Radiologist-Friendly and Automatic Lung Cancer Screening Using Memory Recurrent Networks

Aryan Mobiny, *Member, IEEE,* Hien V. Nguyen, *Member, IEEE,* Supratik K. Moulik, Naveen Garg, and Carol C. Wu

***Abstract*—Most of existing computer aided diagnosis (CAD) systems follow a rigid paradigm where the classifier's decision function is optimized during the training phase, and fixed during the test phase. These systems are often perceived as unfriendly as they do not allow clinicians to provide input. They are also unable to cope with the perpetual changes in data distribution caused by different sensing technologies, imaging protocols, and patient populations. To address these shortcomings, this paper proposes a novel CAD model capable of incorporating expert domain knowledge in real-time to improve its decision function. When the data distribution changes, our classification accuracy on lung nodule data remains above 90% while popular deep networks' accuracies reduce to chance level. We demonstrate that the order of feedback samples affects the final accuracy. An information-gain sorting mechanism is proposed to compute an optimal order of feedback samples. We provide extensive experimental results on two lung nodule datasets to demonstrate that the proposed approach is promising for building a more reliable and radiologist-friendly CAD system.**

***Index Terms*—Domain adaptation, interactive medical diagnosis, lung cancer screening, lung nodule, memory-augmented neural network, radiologist feedback**

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-related deaths [1]. The stage at which diagnosis is made largely determines the overall prognosis of the patient. The five-year relative survival rate is over 50% in early-stage disease, while survival rates drop to less than 5% for late-stage disease [1]. Lung cancer screening of high risk individuals, which is designed to detect the disease at an early stage, has been shown in the National Lung Screening Trial (NLST) to reduce lung cancer mortality by 20% (NLST research team). The main challenge in lung cancer screening is detecting lung nodules [2], [3]. Radiologist fatigue, increasing workload, and stringent turn-around-time requirements are just a few of the factors which negatively impact detection rate for lung nodules. Many studies have documented the occurrence of diagnostic errors in clinical practice, caused by many different contributing factors which can generally be divided into person-specific (such as satisfaction of search etc), nodule-specific (small size, low density) and environment-specific issues (e.g. inadequate equipment, staff shortages, excess workload, etc.) [4], [5].

Computer-aided diagnosis (CAD) systems aim to improve the radiologist's performance in terms of diagnostic accuracy and speed [6]. The role of CAD systems in lung nodule detection and screening has been demonstrated over the years [7], [8], as well as their role in distinguishing benign from malignant nodules [6], [9]. However, automated identification of nodules from non-nodules is quite challenging mainly due to the large variation in sizes, shapes, margins and density of the nodules [10]. The nodules can also occur in different locations (such as peri-fissural, subpleural, endobronchial, perivascular), contributing to the diversified contextual environment around the nodule tissue [11].

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 [13]–[16]. Deep neural networks (DNNs) have an advantage of automatically capturing the image's higher level features directly from the raw input data. This leads to powerful features tuned to specific tasks of medical image analysis [17]–[21]. Recent work has explored deep networks for detecting lung pathology [20], [22]–[25]. In the context of pulmonary nodule classification in Computed Tomography (CT) images, Hua et al. [26] introduced models of a deep belief network and a convolutional neural network that outperforms the conventional hand-crafted features. Setio et al. [27] proposed a multiview convolutional network for lung nodule detection.

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 [28]. 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 expected based on the performance of radiologist and the CAD system in isolation. There are even research reporting no or minimal benefits associated with the presence of CADs on radiologists diagnostic performance (e.g. [29], [30]). Radiologist's *trust* in the CAD system is a key factor in improving the team performance. Inappropriate level of trust in the automation leads to poor performance of the radiologist-

A. Mobiny and H. V. Nguyen are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX, 77004 USA (e-mail: amobiny@uh.edu, and hienvnguyen@uh.edu).

S. K. Moulik ...

N. Garg and C. C. Wu are with the Department of Diagnostic Radiology, University of Texas MD Anderson Cancer Center, Houston, TX, 77030 USA (e-mail: ngarg@mdanderson.org, and ccwu1@mdanderson.org)

CAD team [28]. Radiologists sometimes *under-trust* CADs, preventing them from utilizing its benefits. On the other hand, *over-trust* in automation leads to making diagnostic errors that would not have happened without CAD [30].

Despite the recent advances in deep learning, DNNs still make significant errors that are sometimes obvious to radiologists. To build an appropriate level of trust in machine learning models, it is desirable for radiologists to be able to correct those mistakes on the fly. The CAD model must be able to rapidly incorporate the expert domain knowledge and improve its decision function. We conjecture that this software feature will lead to higher adoption rate of machine learning technologies within the radiology community. The interaction will facilitate clinicians to gain a deeper understanding of how the software works, thereby, facilitate an effective utilization of CADs to increase the efficiency and accuracy of the diagnoses.

Another motivation for the designing an adaptable CAD system is the perpetual problem of *domain shift*. Domain shift occurs when the conditions, or domains, under which the systems were developed differ from those in which we use the systems. While most of existing work focuses on improving the classification accuracy for a static dataset, the problem of adapting a classifier to changes in lung CT data is largely under-investigated. It is especially helpful in the case of lung nodule detection which involves substantial amount of inter-patient and intra-patient variations that heavily deteriorate the generalization performance of deep networks. These variations might be created by the differences in patient populations, image resolutions, scanner types, or imaging protocols. A possible solution is to use domain adaptation (DA) methods [31]–[36]. However, most of existing domain adaptation techniques require a computationally intensive re-training process. This is not acceptable for clinical uses because it can cause a significant delay in diagnoses and treatments.

Motivated by the above challenges, this paper proposes a novel framework for iteratively adapting a lung nodule classifier to changes in the data distribution using only a few feedback examples from clinical experts. This removes the need for annotating many samples from the target domain, as well as re-training the whole deep model from scratch. Recurrent structure of the model enables the radiologist to provide samples one after the other, evaluate the performance enhancement, and stop when satisfied with the model prediction accuracy. Our paper makes the following contributions:

1) We evaluate the performances of deep convolutional networks for the lung nodule detection task. We also study their behaviors in the presence of domain shifts. This is done in two ways: (i) adding different types of never-seen-before distortions to the images of the source domain, and (ii) evaluating our models on a new dataset independent of the training data.
2) A computationally efficient framework is proposed for rapidly adapting deep networks to the radiologists' feedback using a memory-augmented recurrent network. Our extensive experimental results demonstrate that the proposed approach is robust to shifts in the data distribution caused by the variation of sensing technology or patient population.
3) Given a set of feedback samples from radiologists, we propose an effective strategy for finding the optimal order to feed those samples into the recurrent network. Our approach is well-grounded in information theory, and demonstrates to produce significantly better results compared to random ordering.

