IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 13, 2018, accepted September 27, 2018, date of publication October 11, 2018,
date of current version November 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2875406

DeepCXray: Automatically Diagnosing Diseases on Chest X-Rays Using Deep Neural Networks

XIUYUAN XU, QUAN GUO, (Member, IEEE), JIXIANG GUO, (Member, IEEE), AND ZHANG YI, (Fellow, IEEE)
Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu 610065, China

Corresponding author: Zhang Yi (zhangyi@scu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61432012 and Grant U1435213.

ABSTRACT The automatic detection of diseases in images acquired through chest X-rays can be useful in clinical diagnosis because of a shortage of experienced doctors. Compared with natural images, those acquired through chest X-rays are obtained by using penetrating imaging technology, such that there are multiple levels of features in an image. It is thus difficult to extract the features of a disease for further diagnosis. In practice, healthy people are in a majority and the morbidities of different disease vary, because of which the obtained labels are imbalanced. The two main challenges of diagnosis though chest X-ray images are to extract discriminative features from X-ray images and handle the problem of imbalanced data distribution. In this paper, we propose a deep neural network called DeepCXray that simultaneously solves these two problems. An InceptionV3 model is trained to extract features from raw images, and a new objective function is designed to address the problem of imbalanced data distribution. The proposed objective function is a performance index based on cross entropy loss that automatically weights the ratio of positive to negative samples. In other words, the proposed loss function can automatically reduce the influence of an overwhelming number of negative samples by shrinking each cross entropy terms by a different extent. Extensive experiments highlight the promising performance of DeepCXray on the ChestXray14 dataset of the National Institutes of Health in terms of the area under the receiver operating characteristic curve.

INDEX TERMS Chest X-rays, deep neural networks, cross weighted cross entropy loss, imbalanced data, feature extraction.

I. INTRODUCTION

A chest X-ray is a quick and painless procedure that generates images of the internal form of the chest. It is widely used to help diagnose and monitor treatment for a variety of diseases, such as chest injury, pneumonia, emphysema, and lung cancer [1], [2]. To obtain a diagnosis, even experienced doctors must carefully analyze chest X-ray images. Such a procedure is time consuming, and the correctness of the diagnosis depends completely on the experience of the doctor. In practice, however, there is a lack of experienced physicians to provide a high quality of service to all patients, and many patients may not be diagnosed and treated in time. Therefore, it is highly desirable to develop a system that can automatically diagnose certain diseases using images obtained from chest X-rays images.

In the last several years, a large number of studies have been devoted to image classification [3]. Of them, remarkable progress has been made due to the development of deep neural networks [4]–[8]. Convolutional neural networks (CNNs) have manifested powerful abilities to extract the inner representations of images, and have obtained significant success in many areas of computer vision [4], [7]–[11].

CNNs have also achieved promising performance in natural image classification. A 1000-category image classification task [3] was released in 2009. Since then, the accuracy of computer in ranking the top five images has continually improved. In 2016 the error rate was reduced to 5.29% [6], which is competitive with the human-level performance.

The classification of chest X-ray images is different from the classification of natural images. Recognizing X-ray images is more useful than recognizing natural images in the medical fields. Moreover, images of chest X-rays pose new challenges because the features of diseases are very hard to identify. Traditional methods tend to extract particular features from chest X-ray images [12]–[15], but, they are manually designed and ineffective. These methods focus on regions of interest based on texture and shape features. Moreover, the regions of disease activity cannot be discriminated by using texture and shape as features. Diseases have latent features that cannot be manually extracted. Nowadays, deep neural networks, especially deep CNNs, perform well at extracting the inner representations of images. To extract the features of X-ray images effectively, in this study, we trained an InceptionV3 [11] module to extract the features of X-ray images and employ transfer learning to boost training performance.

Another challenge posed by automatic detection of diseases from chest X-ray images is that the images have imbalanced data distribution. Considering the recently released ChestXray14 dataset [1] as an example, it contains 112, 120 images from over 14 classes. Positive samples of one of the disease categories (hernia), are the fewest in number, at 199. By contrast, a large number of images (more than 50, 000) show no diseases and are called negative samples. Such an imbalanced dataset degrades the performances of many diagnostic algorithms because they are easily pulled toward the negative samples. To address the problem of the imbalanced data distribution, common solution in deep neural network training is data augmentation, such as by cropping, rotating, and flipping positive images. A major disadvantage of data augmentation is the limited diversity of positive samples. It cannot introduce new features of diseases because the ''new'' data must actually be derived from available data.

To address the above two challenges, we propose a deep neural network called DeepCXray to diagnose diseases on chest X-rays automatically. The proposed DeepCXray is an end-to-end and optimized classifier that employs InceptionV3 as feature extractor, and is trained using a new objective function. The major novelty of this work lies in the proposed objective function, called cross weighted cross entropy loss (CW-CEL), which allows our network to overcome the problem of too many negative samples by weighting the ratio of positive to negative samples. Extensive experiments show that our method outperforms state-of-the-art methods on a recently released chest X-ray image database by a considerable margin.

The remainder of this paper is organized as follows: In Section II, we introduce some related work in feature-extraction and imbalanced datasets. In Section III, the proposed method is presented in detail. In Section IV, extensive experiments are reported to verify the effectiveness of our method, and the conclusions of this study are drawn in Section V.

II. RELATED WORK

A. COMPUTER-AIDED DIAGNOSTICS

Computer-aided diagnostics(CADs) is technology that can be used to substantially improve the efficiency of a doctor's diagnosis. The relevant methods rely heavily depended on the extracted features. Traditional features-extraction methods are manually designed, and focus on extracting texture and shape features [12]–[15].

A method proposed by Jaeger et al. [15] is based on a graph cut segmentation method. They computed a set of texture and shape features that enabled the classification of X-ray images as normal or abnormal using a binary classifier. Boussaid and Kokkinos [13] proposed the loopy part model to segment ensembles of organs in medical images. Each organ's shape was represented by an acyclic graph while shape consistency was enforced through inter-shape connections. Avni et al. [12] proposed an efficient image categorization and retrieval system based on a local patch representation of the image content using a bag-of-visual-words approach. These methods are all based on manually designed feature extraction. With the development of deep learning, many CAD tools based on deep learning have been proposed [16], [17]. Gulshan *et al.* [16] proposed a deep learning-based method to construct an effective diabetic retinopathy diagnosis system with a performance that was comparable to that of a trained radiologist. Esteva *et al.* [17] constructed a single end-to-end CNN using only pixels and disease labels as inputs. This method outperformed 21 board-certified dermatologists on biopsy-proven clinical images with two critical binary classification use cases: keratinocyte carcinomas versus benign seborrheic keratoses, and malignant melanomas versus benign nevi.

