

# Learning Visual and Textual Representations for Multimodal Matching and Classification

Yu Liu<sup>a</sup>, Li Liu<sup>b,c</sup>, Yanming Guo<sup>b</sup>, Michael S. Lew<sup>a,\*</sup>

<sup>a</sup>*Department of Computer Science, Leiden University, Leiden, 2333 CA, The Netherlands*

<sup>b</sup>*College of System Engineering, National University of Defense Technology, Changsha, Hunan 410073, China*

<sup>c</sup>*Center for Machine Vision and Signal Analysis, University of Oulu, Oulu, 8000, Finland.*

---

## Abstract

Multimodal learning has been an important and challenging problem for decades, which aims to bridge the modality gap between heterogeneous representations, such as vision and language. Unlike many current approaches which only focus on either multimodal matching or classification, we propose a unified network to jointly learn Multimodal Matching and Classification (MMC-Net) between images and texts. The proposed MMC-Net model can seamlessly integrate the matching and classification components. It first learns visual and textual embedding features in the matching component, and then generates discriminative multimodal representations in the classification component. Combining the two components in a unified model can help in improving their performance. Moreover, we present a multi-stage training algorithm by minimizing both of the matching and classification loss functions. Experimental results on four well-known multimodal benchmarks demonstrate the effectiveness and efficiency of the proposed approach, which achieves competitive performance for multimodal matching and classification compared to state-of-the-art approaches.

*Keywords:* Vision and language, multimodal matching, multimodal classification, deep learning

---

---

\*Corresponding author

*Email address:* [m.s.k.lew@liacs.leidenuniv.nl](mailto:m.s.k.lew@liacs.leidenuniv.nl) (Michael S. Lew)

## 1. Introduction

The problem of multimodal analytics has attracted increasing attention due to a drastic growth of multimedia data such as text, image, video, audio, and graphics. Consequently, it has aroused new challenges in unifying different modalities and bridging their semantic gap. Prior work has been dedicated to developing computational models to simulate the human-brain mechanism regarding unifying and processing the multimodal data. In this work, our focus is on jointly modeling the multimodal matching and classification between vision and language. The multimodal research underpins many critical applications in the computer vision field, including image captioning [1, 2, 3], cross-modal retrieval [4, 5, 6], and zero-shot recognition [7, 8, 9, 10].

Specifically, multimodal matching has been studied for decades, with the aim of searching for a latent space, where visual and textual features can be unified to be latent embeddings. The hypothesis is that different modalities have semantically related properties that can be distilled into a common latent space. Early approaches that attempt to learn latent embeddings are mainly developed based on the Canonical Correlation Analysis (CCA) [11], which is effective at maximizing the high correlation between visual and textual features in the latent space. Driven by the increasing progress of deep learning, many works [12, 13, 14, 15] have been dedicated to developing deep matching networks to learn discriminative latent embeddings and train the networks by using a bi-directional rank loss function. They have achieved state-of-the-art performance on many well-known multimodal benchmarks [6, 16, 17, 18].

However, learning latent embeddings is influenced by the notable variance in images or texts. For example in Fig. 1, five sentences annotated by humans are provided to describe the same image. The input image and five sentences are projected into a latent space based on a two-branch network (see Fig. 3). One can observe that these sentences have significant variance on representing the visual content. Although they can consistently describe the main objects in the scene such as ‘girl’ (or ‘child’) and ‘bicycle’ (or ‘bike’), they still present great variance in terms of other objects, *e.g.* ‘bench’, ‘table’ and ‘leaves’. Likewise, the potential variance is also existing in visual embedding features. Consequently, it becomes more difficult to model image and text matching.

To address this issue, we aim to introduce a classification component to learn more robust latent embeddings. Our motivation is that object labels can typically provide more consistent and less biased information than sentences. As can be seen in Fig. 1, object labels contain the most important concepts in the image, for example ‘Person’ and ‘Bicycle’ which are commonly mentioned in all of the five sentences. On the other hand, some visual concepts, which are subjectively described in some of the sentences (*e.g.* ‘leaves’ and ‘sweater’) will not appear in the ground-truth labels. Hence, using the object labels as additional supervisory signals is beneficial to correct the biased descriptions and improve the matching between images and texts.

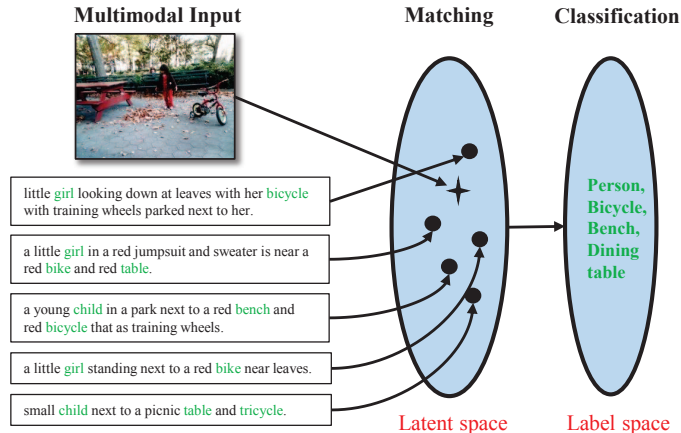


Figure 1: Example of joint multimodal matching and classification. Given one image and its descriptive sentences, they are first co-embedded into a latent space for matching (in red and blue). Then, the visual and textual embedding features are integrated to be a multimodal representation for classification. In the input sentences, the words related to the ground-truth object labels are highlighted in green.

In this work, we propose a unified network for joint Multimodal Matching and Classification (MMC-Net) as illustrated in Fig. 3. First, the matching component transforms the input visual and textual features, respectively, via a couple of fully-connected layers and a fusion module. The matching loss is imposed on the outputs of the two fusion modules to maximize their correlation. Then, the classification component is built upon the visual and textual embedding features. A compact bilinear pooling module is used to generate a multimodal representation vector, based on which the classification loss is computed to predict object labels. In this way, the proposed MMC-Net can jointly learn the latent embeddings and the multimodal representation in a unified model. On the one hand, the classification component is beneficial to alleviate the biased input, so that the model can learn better robust latent embeddings. On the other hand, the matching component is able to bridge the modality gap between vision and language, and therefore combining visual and textual embedding features can produce a discriminative multimodal representation for classification.

The contributions of this work are summarized as follows:

- We propose a novel deep multimodal network (*i.e.* MMC-Net), where the matching and classification components can be seamlessly integrated and help promote each other jointly. MMC-Net is a general architecture that is potentially applicable to diverse multimodal tasks related to matching and classification.
- We present a multi-stage training algorithm by incorporating the matching and classification loss. It can make the matching and classification components more compatible in a unified model.

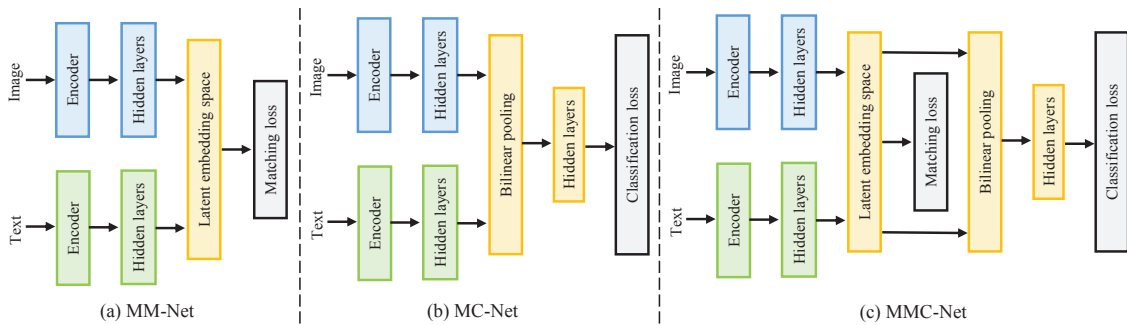


Figure 2: Illustration of three multimodal networks. (a) Multimodal Matching Network. (b) Multimodal Classification Network. (c) Multimodal Matching and Classification Network. Note that, the parameters in the image and text branches are unshared, as drawn in blue and green.

- Results on four well-known multimodal benchmarks demonstrate that MMC-Net outperforms the baseline models that are built for either matching or classification (*i.e.* MM-Net and MC-Net). In addition, our approach achieves competitive performance compared to current state-of-the-art approaches.

The rest of this paper is organized as follows. Sec. 2 summarizes the related work about multimodal matching and classification. We introduce the details of the proposed MMC-Net model in Sec. 3, and the training algorithm in Sec. 4. Comprehensive experiments in Sec. 5 are used to evaluate the approach. Finally, Sec. 6 concludes the paper and discusses the future work.

## 2. Related Work

In this section, we introduce the use of multimodal fusion, and then revisit recent works related to the research of image-text based multimodal matching and classification.

### 2.1. Multimodal Fusion

Human can see, hear and speak simultaneously. Motivated by this, it is beneficial to integrate different modality-specific representations, which can help compensate the limitation of one single modality. Based on various conditions (*e.g.* detectors, sensors and equipments), we can represent the same phenomenon with multimodal representations (*e.g.* image, video, text and audio). In recent years, the growing availability of multiple modalities has triggered a large amount of research efforts on multimodal fusion. Consequently, a wide range of multimodal applications, including action recognition [19, 20, 21], image captioning [1, 2, 3], cross-modal retrieval [4, 5, 6] and zero-shot recognition [7, 8, 9, 10], have been of primary importance in the field of computer vision. For example, Simonyan *et al.* [19] developed a two-stream ConvNet architecture for action recognition in videos, which could integrate spatial and temporal information based on multi-frame dense optical flow. The work of Hu *et al.* [20] presented a joint learning model to simultaneously learn

80 heterogeneous features from different channels (*i.e.* RGB, depth) for RGB-D activity recognition. In this work, our focus is on the applications regarding both vision and language, which will be detailed as follows.

### 2.2. Multimodal Matching

Typically, multimodal matching is posed as a feature embedding problem, which aims to project  
85 heterogeneous representations into a common space. As the multimodal generalization of PCA, CCA [11] learns a pair of linear transformations to maximize the correlation matrix between different modalities. Many extensions [22, 23, 24] were developed to augment the effectiveness of CCA. For instance, Gong *et al.* [25] added a third view with the two-view CCA using high-level image semantics in order to gain a better separation for multimodal data. Ranjan *et al.* [26] proposed a  
90 multi-label CCA approach by introducing multi-label information while learning the cross-modal subspaces. In addition, it is beneficial to build deep CCA models for learning better non-linear projections end-to-end [27, 28]. To promote the linear transformations in CCA, Andrew *et al.* [27] developed a deep CCA model to directly learn a flexible nonlinear mapping. In recent literature, A number of approaches [29, 15, 16, 18] have been dedicated to designing diverse deep matching  
95 networks to search for a more discriminative latent space. Ma *et al.* [15] used multimodal CNNs for encoding both images and sentences, to learn the matching relation between the image and the word fragments. Karpathy *et al.* [14] proposed a novel ranking model that aligned visual and language modalities using a multimodal latent embedding. Wang *et al.* [6] built a simple and efficient matching network that focused on preserving the structure relation of images and texts  
100 in the latent space. Nam *et al.* [17] developed visual and textual attention models and jointly trained them to capture the shared semantics between images and sentences. In Fig. 2(a), we show a general pipeline of multimodal matching networks (MM-Net). It is composed of feature encoders, hidden layers, a latent embedding space, and a matching loss function.

