Lung Cancer Screening Using Adaptive Memory-Augmented Recurrent Networks

Aryan Mobiny, Supratik Moulik, Ilker Gurcan, Tanay Shah, Hien Van Nguyen

***Abstract*—In this paper, we investigate the effectiveness of deep learning techniques for lung nodule classification in computed tomography scans. Using less than 10,000 training examples, our deep networks perform five times better than a standard radiology software. Visualization of the networks' neurons reveals semantically meaningful features that are consistent with the clinical knowledge and radiologists' perception. Our paper also proposes a novel framework for rapidly adapting deep networks to the radiologists' feedback, or change in the data due to the shift in sensor's resolution or patient population. The classification accuracy of our approach remains above 80% while popular deep networks' accuracy is around chance. Finally, we provide in-depth analysis of our framework by asking a radiologist to examine important networks' features and perform blind relabeling of networks' mistakes.**

***Index Terms*—Adaptation, Computed Tomography, Convolutional Neural Network, Memory-augmented Neural Network, One-shot Learning, Pulmonary Lung Nodule.**

I. Introduction

Lung cancer is consistently ranked as the leading cause of the cancer-related deaths all around the world in the past several years, accounting for more than one-quarter (26%) of all cancer deaths [1]. The stage at which it is diagnosed plays a critical role in the lung cancer prognosis. Based on the latest cancer statistics reports [1], the five-year relative survival rate is over 50% in early-stage disease, while that of advanced stage is less than 5%. Therefore, early diagnosis significantly increases the treatment effectiveness and the patient's survival chance. Of all the available imaging modalities, Computed Tomography (CT) of the lung detects more nodules and lung cancers, including early-stage cancers [2]. However, screening several scans from an overwhelming number of different cases on a daily basis puts an increased burden on radiologists which could make them prone to make irreparable mistakes. Many studies have documented the occurrence of radiological errors in clinical practice, caused by many different contributing factors which can generally be divided into person-specific (such as radiologists' complacency, lack of knowledge, faulty reasoning, etc) and environment-specific issues (e.g. inadequate equipment, staff shortages, excess workload, etc.) [3], [4].

That's where computer-aided diagnosis (CAD) system comes into play to help to improve the radiologist's performance in terms of diagnosis accuracy and speed [5]. Over the years, numerous studies have demonstrated the CAD systems potential to assist radiologists in detecting pulmonary lung nodules [6], [7], as well as distinguishing benign from malignant nodules [5], [8]. However, Automated identification of nodules from non-nodules is quite challenging mainly due to the large variations in sizes, shapes, and locations of the nodules. There are also different categories of nodules (such as solitary, pleural, ground-glass opacity, and cavitary) contributing to the diversified contextual environment around the nodule tissue [9]. Moreover, some of the non-nodule candidates look quite similar to the real nodules in the sense of morphological appearance which will strongly hinder the detection process [10].

Generally, a CAD system which is used for detecting pulmonary lung nodules composed of three major stages. The first one is to detect all potential nodules (contiguous structures within the lungs). In the second stage, the main features of the candidate nodules will be extracted. These features are later incorporated into a classifier to separate the FP objects (normal tissues) from TP (real nodules) [11] where increasing the correct detection rate is the major challenge. Here, an intensity-based automated system (introduced in section II) is incorporated to detect nodule candidates and we mainly focus on the second and third stages aiming to improve the classification part by leveraging the power of the state-of-the-art methods and deep models.

The performance of a conventional CAD system depends heavily on the intermediate image processing stages (such as extracting hand-crafted morphological and statistical features) which are both time-consuming and subjective [12]. In recent years, deep learning technology has attracted considerable interest in the computer vision and machine learning community. Deep neural networks (DNNs) have an advantage of automatically capturing the image higher level feature representation directly from the input pixel data. Therefore, deep learning framework can potentially replace the feature computing and selection which makes it suitable in medical image diagnosis [13].

In the context of pulmonary nodule classification in CT images, Hua et al. [14] introduced models of a deep belief network and a convolutional neural network that can outperform the conventional hand-crafted feature computing CAD frameworks. Setio et al. [15] proposed a multiview convolutional network for the purpose of lung nodule candidates detection at a very high sensitivity (85.4%) while reducing the number of false positives.

Knowing the enormous potential of CADs to enhance the radiologists' diagnostic capability, most of the previous studies focused on improving the stand-alone performance of CADs. However, optimizing the quality of the interaction between radiologists and CAD systems as a *team* is often overlooked [16]. Therefore, while the majority of the studies reported a high level of diagnostic performance using a combination of radiologists and CAD systems, the overall team performance is lower than expectations based on the performance of radiologist and the CAD system in isolation. There are even researchs reporting no or minimal benefits associated with the presence of CADs on radiologists diagnostic performance (e.g. [17], [18]).

In this paper, we propose a state-of-the-art "expert in the loop" framework using the idea of one-shot learning which enables the radiologist to modify the output of the CAD system (i.e. CNN classifier) by providing only a few feedbacks. The first advantage of this framework over its counterparts is that it allows radiologists to check the errors made by the network in a real-time manner and track its improvement happening by providing sequential feedbacks. This will eventually enable radiologists to calibrate their trust in the CAD system more effectively and stop the process as soon as they are convinced of the accuracy of the results. Second, there is no need to retrain the network from scratch (which is both time-consuming and inefficient) each time we receive new sets of data from relatively different distributions. The final goal of designing such framework is to be able to rapidly understand and adapt to the new CT scan images.

It's especially helpful for the case of lung nodule classification which involves lots of inter and intra-patient variations that heavily deteriorate the performance generalization. These variations might occur either by the different nodule shapes coming from subjects of various age, gender, race, etc. or by the different image resolution due to change in the scanning modality. In other words, instead of having a static bias, the proposed structure is designed to shift its bias and modify the decision boundary as it acquires more data.

To implement the "expert in the loop" paradigm, we take two different approaches. One is to use the so-called "Siamese network" which was first introduced by Bromely and LeCun [19] and exploited in a signature verification problem. Later, Koch et al. [20] used a modified version of the siamese network with a convolution structure in a one-shot learning setting and achieved promising results in the one-shot classification task using the Omniglot dataset [21].

The second strategy for attaining rapid inference from a few feedbacks is to use a memory-augmented neural network (MANN) proposed by the Google DeepMind team [22]. This structure contains a dedicated, addressable memory storage and is capable of meta-learning, meaning that it learns to bind data representations to their proper class labels by employing a general scheme.

The main contributions can be summarized as follows:

1) We study the discrimination power of deep networks on CT lung nodule classification. Our results show the promising performance of the designed CNNs compared with that of the automated software which works based on the hard-coded rules on the density values. The classification results also demonstrate that the 3D CNNs capable of receiving and analyzing the volumetric CT images are more accurate that the 2D counterparts. This is because the 3D network can exploit and encode the complex environment surrounding the nodules and extract more discriminative representations.
2) We model an interactive framework which enables the radiologist to modify the decision function by providing only a few feedbacks. This system quickly adapts to the new CT image data received from a never-before-seen distribution (which might happen due to changes in patients or scanning modality, etc.). Therefore, there is no need to retrain the network which saves a lot of time and effort.
3) We performed an in-depth analysis of the designed networks. This was done by presenting the network predictions to the radiologists and receiving their feedbacks continuously. Visualizing the networks also helped with understanding the most informative features captured by the network. This was done using both real and synthetic data generated based on the radiologist suggestion. This process finally provides an in-depth perception of the way that the deep convolutional network encodes and analyzes the input images. Finding similarities between the way that a radiologist analyzes a specific image and the network extract information from it helps building the experts trust in the machine.

II. Dataset

The study included 226 unique CT Chest studies (with or without contrast) all from a single institution utilizing GE and Siemens scanners. The data was anonymized via institutional data security protocols.

The DICOM image data was preprocessed utilizing full body automated segmentation software in order to identify structures to the organ level. The segmented lung are subsequently analyzed by the same software to isolate potential nodules within the lung volume. A bounding box ROI with at least 8 voxel padding surrounding the candidate nodule generated and the data is automatically cropped utilizing an $n \times n \times n$ voxel volume where $n = 32 + 16 \times m$ (m is number of addition buffer voxels needed to fully bound the nodule). Therefore, the resulted 3D images are of size $32 \times 32 \times 32$, or $48 \times 48 \times 48$, or $64 \times 64 \times 64$ pixels and so on.

The candidate points are reviewed by at least one board certified radiologist; the points are assigned a label of nodules or non-nodules. From all the generated images (about 7400 images), around 56% were labeled as nodules and the rest were detected to be non-nodules.

