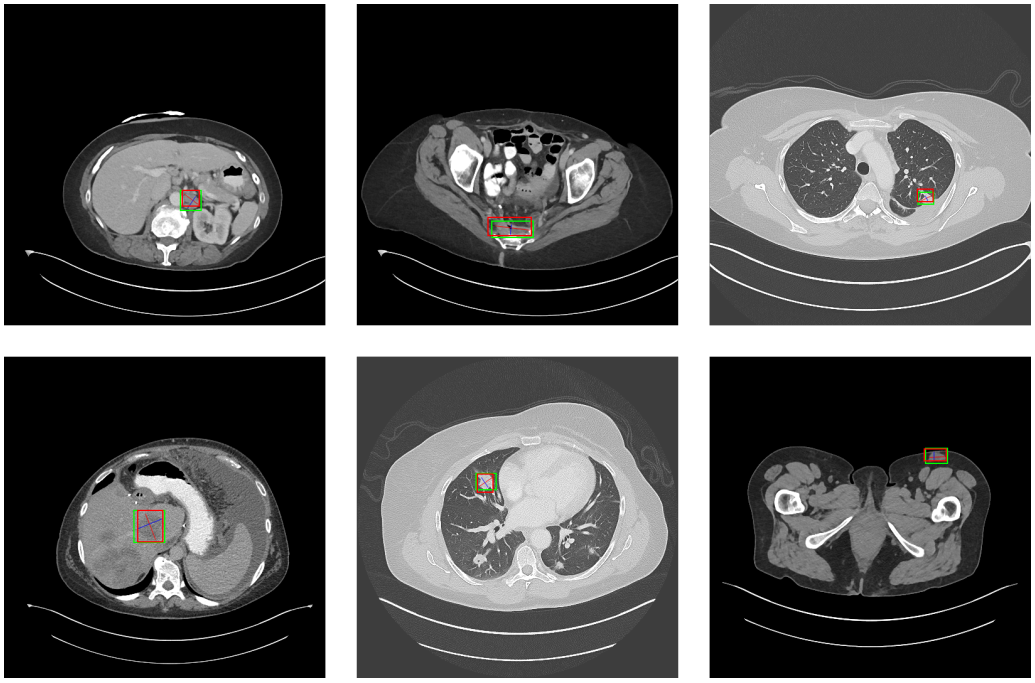

Figure 16. The detection results of SSD 300 in the DeepLesion dataset. The green box marks the location of the lesion.

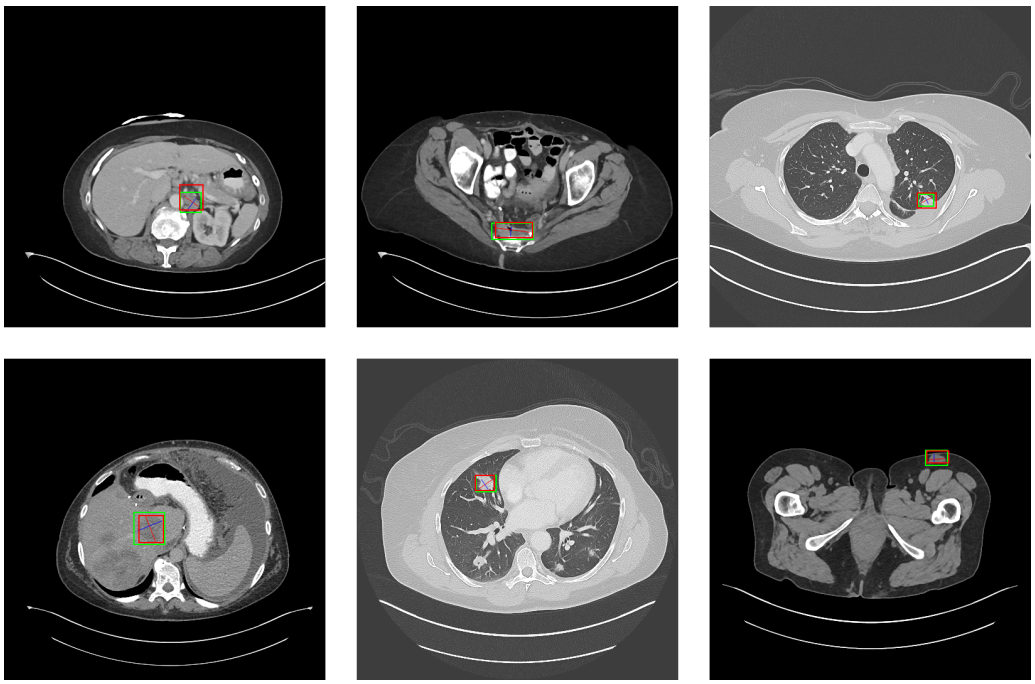

Figure 17. The detection results of SSD 512 in the DeepLesion dataset. The green box marks the location of the lesion.

