scientific reports

OPEN

Searching for pneumothorax in x-ray images using autoencoded deep features

Antonio Sze-To[1], Abtin Riasatian[1] & H. R. Tizhoosh[1,2]✉

Fast diagnosis and treatment of pneumothorax, a collapsed or dropped lung, is crucial to avoid fatalities. Pneumothorax is typically detected on a chest X-ray image through visual inspection by experienced radiologists. However, the detection rate is quite low due to the complexity of visual inspection for small lung collapses. Therefore, there is an urgent need for automated detection systems to assist radiologists. Although deep learning classifiers generally deliver high accuracy levels in many applications, they may not be useful in clinical practice due to the lack of high-quality and representative labeled image sets. Alternatively, searching in the archive of past cases to find matching images may serve as a "virtual second opinion" through accessing the metadata of matched evidently diagnosed cases. To use image search as a triaging or diagnosis assistant, we must first tag all chest X-ray images with expressive identifiers, i.e., deep features. Then, given a query chest X-ray image, the majority vote among the top *k* retrieved images can provide a more explainable output. In this study, we searched in a repository with more than 550,000 chest X-ray images. We developed the Autoencoding Thorax Net (short *AutoThorax* -Net) for image search in chest radiographs. Experimental results show that image search based on *AutoThorax* -Net features can achieve high identification performance providing a path towards real-world deployment. We achieved 92% AUC accuracy for a semi-automated search in 194,608 images (pneumothorax and normal) and 82% AUC accuracy for fully automated search in 551,383 images (normal, pneumothorax and many other chest diseases).

Pneumothorax (collapsed or dropped lung) is an emergency condition when air enters the pleural space, i.e., the space between the lungs and the chest wall[1,2]. It is one of the diseases in the *Category 1* findings that should be communicated to clinicians within minutes in order to take immediate actions to avoid fatalities as recommended by the American College of Radiology (ACR)[3,4]. A graphical illustration of pneumothorax is provided in Fig. 1.

It is generally a serious condition that can be fatal[1]. To prevent patient death, early detection of pneumothorax through application of deep learning may be a viable option[4]. Pneumothorax is typically detected on chest X-ray images by qualified radiologists[2]. However, nowadays radiologists in many have to process many X-ray studies daily[6]. With the increasing workload, the large volume of work for radiologists understandably delays diagnosis and treatment. In this process, experience is absolutely necessary but even the most experienced expert may be prone to miss the subtleties of an image[7,8]. Since a wrong or delayed diagnosis can cause harm to patients[9], it is vital to develop computer-aided approaches to assist radiologists in their daily workflow.

Chest X-ray is the most common medical imaging modality with over 35 millions images taken every year in the U.S. alone[10]. X-ray images allow for inexpensive screening of several chest conditions including pneumothorax[6]. Since hospital daily workloads may result in long queues for radiology images to be read, including images acquired overnight or images without any clinical pre-screening, an automated method of inspecting chest X-rays and prioritizing studies with potentially positive findings for rapid review may reduce the delay in diagnosing and treating pneumothorax[11].

Due to its recent success, an increasing number of studies have adopted "*deep learning*" for processing digital images, referring to the use of Deep Neural Networks (DNN), defined as artificial neuronal networks with more 3 hidden layers[12] (DNNs practically consisting of many more hidden layers), to detect pneumothorax or other thoracic diseases in chest X-ray images[11,13–23]. The deep-learning pneumothorax detection systems can be categorized into two categories: (1) detection methods, i.e., to pinpoint the exact location of certain thoracic diseases

[1]Kimia Lab, University of Waterloo, Waterloo, ON N2L 3G1, Canada. [2]Vector Institute, MaRS Centre, Toronto, ON M5G 1M1, Canada. ✉email: tizhoosh@uwaterloo.ca

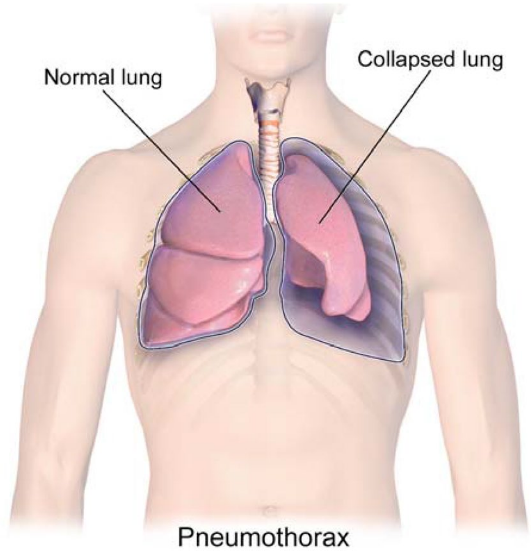


Figure 1. A graphical illustration of pneumothorax. [Medical gallery of Blausen Medical 2014, WikiJournal of Medicine[5]].

in an input chest X-ray image, and (2) classification methods, i.e., to identify the presence of certain thoracic diseases in an input chest X-ray image without highlighting the exact location of the disease.

From a practical point of view, detection systems remain hard to be developed as large high-quality datasets with pixel-level labels are needed to train such systems. These datasets are expensive to obtain as creating representative and accurate labels constitutes tedious manual work for radiologists (for instance, many works focus on large and medium size pneumothorax[11]). Classification systems, on the other hand, are relatively easier to develop as they only need image-label annotation.

So far, more than half a million chest X-ray images with image-level (or global) labels have been released. These are ChestX-ray14[13], CheXpert[19] and MIMIC-CXR[24]. Leveraging such a large amount of labelled images, classification-based systems should not be difficult to train and deploy. Nevertheless, the main drawback of a classification system is that it only outputs a single probability, a number that quantifies the likelihood of the chest X-ray to contain a certain abnormality. This may not be enough to justify the diagnosis.

Image search, as a different approach to classification, not only can provide a probabilistic output like a classifier by taking a weighted majority vote among matched images but also can provide access to the metadata of similar cases from the past, a functionality that a trained deep network for classification cannot offer. Hence, image search allows a comparison of patient histories and treatment profiles due to providing a list of matched cases and not just delivering a classification probability[25]. Therefore, image search may enable a virtual "second opinion" for diagnostic purposes and provide computerized explanations for the decision support. While image search may be more viable for clinical deployment in terms of explainability, its classification performance still needs to be investigated, specifically, whether image search can achieve a identification performance as high as those obtained by classification-based deep learning systems.

In this study, we explored the use of image search based on deep features obtained from DNNs to detect pneumothorax in chest X-ray images. By means of using image search as a classifier, all chest X-ray images were first tagged with a feature vector. Given a query chest X-ray image, the majority voting of the top k retrieved chest X-ray images was then used as a classifier. The corresponding reports and other clinical metadata of the top search results can also be used if available. This is an inherent benefit of image search.

Our contributions in this study are two-folded. Firstly, we developed *AutoThorax*-Net that generates feature vectors from integrating multiple images into one feature vector. Although the benefits of using deep features for processing x-ray images is well-established, in this study we demonstrated breaking down the image into multiple sub-images, here by using the chest symmetry, does in fact provide better results; the separation of left and right lung accompanying the entire image apparently increases the recognition accuracy. Flipping one lung—as a coarse type of image registration—may also contribute to the better feature matching. Experimental results demonstrate that image search on *AutoThorax*-Net features can achieve a higher detection performance compared with using feature vectors solely from the input image. Experimental results also showed that image search based using *AutoThorax*-Net features with the dimensionality reduced by a factor of 12 times can achieve a detection performance, comparable to or outperforming those obtained by existing systems such as CheXNet[13].