Examples of ROI images extracted from the thoracic CT scans, reviewed and labeled by the radiologist are provided in Fig. 1. These images illustrate the demanding task of distinguishing nodules from non-nodule lesions. One reason is that the pulmonary nodules come with large variations in shapes, sizes, types, etc. In Fig. 1, examples of solitary (1), sub-pleural (2), cavitary (3) and ground-glass (4) nodules are depicted. (5) is a more complicated sample containing a mixed solid and ground-glass nodule with irregular margins. While nodules are commonly known as spherical lesions, having a non-spherical shape (12-13) and irregular margins is pretty common. These irregularities can be caused by vessels and/or spiculations (6-11). Other objects and tissues might also appear in the nodule samples, such as single or multiple blood vessels

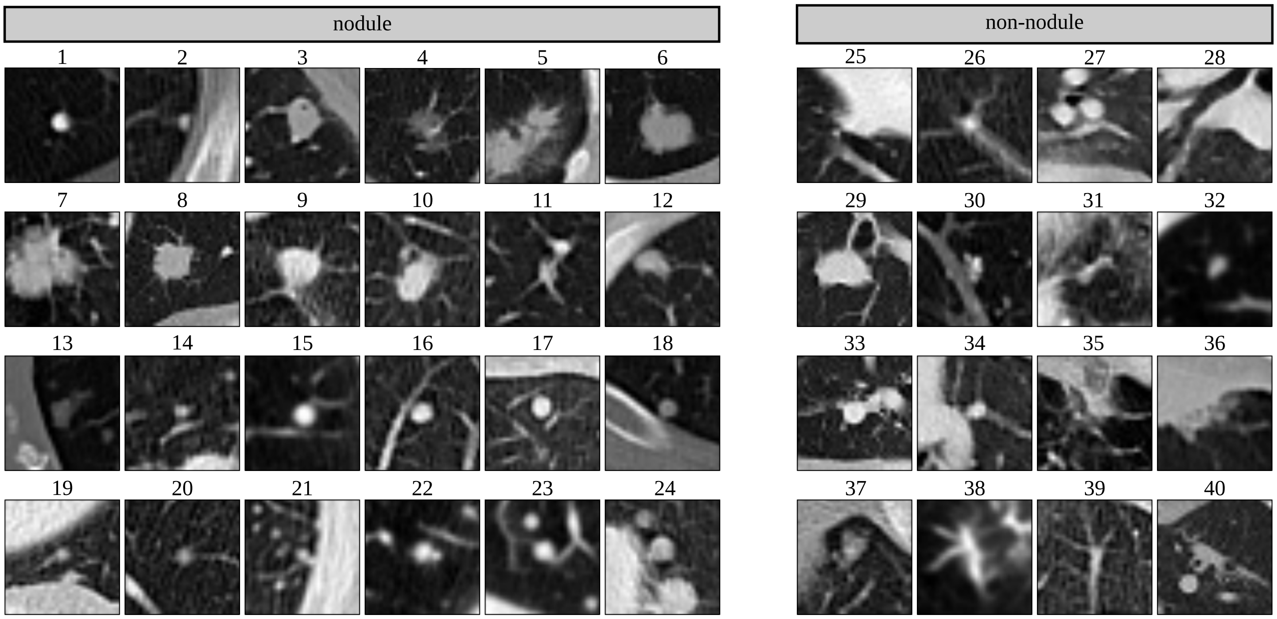

Fig. 1. Sample 32 × 32 ROI images of nodules (Left) and non-nodules (Right) with various sizes, shapes and type. Lesions are located in the center of the box. Each image is a representative 2D axial plane extracted from the middle of the volume.

(14-17), chest wall (18-19), lung recess (19), etc. Moreover, one nodule image can also contain several nodules of different shapes and sizes (21-24) which makes the detection more challenging.

Some non-nodule examples are simple to distinguish, such as images only containing blood vessel (25-26), multiple vessels (27), vessel and airway(28), lymph node (29). However, the second reason which hinders the identification process is the non-nodule candidates mimicking the morphological appearance of the real pulmonary nodules. Examples are calcification (30), short vessels (31-34), scarring (35), infection (36-37), vessels with motion artifact mimicking a ground-glass nodule (38), septical thickening (39). Some images might also contain a nodule, but is centered on another tissue (such as a vessel in (40)) and so is labeled as non-nodule.

The automated image segmentation software has an algorithmic method for separating true from false candidate nodule points. By utilizing the lung segmentation data and the voxel density value, the center of each candidate point is estimated as the point which is the most equidistant from surrounding air density (-200 HU) lung points. Radial density analysis is subsequently performed to determine the symmetry characteristic of the candidate point. Specifically, the nodule is categorized as either mostly spherical or mostly cylindrical. This geometric analysis forms the backbone of the automated nodule analysis. There are several key limitations to the method which necessitated the additaion of machine learning algorithms. First, for practical computation reasons, only limited number of radial direction are analyzed which means that a vessel had to be oriented along the path of the angular steps in order to be determined as such. The symmetry analysis also requires the establishment of an edge threshold which becomes problematic is the setting of inhomogeneous or ground glass nodules. In addition, if a nodule is situation along a blood vessel, the nodule is likely to be mischaracterized as a vessel; given that most metastatic disease to the lungs is spread hematogenously, this is the most clinically significant limitation. Finally, the variations in morphology, homogeneity and overall size contribute to the limitation of the algorithm. Using the radiologist prediction as the ground truth, the software prediction performance is as depicted in Table I. The software could not make any reliable prediction for 117 images (classified as unknown).

TABLE I
AUTOMATED-SOFTWARE PREDICTION RESULTS FOR 7399 IMAGES FROM 226 CT SCANS. THE SOFTWARE WASN'T ABLE TO IDENTIFY 117 IMAGES AND MARKED THEM AS UNKNOWN

		Actual Class	
		nodule	**non-nodule**
Predicted	**nodule**	3554	878
Class	**non-nodule**	617	2233

III. Lung Nodule Classification with Deep CNNs

The traditional models used for supervised learning (here, separating nodule and non-nodule images) usually starts with designing and extracting hand-crafted features (such as texture, geometry, etc.) from input images. A trainable classifier (e.g. linear discriminant analysis, support vector machine, etc.) then categorizes the resulting feature vectors into classes [23].

In the recent years, however, deep artificial neural networks have gained more attention and won numerous contests in machine learning and pattern recognition. After enjoying little success since the 1980s, neural networks are widely successful today due to the large amount of available data and computational resources of modern computers (equipped with faster CPUs and general purpose GPUs) to run much larger models [24]. In contrast to hand-crafted features extracted in traditional methods, deep learning models are capable of automatically extracting features specific to the problem in hand. They made this possible by learning a representation of data via training end-to-end in a supervised fashion.

Inspired by biological processes, convolutional neural network (CNN) is a class of deep neural networks that have successfully been applied in several computer vision application [25], [26] and more specifically, in the area of medical image analysis [27], [28]. In this paper, similar to recent studies reporting promising results in lung nodule classification [15], [10], we leveraged the discrimination power of two powerful convolutional network architectures, known as AlexNet [25] and ResNet [29]. These networks are modified to match with the size of images in hand, as well as improving their performance using the latest advances in deep learning research.

Data preparation and preprocessing: start with the binary classification of real and false nodules (labeled by the radiologist), the entire dataset was fully anonymized. Then all images were resized to $32 \times 32 \times 32$ pixels and separated to training and validation set at patient-level: 158 patients (out of 226) were selected for training (which includes about 70% of the total number of images) and the remaining 68 patients were chosen for validation. To handle the imbalanced data for the classification task, we managed to balance the training set by adding copies of the images of the under-represented class (false nodule class). This gives us the final set of 3D images of the potential lung nodules to be classified. We also cut all these images along x-axis and saved the middle-slices as 2D images. Having both 2D and 3D images enable us to build and train 2D and 3D networks and compare their performance to check how much improvement can be gained by adding the third dimension. Finally, data was scaled by bringing all the pixel intensity values into the range [0, 1] (i.e. unity-based normalization). Since all the intensity values are in the range [0, 4096], it was simply done by dividing the pixel values by 4096.

AlexNet: The famous AlexNet model [25] is selected as a simple yet effective model for the classification task. Some modifications are also applied to the original AlexNet to make it compatible to our data dimension and improving its performance as well. As depicted in Table II, the network contains eight layers (five convolutional layers followed by three fully-connected layers) which is similar to the original AlexNet. These networks are prepared similarly for both 2D and 3D images where in the 3D structure, 2D computations (such as convolutions, pooling, etc.) replaced by their 3D counterparts. Moreover, in both 2D and 3D-Alexnet, the local response normalization layers after the first, second, and last convolutional layers proposed in the original paper are removed and replaced by adding the powerful batch normalization (BN) layers [30] after all layers (both convolution and fully-connected) and before the ReLU nonlinearity. BN allows us to use higher learning rates and converge much faster. It also has regularization effect (like dropout) because of computing the statistics on every mini-batch (rather than the entire training examples) [30]. The ReLU non-linearity is applied to the batch-normalized output of every convolutional and fully-connected layer, except the last fully-connected layer which is followed by a softmax function. Smaller stride (1 instead of 4) and filter size (4 instead of 11) are selected for the first convolutional layer which was demonstrated to results in more distinctive features and fewer dead filters [31]. We also use dropout with the rate set to 50% [25] in the first two fully-connected layers to prevent over-fitting at the expense of later convergence.