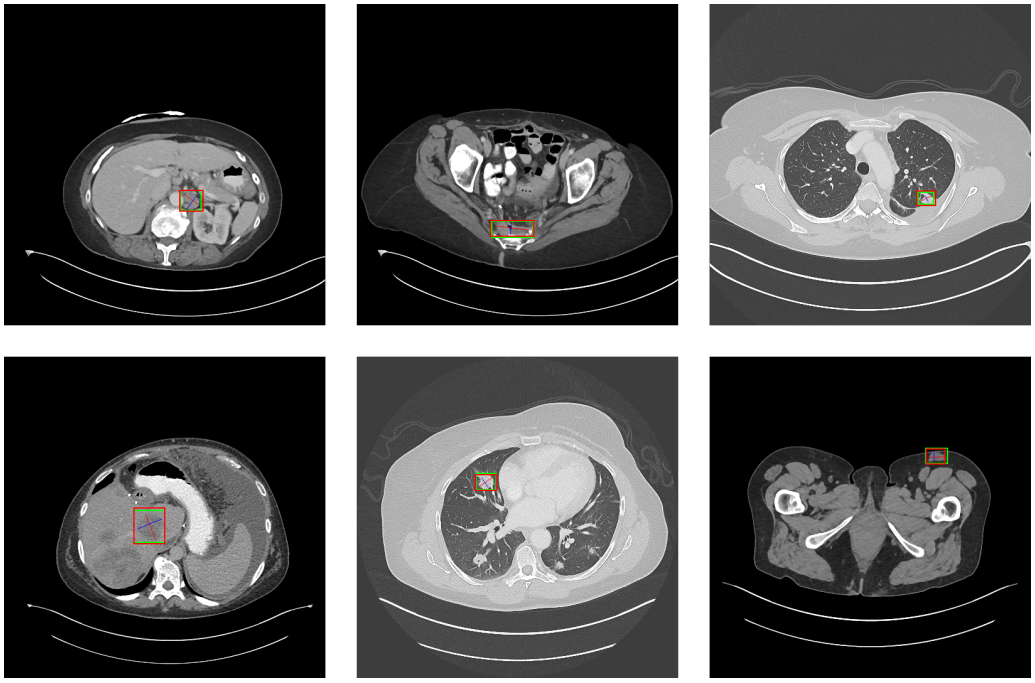

Figure 18. The detection results of SGDN 300 in the DeepLesion dataset. The green box marks the location of the lesion.

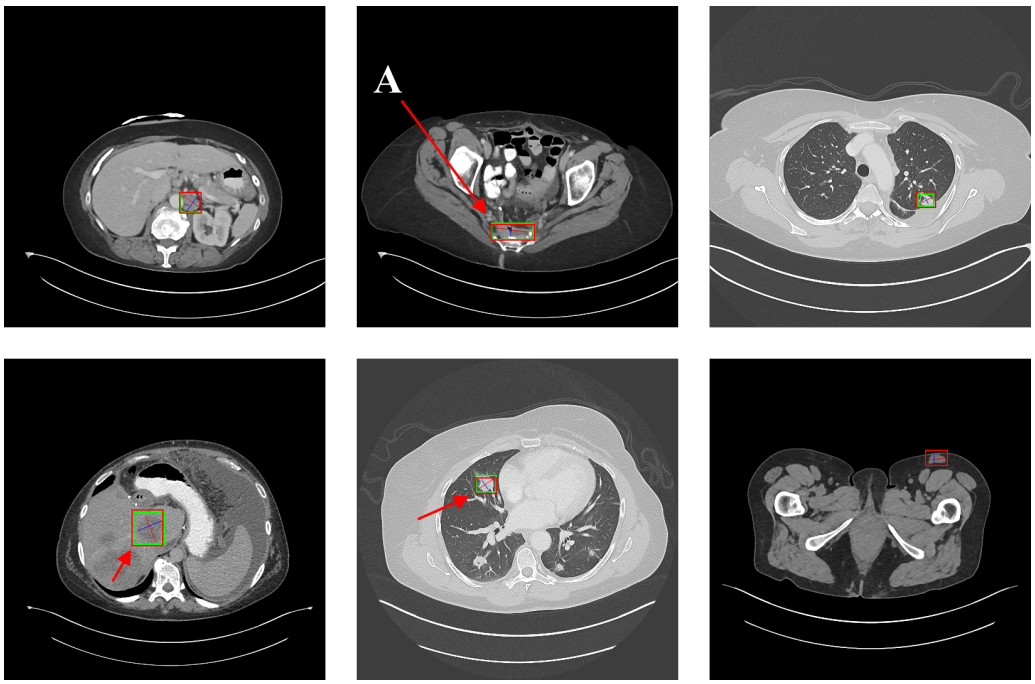

Figure 19. The detection results of SGDN 512 in the DeepLesion dataset. The green box marks the location of the lesion.

