Attend and Imagine: Multi-Label Image Classification With Visual Attention and Recurrent Neural Networks

Fan Lyu, Qi Wu, Fuyuan Hu, Qingyao Wu, and Mingkui Tan

***Abstract*—Real images often have multiple labels, i.e., each image is associated with multiple objects or attributes. Compared to single-label image classification, the multilabel classification problem is much more challenging due to several issues. At first, multiple objects can be anywhere in the image. Second, the importance of different regions in an image is different, and the regions of interest in a multilabel image can be very different from another one. Finally, multiple labels of an image can have label dependencies due to complex image structures. To address these challenges, in this paper, we propose to predict the labels sequentially by applying the recurrent neural networks (RNNs), which are used to encode the label dependencies. When predicting a specific label, we introduce a dynamic attention mechanism to enable the model to focus on only regions of interest in the image. Two benchmark datasets (i.e., Pascal VOC and MS-COCO) are adopted to demonstrate the effectiveness of our work. Moreover, we construct a new dataset, which includes many semantic dependent labels in each image, to verify the effectiveness of our model. Experimental results show that our method outperforms several state-of-the-arts, especially when predicting some semantic relative labels.**

***Index Terms*—Multi-label classification, visual attention, deep learning, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN).**

Manuscript received April 13, 2018; revised August 27, 2018; accepted December 21, 2018. Date of publication January 23, 2019; date of current version July 19, 2019. This work was supported in part by the Natural Science Foundation of China (61876121, 61472267, 61877038, 61602185), in part by the Primary Research & Development Plan of Jiangsu Province (BE2017663), in part by the Fundamental Research Funds for the Central Universities D2172480, and in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams 2017ZT07X183. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Elisa Ricci. *(Fan Lyu and Qi Wu contributed equally to this work.) (Corresponding authors: Fuyuan Hu and Mingkui Tan.)*

F. Lyu was with the Suzhou University of Science and Technology, Suzhou 215009, China. He is now with the College of Intelligence and Computing, Tianjin University, Tianjin 300000, China (e-mail: fanlyu@tju.edu.cn).

Qi Wu is with the Australian Centre for Visual Technologies, School of Computer Science, The University of Adelaide, Adelaide, SA 5005, Australia (e-mail: qi.wu01@adelaide.edu.au).

F. Hu is with the School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215010, China (e-mail: fuyuanhu@mail.usts.edu.cn).

Qingyao Wu and M. Tan are with the School of Software Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: qyw@scut.edu.cn; mingkuitan@scut.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2019.2894964

I. Introduction

During the past few years, thanks to the development of learning theory [1]–[3] and the availability of large-scale datasets like ImageNet [4], significant progress has been made in the task of single-label image classification [5]–[10], where each image is associated with one label only. However, in the real world, what we usually see are multi-label images, where each image contains more than one objects or multiple semantic tags. For example, images from the social media and news usually contain fruitful semantic tags.

In this paper, we construct a new image dataset, called BNID (Breaking News Image Dataset), whose images are all from breaking news. In Fig. 1(a), we show some examples of multi-label images from BNID, whilst in Fig. 1(b), we show some examples of single-label image from ImageNet. Obviously, from Fig. 1(a), the multi-label images are more complicated than single-label images. Specifically, for the single-label images, there is only one primary object (namely one label), which, however, is not possible in many practical applications. Moreover, the object often covers the whole image or lies around the center.

On the contrary, for multi-label images, following issues are observed. At first, they usually contain multiple objects, each of which is associated with one or more labels. Secondly, the multiple objects do not have an explicitly aligned foreground, and the multiple objects in an image can lie anywhere. This implies that the importance of different regions of an image is different, and the regions of interest in one multi-label image can be very different from another one. Third, for a specific kind of objects (such as person), they can have different sizes in different images, which further increases the difficulties of multi-label image learning. At last, for the multiple targets (or labels) in an image, they usually have some dependencies among each other [11]–[13]. For example, a ship will likely appear along with water. This kind of label correlation is not necessarily considered in single-label classification. As a result, directly applying single-label methods on the multi-label image classification is not helpful [14]–[17].

A lot of efforts have been made to handle the multi-label image classification problem [18]. Traditional methods can be divided into two categories—problem transformation based methods [19]–[26] and algorithm adaptation based methods [27]–[32]. Recently, Convolutional Neural Networks (CNNs) [5] based approaches have dominated this area. For example, Gong

(a) Examples of multi-label images from BNID

(b) Examples of single-label images from ImageNet

Fig. 1. The comparison between single-label images and multi-label images. (a) Multi-label image has more than one objects in one image. (b) The only object in the single-label image is always full of the image and aligned center. The objects in it can be anywhere, random size and may have some correlations.

et al. [14] compare several multi-label loss functions and find that a significant performance gain could be obtained by using a top-k ranking objective function. Sharif *et al.* [15] use CNN to extract features and put them in an SVM to classify the relevant labels. Wei *et al.* [17] extract some hypotheses from source image and train them using a pre-trained CNN model.

The aforementioned approaches, however, do not consider the label dependencies, thus they are difficult to model the correlations among labels or objects. Recently, Recurrent Neural Networks (RNNs) have been proved to be powerful to model the sequence correlations in machine translation [33]–[35]. Wang *et al.* [36] propose a unified CNN-RNN framework, which utilizes RNN to learn both the semantic redundancy and the co-occurrence dependency among labels. Jin *et al.* [37] take a similar idea but pay more attention to the annotation length. Liu *et al.* [38] pre-train a CNN model and a RNN model separately. In general, these methods predict the labels sequentially. When predicting each label at each time step, however, they often use the same global image as the input. That is to say, the whole image is used as input when predicting a specific label. However, for practical multi-label images, they are often very complicated and one particular label is often related to one region of the image. Therefore, directly using the whole image as the input to each label may hamper the performance. For example, as shown in Fig. 3, the multi-label images have multiple concerned regions, but most of them are relevant to one label only. Therefore, when predicting one specific label in a step of RNN, it is useless to consider other irrelevant regions.

The attention mechanism has been used in the machine translation, image captioning and visual question answering [33], [39]–[42]. In particular, attention mechanism can be viewed as a feature selection strategy [43], [44] which can be used to foucs on the useful information and ignore the redundancies. In multi-label classification, we can regard the label as a sequence, however, which is denser than the sentence in real context (for a sentence, it usually has inherent syntactic structure and a lot of qualifiers). To this end, in this paper, we propose an attention based method that allows the model to attend on some particular regions only when predicting the label at each time step. Unlike the attention based method in machine translation and image captioning, our model considers a continuous transformation of the attentive area at each step. We note in [45], Zhu *et al.* proposed a similar method called Spatial Regularization Network, but they care more about how to regularize the spatial information with the weighted attention. We follow the encoder-decoder pattern, which is popular in machine translation and image captioning. In the encoder, we extract certain features from a lower convolutional layer of a pre-trained CNN. In the decoder, we first initialize LSTM with the extracted features. Then, at each time step, we compute a dynamic context by using the attention mechanism and inject the context into LSTM. Moreover, to prevent LSTM from forgetting the whole image in the propagation, we also remind our model every step by merging a higher level feature from the image with LSTM. We evaluate our model on two popular multi-label datasets, i.e., Pascal VOC and MS-COCO, and our own dataset BNID. We compare our model with several state-of-the-art multi-label classification methods. Experimental results show that our method achieves better performance. We also visualize our attended regions and the results show that the model can learn to focus on the related parts of the images during predicting specific labels.