The rest of this paper is organized as follows: Section II explains how we adapt a classifier to domain experts' input, and compute an optimal feedback sequence. Section III describes the two datasets used in this study. All the designed experiments and obtained results are presented in Section IV. Section V concludes the paper with future research directions.

II. Methodology

A. Rapid Adaptation of Lung Nodule Classifier

Motivation: Radiologist's *trust* in the computer-aided diagnosis (CAD) systems is a key factor in increasing the adoption rate of CAD to clinical practice. Since radiologists are responsible for the final diagnosis, it is important for them to be able correct errors made by CAD systems. As a result, the system has to be able to rapidly adapt to radiologists' feedback and further refine its decision. Such interaction is crucial for radiologists to understand how CAD systems work. This in turns will help improve efficacy and efficiency of the radiologist-CAD team [28].

Another important reason for making lung nodule classifier adaptive is the perpetual problem of domain shift. That is the problem when training data are different from test data. For lung nodule, difference in CT scanner's sensing technology, reconstruction algorithms, or scanning protocol are common reasons causing discrepancy between training and test data. Variation in the patient population is another factor contributing toward shift in data distribution. It has been shown that the classification accuracy reduces dramatically when training and test data come from different distributions [37]–[40]. Re-training a lung nodule classifier is both expensive and time-consuming, which can disrupt the clinical work-flow.

In this section, we propose a framework for adapting lung nodule classifier, using only few feedback inputs, to never-seen-before data distribution. The proposed approach enables radiologists to review a few errors made by a deep network and incorporate his or her knowledge to correct them. Our classifier uses these feedback to further refine its decision.

Background on Memory Recurrent Network: Due to the sequential nature of feedback provided by the radiologist, and the need to encode and accumulate the information over time, neural networks with recurrent structures are a natural choice. They are equipped with an "internal memory" that captures information about what has been calculated so far. Long-short term memory (LSTM) model is introduced as a modification to vanilla recurrent networks (RNNs), capable of encoding the long-term dependencies [41].

In our lung nodule detection problem, the model must be able to quickly encode and retrieve information, and modify its decisions (if needed) to make accurate inferences only within a *small dataset* called feedback samples. This set of samples are provided as either information correction by

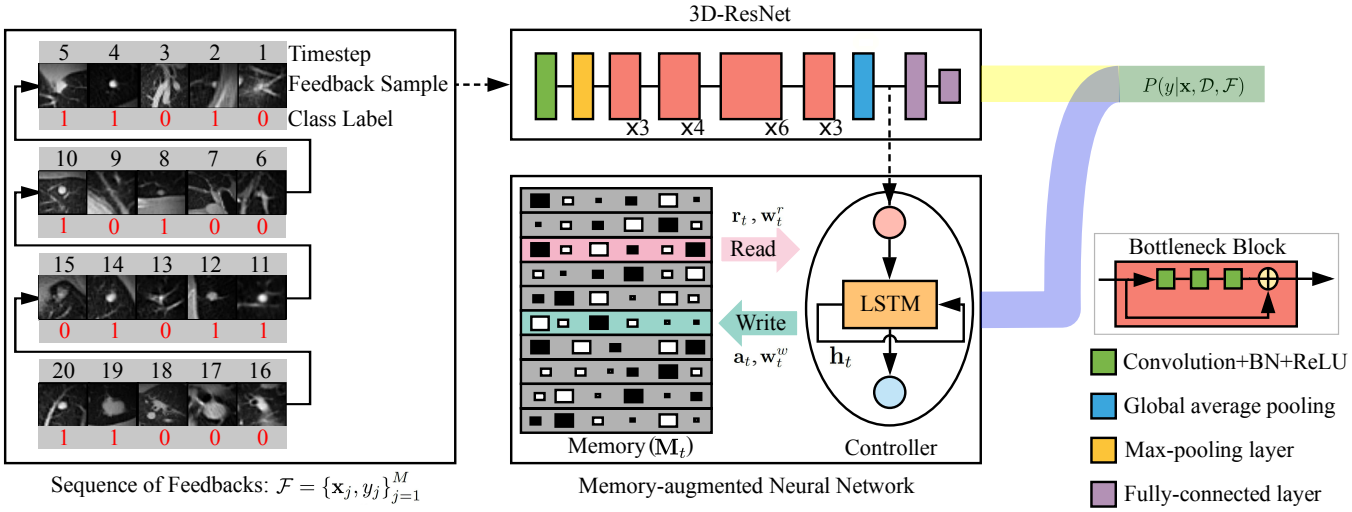

Fig. 1. Architecture of the 3D-LUCRAM classifier, consisting of a 3D-ResNet and a memory-augmented neural network (MANN).

the radiologist or brand-new samples possibly from never-before-seen distributions. Thus the ideal model must learn to capture the cumulative expertise gained *across* domains and continuously adapt to never-before-seen distributions [42]. However, the neural networks with internal memory capacity (such as LSTM) are not able to rapidly encode, store and access a significant amount of new information required at each step. Architectures such as memory networks [43] and Neural Turing Machines (NTMs) are developed as models that meet the requisite criteria. These are external-memory equipped networks capable of rapidly encoding new information and storing them in a stable, addressable representation that can selectively be accessed when needed.

Our approach uses a memory augmented neural network (MANN) [44], a variant of Neural Turing Machine [43], as the main building block for processing sequential feedback information. MANN consists of two main components: a) a LSTM network as the main controller, and b) an external memory bank interacting with the main controller through read and write operations (see Fig. 1). The external memory is 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 reads and writes to the external memory at every time step (i.e. receiving each feedback).