According to Figures 10–19, the proposed model produces the most comprehensive detection results compared to other models. However, there are still a few cases where the shortcomings of SGDN can be seen: the arrows in Figure 19 show that our model is still not accurate at the edge of the lesion. In addition, from these figures, we can see that all the comparison models perform very poorly at the site of arrow A. The difference between the predicted box and the ground truth given by our model at arrow A is the largest compared to other recognition results.

6. Discussion

6.1. Ablation Experiment of Symmetry GANs

This paper uses GAN modules in backbone and attention extraction modules, while GAN models have many branches and foci. The primary purpose of the GAN module in front of the backbone is to enhance the model input. In contrast, the GAN module in the attention extraction module generates an attention mask to enhance the model's robustness. Therefore, for the two GAN modules with different purposes, different GAN models are implemented in this paper, including DCGAN, CVAE-GAN, SAGAN, and SPA-GAN. Several ablation experiments were conducted, and the experimental results are illustrated in Table 3.

Table 3. Results of different implementations of symmetry GAN module on SGDN 512.

Method	*P*	*R*	*mAP*	FPS
No symmetry GANs (baseline)	0.8375	0.8749	0.8526	29
DCGAN + SAGAN	0.9720	0.9858	0.9833	13
CVAE-GAN + SAGAN	0.9691	0.9731	0.9603	11
DCGAN + SPA-GAN	0.9737	0.9845	0.9841	11
CVAE-GAN + SPA-GAN	0.9741	0.9745	0.9742	8

As Table 3 illustrates, using DCGAN and SPA-GAN to implement GAN model A and GAN model B, respectively, can optimize the models' performance, with the three primary metrics reaching 0.9737 and 0.9845. As a comparison, DCGAN is better than CVAE-GAN in the choice of GAN model A. Regardless of the implementation of GAN model B, this may be due to the insufficient depth of the network in CGAN, resulting in ineffective training of the generator and discriminator. By comparing the baseline model, it is apparent that the symmetry GAN module, regardless of the implementation approach, can significantly improve the model's performance by 12.5% in terms of the mAP parameter.

6.2. Ablation Experiment of Data Augmentation Methods

Conventional data augmentation methods are utilized in computer vision applications, including random crop, flip, and translation. However, this work employs sophisticated data augmentation methods such as random erasure and image mixing. We conducted ablation experiments to validate the improved efficacy of various strategies on model performance. Because the four data augmentation methods, random-erase, CutMix, series-MixUp, and Mosaic, entail higher computational complexity than affine transformation-based methods, they exert a more significant effect on the model's training and inference speed. We evaluated the impacts of various incorporations to see if it was beneficial to utilize these strategies. Furthermore, we investigated whether it was viable to employ exclusively affine transformation-based augmentation approaches. Table 4 displays the experimental results.

Table 4. Results of different data augmentation methods.

Random-Erase	CutMix	Series-MixUp	Mosaic	*P*	*R*	*mAP*	FPS
✓	✓	✓	✓	0.9720	0.9858	0.9833	13
	✓	✓	✓	0.9715	0.9861	0.9829	15
✓		✓	✓	0.9711	0.9847	0.9834	13
✓	✓		✓	0.9722	0.9861	0.9808	13
✓	✓	✓		0.9718	0.9822	0.9817	13
				0.9328	0.9356	0.9272	16

Table 4 shows that every data augmentation method can satisfactorily enhance the model's performance. In addition, by observing the variation in the FPS parameter, we can

see that other data augmentation methods have almost no effect on the model speed, except for the random-erase, which slightly affects the model speed. Moreover, when comparing the model performance, the effects of the random-erase and Mosaic methods are similar, since using one of them on the model can realize nearly identical effects. Meanwhile, when merely adopting the affine transformation-based data augmentation method, although the model can be accelerated to 16 FPS, it has little to no substantial effect on model speed improvement. The model's precision, recall, and mAP are only 0.9328, 0.9356, and 0.9272, illustrating a significant downward trend. Hence, taking the model's performance and implementation speed characteristics into account, this paper uses CutMix, series-MixUp, and Mosaic jointly to ascertain that the model performs the best comprehensively.