### 2.3. Multimodal Classification

105 Multimodal classification aims to combine visual and textual features as a multimodal representation, and then uses it to predict class labels. Early studies attempted to use simple fusion modules such as element-wise sum or product. In the work by Ba *et al.* [7], they used a dot product to integrate two features in the last layer, and produced a set of classifier weights for fine-grained classification. Ma *et al.* [30] developed an auto-encoder with the structured regularization to en-  
110 hance the interactions while integrating different modality-specific features. Recently, Bai *et al.* [31] presented an end-to-end trainable neural network for fine-grained image classification through capturing scene textual and visual cues from images. Besides, visual question answering [32, 33, 34]

that is often cast as a multimodal classification problem, relies on an element-wise sum operation to incorporate visual and textual features. To achieve better fused feature, Fukui *et al.* [35] exploited  
115 a multimodal compact bilinear pooling [36] for visual question answering and visual grounding. Compact bilinear pooling is able to capture high-order correlated information between visual and textual features, while using much less parameters than the standard bilinear pooling [37, 38]. One recent work [39] merged the prediction scores from the vision and language streams in a late-processing manner. Figure 2(b) describes the pipeline of multimodal classification networks based  
120 on the bilinear pooling.

#### 2.4. Multimodal Matching and Classification

Unlike the above work, our purpose is to model the multimodal matching and classification tasks in one network. As illustrated in Fig. 2(c), the proposed MMC-Net builds the classification component upon the matching component. Consequently, the whole network can be used for both  
125 matching and classification. Zhang *et al.* [40] developed a deep matching framework that can jointly optimize both classification and similarity constraints for fine-grained image classification. However, their work focused solely on the visual domain without introducing the textual domain. One recent work [41] for zero-exemplar event detection developed a three-branch network that aimed to classify event categories based on the input video and its textual title, by learning to embed the  
130 video feature and the event article feature in the matching component. However, their classification component was only based on the textual embedding, but did not use the visual embedding. Their manner limits the classification performance and discourages the benefit of unifying the matching and classification components. Instead, our classification component allows to combine visual and textual embeddings and can produce more informative multimodal representations.

### 135 3. Multimodal Matching and Classification Network

In this section, we introduce the proposed MMC-Net model and its three key components.

#### 3.1. Overall Architecture

Figure 3 illustrates the overview architecture of MMC-Net, which mainly consists of three components: multimodal input, multimodal matching and multimodal classification. Given an image  
140 and its corresponding text, MMC-Net first utilizes off-the-shelf feature encoders to extract the visual and textual features, respectively. Next, in the multimodal component, two groups of four fully-connected layers are used in both image and text branches to learn a latent space, where its objective is to minimize the matching loss between the related images and texts. Moreover, the multimodal classification component is built upon the visual and textual embedding features. We

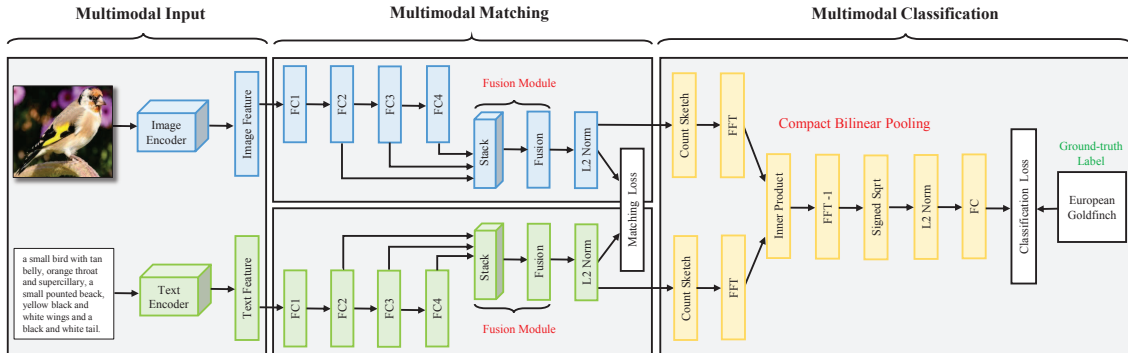


Figure 3: The overview architecture of our proposed MMC-Net for joint multimodal matching and classification. It comprises three key components. (1) The multimodal input aims to capture visual and textual representations from off-the-shelf encoders (e.g. CNN and word2vec). (2) In the matching component, four fully-connected layers in both of the image and text branches are developed to learn the latent embeddings. (3) Based on the visual and textual embedding features, the classification component utilizes a compact bilinear pooling module which can generate a high-order multimodal representation to perform the prediction. The entire network can be trained with a matching loss and a classification loss.

145 employ a compact bilinear pooling module to generate a high-order and efficient multimodal representation. The classification loss is computed with respect to the pre-defined ground-truth labels. Next, we will detail each of the three components.

### 3.2. Multimodal Input

In a data collection with  $N$  matching image-text pairs,  $(x_i, y_i)$  represent the encoded visual and textual features,  $i = 1, \dots, N$ . Taking these features as input instead of the raw data enables to train the entire network effectively. Also, any common feature encoders are potentially applicable for this network.

**Image encoder:** we use the powerful CNN model, ResNet-152 [42], which is pre-trained on the ImageNet dataset [43]. First, the CNN model is recast to its fully convolutional network (FCN) counterpart, to extract richer region representations. Then we set the smaller side of the image to 512 and isotropically resize the other side. The last max-pooling layer in ResNet-152 is averaged to generate a 2048-dimensional feature vector. Compared with the widely-used VGG feature [44] (i.e. 4096-dim), ResNet-152 can provide more discriminative visual representation, while decreasing the feature dimensions (2048 v.s. 4096). The extracted image feature is then fed into the image branch of the matching component.

**Text encoder.** we employ the simple yet efficient word2vec [45] to represent sentence-level texts. It provides a 300-dimensional feature vector, which is often called Mean vector. Notably, more informative text encoders can be developed based on word2vec, for example the Hybrid Gaussian-Laplacian mixture model (HGLMM) [46] that computes a 18000-dimensional feature vector with 30 centers (i.e.  $300 \times 30 \times 2$ ). However, we still use the standard Mean vector due to its high efficiency and low dimensionality.

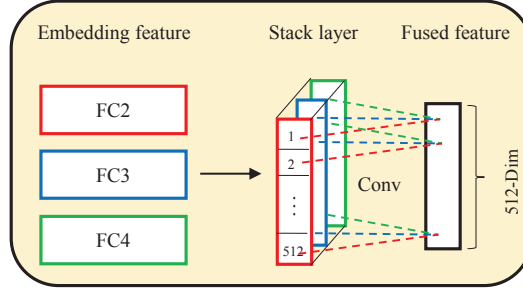


Figure 4: Illustration of the fusion module used in the matching component. A convolutional layer is used to learn weights for different spatial elements in FC2, FC3 and FC4.

### 3.3. Multimodal Matching

The multimodal matching component contains three aspects: latent embedding, fusion module and matching loss.

**Latent embedding.** As shown in Fig. 3, the matching component develops two branches of four fully-connected layers to simultaneously project visual and textual features into a discriminative latent space. Note that the parameters of the two branches (drawn in blue and green) are unshared due to the modality specialization. The channels from FC1 to FC4 are set to  $\{2048, 512, 512, 512\}$  in both of the two branches. First, the input visual and textual features are normalized with the batch normalization (BN) [47]. Then FC1 is regularized by a dropout layer with 0.5 probability, and instead other fully-connected layers are regularized with the BN layer. ReLU is used after the fully-connected layers.

**Fusion module.** Exploiting multi-layer features has been well-studied in many deep neural networks [48, 49, 50, 51], as it allows to take advantage of different levels of hidden representations in the networks. Driven by this, we introduce a fusion module to generate a multi-layer embedding feature. Figure 4 depicts the pipeline of the fusion module. Since the FC2, FC3 and FC4 layers have the same number of channels, it is feasible to stack their feature vectors together. Then we employ a convolutional operation to learn adaptive weights while fusing the three layers.

We denote the stack layer in the two branches as  $S(x_i)$  and  $S(y_i)$ , respectively. The stack layer, a  $512 \times 3$  matrix, is convolved by the convolutional filter, which has a size of  $1 \times 1 \times 3$ . Note that, the three weights are shared over the spatial dimensions of the stack layer. We can compute the fused visual feature  $f(x_i)$  and textual feature  $g(y_i)$  by

$$f(x_i) = W_I^{fuse} \odot S(x_i) + b_I^{fuse}, \quad (1)$$

$$g(y_i) = W_T^{fuse} \odot S(y_i) + b_T^{fuse}, \quad (2)$$



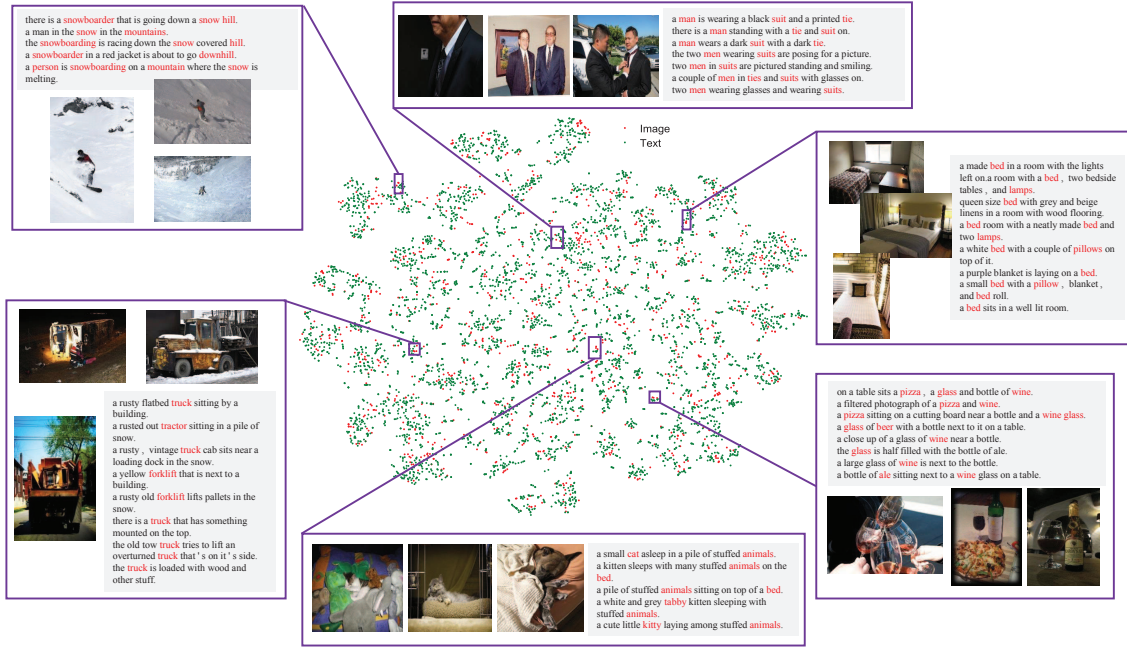


Figure 5: Visualization of the visual and textual embedding features learned in the matching component. Each image (in red) is related to several corresponding texts (in green). We present some images and texts corresponding to the points in the distribution map. The semantic words related to the visual content are shown in red.

where  $W_I^{fuse}$  and  $W_T^{fuse}$  are the fusion weights to be learned (*i.e.* 3 elements)  $b_I^{fuse}$  and  $b_T^{fuse}$  are  
 190 the bias vectors (*i.e.* 512 elements). The operator  $\odot$  represents the convolutional operation.

