Recurrent Networks for Guided Multi-Attention Classification

Xin Dai, Xiangnan Kong, Tian Guo,
John Boaz Lee, and Xinyue Liu
Worcester Polytechnic Institute
Worcester, MA
{xdai5,xkong,tian,jtlee,xliu4}@wpi.edu

Constance Moore
University of Massachusetts Medical School
Worcester, MA
constance.moore@umassmed.edu

ABSTRACT

Attention-based image classification has gained increasing popularity in recent years. State-of-the-art methods for attention-based classification typically require a large training set and operate under the assumption that the label of an image depends solely on a single object (*i.e.*, region of interest) in the image. However, in many real-world applications (*e.g.*, medical imaging), it is very expensive to collect a large training set. Moreover, the label of each image is usually determined jointly by multiple regions of interest (ROIs). Fortunately, for such applications, it is often possible to collect the locations of the ROIs in each training image. In this paper, we study the problem of *guided multi-attention classification*, the goal of which is to achieve high accuracy under the dual constraints of (1) small sample size, and (2) multiple ROIs for each image. We propose a model, called Guided Attention Recurrent Network (GARN), for multi-attention classification. Different from existing attention-based methods, GARN utilizes guidance information regarding multiple ROIs thus allowing it to work well even when sample size is small. Empirical studies on three different visual tasks show that our guided attention approach can effectively boost model performance for multi-attention image classification.

CCS CONCEPTS

• **Information systems → Data mining**; • **Computing methodologies → Neural networks**;

KEYWORDS

Visual attention network; recurrent attention model; brain network classification

ACM Reference Format:
Xin Dai, Xiangnan Kong, Tian Guo, John Boaz Lee, and Xinyue Liu and Constance Moore. 2020. Recurrent Networks for Guided Multi-Attention Classification. In *KDD '20: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 22–27, 2020, San Diego, CA.* ACM, New York, NY, USA, 9 pages. https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

1 INTRODUCTION

Image classification has been intensively studied in recent years in the machine learning community. Many recent work focus on designing deep neural networks, such as Convolutional Neural Networks (CNN), and these have achieved great success on various image datasets. Conventional deep learning methods usually focus on images with relatively "low resolutions" at the level of thousands of pixels (*e.g.*, 28×28, 256×256, and 512×512) [13, 18, 19]. However, many real-world applications (*e.g.*, medical imaging) usually involve images of much higher resolutions. For example, functional Magnetic Resonance Imaging (fMRI) scans usually have millions of voxels, *e.g.*, $512 \times 256 \times 384$ in terms of height, width and depth. Training deep learning models (*e.g.*, CNN) on such images will incur huge computational costs, which grow at least linearly with respect to the number of pixels.

To achieve sublinear computational costs, many attention-based classification techniques (especially hard attention methods) have been proposed [3, 19]. For example, Recurrent Attention Model (RAM) [19] is an attention-based model, trained using reinforcement learning (RL), which maintains a constant computational cost w.r.t. the number of image pixels for image classification. RAM moves its visual attention sensor on the input image and takes a fixed number of glimpses of the image at each step. RAM has demonstrated superior performance on high-resolution image classification tasks, making a strong case for the use of attention-based methods under this setting.

In this paper, we mainly focus on the multi-attention classification problem, where each image involves multiple objects, *i.e.*, regions of interest (ROIs). The label of an image is determined jointly by multiple ROIs through complex relationships. For example, in brain network classification, each fMRI scan contains multiple brain regions whose relationships with each other may be affected by a neurological disease. In order to predict whether a brain network is normal or abnormal, we need to examine the pairwise relationships between different brain regions. If we focus on just a single brain region, we may not have enough information to correctly predict the brain network's label. Many other visual recognition tasks also involve multiple ROIs, as illustrated in Figure 1.

Current work on attention-based models largely assume that a large-scale training set (*e.g.*, millions of images) is available, making it possible to learn ROI locations automatically. However, in many applications like medical imaging, only a small number of training images are available. Such applications raise two unique challenges for attention-based models: (1) It is usually hard to learn the locations of the ROIs directly from the data. (2) Even if the models manage to find the ROIs given the small number of samples, the models can easily overfit, as demonstrated in Figure 2.

One of our key insights is that by learning the locations of the ROIs in addition to the content inside each ROI, an attention-based model can achieve higher accuracy even with small-scale training

KDD '20, August 22–27, 2020, San Diego, CA

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XX.XXXX/XXXXXXX.XXXXXXX

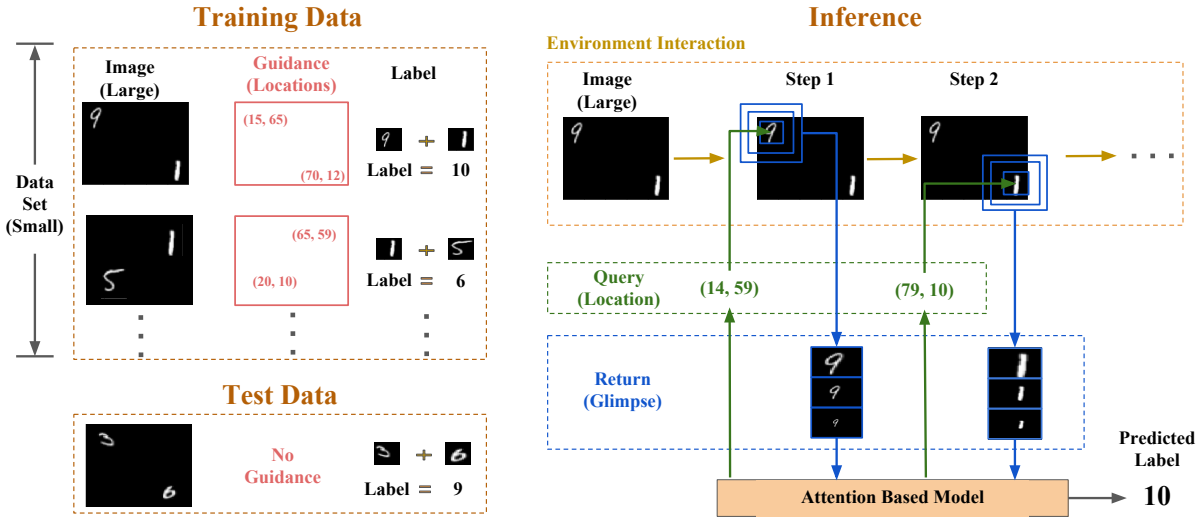


Figure 1: An example of the guided multi-attention classification problem. Each image contains two written digits (ROIs) at varying locations. The label of the image is determined by the sum of the two digits, *e.g.*, the label $10 = (9+1)$. The locations of the digits are provided as guidance to the system in the *small* training set, but are *not* available during inference. An attention-based model moves its visual sensor (controlled by a policy function) over the image and extracts patches (glimpses) to predict the image label.

set. Fortunately, in many applications with a small number of training samples, it is usually possible for human experts to provide the locations of the ROIs, *e.g.*, locations of brain regions. In this paper, we studied a new problem called *guided multi-attention classification*, as shown in Figure 1. The goal of guided multi-attention classification is to train an attention-based model on a small-scale dataset by utilizing the guidance, *i.e.*, the locations of ROIs in each image, to avoid overfitting.

