Recurrent Image Annotation With Explicit Inter-Label Dependencies

Ayushi Dutta[1⋆][0000−0001−6958−5017], Yashaswi Verma[2][0000−0003−2317−2641], and C.V. Jawahar[3][0000−0001−6767−7057]

[1] Target Corporation India Private Limited, India (ayushi.dutta@target.com)
[2] Indian Institute of Technology, Jodhpur, India (yashaswi@iitj.ac.in)
[3] IIIT Hyderabad, India (jawahar@iiit.ac.in)

Abstract. Inspired by the success of the CNN-RNN framework in the image captioning task, several works have explored this in multi-label image annotation with the hope that the RNN followed by a CNN would encode inter-label dependencies better than using a CNN alone. To do so, for each training sample, the earlier methods converted the ground-truth label-set into a sequence of labels based on their frequencies (e.g., rare-to-frequent) for training the RNN. However, since the ground-truth is an unordered *set* of labels, imposing a fixed and predefined sequence on them does not naturally align with this task. To address this, some of the recent papers have proposed techniques that are capable to train the RNN without feeding the ground-truth labels in a particular sequence/order. However, most of these techniques leave it to the RNN to implicitly choose one sequence for the ground-truth labels corresponding to each sample at the time of training, thus making it inherently biased. In this paper, we address this limitation and propose a novel approach in which the RNN is explicitly forced to learn multiple relevant inter-label dependencies, without the need of feeding the ground-truth in any particular order. Using thorough empirical comparisons, we demonstrate that our approach outperforms several state-of-the-art techniques on two popular datasets (MS-COCO and NUS-WIDE). Additionally, it provides a new perspecitve of looking at an unordered set of labels as equivalent to a collection of different permutations (sequences) of those labels, thus naturally aligning with the image annotation task. Our code is available at: https://github.com/ayushidutta/multi-order-rnn

Keywords: Image annotation; Multi-label learning; CNN-RNN framework; Inter-label dependencies; Order-free training

1 Introduction

Multi-label image annotation is a fundamental problem in computer vision and machine learning, with applications in image retrieval [7, 37, 53], scene recognition [1], object recognition [47], image captioning [8], etc. In the last few years,

⋆ The author did most of this work while she was a student at IIIT Hyderabad, India.

deep Convolution Neural Networks (CNNs) such as [22, 39, 40, 15] have been shown to achieve great success in the single-label image classification task [38], which aims at assigning *one* label (or category) to an image from a fixed vocabulary. However, in the multi-label image annotation task, each image is associated with an unordered *subset* of labels from a vocabulary that corresponds to different visual concepts present in that image, such as objects (e.g., *shirt*), attributes (e.g., *green*), scene (e.g., *outdoor*), and other visual entities (e.g., *pavement*, *sky*, etc.). Further, these labels share rich semantic relationships among them (e.g., *forest* is related to *green*, *ferrari* is related to *car*, etc.), thus making it much more challenging than single-label classification.

To model inter-label dependencies, existing works have used a variety of techniques, such as nearest-neighbours based models [33, 12, 45], ranking-based models [14, 2] probabilistic graphical models [27, 28], structured inference models [47, 23, 17, 48], and models comprising of a Recurrent Neural Network (RNN) following a CNN [48, 18, 48, 31, 4] (also referred to as CNN-RNN framework). Among these, CNN-RNN based models have received increasing attention in the recent years [17, 48, 18, 4, 5, 26, 50], particularly due to the capability of an RNN to capture higher-order inter-label relationships while keeping the computational complexity tractable. The earlier models in this direction were motivated by the success of the CNN-RNN framework in the image captioning task [46, 21]. Analogous to the sequence/order of words in a caption, these models proposed to train the RNN for the image annotation task by imposing a fixed and predefined order on the labels based on their frequencies in the training data (e.g., frequent-to-rare or rare-to-frequent). In [18], Jin and Nakayama showed that the order of labels in the training phase had an impact on the annotation performance, and found that rare-to-frequent order worked the best, which was further validated in the subsequent papers such as [48, 31]. However, such an ordering introduces a hard constraint on the RNN model. E.g., if we impose rare-to-frequent label order, the model would be forced to learn to identify the rare labels first, which is difficult since these labels have very few training examples. Further, in RNN, since the future labels are predicted based on the previously predicted ones, any error in the initial predictions would increase the likelihood of errors in the subsequent predictions. Similarly, if we impose frequent-to-rare label order, the model would get biased towards frequent labels and would have to make several correct predictions before predicting the correct rare label(s). In general, any frequency-based predefined label order does not reflect the true inter-label dependencies since when an image has multiple labels, each label is related to many other labels with respect to the global context of that image, though spatially a label may relate more strongly to only a few of them. Additionally, defining such an order makes the model biased towards the dataset-specific statistics.

To address these limitations, some recent papers [4, 5, 26, 50] have proposed techniques that do not require to feed the ground-truth labels to the RNN in any particular sequence. However, these techniques allow the RNN to implicitly choose one out of many possible sequences, which in turn makes it inherently biased. In this paper, we address this limitation using a novel approach in which

the RNN is explicitly forced to learn multiple relevant inter-label dependencies in the form of multiple label orders instead of a fixed and predefined one. Specifically, at any given time-step, we train the model to predict all the correct labels except the one it has selected as the most probable one in the previous time-step. During testing, we max-pool the prediction scores for each label across all the time-steps, and then pick the labels with scores above a threshold. In this way, the best prediction of a label is obtained from its individual prediction path. Additionally, allowing the model to learn and predict from multiple label paths also provides the advantage that in reality there may be more than one sequences that reflect appropriate inter-label dependencies. As one could observe, the proposed idea is closely related to the well-known Viterbi algorithm, and provides a new perspecitve of looking at an unordered set of labels as equivalent to a collection of different permutations of those labels, thus naturally aligning the inherent capability of an RNN (i.e., sequence prediction) with the objective of the image annotation task (i.e., unordered subset prediction). In our experiments on two large-scale multi-label image annotation datasets, we demonstrate that the proposed approach outperforms competing baselines and several state-of-the-art image annotation techniques.