B. DEEP CNNS

CNNs are inspired by biological vision mechanisms [18]. Because of their trainable filters and shared parameters, they are quite effective in computer vision tasks. Moreover, CNNs are an important type of neural network that have made many breakthroughs in computer vision. The first successful application of CNNs was in the recognition of handwritten digits using the Lenet-5 [10]. After Hinton et al. [19] proposed deep belief networks in 2006 and developed the concept of deep learning, CNNs and deep learning developed quickly. In 2012, Krizhevsky et al. [4] constructed a CNN to obtain the best result in an ImageNet competition that was $8.2 - 8.7$ percentage points superior to previous results. Lin et al. [20] noted that a fully connected layer is not necessary in a convolution neural network, and proposed that 1×1 kernels can be used instead. From then on, an increasing number studies have focused on deep CNNs. There were two initial solutions in deep learning. One was the VGG [5] model architecture, and the other one was the famous inception architecture inspired by Lin et al. [20]. He et al. [8] subsequently proposed residual blocks that enable a CNN to go deeper, and Huang et al. [6] proposed dense blocks that can be used for even deeper networks. As CNNs developed, many studies investigated their training. ZFNet, proposed by Zeiler and Fergus [21] enables users to visualize the inner representation learned by convolutional layers, which can help them adjust the parameters effectively.

CNNs are also being used in weakly supervised object localization or visualization. Zhou et al. [22] proposed a method that generates class activation maps (CAMs) using the global average pooling (GAP) layers in CNNs. A CAM for a particular category indicates the discriminative image regions used by the CNN to identify that category. Wang et al. [1] proposed a method to construct a transition layer in a neural network architecture when the model was trained sufficiently well. They combined the transition layer with the output to construct heatmaps for disease localization. In our experiments, to visualize, we employed global average pooling layer to directly calculate the heatmaps.

C. METHODS TO TRAIN IMBALANCED DATASET

With the development of deep learning, which is a data-driven approach, imbalanced data classes have become an unavoidable issue for scientists particularly in medical imaging field. Researchers have obtained promising results on some datasets and the relevant methods can be divided into two categories: data-level and classifier-level [23] approaches.

Data-level methods, are widely used in deep leaning [24]–[27]. The most commonly used method is ***oversampling***. The basic version of oversampling is called random minority oversampling, which simply replicates randomly selected samples from minority classes. It has been shown that oversampling is effective, but it can lead to overfitting [28]. Hence, some methods have been recently proposed to ensure the uniform class distribution of each mini-batch and control the selection of examples from each class. Another popular data-level method is ***undersampling***, which ensures the same number of examples in each class by reducing the number of negative samples (or other majority samples) [24]. The second approach to the problem of imbalanced data classes, classifier-level methods, include the following: *threshold-based methods*, *cost-sensitive learning*, *one-class classification* and *hybrid methods*. For threshold-based methods, the most basic version simply compensates for prior class probabilities [29]. Cost-sensitive learning assigns a different cost to the misclassification of examples from different classes. Cost-sensitive learning with respect to neural networks can also be applied to the inference phase once the classifier has been trained, such as in the threshold moving [30] or post scaling [31] methods. This method can be adapted to modify the learning rate so that examples of higher cost contribute more to updating the weights [32]. In two-category classification tasks, some work has involved a technique that recognizes positive instances instead of discriminating between classes. Then, a new instance is classified using the reconstruction errors between input and output patterns [33]–[35]. Finally, in many cases, using only one of the above methods does not work effectively, or is not suitable for some datasets. Combining methods from one or both of the above-mentioned categories is a useful approach. Recently, two-phase training was introduced and successfully used to train a CNN for brain tumor segmentation [36]. However, these methods mentioned cannot solve the problem of imbalanced multi-label datasets. Modifications to the cross entropy loss function are common method to promote the effectiveness of training. Wen *et al.* [37] proposed a novel loss function called center loss and combined it with softmax cross entropy loss to obtain highly discriminative features for robust face recognition. In terms of addressing imbalance in multi-label classification tasks, Li and Wang [38] proposed RMLS. Wang et al. [1] proposed a loss function to reduce the influence of many negative samples by balancing the errors of negative and positive samples in the loss function. This study amplified the influence of errors to varying degrees. However, the number of positives in a batch is small that cause the weights to increase substantially, which bring training oscillations. Hence, we proposed a loss function that can address this problem and make training more stable.

III. DEEPCXRAY

An automatic disease diagnosis system for images from chest X-rays can be regarded as a multi-label classification task, where the input is a frontal-view chest X-ray image X and the output is a $14-$dimensional vector, where an element labeled "1" denotes the presence of a certain disease, and a label of "0" denotes the absence of it. In this study, we aim to enable the system to learn a mapping F from X to y, $F(X) \rightarrow y$. We define performance index $L(F(X), y)$ to measure the distance between the output of model $F(X)$ and the target y. In this section, we presented our deep neural network, called DeepCXray, to approximate the mapping $F(X)$ by updating weights W to minimize $L(F(X), y)$.

The proposed method is shown in FIGURE 1. We first use X-ray images as inputs to our model. The main part of this model is based on a deep CNN architecture that enables us to obtain the inner representations of these images. The green cubes in the figure are CNNs that are regarded as feature extractors. This module can use a residual block [8], the inception module [7], a dense block [6], and/or local response normalization [4], as shown in the middle blocks indicated by the dashed-line. Following global average pooling, the results of the model are generated for the diagnosis of 14 diseases. To improve the model's performance, we add an auxiliary task and propose a loss function to train the model effectively. In the new loss function, we combine the outputs with the corresponding labels to calculate cross entropy values. The "×" in the right dashed-line block denotes the element-wise product.

A. NETWORK STRUCTURE

The proposed DeepCXray employs InceptionV3 [11] as feature extractor. Because the size of a medical image database is often much smaller than that of a natural image databases, to exploit the potential of deep neural networks and benefit from the representative capacity of a transferable model, we pre-trained InceptionV3 on ImageNet [3], one of the most well-known image datasets. The original InceptionV3 had $1,000$ outputs. To make this output suitable for our task, we replaced the last layer with a layer of 14 neurons. Moreover, we experimentally found that when DeepCXray was trained with an auxiliary task to predict a normal image, a better prediction of the 14 diagnosis can was obtained.

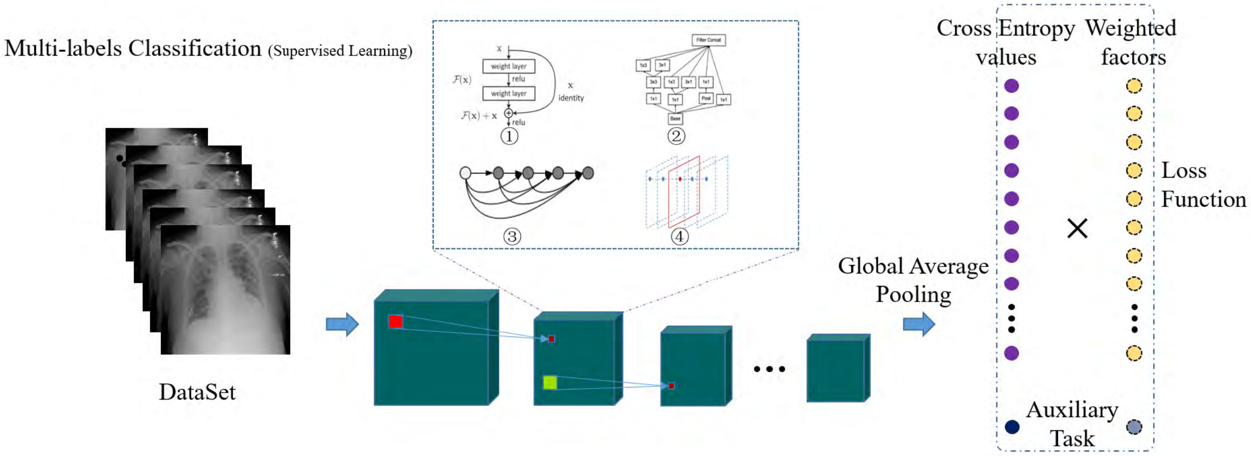


FIGURE 1. DeepCXray for multi-label classification.