Despite its value and significance, the guided multi-attention classification has not been studied in this context so far. The key research challenges are as follows:

Guidance of Attention: One key problem is how to learn a good policy using the guidance information (*i.e.*, ROIs' locations). Such guidance is *only* available during training which requires careful design to ensure that the model still performs well without it at inference time. Moreover, there can be a large number of possible trajectories covering these ROIs in each training image.

Limited number of samples: Conventional attention-based models usually require a large dataset to train the attention mechanism. With small datasets, the attention-based models can easily overfit by using the locations of ROIs instead of the contents in each region to build a classification model. As shown in Figure 2, to avoid overfitting, the classifier of the attention-based model should avoid using the low-resolution glimpse, *i.e.*, containing the ROI locations, but instead focus on the high-resolution glimpse, *i.e.*, containing the content of each ROI. On the other hand, the "locator" network which determines where the sensor should move next, should use the low-resolution glimpse instead.

In this paper, we propose a model, called Guided Attention Recurrent Network (GARN), for the multi-attention classification problem. Different from existing attention-based methods (see Table 1), GARN utilizes the guidance information for multiple ROIs in each image and works well with small training datasets. We designed a new reward mechanism to utilize both the given ROI locations and the label from each training image. We proposed a novel attention model consisting of two separate RNNs that are trained simultaneously. Empirical studies on three different visual tasks demonstrate that our guided attention approach can effectively boost model performance for multi-attention image classification.

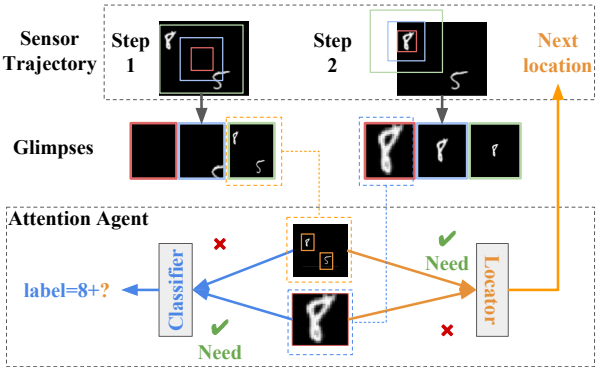


Figure 2: The unique challenge of attention-based classification with only a small number of training samples. A classifier will overfit if it learns to use the locations instead of the contents of ROIs. To prevent overfitting, a classifier should avoid "memorizing" locations in a low-resolution glimpse and focus on the high-resolution glimpse. Meanwhile, a "locator" network should utilize the low-resolution glimpse to determine where to move the sensor next.

2 PROBLEM FORMULATION

In this section, we formally define the multi-attention classification problem. We are given a small set of N training samples $\mathcal{D} = \{(\mathbf{I}_i, \mathcal{R}_i, y_i)\}_{i=1}^{N}$. Here, $\mathbf{I}_i \in \mathbb{R}^{W \times H \times C}$ denotes the i-th image with dimensions $W \times H \times C$ and label $y_i \in \mathcal{L}$. Furthermore, $\mathcal{L}$ represents the label space, *i.e.*, $\{0, 1\}$ for binary classification, and $\{1, \cdots, N_c\}$ for multi-class classification, where N_c is the number of categories.

Table 1: How GARN differs from other attention-based methods. GARN settings are highlighted in red.

Related Work	Base Learner	Supervised Attention	# ROIs	Size of Image	Size of Training Set
Goodfellow et al. [11]	CNN	No	Multiple	Small	Large
Mnih et al. [19]	RAM	No	Single	Large	Large
Ba et al. [3]	RAM	No	Multiple	Large	Large
This Paper (GARN)	RAM	Yes	Multiple	Large	Small

$\mathcal{R}_i = \{\ell_{ij}\}_{j=1}^{n_i}$ is a set of locations of the ROIs in image $\mathbf{I}_i$. Here $\ell_{ij} = (x_{ij}, y_{ij}) \in \mathbb{R}^2$, where $0 \le x_{ij} \le W$ and $0 \le y_{ij} \le H$, indicates the center of the j-th ROI in the i-th image. The label y_i is only determined by the objects/contents within these ROIs.

Region of Interest (ROI): In the multi-attention classification problem, each ROI is a part of the image that contains information pertinent to the label of the image. For instance, in an fMRI image of the human brain, each ROI is one of the brain regions related to a certain neurological disease.

The goal of multi-attention classification is to learn a model $f : \mathbb{R}^{W\times H\times C} \mapsto \mathcal{L}$. Specifically, we are interested in learning an attention-based model, which interacts with a test image $\mathbf{I}$ that iteratively extracts useful information from a test image through multiple steps. In each step, the attention model obtains a glimpse, *i.e.*, patch, $\mathbf{X}_t$ of the image $\mathbf{I}$ around a queried location. The attention-based model contains a policy function for visual attention $\pi(\mathbf{h}_t) = (x_{t+1}, y_{t+1})$. Here, $\mathbf{h}_t$ represents the hidden state of the model at the t-th step of interaction with the image while (x_{t+1}, y_{t+1}) represents the location where the attention mechanism wants to obtain the next glimpse, at step $t+1$, on the test image $\mathbf{I}$.

In this paper, we focus on studying the guided multi-attention classification problem, which has the following properties: (1) training set size (*i.e.*, $|\mathcal{D}|$) is small; (2) image size is large; (3) the class label of each image is related to multiple ROIs – for instance, the sum (label) of multiple digits (ROIs) in an image, or the correlation (label) between the activities of different brain regions (ROIs) in an fMRI scan; and (4) ground-truth locations of ROIs are only provided for a small training set.

3 OUR PROPOSED METHOD: GARN

3.1 RAM Background

Our proposed approach is inspired by the RAM model introduced by Mnih et al. [19]. In RAM, an RL agent interacts with an input image through a sequence of steps. At each step, guided by attention, the agent takes a small patch (or glimpse) of a certain part of the image. The model then updates its internal state with the information provided by the observed glimpse and uses this to decide the next location to focus its attention on. After several steps, the model makes a prediction on the label of the image. Overall, RAM consists of a glimpse network, a core network, a location network, and an action network.