TABLE II
AlexNet Architecture

Layer Name	Output Size	Configuration
Conv1	$32 \times 32 \times 32 \times 16$	4×4, stride 1
MaxPool1	$16 \times 16 \times 16 \times 16$	3×3, stride 2
Conv2	$16 \times 16 \times 16 \times 32$	3×3, stride 1
MaxPool2	$8 \times 8 \times 8 \times 32$	3×3, stride 2
Conv3	$8 \times 8 \times 8 \times 32$	3×3, stride 1
Conv4	$8 \times 8 \times 8 \times 32$	4×4, stride 1
Conv5	$8 \times 8 \times 8 \times 32$	4×4, stride 1
MaxPool3	$4 \times 4 \times 4 \times 32$	3×3, stride 2
FC1, FC2, FC3	$h_1 = 200, h_2 = 75, h_3 = 2$	

ResNet: We also test whether our data can benefit from a more complex state-of-the-art CNN model, i.e. ResNet [29]. We use the 50-layer ResNet with some added modifications (see Table III and Fig. 2 for a more detailed representation) which have almost the same time complexity as the AlexNet, and consists of four so-called bottleneck blocks as described in [29]. Each of the three convolutional layers in a single bottleneck block follows by BN which is applied before the ReLU nonlinearity. A fully-connected layer with 50 hidden units is added before the classification layer which showed being capable of improving the classification results. Similar to AlexNet, a dropout with 50% rate is used in this fully-connected layer.

Loss Function: Let M represent the mini-batch size and $y^{(i)} = (y_1^{(i)}, y_2^{(i)})$ be the one-hot-encoded correct label for the i^{th} image in the mini-batch. We impose a cross-entropy cost function on our binary classifier of the following form:

$$L = \frac{-1}{M} \sum_{i=1}^{M} \sum_{j=1}^{2} y_j^{(i)} \ln p_j^{(i)} \qquad (1)$$

where j indexes the j^{th} output neuron and $p_j^{(i)}$ is the predicted value of the j^{th} neuron in the output layer in response to the i^{th} image in the mini-batch, passed through a softmax function.

Weight initialization: In both networks, weights are initialized by random values drawn from Gaussian distribution with zero mean and 0.01 standard deviation and biases are all initialized by zero.

Learning schedule: We use ADAM optimizer [32] which is a method for efficient stochastic optimization that only requires first-order gradients with little memory requirement. The corresponding parameters are set to $\beta_1 = 0.9$, $\beta_2 = 0.999$,

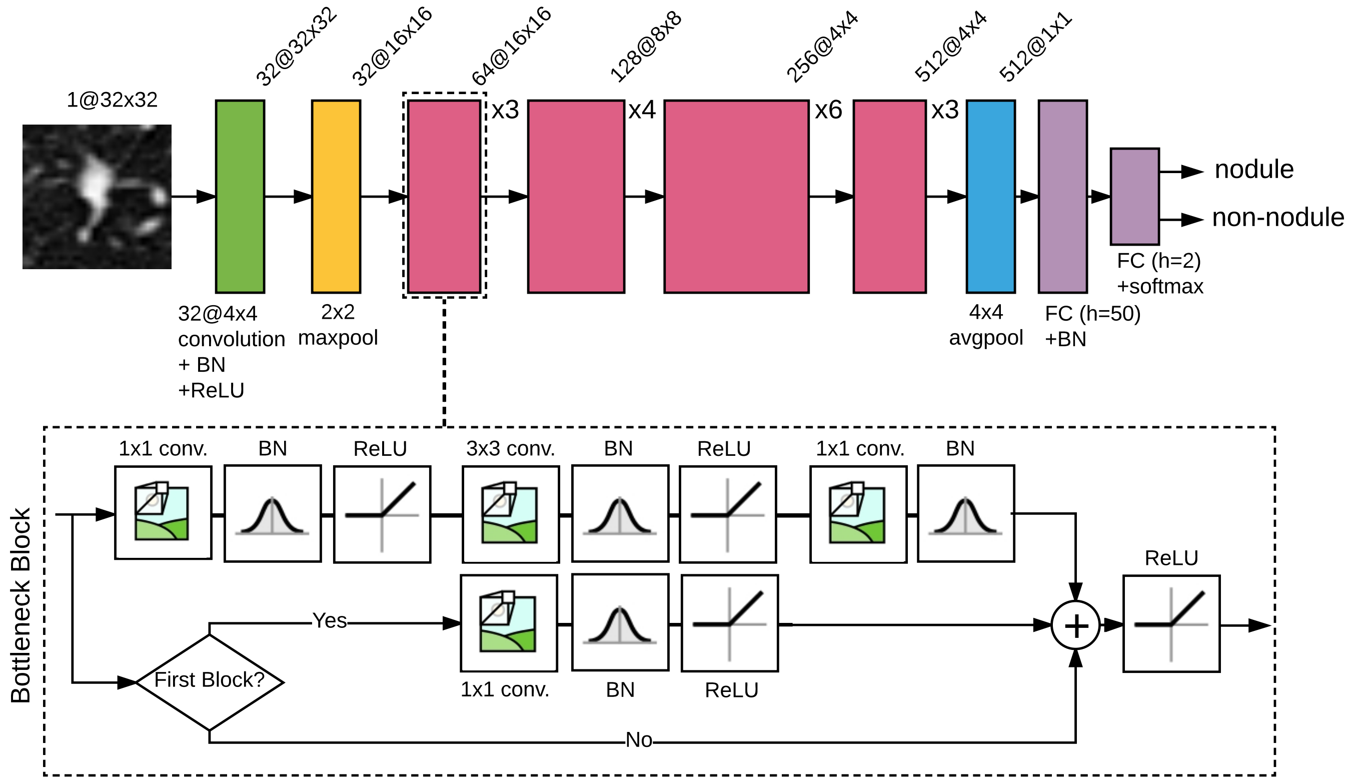

Fig. 2. A configuration of the ResNet model with 50 convolutional layers. This network includes 16 bottleneck blocks. The magnified inset shows how an individual bottleneck block is constructed.

$\epsilon = 10^{-8}$. For both networks, the initial learning rate and mini-batch size are set to 0.001 and 16, respectively.

Data augmentation: is applied to artificially increase the number of training data and reduce over-fitting on models. Real-time data augmentation is done by randomly rotating around the center for 2D images and along all three axes for 3D images. We set the maximum rotation degree to 90^o and 45^o degrees for 2D and 3D networks respectively so as to prevent introducing too much distortion to the images. Afterwards, pepper noise was added to the rotated images. It's similar to applying dropout to input neurons in the visible layer. The dropout rate is set to 5%.

IV. Visualizing the Receptive Fields of the CNN Units

Current deep neural networks result in astonishing performances in a wide variety of detection and classification tasks (such as [25] and [29] on the ImageNet benchmark). These networks also remove the need to carefully design and select morphological and statistical features. However, there is no clear understanding of their remarkable performance. In other words, the nature of the representations learned by the networks is unclear.

Recently, a number of studies introduced novel visualization techniques to realize the representation learned by CNNs. These methods can shed lights on characteristics of CNN models that goes beyond simply measuring their discriminative power. In [31], a new deconvolutional structure was introduced to visualize what activates each single unit. [33] introduced a method to visualize and understand the representations learned by the intermediate layers. They showed that in a scene classification task, the trained CNN is capable of detecting objects automatically, while no supervision is provided for the objects.

TABLE III
ResNet Architecture with 50 Layers

Layer Name	Output Size	Configuration
Conv1	$32 \times 32 \times 32 \times 32$	4×4, stride 1
MaxPool1	$16 \times 16 \times 16 \times 32$	2×2, stride 2
Bottleneck-Block1	$16 \times 16 \times 16 \times 64$	$\begin{bmatrix} \text{1x1, 32} \\ \text{3x3, 32} \\ \text{1x1, 64} \end{bmatrix} \times 3$
Bottleneck-Block2	$8 \times 8 \times 8 \times 128$	$\begin{bmatrix} \text{1x1, 48} \\ \text{3x3, 48} \\ \text{1x1, 128} \end{bmatrix} \times 4$
Bottleneck-Block3	$4 \times 4 \times 4 \times 256$	$\begin{bmatrix} \text{1x1, 64} \\ \text{3x3, 64} \\ \text{1x1, 256} \end{bmatrix} \times 6$
Bottleneck-Block4	$4 \times 4 \times 4 \times 512$	$\begin{bmatrix} \text{1x1, 128} \\ \text{3x3, 128} \\ \text{1x1, 512} \end{bmatrix} \times 3$
AvgPool	$1 \times 1 \times 1 \times 512$	4×4, stride 1
FC1, FC2	$h_1 = 50, h_2 = 2$	

Similarly in our classification task, when training a CNN to differentiate nodules from non-nodules, it's interesting to see what are the most important components or regions within an image to make predictions. This helps to evaluate how efficient the underlying representation is. Units specialized to detect and react to certain properties or objects within images are mimicking the radiologist's decision-making procedure. Proofing this helps to build the radiologist's trust in the CAD system.

To this end, we exploited the method introduced in [33] to investigate the receptive fields (RF) of different units in the CNN. They proposed a method to estimate the empirical size of the RFs of the units by identifying the regions of the image that leads to the high unit activations. For each image, this region can be called the *activation region* which plays the most important role in generating the high activation. Focusing precisely on the activation regions of each image allows us to better understand each unit, as well as the whole network.

To estimate a given unit's RF, we used the images in the validation set as input. Then, 30 images with the highest activations for the given unit was used to compute its RF.

V. Interactive Adaptation Learning Framework

A. Motivation

DNNs have recently been achieving promising results on a variety of pattern recognition tasks (such as biomedical imaging and classification). However, there still exists concerns about the generalization of these networks. Changes in the dataset distribution, commonly known as domain shift, deteriorates the performance of deep networks. In 2010, Saenko et al. [34] showed that training an object classifier on a set of data acquired in a given setting (i.e. the source domain) and testing it in a different setting (i.e. target domain), degrades the classifier performance.