InceptionV3 is a CNN architecture and is the third generation of the inception architecture. This architecture modifies the kernel-size of convolutions in the network. The main idea is that several small kernels can replace a large kernel. To reduce the amount of calculation and total number of all parameters, this study uses kernel factorization to factorize large kernels into small ones. There are two kinds of factorizations: replacing the larger kernel with several small ones, and replacing a large convolution kernel with several asymmetric ones. For examples, a 5×5 kernel can be replaced by two 3×3 kernels. Alternatively, an $n \times n$ kernel can be replaced by a $1 \times n$ kernel followed by an $n \times 1$ kernel. InceptionV3 employs a global average pooling layer and $1,000$ outputs for $1,000-$category classification. However, our task outputs 14 classes. To apply the InceptionV3 model to this $14-$category multi-classification task, we replaced the last layer with 14 neural outputs.

We made several end-to-end binary predictions within a single model. In this model, if the output is a vector of zeroes, the given chest X-ray image is predicted to show no diseases. If there are ones in the output vector, the corresponding diseases are likely to exist. The dataset has imbalanced labels, which result in sparse target vectors.

Because the imbalanced dataset has many negative samples that are labeled zero, the positive samples are too few to effectively train. Some work [39] indicated that multi-task learning is one approach to inductive transfer that improves generalization using the domain information contained in the training signals of related tasks as an inductive bias. In this study, if a sample is indicated as normal, the vector of the target consist of all zeros; if not, there is at least one disease. We added an auxiliary task to predict a given image as that of a normal (disease free) chest or not to train this model effectively. The results of the following experiments indicate that this method can improve the performance of the model.

B. CROSS WEIGHTED CROSS ENTROPY LOSS

We note that the dataset is imbalanced. Methods to address solve the imbalanced dataset problem consist of data-level and classifier-level approaches. Data-level methods, as mentioned above, consist of *undersampling* and *upsampling*. These methods are not suitable for the multi-classification tasks because a sample can belong to multiple categories, i.e., a minority category as well as the majority category. *Undersampling* and *upsampling* thus cannot balance the distribution of such a dataset, although a method proposed in [38] considers the different labels' relevance in a given instance.

Some classifier-level methods are available as well. For instance, Wang et al. [1] proposed a loss layer called the weighted cross entropy loss(W-CEL) to force the network to learn positive examples by minimizing the following equation:

$$\begin{aligned} &L_{W-CEL}(f(\vec{x}), \vec{y}) \\ &\quad = \beta_P \sum_{y_c=1} - \ln(f(x_c)) + \beta_N \sum_{y_c=0} - \ln(1 - f(x_c)), \end{aligned} \tag{1}$$

$$\beta_P = \frac{|P| + |N|}{|P|}, \tag{2}$$

$$\beta_N = \frac{|P| + |N|}{|N|}. \tag{3}$$

For updates, the weights obey the following equations:

$$W \leftarrow W - \Delta W, \tag{4}$$

$$\Delta W = \beta_P \sum_{y_c=1} \frac{-\partial ln(f(x_c))}{\partial W} + \beta_N \sum_{y_c=0} \frac{-\partial ln(1-f(x_c))}{\partial W} \tag{5}$$

where $|P|$ and $|N|$ are the total number of ones and zeros in a batch of image labels. This function has a drawback. When sampling a mini-batch, since P is sparse in general, $|P|$ may equal or close to zero, and then and then β_P goes to infinite or very large, this will result in the training process stop earlier or oscillating. The experiments in this study show this.

To overcome the disadvantages, we propose a loss function, called CW-CEL, which is defined as follows:

$$\begin{aligned} &L_{CW-CEL}(f(\vec{x}), \vec{y}) \\ &\quad = \alpha_N \sum_{y_c=1} - \ln(f(x_c)) + \alpha_P \sum_{y_c=0} - \ln(1 - f(x_c)), \end{aligned} \tag{6}$$

$$\alpha_P = \frac{|P|}{|P| + |N|}, \tag{7}$$

$$\alpha_N = \frac{|N|}{|P| + |N|}. \tag{8}$$

Clearly, the parameters α_N and α_P are controlled in a range between zero and one, thus give the training to be more stable. Moreover, since $\alpha_N + \alpha_P = 1$, it gives a good balance between positive and negative labels.

To weaken the influence of imbalanced labels. Wang et al.'s proposal [1] and our method both optimize the loss function. In contrast to our method, however Wang *et al.* magnified the two terms of the cross entropy loss to some extent, which can lead to the potential problem. The major problem is that $|P|$ can be very small, which leads to larger gradients that causes the training to oscillate sharply. Our method shrinks the two terms of cross entropy loss. When we balance the dataset, the weights are in the range [0, 1], and hence our loss function can avoid oscillation caused by batch samples. The experiments in this study show that this method works well.

C. FINE-GRAINED CROSS WEIGHTED CROSS ENTROPY LOSS

As shown in the above equations, the range of oscillation of the loss values can be limited to a small scale, but, the methods proposed above correct only the imbalance between negatives and positives. In multi-category classification, there is also an imbalance problem between labels. Different labels cannot be balanced based on the methods above. To address this problem, a weight is added to a different class output. This method is as the follows.

$$L_{CW-CEL}(f(\vec{x}), \vec{y}) = \sum_{y_c=1} -\alpha_N^c ln(f(x_c)) + \sum_{y_c=0} -\alpha_P^c ln(1 - f(x_c)), \tag{9}$$

$$\alpha_P^c = \frac{P_c}{N_c + P_c}, \tag{10}$$

$$\alpha_N^c = \frac{N_c}{N_c + P_c}. \tag{11}$$

In this equation, α_P^c is the corresponding disease ratio in a batch. Based upon this, each disease has different proportion, and thus every class can be balanced to some extent by this method.

Other classifier-level methods are available. Threshold moving [30] and post scaling [31] can be employed to modify the learning rate to force more costly examples to contribute more to updating the weights. These methods focus on some hard samples(that is, those that are difficult for the model to discriminate). However, in multi-category classification, a sample can belong to both the majority and the minority categories at the same time, and these approaches cannot address this case. These methods modify only the learning rates, which does not make a difference with respect to the error of each category's. The method, CW-CEL does not address this problem either. However, our method considers this problem and is designed to balance the relationship of each label to the entire dataset.

IV. EXPERIMENTS

In this section, we present the results of experiments conducted on a real-world dataset to evaluate the effectiveness of the proposed method. The parameters of the method were initialized using pre-trained models, and it was then trained end-to-end using Adadelta [40] with initial learning rate $lr = 1e - 5$. We replace the last layer with a one with 15 outputs, in which we used the sigmoid function to generate the probability for each disease and one output for the auxiliary task to predict normal or diseased images. We trained the model with mini-batches of size 16. We also designed the experiments to compare models with and without transfer learning, and compared models with and without an auxiliary task at the same time. All experimental trials were run on the graphics processing unit (GPU) enabled TensorFlow computational framework on a single NVIDIA Tesla K40 running the Ubuntu 14.04 operating system.

