Exploring images with deep learning for classification, retrieval and synthesis

Liu, Y.

Chapter 5

Image-Text Matching for Cross-modal Retrieval

In the previous chapter, we have started the research theme on image retrieval. Nowadays, cross-modal retrieval using vision and language has drawn increasing attention due to the availability of large-scale multimedia data. This observation motivates our research on how we can develop an efficient deep matching network for cross-modal retrieval (RQ 4).

A major challenge in matching visual and textual representations is that they typically have different modality-specific features based on individual feature encoders. Existing approaches take advantage of the power of deep models to learn a discriminative embedding space where related images and texts can be gathered, however, few of them consider maintaining the model complexity. In this chapter, we introduce an efficient approach to couple visual and textual features based on a recurrent residual fusion (RRF) block. RRF adapts the residual learning to the recurrent mechanism, so that it can recursively improve feature embeddings while retaining the shared parameters. In addition, a fusion module is used to integrate the intermediate recurrent outputs and generate a more powerful representation. In the matching network, RRF can be viewed as a feature enhancement component that gathers visual and textual representations into a more discriminative embedding space. Moreover, we present a bi-rank loss function to enforce separability of the two modalities in the embedding space. In the experiments, we verify the effectiveness of the proposed approach on two multi-modal datasets where it can achieve competitive performance with the state-of-the-art approaches.

Keywords

Cross-modal retrieval, Image-text matching, Deep neural networks, Ranking loss

5.1 Introduction

The matching problem between images and texts [49, 50, 51, 52, 53, 54] is one of the most important tasks in the area of multi-modal information retrieval. 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 applications such as image captioning [55, 56], visual question answering [57, 58] and zero-shot recognition [59, 60].

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 [63, 180, 181]. Previously, Canonical Correlation Analysis (CCA) [61] has been a well-known and representative embedding technique for decades. CCA can learn 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, including randomized CCA [182], nonparametric CCA [183], and kernel CCA [184].

Driven by the successful developments of deep learning, more and more works extract powerful visual and textual features from deep neural networks. For example, recent works [50, 51, 52, 53, 55, 185] employ convolutional neural networks (CNNs) [4] to extract deep image features, and learn descriptive text features based on recurrent neural networks (RNNs) [186]. Then they can incorporate deep learning features with traditional embedding techniques (*e.g.* CCA and its variants). In addition, extensive research efforts [49, 62] 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 [52, 53, 55, 76, 181]. Nevertheless, the performance of multi-modal matching is still far from competitive with that of an intra-modal task like image retrieval. In addition, most of prior works are inefficient with respect to the model complexity. Regarding this task, we aim to address **RQ 4: How can we build a deep matching network to unify images and texts into a more discriminative space without increasing the number of network parameters?**

In this chapter, we propose a deep matching network using recurrent residual fusion (RRF) as building blocks for improving feature embeddings. Our 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 is introduced in the third fully-connected layer of the two branches. Specifically, 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 [10]. 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 residual embedding features and provides high performance. Secondly, RRF employs a recurrent mechanism with the 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 not only cross-modal relationships, but also intra-modal relationships (*e.g.* image-image and text-text). 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 contributions of this work are as follows:

- 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.
- 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.
- The experimental results demonstrate that our approach achieves competitive performance on public benchmarks for image-to-text and text-to-image retrieval.

The rest of this chapter is structured as follows. Section 5.2 describes the proposed recurrent residual fusion method. The image-text matching network is presented in Section 5.3. The experimental results are reported in Section 5.4. Section 5.5 summarizes the conclusions.

5.2 Recurrent Residual Fusion

We describe the details of the RRF block (Figure 5.2) with three components: an identity connection, a recurrent connection and a fusion module.

