Learning a Recurrent Residual Fusion Network for Multimodal Matching

Yu Liu Yanming Guo Erwin M. Bakker Michael S. Lew
LIACS Media Lab, Leiden University, Leiden, The Netherlands
{y.liu, y.guo, e.m.bakker, m.s.lew}@liacs.leidenuniv.nl

Abstract

A major challenge in matching between vision and language is that they typically have completely different features and representations. In this work, we introduce a novel bridge between the modality-specific representations by creating a co-embedding space based on a recurrent residual fusion (RRF) block. Specifically, RRF adapts the recurrent mechanism to residual learning, so that it can recursively improve feature embeddings while retaining the shared parameters. Then, a fusion module is used to integrate the intermediate recurrent outputs and generates a more powerful representation. In the matching network, RRF acts as a feature enhancement component to gather visual and textual representations into a more discriminative embedding space where it allows to narrow the cross-modal gap between vision and language. Moreover, we employ a bi-rank loss function to enforce separability of the two modalities in the embedding space. In the experiments, we evaluate the proposed RRF-Net using two multi-modal datasets where it achieves state-of-the-art results.

1. Introduction

Nowadays, multimedia data in various media types (e.g. image, video, text, and audio) is growing exponentially due to the increasing popularity of the Internet and social networks. These heterogeneous data offers us the opportunity to understand the world better, while giving rise to the challenges of understanding and bridging the semantic gap between multiple modalities. Specifically, the matching problem between images and texts [42, 32, 18, 25, 39, 40] is one of the most important tasks in the area of multi-modal research. This task remains challenging due to the heterogenous representations and the cross-modal gap between vision and language, which is also a core issue for other multi-modal tasks such as image captioning [15, 27, 38] and visual question answering [2, 26, 31].

A main line of research for multi-modal matching is to learn a latent embedding space where related images and texts can be unified into similar representations [5, 6, 14]. Particularly, Canonical Correlation Analysis (CCA) [11] has been a well-tested and representative embedding technique for decades. It learns a linear transformation to project two modalities into a common space where their correlations are maximized. Also, some extensive techniques are applied to the classical CCA, such as randomized CCA [30], nonparametric CCA [28], and kernel CCA [7].

Driven by the successful developments of deep learning, more and more works extract powerful visual and textual features using deep neural networks. For example, recent works [18, 32, 15, 20, 39, 25] employ convolutional neural networks (CNNs) [19] to extract deep image features, and capture descriptive text features based on recurrent neural networks (RNNs) [35]. Then they incorporate deep learning features with traditional embedding techniques (e.g. CCA and its variants). Also, extensive research efforts [1, 42] have been dedicated to directly learning a deep CCA model that can be end-to-end trainable. Instead of using CCA, recent works developed a variety of multi-modal deep neural networks to model the matching task [14, 15, 25, 39, 4]. Nevertheless, the multi-modal matching performance is still far from competitive with that of the intra-modal tasks, such as image retrieval and machine translation. One key issue is: *how to improve latent embeddings to unify images and texts into a more discriminative space?*

To address this issue, we propose a deep matching network using recurrent residual fusion (RRF) as building blocks for improving feature embeddings. This new matching network (RRF-Net) has two branches for representing images and texts, respectively. Each branch consists of four fully-connected layers that are used to project a source representation into a common latent space. The proposed RRF building block (in Fig. 2) is introduced in the third fully-connected layer of the two branches. Importantly, RRF integrates three main components to improve the feature embedding procedure in the network.

The first component in RRF is inspired by the residual learning in ResNet [9]. We add an identity connection to sum the input of a fully-connected layer with its output. This component enables the fully-connected layer to learn a residual embedding feature and provides high performance.

Secondly, RRF employs a recurrent mechanism to residual learning by adding a recurrent connection whose direction is inverse to the identity connection. As the parameters of the fully-connected layer are shared during the recurrent procedure, RRF is able to recurrently improve feature embeddings while retaining the parameters. The third component is the use of a fusion module, which aims to integrate intermediate recurrent outputs to generate a more powerful fused output. The fusion module facilitates making use of more complementary information in the intermediate layers and explicitly transferring their effects to the final output. We provide two efficient fusion modules: sum-pooling fusion and convolutional fusion.

Moreover, we present a bi-directional rank loss function (called bi-rank loss), including image-to-text rank loss and text-to-image rank loss, to train the proposed RRF-Net. The original bi-directional loss function only considers the cross-modal relationships between images and texts. Instead, the bi-rank loss can preserve both of the cross-modal and intra-modal relationships (e.g. images-images and texts-texts). As a result, it is able to enforce separability of the two modalities in a unified embedding space. Extensive experiments show remarkable improvements of the bi-rank loss over the original bi-directional loss.

The main contributions of our work can be summarized as follows: (1) We introduce a new RRF building block and adapt it to a deep matching network. RRF provides promising insights into efficiently improving the co-embedding between images and texts. (2) We present a bi-rank loss function to train the RRF-Net for better ensuring the cross-modal and intra-modal constraints in the unified space. (3) The experimental results demonstrate that our approach achieves state-of-the-art performance on public benchmarks for image-to-text and text-to-image retrieval.

2. Related work

In this section, we review the related works and discuss our differences from them.