A. DATASETS

We tested our model on the ChestXray14 dataset which [1] is an imbalanced dataset of images of diseases lungs released by the National Institute of Health. The dataset contained 14 diseases as shown in FIGURE 2, and provided more than 112, 120 pictures from 30, 805 patients labeled using natural language processing. The number of instance of imaging presenting each disease are shown in the histogram in FIGURE 3.

We split the dataset into three parts: 70% was used for training, 20% for testing, and the remaining 10% was reserved for validation. We evaluated the performance of the models based on the validation dataset, during training. We chose the best checkpoint on the validation dataset and employed the model to test the test dataset and obtained the final results. To verify reproducibility of our results, we performed six random trials of the experiments. The results show that the choice of the split had an insignificant effects on performance.

Before entering the dataset into the model, we normalized the pixel values of the images in the range $[-1, 1]$. Because some pre-trained models require three-channel pictures, we converted gray images into three-channel images, where each channel had the same value. To unify the input size, we resized every image to dimensions of 299 × 299.

B. METRIC

To assess the performance of this model using multi-class classification, we used the area under the receiver operating characteristic (AUC). When normalized units are used, the AUC is equal to the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative one (assuming ''positive'' ranks are higher

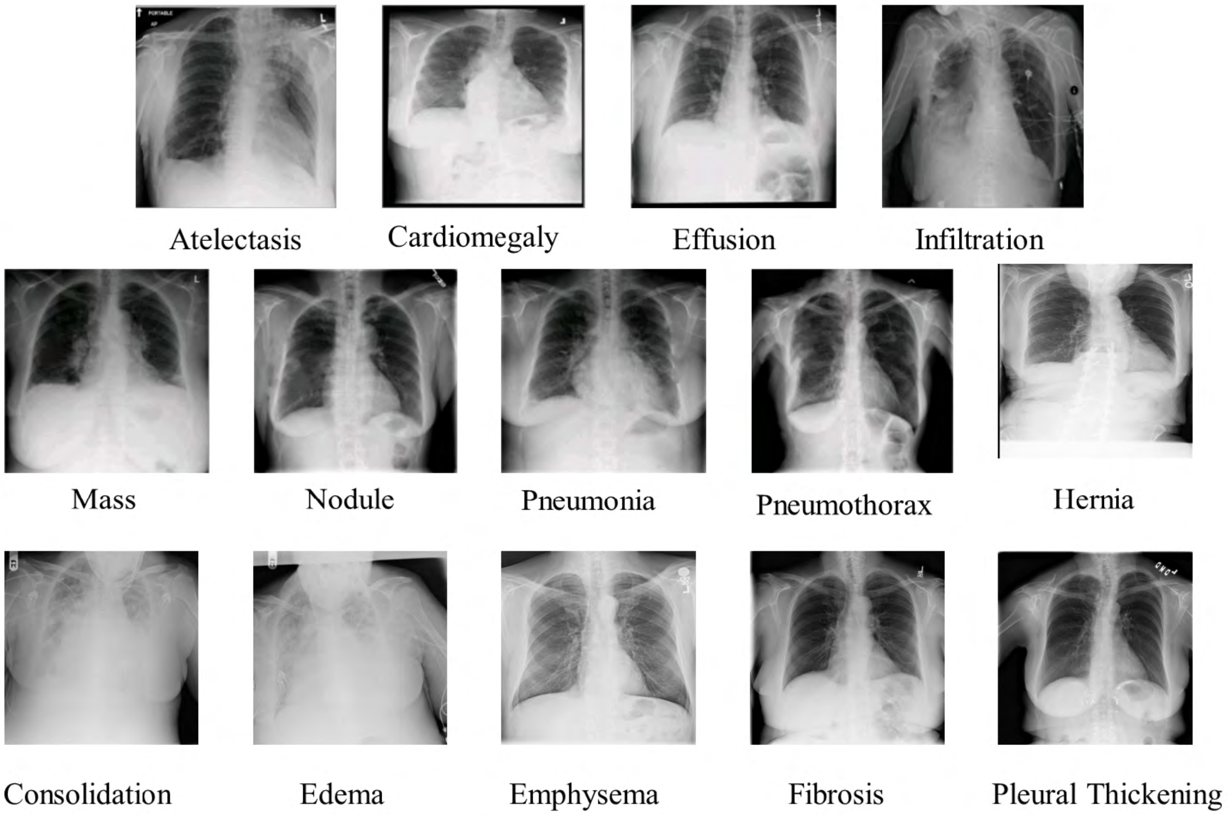


FIGURE 2. Samples of 14 diseases from ChestXray14 dataset.

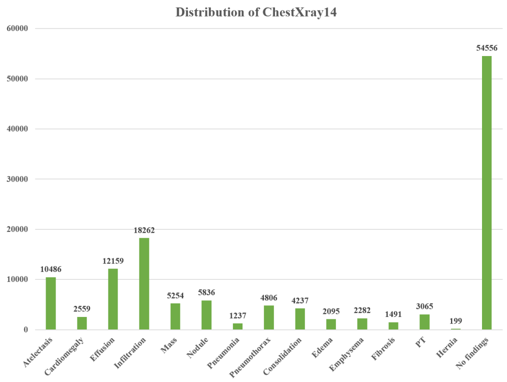


FIGURE 3. Number of instance of disease in the database, which ranges from 18262 to 199 images. There are 54556 normal images.

than "negative"). The AUC is given by

$$\begin{aligned} A &= \int_{-\infty}^{\infty} TPR(T) \cdot (-FPR'(T))dT \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(T' > T) \cdot f_1(T') \cdot f_0(T)dT'dT \\ &= P(pos_1 > pos_0) \end{aligned} \tag{12}$$

where pos_1 is the score for a positive instance, pos_0 the score for a negative instance, and f_0 and f_1 are the probability densities, as defined in the previous section. Moreover, the integral boundaries are reserved because a large value of T has a lower value on the $x - axis$.

C. COMPARISON BETWEEN W-CEL AND CW-CEL ON CHESTXRAY IMAGES

The proposed loss function is an improvement over Weighed Cross Entropy Loss(W-CEL).

In the experiments, every model was trained for 23, 000 steps. We chose a model that delivered the best performance upon the validation dataset to evaluate on the testing dataset. In this section, we compared CW-CEL proposed in this study with W-CEL with and without transfer learning. The results are shown in TABLE 1.

"NT" in the table denotes the models trained without transfer learning. We did not employ a pre-trained model to boost this training process. When trained without transfer learning, the models W-CEL(NT) and CW-CEL(NT) did not converge properly. However, the models with transfer learning obtained the best performance in approximately 9, 000 steps. We evaluated the models without transfer learning for 23, 000 steps, and evaluated the models with transfer learning obtained for 9, 000 steps. The results forW-CEL are much better than those of W-CEL(NT), and highlighted the effectiveness of transfer learning's. To compare, we trained a model with CW-CEL under the same conditions.