• **Glimpse network** takes a sensor-provided glimpse, $\mathbf{X}_t$, of the input image at time t and encodes it into a "retina-like" glimpse representation, $\mathbf{x}_t$.

• **Core network** is a recurrent neural network. It obtains a new internal state by taking the glimpse representation and combining this with its current internal state. The internal state is a hidden representation which encodes the history of interactions between the agent and the input image.

• **Location network** takes the internal state at time t and outputs a location, ℓ_t, which is where the sensor will be deployed at the next step. Each location, ℓ_t, is assigned a corresponding task-based reward.

• **Action network** takes the internal state at time t as input and generates an action a_t. When RAM is applied to image classification, only the final action, which is used to predict the image label, is utilized. The action earns a reward of 1 if the prediction is correct, otherwise reward is 0.

The t-step agent's interactions with the input image can be denoted as a sequence $S_{1:t} = (\mathbf{x}_1, \ell_1, a_1, \mathbf{x}_2, \ell_2, a_2, \cdots, \mathbf{x}_t)$. RAM learns a function which maps $S_{1:t}$ to a distribution over all possible sensor locations and agent actions. The goal is to learn a policy which determines where to move and what actions to take that maximizes reward.

3.2 Dual RNN Structure

Conventional attention-based methods tend to rely on large-scale datasets for training. However, in many real-world applications, such as medical imaging, the number of available images can be relatively small. For instance, the neuroimaging dataset that Zhang et al. [25] studied had less than a hundred samples. As we illustrated in Figure 2, training attention-based methods on smaller scale training data leads to some unique challenges.

Our key insight is as follows. Instead of trying to learn the locations of the various ROIs as well as the relevant content in each of the ROIs using a single network, like conventional approaches, we divide this process into two connected sub-processes. To make the most of the small number of training images and to fully leverage the power of expert-provided guidance (*e.g.*, locations of ROIs), we design a guided multi-attention model with two complementary RNNs (see Figure 3). The first RNN is used to locate ROIs in the image while the second one is used solely for classification. While the two RNNs take patches of an image at the same position as input, we expect them to remember different things about the input due to a difference in their function.

We now introduce our proposed model architecture. In the subsequent discussions, we will use the same notations as [19]. Let $\text{Linear}(\mathbf{x})$ denote a linear transformation $\mathbf{W}^\top\mathbf{x} + \mathbf{b}$ with weight matrix $\mathbf{W}$ and bias $\mathbf{b}$. On the other hand, $\text{Rect}(\mathbf{x}) = \max(\mathbf{x}, 0)$ denotes the ReLU activation.

3.2.1 RNN for Locating ROI. Our RNN for locating ROIs consists of four parts: glimpse sensor, glimpse network, core network, and location network.

• **Glimpse sensor:** Given an image $\mathbf{I}$, a location $\ell = (i, j)$ and a glimpse scale s, the sensor extracts s square patches $\mathbf{P}_m$, for $m =$

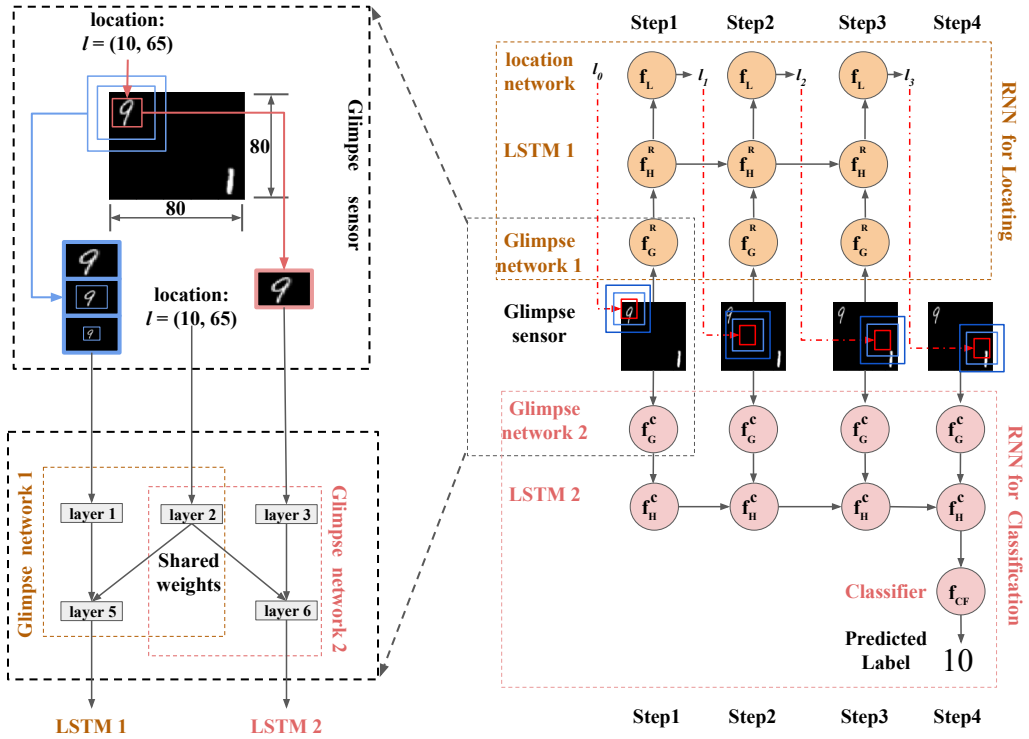


Figure 3: GARN overview. The proposed GARN model consists of two RNNs, one for locating ROIs and the other for classification. The glimpse sensor extracts several image patches of different scales and feeds them to two glimpse networks, f_G^R and f_G^C. f_G^R is the glimpse network of the RNN which locates ROIs while f_G^C belongs to the classification RNN. The glimpses fed to both f_G^R and f_G^C are from the same location given by the network f_L with a potentially different number of glimpse scales.

$1, \cdots, s$, centered at location (i, j). The side of the $(m+1)^{th}$ patch is twice that of the m^{th} patch. All s patches are then scaled to the smallest size, concatenated, and flattened to a vector $\mathbf{x}$.