Related works

Deep learning for analyzing chest x-ray images.

Since the release of the *ChestX-ray14* dataset[13] by National Institute of Health, providing 112,120 frontal-view X-ray images of 30,805 unique patients labelled for 14 diseases (in which each image may have multi-labels), an increasing number of studies have adopted DNNs to develop automated systems to detect diverse diseases on chest X-ray images[11,14–22]. *CheXNet*[14], a DNN with

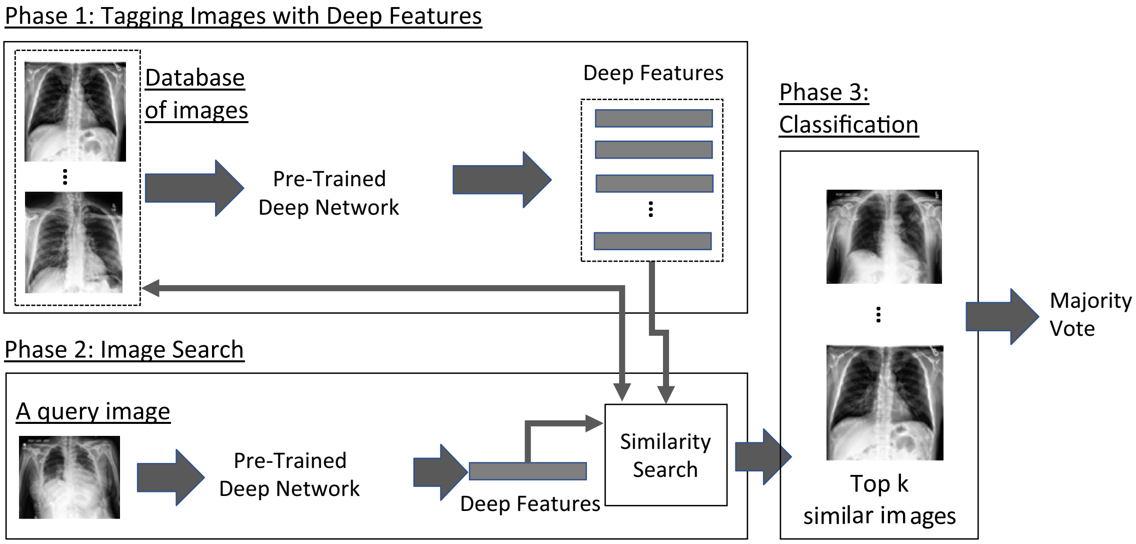


Figure 2. An overview of using image search as a classifier to recognize pneumothorax in chest X-ray images. Phase 1: Tagging images with Features. Phase 2: Image Search (distance calculation between the query features and all other features in the database). Phase 3: Classification (the majority voting of the retrieved images as a classifier).

DenseNet121 architecture[26], has been trained on ChestX-ray14 dataset and achieved radiologist-level detection of pneumonia. Since then, many DNN architectures have been proposed for a variety of tasks ranging from localization[21], lateral and frontal dual chest X-ray reading[15], integration of non-image data in classification[20], attention-guided approaches[27], location-aware schemes[6], weakly-supervised methods[28,29] as well as generative models[8].

For detecting pneumothorax, a recent study collected 13,292 frontal chest X-rays (3107 with pneumothorax) to train a DNN to verify the presence of a large or moderate-sized pneumothorax. Another recent study[4] collected 1003 images (437 with pneumothorax and 566 with no abnormality) to detect pneumothorax with DNNs. So far, there has been no study to investigate pneumothorax detection in a large dataset, perhaps by combining the three large public datasets.

Deep learning for content-based image retrieval.

Retrieving similar images given a query image is known as Content-Based Image Retrieval (CBIR)[30] or Content-Based Medical Image Retrieval (CBMIR)[31] for medical applications. While classification-based methods provide promising results[32], CBMIR systems can assist clinicians by enabling them to compare the case they are examining with previous (already diagnosed) cases and by exploiting the information in corresponding medical reports[31]. It may also help radiologists in faster and more reliably preparing reports for particular diagnosis[31].

While deep learning methods have been applied to CBIR tasks in recent studies[33,34], there has been less attention on exploring deep learning methods for CBMIR tasks[35,36].

One study investigated the performance of DNNs for MR and CT images with human anatomy labels[35]. Another study investigated the retrieval performance of DNNs among multimodal medical images for different body organs[36]. There is also a study exploring hashing deep features into binary codes, testing among lung, pancreas, neuro and urothelial bladder images[37]. Deep Siamese Convolutional Neural Networks[38] have also been tested for CBMIR to minimize the use of expert labels using multiclass diabetic retinopathy fundus images. Another study explored deep learning for CBMIR among multiple modalities for a large number of classes[39]. So far, there has not been any report to validate CBMIR techniques for a challenging case like pneumothorax detection in large datasets. We attempt to close this gap by reporting our results on a large dataset of chest x-ray images by fusion three public datasets.

Methods

Given a query chest X-ray image, the problem is to output whether it contains pneumothorax using image search in archived images through majority vote among retrieved similar images from the archive.

The proposed method of using image search as a classifier comprises of three phases (Fig. 2): (1) Tagging images with deep features (all images in the database are fed into a pre-trained network to extract features), (2) image searching (tagging the query image with features and calculating its distance with all other features in the archive to find the most similar images), (3) classification (majority vote among the labels of retrieved similar images).

Phase 1: tagging images with deep features.

In this phase, all chest X-ray images in the archive are tagged with deep features. To represent a chest X-ray image as a feature vector with a fixed dimension, the last