Recently, a few studies even cast doubt on the generality of DNNs in a less severe cases. In 2013 Szegedy et al. [35] illustrated that adding an imperceptible amount of perturbation can cause a DNN to misclassify an image. Later in 2015, [36] showed that unrecognizable images (generated by evolving the original images successively) easily mislead the state-of-the-art DNNs to recognize the *fooling images* as familiar ones.

In pulmonary lung nodule detection, domain shift is inevitable. Differences in the CT scanners' technical capabilities, such as slice count and reconstruction algorithm, cause changes in CT images and their resolution. Changes in subject population might also shift the data distribution. Moreover, the candidate nodule ROIs (both nodule and non-nodule cases) involve a lot of inter-class variations (see Fig. 1). In the case of real nodule samples, the nodule morphology and the surrounding environment changes drastically from image to image. Therefore, the designed framework must be able to quickly understand and adapt to the images from never-before-seen distributions.

When it comes to interpret medical images, radiologist's *trust* in the CAD system is a key factor in optimizing the interaction between radiologists and automated aids. Inappropriate level of trust in the automation leads to poor performance of the radiologist-CAD team [16]. Radiologists sometimes *under-trust* CADs preventing them from using its potential benefits. On the other hand, over-trust (i.e. too much trust) in automation leads to making diagnostic errors that would not have happened without CAD [18], [37].

In pulmonary lung detection, DNNs make mistakes, some of which are obvious to the radiologist. Having no way to interact with the system and make modifications often causes the radiologist's under-reliance on automation. To improve the image classification task, especially in specialized medical domains such as radiology, we need to incorporate the experts' domain knowledge in an interactive manner.

This can be done by letting the radiologist to freely select a small subset of images classified by the DNN classifier and use his/ her understanding of the image content to relabel it as and when needed. The learning algorithm must be able to rapidly incorporate the modified information (i.e. feedback provided by the radiologist), update its decision function and make more accurate predictions.

Here, we propose a framework which is able to address both concerns under the umbrella of *one-shot learning* setting. That is because of the nature of the problem in hand which requires making rapid inferences from small quantities of data. The proposed structure makes the CAD system adaptable to data coming from various distributions. Therefore, there will be no need to retrain the network from scratch every time receiving a new set of data. This helps spending less time and effort while still making accurate predictions. Whenever receiving a new set of data, the radiologist can review a few errors made by the deep network classifier and incorporate his knowledge to correct them. These corrections will be sent back to the CAD system as feed-backs to update the decision function.

B. One-shot Learning

It has been demonstrated that humans are capable of learning object categories from only one or very few examples, and at a rapid pace [38]. That's because humans make use of existing knowledge of previously learned classes when learning new ones. The best deep network systems, however, need hundreds or thousands of examples to learn new concepts [39]. Therefore, generating new behavior based on a few scraps of information is an ongoing challenge in the contemporary deep learning literature. That's the key motivation for the one-shot learning setting to make the systems use prior knowledge about object categories to classify new objects [40], [41].

More precisely, the one-shot learning approach aims to model the conditional probability distribution of $P(\hat{y}|\hat{x}, S)$ where $S = \{(x_i, y_i)\}_{i=1}^{m}$ is the small set of m examples of input-label-pairs, $\hat{x}$ is the given test example, and $\hat{y}$ a distribution over possible classes. In the deep learning framework, P is parametrized by a neural network trained to perform the mapping.

We examined two different strategies to update the decision function (given the corrections from radiologist) and make modified predictions. These strategies and the associated networks are explained in the following sections.

1) Deep Siamese Neural Network: By definition, Siamese neural network is a class of neural network architectures that contain two (or more) identical sub-networks with shared parameters and accept two (or more) inputs, map each of them to a high-level feature representation on each side, and finally use a metric between the resulted feature vectors to quantify their similarity. This architecture was first introduced in the early 90s by Bromley and LeCun [19] to solve a signature verification problem. In 2005, LeCun et al. [42] proposed a contrastive energy function to learn the similarity metric. Koch et al. later replaced the fully-connected layers with convolutional layers and used the weighted L_1 distance between the feature vectors (followed by a sigmoid activation) as the fixed similarity measure.

In our setting, in order to use the expert's knowledge in an interactive way, we let the radiologist to correct a few errors made by the classifier. In a real-time setting, the radiologist uses his domain knowledge to make these corrections on the unlabeled data. From the network's perspective, it's similar to adding new information which needs to be incorporated to modify the decision function. The whole process is done by picking a few images one by one, modifying their predicted labels and tracking how the prediction improvement is happening. The expert can stop the review and relabeling process whenever satisfied by the network performance.

Mathematically speaking, $P(y|\mathbf{x})$ is the conditional probability computed by the trained CNN network. Given an image with modified label (i.e. an update) of $(\mathbf{x}_i, y_i)$, the goal is to compute $P(y|\mathbf{x}, \mathbf{x}_i, y_i)$ which can be decomposed as:

$$\begin{aligned} P(y|\mathbf{x}, \mathbf{x}_i, y_i) &= \frac{P(\mathbf{x}, \mathbf{x}_i, y, y_i)}{P(\mathbf{x}, \mathbf{x}_i, y_i)} \\ &= \frac{P(y_i|\mathbf{x}, \mathbf{x}_i, y)P(\mathbf{x}, \mathbf{x}_i, y)}{P(\mathbf{x}, \mathbf{x}_i, y_i)} \\ &= \frac{P(y_i|\mathbf{x}, \mathbf{x}_i, y)P(\mathbf{x}_i|\mathbf{x}, y)P(y|\mathbf{x})P(\mathbf{x})}{P(\mathbf{x}, \mathbf{x}_i, y_i)} \\ &= \frac{P(y_i|\mathbf{x}, \mathbf{x}_i, y)P(\mathbf{x}_i|\mathbf{x}, y)P(y|\mathbf{x})P(\mathbf{x})}{P(\mathbf{x}, \mathbf{x}_i, y_i)} \\ &= \frac{P(y_i|\mathbf{x}, \mathbf{x}_i, y)P(\mathbf{x}_i|\mathbf{x}, y)P(y|\mathbf{x})}{P(\mathbf{x}_i, y_i|\mathbf{x})} \end{aligned} \tag{2}$$

Assuming independence, we can take $P(\mathbf{x}_i|\mathbf{x}, y) = P(\mathbf{x}_i)$ and $P(\mathbf{x}_i, y_i|\mathbf{x}) = P(\mathbf{x}_i, y_i)$. Therefore, the only term to focus on is $P(y_i|\mathbf{x}, \mathbf{x}_i, y)$. To compute this term, a Siamese network is trained in a supervised manner using the labeled data in hand. The key point which makes this architecture useful for computing $P(y_i|\mathbf{x}, \mathbf{x}_i, y)$ is that it accepts two input at a time and predicts as if they are similar or not; i.e. have the same label or not. Therefore, after the training, we can pass in the update data, $\mathbf{x}_i$, from one side and the rest from the other side and make inference and get $P(y_i|\mathbf{x}, \mathbf{x}_i, y)$. Based on (2), by multiplying this probability values by the CNN prediction probabilities, $P(y|\mathbf{x})$, we can get the modified prediction values.

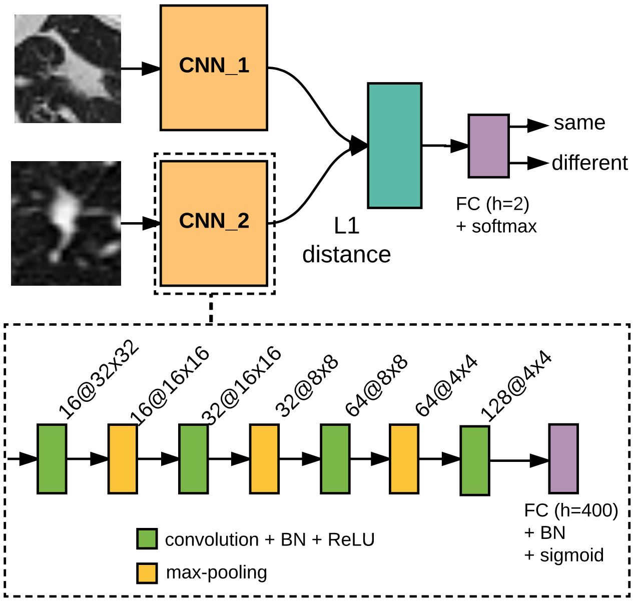

Fig. 3. Best Siamese network structure with convolutional architecture. CNN1 has the same structure as CNN2, with shared weight matrices at each layer.

The structure of the network with the highest validation accuracy is depicted in Fig. 3. As mentioned, the network accepts two inputs (say $\mathbf{x}_1$ and $\mathbf{x}_2$) at a time. The inputs are passed to the twin sub-networks to get their high-level feature representations. Each sub-network consists of a sequence of convolutional layers with filters of size 3 (except the first convolutional layer with filter size of 4) and a fixed stride of 1. Similar to the previously trained CNNs, the number of convolutional filters in each layer is selected as multiples of 8 to optimize the performance. Batch normalization is applied to the output feature maps of each convolution layer, before applying a ReLU activation function. Then it is passed through a max-pooling layer with a filter size and stride of 2. The last convolutional layer, however, is followed by no max-pooling.