• **Glimpse network (f_G^R):** As shown in Figure 3, the glimpse network is composed of 3 fully connected (FC) layers: (1) the first FC network encodes the sensor signal $\mathbf{x}$: $\mathbf{x_h} = \text{Rect}(\text{Linear}(\mathbf{x}))$; (2) the second FC network encodes the location of the sensor ℓ: $\ell_\mathbf{h} = \text{Rect}(\text{Linear}(\ell))$; (3) the third FC network encodes the concatenation of $\mathbf{x_h}$ and $\ell_\mathbf{h}$: $\mathbf{g} = \text{Rect}(\text{Linear}(\mathbf{x_h}, \ell_\mathbf{h}))$. The glimpse representation $\mathbf{g}$ is the output of f_G^R.

• **Core network (f_H^R):** Given the glimpse representation $\mathbf{g}_t$ and hidden internal state $\mathbf{h}_t$ at time step t, the core network updates the internal state using the following rule: $f_H(\mathbf{g}_t, \mathbf{h}_t) = \mathbf{h}_{t+1}$. The hidden state $\mathbf{h}_{t+1}$ now encodes the interaction history of the agent up to time t. We use basic LSTM cells to form f_H.

• **Location network (f_L):** At time step t, the next location ℓ_t is stochastically determined by the location network. We assume that ℓ_t is drawn from a 2D Gaussian distribution. The Gaussian distribution's mean vector $\boldsymbol{\mu}$ is outputted by the location network f_L, which is a fully connected network $\boldsymbol{\mu}_t = \text{Tanh}(\text{Linear}(\mathbf{h}_t))$. The covariance matrix is assumed to be fixed and diagonal.

3.2.2 RNN for Classification. This RNN also consists of four parts: glimpse sensor, glimpse network, core network, action network.

• **Glimpse sensor:** It is similar to the glimpse sensor above, and the two sensors look at the same position at each step. However, in this paper, we use a dual-scale sensor for classification while a triple-scale sensor is used for finding ROIs. Intuitively, this is because the classifier only needs the higher resolution glimpses while the "locator" RNN may benefit from the lower resolution glimpse which covers a wider area.

• **Glimpse network (f_G^C):** Similar to f_G^R, f_G^C is also composed of three FC networks with similar functions. The FC network to encode location is shared with f_G^R. However, it does not share weights with f_G^R for the other two FC networks. This is because the glimpse image here has 1 or 2 scales while f_G^R takes an image with 3 scales.

• **Core network (f_H^C):** The same as f_H^R, but their weights are not shared. f_H^C combines the output of f_G^C at the current step with the previous hidden state to obtain a new hidden state.

• **Action network (f_{CF}):** Takes the last hidden state $\mathbf{h}_n^R$ as input and outputs a label prediction. The action network $f_{CF}(\mathbf{h}_n) = \mathbf{a}_p$ is a three-layer fully connected network with ReLU activations for its hidden layers.

3.3 Reward and Training

The interaction between our model and an image (Figure 4) can be denoted by two sequences. The first, $\mathbf{S}_{1:n}^R = \left(\mathbf{x}_1^R, \ell_1, \mathbf{x}_2^R, \ell_2, \cdots, \mathbf{x}_n^R\right)$, is generated by the RNN for finding ROIs while the second, $\mathbf{S}_{1:n}^C = \left(\mathbf{x}_1^C, \ell_1, \mathbf{x}_2^C, \ell_2, \cdots, \mathbf{x}_n^C, \mathbf{y}\right)$, is encoded by the classification RNN. We can view this as a case of Partially Observable Markov Decision

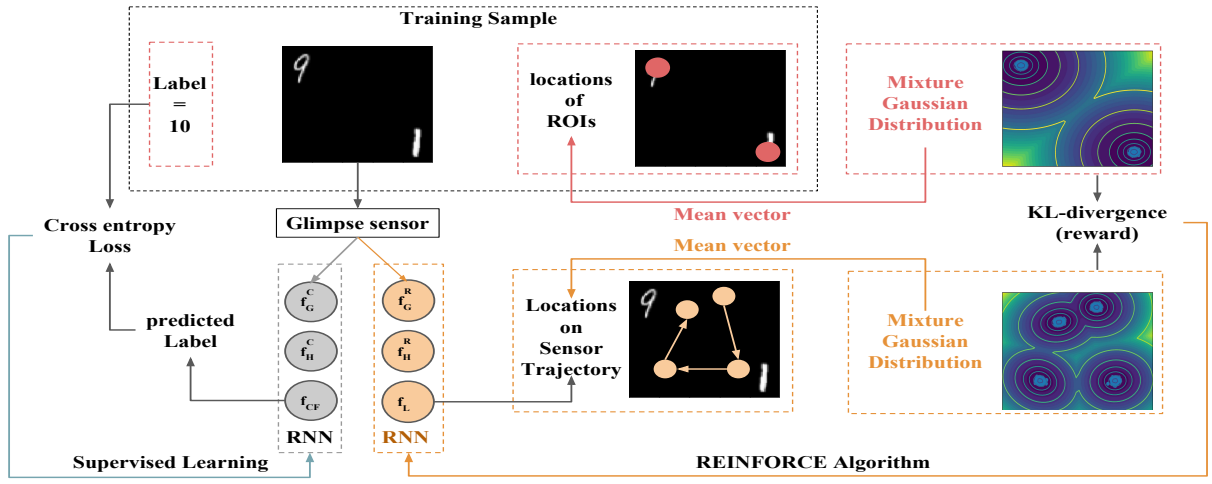


Figure 4: Training overview. The proposed GARN model consists of two RNNs that are trained simultaneously. The RNN for classification is trained using cross-entropy loss. Meanwhile, we trained the RNN for locating ROIs using the KL divergence between two Mixture Gaussian distributions as the reward for the REINFORCE algorithm.

Process [19]. Here, the true state of the environment is static but unknown.

The RNN for classification is trained using cross-entropy loss which is commonly used in supervised learning. Here we mainly discuss the training of the second RNN. We use θ to denote the parameters of the RNN (*i.e.*, f_G^R, f_H^R and f_L). The goal is to learn a policy $\pi(\ell_i|\mathrm{S}^R_{1:i-1};\theta)$ that maximizes the expectation of reward:

$$J(\theta) = \mathbb{E}_{p(\mathrm{S}^R_{1:n};\theta)}\left[\sum_{i=1}^{n} r_{\ell_i|\mathrm{S}^R_{1:i-1}}\right] \tag{1}$$

3.3.1 Reward. We denote $r_{\ell_i|\mathrm{S}^R_{1:i-1}}$ as the reward for the generated location at the i-th step. Originally, in [19], all rewards $r_{\ell_i|\mathrm{S}_{1:i}}$ are set to 1 if the classification is correct, otherwise a uniform reward of 0 is given. However, such assumptions can be problematic when training with only a small number of images, *e.g.*, the model can get high reward by overfitting the training sample without seeing the true ROIs. To mitigate such problem, we designed a reward function based on the ground truth ROI locations:

1. Construct two mixture Gaussian distributions P_1 and P_2, of which the mean vectors correspond to the locations in f_L and the ground truth locations of ROIs, respectively. The standard deviations are hyperparameters, and we used 0.2 by default.