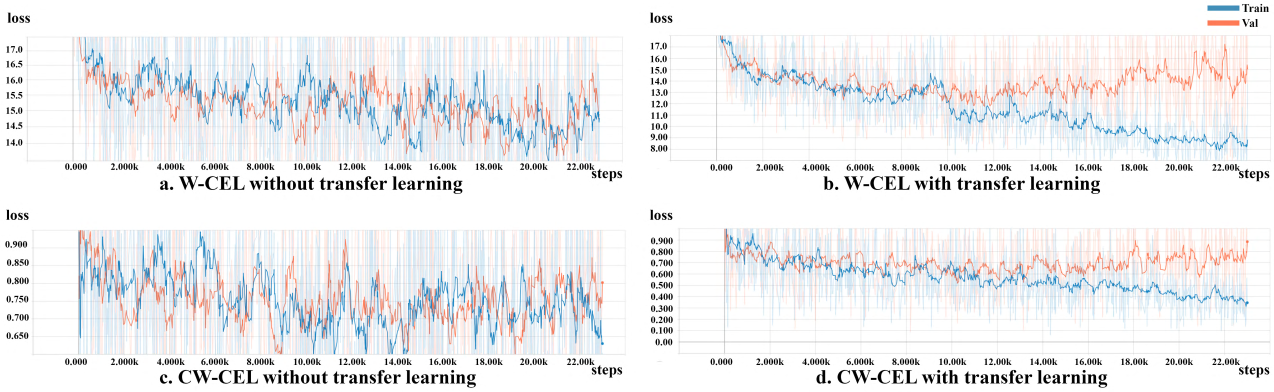


FIGURE 4. The loss curves plotted here for the process of training and validation of the models trained by different methods.

TABLE 1. The comparison between W-CEL and CW-CEL. "NT" denotes that training the model without pre-trained parameters.

Pathology	W-CEL(NT)	W-CEL	CW-CEL(NT)	CW-CEL
Atelectasis	0.6725	0.751	0.7027	0.7833
Cardiomegaly	0.6893	0.8055	0.7794	0.8674
Effusion	0.7505	0.8363	0.8076	0.8576
Infiltration	0.6462	0.6826	0.6648	0.6934
Mass	0.6405	0.7341	0.6836	0.7814
Nodule	0.5877	0.6909	0.6072	0.7264
Pneumonia	0.649	0.6967	0.6596	0.7204
Pneumothorax	0.7166	0.8417	0.7797	0.8822
Consolidation	0.7585	0.7785	0.771	0.7874
Edema	0.8318	0.8819	0.8675	0.8945
Emphysema	0.6661	0.8149	0.7401	0.8806
Fibrosis	0.7139	0.7935	0.7442	0.8295
Pleural Thickening	0.6477	0.7223	0.6846	0.7678
Hernia	0.6572	0.7894	0.7553	0.8324
AUC(Mean)	0.6877	0.7728	0.7315	0.8075

CW-CEL(NT) outperformed W-CEL(NT), whereas CW-CEL outperform than W-CEL. From the table, it is evident that our methods can promote $0.03 \sim 0.05$ mean AUC values of 14 diseases compared to W-CEL.

To visualize the process of training, we plot the loss in the training and validation datasets in FIGURE 4 using Tensorborad with smooth weights of 0.9. In this figure, the blue line denotes training loss, and the orange line denotes validation loss. FIGURES 4-a and 4-b plot the loss curve of W-CEL without and with transfer learning respectively. FIGURES 4-c and 4-d plot the loss curve of CW-CEL without and with transfer learning respectively. The loss value of CW-CEL was smaller than that of W-CEL by one order of magnitude. The range of loss oscillation of CW-CEL was smaller than that of W-CEL. When trained with W-CEL, the model was terminated due to *Inf* and *NaN* errors at times. As an extreme example, if $|P|$ was zero in a batch, which was very likely to happen to the minority categories, β_P is *Inf* and led to a training error.

From FIGURES 4-a and 4-b, we see that the loss with transfer learning was much more stable than that without transfer learning. The latter loss can be decreased to a considerably smaller value than the former. The comparison between FIGURES 4-c and 4-d confirms this. We also see that the model with CW-CEL and W-CEL using transfer learning began over-fitting the training set at almost the same steps. We think the InceptionV3 is a delicate and complex model with many trainable parameters. The chest X-ray images are not sufficient to train such a model. Comparing FIGURES 4-b and 4-d, the two models both over-fitted after a sufficient number of steps. However, the model in FIGURE 4-d shows a smaller degree of over-fitting.

D. COMPARISON WITH OTHER LOSS FUNCTIONS ON CHESTXRAY IMAGES

To evaluate the learning performance of our loss function, we designed an experiment based on the ChestXray14 dataset, in which we used the same architecture, InceptionV3 with pre-trained parameters. However, the loss function was different. The results are shown in TABLE 2.

In this experiment, we compared the results obtained by different loss functions with the same architecture. We compare the mean squared error(MSE) of the loss layer, the CW-CEL layer, and the CW-CEL layer with an auxiliary task. All models were trained with the same number of iterations (approximately $9,000$ and $23,000$) with the same batch size of 16. The results show that our loss function outperformed the other methods. Moreover, $9,000$ steps were sufficient to train our method. MSE(S) and CEL(S), denote the models trained with $23,000$ steps, and the results show that the MSE loss and CEL functions required more iterations for training. Our method needed approximately half the number of update steps. Thus, our loss function can train the model more quickly and well. The results of CEL(S) for the diagnosis of atelectasis, cardiomegaly, effusion and masses were better than those of CW-CEL. However, for others diseases, the other methods perform worse than CW-CEL. These results indicate that our method is superior to the other methods on classes with smaller sizes.

We also added an auxiliary task to determine whether an image was normal in training, and removed this task when testing. For this experiment, we trained the model for approximately $9,000$ steps, the same as for CW-CEL. The results

TABLE 2. Multi-label classification results of the proposed loss function and other loss functions on the ChestXray14 dataset. CW-CEL(AT) denotes CW-CEL with an auxiliary task. CEL(tran.) denotes the architecture that contains a transition layer.

Pathology	MSE	CEL	MSE(S)	CEL(tran.)	CEL(S)	CW-CEL	CW-CEL(AT)	CW-CEL(AT, FG)
Atelectasis	0.6799	0.6908	0.7482	0.7810	0.7898	0.7833	0.7924	0.7838
Cardiomegaly	0.5963	0.5869	0.6547	0.8011	0.8990	0.8674	0.8935	0.8628
Effusion	0.7679	0.8384	0.8483	0.8492	0.8794	0.8576	0.8651	0.8577
Infiltration	0.6304	0.6444	0.6484	0.6696	0.6827	0.6934	0.6957	0.6936
Mass	0.5890	0.6676	0.6346	0.7795	0.8178	0.7814	0.8307	0.7801
Nodule	0.5798	0.5820	0.5461	0.7050	0.7046	0.7264	0.7549	0.7266
Pneumonia	0.6145	0.6664	0.6428	0.6402	0.6916	0.7204	0.8530	0.8326
Pneumothorax	0.6948	0.7570	0.7117	0.8118	0.8658	0.8822	0.8727	0.8583
Consolidation	0.6914	0.7560	0.7523	0.7274	0.7691	0.7874	0.7841	0.7680
Edema	0.7607	0.7761	0.8029	0.7888	0.8691	0.8945	0.9010	0.8908
Emphysema	0.6457	0.6924	0.6715	0.8097	0.8649	0.8806	0.9211	0.8686
Fibrosis	0.6148	0.6177	0.5654	0.7211	0.7505	0.8295	0.8190	0.8077
Pleural Thickening	0.6343	0.6689	0.6678	0.6858	0.7664	0.7678	0.7983	0.7729
Hernia	0.5696	0.5363	0.5740	0.7029	0.8218	0.8324	0.8728	0.8566
AUC(Mean)	0.6763	0.6772	0.6763	0.7480	0.7980	0.8075	0.8325	0.8114