The outputs of the last conolutional layer is flattened into a single vector that is passed to a fully-connected layer with 400 hidden units, followed by batch normalization and sigmoidal activity function. This forms the output vectors $\mathbf{h_1}, \mathbf{h_2} \in \mathbb{R}^{400}$ for each sub-network in response to feeding a single input from each side. Then the L_1 distance metric was computed between these two vectors, which is finally given to the classifier layer (fully-connected layer with 2 unit followed by a softmax). More precisely, the prediction vector is given by:

$$\mathbf{p} = \text{softmax}(\mathbf{W}^T.|\mathbf{h_1} - \mathbf{h_2}| + \mathbf{b}) \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{400\times2}$ and $\mathbf{W} \in \mathbb{R}^{2\times1}$ are the weights and biases linked to the last layer. Entries of $\mathbf{W}$ matrix can be interpreted as the parameters weighting the importance of the component-wise distance [20].

Training schedule: The Siamese network is trained using all possible combinations of size two of the training images in hand. Importantly, data was separated at subject level meaning that we made sure that both images fed to the network at a particular time-step are from the same scan (or patient). Therefore, the network is trained in two level, and learns not only to distinguish the data similarity/ dissimilarity within a scan, but also to capture the inter-scan variations and generalize well.

One-hot encoded true labels are selected as $[1, 0]$ and $[0, 1]$ for different and similar images, respectively. Since we applied the Softmax function in the output (which maps the data onto the interval [0, 1]), standard cross-entropy cost function and back-propagating the error is the suitable choice for training the network (following the approach in [20] and [43]).

We utilized the same learning procedure as the one used for CNNs (ADAM optimizer with initial learning rate of 0.001), batch size of 16, and the same data preprocessing and augmentation. The weights and biases of convolutional and fully-connected layers are initialized as suggested by [20].

2) Recurrent Neural Network with Memory Capacity: Due to the sequential nature of feed-backs provided by the radiologist, and the need to encode and accumulate the information during time, recurrent neural network (RNN) seems to be a proper choice. They have "memory" that captures information about what has been calculated so far. Even though RNNs are specialized for processing sequential values, it's still possible to use their power to process fixed-vector input/outputs in a sequential manner [24].

Theoretical and empirical evidences showed that training vanilla RNNs to learn to store information for a long time is problematic [44]. That is because the gradients propagated over many stages tend to either vanish or explode [45]. Long-short term memory (LSTM) model was introduced by Hochreiter and Schmidhuber [46] to avoid the long-term dependency problem.

In our lung nodule detection problem, the model must be able to learn at two level. One is to learn to quickly make accurate inferences within a small dataset coming from a certain distribution. The second level is to learn capturing the cumulative expertise gained *across* tasks and continuously adapt to new needs [47]; i.e, data from never-before-seen distribution in our case. In other words, the model must be capable of "learning to learn" or "meta-learning".

Generally, the goal of meta-learning is to train a model on a variety of learning tasks, such that it can solve new tasks using only a small number of training samples. While it has been showed that the neural networks with internal memory capacity (such as LSTM) are capable of meta-learning, they are not able to rapidly encode, store and access a significant amount of new information required by each new task. In 2016, Santoro et al. [22] proposed an external-memory-equipped network called memory-augmented neural network (MANN) proved to be capable of meta-learning in tasks with considerable amount of short and long term memory requirements. It was demonstrated by the MANN's superior performance to a LSTM in successfully classifying never-before-seen Omniglot classes after only a few presentations.

MANN uses an external memory which is usually denoted by a matrix $\mathbf{M}_t \in \mathbb{R}^{k \times q}$ where k is the number of memory slots and q is the size of each slot. The model has an LSTM controller that interacts with the external memory with read and write heads at every time step. The reading operation achieved by calculating the following inner product:

$$\mathbf{r}_t = (\mathbf{M}_t)^T . \mathbf{w}_t^r \tag{4}$$

Here, $\mathbf{r}_t$ is the content vector and $\mathbf{w}_t^r \in \mathbb{R}^{k \times 1}$ is the one-hot vector of read weights which leads to retrieving one of the memory slots. To write into the memory, Santoro et al. [22] designed a module called Least Recently Used Access (LRUA) to access the least used memory locations by computing the *least-used* weights vector, $\mathbf{w}_t^{lu}$, at each time step. The write weights $\mathbf{w}_t^w \in \mathbb{R}^{1 \times k}$ which is also a one-hot vector is then computed as:

$$\mathbf{w}_t^w \leftarrow \sigma(\alpha)\mathbf{w}_{t-1}^r + (1 - \sigma(\alpha))\mathbf{w}_{t-1}^{lu} \tag{5}$$

where $\sigma(.)$ is the sigmoid function. Let i be the index of the non-zero element in the one-hot vector $\mathbf{w}_t^w$, then the controller writes in the memory as:

$$\mathbf{M}_t(i) \leftarrow \mathbf{M}_{t-1}(i) + w_t^w(i)\mathbf{a}_t \tag{6}$$

where $\mathbf{a}_t$ is the linear projection of the current hidden state passed through a *tanh* nonlinearity.

The MANN and task structure is presented in Fig. 4. In this setup, a task, or episode, involves presenting labeled images sequentially where y_t is both the target for image $\mathbf{x}_t$ and is presented as input along with $\mathbf{x}_t$ with one step shift in time. For instance, at time t, y_{t-1} is provided as input along with $\mathbf{x}_t$. More importantly, correct labels are randomly shuffled from episode to episode. For example, nodules are labeled as 1 in some episode (so that non-nodule class is labeled as 0) while their labels are randomly changed to 0 in other episodes. This strategy will finally prevent the MANN from learning to simply mapping the samples to their fixed class labels [22].

In our setting, MANN is trained independent of the previously trained convolutional networks, but they are later merge together to form an adaptive system capable of generalizing to new set of nodule images. Suppose $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ is the set of current labeled images in hand which is used for training both CNN and MANN. In the real-time closed-loop setting, $U = \{\mathbf{x}_j, y_j\}_{j=1}^M$ is either the data reviewed and relabeled by the radiologist. These two sets will be used to modify the decision function and make accurate predictions for the newly received data. This can be formulated as follows:

$$\begin{aligned} P(y|\mathbf{x}, D, U) &= \frac{P(y, \mathbf{x}, D, U)}{P(\mathbf{x}, D, U)} \\ &= \frac{P(U|y, \mathbf{x}, D)P(y|\mathbf{x}, D)}{P(U|\mathbf{x}, D)} \\ &= \frac{P(U|y, \mathbf{x}, D)P(y|\mathbf{x}, D)}{\sum_y P(U|\mathbf{x}, y, D)P(y)} \\ &= \frac{P(y|\mathbf{x}, D)}{P(y) + \frac{P(U|\mathbf{x}, 1-y, D)}{P(U|\mathbf{x}, y, D)}P(1-y)} \\ &= \frac{P(y|\mathbf{x}, D)}{P(y) + r(y)(1 - P(y))} \end{aligned} \tag{7}$$

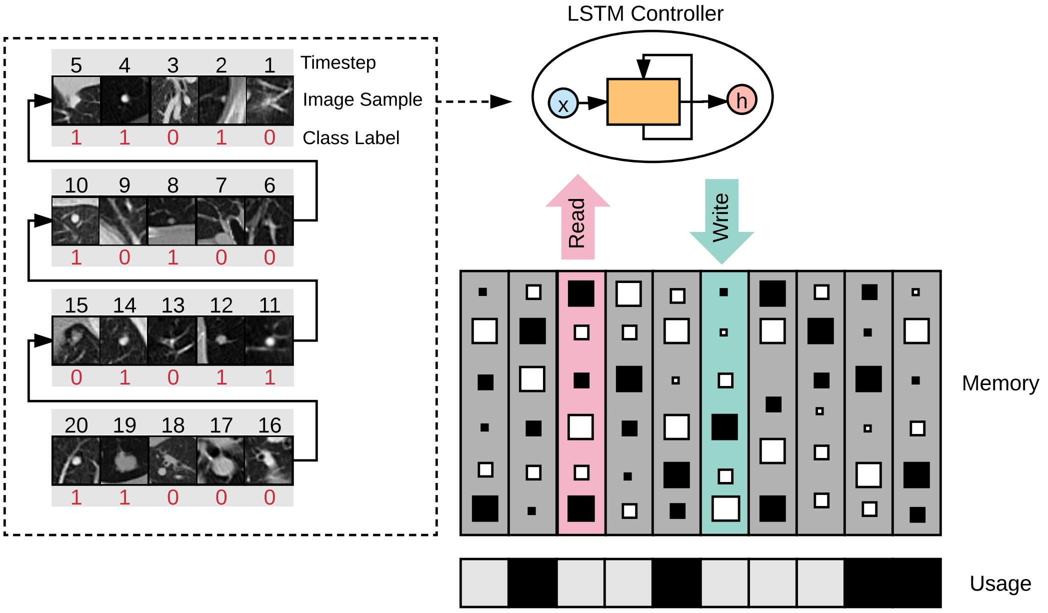

Fig. 4. Example string label and input sequence (left) and memory-augmented neural network (MANN) architecture

Here, $P(y|\mathbf{x}, D)$ is the prediction made by the CNN classifier while $P(y|\mathbf{x}, D, U)$ is the modified decision made by merging the MANN and CNN's predictions using the feedback information (U) provided by the radiologist.