Although the common element-wise operators such as sum-pooling and inner product are simple  
 to compute, they do not adapt the importance of different layers. Another fusion approach is  
 concatenating the three 512-D vectors into one 3\*512-D vector. However, the concatenation output  
 will increase the feature dimensionality and make it more expensive to compute the matching loss.  
 195 To summarize, the convolutional fusion module can provide marked performance improvements,  
 while it has a minimal increase to the total parameters used in the network.

**Matching loss.** As a common practice, the matching distance between  $f(x_i)$  and  $g(y_i)$  is  
 computed with the cosine distance [15, 6, 16]

$$d(f(x_i), g(y_i)) = 1 - \frac{f(x_i) \cdot g(y_i)}{\|f(x_i)\| \cdot \|g(y_i)\|}. \quad (3)$$

Smaller distances indicate more similar image-text pairs. Both  $f(x_i)$  and  $g(y_i)$  are L2-normalized  
 200 before computing their cosine distance. To preserve the similarity constraints in the latent space, we  
 define the matching loss based on an efficient bi-directional rank loss function, similar to [52, 13, 6].  
 The loss function needs to handle the two triplets,  $(x_i, y_i, y_{i,k}^-)$  and  $(y_i, x_i, x_{i,k}^-)$ , where  $x_{i,k}^- \in X_i^-$   
 and  $y_{i,k}^- \in Y_i^-$  are the negative images and texts,  $k = 1, \dots, K$ . To exploit more representative

non-matching pairs, we pick the top  $K$  most dissimilar candidates in each mini-batch. Intuitively, this loss function is designed to decrease the distances of matching pairs (e.g.  $x_i$  and  $y_i$ ) and increase the distances of non-matching pairs (e.g.  $x_i$  and  $y_{i,k}^-$ ,  $y_i$  and  $x_{i,k}^-$ ). Formally, the matching loss based on the fused features is formulated via

$$\begin{aligned} \mathcal{L}_{mat}^{fuse} = & \sum_{i=1}^N \sum_{k=1}^K \max \left[ 0, d(f(x_i), g(y_i)) - d(f(x_i), g(y_{i,k}^-)) + m \right] \\ & + \alpha \max \left[ 0, d(f(x_i), g(y_i)) - d(f(x_{i,k}^-), g(y_i)) + m \right], \end{aligned} \quad (4)$$

where  $m$  is a margin parameter, and  $\alpha$  is used to balance the importance of the two triplets. Minimizing this loss cost will lead to a desirable latent space, where the matching distance  $d(f(x_i), g(y_i))$  should be smaller than any of the non-matching ones  $d(f(x_i), g(y_{i,k}^-))$  and  $d(f(x_{i,k}^-), g(y_i))$ ,  $\forall x_{i,k}^- \in X_i^-$ ,  $\forall y_{i,k}^- \in Y_i^-$ .

In Fig. 5, we make use of the t-SNE algorithm [53] to visualize our embedding features (i.e.  $f(x_i)$  and  $g(y_i)$ ). We use the 1000 images and 5000 texts from the MSCOCO test set. It can be seen that in the distribution map an image feature (in red) is properly surrounded by several related text features (in green), as each image is annotated by five ground-truth matching texts in the dataset. Therefore, this visualization shows that our embedding model can align the images and texts due to learning their semantic correlation. In addition, some images and texts corresponding to the points are shown in the windows. We can see that the embeddings can cluster similar images and texts together despite the significant variations and changes.

### 3.4. Multimodal Classification

The classification component aims to incorporate the visual and textual embedding features and then generates a multimodal representation for predicting object labels. In the following, we detail the classification component including a bilinear pooling module and classification loss.

**Bilinear pooling.** We take advantage of a bilinear pooling module to incorporate visual and textual embedding features learned in the matching component. The bilinear pooling [37] aims to model the pair-wise multiplicative intersection between all elements of two vectors. It can generate more expressive features than other basic operators such as element-wise sum or product. The standard bilinear pooling is formulated with

$$\mathcal{B}(x_i, y_i) = f(x_i)^T g(y_i), \quad (5)$$

Since  $f(x_i)$  and  $g(y_i)$  are  $1 \times M$  vectors (i.e.  $M = 512$ ),  $\mathcal{B}(x_i, y_i)$  becomes an  $M \times M$  matrix that is then reshaped to be a  $1 \times M^2$  vector. Due to the high dimensionality of the bilinear vector

---

**Algorithm 1** CBP with latent embedding features

---

- 1: **Input:**  $f(x_i) \in \mathbb{R}^M, g(y_i) \in \mathbb{R}^M$
  - 2: **Output:**  $\mathcal{B}(x_i, y_i) \in \mathbb{R}^D$
  - 3: **Initialize hash functions:**  $h_1, s_1, h_2, s_2$   
    **For**  $j \leftarrow 1 \cdots M$   
        sample  $h_1[j], h_2[j]$  from  $\{1, \dots, D\}$   
        sample  $s_1[j], s_2[j]$  from  $\{-1, 1\}$   
    **End for**
  - 4: **Compute count sketches:**  
     $\hat{f}(x_i) = [0, \dots, 0], \hat{g}(y_i) = [0, \dots, 0]$   
    **For**  $j \leftarrow 1 \cdots D$   
         $\hat{f}(x_i)[h_1[j]] = \hat{f}(x_i)[h_1[j]] + s_1[j] \cdot f(x_i)[j]$   
         $\hat{g}(y_i)[h_2[j]] = \hat{g}(y_i)[h_2[j]] + s_2[j] \cdot g(y_i)[j]$   
    **End for**
  - 5: **Convolution of Count Sketches:**  
     $\mathcal{B}(x_i, y_i) = \text{FFT}^{-1}(\text{FFT}(\hat{f}(x_i)) \circ \text{FFT}(\hat{g}(y_i)))$ ,  
    where the  $\circ$  denotes element-wise multiplication.
- 

(i.e.  $M^2$ ), we instead use the compact bilinear pooling (CBP) variant [36], which can decrease the dimensionality to  $D$  (where  $D \ll M^2$ ) while retaining the strong discrimination. Different from [35, 36] in which they simply perform the CBP module with the input visual or textual features, we build the CBP module based on the latent embeddings to generate a multimodal feature vector (Fig. 3).  
235

The computational procedure of the CBP module is detailed in Algorithm 1. At first, we initialize several hashing functions from the pre-defined sets. Then, it computes the count sketches [54] to maintain linear projections of a vector with several random vectors. Finally, we make use of the Fast Fourier Transformation (FFT) to compute the convolution of the count sketches, and produce a bilinear vector  $\mathcal{B}(x_i, y_i)$  by an inverse FFT. In particular, the count sketches have the properties:  
240

$$E[\langle \hat{f}(x_i), \hat{g}(y_i) \rangle] = \langle f(x_i), g(y_i) \rangle, \quad (6)$$

$$\text{Var}[\langle \hat{f}(x_i), \hat{g}(y_i) \rangle] \leq \frac{1}{D} (\langle f(x_i), g(y_i) \rangle^2 + \|f(x_i)\|^2 + \|g(y_i)\|^2). \quad (7)$$

Next, the bilinear vector  $\mathcal{B}(x_i, y_i)$  is processed by a signed square-root layer and an L2 normalization layer. Then, we employ a fully-connected layer to estimate the prediction. Assume that there are  $C$  object labels pre-defined in the dataset, the  $j$ -th class probability is predicted with  
245

$$a_{i,j} = \sum_{k=1}^D W_{j,k} \mathcal{B}(x_i, y_i)_k \quad (8)$$

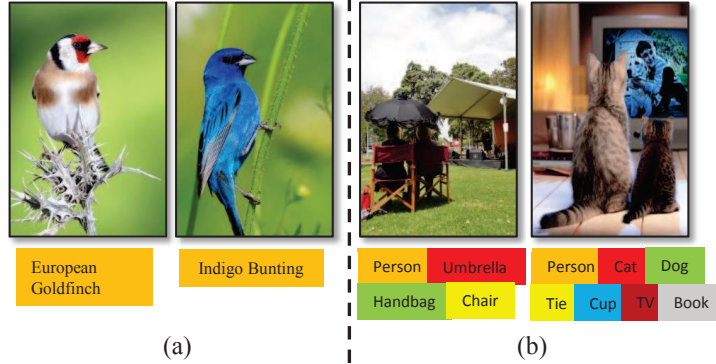


Figure 6: (a) Examples of single-label images from CUB-Bird [55]. (b) Examples of multi-label images from MSCOCO [56].

where  $j = 1, \dots, C$ .  $W_{j,k}$  is the parameter matrix with the size of  $D \times C$ . For simplicity, we do not show the signed square-root and the L2 normalization in this formulation.

250 **Classification loss.** The objective of the classification component is to minimize the loss cost of the prediction with respect to the given ground-truth labels. Fig. 6 shows some images that are annotated by single label or multiple labels. It makes sense to compute different loss functions for single-label and multi-label classification, respectively.

255 1) *Single-label classification.* For example the fine-grained classification in Fig. 6(a), each image is labelled with a fine bird category. To train the classification component, we use the softmax loss function that is represented by

$$\mathcal{L}_{cls} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C \delta(g_i = j) \log p_{i,j}, \quad (9)$$

$$p_{i,j} = \frac{\exp(a_{i,j})}{\sum_{k=1}^C \exp(a_{i,k})}, \quad (10)$$

where  $g_i$  is the ground-truth label corresponding to  $x_i$ .  $\delta(g_i = j)$  is 1 when  $g_i = j$ , otherwise is 0.

260 2) *Multi-label classification.* As shown in Fig. 6(b), images annotated with multiple labels can provide richer information about the visual content. Although many of these labels may appear in the input text, they can still offer complementary labels which are ignored in the text due to less visual attention. We employ the sigmoid cross-entropy loss function to supervise the multi-label classification. The total cost sums up  $K$  of element-wise loss terms

$$\mathcal{L}_{cls} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C g'_{i,j} \log p'_{i,j} + (1 - g'_{i,j}) \log(1 - p'_{i,j}), \quad (11)$$

---

**Algorithm 2** Multi-stage Training Algorithm for MMC-Net.

---

- 1: **The first stage:** train the matching component.  
initialize: learning rate  $\lambda_1$ , training iterations  $T_1$ ,  $t = 0$ .  
**while**  $t < T_1$  **do**  
   $t \leftarrow t + 1$   
  compute the matching loss  $\mathcal{L}_{mat}$  in Eq.(4);  
  update the parameters in the image and text branches:  
     $W_I^{(t)} = W_I^{(t-1)} - \lambda_1^{(t)} \frac{\partial \mathcal{L}_{mat}}{\partial W_I^{(t-1)}}$ ;  
     $W_T^{(t)} = W_T^{(t-1)} - \lambda_1^{(t)} \frac{\partial \mathcal{L}_{mat}}{\partial W_T^{(t-1)}}$ ;  
  **end while**
  - 2: **The second stage:** train the classification component.  
initialize: learning rate  $\lambda_2$  ( $< \lambda_1$ ), training iterations  $T_2$ ,  $t = 0$ .  
**while**  $t < T_2$  **do**  
   $t \leftarrow t + 1$   
  compute the classification loss  $\mathcal{L}_{cls}$  in Eq.(9) or Eq.(11);  
  update the parameters in the compact bilinear pooling module:  
     $W_{CBP}^{(t)} = W_{CBP}^{(t-1)} - \lambda_2^{(t)} \frac{\partial \mathcal{L}_{cls}}{\partial W_{CBP}^{(t-1)}}$ ;  
  **end while**
  - 3: **The third stage:** jointly fine-tune the whole network.  
initialize: learning rate  $\lambda_3$  ( $< \lambda_2$ ), training iterations  $T_3$ ,  $t = 0$ .  
**while**  $t < T_3$  **do**  
   $t \leftarrow t + 1$   
  compute the total loss in Eq.(13);  
  update all the parameters in the network:  
     $W_I^{(t)} = W_I^{(t-1)} - \lambda_1^{(t)} \frac{\partial \mathcal{L}_{total}}{\partial W_I^{(t-1)}}$ ;  
     $W_T^{(t)} = W_T^{(t-1)} - \lambda_1^{(t)} \frac{\partial \mathcal{L}_{total}}{\partial W_T^{(t-1)}}$ ;  
     $W_{CBP}^{(t)} = W_{CBP}^{(t-1)} - \lambda_2^{(t)} \frac{\partial \mathcal{L}_{total}}{\partial W_{CBP}^{(t-1)}}$ ;  
  **end while**
- 

$$p'_{i,j} = \frac{1}{1 + \exp(-a_{i,j})}, \quad (12)$$

265 where  $g'_{i,j} \in \{0, 1\}$  is the ground-truth label indicating the absence or presence of the  $j$ -th class.