6.3. Ablation Experiment of Parallel Activation Functions

The base activation function coefficient k in the parallel activation functions suggested module is heavily empirical. To investigate the model's performance with different parameter configurations, we tried different combinations of k_1, k_2, k_3. Table 5 depicts the experimental results.

Table 5. Experimental results for different combinations of base activation functions. k_1 represents the coefficient of ReLU, k_2 denotes the coefficient of Sigmoid, k_3 indicates the coefficient of Mish.

Model	k_1	k_2	k_3	P	R	mAP
SGDN 300	1.0	0.0	0.0	0.9742	0.9783	0.9765
	0.33	0.33	0.33	0.9771	0.9839	0.9825
	0.6	0.2	0.2	0.9764	0.9855	0.9827
	0.2	0.6	0.2	0.9693	0.9728	0.9714
	0.2	0.2	0.6	0.9722	0.9810	0.9759
SGDN 512	1.0	0.0	0.0	0.9711	0.9724	0.9703
	0.33	0.33	0.33	0.9726	0.9756	0.9733
	0.6	0.2	0.2	0.9720	0.9858	0.9833
	0.2	0.6	0.2	0.9729	0.9738	0.9730
	0.2	0.2	0.6	0.9726	0.9756	0.9733

Through the experiment, we found that the effect of the parallel activation functions module depends largely on the coefficient k before different activation functions; when k is uniformly taken as 0.33 or sigmoid takes the dominant role, the model performance will be seriously degraded; when k_1, k_2, k_3 are taken as 0.2, 0.2, 0.6, i.e., when the Mish function takes the dominant role, the performance of each model is improved.

6.4. CT Image Detection System on iOS

To achieve an end-to-end high-performance CT images model, an intelligent diagnosis system based on our model was developed as an app for iOS using the programming language Swift. The main functional modules of this application are as follows:

1. Section for searching and browsing patient information. Users can loop up patient information as well as look over the historical records. To make it easier for clients to access patient data, we utilize a remote server to connect. Moreover, primary patient information and medical image data are maintained in a database on a remote server.
2. Detect CT images via iOS mobile device camera, suitable for practical application scenarios.
3. Import multiple CT images through the Photos application and detect all images simultaneously.

The workflow of the detection function is shown in Figure 20.

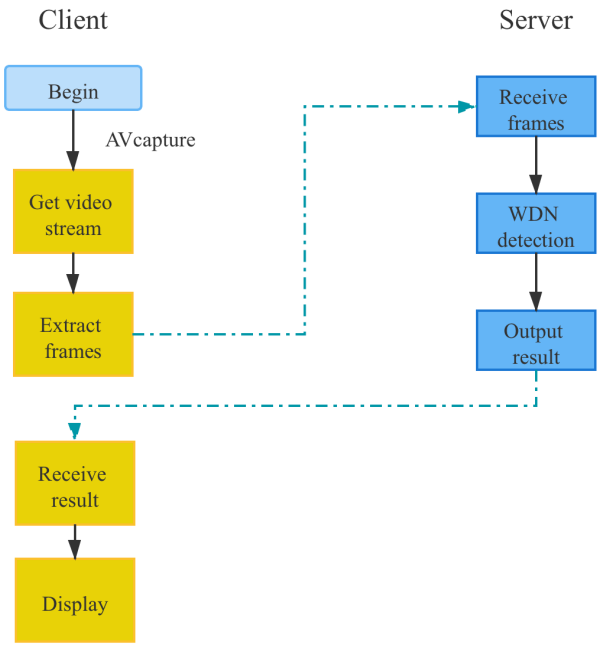

Figure 20. CT image detection system flow chart.

The procedure of detecting CT images by this app is as follows. First, a video stream of CT images is obtained through the iPhone's camera; take the realistic application scenario as an example. Then, the representative frames are obtained and released to the server. Next, the server transfers the received images to the trained model. Finally, the model's output is returned to the iOS end, and the iOS end draws a detection frame based on the returned parameters. Some screenshots of the app in action are shown in Figure 21. The app has been submitted to Apple's App Store.