Training Schedule: The MANN is trained using ADAM optimizer with the same configuration as CNN and Siamese networks and minibatch size is set to 16. A grid search is performed to find the best parameter values. The best validation results are achieved using 128 memory slots of size 40 and LSTM controller of size 200.

VI. Experimental Results

A. Classification

For the binary classification of the unbalanced classes, performance is quantitatively determined via the precision, recall (sensitivity), specificity and error rate metrics where TP and FN samples are the images of the nodule class predicted as nodules and non-nodules, respectively. Validation results for the 2D and 3D CNNs (AlexNet and ResNet) are provided in Fig. 5. The specified point on each of the curves shows the precision and recall values at the default threshold of the classifier (i.e., 0.5). The black point shows the precision-recall of the automated software. Table IV includes the performance values of all CNNs, as well as those of the automated software (which is treated as the baseline result). To get a better comparison, the results for the automated software was also computed and presented over the same validation set. The best result of each column is shown in bold.

As it can be seen in the table, all CNN networks are outperforming the automated software by a large margin proving the strength of the CNN models. Moreover, 3D networks are performing better than their 2D counterparts, decreasing the error rate about 3%. This shows that the 3D networks are

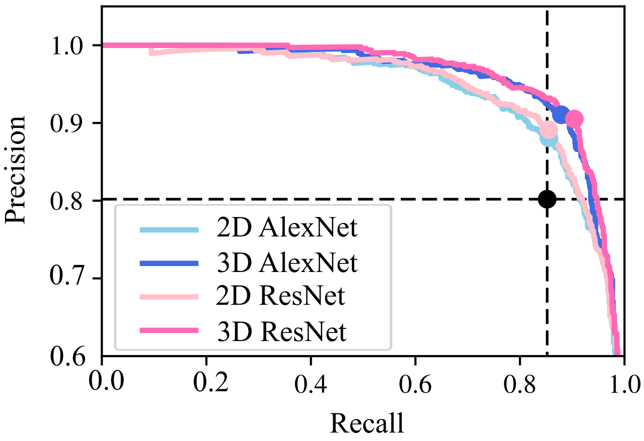

Fig. 5. Precision-recall Curves for the Designed Networks. The points on the curves shows the precision and recall values at the threshold of 0.5. The black dot at the intersection of dotted lines depicts the precision and recall values for the automated software

TABLE IV
THE CLASSIFICATION RESULT OF THE VARIOUS MODELS

	Presicion	**Recall**	**Specificity**	**Error rate**
Automated Software on all images	80.19%	85.21%	71.78%	21.79%
Automated Software on validation set	73.65%	82.13%	70.01%	25.34%
2D AlexNet	88.09%	85.52%	88.32%	13.09%
2D ResNet	89.14%	85.52%	89.47%	12.51%
3D AlexNet	**91.05**%	87.92%	**91.26**%	10.42%
3D ResNet	90.51%	**90.42**%	90.42%	**9.58**%

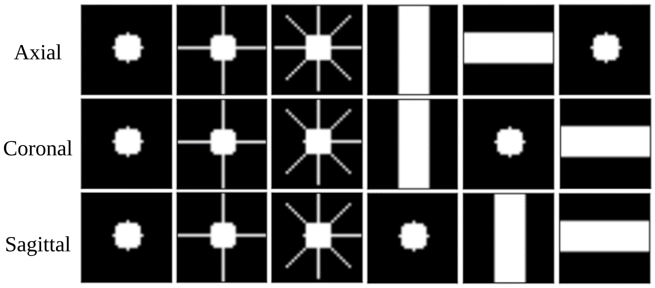

Fig. 6. Examples of synthetic nodule candidates generated for the visualization purpose. These images are mimicking the structures appear in the real images, but with a much simpler appearance. Each column depicts one image of 5 pixels radius from different shape groups. From left to right, the shape groups are: simple sphere, sphere with speculations, sphere with multiple speculations, and vessels along different axes

capable of encoding and exploiting the complicated anatomical surrounding environments of the volumetric image. 3D ResNet achieves the lowest error rate of 9.58%, outperforming 3D AlexNet with the error rate of 10.42%. ResNet also results in a higher sensitivity (recall of 90.42% compared with 87.92% of AlexNet) meaning that it misses less nodules. However, 3D AlexNet gives a slightly higher precision and specificity, reporting fewer false positives. All in all, the classification results support no major benefit in using a more complicated, state-of-the-art CNN model in the case of the data in hand.

B. In-depth Analysis and Visualization of Deep Networks

We compared the performance of the automated algorithm and the superior deep network (ResNet) by counting the number of examples that are correctly classified or misclassified by either of them. These values are presented in terms of percentages in Table V. These values indicates that the designed ResNet is performing almost five times (19.84% of ResNet compared with 4.08% of automated software) better than the standard automated radiology software.

We estimated the RF of units from different layers using the method introduced in [33]. The estimated RFs are later utilized to segment images using the feature maps of different units. Due to simplicity, it is done only for the AlexNet model which has a more straightforward structure.

We also generated 6 groups of synthetic samples of different sizes and shapes. Since some nodule candidate images have a very complicated appearance, it's hard to tell what is the object type that is detected by a specific unit. Therefore, looking at the activation regions of the maximally activating synthetic images with simpler appearance can help to better understand the exact representation learned by each unit. Axial, coronal and sagittal planes of one synthetic example per group is depicted in Fig. 10. Candidates are generated in different sizes with the radius changing from 1 to 15 pixels. The spherical shapes are mimicking the nodule examples while the tube shapes are more like non-nodule cases, especially vessels.

TABLE V
CROSS-COMPARISON OF THE CLASSIFICATION METHODS

Both Correct	**Software correct Only**
72.46%	4.08%
19.84%	5.50%
ResNet Correct Only	**Both Wrong**

Our results shows that object detector units emerge within the network trained to separate nodules from non-nodules. Example of such units of the first and last layers of AlexNet are provided in Fig. 7. The network is trained by the real potential nodule images, and RF-based segmentation is done for both real (left panel) and synthetic (right panel) data. In each image, the area that is not shaded in gray depicts the activation region.

As expected, the first and last convolution layers are learning the low and high level representations, respectively. Comparing the activation regions of the units of con1 and conv5 shows that the activation regions become more semantically meaningful with increasing the depth of layers. Units at the early layers are more responsive to simple shapes and edges. For example, unit 7 and 11 of conv1 are responding to edges created by changes in contrast (i.e. change in color) in different directions. Unit 7 is capturing the edge of vertical chest walls in real data. Since there is no chest wall in synthetic data, this unit is capturing the right edge of the tube shapes which shows the same behavior. The same behavior is observed by unit 11 which gets activated when there is a change in color from top to bottom and from white to black. Unit 16 is responding to the whole white area presented in the image.

Fig. 7 also shows examples of object detector units of the last convolution layer, conv5. Unit 1 is maximally responding to thin horizontal tube shapes which are mostly vessels. Unit 4 is capturing the chest wall on the left side of the ROI. Unit 6 is activated by simple nodules placed at the center of the ROI. Unit 13 is showing a complicated yet interesting behavior. It's searching the environment surrounding real nodules and responds to the objects such as nodule speculations or small vessels. Activation regions in synthetic images shed more light on the behavior of this unit.

C. Adaptation with Siamese and MANN

Our results show that the Siamese network is not capable of accurately determining the similarity/dissimilarity of the provided inputs. The test accuracy at the best validation checkpoint and threshold for the top performing convolutional siamese network is %78.3. The corresponding precision and recall values are %64.9 and %73.7, respectively. This proves that the siamese network is not able to handle too much variation exists in the data. The results also shows the inability of the network to generalize to new sets of images from new subjects. This indicates the need for a more robust model equiped with a memory structure, capable of encoding and accumulating the information provided so far. Then, this model must be able to rapidly access and retrieve the significant amount of information required by the new task.