2. The reward in the Equation (1) is the negative of the Kullback-Leibler divergence between P_1 and P_2, which is commonly used for estimating the difference between two distributions.

$$D_{kl}(P_1||P_2) = \sum_i p_1(i) \ln \frac{P_1(i)}{P_2(i)} \tag{2}$$

When P_1 is exactly the same as P_2, the KL divergence is 0. Hence, the closer the locations of the glimpses are to the actual ROIs, the higher the reward.

Table 2: Summary of experimental datasets.

Characteristic \ Task	Comparing two digits	Adding two digits	Brain network classification
Dataset size	2k-20k	2k-20k	2k-8k
Feature size	80×80	80×80	$91 \times 91 \times 10$
Number of classes	2	19	2
Ratio of the dominant class	0.5	0.09	0.5
Number of ROIs	2	2	4

3.3.2 Gradient Calculation. We use REINFORCE algorithm [23] to maximize J [19]. The gradient of J can be approximately by:

$$\nabla_\theta J = \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n} \nabla_\theta \log\left(\pi\left(\ell_i^j|\mathrm{S}^j_{1:i-1};\theta\right)\right) r^j \tag{3}$$

where m denotes the number of episodes and n denotes the total number of steps.

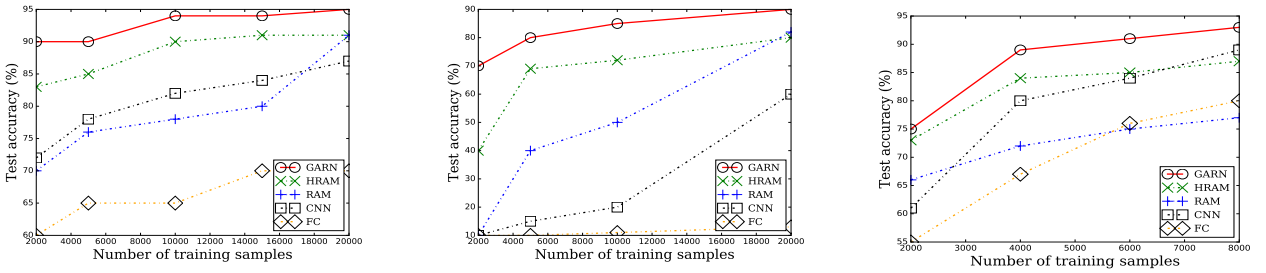
4 EXPERIMENTS

To evaluate our proposed method, GARN, we first conducted experiments on two variants of the MNIST dataset, similar to [3]. We then tested on real-world fMRI data with synthetic regions and labels. More details about each dataset can be found in Table 2.

4.1 Compared Methods

• **Fully Connected Neural Network (FC):** We compare with a fully connected neural network with two hidden layers. The first hidden layer of the FC is composed by 100 neurons, and the second layer by 50. A final classification layer with the appropriate number of outputs is attached at the end.

• **Convolutional Neural Network (CNN):** We designed a CNN that consists of two convolutional layers. Each convolutional layer performs convolution with ReLU activations followed by average pooling. We then connect this to an **FC** network with an architecture that is the same as described above.The convolutional layers have 128 and 256 neurons, respectively. The filter sizes for convolution and pooling are 5×5 and 2×2, respectively.



(a) Comparing two digits (Task 1) (b) Adding two digits (Task 2) (c) Brain network classification (Task 3)

Figure 5: Performance of multi-attention classification on three different tasks. Our proposed guided attention recurrent network (GARN) achieves up to 30% higher accuracy with a small number of training samples, compared to other baseline models. As the number of training samples increases, our GARN model still outperforms others by 5%.

• **Recurrent Attention Model (RAM):** We built a recurrent attention model based on [19] with a sensor crop size of 20×20 and three glimpse scales. In the glimpse network, we use two fully connected networks which each has 128 neurons to encode the cropped image as well as the location vector. Finally, a third FC network with 256 neurons is used to encode the glimpse representation. We use a 256-cell LSTM as our core network. The location network has two layers: the hidden layer has 128 neurons, and an output layer with 2 neurons (using tanh activations) indicating the location coordinates. The action network (classifier) is a fully connected network whose architecture is identical to **FC** described above.

• **Recurrent Attention Model with Hints (HRAM):** To demonstrate the usefulness of guidance information, particularly when training with a small dataset, and also for a fair comparison, we implemented a variant of RAM with hints (*i.e.*, guidance information). Architecture-wise, HRAM is identical to RAM. We trained HRAM with the locations of the ROIs with the standard deviation for calculating KL divergence at 0.2.

• **Guided Attention Recurrent Network (GARN):** This is our proposed model which consists of two RNNs. The RNN for locating ROIs consists of a glimpse network, a core network, and a location network. The RNN for classification consists of another glimpse network, another core network, and an action network (*i.e.*, classifier). Each RNN has the same architecture as their counterpart in the baseline RAM. But the RNN for classification only uses one or two glimpse scales, instead of three, in its glimpse network f_G^C.

4.2 Performance Evaluation

We evaluate the performance of GARN and the other methods on three different classification tasks: *comparing two digits*, *adding two digits*, and *brain network classification*. We introduce each task in more detail in the subsequent discussion. However, before we do so we would first like to highlight two important findings in our performance evaluation:

Importance of Guidance Information: We see in Figures 5a-5c that, across all three tasks, the methods with guidance information (GARN and HRAM) perform substantially better than others when the number of training samples is small. When the number of training samples start to increase, the other methods close the gap in terms of performance but guidance-based methods are still superior.

Importance of Separating Functions: Here, we see in Figures 5a and 5b that when we have sufficient training samples, RAM catches up to HRAM. However, we find that across all three tasks GARN still performs the best. This hints at the importance of using two separate networks that each focus on one of the two important functions: locating ROIs and classification.

4.2.1 Task 1: Comparing Two Digits. In this task, we constructed a new dataset based on the MNIST dataset. For each sample, we randomly selected two MNIST images and embedded them into a black background of size 80 × 80. We randomly sampled two locations around the coordinates (16, 16) and (64, 64) for embedding these two digits. These locations were deliberately chosen to be far-apart in order to force the attention-based methods to learn a policy that has to move for longer distances. We assigned the label 0 to a sample if the digit on the lower right region is larger than the one on the upper left region; otherwise, the label is set to 1.