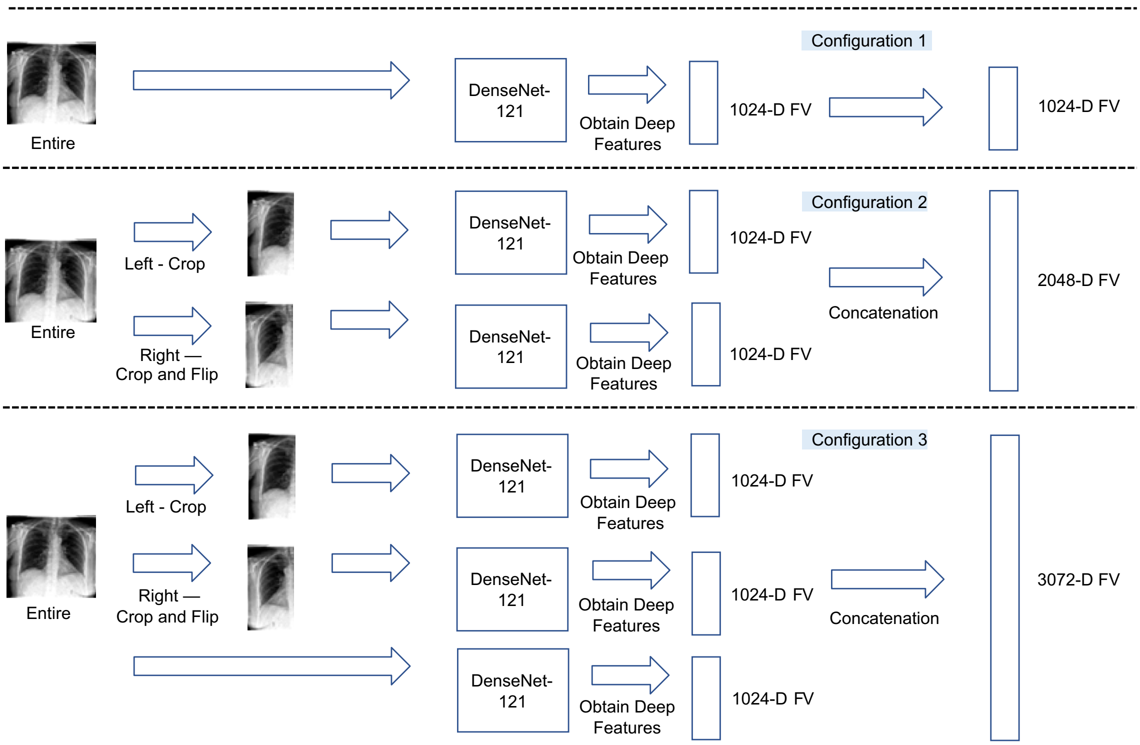
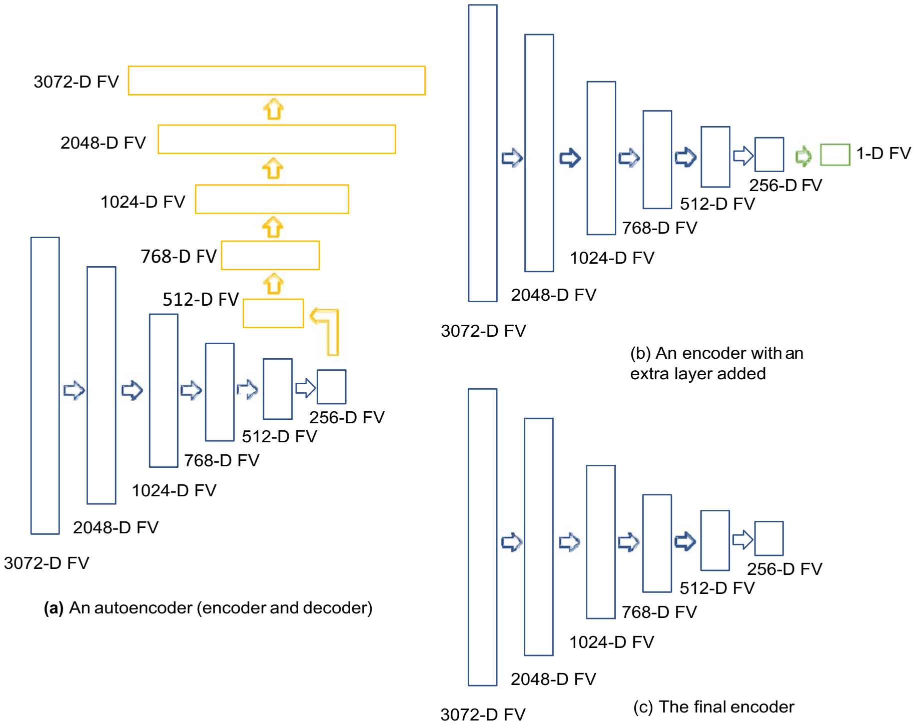


Figure 3. An overview of the three configurations using DenseNet121[26] to extract features from a chest X-ray images. Configuration 1: a feature vector is extracted from the entire chest X-ray image. Configuration 2: two feature vectors are extract from the left chest side and flipped right chest side. The final feature vector is a concatenation of these two feature vectors. Configuration 3: three feature vectors are extract from the left chest side, the flipped right chest side, and the entire chest X-ray image, respectively.

pooling layer may be used as image representation. In other words, the pre-trained deep convolutional neuronal network is considered as a feature extractor to convert a chest X-ray image into an n-dimensional feature vector with $n = 1, 024$ being a typical value for many networks. In our study, DenseNet121[26] is used for converting a chest X-ray image into a feature vector with 1,024 dimensions. DenseNet topology has a strong gradient flow contributing to diverse features and is, compared to many other architectures such as ResNet and EfficientNet, quite compact with *only* 7 million weights. We adopted DenseNet121 also for a fair comparison in experiments with CheXNet, which has also used DenseNet121 as its backbone architecture. Three configurations are explored to extract deep features from a chest X-ray image (Fig. 3):

- *Configuration 1*—a feature vector is extracted from the entire chest X-ray image. Representing the entire image with one feature vector is quite common and assumes that the object or abnormality will be adequately quantified in the single feature vector for the entire image.
- *Configuration 2*—two feature vectors are extracted, one from the left chest side and one from the flipped version of the right chest side. The final feature vector is a concatenation of these two feature vectors. If DenseNet121[26] is adopted as the feature extractor, the feature vector has 2, 048 values. The rational behind this idea is to allow expressive features for each side of the chest to be quantified separately to make feature matching easier for the unsupervised search. As well, flipping the right lung is a registration-like operation to facilitate alignment in matching.
- *Configuration 3*—three feature vectors are extracted as described in previous two configurations. The final feature vector is a concatenation of these three feature vectors. If DenseNet121[26] is adopted as the feature extractor, the dimension of the final feature vector is 3072 real-valued features. The rationale behind this configuration is that matching a combined feature vector that represents the whole image, the left chest side and the flipped right chest side not only provides a global image view but also more focused and aligned attention to each chest side to emphasize their features in the search and matching process.

Phase 2: image search. In this phase, the distance between the deep features of the query chest X-ray image and all chest X-ray images in the database are computed. The chest X-ray images having the shortest distance with those of the query chest X-ray image are subsequently retrieved. The Euclidean distance, as the most commonly used norm for deep feature matching, was used for computing the distance between the deep features of two given chest X-ray images. It is the geometric distance in the multidimensional space recommended when all variables have the same metric[40]. The calculated distances can be sorted to retrieve as many as matched images as desired. The impact of distance norms on retrieval may be investigated in future works.



(a) An autoencoder (encoder and decoder)

(b) An encoder with an extra layer added

(c) The final encoder

Figure 4. An overview of autoencoder topologies: (**a**) an autoencoder, with an encoder (highlighted in blue) and an decoder (highlighted in yellow), is constructed during step 1: Unsupervised end-to-end training with decoder, (**b**) an encoder (highlighted in blue) with an extra layer added (highlighted in green) is constructed during step 2: supervised fine-tuning with labels, (**c**) an encoder (highlighted in blue) is constructed by removing the 1-dimension layer.

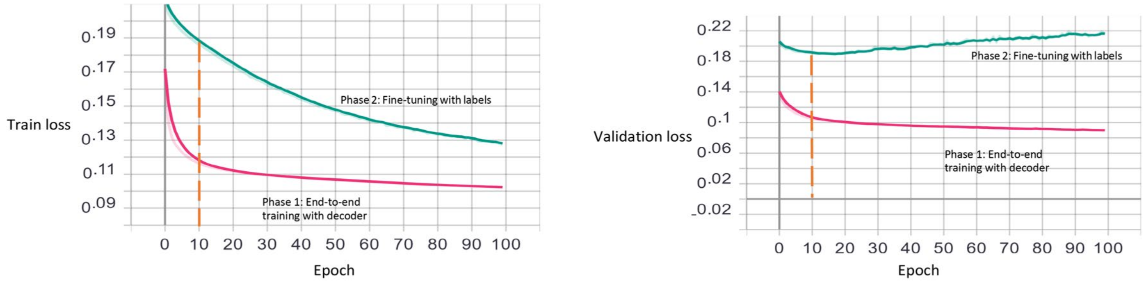
Phase 3: classification. In this phase, the majority vote among the labels of retrieved chest X-ray images is used as a classification decision. For example, given a query chest X-ray image, the top k most similar chest X-ray images are retrieved. If m chest X-ray images are labelled with pneumothorax (with $m \leq k$), the query image is classified as pneumothorax with a class likelihood of m/k. The larger k the more reliable the classification vote will become. This, in turn, requires a large archive of tagged images to increase the probability of finding similar images.