2 Related Work

Multi-label image annotation has been an active area of research from the last two decades. The initial works such as [24, 9, 3, 33, 47, 12, 2, 43, 44] relied on handcrafted local [24, 9, 3] and global [33, 12, 2, 43, 44] features, and explored a variety of techniques such as joint [24, 9] and conditional [3] probabilistic models, nearest-neighbours based models [33, 12, 43], structured inference models [23] and ranking-based models [2, 44].

With the advent of the deep learning era, most of the initial attempts were based on integrating the existing approaches with the powerful features made available by pre-trained deep CNN models. In [11], the authors used a deep CNN model pre-trained on the ImageNet dataset, and fine-tuned it for multi-label image annotation datasets using different loss functions such as softmax, pairwise ranking [19] and WARP [49]. Similarly, other works such as [42, 35, 45] revisited some of the state-of-the-art methods from the pre-deep-learning era, and re-evaluated them using the features extracted from the last fully-connected hidden layer of a pre-trained deep CNN model. Moving further on the similar ideas, Li *et al.* [29] introduced a smooth variant of the hinge-loss (called log-sum-exp pairwise loss, or LSEP loss) especially useful for the multi-label prediction task, and showed it to perform better than the previously known loss functions.

In parallel, there have also been attempts to explicitly model inter-label dependencies prominent in this task using end-to-end deep learning based techniques. One of the early attempts was by Andrea *et al.* [10] who proposed to learn a joint embedding space for images and labels using a deep neural network, thus allowing direct matching between visual (images) and textual (labels) samples in the learned common space, similar to [49]. To capture the underlying relation-

ships between labels in a deep CNN framework, Feng *et al.* [52] proposed a spatial regularization network with learnable convolution filters and attention maps for individual labels, by making use of spatial features from the last convolution layer. On similar lines, several other works such as [32, 13] have also explored the utility of local features and spatial attention. Apart from these, some of the works have also explored techniques such as deep metric learning for multi-label prediction [25], multi-modal learning [36] similar to [10], and Generative Adversarial Networks [41]. It is worth noting that while these approaches remained confined to the available training data, some of the works have demonstrated the advantage of using contextual knowledge coming from external sources. In [20], Johnson *et al.* used a non-parametric approach to find nearest neighbours of an image based on textual meta-data, and then aggregated visual information of an image and with its neighbours using a neural network to improve label prediction. In [17], Hu *et al.* proposed a deep structured neural network that consisted of multiple concept-layers based on the WordNet [34] hierarchy, and trained it to capture inter-label relationships across those layers.

Another class of algorithms that has become popular in the recent past is based on the CNN-RNN framework, that is motivated by the ability of an RNN to model complex inter-label relationships, and at the same time it offers a simple and scalable solution. The earlier attempts were simple adaptations of the CNN-RNN based encoder-decoder models proposed for the image captioning task [46]. These approaches treated multi-label prediction as a sequence prediction problem, where the RNN was trained to predict the labels for a given image in a sequential manner, analogous to predicting a caption [18, 48, 31]. As discussed above, such approaches required a predefined order among the labels at the time of training, and thus constrained the model to predict the labels in that order. Since this does not naturally align with the objective of the image annotation task, some of the recent approaches have proposed order-free techniques that do not require to feed the ground-truth labels to the RNN in any particular order at the time of training. The Order-free RNN model proposed by Shang *et al.* [4] was the first such model that used the concept of "candidate label pool". This pool initially contains all the true labels, and then at each time-step, the most confident label from this pool is used for feedback to the RNN and at the same time removed from this pool. Similar ideas have been proposed in the subsequent works such as [5, 26, 50]. However, these approaches are prone to internally choosing one particular order of labels at the initial time-step, and then iterating over the same sequence in the subsequent time-steps. To address this limitation, we propose a novel approach that forces the RNN model to predict all the correct labels at every time-step, except the one predicted in the previous time-step. As we will show later in Section 3.3, this drives the RNN to learn complex inter-label dependencies in the form of multiple sequences among the labels arising from a given label-set, and thus we call it *Multi-order RNN*. The utility of our approach is also demonstrated in the empirical analysis in which it is shown to outperform all the existing CNN-RNN-based image annotation techniques.

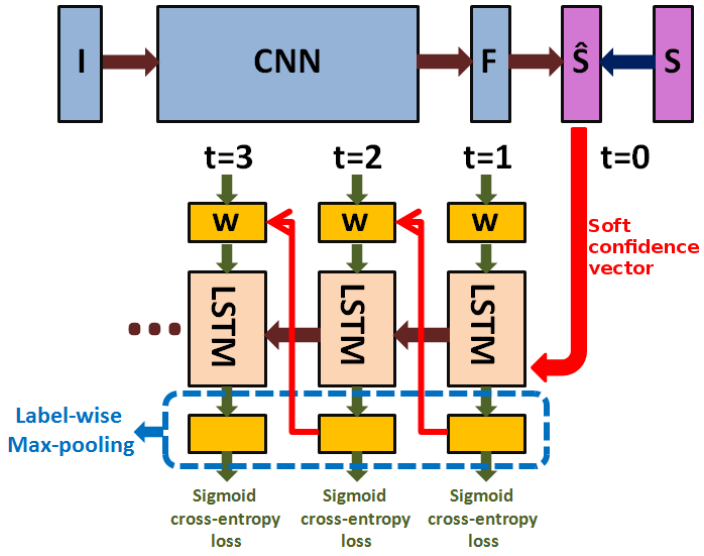

Fig. 1. Overview of the proposed approach. The first component of our model is a deep CNN that that is fine-tuned using the ground-truth (S) from a given dataset. The second component is an LSTM model that uses the soft confidence vector ($\hat{S}$) from the CNN as its initial state. Given a sample, at every time-step, a cross-entropy loss is computed considering all the true labels except the one from the previous time-step as possible candidates for prediction at that time-step. Final predictions are obtained by max-pooling individual label scores across all the time-steps