In the rest of paper, we first discuss the related works in Section II. In Section III, we will illustrate the details of our model. Then, we will show our experimental setting and compare our method with some other current start-of-the-art methods in Section IV. Lastly, we will give a brief conclusion about our work in Section V.

II. Related Work

A lot of efforts have been made in multi-label classification before the popularity of CNN. The traditional methods can be regard as the problem transformation and algorithm adaptation categories [18], [46]. The common problem transformation methods, which transform multi-label problem into multiple single-label problems, contain Binary Relevance (BR) [19], Classifier Chains (CC) [20], [21], Calibrated Label Ranking [22], Label Powerset (LP) [23], [24] and their variants. While the algorithm adaptation methods use some traditional algorithms to adapt data such as Multi-Label k-Nearest Neighbor (ML-kNN) [27], Multi-Label Decision Tree (ML-DT) [28], Ranking Support Vector Machine (Rank-SVM) [29], Collective Multi-Label Classifier (CML) [30] and their variants. Recently, the approaches based on Neural Network have made great progress on this problem. In the following part, we will focus on reviewing some of these works.

A. CNN Multi-Label

CNN can be directly applied on the multi-label image classification problem, such as [14]–[17], [26]. We call this kind of methods CNN Multi-label. Gong *et al.* [14] combine top-k ranking objectives with a CNN architecture to tackle this problem. By defining a weight function for pair-wised ranking labels, they minimize the loss function so that positive labels are ranked higher than negative ones.

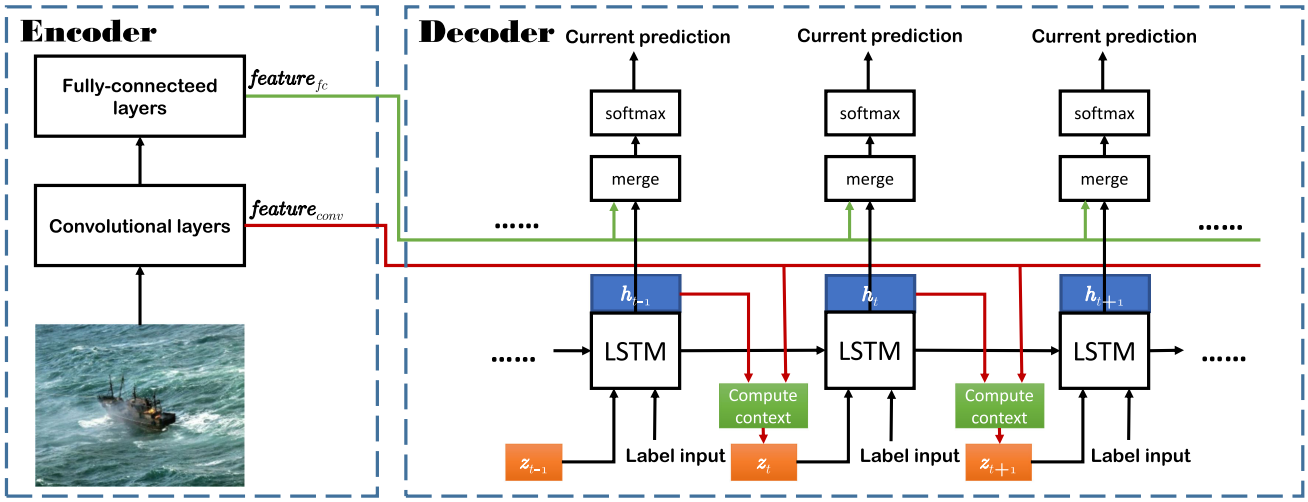


Fig. 2. The schematic of the proposed architecture. The encoder CNN extracts $\boldsymbol{feature}_{conv}$ and $\boldsymbol{feature}_{fc}$ from the final convolutional layer and the final fully-connected layer, respectively. For each step of LSTM, we use the hidden state and $\boldsymbol{feature}_{conv}$ to compute the dynamic attentive context $\boldsymbol{z}_t$, which is described in Section III-C detailedly. We also merge the image embedding computed from $\boldsymbol{feature}_{fc}$ with the output of LSTM every step.

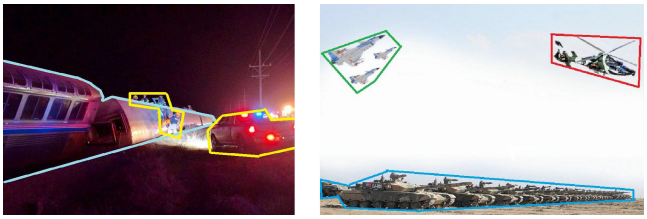


Fig. 3. Examples of multi-label images with multiple regions.

Wei *et al.* [17] provide a regional solution that allows predicting labels independently at the regional level. They use Binarized Normed Gradients (BING) [47] to generate object proposals and further send them into a CNN to compute multi-class scores. A max-pooling operation is applied to obtain final classification results. Yang *et al.* [48] propose a two-stream fully convolutional network that utilizes the readily available privileged bags, instead of hard-to-obtain privileged instances, making the system more general and practical in real world applications. These region-based methods seek to fine-tune the pre-trained model that is trained on other large-scale datasets. Then, they extract many regions from the image and feed them into the previously trained model. The output of each region is filtered by a max-pooling operation.

To extract regions from images, previous studies always use the existing region proposal extraction algorithms and their pre-extracted regions, for example BING [47], Objectness Measure [49], Multiscale Combinatorial Grouping (MCG) [50], etc. By bringing extra regional information of images into the model, the region proposal based methods have improved the vanilla CNN Multi-label to some extent. However, they have several drawbacks. First, they often spend ten more times of computational time to process each image due to the consideration of the extra regional information. Moreover, the existing region proposal based algorithms are also difficult to train, which makes it consume more time to extract regions from one image. This kind of methods forces the model to see some specified rectangular parts of the image, which is close to our approach. In fact, our method is significantly different from it. We list several principal different aspects as follows: 1) Our method has a dynamic attention mechanism, which can be learned by the model itself, but the region-based methods require the external object proposal algorithm that is learned separately; 2) The shape of the attention learned by our method can be various regarding to the label, while it must be a rectangle in the region-based methods; 3) Our method can be trained end-to-end.

B. CNN-RNN

Directly applying CNN to the multi-label classification can only make use of the stereotyped noisy image information, which is much insufficient. The correlations among labels, an important type of information, are ignored by the above methods. Recently, Wang *et al.* [36] utilizes CNN to extract image features and RNN to model the correlations among labels. These methods can be categorized as the CNN-RNN Multi-label method. The CNN-RNN based method benefits by the development of image captioning [39], [51]–[53]. It follows the *encoder-decoder* design pattern, which is from the machine translation [33]–[35]. A CNN module is always set as the encoder, which encodes an image into a fix length vector. Then, the vector is fed into an RNN module and decoded into a sequence of labels. In [36], Wang *et al.* add together the image embedding and the output of the LSTM every step, then passes the combined vector to the final fully connected layer and predict the current label. Liu *et al.* [38] regularize the CNN by the ground truth semantic concepts, then use the prediction to set the LSTM initial states. Jin *et al.* [37] propose an architecture similar to Wang's and argue that the label order really matters the performance.