Image-text matching. CCA [11] and its variants [7, 30, 28] that are based on computing the cross-covariance matrix between two modalities, are able to learn a pair of linear or nonlinear transformations to maximize the cross-modal correlations. Inspired by the powerful generalization of CCA, many research approaches based on CCA were proposed to improve image and text matching. For example, based on the two-view CCA, Gong et al. [5] captured a third view from high-level image semantics to provide a better separability between the classes. Similarly, Ranjan et al. [32] proposed a multi-label CCA approach while learning the cross-modal subspaces. Instead of using hand-crafted kernels in KCCA [7], Andrew et al. [1] developed a deep CCA model to directly learn a flexible nonlinear kernel. Yan [42] alleviated the complexity and overfitting issues while training deep CCA. Our work is different from prior work focused on CCA, but is related to recent deep matching networks [15, 39, 25], which aim to search for a latent unified space where related images and texts are gathered by minimizing a rank loss function. Ma et al. [25] proposed multi-modal CNN for matching images and sentences. Karpathy et al. [15] aligned visual regions and sentences by integrating CNN and RNN. Wang et al. [39] learned a deep structure-preserving embedding in a simple yet efficient network.

Deep fusion networks. Considering complementary representations from intermediate layers in deep networks, multi-layer (or multi-scale) fusion approaches have been well-studied and applied in many vision tasks, such as image-level classification [43, 23] and pixel-level prediction [24, 41]. However, few works investigated the use of deep fusion networks for image and text matching. In this work, our fusion network is able to integrate the intermediate outputs of recurrent residual learning. The residual learning in ResNet [9] has shown its great potential to learn very deep neural networks. Many variants [8, 46, 12, 37] presented more insights into the residual learning mechanism. Also, other works succeeded to apply the residual learning mechanism to diverse vision tasks [3, 16].

Although the general idea of using residual blocks in RNNs has been studied in recent works [21, 44] for image classification and image super-resolution, our proposed RRF that acts as a deep fusion model aims to explore the intermediate features in recurrent residual learning. In addition, we leverage RRF to develop an image-text matching network for improving their latent embedding.

3. Recurrent residual fusion

In this section, we describe the details of building a RRF block (in Fig. 2) with three components: an identity connection, a recurrent connection and a fusion module.

3.1. Identity connection

The basic building block in ResNet [9] adds an extra identity mapping with the traditional non-linear transformations based on convolutional layers. Instead of using a convolutional layer, we develop an identity connection based on a fully-connected layer. As can be seen in Fig. 1(a), our residual block consists of a fully-connected layer (FC), a batch normalization layer (BN) [13] and a Rectified Linear Unit (ReLU) layer [19]. The input and output channels of the FC layer should have the same size. The computation can be presented by

$$h(x) = \sigma(f(x)) + x, \tag{1}$$

where x and $h(x)$ represent the input and output of the building block, respectively. The function $f(\cdot)$ indicates the FC layer, and $\sigma(\cdot)$ is the ReLU activation function.

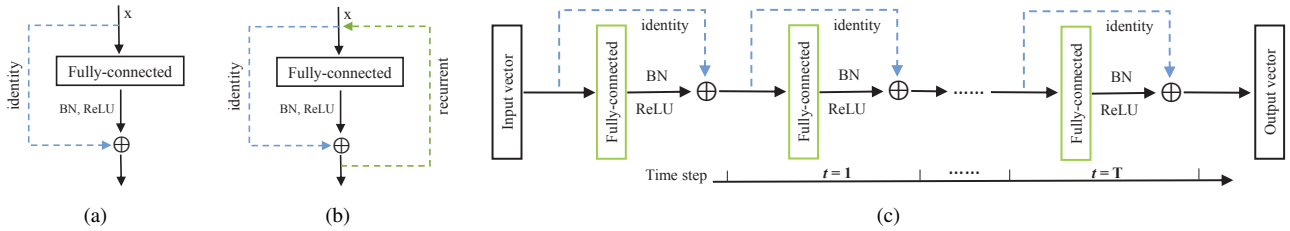


Figure 1: Illustration of basic building blocks. (a) An identity mapping (blue) is added on a fully-connected layer. (b) A recurrent connection (green) is introduced that uses the current output state to update the next input state. (c) We unfold the building block in (b) over recurrent steps, resulting in a very deep network. All fully-connected layers (in green) share the same parameters. t represents the recurrent step, ranging from 1 to T.

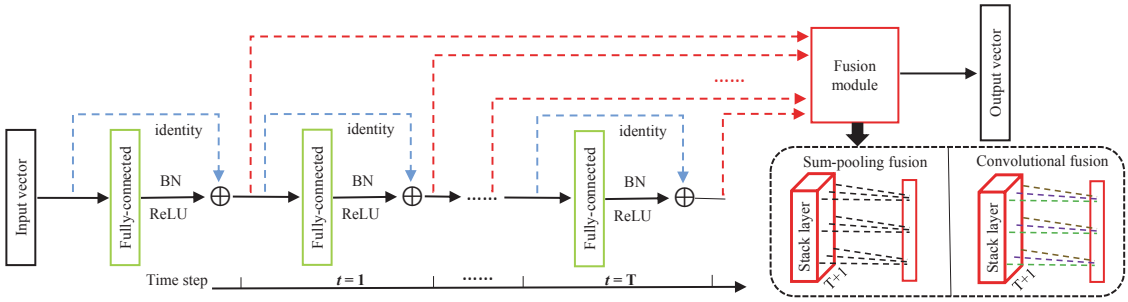


Figure 2: The RRF building block. Built upon recurrent residual learning, we develop a fusion module (in red) to integrate the intermediate output vectors from each recurrent step. The final output vector learns more information than the original output in Fig. 1(c). Specifically, there are two types of fusion modules: the sum-pooling fusion simply fixes equal weights, but the convolutional fusion can learn adaptive weights (drawn in different colors).