3 Approach

Our CNN-RNN framework consists of two components: (i) a deep CNN model that provides a real-valued vectorial representation of an input image, and (ii) an RNN that models image-label and label-label relationships. Let there be N training images $\{I_1, \ldots, I_N\}$, such that each image I is associated with a ground-truth label vector $y = (y^1, y^2, \ldots, y^C)$, where y^c $(\forall c \in \{1, 2, \ldots, C\})$ is 1 if the image I has the c^{th} label in its ground-truth and 0 otherwise, with C being the total number of labels. Also, let $\hat{y}$ denote the vector containing the scores predicted by the RNN corresponding to all the labels at some time-step.

During the training phase, for a given image I, the representation obtained by the CNN is initially fed to the RNN. Based on this, the RNN predicts a label score vector $\hat{y}_t$ at each time-step t in a sequential manner, and generates a prediction path $\pi = (a_1, a_2, ..., a_T)$, where a_t denotes the label corresponding to the maximum score from $\hat{y}_t$ at time t, and T is the total number of time-steps. During inference, we accumulate the scores for individual labels across all the prediction paths using max-pooling, and obtain the final label scores (Figure 1).

Below, first we describe the basic CNN-RNN framework, and then present the proposed Multi-order RNN model.

3.1 Background: CNN-RNN framework

Given a CNN model pre-trained on the ImageNet dataset for the image classification task, the first step is to fine-tune it for a given multi-label image annotation dataset using the standard binary cross-entropy loss. Next, we use the soft confidence (label probability) scores ($\hat{s} \in \mathbb{R}_{+}^{C\times 1}$) predicted by the CNN for an input image, and use this as the interface between the CNN and the RNN. This allows to decouple the learning of these two components, and thus helps in a more efficient joint training [31, 4].

Taking $\hat{s}$ as the input, the RNN decoder generates a sequence of labels $\pi = (a_1, a_2, ..., a_{n_I})$, where a_t is the label predicted at the t^{th} time-step, and n_I is the total number of labels predicted. Analogous to the contemporary approaches, we use the Long Short-Term Memory (LSTM) [16] network as the RNN decoder, which controls message passing between time-steps with specialized gates. At time step t, the model uses its last prediction a_{t-1} as the input, and computes a distribution over the possible outputs:

$$x_t = E \cdot a_{t-1} \tag{1}$$

$$h_t = LSTM(x_t, h_{t-1}, c_{t-1}) \tag{2}$$

$$\hat{y}_t = W \cdot h_t + b \tag{3}$$

where E is the label embedding matrix, W and b are the weight and bias of the output layer, a_{t-1} denotes the *one-hot encoding* of the last prediction, c_t and h_t are the model's cell and hidden states respectively at time t, and $LSTM(\cdot)$ is a forward step of the unit. The output vector $\hat{y}_t$ defines the output scores at t, from which the next label a_t is sampled.

3.2 Multi-order RNN

As introduced earlier, let the ground-truth (binary) label vector of an image I be denoted by $y = (y^1, y^2, \ldots, y^C)$. Also, let y_t denote the ground-truth label vector at time t, and $\hat{y}_t$ be the corresponding predicted label score vector. In practice, since the ground-truth y_t at time-step t is unknown, one could assume that y_t at each time-step is the original label vector y. This would force the model to assign high scores to all the ground-truth labels instead of one particular ground-truth label at each time-step. E.g., let us assume that an image has the labels {*sky, clouds, person*} in its ground-truth, then the model would be forced to predict (i.e., assign high prediction scores to) all the three labels {*sky, clouds, person*} at each time-step based on the most confident label predicted in previous time-step. However, this poses the problem that if the most confident label predicted at time-step t is l, the model may end-up learning a dependency from l to l in the next time-step along with the dependencies from l to other labels. In other

words, there is a high chance that a label which is easiest to predict would be the most confident prediction by RNN at every time-step, and thus the same label would then be repeatedly chosen for feedback to the RNN. To address this, we use a greedy approach that forces the model to explicitly learn to predict a different label. Specifically, if l is the most confident prediction at time-step t, we mask out l in the next time-step; *i.e.*, in the next time-step, we treat l as a negative label rather than positive and learn a dependency from l to all other labels except itself. We explain this mathematically below.

Let l_t be the most confident label with the highest prediction score for an image I at time-step t:

$$l_t = \underset{c \in \{1,2,...,C\}}{\arg\max} \; \hat{y}_t^c \tag{4}$$

Let a_{t-1} be the one-hot encoding corresponding to l_{t-1}. Then we define a label mask $\tilde{a}_t$ at time-step t as:

$$\tilde{a}_t = \neg a_{t-1} \tag{5}$$

In other words, this label mask is a negation of the one-hot encoding of the most confident label from the previous time-step. The mask contains a 0 corresponding to the previously selected label index, and 1 for the rest. Using this, we define a modified ground-truth label vector at time-step t as:

$$y_t = \tilde{a}_t \odot y \tag{6}$$

where $\odot$ represents element-wise multiplication. At time-step $t = 0$, $\tilde{a}_0$ will be a vector with all ones. Using this modified ground-label vector and the predicted label scores at a particular time-step t, we compute the sigmoid cross-entropy loss at that time-step as:

$$\mathcal{L}_t = y_t \cdot log(\sigma(\hat{y}_t)) + (1 - y_t) \cdot log(1 - \sigma(\hat{y}_t)) \tag{7}$$

The above loss is aggregated over all the time-steps and summed over all the training samples to obtain the total loss. This loss is then used to train our model using a gradient descent approach.