Compressing feature dimensionality using autoencoders. The dimensionality of the feature vectors, especially the concatenated ones, may become a computational obstacle but it can be reduced by employing autoencoders. One may use an autoenoder for all configurations but our main motivation was a size reduction for the longest feature vector for configuration 3. Two steps are required to construct an encoder to reduce feature vector dimensionality:

- *Step 1: Unsupervised end-to-end training with a decoder* An autoencoder with the architecture summarized in Fig. 4a is first constructed. A dropout layer[41] with a probability of 0.2 is introduced between each layer to reduce the probability of overfitting. The model is then trained for 10 epochs by backpropagation with outputs being set equal to inputs. The batch size, loss function and optimizer were set to 128, Mean Squared Error and Adam, respectively. The training details are visualized in Fig. 5.
- *Step 2: Supervised fine-tuning with labels* After the training, the decoder in the model is removed as we only need the encoding part as dimensionality reduction. Instead, a one-dimensional fully connected layer of neurons with the sigmoid function as activation function was used in training phase. The network was trained for 10 epochs with the batch size of 128 using binary cross-entropy loss function and Adam optimizer. The training details are visualized in Fig. 5. During training, to deal with class imbalance, individual class weight was set for each class using the following formula:

$$W_{cj} = \frac{S}{C \times S_{cj}} \tag{1}$$

where $w_{c\,j}$ is the class weight of class $c_{\,j}$; C is the total number of classes; S is the total number of training samples; $S_{c\,j}$ is the total number of training samples belonging to class $c_{\,j}$.



(a) Training Loss

(b) Validation Loss

Figure 5. Visualization of training and validation loss in both phases (for the first fold). The weights of the 10th epoch, for both phases, are used in the experiments.

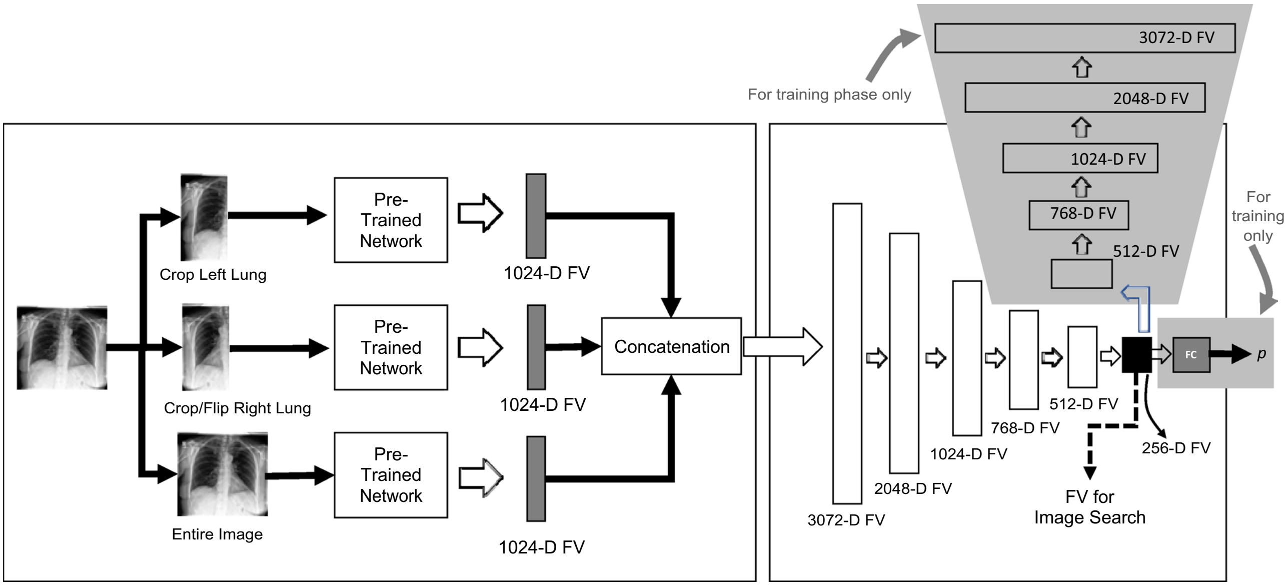


Figure 6. A graphical illustration of *AutoThorax*-Net. Three extracted feature vectors (FVs) (left) are concatenated and fed into into an encoder to compress them into 256 values for image search. During training the same compressed FV is used as input to a fully connected layer (FC) to classify images as pneumothorax with a likelihood of *p*.

The model architecture is summarized in Fig. 4b. Similarly, a dropout layer[41] was introduced between the 256-dimensional layer and the one-dimensional layer to reduce the probability of overfitting. The model is then trained with through backpropagation. After the training, the one-dimensional fully connected layer is removed. The model architecture of the final encoder is summarized in Fig. 4c.

Model architecture.

The architecture of *AutoThorax*-Net to obtain features from a chest X-ray image is illustrated in Fig. 6.

Results

In this section, we first describe the datasets collected and the prepossessing procedures. We then describe the experiments that were conducted, followed by the analysis. The main goal of experiments is to validate the performance of image search via matching deep features. In order to establish performance quantification, we treat search like a classifier by taking a consensus vote guided by the ROC statistics. We also compare the results with the CheXNet (without any modification or fine-tuning) which is an end-to-end deep network specially trained for classifying chest X-ray images.

Dataset collection.

Three large public datasets of chest X-ray images were collected. The first is MIMIC-CXR[24,42], a dataset of 371,920 chest X- rays associated with 227,943 imaging studies. A total of 248,236 frontal chest X-ray images in the training set were used in this study. The second dataset is CheXpert[19], a dataset consisting of 224,316 chest radiographs belonging to 65,240 patients. A total of 191,027 frontal chest X-ray images in the training set were used in this study. The third dataset is ChestX-ray14[13] consisting of 112,120 frontal-view X-ray images of 30,805 patients. All chest X-ray images in this dataset were used in this study.

	MIMIC-CXR[24,42]	CheXpert[19]	ChestX-ray14[13]	Total
+ ve: pneumothorax	11,610	17,693	5,302	34,605
− ve: normal	82,668	16,974	60,361	160,003
Total	94,278	34,667	65,663	194,608

Table 1. A summary of chest X-ray images in the Dataset 1 through combination of three public datasets.

	MIMIC-CXR[24,42]	CheXpert[19]	ChestX-ray14[13]	Total
+ ve: pneumothorax	11,610	17,693	5302	34,605
− ve: Non-pneumothorax	236,626	173,334	106,818	516,778
Total	248,236	191,027	112,120	551,383

Table 2. A summary of chest X-ray images in the Dataset 2 through combination of three public datasets.

In total, 551,383 frontal chest X-ray images were used in our investigations. 34,605 images (6% of of all images) were labelled as pneumothorax. The labels refer to the entire image; the collapsed lungs were not highlighted in any way.