Recently, some researchers apply the attention mechanism to force RNN to see the concerned parts of the original data in the machine translation and the image captioning. Bahdanau

et al. [33] propose a model to search a set of possible positions while generating the target word. Xu *et al.* [39] take hard and soft attention-based methods to generate image descriptions. You *et al.* [40] run a set of attribute detectors to get a list of visual attributes and fuse them into the RNN hidden state.

In CNN-RNN based methods, the labels can be treated as a denser sequence than a sentence because they has no syntactic structure and no unnecessary qualifiers. In this paper, we are on top of this premise and design a dynamic attention mechanism on multi-label classification, which is much different than the vanilla CNN-RNN. The attention mechanism does improve the common CNN-RNN design because it drives the model to focus some important areas and relates them with every predicated labels. The proposed method will be shown in Section III.

III. Method

A. Problem Definition

Given a set of labeled images $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L\}$, where L is the length of the set, multi-label learning seeks to learn to predict all the possible labels for each image in $\mathcal{X}$. For the i-th image $\boldsymbol{x}_i$, we denote the corresponding labels as $\boldsymbol{Y}_i = \{\boldsymbol{y}_{i1}, \boldsymbol{y}_{i2}, \ldots, \boldsymbol{y}_{iC}\}$, where $\boldsymbol{y}_{ij} \in \mathbb{R}^D$ is a one-hot encoding vector (binary index vector in a vocabulary), C is the number of possible labels (or objects) in an image and D is the length of the label dictionary. Here, the number D is fixed and known in advance, whilst the number C is unknown and different when predicting different images. In the traditional methods (such as [14]–[17]), they often determine the number of labels (i.e. the number C) based on some threshold or by picking the top-k ranking labels, where k is a user predefined number. On the contrary, we will propose a new scheme which seeks to inference the number C automatically.

B. Overall Architecture

To predict the multiple labels of one image sequentially, we assume the labels follow some kinds of order. Thus, the labels can be ordered as a specific sequence

$$\tilde{\boldsymbol{Y}}_i = \{\tilde{\boldsymbol{y}}_{i1}, \tilde{\boldsymbol{y}}_{i2}, \ldots, \tilde{\boldsymbol{y}}_{iC}\},\ \tilde{\boldsymbol{y}}_{ij} \in \mathbb{R}^D. \tag{1}$$

Here, the sequence order can be pre-defined by different rules, such as the frequency rule and the dependency rule, which is described detailedly in Section IV-A.

Our overall architecture follows the *encoder-decoder* design pattern, which is able to learn a transformation from one representation to another. In our framework, we use two kinds of features. The first kind features, denoted by $\boldsymbol{feature}_{conv} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_N\}$, are extracted from the final convolutional layer of a pre-trained CNN, where N denotes the number of regions. Specifically, the features in $\boldsymbol{feature}_{conv}$ are constructed by a set of convolutional features extracted from local regions corresponding to different parts of the image. In this way, it helps the model focus on specific parts of an image and obtain a correspondence between the feature vectors and the original image patches. That means more structural and positional information are preserved. The other one is an 4096-D vector $\boldsymbol{feature}_{fc}$ extracted from the last fully-connected layer, and it contains the higher level information.

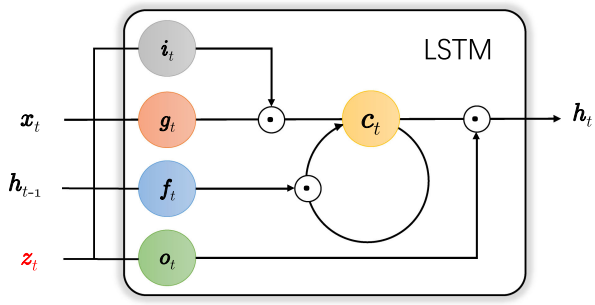


Fig. 4. An architecture of LSTM with a dynamic attentive context $\mathbf{z}_t$.

For the RNN decoder, we use the Long Short-Term Memory (LSTM) [54], a particular form of RNN. LSTM adds three extra gates to an RNN neuron: the input gate controls of the input data it should read; the forget gate controls of the previous state it should forget; the output gate controls of the current state it should output. We join a trainable attention $\boldsymbol{z}_t$ into LSTM and we will explain how to obtain that in the next section. Therefore following [55], the forward passing at step t can be defined as

$$\boldsymbol{g}_t = \tanh(\boldsymbol{W}_{xc}\boldsymbol{y}_t + \boldsymbol{W}_{hc}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{zc}\boldsymbol{z}_t + \boldsymbol{b}_c), \tag{2}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{xi}\boldsymbol{y}_t + \boldsymbol{W}_{hi}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{zi}\boldsymbol{z}_t + \boldsymbol{b}_i), \tag{3}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{xf}\boldsymbol{y}_t + \boldsymbol{W}_{hf}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{zf}\boldsymbol{z}_t + \boldsymbol{b}_f), \tag{4}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{xo}\boldsymbol{y}_t + \boldsymbol{W}_{ho}\boldsymbol{h}_{t-1} + \boldsymbol{W}_{zo}\boldsymbol{z}_t + \boldsymbol{b}_o), \tag{5}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{g}_t, \tag{6}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t), \tag{7}$$

where $\boldsymbol{W}$ are trainable weights and $\boldsymbol{y}_t$ is the input label embedding in step t. The whole architecture is shown in Fig. 2. In our architecture, We initialize the LSTM with the mean of the feature maps:

$$\boldsymbol{c}_0 = I_c\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{a}_i\right), \tag{8}$$

$$\boldsymbol{h}_0 = I_h\left(\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{a}_i\right), \tag{9}$$

where the initial function I can be denoted as a simple Multi-Layer Perceptron (MLP)

$$I(\boldsymbol{x}) = \tanh(\boldsymbol{W}_I\boldsymbol{x} + \boldsymbol{b}_I). \tag{10}$$

This initialization of parameters is quite important. Yang *et al.* [56] consider that the attention mechanism lacks global modeling abilities in common sequential learning. Initializing the memory cell and the hidden state in this way helps LSTM learn the whole non-attentive feature maps and a glance to the original image.

Note that the image features may be forgot if the LSTM sequences are too long. To avoid this issue, we remind it at every step by merging the image embedding from $\boldsymbol{feature}_{fc}$ with the output of the last LSTM.

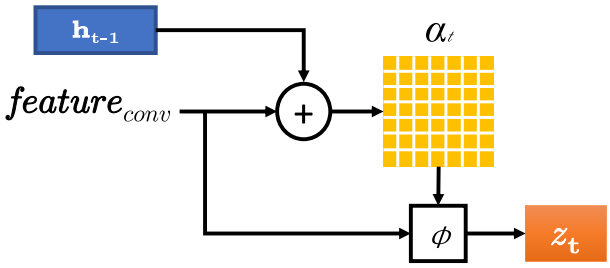


Fig. 5. The computation of $\boldsymbol{z}_t$, which is represented by the "compute context" block (the green one) in Fig. 2.