Reading: For a given input $\mathbf{x}_t$ and the memory matrix $\mathbf{M}_t$ with k rows (slots) of size q at time t, the reading operation is done by a weighted linear combination of all memory slots scaled by a normalized read-weight vector $\mathbf{w}_t^r$ as follows:

$$\mathbf{r}_t = (\mathbf{M}_t)^T . \mathbf{w}_t^r \tag{1}$$

Here, $\mathbf{r}_t$ is the content vector retrieved from the memory, and $\mathbf{w}_t^r \in \mathbb{R}^{k \times 1}$ is the read-weight vector. For a given input $\mathbf{x}_t$, the controller will produce a key $\mathbf{k}_t$ computed as $\mathbf{k}_t = tanh(\mathbf{W}_{hk}\mathbf{h}_t + \mathbf{b}_k)$ from the controller hidden states ($\mathbf{h}_t$). $\mathbf{W}_{hk}$ and $\mathbf{b}_k$ are the corresponding weight matrix and bias respectively. This key will be compared against each memory slot $\mathbf{M}_t(i)$ using the cosine similarity measure $K(.,.)$. Then to specify how much each slot should contribute to $\mathbf{r}_t$, the similarity is used to produce the read-weight vector $\mathbf{w}_t^r$:

$$\mathbf{w}_t^r = \text{softmax}\big[K(\mathbf{k}_t, \mathbf{M}_t(i))\big] \tag{2}$$

where softmax is used to get the normalized weight vector with its elements summing to one. This vector allows the controller to select values similar to previously-seen values, which is called content-based addressing.

Writing: To write into the memory, the controller will interpolate between writing to the most recently read memory rows and writing to the *least-used* memory rows. If $\mathbf{w}_{t-1}^r$ is the read-weight vector at the previous time step, and $\mathbf{w}_{t-1}^{lu}$ is a weight vector that captures the least-used memory location, the write weights $\mathbf{w}_t^w \in \mathbb{R}^{1 \times k}$ is then computed using a learnable sigmoid gate:

$$\mathbf{w}_t^w \leftarrow \sigma(\alpha_t)\mathbf{w}_{t-1}^r + (1 - \sigma(\alpha_t))\mathbf{w}_{t-1}^{lu} \tag{3}$$

where $\sigma(.)$ is the sigmoid function. α_t is a scalar computed as $\alpha_t = \mathbf{w}_\alpha \mathbf{h}_t + b_\alpha$ at each time step, and $\mathbf{w}_t$ and b_t are trainable parameters learned discriminatively through back-propagation. This encourages information to be written into either rarely-used locations of the external memory to preserve recently encoded information, or the last used location to *update* the memory with newer, possibly more relevant information. The i^{th} memory slot at time-step t, $\mathbf{M}_t(i)$, is then updated as:

$$\mathbf{M}_t(i) \leftarrow \mathbf{M}_{t-1}(i) + \mathbf{w}_t^w(i).\,\mathbf{a}_t \tag{4}$$

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

To create the least used weight vector $\mathbf{w}_t^{lu}$, the controller maintains a usage-weight vector $\mathbf{w}_t^u$ which gets updated updated after every read and write step as:

$$\mathbf{w}_t^u \leftarrow \gamma \mathbf{w}_{t-1}^u + \mathbf{w}_t^r + \mathbf{w}_t^w \tag{5}$$

where $\gamma \in [0, 1]$ is a scalar parameter used to determine how quickly previous usage values should decay. The least used weight vector $\mathbf{w}_{t-1}^{lu}$ is a one-hot-encoded vector generated from $\mathbf{w}_{t-1}^u$ by setting its minimum element to 1, and all other elements to 0.

Finally, MANN uses the concatenation of the read content vector and the hidden nodes $(\mathbf{r}_t, \mathbf{h}_t)$ to predict the output. The introduction of an external memory enables the recurrent network to store and retrieve much longer-term information compared to LSTM. This frees up the main controller and increases its capacity of learning highly complicated patterns within the data.

Adaptive Lung Nodule Classifier via Memory Augmented Recurrent Network: Suppose $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ is the initial set of training data. Let $\mathcal{F} = \{\mathbf{x}_j, y_j\}_{j=1}^{M}$ denote the feedback samples provided by physicians. Given a sample $\mathbf{x}$, our goal is to estimate its true label by conditioning on the initial training set and the feedback data. The conditional probability can be written as follows:

$$P(y|\mathbf{x}, \mathcal{D}, \mathcal{F}) = \frac{P(\mathbf{x}, y, \mathcal{D}, \mathcal{F})}{P(\mathbf{x}, \mathcal{D}, \mathcal{F})} = P(y|\mathbf{x}, \mathcal{D}) \frac{P(\mathcal{F}|\mathbf{x}, y, \mathcal{D})}{P(\mathcal{F}|\mathbf{x}, \mathcal{D})}$$

The first term $P(y|\mathbf{x}, \mathcal{D})$ depends only on the initial training set and the given input, but not the feedback $\mathcal{F}$. Thus it can be approximated by training a convolutional neural network (ResNet) on $\mathcal{D}$. The denominator in the second term does not affect the classifier's decision as it is the same for every y. Therefore, we only need to estimate $P(\mathcal{F}|\mathbf{x}, y, \mathcal{D})$ to update the classifier's decision. To this end, we model this likelihood function using a MANN. We then merge together the output of MANN and ResNet to form an adaptive classifier as illustrated in Fig. 1. We call it Lung nodUle Classifier with Rapid Adaptation Mechanism (LUCRAM) and will implement . We expect LUCRAM to generalize well to a new set of nodule images using only few feedback samples. However, in practice, we observe that the 3D-LUCRAM training converges rather slowly. This could be because the recurrent network cannot scale properly to the large input images (in our case, $32 \times 32 \times 32 = 32,768$ pixels). We mitigate this issue by passing images to a ResNet before feeding them into LUCRAM. Specifically, 512-dimensional output of ResNet's average-pooling layer is used as the representation of input images fed to 3D-LUCRAM. Our experimental results show that this modification dramatically improves the convergence speed of LUCRAM.

Training Procedure: We train the system by sequentially feeding the input $\mathbf{x}_t$ and the time-delayed output y_{t-1} to the network, and predict the current label y_t. Each sequence of input images is called an *episode*. This simulates the sequential feedback from radiologists in the evaluation phase. This idea is inspired by the human learning and evolution through generations. Each training episode mimics a learner's lifespan where it learns to optimize its performance. Next episodes are like the next generation learners using the accumulated knowledge to solve a similar problem regardless of a possible shift in the data.