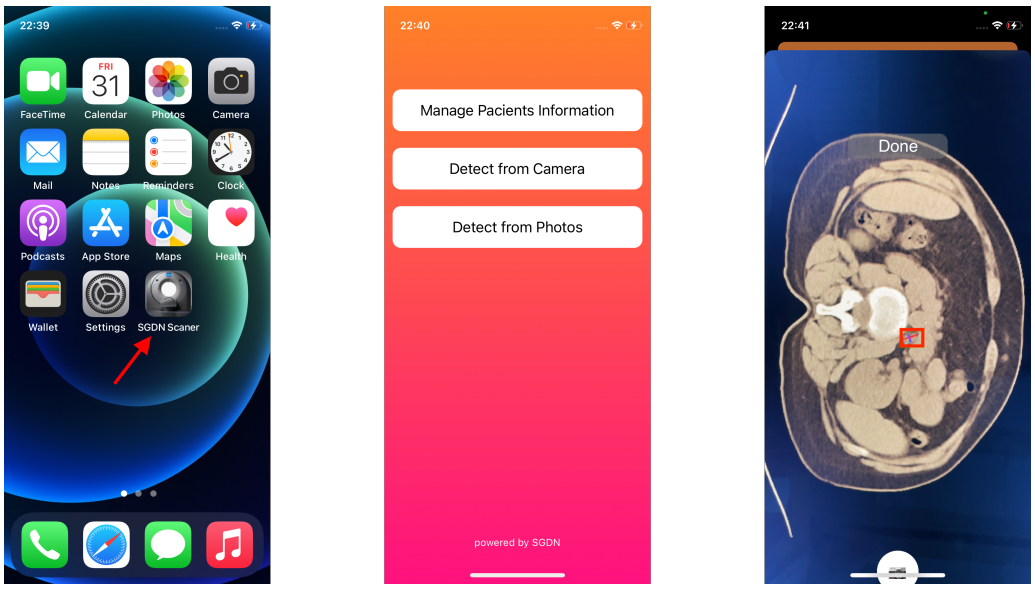

Figure 21. Screenshot on iPhone12 mini. From left to right: screenshot of the app launched on the desktop; screenshot of the function selection after launching; screenshot of the test result, and image used for testing.

Two functional modes were created for this app. The manual mode requires the user to take a picture manually for detection. The automatic mode takes a frame from the video stream every second for automatic detection and result archives. As Figure 20 shows, the detection application is implemented by server arithmetic. Meanwhile, from Table 1, we

can see that the *FPS* of our proposed model is 13. In Table 1, we can see that the FPS of our proposed model is 13. In other words, it takes less than 0.1 s to process a single image. Therefore, the detection speed of the application depends on the practical network environment.

7. Conclusions

7.1. Analysis of Symmetry GAN Detection Network

Lesions arise in body tissues as a result of a variety of causes, having a devastating impact on the human body's vital functions. In recent years, computed tomography (CT) has been dramatically enhanced and widely applied in biomedical research, particularly for lesion detection. Nevertheless, lesion detection in CT images has the following challenges: (1) the image quality drops when reducing the radiation dose to decrease radiational harm to the human body, with the scan and reconstruction variables remaining unchanged; (2) noise interference frequently hampers the image quality; (3) lesion images generally exhibit complex structures due to the intricate conditions of diseased tissue; (4) lesion structures vary from patient to patient; (5) due to pathological variables and external noise interference, the contrast between the oriented object and the background is not sufficient.

Therefore, we present a symmetry GAN detection network (SGDN) based on a one-stage detection network, aiming to address the above challenges. In this paper, we use by far the largest CT medical image dataset—DeepLesion—to identify 22 types of lesions, as shown in Figure 1.

In this paper, the original one-stage detection network has been optimized as follows:

1. Symmetry GAN models: First and foremost, a generative model is added in front of the backbone to expand the input CT image series, which aims to alleviate the general problem of small sample size in medical datasets. Second, GAN models are added to the attention extraction module to generate attention masks. Figure 7 shows the effect of adding GAN models on feature maps, and the results of the experimental part also illustrate that this approach can effectively enhance the robustness of the model. Eventually, on the validation set, the suggested method reaches values of 0.9720, 0.9858, and 0.9833 for P, R, and mAP, respectively. The statistical results demonstrate that the presented model outperforms any other compared model.
2. In order to verify the effectiveness of various implementations of symmetry GANs, in Section 6, we test the performance of GAN model A with DCGAN and CVAE-GAN and that of GAN model B with SAGAN and SPA-GAN, respectively. The experimental results demonstrate that the combination of DCGAN + SPA-GAN has the best performance, reaching values of 0.9737, 0.9845, and 0.9841 for P, R, and mAP, respectively, which further demonstrate the model's improved detection accuracy.
3. This paper presents the use of parallel multi-activation functions to replace single activation functions and theoretically proves that the performance is not inferior to that of single activation functions, as shown in Section 6.3. By applying parallel multi-activation functions, we have improved the performance of SGDN 512 by nearly 1.4%.
4. Meanwhile, the loss function of the detection task is optimized by replacing the IoU loss with a more reasonable $CIoU$ loss.
5. In this study, we encapsulate the model and develop a related application based on the iOS platform, highlighting this model's practical significance in actual scenarios.