C. Visual Attention Model

Our visual attention model uses a dynamic context $\boldsymbol{z}_t$ to represent the relevant part of the image or the attentive feature maps. The context can be computed with an attention operation ϕ. We extract features from the convolutional layer of a pre-trained VGG-Net [6], then N regions are obtained. For all regions, we give a positive weight $\boldsymbol{\alpha}_t = \{\alpha_{t1}, \alpha_{t2}, \ldots, \alpha_{tN}\}$ to decide which locations are the right attentive places for the next label. Its element α_{ti} is computed by

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{N} \exp(e_{tk})}, \tag{11}$$

where

$$e_{ti} = g(\boldsymbol{a}_i, \boldsymbol{h}_{t-1}). \tag{12}$$

$\boldsymbol{e}_t = \{e_{t1}, e_{t2}, \ldots, e_{tN}\}$ represents the energy of the weight $\boldsymbol{\alpha}_t$. It reflects the importance of the feature $\boldsymbol{a}_i$ as well as the hidden state $\boldsymbol{h}_{t-1}$ and decides the next state of LSTM. Therefore, the LSTM is forced to pay more attention to these regions with larger weight. In Equation (12), g is a simple MLP.

Then, we can compute the dynamic context $\boldsymbol{z}_t$ as follows,

$$\boldsymbol{z}_t = \beta_t \cdot \phi(\{\boldsymbol{a}_i\}, \boldsymbol{\alpha}_t) = \beta_t \sum_{i=1}^{N} \alpha_{ti} \boldsymbol{a}_i. \tag{13}$$

Note that we use the sum of all weighted $\boldsymbol{a}_i$ to compute the expected aligned contexts. In the machine translation, this model scores the contribution of each inputs when predicting an output word. However, the sequence of a sentence lies in the 1-D space, which in the image it is 2-D. Consider this as shown in Fig. 5, we first compute $\boldsymbol{\alpha}_t$ by $\boldsymbol{feature}_{conv}$ and the previous hidden state $\boldsymbol{h}_{t-1}$. Then, with the weight $\boldsymbol{\alpha}_t$, we can obtain the attentive context $\boldsymbol{z}_t$ from $\boldsymbol{feature}_{conv}$. This attentive context shows how the weight influence the feature maps, so we regard their set $\boldsymbol{z}_t$ as another special features and feed them into LSTM as the next input. In a conventional model, the input of LSTM is usually the label embedding $\boldsymbol{x}_t$, computed from the ordered label sequence, and the previous hidden state $\boldsymbol{h}_{t-1}$. For convenience, we concatenate the label embedding and the dynamic context $\boldsymbol{z}_t$ as the uniform input of LSTM. That means for every step of the LSTM's recurrence, the model must take the possible area into account and overlook some unimportant information.

In Equation 13, we also add a selective gate β_t to the dynamic context $\boldsymbol{z}_t$. The gate controls of whether the current attentive context should consider more about the correlation with the previous labels. It is computed by the current hidden state:

$$\beta_t = \sigma(\boldsymbol{W}_{\beta t} \boldsymbol{h}_{t-1} + \boldsymbol{b}_{\beta t}). \tag{14}$$

To avoid the attention areas over-concentrated on one little point, we additionally add an attention regular term to the loss function. Thus, for one training instance $(\boldsymbol{x}_i, \tilde{\boldsymbol{Y}}_i)$, the loss function can be written as

$$\mathcal{L}(\boldsymbol{x}_i, \tilde{\boldsymbol{Y}}_i) = -\log(P(\tilde{\boldsymbol{Y}}_i | \boldsymbol{x}_i)) + \lambda \sum_{i}^{N} \left(1 - \sum_{t}^{T} \boldsymbol{\alpha}_t\right)^2, \tag{15}$$

where λ is the regular parameter, and the concentrated level is related to its value. T is the maximal sequence length.

IV. Experiments

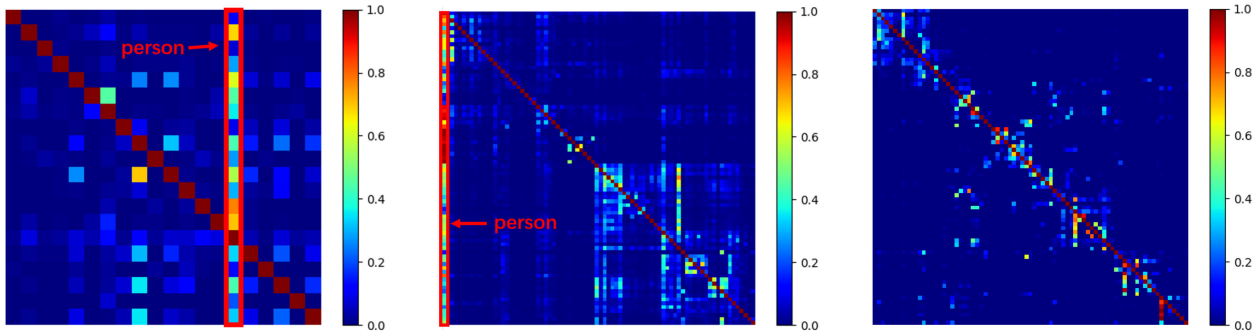
A. Datasets

We use two popular multi-label datasets to evaluate the performance, i.e., Pascal VOC [57] and Microsoft COCO [58]. Multi-label classification is required in many practical applications. For example, due to the convenience of the Internet, millions of news images are uploaded everyday. To this end, we construct a new dataset—Breaking News Image Dataset (BNID), collected by us from the Internet.

1) Two Benchmark Datasets: The PASCAL Visual Object Classes Challenge (Pascal VOC) dataset [57] and Microsoft COCO (MS-COCO) dataset [58] are two widely used multi-label classification datasets. Pascal VOC 2007 has 5,011 training examples and 4,952 testing examples of 20 classes. MS-COCO dataset [58] has 123,287 images (82,783 training and 40,504 testing examples) of 80 different classes.

2) BNID Dataset: The Breaking News Image Dataset (BNID), collected by us from the Internet, is made up of 9,986 online images with respect to the breaking news in recent years, for example, "battle, major conference, economic crisis". Due to the convenience of the Internet, millions of news images are uploaded everyday. It is necessary and meaningful for the government and the media to take control and make use of these images. BNID has 77 different classes, where more than half are abstract representations that are not directly related to objects (such as law, policy, G20), and the rest are the concrete objects or events (such as politician, fighter). We sample 1,000 testing examples from all data, and the rest 8,986 examples are for training.

To study the label dependencies of the above datasets, we show the co-occurrence rate of the labels (in Fig. 6), i.e., the proportion of one label appearing with another label simultaneously in the amount of the latter. We can see some labels have strong correlations with other labels while others do not. However, one interesting observation is that in VOC and MS-COCO, the correlations of "person" (the most colorful column) with all other labels are very high. This means when faced with the label "person", the model can not well predict the next label because it has strong correlations with nearly all the labels.



(a) label correlations among VOC (b) label correlations among MS-COCO (c) label correlations among BNID

Fig. 6. Visualization of label correlations on VOC, MS-COCO and BNID. The confusion matrix means the probability that one label appears with another one at the same time. Obviously, the label correlations of VOC and MS-COCO are very dense, which means most labels can not decide which is the most related label in different contexts. Moreover, the correlation of label "person" is stronger than other labels, which means when the first prediction is "person", it is difficult to decide which is the most related label.