3.2. Recurrent connection

RNNs [10, 35] are widely-used for modeling sequential contexts in tasks like machine translation and image captioning. We seek to introduce the recurrent mechanism to the residual learning block. As can be seen in Fig. 1(b), we add a recurrent connection whose direction is inverse to the identity connection. As a result, the current output can be used as the next input, and then the next input continues adding an identity mapping to the residual mapping to compute the next output. As the fully-connected parameters are shared during the recurrent procedure, the whole structure is able to become much deeper without consuming more parameters. We unfold the structure across recurrent steps in Fig. 1(c). Assume that there are T recurrent steps in total, so the structure has $T+1$ layers inside, and each layer uses the same parameters as drawn in green. Mathematically, the recurrent residual procedure is formulated via

$$x_t = h(x_{t-1}) \tag{2}$$

$$f(x_t) = w \cdot x_t + b \tag{3}$$

$$h(x_t) = \sigma(f(x_t)) + x_t \tag{4}$$

where $t = 1, \ldots, T$ and $x_0 = x$ is the original input vector. x_t is updated by the previous output $h(x_{t-1})$ which adds the residual mapping $f(x_t)$ with the identity mapping x_t. The parameters w, b indicate the shared weights and bias in the fully-connected layer. Note that the parameters used in the BN layer are not shared during recurrence, however, the number of these parameters is much lower than that of the total parameters in the model. The input vector can be refined over recurrence while maintaining the efficiency due to tying the shared parameters. Finally, the output vector learns to be a more discriminative representation.

3.3. Fusion module

Typically, a plain network can learn multiple representations from bottom layers to top layers, however, the final output only connects with the topmost layer. For example in Fig. 1(c), the output vector is directly affected by the result at the last recurrent step. Although the recurrent procedure can transfer the effects of intermediate layers to the final output, their effects are implicit and indirect compared with the topmost layer. Therefore, we develop a fusion module to explicitly aggregate the intermediate layers involved in the recurrent procedure. Figure 2 highlights the fusion module in red. Specifically, several new side branches (dot lines in red) are generated from intermediate layers and then merged into a fusion module. As the intermediate layers have the same dimension, the fusion module is able to integrate them without adding extra new transition layers. In a fusion module, $T+1$ side outputs are stacked as a layer S. S has $1 \times N \times (T+1)$ size, where N is the dimension of each side output. Based on S, we employ two fusion methods to compute a fused output vector: sum-pooling fusion and convolutional fusion.

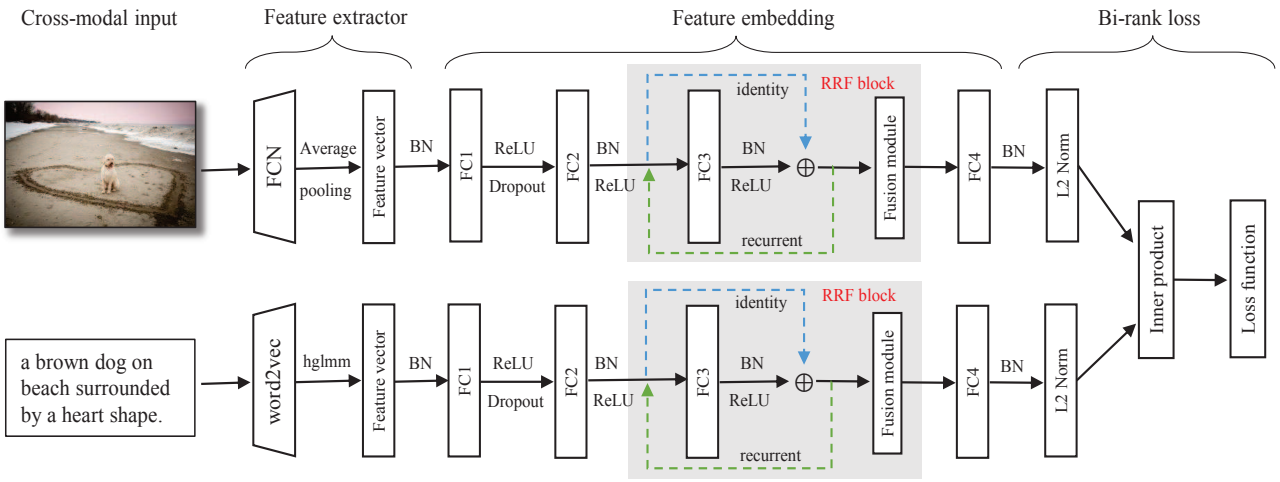


Figure 3: The overview architecture of the proposed RRF-Net for image and text matching. This two-branch network comprises three key steps: (1) feature extractors are used for capturing visual and textual representations. (2) Four fully-connected layers (from FC1 to FC4) in two branches are used for learning feature embeddings. Importantly, a RRF block is built upon the FC3 layer to improve its embedding capability. The details inside the RRF block are described in Fig. 2. (3) After normalizing the two output vectors and computing their inner product, we employ a bi-rank loss to train the entire network.

Sum-pooling fusion. As can be seen in the right bottom of Fig. 2, it computes a summation across the feature channels of the stack layer S. The fused output vector S_{sum} is represented by

$$S_{sum} = \sum_{i=0}^{T} h(x_i) = \sum_{i=0}^{T} \sigma(f(x_i)) + x_i. \tag{5}$$

The sum-pooling fusion supposes that each side branch has the same importance without learning any weights.