show that the model trained with the auxiliary task outperformed the model trained without it. We give the results of fine-grained CW-CEL. Although this model's results were better than the baseline [1], it did not outperform CW-CEL. Our methods were designed to balance different labels. There is a significant difference between the distributions of normal and abnormal classes. However, the difference among the distributions of diseases were insignificant. We suspect that the fine-grained weights of CW-CEL drastically changed in each batch owing to the changes to the number of samples in each class. This led to unstable training.

CEL(trans.) denotes the CEL model with a transition layer. Comparing its results with those of CEL(S), we see that the transition layer did not improve overall multi-label classification performance.

To evaluate the model's reproducibility, we performed six experiments in which we randomly split the entire dataset into two parts: 70% for training, 10% for validation and 20% for testing as before. We then compared the model's AUC values for all 14 diseases. We show the results in FIGURE 5. We also randomly conducted statistical testing on the experiments the six times, and the results are shown in TABLE 3. The results shown in this table were generated by the CW-CEL(AT) in TABLE 2. CW-CEL(mean) denotes the corresponding mean AUC values of the diseases for six random experiments, whereas CW-CEL(std) denotes the corresponding standard deviation, and where the standard deviations of almost all diseases were around at 0.01. From the FIGURE 5 and the TABLE 3, the results show that the model's results are reproducible, and it is efficient.

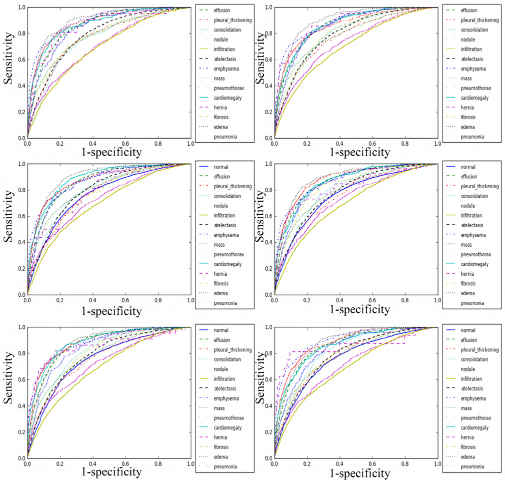


FIGURE 5. AUC results of six experiments with randomly split datasets.

TABLE 3. The statistic testing for six random experiments.

Pathology	CW-CEL(Mean)	CW-CEL(std)
Atelectasis	0.7882	0.0061
Cardiomegaly	0.8710	0.0119
Effusion	0.8620	0.0018
Infiltration	0.6957	0.0044
Mass	0.7961	0.0175
Nodule	0.7279	0.0150
Pneumonia	0.8457	0.0088
Pneumothorax	0.8652	0.0147
Consolidation	0.7916	0.0105
Edema	0.8917	0.0078
Emphysema	0.8970	0.0163
Fibrosis	0.8042	0.0165
Pleural Thickening	0.7816	0.0147
Hernia	0.8731	0.0309
Mean	0.8208	0.0126

E. COMPARISON WITH OTHER STATE-OF-THE-ART RESULTS

We compared the per-class (AUC) of the model with previous state-of-the-art results reported by CheXNet [2], Yao et al. [41] and Wang et al. [1] as shown in TABLE 4.

Our DeepCXray achieved close to state-of-the-art results on all 14 diseases classes. We obtained the best results for pneumonia. Rajpurkar et al. [2] reported that CheXNet can perform as well as human radiologists for this disease (an AUC of 0.788). Our best AUC for it is 0.8530. The results of AUC for other five trials were 0.8558, 0.8417, 0.8310, 0.8480, 0.8451. Pneumonia was labeled for 1237 times, which was 1% of the total number of all images. Further,

TABLE 4. Recent state-of-the-art results reported for the 14 diseases in the ChestXray14 dataset.

Pathology	Wang. et al. [1]	Yao, Li. et al. [41]	CheXNet [2]	DeepCXray(ours)
Atelectasis	0.7158	0.772	0.8209	0.7924
Cardiomegaly	0.8065	0.904	0.9048	0.8935
Effusion	0.7843	0.859	0.8831	0.8651
Infiltration	0.6089	0.695	0.7204	0.6957
Mass	0.7057	0.792	0.8618	0.8307
Nodule	0.6706	0.717	0.7766	0.7549
Pneumonia	0.6326	0.7130	0.7632	**0.8530**
Pneumothorax	0.8055	0.841	0.8932	0.8727
Consolidation	0.7078	0.788	0.7939	0.7841
Edema	0.8345	0.882	0.8932	**0.9010**
Emphysema	0.8149	0.829	0.9260	0.9211
Fibrosis	0.7688	0.767	0.8044	**0.8190**
Pleural Thickening	0.7082	0.765	0.8138	0.7983
Hernia	0.7667	0.914	0.9387	0.8728
AUC(mean)	0.7379	0.8027	0.8424	0.8325

TABLE 5. Results upon four diseases that contains the least samples.

Pathology	Hernia	Pneumonia	Fibrosis	Edema
Wang. et al. [1]	0.7667	0.6326	0.7688	0.8345
Yao, Li. et al. [41]	0.914	0.713	0.7670	0.8820
CheXNet [2]	0.9387	0.7632	0.8044	0.8932
DeepCXray(ours)	0.8834	**0.8530**	**0.8190**	**0.9010**

we removed the four diseases that contained the smallest number of samples for a comparison. The results are shown in TABLE 5. Our model outperformed the other models for three of diseases, excluding disease(henia). When evaluating the result by disease(henia), only 40 samples were labeled as those of henia. We conclude that our model (DeepCXray) can improve the performance for a minority category significantly by excluding other factors. The W-CEL method in this table is from Wang *et al.* [1], and our models outperformed theirs.

F. EVALUATION OF THE PROPOSED LOSS ON DIFFERENT ARCHITECTURES

In this section, we report the assessment of the proposed loss function on different architectures. We performed experiments on the most popular and powerful CNNs, i.e., residual block [8], inception [7], [11], and VGG [9]. These architectures have played a significant role in the development of CNNs.