TABLE I
COMPARISONS ON PASCAL VOC 2007

	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
INRIA [59]	77.2	69.3	56.2	66.6	45.5	68.1	83.4	53.6	58.3	51.1	62.2	45.2	78.4	69.7	86.1	52.4	54.4	54.3	75.8	62.1	63.5
FV [60]	75.7	64.8	52.8	70.6	30.0	64.1	77.5	55.5	55.6	41.8	56.3	41.7	76.3	64.4	82.7	28.3	39.7	56.6	79.7	51.5	58.3
CNN-SVM [15]	88.5	81.0	83.5	82.0	42.0	72.5	85.3	81.6	59.9	58.5	66.5	77.8	81.8	78.8	90.2	54.8	71.1	62.6	87.4	71.8	73.9
I-FT [17]	91.4	84.7	87.5	81.8	40.2	73.0	86.4	84.8	51.8	63.9	67.9	82.7	84.0	76.9	90.4	51.5	79.9	54.3	89.5	65.8	74.5
HCP-1000C [17]	95.1	90.1	92.8	89.9	51.5	80.0	91.7	91.6	57.7	77.8	70.9	89.3	89.3	85.2	93.0	64.0	85.7	62.7	94.4	78.3	81.5
HCP-2000C [17]	96.0	92.1	93.7	**93.4**	58.7	84.0	**93.4**	92.0	62.8	**89.1**	76.3	91.4	**95.0**	87.8	93.1	**69.9**	90.3	68.0	96.8	80.6	85.2
CNN-RNN [36]	96.7	83.1	**94.2**	92.8	**61.2**	82.1	89.1	**94.2**	64.2	83.6	70.0	92.4	91.7	84.2	93.7	59.8	**93.2**	**75.3**	**99.7**	78.6	84.0
CNN Multi-label	96.5	89.5	92.8	92.8	58.0	**85.6**	90.0	92.5	70.3	85.2	76.0	91.7	89.9	87.0	94.3	68.8	87.1	68.3	97.3	80.7	84.7
CNN-RNN Inject	96.1	90.6	92.8	92.2	51.6	79.5	88.8	90.8	70.5	82.1	72.4	90.8	94.0	87.8	95.0	64.7	87.6	64.5	97.1	77.0	83.3
CNN-RNN Merge	96.7	91.1	93.8	92.7	55.4	81.7	90.2	91.0	67.6	83.9	75.4	91.7	94.3	90.8	94.8	65.5	86.6	67.5	97.1	79.4	84.4
Ours-frequency	97.0	**92.5**	93.8	93.3	59.3	82.6	90.6	92.0	**73.4**	82.4	**76.6**	92.4	94.2	**91.4**	95.3	67.9	88.6	70.1	96.8	81.5	**85.6**
Ours-dependency	**97.2**	92.0	94.0	92.9	57.8	84.2	91.1	93.3	70.8	82.6	**76.6**	**92.9**	94.4	90.5	**95.7**	67.0	86.3	69.7	97.6	**83.0**	85.5

B. Baselines and Parameter Settings

The mainly comparative methods are CNN Multi-label, CNN-RNN. CNN Multi-label is the direct method that applies CNN to Multi-label classification. CNN-RNN model has two different categories, i.e., the *inject* and *merge* architectures. The inject architecture injects the feature vector into the RNN module, as well as the previous labels. The merge architecture merges the output of RNN with the feature vector every step before passing to the final output layer.

For the proposed method, we use VGG16 [6] as our back-bone model of the encoder CNN. The $\boldsymbol{feature}_{conv}$ are extracted from the last convolutional layer and $\boldsymbol{feature}_{fc}$ are extracted from the last FC-4096 layer. The parameters of VGG16 are pre-trained on ImageNet.

Moreover, we set both embedding size and the state size of LSTM to be 512. A special start label and a special end label are inserted into the prediction path. Every label has been represented as a one-hot form, and the length is the size of the vocabulary dictionary including the start and the end label. The learning rates are 10^{-3}, 2×10^{-3} and 10^{-1} for the convolutional layers, the first two fully-connected layers, and the later layers. We take 60 epochs in total and decrease the learning rate to one-tenth of the current rate every 20 epochs. The momentum is 0.9 and the weight decay is 5×10^{-4}.

The order of labels is quite important, and different label orders would make difference to the performance. In [37], the authors argue label orders do affect results. Inspired by [37], our proposed method produces two kinds of models, relying on two different sequence ordering rules, i.e., the frequency rule and the dependency rule. Here, the frequency rule reconstructs the multiple labels by the global statistical information, which means order with high frequency always prior to the low-frequency one, which is the same as the "frequent-first" in [37]. For example, there are more "person"s than "car"s in Pascal VOC 2007 in statistic, so the labels "car" and "person" should always follow the order "person, car". The dependency rule keeps the first label in statistic too but sorts the rest of labels by their own correlations, which is shown in Fig. 6. For example, if the most relevant label to "person" is the label "car" when the prediction contains more than three labels but including "person" and "car", the "car" should closely next to the "person". We use "Ours-frequency" and "Ours-dependency" to represent which rule the proposed method follows, respectively.

C. Evaluation Metrics

We use the following metrics to evaluate the methods. $\downarrow$ means the lower the metric is, the better the performance is, while $\uparrow$ is on the contrary.

- Precision (P@k): precision rate on top k;
- Recall (R@k): the recall rate on top k;
- F1-Score (F@k): the harmonic mean between precision and recall on top k;

TABLE II
COMPARISONS ON MS-COCO

Method	P@3/5(%)↑	R@3/5(%)↑	F@3/5(%)↑	H@3/5↓	A@3/5(%)↑	1-err(%)↓	cov↓	rloss↓	mAP(%)↑
CNN Multi-label	58.78/42.23	**71.96**/81.14	64.70/55.55	0.029/0.046	46.25/36.98	**11.25**	49.31	0.032	63.94
CNN-RNN Inject	57.78/41.41	70.66/79.60	63.58/54.48	0.030/0.046	45.23/36.07	12.82	48.43	0.046	60.73
CNN-RNN Merge	57.92/41.18	70.87/79.24	63.74/54.19	0.030/0.047	45.32/35.79	12.12	**47.81**	0.050	61.54
CNN-RNN [36]	-/-	-/-	-/-	-/-	-/-	-	-	-	61.20
Ours-frequency	**59.11/42.79**	71.91/**81.64**	**64.88/56.15**	0.029/**0.045**	46.51/**37.57**	11.50	49.83	**0.028**	**64.72**
Ours-dependency	57.56/41.82	70.03/79.81	63.19/54.88	0.030/0.046	**46.52**/36.48	13.67	49.99	0.035	62.00

TABLE III
COMPARISONS ON BNID

Method	P@3/5(%)↑	R@3/5(%)↑	F@3/5(%)↑	H@3/5↓	A@3/5(%)↑	1-err(%)↓	cov↓	rloss↓	mAP(%)↑
CNN Multi-label	51.04/36.33	62.96/74.28	56.37/48.79	0.031/0.049	45.53/34.68	37.60	50.50	0.072	49.57
CNN-RNN Inject	51.27/36.16	62.93/73.63	56.51/48.51	0.030/0.049	46.14/34.59	35.38	50.85	0.058	50.50
CNN-RNN Merge	52.45/37.19	64.60/75.87	57.89/49.92	0.029/0.048	47.37/35.60	33.87	50.42	0.057	52.08
Ours-frequency	**52.62**/37.35	64.75/76.17	**58.05**/50.13	0.029/0.048	**47.80**/35.70	35.58	**50.25**	0.056	53.48
Ours-dependency	52.45/**37.62**	**64.76/76.83**	57.96/**50.51**	0.029/**0.047**	47.13/**36.12**	**33.16**	50.71	**0.049**	**54.81**