Implementation and parameter setting. We used the library *Keras* (http://keras.io/) v2.2.4 with Tensorflow backend[43] to implement the approach. As we used a pre-trained network for feature extraction, the DenseNet121 was selected[26], and the weight file was obtained through the default setting of Keras. For CheXNet[14], the weight file was downloaded from GitHub (https://github.com/brucechou1983/CheXNet- Keras). All images were resized to 224 224 before feeding into networks. All other parameters were default values unless otherwise specified. All experiments were run on a computer with 64.0 GB DDR4 RAM, an Intel Core i9-7900X @3.30 GHz CPU (10 Cores) and one GTX 1080 graphic card.

Performance evaluation. Following relevant literature[14,19], the performance of classification was evaluated by the area under the curve (AUC) for the receiver operating characteristic curve (ROC curve) to enable the comparison over a range of prediction thresholds. As a tenfold cross-validation was conducted in the experiments, average ROC was computed with 95% confidence interval.

Dataset preparation & preprocessing. There is a concern for ChestX-ray14[13] dataset that its chest X-ray images with chest tubes were frequently labelled with Pneumothorax[44,45]. As we combined ChestX-ray14 with CheXpert[19], and MIMIC-CXR[24,42] datasets in our experiments, the concern was mitigated to address the bias.

Dataset 1 (semi-automated detection). If there is a suspicion of pneumothorax by the user, then the search will limited to the archived images that are either normal (i.e., no finding) or pneumothorax. We This is a dataset comprising of 34,605 pneumothorax chest X-ray images and 160,003 normal chest X-ray images. Searching in this dataset means there is already a suspicion by the expert that the image may contain pneumothorax, hence the search is guided to only search within archived images that are diagnosed as either pneumothorax or normal (no finding). The pneumothorax images were obtained from the collected frontal chest x-ray images with the label "Pneumothorax". They were considered as the positive (+ ve) class. The normal images were obtained from the collected frontal chest x-ray images with label the "No Finidng". These chest X-ray images were considered as the negative (− ve) class. A summary of dataset 1 is provided in Table 1.

Dataset 2 (fully-automated detection). If there is no concrete suspicion from user, we match the input image against all other images regardless of their tagged disease label. This dataset is comprising of 34,605 pneumothorax chest X-ray images and 516,778 non-pneumothorax chest x-ray images. Searching in this dataset means the computer is automatically searching in all images to verify the likelihood of pneumothorax without any guidance of the expert. The pneumothorax images were obtained from the collected frontal chest X-ray images with the label "Pneumothorax". They were considered as the positive (+ ve) class. The non-pneumothorax images were obtained from the collected frontal chest X-ray images without the label "Pneumothorax", meaning that they may contain cases such as normal, pneumonia, edema, cardiomegaly and more. They were considered as the negative (-ve) class. A summary of dataset 2 is provided in Table 2.

First experiment series: semi-automated solution. The first experiments series focuses on a "semi-automated" solution for pneumotharx. We confine the search and classification to cases that are either normal or diagnosed with pneumotharx (Dataset 1). We test all three configurations (Fig. 3), CheXNet, and the proposed *AutoThorax*-Net.

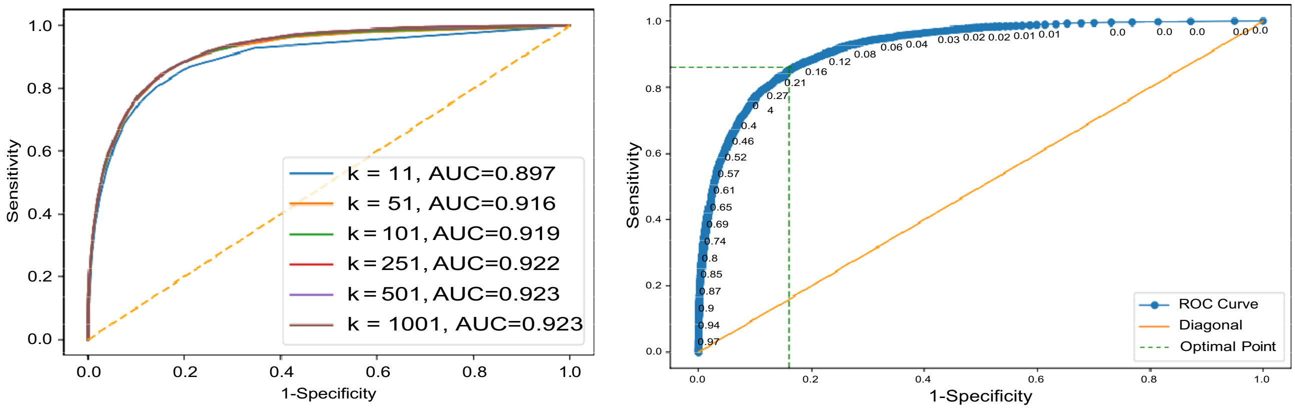


Figure 7. Analysis for Dataset 1. Left: Sample ROC curves for the proposed *AutoThorax*-Net for different k values in one fold and their area under the curve (AUC), Right: corresponding ROC thresholds for $k = 1001$ to select the sensitivity–specificity trade-off using Youden's index.

Experimental workflow. All images of Dataset 1 for all configurations were first tagged with deep features. We constructed the receiver operating characteristics (ROC) curve for the dataset to find the trade-off between sensitivity and specificity (Fig. 7). We used Youden's index[46] to find the trade-off position on the ROC curve providing the threshold for match selection. The Youden's index can be calculated as "*sensitivity* + *specificity* 1". A standard tenfold cross-validation was adopted for tests that showed a very low standard deviation for all experiments apparently due to the large size of the datasets. All tagged chest X-ray images were divided into 10 groups. In each fold, one group of chest X-ray images was used as validation set, while the remaining chest X-ray images were used as "archived" images to be searched. The above process was repeated 10 times, such that in each fold a different group of chest X-ray images was used as the validation set. In each fold, an encoder was trained using the archived set of that fold. The encoder was then used for compressing deep features for each chest X-ray image in the validation set.

The parameters of the encoder construction process are described as follows:

Step 1 Unsupervised end-to-end training with decoder: The training epoch and batch size were set as 10 and 128, respectively. The loss function was chosen as mean-squared-error. Adam optimizer[47] was used. The dropout rate was set to 0.2, i.e., a probability of 20% setting the neuron output as zero to counteract possible overfitting.

Step 2 Supervised fine-tuning with labels: The loss function was chosen as binary cross-entropy. Other parameters remained the same as in Step 1.

Given a query chest X-ray image from validation sets, image search was conducted on the archived set to retrieve k similar images for each query image. The consensus vote among the top k retrieved chest X-ray images subsequently determines whether the query image is pneumothorax. Results were generated with k 11, 51, 101, 251, 501, 1001. As we were using a large number of archived images, one was excepting to see better results for higher k values.

Results. Experimental results on Dataset 1 are summarized in Table 3 for *AutoThorax*-Net, ChexNet and all three feature configurations from Fig. 3. We calculated area under the curve (AUC), sensitivity and specificity for all 10 folds. Standard deviations were quite low (< 1%), hence not reported. Figure 8 shows the confusion matrices for both *AutoThorax*-Net and ChexNet.

The average sensitivity and specificity obtained by Configuration 3 for $k = 1001$ are higher than those obtained by Configuration 1 although they have almost the same AUC. Configuration 1 shows higher sensitivity for $k = 11$ (86% versus 83%) but its specificity is lower than Configuration 3 (76% versus 80%). Configuration 2 delivers the same AUC in range 88% but in individual comparison is always worse than other configurations with lower sensitivity and specificity.