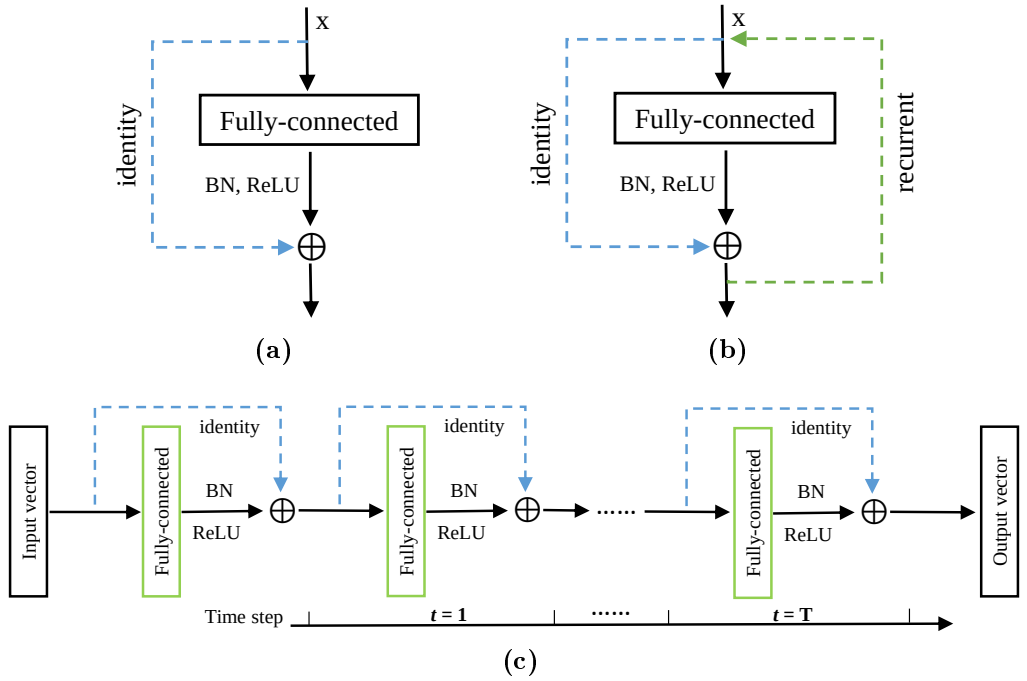

Figure 5.1: Illustration of basic building blocks. (a) An identity mapping (blue) is added to a fully-connected layer. (b) A recurrent connection (green) is introduced that uses the current output state to update the next input state. (c) We unfold the building block in (b) over recurrent steps, resulting 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.

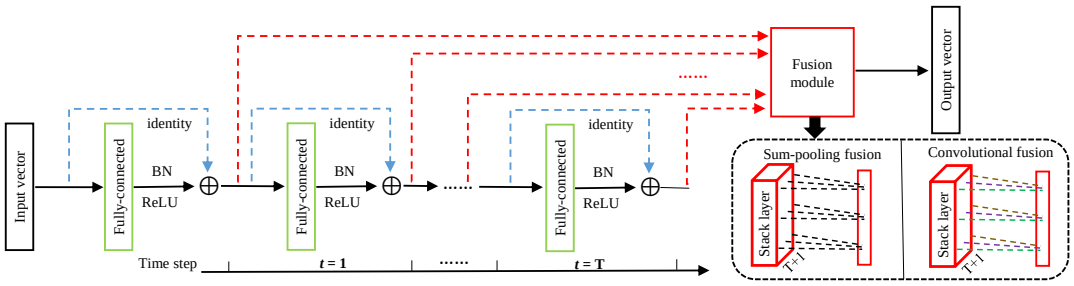

Figure 5.2: The RRF building block. Built upon recurrent residual learning, we develop a fusion module (in red) to integrate the intermediate output vectors from each recurrent step. The final output vector learns more information than the original output in Figure 5.1c. Specifically, there are two types of fusion modules: the sum-pooling fusion simply fixes equal weights, but the convolutional fusion can learn adaptive weights (drawn in different colors).