Although the suggested model has exceeded other compared models, limitations still exist. Firstly, the model still does not perform satisfactorily in the detection masks at the boundary. Second, the model's utilization of the spatio-temporal information contained in the CT image series still needs to be improved. These demerits will be addressed in the future by the researchers of this paper.

7.2. Future Work

The presented method still has a few flaws when the detecting lesion's size is very small, as shown in Figure 19. The dataset used in this paper is a series of CT images. Nonetheless, the present model does not effectively utilize the spatio-temporal continuity of image sequences. Therefore, in the future, the authors of this paper will optimize the proposed model to extract image sequences with continuous features as much as possible to further optimize the model. Additionally, the parameter k of the parallel activation functions used in this paper is set empirically by a human. The authors will attempt to involve this parameter in the network training in future optimization.

Author Contributions: Conceptualization, Y.Z.; methodology, Y.Z.; validation, Y.Z. and Z.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.Z., S.W., S.H. and J.L.; visualization, Y.Z. and D.F.; supervision, Y.Z. and J.F.; project administration, C.L.; funding acquisition, C.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61202479.

Acknowledgments: We are grateful to the Edison Coding Club of CIEE at China Agricultural University for their strong support during our thesis writing.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You Only Look Once
SSD	Single Shot Detection
GAN	Generative Adversarial Networks
DCGAN	Deep Convolutional Generative Adversarial Networks
SPA-GAN	Spatial Attention GAN
SAGAN	Self-Attention Generative Adversarial Networks

References

1. Sun, L.; Wang, J.; Huang, Y.; Ding, X.; Greenspan, H.; Paisley, J. An adversarial learning approach to medical image synthesis for lesion detection. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2303–2314. [CrossRef] [PubMed]
2. Pedro, M.T.; Eissler, A.; Schmidberger, J.; Kratzer, W.; Wirtz, C.R.; Antoniadis, G.; Koenig, R.W. Sodium Fluorescein–Guided Surgery in Peripheral Nerve Sheath Tumors: First Experience in 10 Cases of Schwannoma. *World Neurosurg.* **2019**, *124*, e724–e732. [CrossRef] [PubMed]
3. Fyllingen, E.H.; Bø, L.E.; Reinertsen, I.; Jakola, A.S.; Sagberg, L.M.; Berntsen, E.M.; Salvesen, Ø.; Solheim, O. Survival of glioblastoma in relation to tumor location: A statistical tumor atlas of a population-based cohort. *Acta Neurochir.* **2021**, *163*, 1895–1905 [CrossRef] [PubMed]
4. Sera, T. Computed tomography. In Transparency in Biology; Springer: Berlin/Heidelberg, Germany, 2021; pp. 167–187.
5. Shen, Y.; Sun, S.; Xu, F.; Liu, Y.; Yin, X.; Zhou, X. CT Image Reconstruction via Nonlocal Low-Rank Regularization and Data-Driven Tight Frame. *Symmetry* **2021**, *13*, 1873. [CrossRef]
6. Li, G.; Jiang, D.; Zhou, Y.; Jiang, G.; Kong, J.; Manogaran, G. Human lesion detection method based on image information and brain signal. *IEEE Access* **2019**, *7*, 11533–11542. [CrossRef]
7. Zhang, Y.; Liu, X.; Wa, S.; Liu, Y.; Kang, J.; Lv, C. GenU-Net++: An Automatic Intracranial Brain Tumors Segmentation Algorithm on 3D Image Series with High Performance. *Symmetry* **2021**, *13*, 2395. [CrossRef]
8. Kremer, S.; Lersy, F.; de Sèze, J.; Ferré, J.C.; Maamar, A.; Carsin-Nicol, B.; Collange, O.; Bonneville, F.; Adam, G.; Martin-Blondel, G.; et al. Brain MRI findings in severe COVID-19: A retrospective observational study. *Radiology* **2020**, *297*, E242–E251. [CrossRef]
9. Qin, C.; Liu, F.; Yen, T.C.; Lan, X. 18F-FDG PET/CT findings of COVID-19: A series of four highly suspected cases. *Eur. J. Nucl. Med. Mol. Imaging* **2020**. [CrossRef]
10. Beregi, J.; Greffier, J. Low and ultra-low dose radiation in CT: Opportunities and limitations. *Diagn. Interv. Imaging* **2019**, *100*, 63–64. [CrossRef]
11. Leyendecker, P.; Faucher, V.; Labani, A.; Noblet, V.; Lefebvre, F.; Magotteaux, P.; Ohana, M.; Roy, C. Prospective evaluation of ultra-low-dose contrast-enhanced 100-kV abdominal computed tomography with tin filter: Effect on radiation dose reduction and image quality with a third-generation dual-source CT system. *Eur. Radiol.* **2019**, *29*, 2107–2116. [CrossRef]