AutoThorax-Net has clearly the highest AUC (92%). ChexNet delivers an AUC of 88% similar to the three search configurations. The highest sensitivity is 86% achieved by all tested methods. However, *AutoThorax*-Net also provides a specificity of 84% whereas the specificity of all other methods, including ChexNet, are in the 70% range.

To verify that the improvements of the proposed methods are significant, we have performed the two-sided Wilcoxon Signed-Rank test[48] between the performance of CheXNet and our best performing configuration which is *AutoThorax*-Net with $k = 1001$. Our results, shown in Table 3, suggest that AUC and Specificity are improved from 88 to 92 and 76 to 84, respectively. The calculated p-values for these two metrics, both 0.005, are smaller than 0.05 and reject the null hypothesis which means significant differences exist between the performance of *AutoThorax*-Net with $k = 1001$ and CheXNet with respect to these two metrics.

Second experiment series: automated solution. In these experiments, we investigated the possibility of constructing a "fully automated" solution by searching the entire archive, i.e., Dataset 2. We summarize the experimental workflow, and report the results with some analysis.

Method	Sensitivity	Specificity	AUC
CheXNet[14] classifier	86	76	88
Search via *AutoThorax*-net features ($k = 1001$)	86	84	92
Search via *AutoThorax*- net features ($k = 501$)	86	83	92
Search via *AutoThorax*- net features ($k = 251$)	84	85	92
Search via *AutoThorax*- net features ($k = 101$)	85	84	92
Search via *AutoThorax*- net features ($k = 51$)	85	84	92
Search via *AutoThorax*- net features ($k = 11$)	81	86	90
Search via configuration 3 (3072 features, $k = 1001$)	85	74	88
Search via configuration 3 (3072 features, $k = 501$)	84	76	88
Search via configuration 3 (3072 features, $k = 251$)	81	79	89
Search via configuration 3 (3072 features, $k = 101$)	84	78	89
Search via configuration 3 (3072 features, $k = 51$)	86	77	89
Search via Configuration 3 (3072 features, $k = 11$)	83	80	88
Search via configuration 2 (2048 features, $k = 1001$)	86	70	87
Search via configuration 2 (2048 features, $k = 501$)	85	73	88
Search via configuration 2 (2048 features, $k = 251$)	84	75	88
Search via configuration 2 (2048 features, $k = 101$)	84	77	88
Search via configuration 2 (2048 features, $k = 51$)	79	81	88
Search via configuration 2 (2048 features, $k = 11$)	80	81	87
Search via configuration 1 (1024 features, $k = 1001$)	80	78	88
Search via configuration 1 (1024 features, $k = 501$)	81	78	88
Search via configuration 1 (1024 features, $k = 251$)	80	80	89
Search via configuration 1 (1024 features, $k = 101$)	81	80	89
Search via configuration 1 (1024 features, $k = 51$)	83	80	89
Search via configuration 1 (1024 features, $k = 11$)	86	76	88

Table 3. A summary of classification performance using image search as a classifier on Dataset 1. The numbers (in percentage) are the result of averaging 10 folds with very low standard deviation (< 1%).

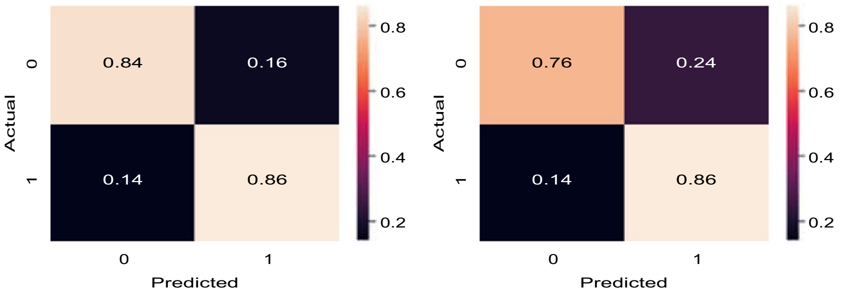


Figure 8. Dataset 1: Confusion matrices for the best results of *AutoThorax*-Net (left) and CheXNet (right).

Experimental workflow. We constructed the receiver operating characteristics (ROC) curve for Dataset 2 to find the trade-off between sensitivity and specificity (Fig. 9). We used Youden's index to find the trade-off position on the ROC curve providing the threshold for match selection. A standard tenfold cross-validation was adopted for testing that showed a very low standard deviation (< 1%) for all experiments apparently due to the large size of the datasets. All chest X-ray images were divided into 10 folds. In each fold, one group of chest X-ray images was used as validation set, while the remaining chest X-ray images were used as archived set. The above process was repeated 10 times, such that in each fold a different group of chest X-ray images was used as the validation set. In each fold, an encoder was trained using the archived set of that fold. The encoder was then used for compressing deep features for each chest X-ray image in the validation set.

The parameters of the encoder construction process were set as before described for Dataset 1.

For image search, given a chest X-ray image (from the validation set), the compressed deep feature was used for searching in the archived set. The consensus vote among the top k retrieved chest X-ray images to classify the query image from the validation set. Experiments were conducted with k 11, 51, 101, 251, 501, 1001 to observe the effect of more retrievals on consensus voting. For comparison, CheXNet[14] was adopted as a baseline to be applied to the validation set in each fold.

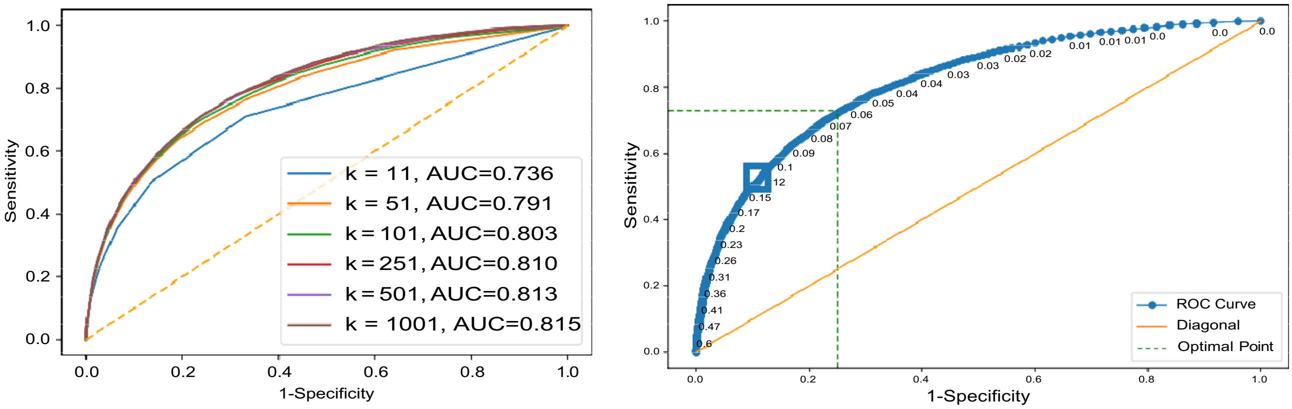


Figure 9. Analysis for Dataset 2. Left: Sample ROC curves for the proposed *AutoThorax*-Net for different k values in one fold and their area under the curve (AUC), Right: corresponding ROC thresholds for $k = 1001$ to select the sensitivity–specificity trade-off using Youden's index.