Identity connection

The basic building block in ResNet [10] 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 on top of a fully-connected layer. As can be seen in Figure 5.1a, our residual block consists of a

fully-connected layer (FC), a batch normalization layer (BN) [135] and a Rectified Linear Unit (ReLU) layer [4]. The input and output channels of the FC layer should have the same size. The computation can be presented by

$$h(x) = \sigma(f(x)) + x, \tag{5.1}$$

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

Recurrent connection

RNNs [186, 187] are proposed 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 Figure 5.1b, 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 Figure 5.1c. 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

$$x_t = h(x_{t-1}) \tag{5.2}$$

$$f(x_t) = w \cdot x_t + b \tag{5.3}$$

$$h(x_t) = \sigma(f(x_t)) + x_t \tag{5.4}$$

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.

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 Figure 5.1c, 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 5.2 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 is of size $1 \times N \times (T+1)$, 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.

(1) *Sum-pooling fusion.* As can be seen in the right bottom of Figure 5.2, it computes a summation across the feature channels of the stack layer S. The fused output vector S_{sum} is represented by

$$S_{sum} = \sum_{i=0}^{T} h(x_i) = \sum_{i=0}^{T} \sigma(f(x_i)) + x_i. \tag{5.5}$$

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

(2) *Convolutional fusion.* Normally, each side branch (or intermediate layer) may influence the output vector with different importance. 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}:

$$S_{conv} = w_f * S + b_f \tag{5.6}$$

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 number of parameters used in a deep network.

In summary, the RRF block incorporate the above three components 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. Unlike 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.

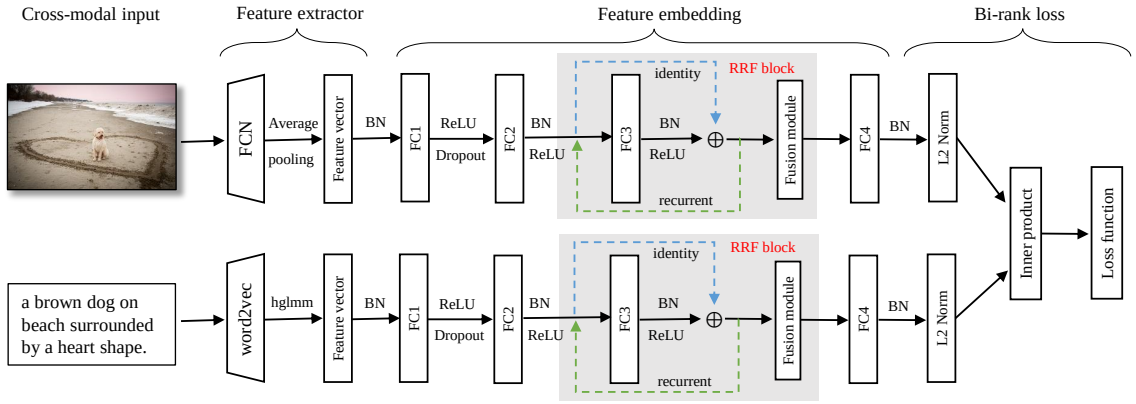

Figure 5.3: The overview architecture of the proposed RRF-Net for image and text matching. This two-branch network comprises three key steps: (1) feature extractors are used for capturing visual and textual representations. (2) Four fully-connected layers (from FC1 to FC4) in two branches are used for 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 Figure 5.2. (3) After normalizing the two output vectors and computing their inner product, we employ a bi-rank loss to train the entire network.

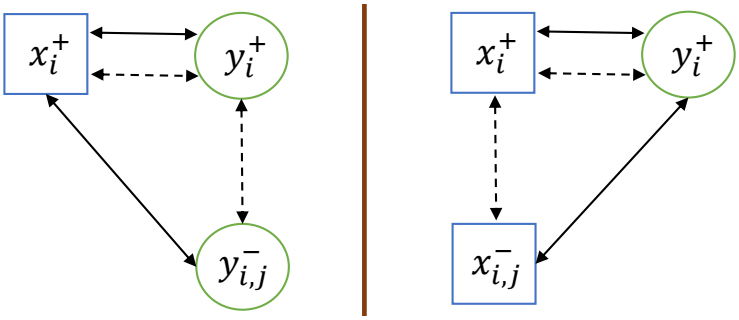

Figure 5.4: Illustration of computing the bi-rank loss that are used to train the RRF-Net. Left: image-to-text rank loss; Right: text-to-image rank loss. x and y indicate the image and text, respectively.