More importantly, labels are randomly shuffled from episode to episode. For example, nodules can be labeled as 1 in one episode and 0 in another. Note that labels are consistent within the same episode. This strategy helps prevent the LUCRAM from learning a fixed mapping from samples to their class labels, but a dynamic binding between image features and the labels provided in the feedback. Consider the scenario where we do not shuffle the labels, the network can simply predict the ground-truth labels from input images instead of relying on the labels provided in the feedback. This is undesirable as the model becomes insensitive to the feedback information. Similar strategy was employed to learn dynamic mappings for different tasks [44]. The shuffling is limited to the training phase only, and is removed during the evaluation. The network's parameters are optimized through maximizing the cross entropy between predicted probabilistic scores and the ground truth labels. We train the network end-to-end using ADAM optimizer with the same configuration as the CNN and minibatch size is set to 16. A random 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.

B. Sorting the Feedback Samples

As will be shown in the results section, the proposed memory-augmented recurrent architecture is capable of adapting to the new domain and improving its prediction when receiving a few sequential feedback (provided by the radiologist), even when feedback samples are coming from a different distribution. This also enables the radiologists to sequentially provide as many feedback as required and stop when satisfied with the network performance. However, feedback samples naturally does not correspond to an ordered sequence, but to an unordered set of inputs. In this case, an important invariance property that must be satisfied is that shuffling the order of inputs should not alter the network prediction.

Does the order matter? We conducted experiments to check if the order of input feedback impacts the performance of the adaptive model. This is done by consecutively selecting random sets of inputs with different length (mimicking feedback samples from radiologist) from LUNA-16 data, randomly shuffling the order of samples (to get several permuted sequences), and fed each permuted sequence to the model. We then freezed the model and evaluated its performance over a fixed set of thousand samples.

The changes of prediction accuracies obtained by reordering the feedback samples are presented in Fig. 2. Performance variation is measured by the semi-interquartile range (IQR) (one half the difference between the 75^{th} percentile and the 25^{th} percentile) of all prediction accuracies achieved by reordering the instances. In other words, each single point in a box-plot is corresponding to the changes (measured by semi-IQR) in the prediction accuracy when receiving a set of feedback samples in different orders.

Our results demonstrates that the *a priori* choice of ordering of the data to be presented to the model matters; especially for shorter sequence of feedback. Similar observations have been made in the context of sequence-to-sequence (seq2seq) learning where the order in which input data is shown to the model has an impact on the learning performance [45].

According to Fig. 2, the fluctuations in prediction accuracy is less severe in 3D-LUCRAM than in 2D-LUCRAM. 3D-LUCRAM's accuracy often alters about 5% and 2% (on

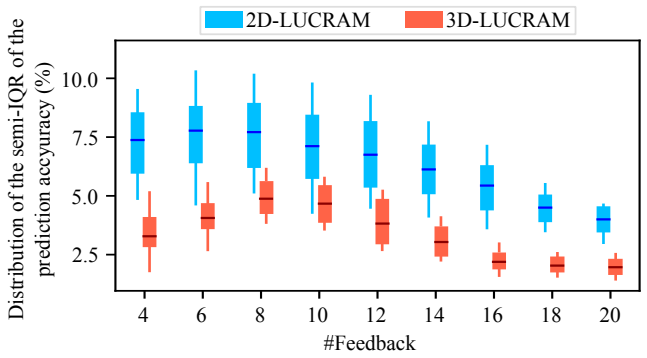

Fig. 2. Distribution of the semi-IQR of the prediction accuracies for feedback sequences of different length. Each point in the distributions corresponds to the semi-IQR of all accuracies obtained from re-ordering a set of provided feedbacks.

average) after 10 and 20 feedback respectively. These values are 7.5% and 5% for 2D-LUCRAM after the same number of feedback.

Information-Gain Sorting: Now that we know the order matters, the question is how should we represent a set of input feedback where an obvious, natural order cannot be determined? As discussed earlier, the proposed framework modifies its decisions and adapts sequentially. This is done through updating the controller states and memory contents according to the samples provided so far. In other words, LUCRAM uses the prediction error of the former samples for its later decisions which means learning more from uncertain samples, the ones with higher prediction error.

Given this mechanism, the more informative the samples that are presented earlier in the sequence, the richer will be the encoded knowledge (which also includes the ground truth label) of the new domain. Therefore, we propose an algorithm aimed at re-ordering the samples according to their informativeness, i.e. how well an instance helps reduce the model uncertainty. We argue that an intuitive approach is to feed the more complex (uncertain) instances earlier in the sequence. This mechanism enables network to quickly adapt to the new domain while deferring specific kinds of minor modifications until the domain knowledge is available. In fact, when making later decisions, the network has access to the entire, more complicated information that has been presented earlier and encoded in its states and memory.

In information theory, Entropy is the measure commonly used to quantify the information content produced by *one* stochastic source of data. Kullback-Leibler divergence, also known as information gain, is a measure of comparing *two* distributions: $D_{KL}(p|q)$ quantifies how similar a *true* probability distribution, p, is to a candidate distribution q. Kullback-Leibler divergence can be decomposed into two terms:

$$D_{KL}(p|q) = \sum_i p_i \log \frac{p_i}{q_i} = -\sum_i p_i \log q_i - H(p) \qquad (6)$$

where the first term is often called cross-entropy and $H(p)$ is the entropy of p. In our task, p will be the distribution of true labels which is held fixed, so $H(p)$ doesn't change with the parameters of the model. Thus, cross-entropy and D_{KL} are equivalent (differ by a constant factor for all q) and can be used interchangeably to measure the information gain.

Algorithm 1 Information-Gain Sorting (IGS) returns a sorted sequence given a set of L labeled feedback samples. Re-ordering of the instances is done according to their informativeness measured as the prediction entropy.

procedure IGS($\mathcal{F}$) ▷ $\mathcal{F} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{L}$
  $\mathcal{S} \leftarrow \{\}$
  while $|\mathcal{F}| > 0$ **do** ▷ So long as some feedback left
    for $\{(\mathbf{x}_i, y_i)\}$ **in** $\mathcal{F}$ **do**
      $p_i \leftarrow P(\mathbf{x}_i|\mathcal{S};\theta)$
      $e_i = \sum_{j=1}^{c} y_{ij} log(p_{ij})$ ▷ e for entropy
    end for
    $i^* = \text{argmax}_i(e_i)$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\mathbf{x}_{i^*}, y_{i^*})\}$
    $\mathcal{F} \leftarrow \mathcal{F} - \{(\mathbf{x}_{i^*}, y_{i^*})\}$
  end while
  return S ▷ The sorted sequence of feedback
end procedure

Therefore, we used cross-entropy between the ground truth provided by radiologist (y_i) and recurrent structure's current prediction (p_i) to model the amount of information (i.e. average number of bits) the network gains at a certain time. In fact, the value of cross-entropy tells us how important a given feedback sample is at a certain time step and is used to decide the ordering of the samples of a given set. This method is called *Information-Gain Sorting*, shortened as IGS, and summarized as in Algorithm 1. At each time-step, the remaining feedback samples in $\mathcal{F}$ (i.e. the ones have not been selected yet) are fed to the model one after the other, and the one with higher entropy is selected as the most informative sample at that time. Then this sample is excluded from $\mathcal{F}$ and added to the sorted sequence $\mathcal{S}$.