Method	Sensitivity	Specificity	AUC
CheXNet[14] classifier	72	67	77
Search via *AutoThorax*- net features ($k = 1001$)	73	75	82
Search via *AutoThorax*- net features ($k = 501$)	73	75	82
Search via *AutoThorax*- net features ($k = 251$)	72	75	82
Search via *AutoThorax*- net features ($k = 101$)	69	78	81
Search via *AutoThorax*- net features ($k = 51$)	70	75	80
Search via *AutoThorax*- net features ($k = 11$)	72	67	74
Search via configuration 3 (3072 features, $k = 1001$)	72	63	75
Search via configuration 3 (3072 features, $k = 501$)	70	67	76
Search via configuration 3 (3072 features, $k = 251$)	71	67	76
Search via configuration 3 (3072 features, $k = 101$)	74	65	77
Search via configuration 3 (3072 features, $k = 51$)	65	74	76
Search via configuration 3 (3072 features, $k = 11$)	72	65	72
Search via configuration 2 (2048 features, $k = 1001$)	67	66	74
Search via configuration 2 (2048 features, $k = 501$)	64	70	75
Search via configuration 2 (2048 features, $k = 251$)	74	61	75
Search via configuration 2 (2048 features, $k = 101$)	70	66	75
Search via configuration 2 (2048 features, $k = 51$)	73	63	75
Search via configuration 2 (2048 features, $k = 11$)	68	66	70
Search via configuration 1 (1024 features, $k = 1001$)	73	61	74
Search via configuration 1 (1024 features, $k = 501$)	67	68	75
Search via configuration 1 (1024 features, $k = 251$)	67	69	75
Search via configuration 1 (1024 features, $k = 101$)	70	65	75
Search via configuration 1 (1024 features, $k = 51$)	67	68	74
Search via configuration 1 (1024 features, $k = 11$)	71	60	69

Table 4. A summary of classification performance using image search as a classifier on Dataset 2. The numbers (in percentage) are the result of averaging 10 folds with very low standard deviation ($< 1\%$).

Results. Experimental results on Dataset 2 are summarized in Table 4. Figure 10 shows the confusion matrices for *AutoThorax*-Net and ChexNet.

The highest AUC of 82% is achieved by *AutoThorax*-Net for $k = 251$, 501 and 1001. The highest sensitivity of 74% is achieved by Configuration 2 (for $k = 251$) and Configuation 3 (for 101). However, they both deliver low specificity values of 61% and 65%, respectively. The second highest sensitivity of 73% is achieved by Configuration 1, Configuration 2 and *AutoThorax*-Net. Their specificity is 61%, 63% and 75%, respectively. *AutoThorax*-Net can clearly provide a higher and more reliable trade-off between sensitivity and specificity in a fully automated setting when applied on a large archive of X-ray images.

To verify that the improvements of the proposed methods are significant, we have performed the two-sided Wilcoxon Signed-Rank test[48] between the performance of CheXNet and our best performing configuration which is *AutoThorax*-Net with $k = 1001$. Our results, shown in Table 4, suggest that AUC and Specificity are improved

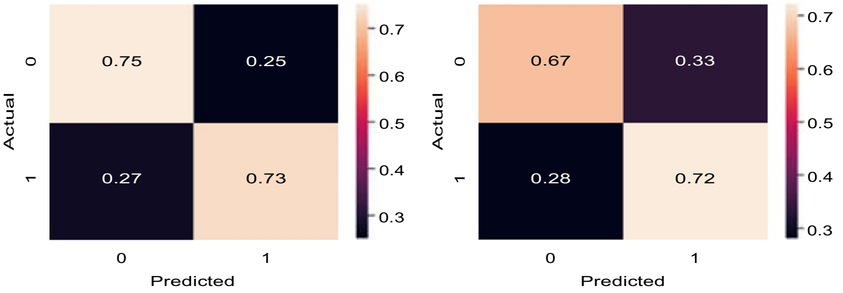


Figure 10. Dataset 2: Confusion matrices for the best results of *AutoThorax*-Net (left) and CheXNet (right).

from 77 to 82 and 67 to 75, respectively. The calculated p-values for these two metrics, both 0.005, are smaller than 0.05 and reject the null hypothesis which means significant difference exist between the performance of *AutoThorax*-Net with $k = 1001$ and CheXNet with respect to these two metrics.

Comparing Autoencoder against PCA. As one of the main contributions of the *AutoThorax*-Net is encoding the concatenated feature vector (i.e., reducing the dimensionality), the question arises whether the same level of performance can be achieved by traditional algorithms such as the principal component analysis (PCA). We did run the tenfold cross validation on both dataset configurations for $k = 11$ and $k = 51$. We observed in all settings that the performance of autoencoder was better than PCA. For instance, for the second experiment, PCA achieved 72% and 76% AUC for $k = 11$ and $k = 51$, respectively, while autoencoder achieved 74% and 80% AUC. As the performance of the dimensionality reduction is independent of k, one expects that a more capable compression should already manifest itself for any k. However, as we are using the compressed/encoded features for image search, good performance is expected to be particularly visible for a small number of matched cases.

Discussions

In our investigations, we experimented with image search as a classifier to detect pneumothorax based on autoencoded concatenated features applied on more than half a million chest X-ray images obtained through the merging three large public datasets.

In our experiments, we verified that the use of image search as a classifier with *AutoThorax*-Net as a feature extractor can improve classification performance. This was demonstrated by analysing the ROC curves to find the trade-off for each individual approach. We further confirmed that compressing concatenated deep features via autoencoders further improves the results of image search. This indicates that image search as a classifier is a viable and more conveniently explainable solution for the practice of diagnostic radiology when reports and history of evidently diagnosed cases of similar cases are readily available.

Please note that some of the folds we used may contain images that ChextNet has already seen during its training. This may bring a slight inflation of performance numbers for ChexNet. We ignored this unfair advantage for ChexNet over our AutoThorax-Net since we had to exploit the mixture of three public datasets and apply k-fold cross validation for maximum data usage and decreasing data bias.

Received: 11 May 2020; Accepted: 20 April 2021
Published online: 10 May 2021

References

1. Imran, J. B. & Eastman, A. L. Pneumothorax. *JAMA* **318**, 974–974 (2017).
2. Zarogoulidis, P. *et al.* Pneumothorax: from definition to diagnosis and treatment. *J. Thoracic Dis.* **6**, S372 (2014).
3. Larson, P. A., Berland, L. L., Griffith, B., Kahn, C. E. Jr. & Liebscher, L. A. Actionable findings and the role of it support: report of the acr actionable reporting work group. *J. Am. Coll. Radiol.* **11**, 552–558 (2014).
4. Gooßen, A. *et al.* Deep learning for pneumothorax detection and localization in chest radiographs. arXiv preprint arXiv:1907.07324 (2019).
5. Blausen.com. Medical gallery of blausen medical 2014. *WikiJournal Med* **1**, https://doi.org/10.15347/wjm/2014.010 (2014).
6. Guendel, S. *et al.* Learning to recognize abnormalities in chest x-rays with location-aware dense networks. In *Iberoamerican Congress on Pattern Recognition*, 757–765 (Springer, 2018).
7. Kelly, B. S., Rainford, L. A., Darcy, S. P., Kavanagh, E. C. & Toomey, R. J. The development of expertise in radiology: in chest radiograph interpretation, "expert" search pattern may predate "expert" levels of diagnostic accuracy for pneumothorax identification. *Radiology* **280**, 252–260 (2016).
8. Mao, C., Yao, L., Pan, Y., Luo, Y. & Zeng, Z. Deep generative classifiers for thoracic disease diagnosis with chest x-ray images. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 1209–1214 (IEEE, 2018).
9. Ker, J., Wang, L., Rao, J. & Lim, T. Deep learning applications in medical image analysis. *IEEE Access* **6**, 9375–9389 (2017).
10. Kamel, S. I., Levin, D. C., Parker, L. & Rao, V. M. Utilization trends in noncardiac thoracic imaging, 2002–2014. *J. Am. Coll. Radiol.* **14**, 337–342 (2017).
11. Taylor, A. G., Mielke, C. & Mongan, J. Automated detection of moderate and large pneumothorax on frontal chest x-rays using deep convolutional neural networks: a retrospective study. *PLoS Med.* **15**, e1002697 (2018).
12. Bengio, Y. *Learning Deep Architectures for AI* (Now Publishers Inc, 2009).