5.3 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 5.3 illustrates the architecture of the network, and we will describe its three key steps as below.

5.3.1 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 [10] pre-trained on ImageNet [5]. To efficiently extract dense region representations, CNN models are first recast to fully convolutional networks (FCNs) [26]. 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) [51] which is built based on word2vec model [188]. For each sentence, HGLMM computes one 18000-dimensional vector with 30 centers (*i.e.* 300*30*2). To decrease the memory cost [53], we also use PCA to reduce the dimension from 18000 to 6000. Finally, the 6000-dimensional vector acts as a powerful feature.

5.3.2 Feature embedding

To learn a discriminative embedding space, we develop four fully-connected layers on top of the two feature extractors. Their channels are {2048, 512, 512, 512} in both 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 batch normalization (BN) [135].

The core component in each branch is the FC3 layer as it introduces the RRF building block. RRF increases the FC3 layer to depth $T + 1$ 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.

5.3.3 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

$$s(x, y) = 1 - \frac{f(x) \cdot f(y)}{||f(x)|| \cdot ||f(y)||}. \tag{5.7}$$

Smaller distances indicate larger similarities. To train the network, we define a bi-rank loss function, including image-to-text and 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 Figure 5.4. Finally, this loss function is expressed with

$$\begin{aligned} l_{i2t} = \sum_{j=1}^{N} \Big(& \alpha_1 \max[0, s(x_i^+, y_i^+) - s(x_i^+, y_{i,j}^-) + m] \\ & + \alpha_2 \max[0, s(x_i^+, y_i^+) - s(y_i^+, y_{i,j}^-) + m] \Big), \end{aligned} \tag{5.8}$$

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

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 Figure 5.4. The text-to-image rank loss is as follows

$$\begin{aligned} l_{t2i} = \sum_{j=1}^{N} \Big(& \alpha_1 \max[0, s(y_i^+, x_i^+) - s(y_i^+, x_{i,j}^-) + m] \\ & + \alpha_2 \max[0, s(y_i^+, x_i^+) - s(x_i^+, x_{i,j}^-) + m] \Big), \end{aligned} \tag{5.9}$$

Full objective

The objective is to minimize the total loss by adding the two rank loss functions

$$l(x_i^+, y_i^+, X_i^-, Y_i^-) = \frac{\beta_1 l_{i2t} + \beta_2 l_{t2i}}{N}, \tag{5.10}$$

where the weights β_1 and β_2 control the importance of the two terms of one-directional rank loss. Compared with [53] which searches for extra positive intra-modal pairs, our bi-rank loss directly uses the negative intra-modal pairs and needs a minimal amount of additional computations.

Table 5.1: Evaluation results on the proposed 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).

Method	Image to Text		Text to Image	
	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

5.4 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.

Datasets. (1) Flickr30K [189]: following the dataset splits in [190], we use 29,783 training images, 1,000 validation images and 1,000 test images. Each image is annotated by five sentence-level texts. It has $29,783 * 5 = 148,915$ training pairs. (2) MSCOCO [117]: it consists of 82,783 training images and 40,504 validation images. 1,000 test images are selected from the validation set [190]. We choose five sentences for each image and generate $82,783 * 5 = 413,915$ training pairs.

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 [53], 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 method. It uses the same 4-layer plain network in Figure 5.3 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.

5.4.1 Results and discussion

To measure the performance of image-text retrieval, we adopt the evaluation metric R@K which is the recall rate of a correctly retrieved ground-truth at top K candidates (*e.g.* $K = 1, 5, 10$) [55].

Figure 5.5: 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.

Evaluation for the RRF-Net

In Table 5.1, 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 $T = 3$. We believe that evaluating more recurrent steps is still promising in future research. The first and second columns in Figure 5.5 compare the examples between the baseline and the RRF-Net.