Label Prediction Once the model is trained, for a given test image, first we obtain the soft confidence (probability) scores from the CNN and initiate the LSTM using them. Then, the LSTM network is iterated for T time-steps, resulting in T prediction score vectors $(\hat{y}_1, \hat{y}_2, ..., \hat{y}_T)$, where each $\hat{y}_t = (\hat{y}_t^1, \hat{y}_t^2, \ldots, \hat{y}_t^C)$ denotes the scores for all the C labels at times-step t. We employ label-wise max-pooling to integrate the scores across all the time-steps into the final result $\hat{y} = (\hat{y}^1, \hat{y}^2, ..., \hat{y}^C)$, where: $\hat{y}^c = max(\hat{y}_1^c, \hat{y}_2^c, ..., \hat{y}_T^c)$, $\forall c = 1, \ldots, C$. The final predicted label probability distribution $\hat{p}$ is obtained as $\hat{p}_i = \sigma(\hat{y})$. Since we use the sigmoid function, finally we assign all those labels whose probability scores are greater than 0.5. We carry out the LSTM iterations for a fixed number of 'T' time-steps which is determined experimentally. Interestingly, unlike methods that do a predefined label order based training and sequential label prediction, our model does not require to predict an "end-of-sequence" ($<$ EOS $>$) token.

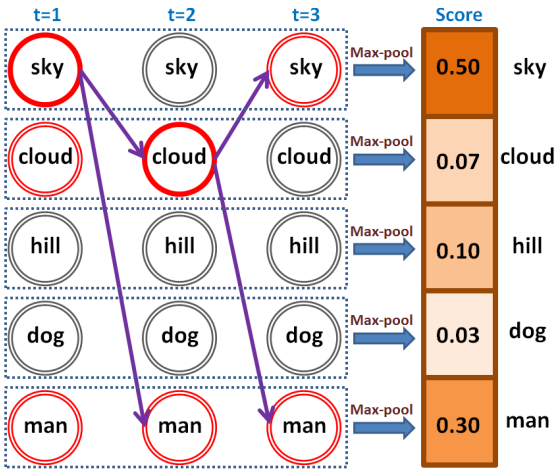

Fig. 2. An example of multiple inter-label dependencies that can be learned using the proposed approach. Please see Section 3.3 for details.

3.3 Discussion

Now we will discuss how the proposed approach learns multiple relevant inter-label dependencies (and not all possible permutations, since many of them will not be meaningful). Algorithmically, in the first time-step, the LSTM is trained to predict all the true labels by the loss $\mathcal{L}_1$, analogous to a CNN that can be trained to predict all the true labels by a single look at an image. As the LSTM strives to predict all the labels, the most confident label predicted by it is given as the feedback for the next time-step. In the second time-step, the model is trained to predict all the true labels except the most confident label predicted by it in the previous time-step, and this continues for a fixed number of time-steps.

Let us try to understand this through an example illustrated in Figure 2. Let $\{sky,\ cloud,\ hill,\ dog,\ man\}$ be the complete label-set (vocabulary), and let $\{sky,\ cloud,\ man\}$ be the ground-truth set of labels for a given image. **During training**, at $t = 1$, the model is forced to predict all the positive labels correctly. Suppose it selects *sky* as the most confident label. Then at $t = 2$, the model is trained to learn dependencies from *sky* to both *cloud* and *man*, while keeping *sky* as a negative label. Suppose the model now selects *cloud* as the label with the highest confidence. Then at $t = 3$, the model will be trained to predict both *sky* and *man*. In this way, the model explicitly learns multiple dependency paths $sky \rightarrow cloud \rightarrow sky$ and $sky \rightarrow cloud \rightarrow man$. In other words, it learns not only to go back to *sky* from *cloud*, but also learns to predict *man* based on the confidence that the image already has $\{sky,\ cloud\}$. In this way, the label correlations that are hard to learn initially get learned at later time-steps based on their dependencies on other labels in the ground-truth. **During testing**, given an input image, the most confident label at each time-step is fed as input

to the LSTM, and this is repeated for a fixed number of time-steps. At the end, the prediction scores across all the time-steps are label-wise max-pooled, and the labels above a threshold are assigned.

The proposed training approach is a greedy approach that is self-guided by the training procedure. Let $\{l_1, l_2, l_3, l_4, l_5\}$ be the complete set of labels, and let $\{l_1, l_3, l_5\}$ be the true labels for a given image. If the model is most confident at predicting l_1 at the first time-step, we train it to predict $\{l_3, l_5\}$ in the next time-step. If the model had predicted l_3 first, the training at the next time-step would have been for $\{l_1, l_5\}$. In case the model predicts an incorrect label, say l_2, as the most confident prediction, it will be given as the feedback to the LSTM for the next time-step. However, since we penalize all the true negative labels (not present in the ground-truth) at every time-step, the model learns not to predict them. By self-learning multiple inter-label dependencies, the model inherently tries to learn label correlations. E.g., if l_3 is chosen immediately after l_1 by the model, there is a strong likelihood that $\{l_1, l_3\}$ co-occur often in practice, and it makes more sense to predict l_3 when l_1 is present. In contrast, in a predefined order based training, the label dependencies are learned in some specified order, which may not be an appropriate order in practice. Also, there may not be a single specific order that reflects the label dependencies in an image. Since the proposed approach can learn multiple inter-label dependencies paths, it can mitigate both these issues.