Figure 5a compares the test accuracies of our proposed GARN and the four baseline models. When there are only 2k training samples, GARN achieves 6% higher accuracy than the best performing baseline HRAM – RAM modified with additional guidance information. This highlights the importance of designing separate RNNs for locating ROIs location and classification. In addition, the improved test accuracy of HRAM over RAM, especially for smaller training datasets, highlights the importance of using ROIs' locations during training, whenever possible.

4.2.2 Task 2: Adding Two Digits. Next we evaluated our proposed model on determining the sum of two digits embedded in an image. We used the same training images from Task 1 and labeled each sample with one out of 19 possible classes. This task is inherently more difficult than the first task due to the larger number of classes and the fact that images with the same label can look very different, *e.g.*, an image consisting of 1 and 9 and an image of 2 and 8 both have the same label.

In Figure 5b, we demonstrate that GARN outperforms all baselines for training datasets with size ranging from 2k to 20k samples. Interestingly, when there are only 2k training samples, all baselines but HRAM perform poorly – similar to random guessing. HRAM increases the test accuracy by 30%, again indicating the usefulness of providing guidance information in settings when we only have limited data. Lastly, GARN achieves more than 70% test accuracy even with 2k training samples and gradually increases its accuracy

to 90% with 20k training samples. Our results indicate that GARN is effective in avoiding overfitting even for relatively complex tasks, with very small number of training samples.

4.2.3 Task 3: Brain Network Classification in fMRI. Lastly, we studied the performance of GARN on a brain network classification problem that reflects settings in the real-world. At a high level, this classification task aims to determine whether a human subject has a certain brain disorder (*e.g.*, concussion, bipolar disorder or Alzheimer disease) from fMRI data. An fMRI sample is a 4D image. Essentially, it is a series of 3D brain images captured over time. From a given fMRI sample, we can construct a weighted graph called a functional brain network with nodes in the graph denoting regions and time-series correlations between regions being the weighted edges. Such correlations are calculated from associated time sequences and reflect the functional interactions between brain regions [7]. In this work, we used regions in the Default Mode Network (DMN), one of the most prominent function networks [1]. We designed a classification taks which requires understanding of the relationships between different regions in DMN. Figure 6 summarizes the steps in constructing the dataset.

In more details, we constructed a synthetic brain network dataset from real-world fRMI data with 31 samples following these steps:

(1) We normalize the brain shape of all subjects by aligning them to the MNI152 standard brain template [2]. This allows us to align all the regions from different fMRI images and helps us identify brain ROIs.
(2) For each raw fMRI image, we carefully select six regions of the DMN. These regions are: left/right posterior cingulate gyrus, left/right angular gyrus, and left/right Medial frontal gyrus [22]. We further combine the regions that are visually close to each other, *e.g.*, the left/right posterior cingulate gyrus, and the left/right Medial frontal gyrus.
(3) To ensure all four DMN regions are included, we extracted a 3D slice with shape = [width = 91, height = 91, time length = 10] at the position $z = 51$ from each fMRI image. This gives us a total of 31 fMRI images which we used as a basis to construct a larger synthetic dataset. We used two complementary approaches (Figure 6-2), *i.e.*, associating each fMRI image with randomly generated time sequences and changing the DMN locations by randomly scaling each fMRI image.
(4) To determine the label for each new fMRI image, we first built a simple brain network that is a complete graph of four DMN locations. We then calculated the Pearson correlation between each pair of DMN locations based on their time sequences. An fMRI image is labeled as "normal" if all pairwise correlations are higher than 0.6, otherwise it is labeled as "abnormal".

We can see from Figure 5c that our proposed GARN significantly outperforms all baselines by up to 2%-20% accuracy, even with a small number of training samples. HRAM achieves about 8% higher accuracy compared to RAM, suggesting the usefulness of utilizing ROIs locations during training. Lastly, neither the CNN nor the FC models work well with small training dataset.

[1] https://en.wikipedia.org/wiki/Default_mode_network
[2] https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases

4.3 Discussion on Parameters

We evaluated two important hyperparamaters, *i.e.*, the number of glimpses and the number of sensor scales.

The number of glimpses represents how many chances we give the model to move the sensor around. More glimpses equals a longer sensor trajectory which typically corresponds to a higher likelihood of gathering more information from the image. In Figure 7, we compared the test accuracies of models given different numbers of glimpses. For tasks one and two which only contain two ROIs, we set the glimpse number to be four and eight, respectively. For task three, we set the glimpse number to be five, ten, and twenty, respectively. The choices of glimpse numbers are based on the number of ROIs to increase the likelihood of capturing ROIs with stochastically generated locations. In Figure 7a and Figure 7b, we can see that GARN achieves higher accuracies with eight glimpses than four glimpses. The accuracy gap decreases as the training samples increases. This is likely because the four-glimpse agent has fewer chances of hitting all the ROIs. Figure 7c shows the impact of different number of glimpses on brain classification task. Given that there are four ROIs in the Default Mode Network, the minimal required number of glimpses is higher than the first two tasks. Having access to more training samples can alleviate the need for more glimpses per sample, as indicated by the shrinking accuracy gaps between ten and twenty glimpses at 8k training samples. Our results suggest that our GARN can effectively avoids overfitting on smaller datasets.

Next we discuss the impact of the number of sensor scales on test accuracy. Recall that our GARN uses two glimpse networks, f_G^R and f_G^C, to locate ROIs and for classification. Each glimpse network can be configured with a different number of sensor scales for each glimpse. We used three scales for f_G^R, similar to the original RAM. We vary the number of sensor scales from one to three for f_G^C which is the agent for classification as demonstrated in Figure 8a.

In Figure 8b and 8c, we compared the test accuracies for different number of sensor scales. Our results show that for both tasks, using fewer scales under smaller training samples achieve higher test accuracies. This suggests that using more and larger scales may lead to overfitting especially when the training datasets are small. One potential reason is that larger scale contains information, *e.g.*, black background, that is not useful for classification. However, such information can be useful for locating ROIs. This suggests that it is useful to separately configure the number of scales for locating ROIs and classification, as we did in GARN by designing two separate RNNs.

5 RELATED WORK

To the best of our knowledge, this work is the first to address the problem of guided multi-attention classification.