12. Zhang, C.; Shen, X.; Cheng, H.; Qian, Q. Brain tumor segmentation based on hybrid clustering and morphological operations. *Int. J. Biomed. Imaging* **2019**, *2019*, 7305832. [CrossRef] [PubMed]
13. Chen, C.; Xiao, R.; Zhang, T.; Lu, Y.; Guo, X.; Wang, J.; Chen, H.; Wang, Z. Pathological lung segmentation in chest CT images based on improved random walker. *Comput. Methods Programs Biomed.* **2021**, *200*, 105864. [CrossRef] [PubMed]
14. Singh, R.; Digumarthy, S.R.; Muse, V.V.; Kambadakone, A.R.; Blake, M.A.; Tabari, A.; Hoi, Y.; Akino, N.; Angel, E.; Madan, R.; et al. Image quality and lesion detection on deep learning reconstruction and iterative reconstruction of submillisievert chest and abdominal CT. *Am. J. Roentgenol.* **2020**, *214*, 566–573. [CrossRef] [PubMed]
15. Qin, X.; Hu, X.; Wang, Q.; Zeng, J.; Chen, J. A rare acute neck pain cause that can have misdiagnosis or missed diagnosis-crowned dens syndrome: Description of two cases and a literature analysis. *Quant. Imaging Med. Surg.* **2021**, *11*, 4491. [CrossRef] [PubMed]
16. Joskowicz, L.; Cohen, D.; Caplan, N.; Sosna, J. Inter-observer variability of manual contour delineation of structures in CT. *Eur. Radiol.* **2019**, *29*, 1391–1399. [CrossRef] [PubMed]
17. Yan, K.; Wang, X.; Lu, L.; Summers, R.M. DeepLesion: Automated mining of large-scale lesion annotations and universal lesion detection with deep learning. *J. Med. Imaging* **2018**, *5*, 036501. [CrossRef] [PubMed]
18. Hatt, M.; Parmar, C.; Qi, J.; El Naqa, I. Machine (deep) learning methods for image processing and radiomics. *IEEE Trans. Radiat. Plasma Med. Sci.* **2019**, *3*, 104–108. [CrossRef]
19. Ann, E.T.L.; Hao, N.S.; Wei, G.W.; Hee, K.C. Feast In: A Machine Learning Image Recognition Model of Recipe and Lifestyle Applications. In *MATEC Web of Conferences*; EDP Sciences: Ulys, France, 2021; Volume 335, p. 04006.
20. Gu, H.; Wen, F.; Wang, B.; Lee, A.K.; Xu, D. Machine Learning-based image recognition for visual inspections. In *SNAME Maritime Convention*; OnePetro: Tacoma, WA, USA, 2019.
21. Fujiyoshi, H.; Hirakawa, T.; Yamashita, T. Deep learning-based image recognition for autonomous driving. *IATSS Res.* **2019**, *43*, 244–252. [CrossRef]
22. Sim, H.S.; Kim, H.I.; Ahn, J.J. Is deep learning for image recognition applicable to stock market prediction? *Complexity* **2019**, *2019*. [CrossRef]
23. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
24. Benjdira, B.; Khursheed, T.; Koubaa, A.; Ammar, A.; Ouni, K. Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3. In Proceedings of the 2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS), Muscat, Oman, 5–7 February 2019; pp. 1–6.
25. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
26. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
27. Jocher, G. yolov5. 2020. Available online: https://zenodo.org/record/5563715#.Yej9u9BByUk (accessed on 30 December 2021).
28. Zhai, S.; Shang, D.; Wang, S.; Dong, S. DF-SSD: An improved SSD object detection algorithm based on DenseNet and feature fusion. *IEEE Access* **2020**, *8*, 24344–24357. [CrossRef]
29. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
30. Zhang, Y.; Wa, S.; Sun, P.; Wang, Y. Pear Defect Detection Method Based on ResNet and DCGAN. *Information* **2021**, *12*, 397. [CrossRef]
31. Zhang, Y.; Wa, S.; Liu, Y.; Zhou, X.; Sun, P.; Ma, Q. High-Accuracy Detection of Maize Leaf Diseases CNN Based on Multi-Pathway Activation Function Module. *Remote Sens.* **2021**, *13*, 4218. [CrossRef]
32. Zhang, Y.; He, S.; Wa, S.; Zong, Z.; Liu, Y. Using Generative Module and Pruning Inference for the Fast and Accurate Detection of Apple Flower in Natural Environments. *Information* **2021**, *12*, 495. [CrossRef]
33. Roy, M.; Mali, K.; Chatterjee, S.; Chakraborty, S.; Debnath, R.; Sen, S. A study on the applications of the biomedical image encryption methods for secured computer aided diagnostics. In Proceedings of the 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 4–6 February 2019; pp. 881–886.
34. Du, G.; Cao, X.; Liang, J.; Chen, X.; Zhan, Y. Medical image segmentation based on u-net: A review. *J. Imaging Sci. Technol.* **2020**, *64*, 20508-1–20508-12. [CrossRef]
35. Savelli, B.; Bria, A.; Molinara, M.; Marrocco, C.; Tortorella, F. A multi-context CNN ensemble for small lesion detection. *Artif. Intell. Med.* **2020**, *103*, 101749. [CrossRef]
36. Liu, Y.; Ma, Z.; Liu, X.; Ma, S.; Ren, K. Privacy-Preserving Object Detection for Medical Images With Faster R-CNN. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 69–84. [CrossRef]
37. Varshni, D.; Thakral, K.; Agarwal, L.; Nijhawan, R.; Mittal, A. Pneumonia Detection Using CNN based Feature Extraction. In Proceedings of the 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 20–22 February 2019; pp. 1–7. [CrossRef]
38. Yan, X.; Cui, B.; Xu, Y.; Shi, P.; Wang, Z. A method of information protection for collaborative deep learning under gan model attack. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2019**, *18*, 871–881. [CrossRef]
39. Ben-Cohen, A.; Klang, E.; Raskin, S.P.; Soffer, S.; Ben-Haim, S.; Konen, E.; Amitai, M.M.; Greenspan, H. Cross-modality synthesis from CT to PET using FCN and GAN networks for improved automated lesion detection. *Eng. Appl. Artif. Intell.* **2019**, *78*, 186–194. [CrossRef]