As we can observe, there is a possibility that the sequence of the most confident labels that the LSTM model predicts as it iterates over time-steps is $\{l_1, l_5, l_1, l_5, \ldots\}$. Algorithmically, this implies that l_1 was the most confident label at $t = 1$, l_5 at $t = 2$, l_1 at $t = 3$, l_5 at $t = 4$, and so on. Intuitively, this would mean that at each time-step, while the previous most confident label guides the LSTM model, the model is still forced to learn that labels dependencies with all the remaining correct labels including the (current) most confident one. While this particular behavior is probably not the best one, this would still facilitate the model to encode multiple (even bidirectional) inter-label dependencies unlike any existing CNN-RNN-based method. At this point, if we had maintained a pool of all the true labels predicted until a certain time-step and forced the model to predict a label only from the remaining ones, then this would have resulted in forcing some improbable dependencies among the labels, which is not desirable. We would like to highlight that this is exactly what was proposed in [4], and our algorithm elegantly relaxes this hard constraint imposed on LSTM. As evident from the empirical analyses, this results in achieving significantly better performance than [4].

4 Experiments

4.1 Datasets

We experiment using two popular and large-scale image annotation datasets: MS-COCO [30] and NUS-WIDE [6]. The MS-COCO dataset has been used for various object recognition tasks in the context of natural scene understanding. It

Table 1. Comparison of the proposed approach with a vanilla CNN using binary cross-entropy loss, and CNN-RNN models trained with different label-ordering methods on the MS-COCO dataset

Metric→	Per-label			Per-image		
Method↓	P_L	R_L	$F1_L$	P_I	R_I	$F1_I$
CNN (Binary Cross-Entropy)	59.30	58.60	58.90	61.70	65.00	63.30
CNN-RNN (frequent-first)	70.27	56.49	62.63	72.15	64.53	68.13
CNN-RNN (rare-first)	65.68	61.32	63.43	70.82	64.73	67.64
CNN-RNN (lexicographic order)	70.98	55.86	62.52	74.14	62.35	67.74
Multi-order RNN (Proposed)	**77.09**	**64.32**	**70.13**	**84.90**	**75.83**	**80.11**

contains 82,783 images in the training set, 40,504 images in the validation set, and a vocabulary of 80 labels with around 2.9 labels per image. Since the ground-truth labels for the test set are not publicly available, we use the validation set in our comparisons following the earlier papers. The NUS-WIDE dataset contains 269,648 images downloaded from Flickr. Its vocabulary contains 81 labels, with around 2.4 labels per image. Following earlier papers, we discard the images without any label, that leaves us with 209,347 images. For empirical comparisons, we split into around 125,449 images for training and 83,898 for testing by adopting the split originally provided along with this dataset.

4.2 Evaluation Metrics

For empirical analyses, we consider both per-label as well as per-image evaluation metrics. In case of per-label metrics, for a given label, let a_1 be the number of images that contain that label in the ground-truth, a_2 be the number of images that are assigned that label during prediction, and a_3 be the number of images with correct predictions ($a_3 \leq a_2$ and $a_3 \leq a_1$). Then, for that label, precision is given by $\frac{a_3}{a_2}$, and recall by $\frac{a_3}{a_1}$. These scores are computed for all the labels and averaged to get mean per-label precision P_L and mean per-label recall R_L. Finally, the mean per-label F1 score is computed as the harmonic mean of P_L and R_L; i.e., $F1_L = \frac{2 \times P_L \times R_L}{P_L + R_L}$.

In case of per-image metrics, for a given (test) image, let b_1 be the number of labels present in its ground-truth, b_2 be the number of labels assigned during prediction, and b_3 be the number of correctly predicted labels ($b_3 \leq b_2$ and $b_3 \leq b_1$). Then, for that image, precision is given by $\frac{b_3}{b_2}$, and recall by $\frac{b_3}{b_1}$. These scores are computed for all the test images and averaged to get mean per-image precision P_I and mean per-image recall R_I. Finally, the mean per-image F1 score is computed as the harmonic mean of P_I and R_I; i.e., $F1_I = \frac{2 \times P_I \times R_I}{P_I + R_I}$.

4.3 Implementation Details

We use ResNet-101 [15] pre-trained on the ILSVRC12 1000-class classification dataset [38] as our CNN model, and it is fine-tuned for both NUS-WIDE and MS-

Table 2. Comparison of the proposed approach with the state-of-the-art methods on the MS-COCO dataset

	Metric→	Per-label			Per-image		
	Method↓	P_L	R_L	$F1_L$	P_I	R_I	$F1_I$
CNN based	TagProp [12]	63.11	58.29	60.61	58.17	71.07	63.98
	2PKNN [45]	63.77	55.70	59.46	54.13	66.95	59.86
	MS-CNN+LQP [36]	67.48	60.93	64.04	70.22	67.93	69.06
	WARP [49, 11]	57.09	55.31	56.19	57.54	70.03	63.18
	LSEP [29]	73.50	56.40	63.82	76.30	61.80	68.29
	SRN [52]	**85.20**	58.80	67.40	**87.40**	62.50	72.90
	S-Cls [32]	–	–	69.20	–	–	74.00
	ACfs [13]	77.40	**68.30**	**72.20**	79.80	**73.10**	**76.30**
	WGAN-gp [41]	70.50	58.70	64.00	72.30	64.60	68.20
	RETDM [25]	**79.90**	55.50	65.50	81.90	61.10	70.00
CNN-RNN based	SR-CNN-RNN [31]	67.40	59.83	63.39	76.63	68.73	72.47
	Order-free RNN [4]	71.60	54.80	62.10	74.20	62.20	67.70
	Recurrent-Attention RL [5]	78.80	57.20	66.20	84.00	61.60	71.10
	Attentive RNN [26]	71.90	59.60	65.20	74.30	69.70	71.80
	MLA [50]	68.37	60.39	64.13	72.16	66.71	69.33
	PLA [50]	70.18	61.96	65.81	73.75	67.74	70.62
	Multi-order RNN (Proposed)	77.09	**64.32**	**70.13**	**84.90**	**75.83**	**80.11**

COCO datasets separately. For our LSTM (RNN) model, we use 512 cells with the *tanh* activation function and 256-dimensional label-embedding. We train using a batch size of 32 with RMSProp Optimiser and learning-rate of $1e-4$ for 50 epochs, and recurse the LSTM for $T = 5$ time-steps.