13. Wang, X. *et al.* Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2097–2106 (2017).
14. Rajpurkar, P. *et al.* Chexnet: radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225 (2017).
15. Rubin, J. *et al.* Large scale automated reading of frontal and lateral chest x-rays using dual convolutional neural networks. arXiv preprint arXiv:1804.07839 (2018).
16. Rajpurkar, P. *et al.* Deep learning for chest radiograph diagnosis: a retrospective comparison of the chexnext algorithm to practicing radiologists. *PLoS Med.* **15**, e1002686 (2018).
17. Dunnmon, J. A. *et al.* Assessment of convolutional neural networks for automated classification of chest radiographs. *Radiology* **290**, 537–544 (2018).
18. Feng, Y., Teh, H. S. & Cai, Y. Deep learning for chest radiology: a review. *Curr. Radiol. Reports* **7**, 24 (2019).
19. Irvin, J. *et al.* Chexpert: a large chest radiograph dataset with uncertainty labels and expert comparison. arXiv preprint arXiv:1901.07031 (2019).
20. Baltruschat, I. M., Nickisch, H., Grass, M., Knopp, T. & Saalbach, A. Comparison of deep learning approaches for multi-label chest x-ray classification. *Sci. Rep.* **9**, 6381 (2019).
21. Rakshit, S., Saha, I., Wlasnowolski, M., Maulik, U. & Plewczynski, D. Deep learning for detection and localization of thoracic diseases using chest x-ray imagery. In *International Conference on Artificial Intelligence and Soft Computing*, 271–282 (Springer, 2019).
22. Sze-To, A. & Wang, Z. tchexnet: Detecting pneumothorax on chest x-ray images using deep transfer learning. In *International Conference on Image Analysis and Recognition*, 325–332 (Springer, 2019).
23. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
24. Johnson, A. E. *et al.* Mimic-cxr: a large publicly available database of labeled chest radiographs. arXiv preprint arXiv:1901.07042 (2019).
25. Zhang, X., Liu, W., Dundar, M., Badve, S. & Zhang, S. Towards large-scale histopathological image analysis: hashing-based image retrieval. *IEEE Trans. Med. Imaging* **34**, 496–506 (2014).
26. Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700–4708 (2017).
27. Guan, Q. *et al.* Diagnose like a radiologist: attention guided convolutional neural network for thorax disease classification. arXiv preprint arXiv:1801.09927 (2018).
28. Yao, L., Prosky, J., Poblenz, E., Covington, B. & Lyman, K. Weakly supervised medical diagnosis and localization from multiple resolutions. arXiv preprint arXiv:1803.07703 (2018).
29. Yan, C., Yao, J., Li, R., Xu, Z. & Huang, J. Weakly supervised deep learning for thoracic disease classification and localization on chest x-rays. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, 103–110 (ACM, 2018).
30. Zhou, W., Li, H. & Tian, Q. Recent advance in content-based image retrieval: a literature survey. arXiv preprint arXiv:1706.06064 (2017).
31. Das, P. & Neelima, A. An overview of approaches for content-based medical image retrieval. *Int. J. Multimed. Inf. Retrieval* **6**, 271–280 (2017).
32. Camlica, Z., Tizhoosh, H. R. & Khalvati, F. Medical image classification via svm using lbp features from saliency-based folded data. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, 128–132 (IEEE, 2015).
33. Wan, J. *et al.* Deep learning for content-based image retrieval: a comprehensive study. In *Proceedings of the 22nd ACM international Conference on Multimedia*, 157–166 (ACM, 2014).
34. Tzelepi, M. & Tefas, A. Deep convolutional learning for content based image retrieval. *Neurocomputing* **275**, 2467–2478 (2018).
35. Sklan, J. E., Plassard, A. J., Fabbri, D. & Landman, B. A. Toward content-based image retrieval with deep convolutional neural networks. In *Medical Imaging 2015: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 9417, 94172C (International Society for Optics and Photonics, 2015).
36. Qayyum, A., Anwar, S. M., Awais, M. & Majid, M. Medical image retrieval using deep convolutional neural network. *Neurocomputing* **266**, 8–20 (2017).
37. Qiu, C., Cai, Y., Gao, X. & Cui, Y. Medical image retrieval based on the deep convolution network and hash coding. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 1–6 (IEEE, 2017).
38. Chung, Y.-A. & Weng, W.-H. Learning deep representations of medical images using siamese cnns with application to content-based image retrieval. arXiv preprint arXiv:1711.08490 (2017).
39. Owais, M., Arsalan, M., Choi, J. & Park, K. R. Effective diagnosis and treatment through content-based medical image retrieval (cbmir) by using artificial intelligence. *J. Clin. Med.* **8**, 462 (2019).
40. Mercioni, M. A. & Holban, S. A survey of distance metrics in clustering data mining techniques. In *Proceedings of the 2019 3rd International Conference on Graphics and Signal Processing*, 44–47 (2019).
41. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
42. Goldberger, A. L. *et al.* Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation* **101**, e215–e220 (2000).
43. Abadi, M. *et al.* Tensorflow: a system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 265–283 (2016).
44. Zhang, Y., Wu, H., Liu, H., Tong, L. & Wang, M. D. Mitigating the effect of dataset bias on training deep models for chest x-rays. arXiv preprint arXiv:1910.06745 (2019).
45. Zech, J. R. *et al.* Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS Med.* **15**(11), e1002683 (2018).
46. Youden, W. J. Index for rating diagnostic tests. *Cancer* **3**, 32–35 (1950).
47. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
48. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006).
49. Bressem, K. K. *et al.* Comparing different deep learning architectures for classification of chest radiographs. *Sci. Rep.* **10**, 1–16 (2020).
50. Yang, X. *et al.* Covid-ct-dataset: a ct scan dataset about covid-19 (2020). 2003.13865.
51. Pham, T. D. A comprehensive study on classification of covid-19 on computed tomography with pretrained convolutional neural networks. *Sci. Rep.* **10**, 1–8 (2020).

Acknowledgements

This research was a “Path-Finder Project” financially supported by the Vector Institute, Toronto, Canada.

Author contributions

A.S. has implemented the approach, run all initial experiments, and written the first draft of the paper. A.R. has run the second round of experiments with ROC curves to calculate sensitivity and specificity as well PCA tests. H.R.T has proposed the idea, conceptually designed the solution, analyzed the results and edited the paper and rewritten several parts.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-89194-4.

Correspondence and requests for materials should be addressed to H.R.T.

Reprints and permissions information is available at www.nature.com/reprints.