Convolutional fusion. Normally, each side branch (or intermediate layer) may have different important influence on the output vector. Therefore, we use a convolutional layer in the fusion module to learn adaptive weights (or importance) for better fusing side branches. The filter f in the convolutional layer has $1 \times 1 \times (T+1)$ dimensions. S is convolved by f to generate the fused vector S_{conv}:

$$S_{conv} = w_f * S + b_f \tag{6}$$

where w_f and b_f represent the weights and bias, respectively. It is worth noting that these additional parameters (i.e. $T+1$) are a minimal increase to the total parameters used in a deep network.

In summary, the RRF block incorporate the above three components (Sec. 3.1, 3.2, 3.3) and inherits their individual advantages. It acts as a feature enhancement to the power of the input vector and aims to generate a more informative output vector. Different from other deep fusion networks in which different layers are aggregated, RRF delves into improving the discrimination of one layer over recurrence. Also, RRF is a general structure that can be potentially applied to many existing layers in a deep network.

4. Matching network

In this section, we present a new deep matching network called RRF-Net, where the RRF blocks are introduced to improve latent embeddings between images and texts. Figure 3 illustrates the architecture of the network, and we will describe its three key steps as below.

4.1. Feature extractor

As a common practice, we capture visual and textual features using off-the-shelf feature extractors. Taking these features as input instead of the raw data can ease the training procedure and lead to fast convergence.

Image feature extractor: we choose the powerful ResNet-152 [9] pre-trained on ImageNet [33]. To efficiently extract dense region representations, CNN models are first recast to fully convolutional networks (FCNs) [24]. Given one input image, we set its smaller side to 512 and isotropically resize the other side. The last max-pooling layer in the ResNet-152 model is averaged to generate a 2048-dimensional visual feature vector.

Text feature extractor: we employ the Hybrid Gaussian-Laplacian mixture model (HGLMM) [18] which is built based on word2vec model [29]. For each sentence, HGLMM computes one 18000-dimensional vector with 30 centers (i.e. 300*30*2). To decrease the memory cost [39], we also use PCA to reduce the dimension from 18000 to 6000. Finally, the 6000-dimensional vector acts as a powerful sentence representation.

4.2. Feature embedding

To learn a discriminative embedding space, we develop four fully-connected layers on top of the two feature ex-

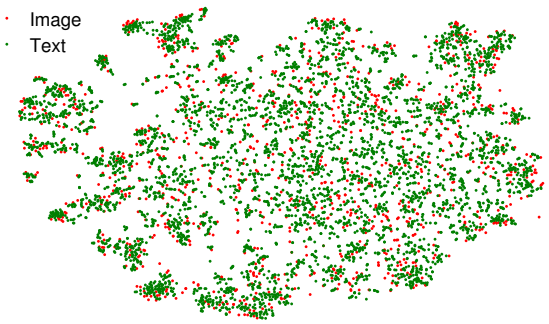


Figure 4: Visualization of our embedding on the Flickr30K test set (1000 images and 5000 texts).

tractors. Their channels are {2048, 512, 512, 512} in both two branches. Note that the parameters in each branch are unshared as they are responsible for different modalities. Specifically, ReLU is used for FC1, FC2 and FC3, but not for FC4. A dropout layer with 0.5 probability is added after FC1, and other FC layers are regularized with BN [13].

The core component in each branch is the FC3 layer as it introduces the RRF building block. RRF increases the FC3 layer to $T+1$ depth while retaining the parameters. Consequently, it facilitates deeper learning of latent embeddings and further unifies the visual and textual representations. Notably, the BN layer after FC3 learns unshared parameters during recurrent steps, however, these few extra parameters raise little cost to the entire network. Moreover, a RRF block can be imposed on any fully-connected layer. But in the current architecture, FC3 is more suitable than other layers. Also, we observe that using only a RRF block seems sufficient for enhancing feature embeddings.

In Fig. 4, we use the t-SNE algorithm [36] to visualize the embedding features learned in the RRF-Net (with $T = 3$). Since an image has five ground-truth matching texts, we can see that each image is surrounded by several texts. This example shows that RRF-Net is potential to align the distributions of images and texts and to preserve their intrinsic correlation.

4.3. Bi-rank loss

After unifying images and texts into a joint embedding space, the next step is to compare their similarities. Given an image x and a text y, their FC4 embedding features are denoted as $f(x)$ and $f(y)$. We compute the similarity $s(x, y)$ with the cosine distance

$$s(x,y) = 1 - \frac{f(x) \cdot f(y)}{||f(x)|| \cdot ||f(y)||}. \tag{7}$$

Smaller distances indicate larger similarities. To train the network, we define a bi-rank loss function, including image-to-text rank loss and text-to-image rank loss.

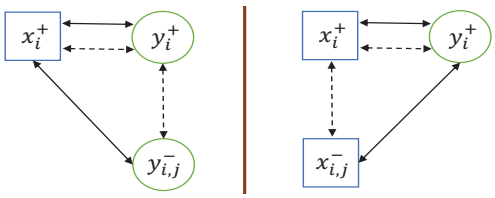


Figure 5: Illustration of computing bi-rank loss. Left: image-to-text rank loss; Right: text-to-image rank loss.