## 4. Training and Inference

This section describes the training procedure of the MMC-Net model. Also, we present the inference manner for multimodal matching and classification.

### 4.1. Multi-stage Training

270 The optimization objective in the model is to minimize the total training loss which merges the matching and classification loss together

$$\min_W \mathcal{L}_{total} = \mathcal{L}_{mat} + \beta \mathcal{L}_{cls}, \quad (13)$$

where the parameter  $\beta$  is used to regulate the two loss terms. The parameters  $W$  in the network mainly contains  $W_I$  and  $W_T$  in the image and text branches, and  $W_{CBP}$  in the compact bilinear pooling module.

275 We propose a multi-stage training algorithm to better model the matching and classification components. As summarized in Algorithm 2, the training procedure consists of three stages. During the first stage we train the matching component with the loss  $\mathcal{L}_{mat}$ . For the second stage, we need to learn the parameters in the classification component using the loss  $\mathcal{L}_{cls}$ . In this stage, only the parameters in the classification component can be updated whereas the parameters in the matching  
280 component are all frozen. In the third stage, the model is initialized by the parameters learned in the first and second stages. It aims to jointly fine-tune the whole network based on the total loss  $\mathcal{L}_{total}$ . Due to using this multi-stage fashion, it is feasible to promote the training of the entire network and maintain the high performance.

Note that, the training of the FFT and inverse FFT in the CBP module also follows the chain  
285 rule of the backward propagation. As for  $\mathcal{B}(x_i, y_i)$ , the partial derivatives of  $\mathcal{L}_{cls}$  with respect to  $\hat{f}(x_i)$  and  $\hat{g}(y_i)$  can be expressed with

$$\frac{\partial \mathcal{L}_{cls}}{\partial \hat{f}(x_i)} = \text{FFT}^{-1} \left( \text{FFT} \left( \frac{\partial \mathcal{L}_{cls}}{\partial \mathcal{B}} \right) \circ \text{FFT}(\hat{g}(y_i)) \right), \quad (14)$$

$$\frac{\partial \mathcal{L}_{cls}}{\partial \hat{g}(y_i)} = \text{FFT}^{-1} \left( \text{FFT} \left( \frac{\partial \mathcal{L}_{cls}}{\partial \mathcal{B}} \right) \circ \text{FFT}(\hat{f}(x_i)) \right), \quad (15)$$

Similarly, it is straightforward to induce the partial derivatives for any variables in the model.

#### 4.2. Inference

290 We present the inference manner for multimodal matching and classification, respectively.

**Multimodal matching.** For the image-to-text matching, given a query image  $x_q$ , its purpose is to search for relevant texts *w.r.t*  $x_q$  from a text database  $Y$ . Likewise, the text-to-image matching aims to retrieve related images from an image database  $X$ , given a query text  $y_q$ . In the MMC-Net model, the fused visual and textual features learned in the fusion module are used to compare  
295 the matching distance, denoted as  $d(f(x_q), g(y_i))$  or  $d(f(x_i), g(y_q))$ , where  $y_i \in Y, x_i \in X$ . The  $k$ -nearest neighbor ( $k$ -NN) search is used to find the top- $k$  most similar candidates.

**Multimodal classification.** Its inference is based on the probabilities predicted by the last fully-connected layer in the classification component. For the single-label case, the element that has the maximum probability corresponds to the predicted class. As for the multi-label case, the items  
300 whose probabilities in the prediction are more than 0.5 are estimated to contain the corresponding object classes.

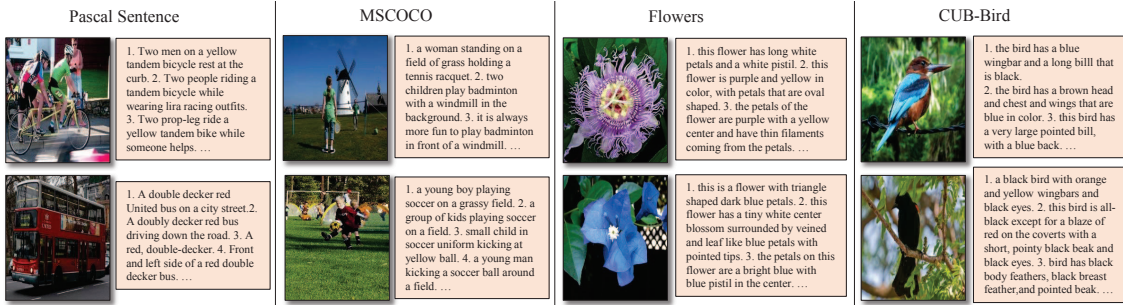


Figure 7: Example of four multimodal datasets. Three textual descriptions are listed for each image.

## 5. Experiments

In this section, we evaluate the performance of the proposed MMC-Net on four well-known multimodal benchmarks. We first introduce the configuration in the experiments, including the datasets, evaluation metrics, parameter settings and baseline models. Then we assess the performance of MMC-Net for tasks of multimodal matching and classification and compare its results with those of the baseline models. Furthermore, we conduct the ablation study to fully analyze MMC-Net. Lastly, we compare our results with other state-of-the-art approaches.

### 5.1. Dataset Settings

We performed the experiments on four well-known multimodal datasets: Pascal Sentence [57], MSCOCO [56], Flowers [58] and CUB-Bird [55]. Some image and text examples are shown in Fig. 7.

**Pascal Sentence** [57]. It contains 1000 images from 20 categories (50 images per category), and one image is described by five different sentences. We pick 800 images for training (40 images per category), 100 images for validation (5 images per category), and 100 images for test (5 images per category). In total, there are  $40 * 20 * 5 = 4000$  image-text training pairs,  $5 * 20 * 5 = 500$  validation pairs, and  $5 * 20 * 5 = 500$  test pairs.

**MSCOCO** [56]. It includes 82783 training images and 40504 validation images in total. We pick five descriptive sentences for one image and generate  $82783 * 5 = 413915$  training pairs. For a fair comparison, we use the same 1000 test images used in recent works [15, 6, 16].

**Flowers** [58]. This dataset [58] contains 102 classes with a total of 8189 images. 2040 images (train+val) are used in the training stage and the rest 6149 images are for testing. Reed *et al.* [8] collected fine-grained visual descriptions for these images by using the Amazon Mechanical Turk (AMT) platform. One image is described by ten sentence-level descriptions. Therefore, we can obtain  $2040 * 10 = 20400$  training pairs and  $6149 * 10 = 61490$  testing pairs.

**CUB-Bird** [55]. It contains 11,788 bird images from 200 categories. 5994 images are for training, and 5794 images are for testing. Similarly, ten sentences are provided to describe one

image [8]. As a result, it has  $5994 * 10 = 59940$  pairs for training, and  $5794 * 10 = 57940$  pairs for testing.

### 330 5.2. Evaluation Metrics

We evaluate the performance of multimodal matching and multimodal classification, separately.

**Multimodal matching.** We employ the widely-used retrieval metric R@K, which is the recall rate of a correctly retrieved ground-truth at top  $K$  candidates (e.g.  $K = 1, 5, 10$ ) [14, 3]. It includes results of both image-to-text (I→T) and text-to-image retrieval (T→I).

335 **Multimodal classification.** We compute the Top-1 classification accuracy for Pascal Sentence, Flowers and CUB-Bird. Since MSCOCO is a multi-label classification dataset, we evaluate the performance on it using the average precision (AP) across multiple classes.

### 5.3. Implementation Details

We implemented the proposed approach based on the publicly available Caffe library [59]. It is important to shuffle the training samples randomly during the data preparation stage. The hyper-parameters were evaluated on the validation set of each dataset. For instance, we set  $\alpha = 2$  and  $m = 0.1$  while computing the matching loss function on all the datasets. The number of non-matching pairs in the negative sets was  $K = 20$  for Pascal Sentence, Flowers and CUB-Bird, and  $K = 50$  for MSCOCO. We used a mini-batch size of 128 for Pascal Sentence, Flowers and CUB-Bird, and 1500 for MSCOCO. Note that, we use a larger  $K$  and min-batch size for MSCOCO, because it has enormously more training samples, compared to the other three datasets. We trained the model using SGD with a weight decay of 0.0005, a momentum of 0.9. The learning rate was initialized with 0.1 and was divided by 10 when the loss stops decreasing.

### 5.4. Baseline Models

350 To verify the effectiveness of the proposed MMC-Net, we implemented two baseline models: MM-Net and MC-Net.

**MM-Net:** a baseline model for multimodal matching as illustrated in Fig. 2(a). It only contains the matching component of the MMC-Net (see Fig. 3), which is trained with the matching loss.

355 **MC-Net:** a baseline model for multimodal classification as illustrated in Fig. 2(b). It has the similar architecture as the MMC-Net, however, it does not compute the matching loss between visual and textual features. MC-Net is only trained with the classification loss.

### 5.5. Results on Multimodal Matching

We conducted the cross-modal retrieval experiments on the four datasets. To verify the effectiveness of adding a classification component in MMC-Net, we use the baseline MM-Net for comparison.



Table 1: Image-to-text retrieval results compared between MMC-Net and MM-Net. The proposed MMC-Net can outperform the baseline MM-Net with considerable gains across all the four datasets.

Method	Pascal Sentence			MSCOCO			Flowers			CUB-Bird		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
MM-Net	47.0	85.0	92.0	55.5	84.2	91.4	58.1	82.5	88.5	32.5	61.4	72.5
MMC-Net	52.0	87.0	93.0	57.0	85.8	92.7	78.7	93.9	96.0	39.2	66.9	76.4

Table 2: Text-to-image retrieval results compared between MMC-Net and MM-Net. Compared to MM-Net, MMC-Net can achieve better retrieval results on the four datasets.