TABLE IV
ABLATION STUDIES ON PASCAL VOC 2007

Situation	mAp(%)
w/o global/local features	83.9
only local features	84.8
only global features	85.2
w/ global/local features	85.6

- Hamming loss (H@k): the fraction of misclassified instance-label pairs on top k;
- Accuracy (A@k): accuracy on top k;
- One error (1-err): how many times the top-ranked label is not in the set of relevant labels of the example;
- Coverage (cov): how far, on average, we need to go down the list of ranked labels in order to cover all the relevant labels of the example;
- Rank loss (rloss): the average fraction of label pairs that are reversely ordered for the particular example;
- Mean average precision (mAP): average precision is the average fraction of labels ranked higher than a particular label, and mAP is the mean average precision across all labels.

D. Performance on Pascal VOC

We first evaluate our method on Pascal VOC 2007. We compare our method with other start-of-the-art methods with mAP. The results are shown in TABLE I, and our method achieves the start-of-the-art performance. Besides the three baseline methods, we also compare with other popular methods including I-FT [17], HCP-1000C/HCP-2000C [17] and CNN-RNN [36]. I-FT takes the similar idea with CNN Multi-label, and HCP-1000C/HCP-2000C use pre-trained region information to guide the fine-tuning of the model pre-trained on ImageNet 1000 classes or on additional classes. CNN-RNN is the similar to CNN-RNN Inject and CNN-RNN Merge, but it also uses beam search to improve the performance. We see that our method outperforms these state-of-the-art methods. Our method achieves

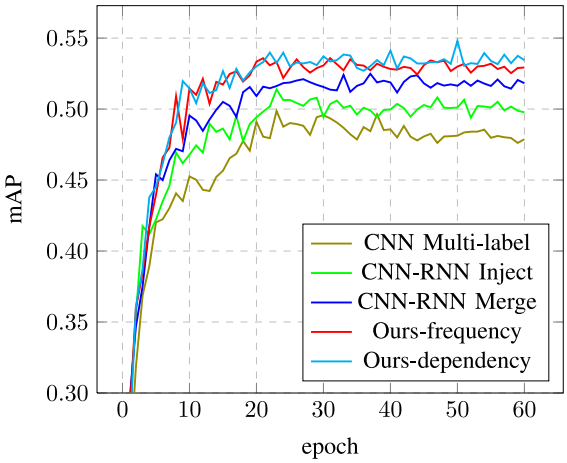
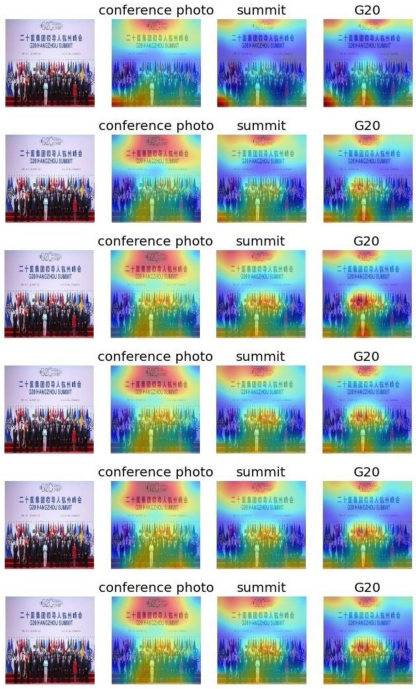


Fig. 7. The changing tread of mAP scores during training on BNID.

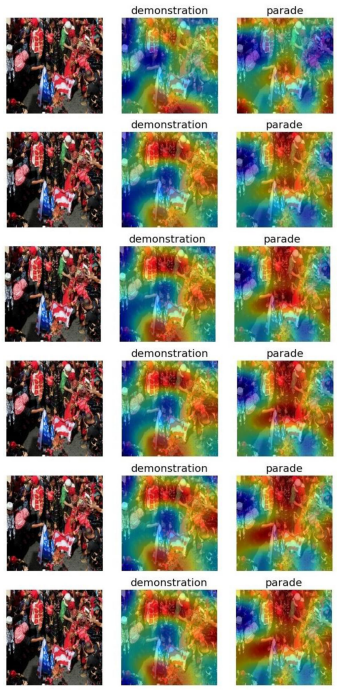
85.6% mAP (with the frequency rule), while CNN-RNN is only 84.0%, our CNN Multi-label is 84.7% and our CNN-RNN Merge is 84.4%. Note that HCP-2000C gets 85.2%, which is close to our results, because the method has quite good regions and pre-trained model. Specifically, our methods (with the frequency rule or the dependency rule) achieve the best Average Precision (AP) Score on {plane, bike, chair, table, dog, motor, person and tv}. Our attention mechanism takes not only the image global information but also the attentive local information to focus on the structural object classification. Hence we are good at predicting smaller object such as cat, tv, etc.

E. Performance on MS-COCO

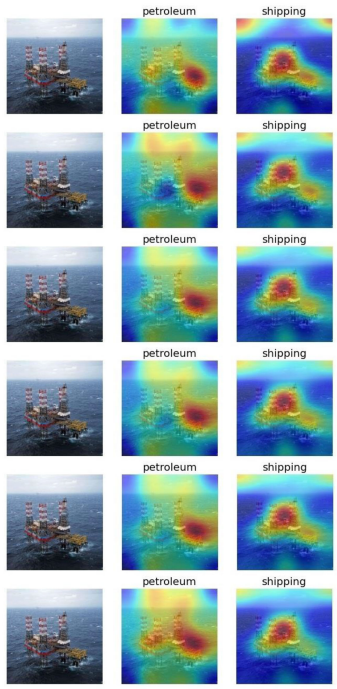
We also evaluate our model on MS-COCO. We can see the results in TABLE II that our method with the frequency rule is better than others in most evaluation metrics. With the frequency rule, the mAP of our method is 64.72%, outperforming the second place, i.e., CNN Multi-label, whose mAP is 63.94%. However, the CNN-RNN-based methods have relatively low performances. There might be two reasons, according to the



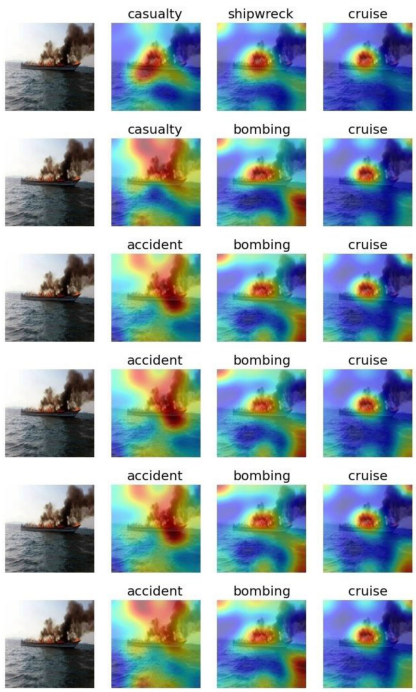
(a) Ground truth: conference photo, summit, G20