Image-to-text rank loss. For an input image x_i^+, its matching text is represented by y_i^+. To obtain more representative non-matching pairs, we collect the top N most dissimilar texts in each mini-batch as a negative text set Y_i^-. Then, we compute the triplet rank loss for $\{x_i^+, y_i^+, y_{i,j}^-\}$, where $y_{i,j}^- \in Y_i^-$ and $j = 1, 2, \ldots, N$. First, the matching cross-modal similarity $s(x_i^+, y_i^+)$ should be larger than any of the non-matching cross-modal similarities $s(x_i^+, y_{i,j}^-)$. Second, we further constrain the intra-modal similarity $s(y_i^+, y_{i,j}^-)$ from exceeding $s(x_i^+, y_i^+)$. This loss can ensure both the cross-modal (i.e. image-text) and the intra-modal (i.e. text-text) relations. An example is shown in the left of Fig. 5. Finally, this loss function is expressed with

$$\begin{aligned} l_{i2t} = \sum_{j=1}^{N} \Big(& \alpha_1 \max[0, s(x_i^+, y_i^+) - s(x_i^+, y_{i,j}^-) + m] \\ & + \alpha_2 \max[0, s(x_i^+, y_i^+) - s(y_i^+, y_{i,j}^-) + m] \Big), \end{aligned} \tag{8}$$

where α_1 and α_2 measure the importance of the two terms. m is a margin parameter.

Text-to-image rank loss. Given one text y_i^+, we collect its top N most dissimilar images in each mini-batch as a negative image set X_i^-. Similarly, we compare the similarities within each triplet $\{y_i^+, x_i^+, x_{i,j}^-\}$, where $x_{i,j}^- \in X_i^-$. Their relations can be seen in the right of Fig. 5. The text-to-image rank loss can be computed by

$$\begin{aligned} l_{t2i} = \sum_{j=1}^{N} \Big(& \alpha_1 \max[0, s(y_i^+, x_i^+) - s(y_i^+, x_{i,j}^-) + m] \\ & + \alpha_2 \max[0, s(y_i^+, x_i^+) - s(x_i^+, x_{i,j}^-) + m] \Big), \end{aligned} \tag{9}$$

Overall loss. Our bi-rank loss adds the above two rank loss functions together by

$$l(x_i^+, y_i^+, X_i^-, Y_i^-) = \frac{\beta_1 l_{i2t} + \beta_2 l_{t2i}}{N}, \tag{10}$$

where the weights β_1 and β_2 control the importance of the two terms of one-directional rank loss. Compared with [39] which searches for extra positive intra-modal pairs, our bi-rank loss directly uses the negative intra-modal pairs, and therefore has minimal additional computation.

Table 1: Evaluation for the RRF-Net on the Flickr30K test set. Higher R@K is better. All of the four RRF-Net models outperform the baseline. When $T = 3$, it obtains better performance (in bold).

	Image to Text		Text to Image	
Method	R@1	R@5	R@1	R@5
Baseline	45.0	75.5	33.6	66.5
RRF-Net, T=1	46.4	76.1	34.3	67.3
RRF-Net, T=2	46.9	76.8	34.8	67.7
RRF-Net, T=3	**47.6**	**77.4**	**35.4**	**68.3**
RRF-Net, T=4	46.2	76.6	35.1	67.6

Table 2: Evaluation for fusion modules on the Flickr30K test set. The convolutional fusion shows better results by learning adaptive weights.

	Image to Text		Text to Image	
Method	R@1	R@5	R@1	R@5
RRF-Net w/o fusion module	45.8	75.9	34.2	67.1
RRF-Net with sum fusion	47.1	76.8	35.0	67.6
RRF-Net with conv fusion	47.6	77.4	35.4	68.3

5. Experiments

In this section, we evaluate our approach and report its results on two widely-used multi-modal datasets for bi-directional image-text retrieval.

5.1. Datasets

Flickr30K [45]: following the dataset splits in [27], we use 29783 training images, 1000 validation images and 1000 test images. Each image is annotated by five sentence-level texts. It has $29783 * 5 = 148915$ training pairs.

MSCOCO [22]: it consists of 82783 training images and 40504 validation images. 1000 test images are selected from the validation set [27]. We choose five sentences for each image and generate $82783 * 5 = 413915$ training pairs.

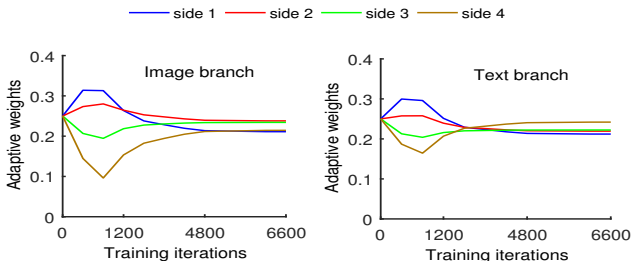
5.2. Implementation details

The hyper-parameters are evaluated on the validation set of each dataset. To be more specific, the parameters $\{\alpha_1, \alpha_2, \beta_1, \beta_2\}$ are set with $\{1, 0.5, 2, 1\}$, and $m = 0.1$. Following [39], the number of non-matching pairs is $N = 50$. We trained the model with a weight decay of 0.0005, a momentum of 0.9, and a mini-batch size of 1500. The learning rate was initialized with 0.1 and is divided by 10 when the decrease in loss stabilizes. It is necessary to shuffle the training samples randomly.

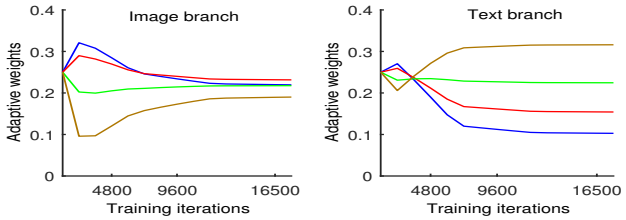
Baseline: it uses the same 4-layer plain network in Fig. 3 but excludes the RRF block from the FC3 layer. We employed the same hyper-parameters for training the RRF-Net model and the baseline model.

5.3. Results

For image-text retrieval, we adopted the evaluation metric R@K which is the recall rate of a correctly retrieved ground-truth at top K candidates (e.g. $K = 1, 5, 10$) [15].