Method	Pascal Sentence			MSCOCO			Flowers			CUB-Bird		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
MM-Net	38.4	80.6	88.6	44.7	79.5	89.5	32.7	46.4	52.9	18.3	25.6	28.8
MMC-Net	41.0	81.2	92.5	46.2	80.8	90.5	43.6	54.8	58.6	25.8	31.4	34.5

360 Table 1 and Table 2 report the results of image-to-text and text-to-image retrieval, respectively. Overall, MMC-Net can achieve considerable improvements over MM-Net for both I→T and T→I retrieval. These results reveal that the classification component in MMC-Net can help in improving the learning of embedding features in the matching component. Moreover, we can observe more insights from these results as follows:



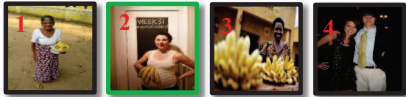




- 365 • By comparison with MM-Net, MMC-Net yields more performance gains on Flowers and CUB-Bird than Pascal Sentence and MSCOCO. For example, the performance gap between MMC-Net and MM-Net is below 5% on Pascal Sentence and MSCOCO, but above 5% on Flowers and CUB-Bird across all the measurements. One reason is that both Flowers and CUB-Bird are fine-grained datasets, and the textual descriptions cannot fully represent the discrimination among different samples. Hence, the results of MM-Net are limited on these two datasets. 370 Instead, MMC-Net can make use of fine-grained class labels to enhance the discriminative abilities when matching images and texts.
- The results of T→I retrieval are lower than those of the I→T retrieval on the four datasets. This is because each image can retrieve several related textual descriptions, but one text 375 corresponds to only one matched image. We believe that refining the datasets is a favorable solution to narrow the performance gap between the I→T and T→I retrieval.
- For Flowers and CUB-Bird, their results are still not satisfactory, especially for the T→I retrieval. Currently, the fine-grained multimodal matching still remains challenging in the research field, but it is a promising research direction in the future.

380 In addition, we present the qualitative retrieval results as shown in Fig. 8. We can observe that MMC-Net obtains better retrieved candidates than MM-Net, for both I→T and T→I retrieval. Furthermore, we visualize the visual and textual embedding features learned in the matching component of MMC-Net. As mentioned earlier in 5, it has shown the embedding map with the MSCOCO test

set. Similarly, we illustrate the embedding features with the Pascal Sentence test set that consists of 100 images and 500 texts. As shown in Fig. 9(a), each point corresponds to one sample (an image or a text) from the 20 Pascal categories. Also, we detail the embedding features per category in Fig. 9(b1) to (b20). It is clear to observe the matching relation between images and texts.

Query Image	MM-Net: Retrieved texts	MMC-Net: Retrieved texts
Pascal Sentence 	<ol style="list-style-type: none"> <li>1. People riding tandem bicycle.</li> <li>2. Two prop-leg ride a yellow tandem bike while someone helps.</li> <li>3. Young man wearing jeans and helmet rides his motorcycle in front of a small crowd.</li> <li>4. A man wearing a helmet does a wheelie on a motorcycle as a crowd watches.</li> </ol>	<ol style="list-style-type: none"> <li>1. Two prop-leg ride a yellow tandem bike while someone helps.</li> <li>2. People riding tandem bicycle.</li> <li>3. Two people riding a tandem bicycle while wearing lira racing outfits.</li> <li>4. Young man wearing jeans and helmet rides his motorcycle in front of a small crowd.</li> </ol>
MSCOCO 	<ol style="list-style-type: none"> <li>1. a man putting together a kite on the floor of a room.</li> <li>2. man folding banner while holding stick in unfinished carpet.</li> <li>3. a man folding a giant paper airplane on the floor.</li> <li>4. a tiny toddler carries a giant bookbag and bag.</li> </ol>	<ol style="list-style-type: none"> <li>1. a man putting together a kite on the floor of a room.</li> <li>2. man folding banner while holding stick in unfinished carpet.</li> <li>3. a man folding a giant paper airplane on the floor.</li> <li>4. a man inside a room putting together a white kite.</li> </ol>
Flowers 	<ol style="list-style-type: none"> <li>1. this flower is pink and white in color, with petals that have pink veins.</li> <li>2. this pink flower has several filaments sticking out of the receptacle.</li> <li>3. this flower has pale pink petals with veins and a white center.</li> <li>4. this flower has petals that are pink with long stamen.</li> </ol>	<ol style="list-style-type: none"> <li>1. this flower is pink and white in color, with petals that have pink veins.</li> <li>2. this flower has pale pink petals with veins and a white center.</li> <li>3. this flower has very light pink petals that have darker pink veins, a yellow ovary, and white stamen.</li> <li>4. this pink flower has several filaments sticking out of the receptacle.</li> </ol>
CUB-Bird 	<ol style="list-style-type: none"> <li>1. a dark brown beak with a long beak and large wingspan.</li> <li>2. this bird has a dark grey color, with a large bill and long wingspan.</li> <li>3. this dull colored bird is brown all over, has large wings and a long large bill.</li> <li>4. a bird with a large, hooked bill, white superciliary and cheek patch, brown crown, and brown body.</li> </ol>	<ol style="list-style-type: none"> <li>1. a dark brown beak with a long beak and large wingspan.</li> <li>2. large bird that is complete brown, with white stripes littering it's wings and a long blunted bill.</li> <li>3. a bird with a large, hooked bill, white superciliary and cheek patch, brown crown, and brown body.</li> <li>4. this dull colored bird is brown all over, has large wings and a long large bill.</li> </ol>

(a) Image-to-text retrieval

Query Text	MM-Net: Retrieved images	MMC-Net: Retrieved images
Pascal Sentence An Swiss-Air flight has just taken off from a runway.		
MSCOCO a woman in white shirt holding bananas next to door.		
Flowers the bright orange petals are highlighted by brown spots and the prominent stamen are topped with dark brown anthers.		
CUB-Bird this bird is light brown, has a long hooked bill, and looks dumb.		

(b) Text-to-image retrieval

Figure 8: Image-text retrieval examples on the datasets. For (a) image-to-text retrieval, the ground-truth matching texts are in green. For (b) 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. For the I→T and T→I retrieval, MMC-Net can retrieve more accurate candidates than MM-Net.

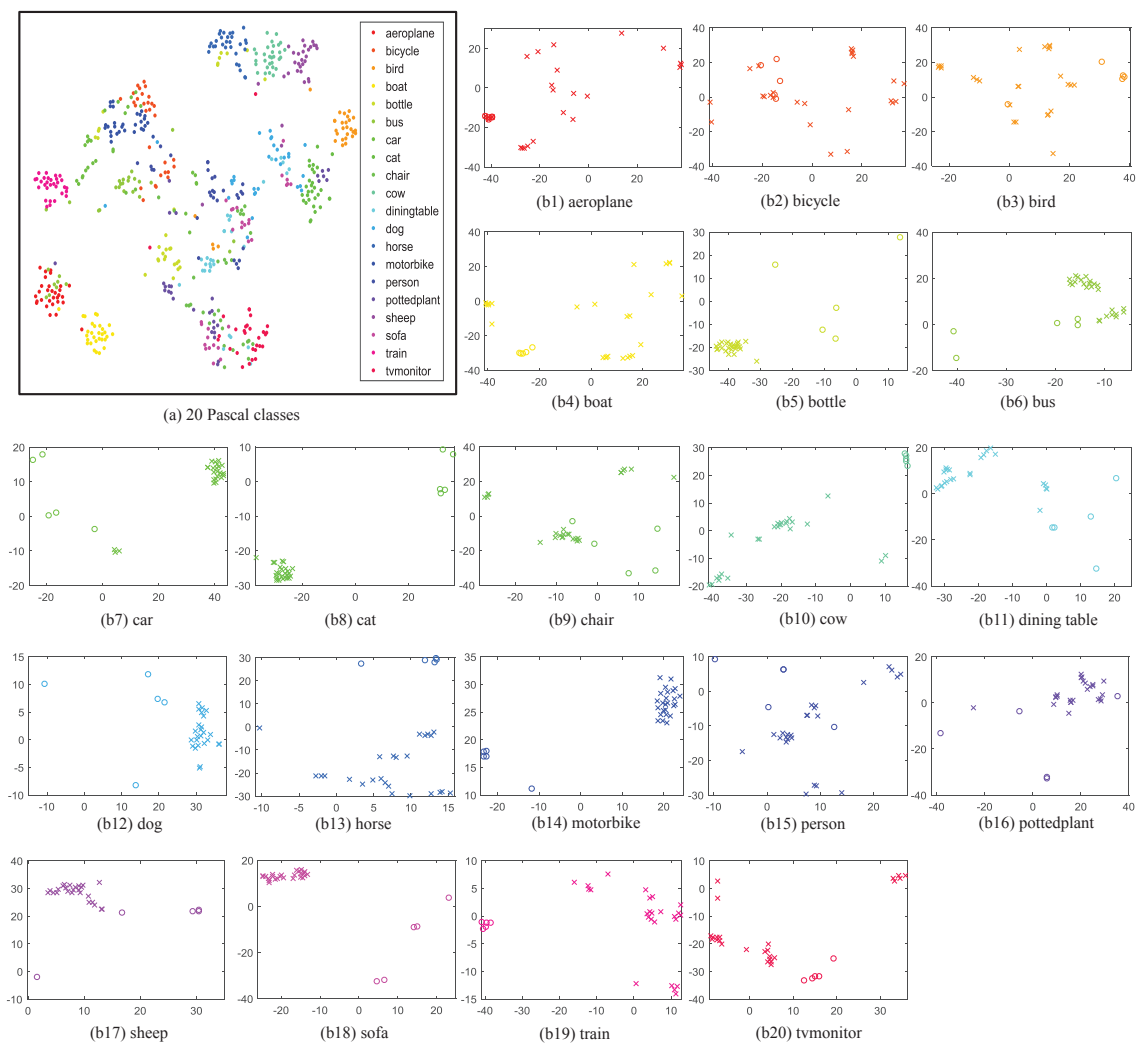


Figure 9: Visualization of the embedding features of the Pascal Sentence test set. (a) 100 images and 500 texts are projected to the 2-D space based on the t-SNE algorithm. They are labelled with the corresponding categories. (b1)-(b20) The embedding map for each category. The images and texts are described by 'O' and 'X', respectively. For some categories (e.g. 'bicycle', 'bird', 'boat'), we can see that MMC-Net can learn the desirable matching between images and texts, but it is still difficult for other categories (e.g. 'bus', 'cat', 'motorbike').

### 5.6. Results on Multimodal Classification

Next, we conducted the multimodal classification experiments on the datasets. To demonstrate the benefit of using a matching component for classification, we compare the MMC-Net model with the baseline MC-Net model. Table 3 reports the classification results, where MMC-Net achieves consistent improvements over MC-Net across all the four datasets. It shows that the matching component is able to promote the classification component due to combining the embedding features to generate more discriminative multimodal representations. Also, MMC-Net has a generalization ability for different types of classification datasets, including either natural images or fine-grained images.

In addition, we show some classification examples in Fig. 10. It can be seen that MMC-Net can predict more accurate classes than MC-Net. Note that MSCOCO has multiple ground-truth labels. Furthermore, we visualize the multimodal representation captured from the CBP module in MMC-Net. Figure 11(a) and (b) illustrate the multimodal features with the Flowers and CUB-Bird test images, respectively. We can observe clear separations among different categories.

Table 3: Comparison of the multimodal classification accuracy between MMC-Net and MC-Net. For the four datasets, MMC-Net can outperform MC-Net with consistent performance gains.