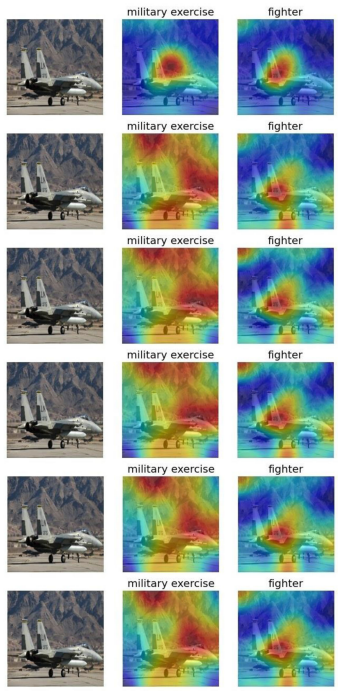
(b) Ground truth: demonstration, parade



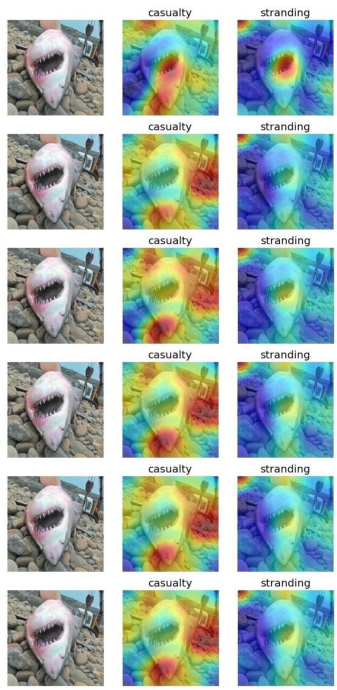
(c) Ground truth: petroleum, shipping



(d) Ground truth: accident, bombing, cruise



(e) Ground truth: millitart ecercise, fighter



(f) Ground truth: casualty, stranding

Fig. 8. Visualization of our attentive images on BNID. We show the predictions and the relative attentive areas of images every 10 epochs.

Fig. 6: 1) The label "person" has high correlations with all other labels, which may difficult to decide the next label when it is the first predicted label. 2) In VOC and MS-COCO, the correlations of every labels are very dense, which means it is difficult to decide which is the most related label in various contexts. For each label, the most related label is different, so the correlations should be sparse enough to highlight the most related one. Another evidence to prove the confusing correlation of MS-COCO is that when our model is trained with the dependency rule, the performance can be worse (62.00%). The dependency rule has less power on MS-COCO, because the dense correlations of the dataset may lead to disorder, which hampers the predictions.

Our overall architecture also takes a CNN-RNN framework, but with our attention mechanism, the confusing label correlations can be guided by the attentive area. Thus, our methods with both rules are better than the CNN-RNN based methods. The model also attend on regions, and combines the local information with the label dependency. Therefore, our methods outperform the CNN-based methods.

F. Performance on BNID

We compare the proposed method with start-of-the-art methods on BNID dataset, and TABLE III reports the results. BNID dataset is quite a complex dataset, including 77 different classes in total. More than half of them are the abstract concepts, which means for the traditional methods based on recognizing the concrete objects are in little effect. Our method outperforms them for almost all metrics, especially the mAP. By contrast, CNN Multi-label is much weaker than other methods in all the metrics. One possible reason is the image in BNID is too complex, and it is difficult for CNN to describe the whole features. Although CNN is good at analyzing the image with only concrete objects, the ability of processing correlations is weak. CNN-RNN Inject is able to learn some effective correlations among labels, but it only makes use of the image information at the begging of the LSTM. The rest recurrence of the LSTM predict the label only by the label dependency, where the image information is much weakened. CNN-RNN Merge is superior to CNN Multi-label and CNN-RNN Inject but is worse than the proposed method, since the proposed method joins the useful attention mechanism and guides the model to predict the next label. The mAP of our method can reach up to 53.48%. As shown in Fig. 7, the mAP of our method converges fast to 51.88% after 10 epochs, while it is 45.19%, 45.72% and 50.2% for CNN Multi-label, CNN-RNN Inject, and CNN-RNN Merge. We find that in BNID, the proposed method is more powerful in some material objects such as some persons, “fighter”, and “train”. However, the performance is still good in many abstract concepts, such as “peace”, “system”, and “independence”. Although the attention mechanism cannot understand where to focus when occurring the abstract concepts, it becomes much defocusing to analyze the whole image, since the correlations provide the alimentative information to the model while coming across the abstract labels. We also see that when we replace the frequency rule of the prediction path with the dependency rule, the performance can be improved further.

G. Ablation Studies

To verify the effectiveness of global and local features, we add ablation studies for four different situations on Pascal VOC 2007, i.e. 1) without global/local features; 2) with only local features; 3) with only global features; 4) with global/local features. The results are shown in TABLE IV. For fair, we set all situations with same settings (including with the dependency label order). Obviously, the proposed model is able to obtain good enough result (85.2%) with only global features, which means it is significant to remind RNN the information of the original image at every step. We also find that using the local features to compute image context can also improve the performance, in other words, the attention mechanism is effective. Our proposed model, combining with both global and local features, achieves the best performance (85.6%), which means when doing multi-label image classification with CNN-RNN structure, the information of image from coarse to fine is all important.

H. Visualization of Attention

We visualize our attentive areas for the image by upsampling the attention weights with a factor of $2^4 = 16$ and applying a Gaussian filter. In Fig. 8, we show the predictions and the relative attentive areas of images every 10 epochs. We find that for the abstract concepts such as “conference photo”, “G20”, and “military exercise”, the attention mechanism pays more attention to the whole image and the attentive areas is wide. However, when occurring solid objects such as “cruise” and “fighter”, the attentive area would be more concentrated. This is very like the human thinking that people observe the global image and consider the relationship in the image, when they adjust whether one image belongs to an abstract concepts. We also see that with the epoch goes, the attentive areas is adjusted by themselves. The attention mechanism tries to involve exact areas into the model, so that the areas is gradually close to the real areas.

V. Conclusion

In this paper, we propose a novel attentive multi-label classification method. Our model uses a dynamic attention mechanism, which generates attentive area to guide LSTM to predict sequence labels. The results show our approach is superior to other state-of-the-art methods. We also find that our approach has quite better performance on predicting abstract representations, which has been validated on our BNID dataset.

References

[1] J. T. Zhou, I. W. Tsang, S.-S. Ho, and K.-R. Müller, “N-ary decomposition for multi-class classification,” *Mach. Learn.*, 2018.
[2] J. T. Zhou *et al.*, “Learning with annotation of various degrees,” *IEEE Trans. Neural Netw. Learn. Syst.*, 2019.
[3] J. T. Zhou *et al.*, “Transfer hashing: From shallow to deep,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6191–6201, Dec. 2018.
[4] J. Deng *et al.*, “Imagenet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.
[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, vol. 25, pp. 1097–1105.
[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Representations*, 2015.
[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[8] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proc. Assoc. Adv. Artif. Intell.*, 2017, vol. 4, 12 pages.
[9] Y. Guo *et al.*, “The shallow end: Empowering shallower deep-convolutional networks through auxiliary outputs,” 2016, arXiv preprint arXiv:1611.01773.
[10] Y. Guo, Q. Wu, C. Deng, J. Chen, and M. Tan, “Double forward propagation for memorized batch normalization,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3134–3141.