(a) Flickr30K



(b) MSCOCO

Figure 6: Analysis of adaptive weights learned in the convolutional fusion. The weights in the image and text branches are shown separately. All weights tend to be stable during the training stage.

Evaluation for the RRF-Net. In Table 1, we show the results of four RRF-Net models with $T = 1, 2, 3, 4$ (here we use the convolutional fusion). Compared with the baseline model, all four RRF-Net models achieved considerable improvements. This verifies the effectiveness of imposing RRF blocks in a deep matching network. We can observe that, the results when $T = 3$ are superior to other time steps. The drop of performance from T=3 and T=4 may be due to the potential overfitting in the model. It shows a trade-off between the number of recurrent steps and the test performance. The following experiments are performed with $T = 3$. Nevertheless, we believe that evaluating more recurrent steps is still promising in future research. The first and second columns in Fig. 7 compare the retrieval examples between the baseline and the RRF-Net.

Evaluation for fusion modules. Recall that we define two types of fusion modules. Table 2 compares their quantitative results. First, we trained a RRF-Net model without using any fusion module, which is actually a recurrent residual model in Fig. 1(c). By comparison, we can see that using fusion modules can achieve remarkable improvements. This evaluation reveals the benefit of integrating the intermediate recurrent layers. Moreover, the advantage of the sum-pooling fusion is that it is parameter-free, however, the convolution fusion yields better results than the sum-pooling fusion due to learning adaptive weights. In the following, we implemented the RRF-Net model with the convolutional fusion.

Furthermore, we analyzed $T + 1$ adaptive weights learned in the convolutional fusion module. When $T = 3$,

Table 3: Comparison with the state-of-the-art results. The best and second results are in bold and underline, respectively.

	Flickr30K dataset						MSCOCO dataset					
	Image to Text			Text to Image			Image to Text			Text to Image		
Method	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
DVSA [15]	22.2	48.2	61.4	15.2	37.7	50.5	38.4	69.9	80.5	27.4	60.2	74.8
SC-NLM [17]	23.0	50.7	62.9	16.8	42.0	56.5	NA	NA	NA	NA	NA	NA
Mean vector [18]	24.8	52.5	64.3	20.5	46.3	59.3	33.2	61.8	75.1	24.2	56.4	72.4
Deep CCA [42]	27.9	56.9	68.2	26.8	52.9	66.9	NA	NA	NA	NA	NA	NA
GMM+HGLMM [18]	35.0	62.0	73.8	25.0	52.7	66.0	39.4	67.9	80.9	25.1	59.8	76.6
m-RNN [27]	35.4	63.8	73.7	22.8	50.7	63.1	41.0	73.0	83.5	29.0	42.2	77.0
RNN-FV [20]	35.6	62.5	74.2	27.4	55.9	70.0	41.5	72.0	82.9	29.2	64.7	80.4
mCNN(ensemble) [25]	33.6	64.1	74.9	26.2	56.3	69.6	42.8	73.1	84.1	32.6	68.6	82.8
DSPE [39]	40.3	68.9	79.9	29.7	60.1	72.1	50.1	79.7	89.2	39.6	75.2	86.9
2WayNet [4]	**49.8**	67.5	NA	**36.0**	55.6	NA	55.8	75.2	NA	39.7	63.3	NA
RRF-Net	47.6	**77.4**	**87.1**	35.4	**68.3**	**79.9**	**56.4**	**85.3**	**91.5**	**43.9**	**78.1**	**88.6**

Table 4: Comparison between the bi-rank loss and the original bi-directional loss on the Flickr30K test set.

	Image to Text		Text to Image	
Method	R@1	R@5	R@1	R@5
Baseline, bi-directional	43.4	73.8	32.5	65.4
Baseline, bi-rank	45.0	75.5	33.6	66.5
RRF-Net, bi-directional	46.4	76.5	34.1	67.4
RRF-Net, bi-rank	47.6	77.4	35.4	68.3

Table 5: Model ensemble results on the Flickr30K test set. Merging more models is significant to obtain better results.

	Image to Text		Text to Image	
Method	R@1	R@5	R@1	R@5
RRF-Net, $M = \{3\}$	47.6	77.4	35.4	68.3
RRF-Net, $M = \{1, 3\}$	49.1	78.4	36.8	69.8
RRF-Net, $M = \{1, 2, 3\}$	50.3	79.2	37.4	70.4
RRF-Net, $M = \{1, 2, 3, 4\}$	**50.8**	**79.5**	**37.6**	**70.9**

four weights are learned for the side branches which are called side 1, side 2, side 3 and side 4 here. Figure 6 visualizes the changes of the four weights during the training iterations on Flickr30K and MSCOCO. They were initialized with a equal weight that is 0.25. We can see that these weights fluctuate significantly in the early training stage, but tend to be stable later. These results provide deeper insights towards the convolutional fusion module. Each side branch has its individual contribution to the whole network.

Evaluation for the bi-rank loss. Table 4 presents the quantitative comparison between the bi-rank loss and the original bi-directional loss. Actually, the original bi-directional loss is a specific case of the bi-rank loss. We implemented the bi-directional loss by setting $\{\alpha_1, \alpha_2, \beta_1, \beta_2\}$ with $\{1, 0, 2, 0\}$. The baseline and RRF-Net models are both evaluated in this test. In summary, it can be seen that the bi-rank loss brings about 1% performance improvements compared with the bi-directional loss.