4.4 Results and Discussion

Comparison with baselines In Table 1, we compare the results of various baselines with our approach, including a CNN model trained with the standard binary-cross entropy loss, and three CNN-RNN style models trained using different schemes for ordering labels at the time of training the RNN module. Here, we observe that the CNN-RNN style models outperform the CNN model, indicating the advantage of using an RNN. We also notice that among the three baseline CNN-RNN models, CNN-RNN (frequent-first) outperforms others with the per-image metrics, and CNN-RNN (rare-first) outperforms others with the per-label metrics. This is expected since the rare-first order assigns more importance to less frequent labels, thus forcing the model to learn to predict them first.

In general, the proposed Multi-order RNN technique consistently outperforms all the baselines by a large margin, providing an improvement of 10.6% in terms of $F1_L$ and 17.6% in terms of $F1_I$ compared to the best performing baseline, thus validating the need for automatically identifying and learning with

Table 3. Comparison of the proposed approach with the state-of-the-art methods on the NUS-WIDE dataset

	Metric→	Per-label			Per-image		
	Method↓	P_L	R_L	$F1_L$	P_I	R_I	$F1_I$
CNN based	TagProp [12]	49.45	**59.13**	53.86	52.37	**74.21**	61.41
	2PKNN [45]	47.94	**55.76**	51.55	51.90	73.00	60.67
	LSEP [29]	**66.70**	45.90	54.38	**76.80**	65.70	70.82
	WARP [49, 11]	44.74	52.44	48.28	53.81	**75.48**	62.83
	CMA [51]	–	–	55.50	–	–	70.00
	MS-CMA [51]	–	–	55.70	–	–	69.50
	WGAN-gp [41]	**62.40**	50.50	55.80	71.40	70.90	71.20
CNN-RNN based	SR-CNN-RNN [31]	55.65	50.17	52.77	70.57	71.35	70.96
	Order-free RNN [4]	59.40	50.70	54.70	69.00	71.40	70.20
	Attentive RNN [26]	44.20	49.30	46.60	53.90	68.70	60.40
	PLA [50]	60.67	52.40	**56.23**	71.96	72.79	**72.37**
	Multi-order RNN (Proposed)	60.85	54.43	**57.46**	**76.50**	73.06	**74.74**

multiple label orders as being done in the proposed approach, rather than using a fixed and predefined one as in the baselines.

Comparison with the state-of-the-art In Table 2 and 3, we compare the performance of the proposed Multi-order RNN with both CNN based as well CNN-RNN based methods. In each column, we highlight the best result in red and the second best result in blue. Among the CNN-RNN based methods, SR-CNN-RNN [31] is the state-of-the-art method that uses rare-to-frequent order of labels for training the RNN, and Order-free RNN [4], Recurrent-Attention RL [5], Attentive RNN [26], MLA [50] and PLA [50] are the methods that do not require to feed the ground-truth labels in any particular order. Among the CNN based methods, TagProp [12] and 2PKNN [45] are state-of-the-art nearest-neighbour based methods that are evaluated using the ResNet-101 features (extracted from the last fully-connected layer), and others are end-to-end trainable models, with WARP [49, 11] and LSEP [29] being trained using a pairwise ranking loss. From the results, we can make the following observations: (a) CNN based methods achieve the maximum average precision in all the cases (SRN [52] on the MS-COCO dataset and LSEP [29] on the NUS-WIDE dataset). However, except ACfs [13], all other methods generally fail to manage the trade-off between precision and recall, thus resulting in low F1 scores. (b) On the MS-COCO dataset, the CNN based approaches perform better than the existing CNN-RNN based approaches. The proposed Multi-order RNN approach brings a big jump in the performance of this class of methods, making it either comparable to or better than the former one. (c) Compared to the existing CNN-RNN based approaches, Multi-order RNN not only achieves higher average precision and recall,

Table 4. Comparison between SR-CNN-RNN [31] and the proposed Multi-order RNN approach based on top-1 accuracy

NUS-WIDE		**MS-COCO**	
SR-CNN-RNN	Multi-order RNN	SR-CNN-RNN	Multi-order RNN
68.06	84.05	81.44	93.49

but also manages the trade-off between the two better than others by achieving an increase in both average precision as well recall, thus achieving the best F1 score in all the cases. (d) In terms of $F1_L$, Multi-order RNN is inferior only to ACfs [13] by 2.07% on the MS-COCO dataset, and outperforms all the methods on the NUS-WIDE dataset with its score being 1.23% more than the second best method. In terms of $F1_I$, Multi-order RNN outperforms all the methods on both the datasets, with its score being 3.81% (on MS-COCO) and 2.37% (on NUS-WIDE) better than the second best methods.