[11] B. K. Bao, T. Li, and S. Yan, "Hidden-concept driven multilabel image annotation and label ranking," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 199–210, Feb. 2012.

[12] M. Tan *et al.*, "Learning graph structure for multi-label image classification via clique generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4100–4109.

[13] X. Ding *et al.*, "Multi-instance multi-label learning combining hierarchical context and its application to image annotation," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1616–1627, Aug. 2016.

[14] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, "Deep convolutional ranking for multilabel image annotation," 2013, arXiv preprint arXiv:1312.4894.

[15] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 806–813.

[16] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1717–1724.

[17] Y. Wei *et al.*, "CNN: Single-label to multi-label," 2014, arXiv preprint arXiv:1406.5726.

[18] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[19] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.

[20] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2009, pp. 254–269.

[21] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, 2011.

[22] J. Fürnkranz, E. Hüllermeier, E. L. Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Mach. Learn.*, vol. 73, no. 2, pp. 133–153, 2008.

[23] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *Proc. Eur. Conf. Mach. Learn.*, 2007, pp. 406–417.

[24] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1079–1089, Jul. 2011.

[25] H. Y. Lo, J. C. Wang, H. M. Wang, and S. D. Lin, "Cost-sensitive multi-label learning for audio tag annotation and retrieval," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 518–529, Jun. 2011.

[26] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu, "Multilabel image classification with regional latent semantic dependencies," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2801–2813, Oct. 2018.

[27] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.

[28] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *Proc. 5th Eur. Conf. Princ. Data Mining Knowl. Discovery*, 2001, pp. 42–53.

[29] A. Elisseeff *et al.*, "A kernel method for multi-labelled classification," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst., Natural Synthetic*, vol. 14, 2001, pp. 681–687.

[30] N. Ghamrawi and A. McCallum, "Collective multi-label classification," in *Proc. 14th ACM Int. Conf. Inf. Knowl. Manage*, 2005, pp. 195–200.

[31] J. Wu *et al.*"Weak labeled active learning with conditional label dependence for multi-label image classification," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1156–1169, Jun. 2017.

[32] J. Xu, V. Jagadeesh, and B. S. Manjunath, "Multi-label learning with fused multimodal bi-relational graph," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 403–412, Feb. 2014.

[33] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, arXiv preprint arXiv:1409.0473.

[34] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Emp. Methods Natural Lang. Process.*, Doha, Qatar, Oct. 25–29, 2014, pp. 1724–1734.

[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[36] J. Wang *et al.*, "CNN-RNN: A unified framework for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2285–2294.

[37] J. Jin and H. Nakayama, "Annotation order matters: Recurrent image annotator for arbitrary length image tagging," in *Proc. 23rd Int. Conf. Pattern Recognit.*, Cancún, Mexico, Dec. 4–8, 2016, pp. 2452–2457.

[38] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, and C. Sun, "Semantic regularisation for recurrent image annotation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4160–4168.

[39] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[40] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4651–4659.

[41] C. Gan, Y. Li, H. Li, C. Sun, and B. Gong, "VQS: Linking segmentations to questions and answers for supervised attention in VQA and question-focused semantic segmentation," in *Proc. IEEE Int. Conf. Comp. Vis*, 2017, vol. 3, pp. 1829–1838.

[42] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "StyleNet: Generating attractive visual captions with styles," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 955–964.

[43] M. Tan, I. W. Tsang, and L. Wang, "Towards ultrahigh dimensional feature selection for big data," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1371–1429, 2014.

[44] Z. Zhuang *et al.*, "Discrimination-aware channel pruning for deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 883–894.

[45] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang, "Learning spatial regularization with image-level supervisions for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5513–5522.

[46] Y. Huang, W. Wang, and L. Wang, "Unconstrained multimodal multi-label learning," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1923–1935, Nov. 2015.

[47] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Bing: Binarized normed gradients for objectness estimation at 300fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3286–3293.

[48] H. Yang, J. T. Zhou, J. Cai, and Y. S. Ong, "MIML-FCN+: Multi-instance multi-label learning via fully convolutional networks with privileged information," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1577–1585.

[49] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[50] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 328–335.

[51] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3156–3164.

[52] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille, "Explain images with multimodal recurrent neural networks," 2014, arXiv preprint arXiv:1410.1090.

[53] Q. Wu, C. Shen, A. v. d. Hengel, P. Wang, and A. Dick, "Image captioning and visual question answering based on attributes and their related external knowledge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1367–1381, Jun. 2018.

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

[55] W. Zaremba and I. Sutskever, "Learning to execute," 2014, arXiv preprint arXiv:1410.4615.

[56] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, and R. R. Salakhutdinov, "Review networks for caption generation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2361–2369.

[57] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[58] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

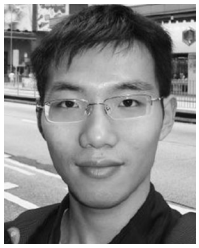
[59] H. Harzallah, F. Jurie, and C. Schmid, "Combining efficient object localization and image classification," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, 2009, pp. 237–244.

[60] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.

Fan Lyu received the B.S. and M.S. degrees in electronic and information engineering, Suzhou University of Science and Technology, Suzhou, China, in 2015 and 2018, respectively. He is currently working toward the Ph.D. degree with the College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests include image processing, multi-label classification, and image captioning.

Qi Wu received the M.Sc. and Ph.D. degrees in computer science from the University of Bath, Bath, U.K., in 2011 and 2015, respectively. His educational background is primarily in computer science and mathematics. He is currently a Lecturer (Assistant Professor) with The University of Adelaide, Adelaide, SA, Australia, and also an Associate Investigator with the Australia Centre for Robotic Vision (ACRV). He is the ARC Discovery Early Career Researcher Award Fellow between 2019–2021. He was working with Prof. Anton van den Hengel, Prof. Ian Reid, and Prof Chunhua Shen. Prior to joining the ACRV, he worked as a Senior Research Associate with the Australia Centre for Visual Technology. He is the CTO of the Vismarty Company, Ltd.

Fuyuan Hu is currently working toward the Ph.D. degree with Northwestern Polytechnical University, Xian, China. He is a Visiting Ph.D. Student with the City University of Hong Kong. He was a Postdoctoral Researcher with Vrije Universiteit Brussel, Brussel, Belgium. He is a Professor in computer vision and machine learning with the Suzhou University of Science and Technology. His research interests include graphical models, structured learning, and tracking.



Qingyao Wu received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 2013. He is an Associate Professor with the School of Software Engineering, South China University of Technology. He worked as a Postdoc Research Fellow with the School of Computer Engineering, Nanyang Technological University until 2015. His research interests include machine learning, data mining, big data research, computer vision, and bioinformatics.

Mingkui Tan received the bachelor's degree in environmental science and engineering in 2006 and the master's degree in control science and engineering in 2009, both from Hunan University, Changsha, China, and the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2014. From 2014–2016, he worked as a Senior Research Associate on computer vision with the School of Computer Science, University of Adelaide, Adelaide, SA, Australia. Since 2016, he has been with the School of Software Engineering, South China University of Technology, Guangzhou, China, where he is currently a Professor. His research interests include machine learning, sparse analysis, deep learning, and large-scale optimization.