5.4. Comparison with the state-of-the-art

We compared our results with the state-of-the-art approaches in Table 3. Overall, RRF-Net achieves competitive (and often better) performance on both Flickr30K and MSCOCO datasets. On the FLICKR30K dataset, DSPE [39] and 2WayNet [4] lead recent state-of-the-art results. Although 2WayNet has the best R@1 results on Flickr30K, the proposed RRF-Net significantly outperforms it on the R@5 accuracy. Additionally, our approach on MSCOCO outperforms the top state-of-the-art approaches.

Recall that we used the ResNet-152 model to extract visual features. To provide more comparison, we were also curious about the performance when using another well-known CNN: VGG-19 [34]. For Flickr30K, RRF-Net yields R@1=42.1 and 31.2 for image-to-text and text-to-image retrieval, respectively. This was not as high as the proposed RRF-Net performance, but still higher than DSPE [39]. Therefore, RRF-Net presents consistently high performance for diverse feature extractors.

5.5. Model ensemble

Although the performance of different RRF-Net models varies, it is beneficial to integrate the retrieved results from multiple models at the test stage. To integrate the strengths of individual RRF-Net models, we employ a simple yet efficient ensemble approach by computing the averaged similarity $s^{'}(x, y)$ given a test pair (x, y):

$$s^{'}(x, y) = \frac{\sum_{m \in M} s^m(x, y)}{|M|}, \tag{11}$$

where M is the index set. $s^m(x, y)$ is the similarity computed by the RRF-Net model with $T = m$. For example when $M = \{1, 3\}$, the model ensemble merges the RRF-Net models with $T = 1$ and $T = 3$. As reported in Table 5, merging the four models (i.e. $M = \{1, 2, 3, 4\}$) together can significantly improve the performance compared with the single RRF-Net model (i.e. $M = \{3\}$). This ensemble approach can refine the retrieved candidates

Table 6: Results on cross-dataset generalization between the Flickr30K and MSCOCO datasets.

		Image to Text			Text to Image		
Cross-dataset	Method	R@1	R@5	R@10	R@1	R@5	R@10
Train: Flickr30K, Test: MSCOCO	Baseline	23.7	52.0	64.0	17.9	42.8	57.4
	RRF-Net, T=3	24.8	53.0	64.8	18.8	44.1	58.5
	RRF-Net, M=1,2,3,4	**26.4**	**54.3**	**66.2**	**20.3**	**45.6**	**59.8**
Train: MSCOCO, Test: Flickr30K	Baseline	27.2	52.6	65.1	20.4	41.5	52.8
	RRF-Net, T=3	28.8	53.8	66.4	21.3	42.7	53.7
	RRF-Net, M=1,2,3,4	**31.5**	**56.0**	**68.6**	**23.4**	**45.0**	**56.8**

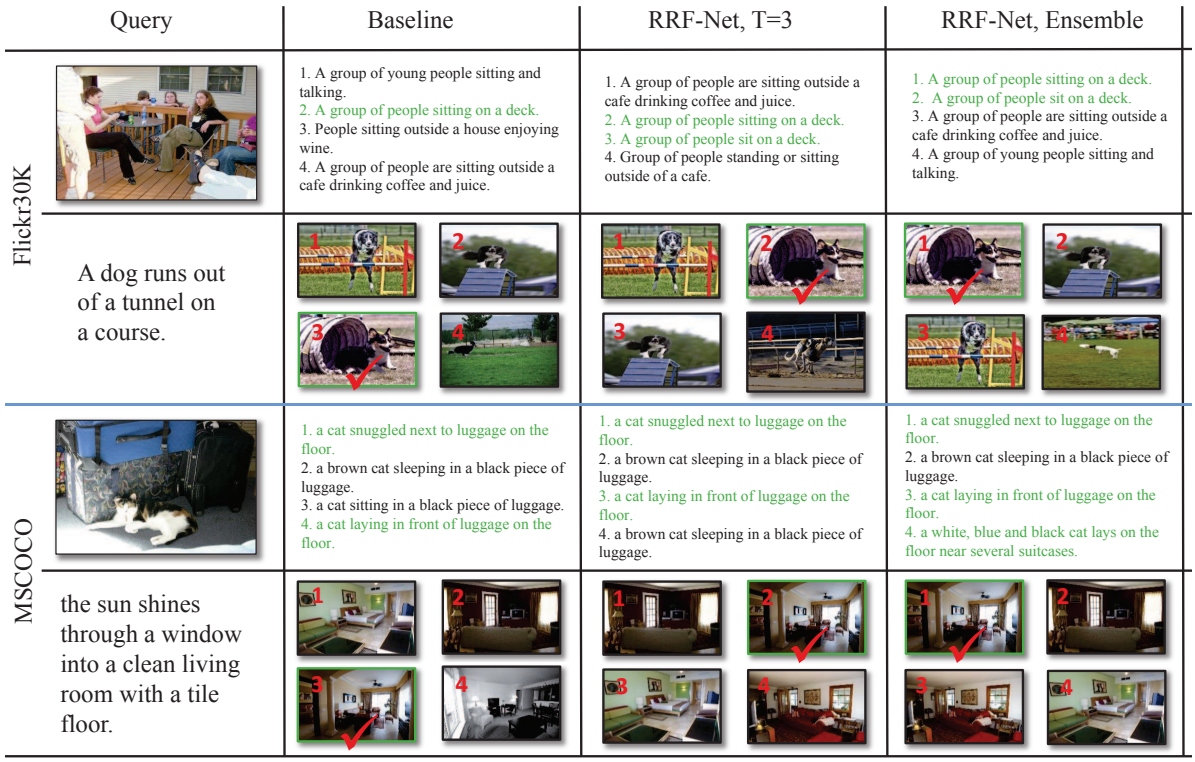


Figure 7: Qualitative results on Flickr30K and MSCOCO. First column: the baseline model; Second column: RRF-Net model with $T = 3$; Third column: the ensemble model with $M = \{1, 2, 3, 4\}$. For image-to-text retrieval, the ground-truth matching texts are in green. For text-to-image retrieval, the red number in the upper left corner of one image is the ranking order, and the green frame corresponds to the ground-truth matching image.

without increasing the training complexity. In Fig. 7, the third column shows its retrieval results.