GT: {person, window}
SR-CNN-RNN: <start>, temple, <end>
Multi-order RNN: {temple, person}

t=1	temple(0.69)*	person(0.43)	sky(0.21)	animal(0.02)	buildings(0.02)
t=2	**person***(0.61)	temple(0.21)	sky(0.14)	buildings(0.03)	animal(0.03)
t=3	temple(0.58)	sky(0.09)	statue(0.03)	buildings(0.02)	water(0.01)
t=4	person(0.54)	temple(0.13)	sky(0.05)	buildings(0.05)	animal(0.05)
t=5	temple(0.47)	sky(0.05)	statue(0.03)	buildings(0.03)	plants(0.01)

Fig. 3. Comparison between the prediction of SR-CNN-RNN and Multi-order RNN for an example image. For Multi-order RNN, we show the top five labels along with their probabilities obtained at each time-step. * indicates the max pooled values

Additional analysis As discussed before, earlier CNN-RNN based image annotation approaches [18, 48, 31] had advocated the use of rare-to-frequent label order at the time of training, and SR-CNN-RNN [31] is the state-of-the-art method from this class of methods. Here, we further analyze the performance of our model against SR-CNN-RNN and consider "top-1 accuracy" as the evaluation metric that denotes the percentage of images with the correct top-1 predicted label. This label is obtained by doing one iteration of LSTM at the time of inference for both the methods (note that the training process is unchanged for both the methods, and both of them use ResNet-101 as their CNN model). As we can see in Table 4, the top-1 accuracy of Multi-order RNN is much higher compared to SR-CNN-RNN. This is so because in practice, many of the rare labels correspond to specific concepts that are difficult to learn as well as predict. In such a scenario, if the first predicted label is wrong, it increases the likelihood

		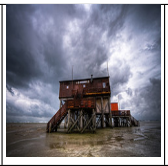		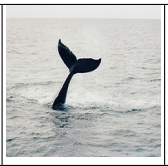	
Ground-truth	clouds, sky	clouds, sky, water, beach, buildings	clouds, sky, window	ocean, water, waves	animal, dog
Multi-order RNN	clouds, house, sky	clouds, sky, water	clouds, sky, vehicle, window	animal, ocean, water, waves	animal, dog, sky

Fig. 4. Annotations for example images from the NUS-WIDE dataset. The labels in blue are the ones that match with the ground-truth, and the labels in red are the ones that are depicted in the corresponding images but missing in their ground-truth

of the subsequent ones also being wrong. However, our model explicitly learns to choose a salient label based on which it can predict other labels, and thus achieves a higher top-1 accuracy. We further illustrate this in Figure 3 where for a given image, the SR-CNN-RNN approach predicts only one (incorrect) label *temple*. Interestingly, while Multi-order RNN also predicts *temple* as the most confident label at $t = 1$, it predicts a correct label *person* at $t = 2$ with probability > 0.5, showing its correlation with *temple* learned by the model.

Finally, we present some qualitative results in Figure 4. From these results, we can observe that our model correctly predicts most of the ground-truth labels. Moreover, the additional labels that are predicted but missing in the ground-truth are actually depicted in their corresponding images. These results further validate the capability of our model to learn complex inter-label relationships.

5 Summary and Conclusion

While recent CNN-RNN based multi-label image annotation techniques have been successful in training RNN without the need of feeding the ground-truth labels in any particular order, they implicitly leave it to RNN to choose one order for those labels and then force it to learn to predict them in that sequence. To overcome this constraint, we have presented a new approach called Multi-order RNN that provides RNN the flexibility to explore and learn multiple relevant inter-label dependencies on its own. Experiments demonstrate that Multi-order RNN consistently outperforms the existing CNN-RNN based approaches, and also provides an intuitive way of adapting a sequence prediction framework for the image annotation (subset prediction) task.

Acknowledgement: YV would like to thank the Department of Science and and Technology (India) for the INSPIRE Faculty Award 2017.

References

1. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern recognition **39**(9), 1757–1771 (2004)
2. Bucak, S.S., Jin, R., Jain, A.K.: Multi-label learning with incomplete class assignments. In: CVPR (2011)
3. Carneiro, G., Chan, A.B., Moreno, P.J., Vasconcelos, N.: Supervised learning of semantic classes for image annotation and retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **29**(3), 394–410 (2007)
4. Chen, S.F., Chen, Y.C., Yeh, C.K., Wang, Y.C.F.: Order-free rnn with visual attention for multi-label classification. In: AAAI (2018)
5. Chen, T., Wang, Z., Li, G., Lin, L.: Recurrent attentional reinforcement learning for multi-label image recognition. In: AAAI. pp. 6730–6737 (2018)
6. seng Chua, T., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: A real-world web image database from national university of singapore. In: In CIVR (2009)
7. Escalante, H.J., Hérnadez, C.A., Sucar, L.E., Montes, M.: Late fusion of heterogeneous methods for multimedia image retrieval. In: MIR (2008)
8. Fang*, H., Gupta*, S., Iandola*, F., Srivastava*, R.K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J.C., Zitnick, C.L., Zweig, G.: From captions to visual concepts and back. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1473–1482 (2015)
9. Feng, S.L., Manmatha, R., Lavrenko, V.: Multiple Bernoulli relevance models for image and video annotation. In: CVPR (2004)
10. Frome, A., Corrado, G., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: A deep visual-semantic embedding model. In: Neural Information Processing Systems (NIPS) (2013)
11. Gong, Y., Jia, Y., Leung, T.K., Toshev, A., Ioffe, S.: Deep convolutional ranking for multilabel image annotation. In: ICLR (2014)
12. Guillaumin, M., Mensink, T., Verbeek, J., Schmid, C.: TagProp: Discriminative metric learning in nearest neighbour models for image auto-annotation. In: ICCV (2009)
13. Guo, H., Zheng, K., Fan, X., Yu, H., Wang, S.: Visual attention consistency under image transforms for multi-label image classification. In: CVPR. pp. 729–739 (2019)
14. Hariharan, B., Zelnik-Manor, L., Vishwanathan, S.V.N., Varma, M.: Large scale max-margin multi-label classification with priors. In: ICML (2010)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
16. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(9), 1735–1780 (Nov 1997). https://doi.org/10.1162/neco.1997.9.8.1735, `http://dx.doi.org/10.1162/neco.1997.9.8.1735`
17. Hu, H., Zhou, G.T., Deng, Z., Liao, Z., Mori, G.: Learning structured inference neural networks with label relations. In: CVPR (2016)
18. Jin, J., Nakayama, H.: Annotation order matters: Recurrent image annotator for arbitrary length image tagging. In: ICPR (2016)
19. Joachims, T.: Optimizing search engines using clickthrough data. In: KDD (2002)
20. Johnson, J., Ballan, L., Fei-Fei, L.: Love thy neighbors: Image annotation by exploiting image metadata. In: ICCV (2015)