III. Dataset

We used two different sets of CT images in our experiments. These datasets are collected independently, and thus it is reasonable to consider their images as samples from different distributions. We randomly selected 70% of the samples of the first set (i.e. our data) for training (either CNNs or LUCRAM) and the remaining 30% for evaluating the network performance on a data from the same distribution. The second set, however, is the publicly available dataset used in LUNA-16 challenge. This dataset is assumed as samples from *never-seen-before* distribution (target domain) and is used to examine the adaptability of both CNN and LUCRAM models.

A. Our Dataset

The dataset includes 226 unique CT Chest scans captured by General Electric and Siemens scanners at a single hospital on a single day. The data was preprocessed by an automated segmentation software in order to identify structures to the organ level. From within the segmented lung tissue, a set of potential nodule point is generated based on the size and shape of regions within the lung which exceeds the air Hounsfield

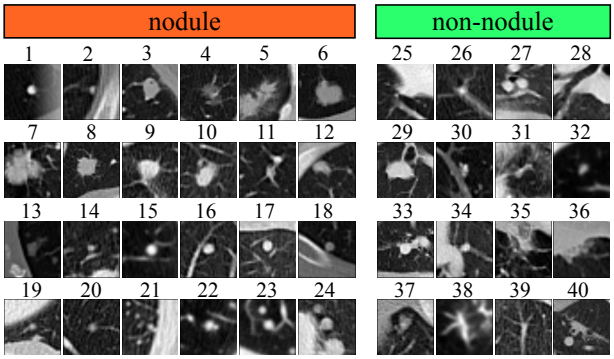

Fig. 3. Sample images of nodules (Left) and non-nodules (Right) selected from our dataset. Each image is a slice along 2D axial plane in the middle of the volume.

Unit (HU) threshold. Additional filters, based on symmetry and other common morphological characteristics, are applied to decrease the false positive rate while maintaining very high sensitivity.

Bounding boxes with at least 8 voxels padding surrounding the candidate nodules are cropped and resized to $32 \times 32 \times 32$ pixels. Each generated candidate is reviewed and annotated by at least one board certified radiologist. From all the generated images (about 7400 images), around 56% were labeled as nodules and the rest non-nodules. Figure 3 shows examples of extracted candidates and the corresponding labels provided by radiologists. These images illustrate the highly challenging task of distinguishing nodules from non-nodule lesions. The first reason is that the pulmonary nodules come with large variations in shapes, sizes, types, etc. In Figure 3, 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, they also often have a non-spherical shape (12-13) and irregular margins. 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 (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).

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. For all these reasons, the detection and classification of lung nodules is a challenging task, even for experienced radiologists.

B. LUNA-16 Challenge Dataset

We also use the candidate nodules provided by the LUNA-16 challenge [46] to evaluate our proposed architecture and emphasis its generalization on a dataset other than our own data. This dataset is a subset of LIDC-IDRI data [47], the largest publicly available reference database for lung nodules including a total of 1018 CT scans. It consists of regular dose and low-dose CT scans collected from a wide range of scanner models and acquisition parameters from seven different participating academic institutions. This makes the whole dataset heterogeneous which makes it suitable for evaluating the generalization performance of the proposed framework.

LUNA-16 includes candidate nodules generated from only 888 scans and labeled based on the criteria explained in [46]. This results in 750K candidate nodules, containing only about 1500 true nodules. Original 3D image patches are of size $64 \times 64 \times 64$. We down-sampled and halved their size through a bicubic interpolation over 4×4 pixel neighborhood to match our dataset dimensions.

IV. Experiments and Results

A. Nodule Detection task with Baseline CNNs

As the baseline for our experiments on adaptation to new domain, we train two CNNs on our data and evaluate their performance on both our data and LUNA-16. These networks are prepared for both two and three dimensional input data using 2D and 3D convolution and pooling operations.

Modified Architectures: We modify two well-known deep network architectures, AlexNet [14] and ResNet-50 [48], to make them compatible with our 3D data and improve their performance. The final baseline architectures are the result of an extensive hyperparameter search over filter sizes, number of channels, and learning rates.

The modified AlexNet contains eight layers (five convolutional layers followed by three fully-connected layers) similar to the original AlexNet. The local response normalization layers after the first, second, and last convolutional layers proposed in the original paper are replaced by batch normalization (BN) layers [49]. We also add BN before all other ReLU layers. Empirically, BN enables much faster convergence rate during training. It also has regularization effect like dropout because of computing the statistics on every mini-batch (rather than the entire training examples) [49]. We use a smaller stride of 1 and filter size of 4 for the first convolutional layer. This results in to more distinctive features and fewer dead filters, as demonstrated in [50]. The dropout rate is set to 50% [14] in the first two fully-connected layers to prevent over-fitting.

We also modify the 50-layer ResNet described in [48] to fit our data dimensions. At the top of the network, we use a convolutional layer including 32 filters of size 4×4 with stride 2, followed by a 2×2 max-pooling layer with stride 2. These are followed by four so-called bottleneck blocks which are described in details in [48]. Each of the three convolutional layers in a single bottleneck block is followed by a BN applied before the ReLU nonlinearity. A fully-connected layer with 50 hidden units is also added to the network and before the classification layer. This layer helps to improve our classification results. Similar to AlexNet, a dropout with 50% rate is used in this fully-connected layer. We also apply 2D versions of these networks to 2D slices of lung nodule volumes along the x-axis. We choose x-axis because it contains most information according to radiologists' feedback. Having

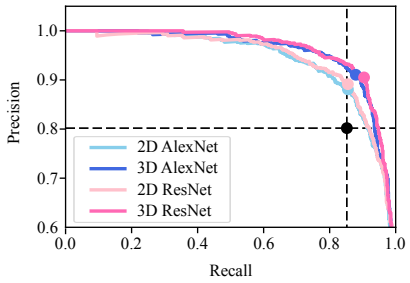

Fig. 4. Precision-recall Curves for the baseline networks vs. the automated software. The points on the curves shows the precision and recall values at the threshold of 0.5. The black dot at the intersection of dotted lines depicts the precision and recall values for the automated software which has an algorithmic method for separating true from false candidate points based on the voxel density and the symmetry charactristic of the candidate point.

both 2D and 3D images enable us to compare to quantify the contribution of the third dimension.