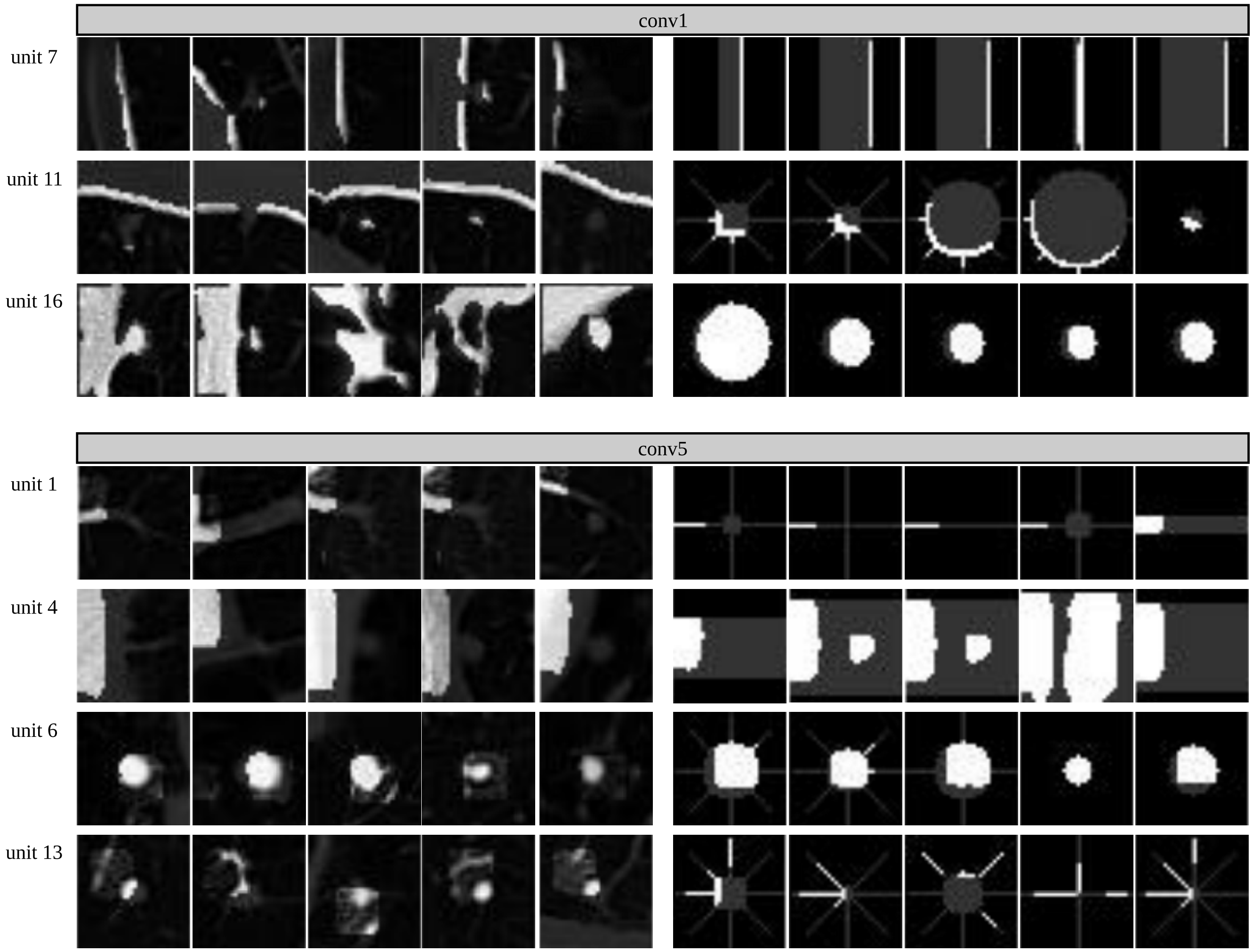

Fig. 7. Examples of object detector units from the first and last convolution layers of the AlexNet. The unshaded area in each image depicts the activation region. Each row contains the five maximally activating images. Results are presented for both real (Left) and synthetic (Right) images and the same units.

The memory-augmented network, in contract, shows a remarkable performance. The classification accuracies for the MANN network is provided in the top panel of Fig. 8. The accuracies are computed for up to 10 feed-backs. For example, the 1st instance accuracy is the classification accuracy for just the first presentation of samples of each class. The second instance accuracy is the classification accuracy of the second observation of both classes, and so on. The curves are plotted only for the first, second, fifth and tenth instances. The first instance accuracy is above chance level which indicates that the MANN is performing *educated guess* for new data samples based on the images it has already seen and stored in the memory. The network also exhibits high classification accuracy on just the second presentation of a sample from a class (above 80%) after only 40,000 episodes reaching up to 95.1% and 95.6% accuracy by the fifth and tenth feed-backs. Training is conducted up to 250,000 episodes where no further improvement was observed in the accuracies.

After every 1,000 training episodes with randomly chosen labels, the network is given a series of validation episodes (i.e. 500 episodes, each containing 10 image of each class). In these episodes, no further learning happened, and the network is to predict the class label for never-before-seen samples pulled from a disjoint validation set. The average accuracies (± std) are depicted in the bottom panel of Fig. 8. Again, the first instance accuracy is above chance level. The highest validation accuracy is achieved at the 226,000th episode with 60.58% and 72.04% for the first and second feed-backs, reaching up to 78.78% and 87.75% by the fifth and tenth, respectively.

We also investigated the robustness of the trained networks by applying different types of distortions with various intensities to the images. This provides a good simulation of cases in which the new images are coming from different sources. Using different types of scanners can cause remarkable changes in the image appearance and resolution. The results for the best trained networks are presented in Fig. 9. The curves indicates that MANN is less affected by distortions while the memory-less networks performance shrinks to chance level rapidly with increasing the noise level.

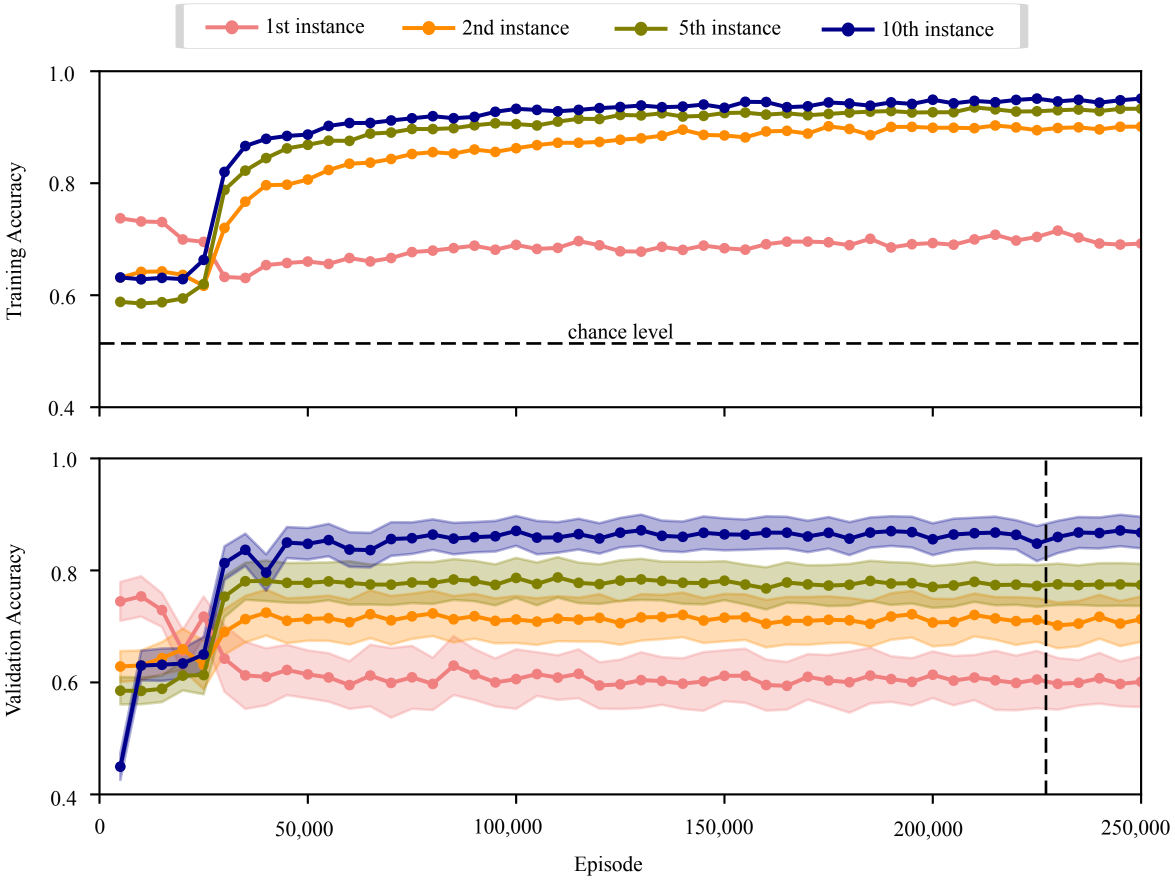

Fig. 8. Training (top) and validation-set (bottom) accuracies of nodule/ non-nodule classification using MANN. The vertical dashed line in the bottom panel depicts the results at the episode with the best validation accuracy

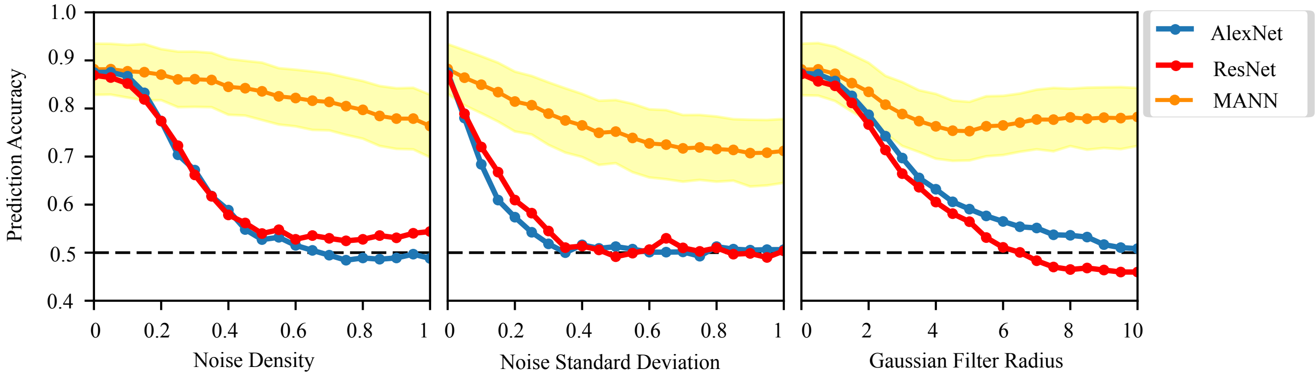

Fig. 9. Comparison of the validation-set classification accuracies of AlexNet, ResNet, and MANN in response to applying different type of distortions on images; namely, salt and pepper noise (Left), Gaussian noise (Middle), and blurring (Right). The yellow shaded areas depicts the standard deviation of MANN's accuracy in the 500 conducted episodes