40. Zhu, J.; Yang, G.; Lio, P. How can we make GAN perform better in single medical image super-resolution? A lesion focused multi-scale approach. In Proceedings of the 2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019), Venice, Italy, 8–11 April 2019; pp. 1669–1673.
41. Rashid, H.; Tanveer, M.A.; Khan, H.A. Skin lesion classification using GAN based data augmentation. In Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 23–27 July 2019; pp. 916–919.
42. Hammami, M.; Friboulet, D.; Kechichian, R. Cycle GAN-Based Data Augmentation for Multi-Organ Detection in CT Images Via Yolo. In Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, 25–28 October 2020; pp. 390–393.
43. Finck, T.; Li, H.; Grundl, L.; Eichinger, P.; Bussas, M.; Mühlau, M.; Menze, B.; Wiestler, B. Deep-learning generated synthetic double inversion recovery images improve multiple sclerosis lesion detection. *Investig. Radiol.* **2020**, *55*, 318–323. [CrossRef]
44. Qin, Z.; Liu, Z.; Zhu, P.; Xue, Y. A GAN-based image synthesis method for skin lesion classification. *Comput. Methods Programs Biomed.* **2020**, *195*, 105568. [CrossRef]
45. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
46. Sural, S.; Qian, G.; Pramanik, S. Segmentation and histogram generation using the HSV color space for image retrieval. In Proceedings of the International Conference on Image Processing, Rochester, NY, USA, 22–25 September 2002; Volume 2, p. II.
47. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. Mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
48. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6023–6032.
49. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
50. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
51. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
52. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016, pp. 21–37.
53. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
54. Everingham, M. The PASCAL Visual Object Classes Challenge 2007. 2007. Available online: http://host.robots.ox.ac.uk/pascal/VOC/voc2007/ (accessed on 30 December 2021).
55. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768.
56. Yadav, S.; Shukla, S. Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In Proceedings of the 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, India, 27–28 February 2016; pp. 78–83.
57. Misra, D. Mish: A self regularized non-monotonic neural activation function. *arXiv* **2019**, arXiv:1908.08681.
58. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
59. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
60. Huang, G.; Liu, Z.; Laurens, V.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
61. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
62. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 12993–13000.
63. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef] [PubMed]