21. Johnson, J., Karpathy, A., Fei-Fei, L.: Densecap: Fully convolutional localization networks for dense captioning. CVPR pp. 4565–4574 (2015)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25, pp. 1097–1105 (2012)
23. Lan, T., Mori, G.: A max-margin riffled independence model for image tag ranking. In: Computer Vision and Pattern Recognition (CVPR) (2013)
24. Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures. In: NIPS (2003)
25. Li, C., Liu, C., Duan, L., Gao, P., Zheng, K.: Reconstruction regularized deep metric learning for multi-label image classification. IEEE transactions on neural networks and learning systems (2019)
26. Li, L., Wang, S., Jiang, S., Huang, Q.: Attentive recurrent neural network for weak-supervised multi-label image classification. In: ACM Multimedia. pp. 1092–1100 (2018)
27. Li, Q., Qiao, M., Bian, W., Tao, D.: Conditional graphical lasso for multi-label image classification. In: CVPR (2016)
28. Li, X., Zhao, F., Guo, Y.: Multi-label image classification with a probabilistic label enhancement model. In: Proc. Uncertainty in Artificial Intell (2014)
29. Li, Y., Song, Y., Luo, J.: Improving pairwise ranking for multi-label image classification. In: CVPR (2017)
30. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnic, C.L.: Microsoft COCO: Common objects in contex. In: ECCV (2014)
31. Liu, F., Xiang, T., Hospedales, T.M., Yang, W., Sun, C.: Semantic regularisation for recurrent image annotation. In: CVPR (2017)
32. Liu, Y., Sheng, L., Shao, J., Yan, J., Xiang, S., Pan, C.: Multi-label image classification via knowledge distillation from weakly-supervised detection. In: ACM Multimedia. pp. 700–708 (2018)
33. Makadia, A., Pavlovic, V., Kumar, S.: A new baseline for image annotation. In: ECCV (2008)
34. Miller, G.A.: Wordnet: a lexical database for english. Communications of the ACM (CACM) **38**(11), 39–41 (1995)
35. Murthy, V.N., Maji, S., Manmatha, R.: Automatic image annotation using deep learning representations. In: ICMR (2015)
36. Niu, Y., Lu, Z., Wen, J.R., Xiang, T., Chang, S.F.: Multi-modal multi-scale deep learning for large-scale image annotation. IEEE Transactions on Image Processing **28**, 1720–1731 (2017)
37. Rasiwasia, N., Pereira, J.C., Coviello, E., Doyle, G., Lanckriet, G.R.G., Levy, R., Vasconcelos, N.: A new approach to cross-modal multimedia retrieval. In: ACM MM (2010)
38. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015)
39. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
40. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
41. Tsai, C.P., yi Lee, H.: Adversarial learning of label dependency: A novel framework for multi-class classification. ICASSP pp. 3847–3851 (2019)
42. Uricchio, T., Ballan, L., Seidenari, L., Bimbo, A.D.: Automatic image annotation via label transfer in the semantic space. CoRR **abs/1605.04770** (2016)

43. Verma, Y., Jawahar, C.V.: Image annotation using metric learning in semantic neighbourhoods. In: ECCV (2012)
44. Verma, Y., Jawahar, C.V.: Exploring SVM for image annotation in presence of confusing labels. In: BMVC (2013)
45. Verma, Y., Jawahar, C.V.: Image annotation by propagating labels from semantic neighbourhoods. Int. J. Comput. Vision **121**(1), 126–148 (2017)
46. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: CVPR (2015)
47. Wang, C., Blei, D., Fei-Fei, L.: Simultaneous image classification and annotation. In: Proc. CVPR (2009)
48. Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., Xu, W.: CNN-RNN: A unified framework for multi-label image classification. In: CVPR (2016)
49. Weston, J., Bengio, S., Usunier, N.: WSABIE: Scaling up to large vocabulary image annotation. In: IJCAI (2011)
50. Yazici, V.O., Gonzalez-Garcia, A., Ramisa, A., Twardowski, B., van de Weijer, J.: Orderless recurrent models for multi-label classification. CoRR **abs/1911.09996** (2019)
51. You, R., Guo, Z., Cui, L., Long, X., Bao, Y., Wen, S.: Cross-modality attention with semantic graph embedding for multi-label classification. CoRR **abs/1912.07872** (2019)
52. Zhu, F., Li, H., Ouyang, W., Yu, N., Wang, X.: Learning spatial regularization with image-level supervisions for multi-label image classification. In: CVPR. pp. 2027–2036 (2017)
53. Zhuang, Y., Yang, Y., , Wu, F.: Mining semantic correlation of heterogeneous multimedia data for cross-media retrieval. IEEE Transactions on Multimedia **10**(2), 221–229 (2008)