VGGNet [9] explores the effect of the relationship between depth and width on performance. Simonyan *et al.* constructed two convolutional neural networks, VGG16 and VGG19, and achieved state-of-the-art results on ILSRC 2014. In the same year, Google InceptionNet [7], [11], [42], [43] was developed, and it can reduce the amount of calculation and the number of parameters. It has 22 layers, considerably deeper than AlexNet [4] and VGGNet. However the number of its parameters is only $\frac{1}{12}$ that of AlexNet's, even though it performs much better. Note that InceptionNet replaces the fully-connected layer with a global pooling layer, which can increase the speed of model training, and can reduce the probability of overfitting and the number of parameters further. This vision of inception was implemented in InceptionV1 [7]. Subsequent, versions of inception have been proposed. In which batch normalization and factorization on bigger convolution kernels have been introduced. ResNet was proposed by He *et al.* [8]. It allows for the transfer of the bottom of the original input to the top layers, which changes the process from one of learning the entire output to learning the difference between the input and the output. This architecture has enabled deep neural networks to become even deeper.

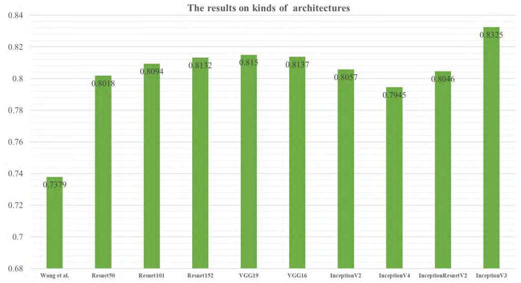
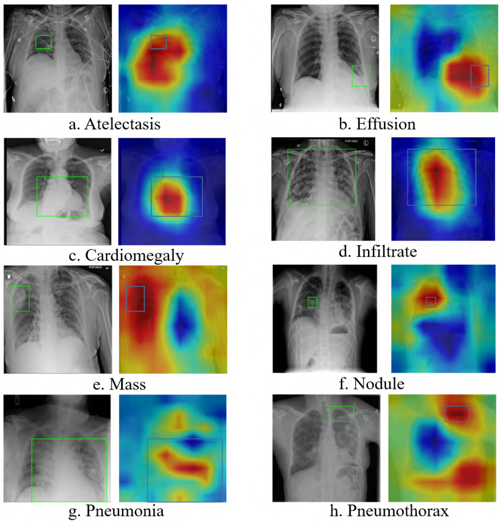


FIGURE 6. Mean AUC results. The model based on our loss function can outperform the baseline proposed by Wang et. al. [1].

To show that our loss function can be combined with some popular architectures, we trained different models. To eliminate the influence of irrelevant factors, the parameters of all models' were pre-trained on the ImageNet dataset. During the training process, we use a batch size of 16, the same learning algorithm as before, and a learning rate initialized at 10^{-5} and updated after 9, 000 steps. The results are shown in the TABLE 6. The results obtained by Resnet52, Resnet101, and Resnet152 [8] show that when the architecture was deeper, the results were better. Furthermore, The results of VGG16 and VGG19 [9] confirm this conclusion. The results obtained by InceptionV2 [42] and InceptionV3 [11] show that a deeper architecture yields better performance, and reveal that the kernel factorization can improve the models' performance. Because these methods reduce the total number of parameters, this makes training easier. Comparing the results of InceptionV4 [43] with those of InceptionV3, the former architecture was deeper than the latter, requiring

TABLE 6. **Results of combining several architectures with our loss function. Our loss function can perform stably with all kinds of architectures on this dataset. The best results are based on InceptionV3.**

Architecture	Resnet50	Resnet101	Resnet152	VGG16	VGG19	InceptionV2	InceptionV4	InceptionResnetV2	InceptionV3
Atelectasis	0.7737	0.7786	0.7728	0.7844	0.7805	0.7812	0.7815	0.7772	0.7924
Cardiomegaly	0.8747	0.8781	0.8824	0.9025	0.9127	0.8752	0.8269	0.8561	0.8935
Effusion	0.8545	0.8655	0.8641	0.8686	0.8705	0.8611	0.8311	0.8603	0.8651
Infiltration	0.6833	0.6814	0.6855	0.6946	0.6936	0.6895	0.6853	0.684	0.6957
Mass	0.7829	0.7961	0.7905	0.7995	0.8111	0.8131	0.7981	0.7931	0.8307
Nodule	0.6876	0.8649	0.7035	0.7195	0.7126	0.7205	0.6978	0.714	0.7549
Pneumonia	0.7681	0.7649	0.78	0.8336	0.8379	0.8325	0.8154	0.8202	0.8530
Pneumothorax	0.8573	0.8649	0.8641	0.847	0.8534	0.8565	0.8594	0.8608	0.8727
Consolidation	0.7630	0.7671	0.7584	0.7834	0.7872	0.7924	0.7548	0.7825	0.7841
Edema	0.8787	0.8871	0.8879	0.8917	0.8961	0.8868	0.8343	0.8789	0.9010
Emphysema	0.8922	0.8817	0.8949	0.8705	0.8799	0.8888	0.8767	0.8688	0.9211
Fibrosis	0.7786	0.7998	0.8089	0.7954	0.7995	0.8025	0.7923	0.7826	0.8190
Pleural Thickening	0.7559	0.7731	0.7721	0.7726	0.7716	0.7753	0.7482	0.7593	0.7983
Hernia	0.8740	0.8799	0.9195	0.8187	0.8037	0.8579	0.8214	0.8262	0.8728
AUC(Mean)	0.8018	0.8094	0.8132	0.8137	0.8150	0.8057	0.7945	0.8045	**0.8325**


a. Atelectasis

b. Effusion

c. Cardiomegaly

d. Infiltrate

e. Mass

f. Nodule

g. Pneumonia

h. Pneumothorax

FIGURE 7. **Diagnosis visualisation based on classification activation map.**

more inception modules to be added to the architecture, rendering it more complex. A more complex model is more challenging to train. We assumed that the scale of the dataset was not too large to train a complex model such as InceptionV4. From InceptionV4 to InceptionResnetV2 [43], the results improved again. Because residual networks are easier to train, the method incorporating inception modules with residual blocks can improve the performance of architecture.

We show the above results in FIGURE 6. All architectures, when combined with our loss function, outperformed the baseline results reported by Wang *et al.* [1].

G. VISUALISATION BASED ON CLASSIFICATION ACTIVATION MAP

In this dataset, there were 984 images containing diagnosis localization ground truth bounding boxes. There were eight kinds of diseases for which bounding boxes were available.

We employed the classification activation map(CAM) [22] to visualize the diagnosis features. The results are shown in FIGURE 7. For each disease, we plotted two images, one for the raw image and ground truth (left), and the other for the corresponding heat map obtained by CAM (right). For example, in FIGURE 7-*a*, the green bounding box indicates the ground truth. The image to the right shows the localization heatmap obtained by CAM. The blue bounding boxes to the right shows the ground truth. In these heatmaps, a region with warmer colors is more likely to be a lesion. We used a simple threshold to segment the heat map. The original heatmap had only one channel, and its values were integers in the range [0, 255]. The region with values close to 255 was the region of interest(ROI). We determined the bounding box for the largest connected component in the threshold map with a threshold of 200. From the examples shown in FIGURE 7, the ROI obtained by CAM was close to the ground truth bounding boxes.