Image classification and object recognition: Image classification has become a widely studied topic. Over the past decade, deep neural networks such as CNNs have achieved significant improvement in image classification accuracy [13]. However, these CNNs often incur a disproportionately high computation cost to detect a

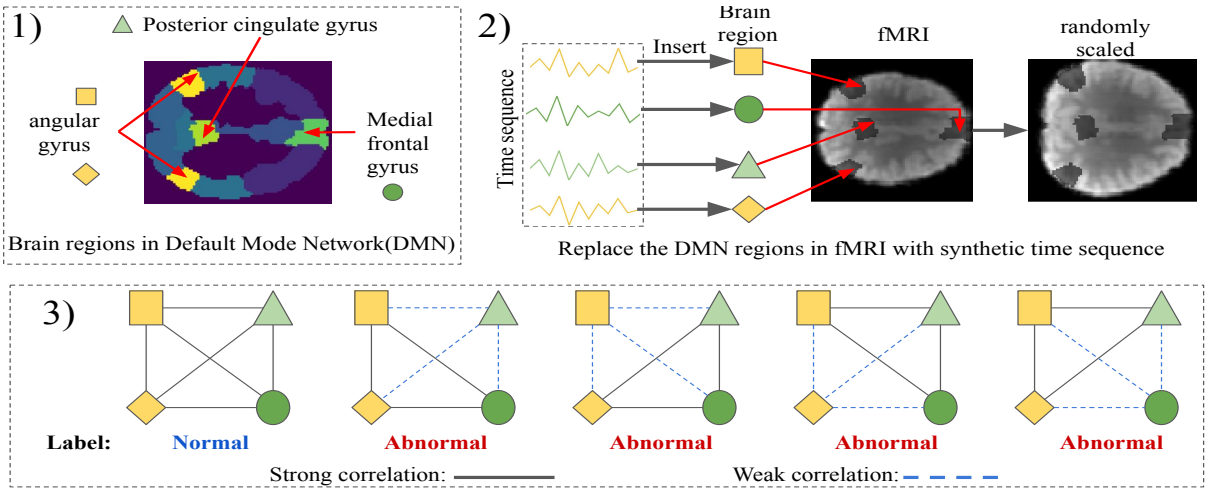
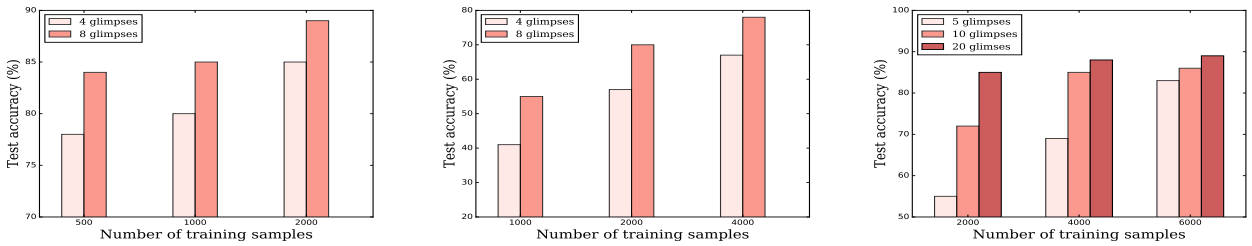
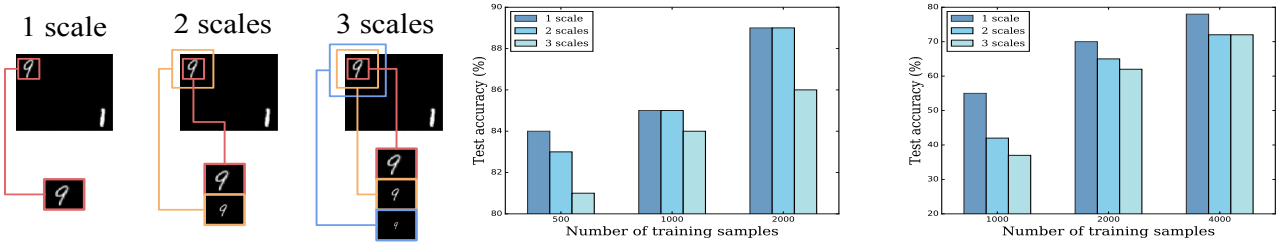


Figure 6: The brain network classification task on fMRI data.



(a) The task of comparing two digits **(b) The task of adding two digits** **(c) The task of classification on fMRI**

Figure 7: Performance of GARN with different number of glimpses. The number of glimpses heavily depends on the number of ROIs. More glimpses help avoid overfitting, but the benefits decrease as the number of training samples increase.



(a) number of scales **(b) The task of comparing two digits** **(c) The task of adding two digits**

Figure 8: Performance of GARN with different number of sensor scales. Having smaller number of scales for the classification RNN helps to avoid overfitting with fewer training samples. This also indicates the need for designing two seperate RNNs in multi-attention classification problem.

small object in a large image. A number of works [1, 10, 11] have attempted to address this problem of high computational cost, but in a non end-to-end way. Others [2, 8, 9], on the other hand, have formulated the task of object detection as a decision task, similar to our work.

Classification on fMRI data: The task of classifying fMRI data can be formulated as a special case of multi-object image classification. Most recent work analyzing fMRI study one or more of the following related sub-tasks: brain region detection [17, 26], brain network discovery, and classification [6, 28]. However, neuroimaging datasets are inherently quite challenging to work with due to their high noise, their high dimensionality, and small sample sizes. It was not until very recently that researchers started to propose end-to-end solutions, such as CNN based methods [20] which solve both brain network discovery and classification coherently [15]. Different from existing work, we use a guided attention-based model which can locate brain regions and do classification as well, without requiring additional information such as time sequences from ROIs as input [15].

Attention model: Recently, researchers have begun to explore attention-based deep learning models for visual tasks [3, 9, 14, 21] and natural language processing [4, 24]. Specifically, Mnih et al. [19] proposed the recurrent attention model (RAM) to tackle the issue of high computation complexity when dealing with large images.

Other work based on RAM have also tackled the problems of multi-object recognition and depth-based person identification [3, 12]. Most recently, Tariang [5] proposed a recurrent attention model to classify natural images and computer generated images. The structure and training method are similar with [3, 19], while it uses a CNN to implement its glimpse network. Meanwhile, Zhao [27] combined a recurrent convolutional network with recurrent attention for pedestrian attribute recognition, which uses a soft attention mechanism instead of the hard attention used by RAM. Another recent study leveraging the soft attention mechanism is [30], which uses recurrent attention residual modules to refine the feature maps learned by convolutional layers. In the areas of person identification, sequence generation, image generation, some other works [16, 29] are also utilize both attentional processing as well as RNNs.

6 CONCLUSION

In this paper, we first formulated the Guided Multi-Attention Classification problem. We then proposed the use of a guided attention recurrent network (GARN) to solve the problem. Our proposed method addresses the challenges of training with only a small number of samples by effectively leveraging the guidance information in the form of ROI locations. Specifically, GARN learns to identify the locations of ROIs and to perform classifications using two separate RNNs. We performed extensive evaluations on three multi-attention classification tasks. Our results across all three tasks demonstrated that GARN outperforms all baseline models. In particular, when the training set size is limited, we observed up to a 30% increase in performance.