Training Procedure: For both networks, training is done using ADAM optimizer [51] and cross-entropy loss function. We perform data augmentation by randomly rotate nodule volumes 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 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 is similar to applying dropout to the visible layer, i.e., the input. The dropout rate is set to 5% and all the model weights are initialized using Xavier initialization [52].

Baseline CNNs Performance: For the binary classification of the unbalanced classes, performance is quantitatively determined via the precision, recall (sensitivity), specificity and error rate metrics. Test results for the 2D and 3D CNNs are provided in Figure 4. 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 I 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 automated software performance was also evaluated over the same test set. The best result of each column is shown in bold.

According to Table I, 3D networks improve over their 2D counterparts by 3% error points. This shows that the 3D networks are 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. However, looking at the precision-recall curve in Fig. 4, there's no strong evidence for us to prefer one network over the other for the lung nodule classification task.

How do CNNs respond to domain shifts? To examine the adaptability of the trained CNN models, we tested them on

TABLE I
PREDICTION ACCURACY OF POPULAR DEEP NETWORKS TRAINED ON OUR DATA AND TESTED ON BOTH OUR DATA (TOP) AND LUNA-16 (BOTTOM)

		Presicion	Recall	Specificity	Accuracy
Our Data	2D-AlexNet	88.09%	85.52%	88.32%	86.91%
	2D-ResNet	89.14%	85.52%	89.47%	87.49%
	3D-AlexNet	**91.05**%	87.92%	**91.26**%	89.58%
	3D-ResNet	90.51%	**90.42**%	90.42%	**90.42**%
LUNA-16	2D-AlexNet	71.13%	72.20%	70.70%	71.40%
	2D-ResNet	**80.82**%	**79.20**%	**81.20**%	**80.20**%
	3D-AlexNet	60.00%	57.60%	62.30%	60.00%
	3D-ResNet	73.30%	62.10%	71.80%	70.70%

samples from the LUNA-16 dataset. As mentioned, this dataset includes a large number of variations due to differences in subject populations, image resolution and scanner types. LUNA-16 and our dataset were collected independently. Thus it is reasonable to believe that its samples come from distributions different from that of our data which is used for training the models.

The performances of the baseline CNNs trained on our data and evaluated on LUNA-16 samples are presented in the lower part of Table I. Interestingly, 2D networks are generalizing better with the 2D-ResNet achieving the best prediction accuracy of 80.20% where 3D-ResNet suffer a more severe drop to 70.70%. The results demonstrate the CNN networks' lack of generalization to domain shift. This emphasizes the challenge of designing a framework that not only performs well on a provided set of samples, but also quickly adapts to a new domain by providing only a few samples without the need to retrain the whole system from scratch; especially in the medical imaging field where time annotation process is time consuming, costly, and prone to human errors.

B. Evaluation of Adaptive Classifier

Adaptation to feedback: We first verify the effectiveness of LUCRAM by computing its classification accuracy after a few number of feedback. Note that this experiment only uses the training and test sets coming from our data. Each episode consists of a set of feedback samples (i.e. labeled samples) selected randomly from either of the two classes. In the test phase, no further learning occurred and the network was to predict the class labels for samples pulled from a disjoint set. The training and test accuracies of LUCRAM over the episodes are provided in the left and right panels of Fig. 5, respectively. The accuracies are computed for up to 20 feedback instances. For example, the 2^{nd} instance accuracy is the classification accuracy for just the first two samples. The 4^{th} instance accuracy is the classification accuracy of the first four instances, and so on.

Table II summarizes the accuracies over the test set. For both architectures, the 2^{nd} instance prediction accuracy (i.e. observing one sample from each class) is above chance level which indicates that the networks perform *educated guess* for new data samples based on the images it has already seen and stored in the memory. 2D-LUCRAM achieves the highest test accuracy after 226K episodes with 60.58% and 72.04% for the second and forth feedback, reaching up to

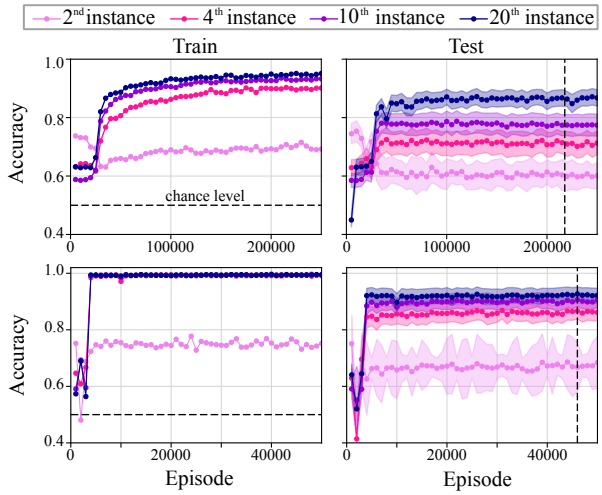

Fig. 5. Training (Left) and test (Right) accuracies of nodule/ non-nodule classification using 2D-LUCRAM (Top) and 3D-LUCRAM (Bottom) on the source domain. In the right panel and at each specific episode, the test accuracy is presented for a network that is trained for that many episodes and is computed as the average accuracy (±std) over 500 sequence of images (of length 20) selected randomly from the whole test set. The vertical dashed line in the right panel depicts the results at the episode with the best test accuracy

78.78% and 87.75% by the tenth and twentieth, respectively. In contrast, 3D-LUCRAM reaches the highest accuracy after only 46K episode with 66.78% and 86.70% for the second and forth feedback, reaching up to 90.60% and 92.68% by the tenth and twentieth, respectively. 3D-LUCRAM significantly outperforms 2D-LUCRAM both in terms of convergence rate and accuracy in adaptive classification setting.