V. CONCLUSIONS AND FUTURE WORK

In this study, a model called DeepCXRay was proposed to diagnose diseases automatically to address two kinds of issues in medical images. First, to effectively extract features from X-ray images, we employed a novel CNN architecture using InceptionV3 to extract features automatically. Because there no dataset is large enough to train the model, we used a model pretrained on ImageNet. Second, to address the problems of imbalanced datasets, we developed a loss function called CW-CEL. We combined the positive and negative weights with the cross-entropy terms to obtain state-of-the-art results. The results of experiments show that when a dataset is substantially imbalanced, the proposed loss function is much more effective.

In future research, we will study heuristic methods that can control the two terms automatically by learning algorithms based not only on samples batches. Of course, the problem with CADs is a sufficient amount of data are not always sufficient for some diseases. Thus, another direction of future work will involve constructing a suitable architecture, and using the limited amount of data with multiple label to train it while avoiding over-fitting.

REFERENCES

[1] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "ChestX-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proc. CVPR*, Jul. 2017, pp. 3462–3471.

[2] P. Rajpurkar *et al.* (2017). "CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning." [Online]. Available: https://arxiv.org/abs/1711.05225

[3] D. Jia, D. Wei, S. Richard, L. J. Li, K. Li, and F.-F. Li, "ImageNet: A large-scale hierarchical image database," in *Proc. CVPR*, Jun. 2009, pp. 248–255.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[5] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[6] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. (2016). "Densely connected convolutional networks." [Online]. Available: https://arxiv.org/abs/1608.06993

[7] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[8] K. He, X. Zhang, S. Ren, and J. Sun. (2015). "Deep residual learning for image recognition." [Online]. Available: https://arxiv.org/abs/1512.03385

[9] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[12] U. Avni, H. Greenspan, E. Konen, M. Sharon, and J. Goldberger, "X-ray categorization and retrieval on the organ and pathology level, using patch-based visual words," *IEEE Trans. Med. Imag.*, vol. 30, no. 3, pp. 733–746, Mar. 2011.

[13] H. Boussaid and I. Kokkinos, "Fast and exact: ADMM-based discriminative shape segmentation with loopy part models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 4058–4065.

[14] S. Hermann, "Evaluation of scan-line optimization for 3D medical image registration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3073–3080.

[15] S. Jaeger *et al.*, "Automatic tuberculosis screening using chest radiographs," *IEEE Trans. Med. Imag.*, vol. 33, no. 2, pp. 233–245, Feb. 2014.

[16] V. Gulshan *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *J. Amer. Med. Assoc.*, vol. 316, no. 22, pp. 2402–2410, 2016, doi: 10.1001/jama.2016.17216.

[17] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017. [Online]. Available: https://www.nature.com/articles/nature21056

[18] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*, S.-I. Amari and M. A. Arbib, Eds. Berlin, Germany: Springer, 1982, pp. 267–285.

[19] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006. [Online]. Available: 10.1162/neco.2006.18.7.1527

[20] M. Lin, Q. Chen, and S. Yan. (2013). "Network in network." [Online]. Available: https://arxiv.org/abs/1312.4400

[21] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision—ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 818–833.

[22] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[23] M. Buda, A. Maki, and M. A. Mazurowski. (2017). "A systematic study of the class imbalance problem in convolutional neural networks." [Online]. Available: http://arxiv.org/abs/1710.05381

[24] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, May 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417416307175

[25] G. Levi and T. Hassncer, "Age and gender classification using convolutional neural networks," in *Proc. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2015, pp. 34–42.

[26] N. Jaccard, T. W. Rogers, E. J. Morton, and L. D. Griffin, "Detection of concealed cars in complex cargo X-ray imagery using deep learning," *J. X-ray Sci. Technol.*, vol. 25, no. 3, pp. 323–339, 2017.

[27] A. Janowczyk and A. Madabhushi, "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases," *J. Pathol. Informat.*, vol. 7, no. 1, p. 29, 2016.

[28] K.-J. Wang, B. Makond, K.-H. Chen, and K.-M. Wang, "A hybrid classifier combining SMOTE with PSO to estimate 5-year survivability of breast cancer patients," *Appl. Soft Comput.*, vol. 20, pp. 15–24, Jul. 2014.

[29] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate Bayesian *a posteriori* probabilities," *Neural Comput.*, vol. 3, no. 4, pp. 461–483, 1991.

[30] Z.-H. Zhou and X.-Y. Liu, "Training cost-sensitive neural networks with methods addressing the class imbalance problem," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 1, pp. 63–77, Jan. 2006.

[31] S. Lawrence, I. Burns, A. Back, A. C. Tsoi, and C. L. Giles, *Neural Network Classification and Prior Class Probabilities*, vol. 1524. Springer, 1998, ch. 14, pp. 299–314.

[32] M. Z. Kukar and I. Kononenko, "Cost-sensitive learning with neural networks," in *Proc. ECAI*, 1998, pp. 445–449.

[33] J. Nathalie, M. Catherine, and G. Mark, "A novelty detection approach to classification," in *Proc. Int. Joint Conf. Artif. Intell.*, 1995, pp. 518–523.

[34] N. Japkowicz, S. J. Hanson, and M. A. Gluck, "Nonlinear autoassociation is not equivalent to PCA," *Neural Comput.*, vol. 12, no. 3, pp. 531–545, 2000.

[35] H. Sohn, K. Worden, and C. R. Farrar, "Novelty detection using auto-associative neural network," in *Proc. Symp. Identificat. Mech. Syst.*, 2001, pp. 187–199.

[36] M. Havaei *et al.*, "Brain tumor segmentation with deep neural networks," *Med. Image Anal.*, vol. 35, pp. 18–31, Jan. 2017.

[37] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Computer Vision—ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 499–515.

[38] L. Li and H. Wang. (2016). "Towards label imbalance in multi-label classification with many labels." [Online]. Available: http://arxiv.org/abs/1604.01304

[39] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[40] M. D. Zeiler. (2012). "ADADELTA: An adaptive learning rate method." [Online]. Available: https://arxiv.org/abs/1212.5701

[41] L. Yao, E. Poblenz, D. Dagunts, B. Covington, D. Bernard, and K. Lyman. (2017). "Learning to diagnose from scratch by exploiting dependencies among labels." [Online]. Available: https://arxiv.org/abs/1710.10501

[42] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167

[43] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. AIAA*, 2017, pp. 1–12.

XIUYUAN XU received the B.S. degree in mechanical engineering from the College of Computer Science, Sichuan University, Chengdu, China, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include neural networks.

QUAN GUO (S'15–M'18) received the Ph.D. degree from Sichuan University, Chengdu, China, in 2017. He is currently with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University. His current research interests include neural networks.

JIXIANG GUO (M'15) received the Ph.D. degree from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, in 2011. She is currently an Assistant Professor with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, China. Her research interests include neural networks and computer-assisted medical applications.

ZHANG YI (M'08–SM'09–F'16) received the Ph.D. degree in mathematics from the Institute of Mathematics, Chinese Academy of Science, Beijing, China, in 1994. He is currently a Professor with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, China. He is the co-author of three books: *Convergence Analysis of Recurrent Neural Networks* (Kluwer Academic Publishers, 2004), *Neural Networks: Computational Models and Applications* (Springer, 2007), and *Subspace Learning of Neural Networks* (CRC Press, 2010). His current research interests include neural networks and big data. He was an Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems from 2009 to 2012. He has been an Associate Editor of the IEEE Transactions on Cybernetics since 2014.