Method	Pascal Sentence	MSCOCO	Flowers	CUB-Bird
MC-Net	71.0	77.6	94.0	80.7
MMC-Net	74.0	79.3	95.2	82.4





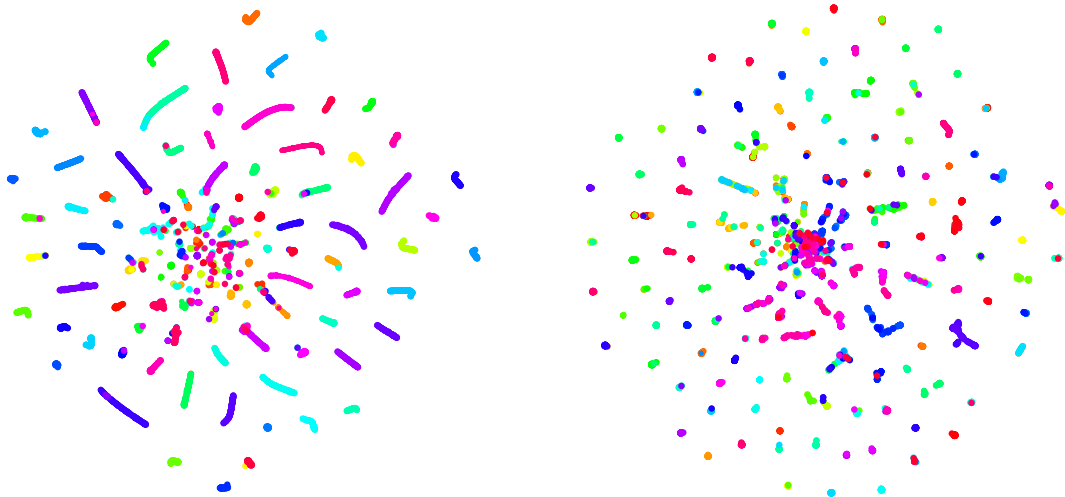
	Pascal Sentence	MSCOCO	Flowers	CUB-Bird
	 A striped sofa and office chairs are near a ping pong table.	 a tennis player wiping his face off with a towel.	 the petals of the flower are purple in color and have green stems with green sepals.	 a bird with a medium yellow bill, white body webbed feet and gray wings.
MC-Net	1. chair 2. tv/monitor 3. sofa 4. diningtable 5. bottle	1. person 2. chair 3. sports ball 4. tennis racket 5. dining table	1. bolero deep blue 2. garden phlox 3. canterbury bells 4. bougainvillea 5. snapdragon	1. Glaucous winged Gull 2. Ring billed Gull 3. California Gull 4. Herring Gull 5. Heermann Gull
MMC-Net	1. sofa 2. chair 3. Diningtable 4. tv/monitor 5. potted plant	1. person 2. tennis racket 3. chair 4. bench 5. sports ball	1. canterbury bells 2. bolero deep blue 3. foxglove 4. stemless gentian 5. garden phlox	1. Herring_Gull 2. California_Gull 3. Western_Gull 4. Ring_billed_Gull 5. Slaty_backed_Gull

Figure 10: Multimodal classification examples on the datasets. Given an input image-text pair, the Top-5 predictions are estimated based on MC-Net and MMC-Net. The ground-truth classes are in green. By comparison, MMC-Net obtains more accurate predictions than MC-Net.



(a) 102 flower categories

(b) 200 bird categories

Figure 11: Visualizing the multimodal features learned in the classification component of MMC-Net. (a) 6149 images from the Flowers test set. (2) 5794 images from the CUB-Bird test set. Images are properly grouped into different clusters as shown in color.

Table 4: Effect of the mini-batch size on the performance of MMC-Net. We train the model with different mini-batch sizes and compare their retrieval results on MSCOCO.

Method	Image-to-Text			Text-to-Image		
	R@1	R@5	R@10	R@1	R@5	R@10
batch size=100	42.5	74.6	87.4	36.6	73.8	86.8
batch size=250	52.6	83.3	91.7	43.0	79.5	89.4
batch size=500	56.6	85.3	92.7	46.0	80.5	90.1
batch size=1000	56.2	85.8	93.0	46.5	80.5	90.1
batch size=1500	57.0	85.8	92.7	46.2	80.8	90.5
batch size=2000	56.7	85.5	92.8	46.7	80.6	90.4

### 5.7. Ablation Study

In the following, we perform an ablation study to provide more insights into MMC-Net.

#### 5.7.1. Analysis of parameters

405 First of all, we analyze the effects of three key parameters used in MMC-Net.

**Effect of the mini-batch size.** Since the loss function for multimodal matching aims to search for hard negative samples, it is essential to define a large mini-batch to increase the search space. For example, we selected a mini-batch size of 1500 for MSCOCO due to its large-scale data. To study the effect of varying different batch sizes, we used different batch sizes to train MMC-Net and tested their performance. Considering the number of negative pairs in each mini-batch is  $K = 50$  410 for MSCOCO, we varied the batch size with 100, 250, 500, 1000, 1500 and 2000. Table 4 compares the retrieval results on MSCOCO with different batch sizes. We can observe that the performance is low when the batch size is 100. By increasing the size to 500, it can achieve significant gains across

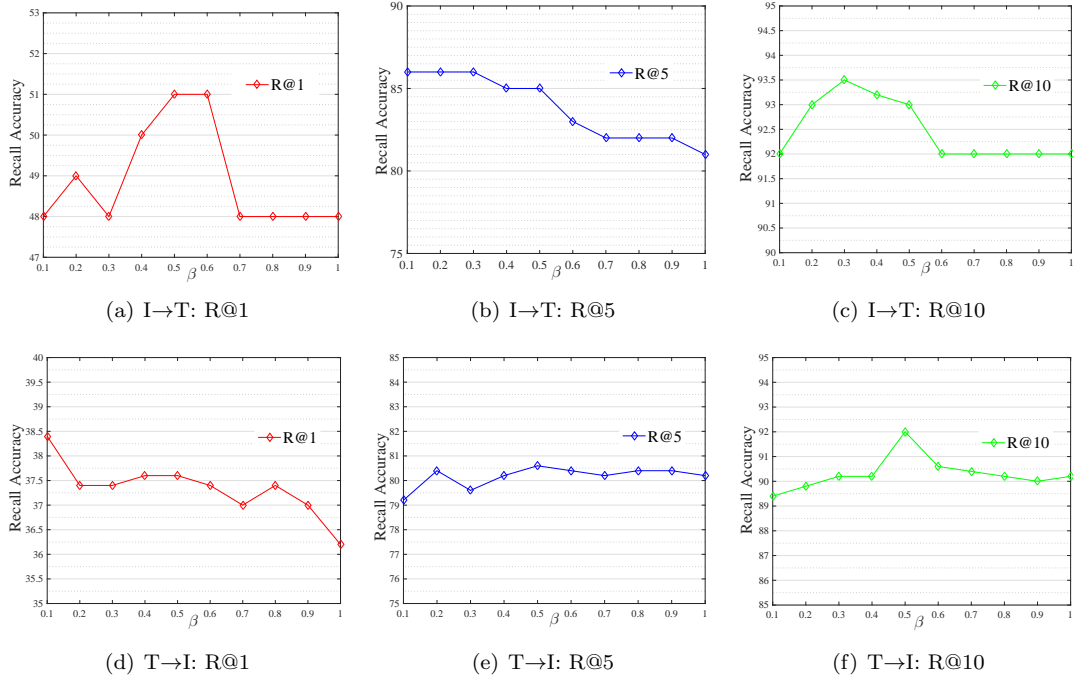


Figure 12: Effect of the parameter  $\beta$  on the performance of MMC-Net. The retrieval results on Pascal Sentence are reported. We select  $\beta = 0.5$  by comparing these results.

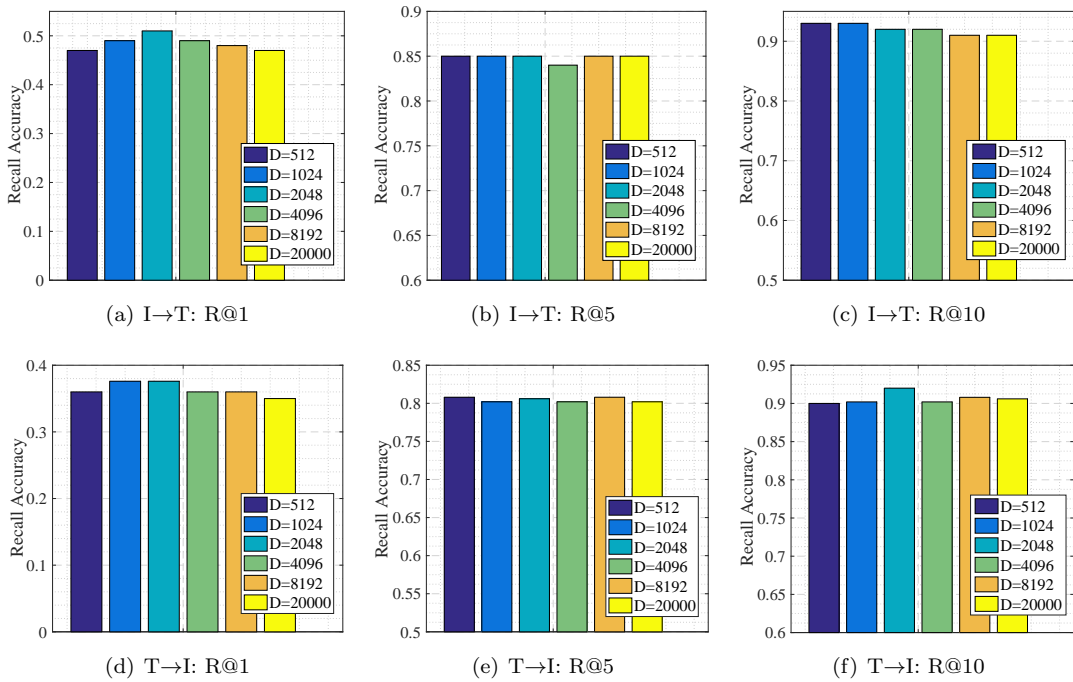


Figure 13: Effect of the parameter  $D$  on the performance of MMC-Net. We present the retrieval results on Pascal Sentence by using different sizes of  $D$ . We select  $D = 2048$  that can bring better results.

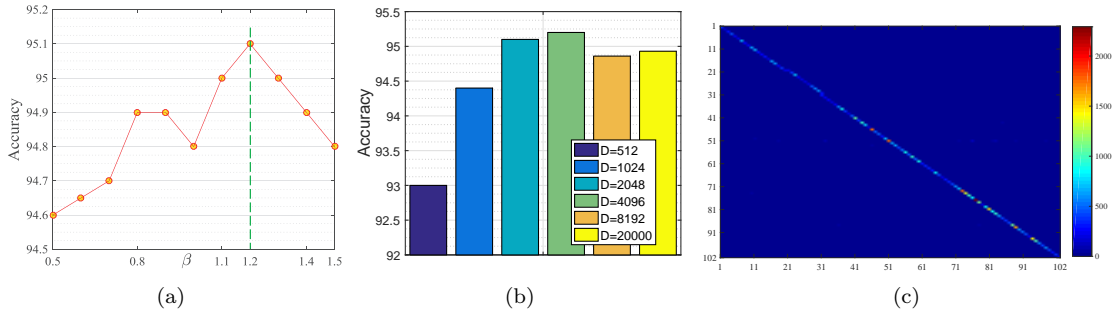


Figure 14: Effect of the parameters on the performance of MMC-Net. We report the Top-1 classification results on Flowers. (a) Analysis of the parameter  $\beta$ . (b) Analysis of the parameter  $D$ . (c) Confusion matrix of 102 Flowers classes. The diagonal line demonstrates the high accuracy per flower class.

all the measurements. We further raise the size to 2000, however there is no important influence  
 415 on the results. Finally, we select batch size=1500 due to its slightly superior results.

**Effect of the Parameter  $\beta$ .** Recall that MMC-Net is trained by integrating the matching and classification loss, we use the parameter  $\beta$  to balance the weights of the two loss functions as defined in Eq. 13. This experiment aims to analyze the effect of  $\beta$  on the performance. Figure 12 shows the cross-modal retrieval results on Pascal Sentence. The R@1, R@5 and R@10 results are  
 420 shown separately, when  $\beta$  varies from 0.1 to 1. We pick  $\beta = 0.5$  by fully comparing these results.