Adaptation to domain shift: We demonstrate the robustness of our adaptive classifier to domain shifts through two experiments:

1) By applying different types of *never-seen-before* distortions with various intensities to the images. One type of distortion is random removal of pixels and replacing them with zero and one values. This creates salt-and-pepper noise effect. To simulate the scenario where the test images have different resolution compared to train image, we apply Gaussian blurring on test images. We allow 3D-LUCRAM to use only 10 samples from each new domain for adaptation. Figure 6 compares the accuracy of the adaptive classifier against that of 3D-AlexNet and 3D-ResNet. The adaptive classifier outperforms deep networks under all kinds and intensities of noises. As the noise level increases, ResNet and AlexNet accuracies are dramatically decreased. In contrast, the classification accuracy of our adaptive classifier remains above 80% even when accuracies of two popular deep networks are reduced to chance. This experiment indicates that the adaptive classifier is highly robust against changes in data distribution.

2) Similar to what we did for CNNs, we use the trained networks to classify another set of images of LUNA-16 challenge. As mentioned, this data is collected independent of our training data and from many different scanner models with various sensor resolution. This enables us to check the networks' adaptability and performance robustness in a more natural setting. Classification results of the proposed adaptive architecture are presented in the bottom panel of

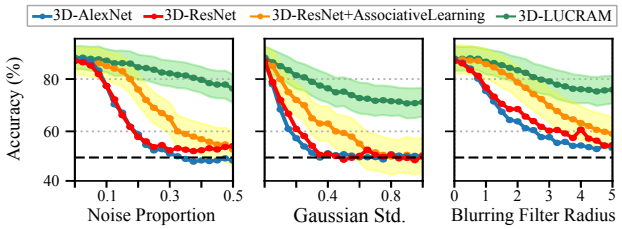

Fig. 6. Comparison of the test classification accuracies in response to applying different type of distortions on images; namely, salt and pepper noise (Left), Gaussian noise (Middle), and blurring (Right). The shaded areas depicts the standard deviation of accuracy over 500 conducted episodes. The black dashed lines depict the chance level.

TABLE II
TEST CLASSIFICATION ACCURACIES (%) FOR LUCRAM ON BOTH SOURCE (OUR DATA) AND TARGET (LUNA-16) DOMAINS.

Domain	Model	Feedback Instance (% correct)					
		2^{nd}	4^{th}	6^{th}	8^{th}	10^{th}	20^{th}
Source	2D-LUCRAM	60.58	72.04	75.59	77.24	78.78	87.75
	3D-LUCRAM	**66.78**	**86.70**	**88.78**	**90.00**	**90.60**	**92.68**
Target	Associative Learning [53]	**70.28**	71.10	71.98	72.87	73.60	81.95
	2D-LUCRAM	59.65	67.78	71.13	73.31	75.10	85.54
	3D-LUCRAM	66.12	71.78	74.63	76.40	78.05	89.04
	3D-LUCRAM + IGS	67.14	**73.17**	**78.28**	**84.27**	**86.93**	**91.08**

Table II. While the CNNs' classification performance drops dramatically (see Table I), 2D and 3D-LUCRAM show a robust performance reaching up to 85.54% and 89.04% after 20 feedback.

We also compare our results with those achieved using the idea of learning by association method proposed by [53]. We modify the model to make the comparison fair. Network structures used in the original paper is replaced with our 3D ResNet-50 structure. First, we train the network on the source domain to minimize the loss function (combination of visit, walker and classification loss) and then test its robustness against the same distortions (see Fig. 6). While the network trained via associative learning performs better than regular CNNs, our proposed structure outperforms it by a significant margin.

For the domain adaptation task, we used the network pre-trained under associative learning regime and fine-tune it using different number of labeled data (feedback samples) from target domain. We ran the fine-tunning for as many iteration as required and reported the best results achieved. A subset of 100 samples from the remaining training data is randomly selected as the required set of unlabeled samples for this approach. Prediction accuracies on the test set is presented in Table II. Our proposed structure outperforms the network trained via associative learning in terms of adapting to the target domain when only few feedback samples are provided. LUCRAM's prediction accuracy reaches up to 90% after 20 feedback, while this value is almost 82% for associative learning for the same number of feedback (labeled samples) and it requires about 200 labeled sample to reach up to 88.5% where its performance saturates and doesn't improve anymore. Moreover, our framework doesn't need any unlabeled data, nor it requires exhaustive fine-tuning on the target domain.

Information Gain Sorting Experiment: Example of a sequence of feedback and its reordered version according to IGS is shown in Fig. 7. It can be inferred that sorting the target domain samples according to their informativeness leads to

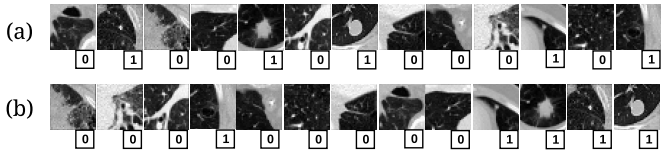

Fig. 7. Sample sequence of thirteen feedback samples ordered randomly (a) and according to IGS (b). Image label is presented in a square below each sample. Each label is used to make prediction in the next time-step.

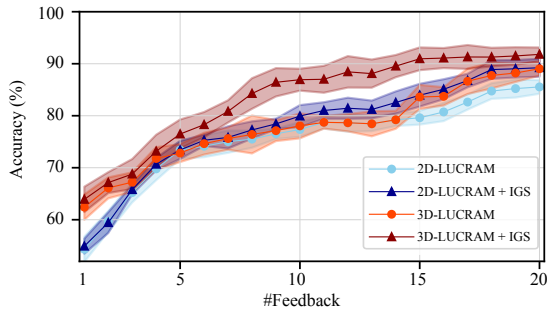

Fig. 8. Comparing the prediction accuracies (±std) of the models when receiving a set of feedback samples both in a random order and as a sequence of feedback sorted according the information gain (IGS). Feedbacks are samples from LUNA-16 data and are fed to the networks that are pre-trained on our data.

a sequence with the more complex, uncertain samples placed earlier in the sequence.

Fig. 8 depicts the performance of both LUCRAM with and without IGS. As shown, the proposed sorting method performs only as well as selecting randomly when only the first few feedback samples are provided. This is due to the poor performance (almost random) of the network after the first few feedback. Therefore, sorting the samples according to their entropy is similar to selecting them randomly. However, as the number of feedback inputs increases, the gap between the random selection and IGS starts to increase. Using one-tailed t-test at the significance level of 0.05, the difference between the two becomes significant after only 10 and 5 feedback inputs for 2D and 3D LUCRAM respectively. This experimentally proves that sorting the provided feedback samples can significantly improve the speed at which the network encodes and retrieves the information of the input domain.