REFERENCES

[1] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. 2010. What is an object?. In *Proc. 2010 IEEE Conf. Computer Vision and Pattern Recognition (CVPR'10)*. 73–80.

[2] Bogdan Alexe, Nicolas Heess, Yee W Teh, and Vittorio Ferrari. 2012. Searching for objects driven by context. In *Advances in Neural Information Processing Systems 25 (NeurIPS'12)*. 881–889.

[3] Jimmy Ba, Volodymyr Mnih, and Koray Kavukcuoglu. 2015. Multiple object recognition with visual attention. In *Proc. 3rd Int. Conf. Learning Representations (ICLR'15)*.

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proc. 3rd Int. Conf. Learning Representations (ICLR'15)*.

[5] Diangarti Bhalang Tarianga, Prithviraj Senguptab, Aniket Roy, Rajat Subhra Chakraborty, and Ruchira Naskar. 2019. Classification of Computer Generated and Natural Images based on Efficient Deep Convolutional Recurrent Attention Model. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[6] Tom Brosch, Roger Tam, AlzheimerâĂŹs Disease Neuroimaging Initiative, et al. 2013. Manifold learning of brain MRIs by deep learning. In *Proc. 16th Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI'13)*. 633–640.

[7] Ed Bullmore and Olaf Sporns. 2009. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience* 10, 3 (2009), 186–198.

[8] Nicholas J Butko and Javier R Movellan. 2009. Optimal scanning for faster object detection. In *Proc. 2009 IEEE Conf. Computer Vision and Pattern Recognition (CVPR'09)*. 2751–2758.

[9] Misha Denil, Loris Bazzani, Hugo Larochelle, and Nando de Freitas. 2012. Learning where to attend with deep architectures for image tracking. *Neural Computation* 24, 8 (2012), 2151–2184.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. 2014 IEEE Conf. Computer Vision and Pattern Recognition (CVPR'14)*. 580–587.

[11] Ian J Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. 2014. Multi-digit number recognition from street view imagery using deep convolutional neural networks. In *Proc. 2nd Int. Conf. Learning Representations (ICLR'14)*.

[12] Albert Haque, Alexandre Alahi, and Li Fei-Fei. 2016. Recurrent attention models for depth-based person identification. In *Proc. 2016 IEEE Conf. Computer Vision and Pattern Recognition (CVPR'16)*. 1229–1238.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NeurIPS'12)*. 1097–1105.

[14] Hugo Larochelle and Geoffrey E Hinton. 2010. Learning to combine foveal glimpses with a third-order Boltzmann machine. In *Advances in Neural Information Processing Systems 23 (NeurIPS'10)*. 1243–1251.

[15] John Boaz Lee, Xiangnan Kong, Yihan Bao, and Constance Moore. 2017. Identifying Deep Contrasting Networks from Time Series Data: Application to Brain Network Analysis. In *Proc. 17th SIAM Int. Conf. Data Mining (SDM'17)*. 543–551.

[16] Jun Liu, Gang Wang, Ping Hu, Ling-Yu Duan, and Alex C Kot. 2017. Global context-aware attention LSTM networks for 3D action recognition. In *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR'17)*.

[17] Arthur Mensch, Gaël Varoquaux, and Bertrand Thirion. 2016. Compressed online dictionary learning for fast resting-state fMRI decomposition. In *Proc. 13th IEEE Int. Symposium on Biomedical Imaging (ISBI'16)*. 1282–1285.

[18] Simon Mezgec and Barbara Koroušić Seljak. 2017. NutriNet: A Deep Learning Food and Drink Image Recognition System for Dietary Assessment. *Nutrients* 9, 7 (2017), 657.

[19] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. 2014. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems 27 (NeurIPS'14)*. 2204–2212.

[20] Dong Nie, Han Zhang, Ehsan Adeli, Luyan Liu, and Dinggang Shen. 2016. 3D deep learning for multi-modal imaging-guided survival time prediction of brain tumor patients. In *Proc. 19th Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI'16)*. 212–220.

[21] Charlie Tang, Nitish Srivastava, and Russ R Salakhutdinov. 2014. Learning generative models with visual attention. In *Advances in Neural Information Processing Systems 27 (NeurIPS'14)*. 1808–1816.

[22] Nathalie Tzourio-Mazoyer, Brigitte Landeau, Dimitri Papathanassiou, Fabrice Crivello, Olivier Etard, Nicolas Delcroix, Bernard Mazoyer, and Marc Joliot. 2002. Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. *Neuroimage* 15, 1 (2002), 273–289.

[23] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8 (1992), 229–256.

[24] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *Proc. 32nd Int. Conf. Machine Learning (ICML'15)*. 2048–2057.

[25] Jingyuan Zhang, Bokai Cao, Sihong Xie, Chun-Ta Lu, Philip S. Yu, and Ann B. Ragin. 2016. Identifying Connectivity Patterns for Brain Diseases via Multi-side-view Guided Deep Architectures. In *Proc. 16th SIAM Int. Conf. Data Mining (SDM'16)*. 36–44.

[26] Yudong Zhang, Zhengchao Dong, Preetha Phillips, Shuihua Wang, Genlin Ji, Jiquan Yang, and Ti-Fei Yuan. 2015. Detection of subjects and brain regions related to Alzheimer's disease using 3D MRI scans based on eigenbrain and machine learning. *Frontiers in Computational Neuroscience* 9 (2015), 66.

[27] Xin Zhao, Liufang Sang, Guiguang Ding, Jungong Han, Na Di, and Chenggang Yan. 2019. Recurrent attention model for pedestrian attribute recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 9275–9282.

[28] Luping Zhou, Lei Wang, Lingqiao Liu, Philip Ogunbona, and Dinggang Shen. 2013. Discriminative brain effective connectivity analysis for Alzheimer's disease: a kernel learning approach upon sparse Gaussian Bayesian network. In *Proc. 2013 IEEE Conf. Computer Vision and Pattern Recognition (CVPR'13)*. 2243–2250.

[29] Zhen Zhou, Yan Huang, Wei Wang, Liang Wang, and Tieniu Tan. 2017. See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification. In *Proc. 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR'17)*. 6776–6785.

[30] Lei Zhu, Zijun Deng, Xiaowei Hu, Chi-Wing Fu, Xuemiao Xu, Jing Qin, and Pheng-Ann Heng. 2018. Bidirectional feature pyramid network with recurrent attention residual modules for shadow detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 121–136.