5.6. Cross-dataset generalization

The cross-dataset generalization of image-text matching models was minimally investigated in prior works. To highlight this important issue, we conducted the cross-dataset experiments between Flickr30K and MSCOCO. Specifically, we trained a model on the Flickr30K training set and evaluated its performance on the MSCOCO 1000 test images, and vice verse. The cross-dataset results are reported in Table 6. We can see that the RRF-Net model with $T = 3$ outperforms the baseline for the two cross-dataset configurations. In addition, the ensemble method can further bring more improvements. The cross-dataset generalization problem still remains challenging, but it will be a particularly promising topic for multi-modal tasks.

6. Conclusion

In this work, we proposed the RRF block and RRF-Net which can bridge the gap between image and text features in a deep matching network. In addition, a bi-rank loss function was presented for enhancing the matching constraints. This work can provide promising insights towards efficiently narrowing the semantic gap between vision and language. Experiments showed that RRF-Net achieved competitive or even state-of-the-art results on Flickr30K and MSCOCO. Since RRF-Net is able to learn discriminative embedding features, it is promising that RRF-Net could be seamlessly integrated into other multi-modal applications like image captioning and VQA.

Acknowledgments This work was supported mainly by the LIACS Media Lab at Leiden University and in part by the China Scholarship Council. We are also grateful to the support of NVIDIA with the donation of GPU cards.

References

[1] G. Andrew, R. Arora, K. Livescu, and J. Bilmes. Deep canonical correlation analysis. In *ICML*, 2013. 1, 2

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. VQA: Visual question answering. In *ICCV*, 2015. 1

[3] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: Object detection via region-based fully convolutional networks. In *NIPS*, 2016. 2

[4] A. Eisenschtat and L. Wolf. Linking image and text with 2-way nets. In *CVPR*, 2017. 1, 7

[5] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *Int. J. Comput. Vision*, 106(2):210–233, 2014. 1, 2

[6] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*, 2014. 1

[7] D. R. Hardoon, S. R. Szedmak, and J. R. Shawe-taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Comput.*, 16(12):2639–2664, 2004. 1, 2

[8] K. He, X. Zhang, S. Ren, , and J. Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 2

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 4

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *NeuralComputing*, 9(8):1735–1780, 1997. 3

[11] H. Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936. 1, 2

[12] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 2

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 2, 5

[14] A. Karpathy, A. Joulin, and F. Li. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014. 1

[15] A. Karpathy and F.-F. Li. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 1, 2, 6, 7

[16] J.-H. Kim, S.-W. Lee, D.-H. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal residual learning for visual qa. In *NIPS*, 2016. 2

[17] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 7

[18] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015. 1, 4, 7

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2

[20] G. Lev, G. Sadeh, B. Klein, and L. Wolf. RNN fisher vectors for action recognition and image annotation. In *ECCV*, 2016. 1, 7

[21] Q. Liao and T. A. Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *CoRR*, abs/1604.03640, 2016. 2

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[23] Y. Liu, Y. Guo, and M. S. Lew. On the exploration of convolutional fusion networks for visual recognition. In *International Conference on MultiMedia Modeling (MMM)*, 2017. 2

[24] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2, 4

[25] L. Ma, Z. Lu, L. Shang, and H. Li. Multimodal convolutional neural networks for matching image and sentence. In *ICCV*, 2015. 1, 2, 7

[26] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015. 1

[27] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *ICLR*, 2015. 1, 6, 7

[28] T. Michaeli, W. Wang, , and K. Livescu. Nonparametric canonical correlation analysis. In *ICML*, 2016. 1, 2

[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 4

[30] P. Mineiro and N. Karampatziakis. A randomized algorithm for cca. In *NIPS workshop*, 2014. 1, 2

[31] H. Noh, P. Hongsuck Seo, and B. Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *CVPR*, 2016. 1

[32] V. Ranjan, N. Rasiwasia, and C. V. Jawahar. Multi-label cross-modal retrieval. In *ICCV*, 2015. 1, 2

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, pages 1–42, 2015. 4

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 7

[35] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 1, 3

[36] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. *JMLR*, 9:2579–2605, 2008. 5

[37] A. Veit, M. Wilber, and S. Belongie. Residual networks behave like ensembles of relatively shallow networks. In *NIPS*, 2016. 2

[38] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 1

[39] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. 1, 2, 4, 5, 6, 7

[40] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, and S. Yan. Cross-modal retrieval with cnn visual features: A new baseline. *IEEE Transactions on Cybernetics*, pages 1–12, 2016. 1

[41] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, 2015. 2

[42] F. Yan and K. Mikolajczyk. Deep correlation for matching images and text. In *CVPR*, 2015. 1, 2, 7

[43] S. Yang and D. Ramanan. Multi-scale recognition with DAG-CNNs. In *ICCV*, 2015. 2

[44] W. Yang, J. Feng, F. Zhao, J. Liu, Z. Guo, and S. Yan. Deep edge guided recurrent residual learning for image super-resolution. *arXiv*, 2016. 2

[45] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. In *ACL*, 2014. 6

[46] S. Zagoruyko and N. Komodakis. Wide residual networks. In *BMVC*, 2016. 2