The best performing model among all is the 3D-LUCRAM model equipped with IGS. This model achieves 76.5%, 86.9%, 91.0%, and 91.1% prediction accuracy after 5, 10, 15 and 20 feedback samples of the LUNA-16 data respectively. This is significantly higher than 72.8%, 78.1%, 83.6%, and 89.0% of 3D-LUCRAM (same network with no use of IGS) on the exact same feedback samples. More importantly, 91.1% accuracy of the 3D-LUCRAM with IGS on LUNA-16 data after 20 feedback is close to the 92.68% of the 3D-LUCRAM in the source domain (second row of Table II).

Interestingly, 2D-LUCRAM with IGS performs almost always as well as 3D-LUCRAM without IGS. It shows that feeding the samples in a proper order (determined by IGS) is as important as using the whole volume of the 3D images instead of the 2D center slice along axial plane which is used in 2D-LUCRAM.

V. Conclusions

This paper systematically evaluates the adaptation of deep networks to domain shift. We found that while deep CNNs achieve state-of-the art performance on a set of data, they do not perform well in response to domain shift. We propose a practical adaptive classifier capable of taking a few feedback inputs from radiologists to refine its decision accordingly. This removes the need to re-train the network from scratch (or fine-tune it on the new data) which requires a significant amount of time, computation resources, and human effort. Our experimental results have demonstrated that when data change, the proposed classifier maintains a good performance while popular deep networks' accuracy reduces to chance level. Finally, we show that since the proposed networks gets updated through the sequential presentation of samples, the order of feedback samples matters. We experimentally proved that the most optimal way to feed the samples to the model is to sort them according to the information gained through each samples. This strategy combined with our proposed recurrent model lead to a model that adapts almost perfectly for the lung nodule detection task. The future work will explore different optimization strategies to speed up the convergence of LUCRAM.

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] N. L. S. T. R. Team *et al.*, "Reduced lung-cancer mortality with low-dose computed tomographic screening," *N Engl J Med*, vol. 2011, no. 365, pp. 395–409, 2011.

[3] N. Horeweg, E. T. Scholten, P. A. de Jong, C. M. van der Aalst, C. Weenink, J.-W. J. Lammers, K. Nackaerts, R. Vliegenthart, K. ten Haaf, U. A. Yousaf-Khan *et al.*, "Detection of lung cancer through low-dose ct screening (nelson): a prespecified analysis of screening test performance and interval cancers," *The Lancet Oncology*, vol. 15, no. 12, pp. 1342–1350, 2014.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

[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] 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.

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

[14] 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.

[15] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[16] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[17] 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.

[18] F. C. Ghesu, E. Krubasik, B. Georgescu, V. Singh, Y. Zheng, J. Hornegger, and D. Comaniciu, "Marginal space deep learning: efficient architecture for volumetric image parsing," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1217–1228, 2016.

[19] H. Greenspan, B. van Ginneken, and R. M. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.

[20] Y. Bar, I. Diamant, L. Wolf, S. Lieberman, E. Konen, and H. Greenspan, "Chest pathology detection using deep learning with non-medical training," in *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*. IEEE, 2015, pp. 294–297.

[21] 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.

[22] D. Kumar, A. Wong, and D. A. Clausi, "Lung nodule classification using deep features in ct images," in *Computer and Robot Vision (CRV), 2015 12th Conference on*. IEEE, 2015, pp. 133–138.

[23] J.-Z. Cheng, D. Ni, Y.-H. Chou, J. Qin, C.-M. Tiu, Y.-C. Chang, C.-S. Huang, D. Shen, and C.-M. Chen, "Computer-aided diagnosis with deep learning architecture: applications to breast lesions in us images and pulmonary nodules in ct scans," *Scientific reports*, vol. 6, p. 24454, 2016.

[24] W. Sun, B. Zheng, and W. Qian, "Computer aided lung cancer diagnosis with deep learning algorithms," in *Medical Imaging 2016: Computer-Aided Diagnosis*, vol. 9785. International Society for Optics and Photonics, 2016, p. 97850Z.

[25] A. Mobiny and H. Van Nguyen, "Fast capsnet for lung cancer screening," *arXiv preprint arXiv:1806.07416*, 2018.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, 2018.

[32] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4068–4076.

[33] J. Hu, J. Lu, and Y.-P. Tan, "Deep transfer metric learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 325–333.

[34] T. Xiao, H. Li, W. Ouyang, and X. Wang, "Learning deep feature representations with domain guided dropout for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1249–1258.

[35] K. Kamnitsas, C. Baumgartner, C. Ledig, V. Newcombe, J. Simpson, A. Kane, D. Menon, A. Nori, A. Criminisi, D. Rueckert *et al.*, "Unsupervised domain adaptation in brain lesion segmentation with adversarial networks," in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 597–609.

[36] B. Wang, M. Prastawa, A. Saha, S. P. Awate, A. Irimia, M. C. Chambers, P. M. Vespa, J. D. Van Horn, V. Pascucci, and G. Gerig, "Modeling 4d changes in pathological anatomy using domain adaptation: Analysis of tbi imaging using a tumor database," in *International Workshop on Multimodal Brain Image Analysis*. Springer, 2013, pp. 31–39.

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

[38] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1785–1792.

[39] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 999–1006.

[40] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2066–2073.

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

[42] 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.

[43] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[44] 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.

[45] O. Vinyals, S. Bengio, and M. Kudlur, "Order matters: Sequence to sequence for sets," *arXiv preprint arXiv:1511.06391*, 2015.

[46] A. A. A. Setio, A. Traverso, T. De Bel, M. S. Berens, C. van den Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts *et al.*, "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge," *Medical image analysis*, vol. 42, pp. 1–13, 2017.

[47] S. G. Armato, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, B. Zhao, D. R. Aberle, C. I. Henschke, E. A. Hoffman *et al.*, "The lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans," *Medical physics*, vol. 38, no. 2, pp. 915–931, 2011.

[48] 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.

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

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

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

[52] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[53] P. Haeusser, A. Mordvintsev, and D. Cremers, "Learning by association-a versatile semi-supervised training method for neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 3, no. 5, 2017, p. 6.