**Effect of the Parameter  $D$ .** In the classification component, a CBP module can integrate visual and textual embedding features into a  $D$ -dimension multimodal vector. In this experiment, we analyze  $D$  with  $\{512, 1024, 2048, 4096, 8192, 20000\}$ , which are all significantly lower than the original bilinear pooling vector (*i.e.*  $512 \times 512 = 262144$ ). In Fig. 13, we present the compared  
 425 results on Pascal Sentence. When  $D = 2048$ , MMC-Net can achieve better results compared to others.

Since MSCOCO is also composed of scene images like Pascal Sentence, it is straightforward and general to employ the same parameters  $\beta$  and  $D$ . In contrast, Flowers and CUB-Bird are commonly used for fine-grained recognition. It is needed to evaluate their parameters different  
 430 from Pascal Sentence and MSCOCO. To this end, we estimated the effects of the parameters on the classification accuracy of Flowers, and then applied the same parameters to CUB-Bird for generalization. Figure 14 presents the analysis of parameters on Flowers. As for the parameter  $\beta$  shown in Fig. 14(a), the best precision accuracy, 95.1%, is reached by  $\beta = 1.2$ . As shown in Fig. 14(b), the accuracy is maximized (*i.e.* 95.2%) when  $D = 4096$ . In the experiments, we set  
 435  $\beta = 1.2$  and  $D = 4096$  for Flowers and CUB-Bird. Additionally, we show the confusion matrix of 102 Flowers categories in Fig. 14(c).

Table 5: Analysis of the fusion module used in MM-Net and MMC-Net. The R@K results on Pascal Sentence are reported. By comparison, the convolutional fusion module can achieve better results than others.

Method	Fusion module	Image to Text			Text to Image		
		R@1	R@5	R@10	R@1	R@5	R@10
MM-Net	No	45.0	82.0	91.0	35.6	75.8	87.0
MM-Net	summation	46.0	83.0	91.0	36.8	77.6	87.6
MM-Net	Multiplication	46.0	84.0	91.0	37.2	78.4	87.6
MM-Net	Convolution	47.0	85.0	92.0	38.4	80.6	88.6
MMC-Net	No	51.0	85.0	92.0	37.6	80.6	92.0
MMC-Net	summation	51.0	86.0	92.0	38.4	81.0	92.0
MMC-Net	Multiplication	51.0	86.0	92.0	39.0	81.0	92.0
MMC-Net	Convolution	52.0	87.0	93.0	41.0	81.2	92.5

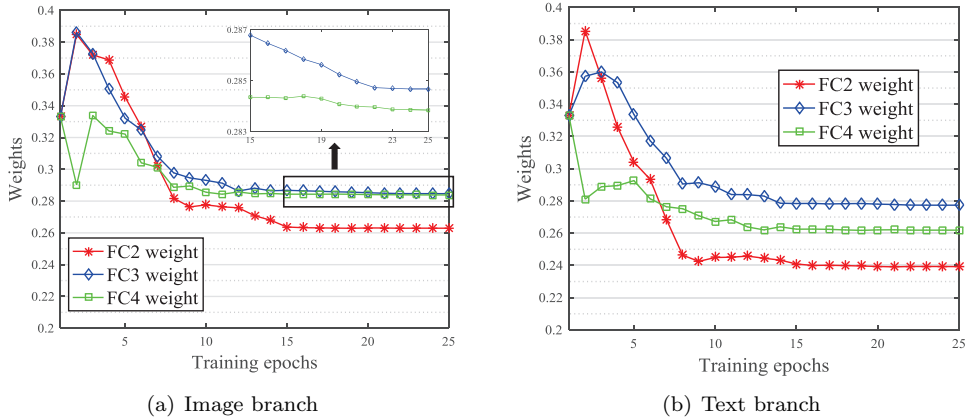


Figure 15: Analysis of adaptive weights learned in the fusion module of the image branch and text branch. This test is performed on Pascal Sentence.

### 5.7.2. Analysis of the fusion module.

This test aims to verify the effectiveness of using the fusion module in the matching component. We build a convolutional fusion module in MMC-Net which can also be applied on the baseline  
440 MM-Net. In Table 5, we report the results for both MMC-Net and MM-Net on the Pascal Sentence test set. We can see that using a fusion module can improve all R@K performance measurements by a considerable margin, compared to the counterparts without using any fusion module. For an additional comparison, we further implement two simple fusion modules: element-wise summation and multiplication. Their results are inferior to those of the convolutional fusion, because they do  
445 not consider the weights of different layers. Instead, the convolutional fusion can learn adaptive weights to produce a superior fused feature while spending only three parameters. All the weights can be learned dynamically and adaptively with other network parameters, without introducing any manual tuning.

Moreover, we delve into analyzing the adaptive weights of different layers learned in the convo-  
450 lutional fusion module. Figure 15 demonstrates their distributions during the training procedure. Since there are three layers (*i.e.* FC2, FC3, FC4) in the fusion module, we initialize their weights



Table 6: Analysis of the CBP module in MMC-Net. The R@K results on Pascal Sentence are reported, which demonstrate the effectiveness and efficiency of using the CBP module.

Method	Dimension	Image to Text			Text to Image		
		R@1	R@5	R@10	R@1	R@5	R@10
MMC-Net with FC	1024	50.0	86.0	92.0	39.6	80.4	90.0
MMC-Net with BP	262144	53.0	88.0	93.0	41.5	81.5	92.5
MMC-Net with CBP	2048	52.0	87.0	93.0	41.0	81.2	92.5

with 0.33. It can be seen that the weights in both of image and text branches tend to be stable after a number of training epochs. In particular, the weight of the FC2 layer is smallest, which demonstrates that its feature representation is less powerful than those of the FC3 and FC4 layers. In addition, the FC4 layer is less important than the FC3 layer. This implies that increasing the depth may not improve the representation learning any more. Hence, we do not develop more layers behind the FC4. Lastly, all the three layers play essential roles in the fusion module, even though they learn individual and different weights.

### 5.7.3. Analysis of the CBP module.

We conduct this experiment to test the use of the CBP module in MMC-Net. For comparison, we present two other methods to integrate the visual and textual features. For the first method, we concatenate the two features to construct a multimodal representation and then feed it into a fully-connected (FC) layer to perform the classification. The second one is using the traditional bilinear pooling (BP) to produce a high-order multimodal representation. Table 6 reports the compared results of different classification modules. The model with CBP can obtain considerable improvements over the one with FC. The MMC-Net with BP achieves better results than other methods, while its multimodal representation has higher dimensionality. On the contrary, CBP can maintain the high accuracy and efficiency.

### 5.7.4. Analysis of combining vision and language

This experiment is used to verify the advantage of incorporating visual and textual representations. As reported in Table 7, we compare the results between combining visual and textual features (*i.e.* MMC-Net) and using only visual features. We can observe that combining vision and language can achieve significantly superior accuracies on Flowers and CUB-Bird. Although visual features can enable the models to achieve promising performance, the informative textual features

Table 7: Analysis of combining vision and language. We report the Top-1 classification rates on Flowers and CUB-Bird. The model with both vision and language outperforms the model with only vision.

Method	Flowers	CUB-Bird
Only Vision	92.2	78.8
Vision and Language	95.2	82.4

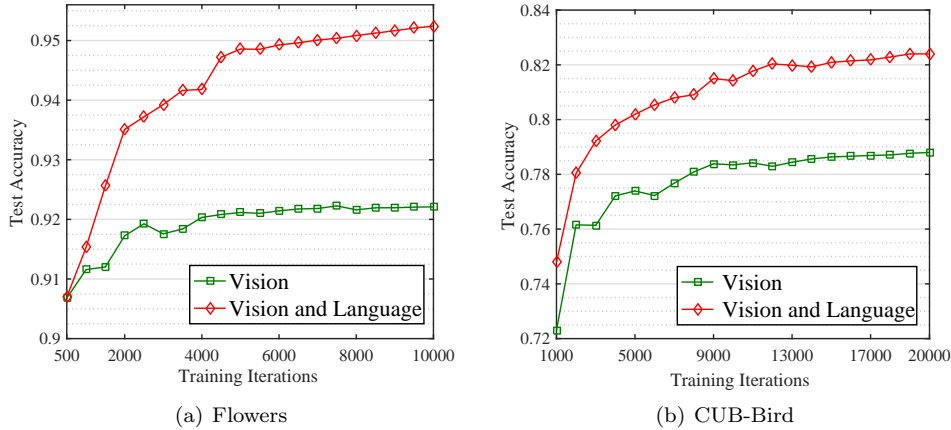


Figure 16: Illustration of the test classification rates during the training iterations. Incorporating language and vision is significant to improve the performance, compared to only using visual information.

Table 8: Analysis of image encoders. The image feature dimensions are also presented. MMC-Net has better matching results on MSCOCO than DSPE [6].

Method	Image encoder	Dimension	Image to Text			Text to Image		
			R@1	R@5	R@10	R@1	R@5	R@10
DSPE	VGG-19	4096	40.7	74.2	85.3	33.5	68.7	83.2
MMC-Net	VGG-19	4096	46.0	79.7	89.2	38.9	73.5	87.5
DSPE	ResNet-152	2048	53.1	82.7	90.2	43.5	78.2	88.9
MMC-Net	ResNet-152	2048	57.0	85.8	92.7	46.2	80.8	90.5

475 can further help improve the classification accuries. This shows the effectiveness of capturing multi-modal representations from both vision and language. Furthermore, Fig. 16 analyzes the test rates during the training iterations. It can be seen that the vision and language model can consistently outperform the vision model in the entire training stage.

### 5.7.5. Analysis of image encoders

480 As aforementioned in Sec. 3.2, we employ the ResNet-152 model to encode the input image. In this experiment, we aim to study the effect of different image encoders. For a fair comparison with DSPE [6], we provide the results of MMC-Net with VGG-19. Also, we implement the DSPE with ResNet-152. Table 8 reports the compared results on MSCOCO. For both VGG-19 and ResNet-152, our MMC-Net can outperform DSPE across all the measurements. We should realize that the improvements of MMC-Net come from two aspects. First, the matching component in MMC-Net has more layers than that of DSPE, *i.e.* four layers *v.s.* two layers. Second, MMC-Net utilizes a classification component to help improve the matching performance. This is the main motivation in this work. Note that, both MMC-Net and DSPE in Table 8 use the Mean vector to encode the input text. In [6], they also present another expensive textual representation using the Hybrid  
 485  
 490 Gaussian-Laplacian mixture model (HGLMM) [46], *i.e.* a 18000-dimension vector. Currently, we

do not introduce HGLMM to MMC-Net, even though it can help increase the performance.

Table 9: Comparison with other state-of-the-art approaches on Pascal Sentence for image-text retrieval. Best results are in bold face. The CNN and RCNN models for [60] and [13] are based on AlexNet [61].

Method	Image encoder	Text encoder	Image to Text		Text to Image	
			R@1	R@5	R@1	R@5
SDT-RNN [60]	CNN	DT-RNN	23.0	45.0	16.4	46.6
kCCA [60]	CNN	Word vector	21.0	47.0	16.4	41.4
DeViSE [52]	AlexNet	skip-gram	17.0	57.0	21.6	54.6
SDT-RNN [60]	RCNN	DT-RNN	25.0	56.0	25.4	65.2
DFE [13]	RCNN	Word vector	39.0	68.0	23.6	65.2
MDL-CW [62]	feature from [63]	feature from [63]	34.0	70.0	35.2	72.6
Mean Vector [46]	VGG-16	Mean vector	52.5	83.2	<b>44.9</b>	84.9
GMM+HGLMM [46]	VGG-16	HGLMM	<b>55.9</b>	86.2	44.0	<b>85.6</b>
Proposed MMC-Net	ResNet-152	Mean vector	52.0	<b>87.0</b>	41.0	81.2

Table 10: Comparison with other state-of-the-art approaches on MSCOCO for image-text retrieval. Best results are in bold face.