VII. Conclusion

This paper explored different architectures of deep learning techniques for lung nodule classification. Experimental results demonstrated that deep convolutional neural networks produces promising results in comparison against traditional radiology software. We also showed the feasibility of rapidly adapting deep networks to feedback or change in data. Our approach significantly outperformed state-of-the-art architectures of deep networks such as residual networks and alexnet. Our paper provided unique insight into the network behaviors by combining visualization techniques and radiologist's input. In the future, we will extend our technique to other applications

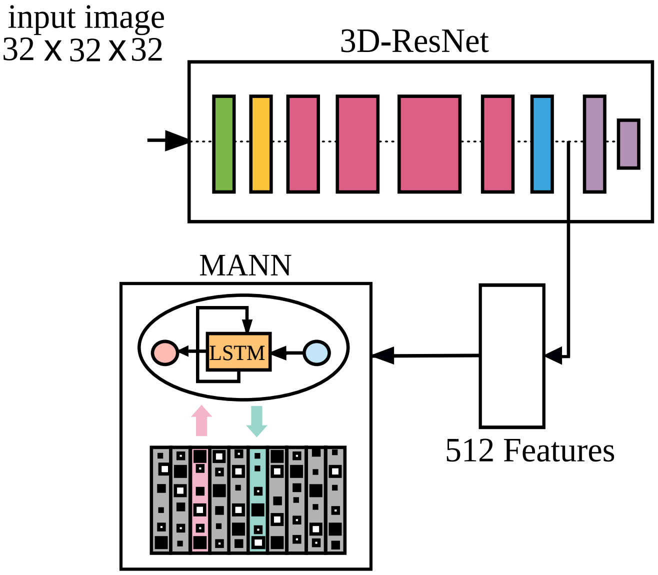

Fig. 10. Examples of synthetic nodule candidates generated for the visualization purpose. These images are mimicking the structures appear in the real images, but with a much simpler appearance. Each column depicts one image of 5 pixels radius from different shape groups. From left to right, the shape groups are: simple sphere, sphere with speculations, sphere with multiple speculations, and vessels along different axes

of medical imaging. We will also investigate different read and write mechanisms to improve the effectiveness of our memory recurrent network. Finally, an interesting research direction is whether our proposed network can discover mislabeled data. This is important as many medical applications intrinsically come with label noises.

Acknowledgment

The authors would like to thank Garglet LLC for supporting our research.

References

[1] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2017," *CA: a cancer journal for clinicians*, vol. 67, no. 1, pp. 7–30, 2017.

[2] V. P. Doria-Rose and E. Szabo, "Screening and prevention of lung cancer," *Lung cancer: a multidisciplinary approach to diagnosis and management*, vol. 2, 2010.

[3] A. Brady, R. Ó. Laoide, P. McCarthy, and R. McDermott, "Discrepancy and error in radiology: concepts, causes and consequences," *The Ulster medical journal*, vol. 81, no. 1, p. 3, 2012.

[4] A. P. Brady, "Error and discrepancy in radiology: inevitable or avoidable?" *Insights into imaging*, pp. 1–12, 2016.

[5] T. N. Shewaye and A. A. Mekonnen, "Benign-malignant lung nodule classification with geometric and appearance histogram features," *arXiv preprint arXiv:1605.08350*, 2016.

[6] K. Awai, K. Murao, A. Ozawa, M. Komi, H. Hayakawa, S. Hori, and Y. Nishimura, "Pulmonary nodules at chest ct: effect of computer-aided diagnosis on radiologists detection performance," *Radiology*, vol. 230, no. 2, pp. 347–352, 2004.

[7] B. Sahiner, H.-P. Chan, L. M. Hadjiiski, P. N. Cascade, E. A. Kazerooni, A. R. Chughtai, C. Poopat, T. Song, L. Frank, J. Stojanovska *et al.*, "Effect of cad on radiologists' detection of lung nodules on thoracic ct scans: analysis of an observer performance study by nodule size," *Academic radiology*, vol. 16, no. 12, pp. 1518–1530, 2009.

[8] S. L. A. Lee, A. Z. Kouzani, and E. J. Hu, "Random forest based lung nodule classification aided by clustering," *Computerized medical imaging and graphics*, vol. 34, no. 7, pp. 535–542, 2010.

[9] M. Firmino, A. H. Morais, R. M. Mendoça, M. R. Dantas, H. R. Hekis, and R. Valentim, "Computer-aided detection system for lung cancer in computed tomography scans: review and future prospects," *Biomedical engineering online*, vol. 13, no. 1, p. 41, 2014.

[10] Q. Dou, H. Chen, L. Yu, J. Qin, and P.-A. Heng, "Multilevel contextual 3-d cnns for false positive reduction in pulmonary nodule detection," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 7, pp. 1558–1567, 2017.

[11] J. Dehmeshki, X. Ye, X. Lin, M. Valdivieso, and H. Amin, "Automated detection of lung nodules in ct images using shape-based genetic algorithm," *Computerized Medical Imaging and Graphics*, vol. 31, no. 6, pp. 408–417, 2007.

[12] W. Shen, M. Zhou, F. Yang, D. Yu, D. Dong, C. Yang, Y. Zang, and J. Tian, "Multi-crop convolutional neural networks for lung nodule malignancy suspiciousness classification," *Pattern Recognition*, vol. 61, pp. 663–673, 2017.

[13] Y. Xu, T. Mo, Q. Feng, P. Zhong, M. Lai, I. Eric, and C. Chang, "Deep learning of feature representation with multiple instance learning for medical image analysis," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1626–1630.

[14] K.-L. Hua, C.-H. Hsu, S. C. Hidayati, W.-H. Cheng, and Y.-J. Chen, "Computer-aided classification of lung nodules on computed tomography images via deep learning technique," *OncoTargets and therapy*, vol. 8, 2015.

[15] A. A. A. Setio, F. Ciompi, G. Litjens, P. Gerke, C. Jacobs, S. J. van Riel, M. M. W. Wille, M. Naqibullah, C. I. Sánchez, and B. van Ginneken, "Pulmonary nodule detection in ct images: false positive reduction using multi-view convolutional networks," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1160–1169, 2016.

[16] W. Jorritsma, F. Cnossen, and P. van Ooijen, "Improving the radiologist–cad interaction: designing for appropriate trust," *Clinical radiology*, vol. 70, no. 2, pp. 115–122, 2015.

[17] T. Drew, C. Cunningham, and J. M. Wolfe, "When and why might a computer-aided detection (cad) system interfere with visual search? an eye-tracking study," *Academic radiology*, vol. 19, no. 10, pp. 1260–1267, 2012.

[18] B. de Hoop, D. W. De Boo, H. A. Gietema, F. van Hoorn, B. Mearadji, L. Schijf, B. van Ginneken, M. Prokop, and C. Schaefer-Prokop, "Computer-aided detection of lung cancer on chest radiographs: effect on observer performance," *Radiology*, vol. 257, no. 2, pp. 532–540, 2010.

[19] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in Neural Information Processing Systems*, 1994, pp. 737–744.

[20] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML Deep Learning Workshop*, vol. 2, 2015.

[21] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," in *Proceedings of the Cognitive Science Society*, vol. 33, no. 33, 2011.

[22] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "One-shot learning with memory-augmented neural networks," *arXiv preprint arXiv:1605.06065*, 2016.

[23] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[26] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[27] A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, "Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2013, pp. 246–253.

[28] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[30] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[31] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[32] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[34] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," *Computer Vision–ECCV 2010*, pp. 213–226, 2010.

[35] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[36] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 427–436.

[37] K. H. Lee, J. M. Goo, C. M. Park, H. J. Lee, and K. N. Jin, "Computer-aided detection of malignant lung nodules on chest radiographs: effect on observers' performance," *Korean journal of radiology*, vol. 13, no. 5, pp. 564–571, 2012.

[38] F. F. Li, R. VanRullen, C. Koch, and P. Perona, "Rapid natural scene categorization in the near absence of attention," *Proceedings of the National Academy of Sciences*, vol. 99, no. 14, pp. 9596–9601, 2002.

[39] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638.

[40] F.-F. Li, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[41] L. Fei-Fei, "Knowledge transfer in learning to recognize visual objects classes," in *International Conference on Development and Learning*, 2006, pp. 1–8.

[42] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 539–546.

[43] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.

[44] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[45] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[46] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[47] C. Giraud-Carrier, R. Vilalta, and P. Brazdil, "Introduction to the special issue on meta-learning," *Machine learning*, vol. 54, no. 3, pp. 187–193, 2004.