Evaluation for fusion modules

Recall that we define two types of fusion modules. Table 5.2 compares their quantitative results. First, we trained a RRF-Net model without using any fusion module, which is actually a recurrent residual model in Figure 5.1c. By comparison, we can

Table 5.2: Evaluation for fusion modules on the Flickr30K test set. The convolutional fusion shows better results by learning adaptive weights.

Method	Image to Text		Text to Image	
	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

Table 5.3: Compared results (R@K) between the bi-rank loss and the original bi-directional loss on the Flickr30K test set.

Method	Image to Text		Text to Image	
	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

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.

Evaluation for the bi-rank loss

Table 5.3 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 $\sim$1% performance improvements compared with the bi-directional loss.

5.4.2 Comparison with other approaches

We compared our results with the state-of-the-art approaches in Table 5.4. Overall, RRF-Net achieves competitive (and often better) performance on both Flickr30K and MSCOCO datasets. On the FLICKR30K dataset, DSPE [53] and 2WayNet [76] lead recent state-of-the-art results. Although 2WayNet has the best R@1 results on Flickr30K, the proposed RRF-Net outperforms it on the R@5 accuracy. Additionally, our approach on MSCOCO outperforms the top state-of-the-art approaches.

Table 5.4: Comparison with the state-of-the-art approaches on Flickr30K and MSCOCO for cross-modal retrieval. Our RRF-Net can compete with 2WayNet [76] on the Flickr30K dataset and achieve superior results on the MSCOCO dataset.

Method	Flickr30K dataset						MSCOCO dataset					
	Image to Text			Text to Image			Image to Text			Text to Image		
	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 [55]	22.2	48.2	61.4	15.2	37.7	50.5	38.4	69.9	80.5	27.4	60.2	74.8
UVSE [66]	23.0	50.7	62.9	16.8	42.0	56.5	-	-	-	-	-	-
Mean vector [51]	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 [49]	27.9	56.9	68.2	26.8	52.9	66.9	-	-	-	-	-	-
VQA-aware [191]	33.9	62.5	74.5	24.9	52.6	64.8	50.5	80.1	89.7	37.0	70.9	82.9
GMM+HGLMM [51]	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 [190]	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 [185]	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) [52]	33.6	64.1	74.9	26.2	56.3	69.6	42.8	73.1	84.1	32.6	68.6	82.8
HM-LSTM [65]	38.1	-	76.5	27.7	-	68.8	43.9	-	87.8	36.1	-	86.7
DSPE [53]	40.3	68.9	79.9	29.7	60.1	72.1	50.1	79.7	89.2	39.6	75.2	86.9
sm-LSTM [68]	42.5	71.9	81.5	30.2	60.4	72.3	53.2	83.1	91.5	40.7	75.8	87.4
2WayNet [76]	**49.8**	67.5	-	**36.0**	55.6	-	55.8	75.2	-	39.7	63.3	-
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 5.5: Model ensemble results (R@K, $K = 1, 5$) on the Flickr30K test set. Merging more models is significant to obtain better results.

Method	Image to Text		Text to Image	
	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**

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 [7]. 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 [53]. Therefore, RRF-Net presents consistently high performance for diverse feature extractors.

5.4.3 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):

$$s^{'}(x, y) = \frac{\sum_{m \in M} s^{m}(x, y)}{|M|}, \tag{5.11}$$

where M is the index set, and $s^m(x, y)$ 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 5.5, 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 without increasing the training complexity. In Figure 5.5, the third column shows its retrieval results.

5.5 Chapter Conclusions

In this chapter, we have exploited the RRF block and RRF-Net which can bridge the gap between image and text features in a deep matching network. RRF can be viewed as a feature enhancement component to gather visual and textual representations into a more discriminative embedding space. In addition, we have presented a bi-rank loss function to enhancing the matching constraints in the embedding space. Experiments showed that RRF-Net can achieve competitive performance on the datasets, Flickr30K and MSCOCO.

Future work. This work can provide promising insights towards how to efficiently narrowing the semantic gap between vision and language. Image-text matching is a fundamental technique for many multi-modal research tasks. Therefore, it is promising that the RRF building block could be seamlessly integrated into other multi-modal systems like image captioning and visual question answering.