Method	Image encoder	Text encoder	Image to Text			Text to Image		
			R@1	R@5	R@10	R@1	R@5	R@10
DVSA [14]	RCNN	RNN	38.4	69.9	80.5	27.4	60.2	74.8
Mean vector [46]	VGG-16	Mean vector	33.2	61.8	75.1	24.2	56.4	72.4
GMM+HGLMM [46]	VGG-16	HGLMM	39.4	67.9	80.9	25.1	59.8	76.6
m-RNN [3]	VGG-16	RNN	41.0	73.0	83.5	29.0	42.2	77.0
RNN-FV [64]	VGG-19	RNN	41.5	72.0	82.9	29.2	64.7	80.4
mCNN(ensemble) [15]	VGG-19	CNN	42.8	73.1	84.1	32.6	68.6	82.8
DSPE [6]	VGG-19	Mean vector	40.7	74.2	85.3	33.5	68.7	83.2
DSPE [6]	VGG-19	HGLMM	50.1	79.7	89.2	39.6	75.2	86.9
2WayNet [16]	VGG-16	HGLMM	55.8	75.2	-	39.7	63.3	-
Proposed MMC-Net	ResNet-152	Mean vector	<b>57.0</b>	<b>85.8</b>	<b>92.7</b>	<b>46.2</b>	<b>80.8</b>	<b>90.5</b>

### 5.8. Comparison with Other Approaches

For Pascal Sentence and MSCOCO, we compare our matching results with other state-of-the-art approaches. As reported in Table 9 and 10, MMC-Net can achieve competitive performance with the state-of-the-art. To be more specific, the method in [46] is effective on small-scale datasets, so it can obtain state-of-the-art results on Pascal Sentence. However, it does not have a strong generalization on large-scale datasets, for example their results on MSCOCO are not quite competitive. In contrast, the proposed MMC-Net maintains the high performance on both of small-scale and large-scale datasets. Moreover, we show the image and text encoders used in different approaches. Both of DSPE [6] and 2WayNet [16] extracted the visual features based on the VGG-19 model, while they rely on a more complicated HGLMM textual representation [46] than the Mean vector used in MMC-Net. As early discussed (Sec. 3.2), we did not use the HGLMM representation in order to maintain the training efficiency. For a fair comparison, MMC-Net with VGG-19 and Mean vector (in Table 8) can outperform DSPE with significant improvements, and can compete with

Table 11: Comparison with other approaches on Flowers and CUB-Bird. Best results are in bold face. The methods in the upper part fine-tune the original CNN models, however, the ones in the lower part do not perform the fine-tuning process. We do not use the bounding box annotations in the datasets. Note that, we use the numbers to describe the depth of the image encoders. The dimension of MMC-Net indicates the multimodal representation extracted from CBP.

Method	Image encoder	Finetune	Dimension	Flowers	CUB-Bird
Deep Optimized [65]	CNN-16	Yes	4096	91.3	67.1
Part R-CNN [66]	DeCAF-8	Yes	4096	-	76.5
Two-level attention [67]	AlexNet-8	Yes	4096	-	77.9
Deep LAC [68]	AlexNet-8	Yes	12288	-	80.3
NAC-const [69]	AlexNet-8	Yes	4096	91.7	68.5
NAC-const [69]	VGG-19	Yes	4096	<b>95.3</b>	81.0
Bilinear CNN [38]	VGG-16	Yes	250k	-	84.0
PD+FC+SWFV-CNN [70]	VGG-16	Yes	70k	-	<b>84.5</b>
MsML+ [71]	DeCAF-8	No	134016	89.5	67.9
BoSP [72]	VGG-16	No	5120	94.0	-
RI-Deep [73]	VGG-19	No	4096	94.0	72.6
ProCRC [74]	VGG-19	No	5120	94.8	78.3
MG-CNN [75]	VGG-19	No	12288	-	81.7
Proposed MMC-Net	ResNet-152	No	4096	<b>95.2</b>	<b>82.4</b>

505 2WayNet while it uses the HGLMM representation. Lastly, we clarify that any common feature encoders for images and texts can be potentially adopted to MMC-Net. Exploring more efficient feature encoders is a fundamental and promising work.

For Flowers and CUB-Bird, we compare the fine-grained classification results with the state-of-the-art. Table 11 reports the comparison details. Since the compared methods do not utilize 510 textual representations, we instead show the CNN model used in the image encoder and the network depth. Note that, these approaches are divided into two groups based on whether the CNN model is finetuned on the target dataset. First, it can be seen that, MMC-Net achieves better results than other approaches without performing the fine-tuning step. Second, MMC-Net can even compete with the approaches with the finetuning step. For example, our results on Flowers is competitive 515 with NAC-const [69]. Also, our approach is superior over most approaches on CUB-Bird, except Bilinear CNN [38] and PD+FC+SWFV-CNN [70]. However, we can see that both [38] and [70] produce a significantly more expensive feature vector than MMC-Net. We should realize that additional fine-tuning techniques have potential to improve performance, but are not the focus of this work. Our competitive results are partly due to the use of the ResNet-152 model, while we believe this should not decrease the effectiveness of our approach.

Table 12: Summary of the parameters used in the MMC-Net for matching and classification, and the time for running the multi-stage training algorithm.

Dataset	#Params for matching	#Params for classification	Time (hours)
Pascal Sentence	~8 millions	~41,000	~0.3
MSCOCO	~8 millions	~164,000	~7.0
Flowers	~8 millions	~418,000	~0.5
CUB-Bird	~8 millions	~820,000	~1.3

520

### 5.9. Computational Cost

We conducted the experiments on a NVIDIA TITAN X card with 12 GB memory. In practice, we first extracted visual and textual features for all training samples using the off-the-shelf feature encoders. Then, we take as input these input features for the matching and classification components. Since the network parameters in MMC-Net are not expensive, it is feasible and rewarding to use a large mini-batch size to improve the training (in Sec.5.3). In Table 12, we show the training parameters in the matching and classification component, and the multi-stage training time cost on the four datasets. The MSCOCO dataset consumes more training time due to its large-scale data. In summary, MMC-Net is an efficient network with a decent model complexity.

## 6. Conclusion and Future Work

In this work, we proposed a unified network for joint multimodal matching and classification. The proposed MMC-Net can simultaneously learn latent embeddings in the matching component, and generate a multimodal representation vector in the classification component. Consequently, the two components can help promote each other by combining their loss functions together. We evaluated our approach on four well-known multimodal datasets. The experimental results demonstrated the robustness and effectiveness of the MMC-Net model, compared to the baseline models. In addition, our approach achieved competitive results with the state-of-the-art approaches. The results showed its promising generalization for diverse multimodal tasks related to matching or classification.

In the future, it is feasible to advance the three components in the MMC-Net. For example, fine-tuning the feature encoders on the target datasets, adding intermediate supervisory signals in the matching component, and improving the compact bilinear pooling module in the classification component. In addition, it is straightforward to adapt MMC-Net to a wider variety of multimodal tasks, including image captioning, visual question answering, and video summarization. Moreover, the attention mechanism is potential to be introduced in the MMC-Net.

### 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

### References

- 555 [1] O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and tell: A neural image caption generator, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3156–3164, 2015.
- [2] A. Karpathy, L. Fei-Fei, Deep Visual-Semantic Alignments for Generating Image Descriptions, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3128–3137, 2015.
- [3] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, A. Yuille, Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN), in: International Conference on Learning Representations (ICLR), 2015.
- 560 [4] F. Feng, X. Wang, R. Li, Cross-modal Retrieval with Correspondence Autoencoder, in: ACM International Conference on Multimedia (MM), 7–16, 2014.
- [5] D. Rafailidis, S. Manolopoulou, P. Daras, A unified framework for multimodal retrieval, *Pattern Recognition* 46 (12) (2013) 3358–3370.
- [6] L. Wang, Y. Li, S. Lazebnik, Learning Deep Structure-Preserving Image-Text Embeddings, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5005–5013, 2016.
- 565 [7] J. Lei Ba, K. Swersky, S. Fidler, R. Salakhutdinov, Predicting Deep Zero-Shot Convolutional Neural Networks Using Textual Descriptions, in: IEEE International Conference on Computer Vision (ICCV), 4247–4255, 2015.
- [8] S. Reed, Z. Akata, H. Lee, B. Schiele, Learning Deep Representations of Fine-Grained Visual Descriptions, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 49–58, 2016.
- 570 [9] E. Kodirov, T. Xiang, S. Gong, Semantic Autoencoder for Zero-Shot Learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3174–3183, 2017.
- [10] L. Zhang, T. Xiang, S. Gong, Learning a Deep Embedding Model for Zero-Shot Learning, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021–2030, 2017.
- [11] H. Hotelling, Relations between two sets of variates, *Biometrika* 28 (1936) 321–377.
- 575 [12] R. Kiros, R. Salakhutdinov, R. S. Zemel, Unifying visual-semantic embeddings with multimodal neural language models, in: Neural Information Processing Systems (NIPS), Deep Learning Workshop, 2014.
- [13] A. Karpathy, A. Joulin, F. Li, Deep Fragment Embeddings for Bidirectional Image Sentence Mapping, in: Neural Information Processing Systems (NIPS), 1889–1897, 2014.
- [14] A. Karpathy, F.-F. Li, Deep Visual-Semantic Alignments for Generating Image Descriptions, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3128–3137, 2015.
- 580 [15] L. Ma, Z. Lu, L. Shang, H. Li, Multimodal Convolutional Neural Networks for Matching Image and Sentence, in: IEEE International Conference on Computer Vision (ICCV), 2623–2631, 2015.
- [16] A. Eisenschat, L. Wolf, Linking Image and Text with 2-Way Nets, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4601–4611, 2017.
- 585 [17] H. Nam, J.-W. Ha, J. Kim, Dual Attention Networks for Multimodal Reasoning and Matching, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 299–307, 2017.
- [18] Y. Liu, Y. Guo, E. M. Bakker, M. S. Lew, Learning a Recurrent Residual Fusion Network for Multimodal Matching, in: IEEE International Conference on Computer Vision (ICCV), 4107–4116, 2017.
- [19] K. Simonyan, A. Zisserman, Two-stream Convolutional Networks for Action Recognition in Videos, in: Neural Information Processing Systems (NIPS), 568–576, 2014.
- 590 [20] J. F. Hu, W. S. Zheng, J. Lai, J. Zhang, Jointly Learning Heterogeneous Features for RGB-D Activity Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (11) (2017) 2186–2200.
- [21] S. Lai, W. S. Zheng, J. F. Hu, J. Zhang, Global-Local Temporal Saliency Action Prediction, *IEEE Transactions on Image Processing* 27 (5) (2018) 2272–2285.

- 595 [22] P. L. Lai, C. Fyfe, Kernel and Nonlinear Canonical Correlation Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10 (05) (2000) 365–377.
- [23] P. Mineiro, N. Karampatziakis, A Randomized Algorithm for CCA, in: *Neural Information Processing Systems (NIPS) workshop*, 2014.
- [24] T. Michaeli, W. Wang, , K. Livescu, Nonparametric Canonical Correlation Analysis, in: *International Conference on Machine Learning (ICML)*, 1967–1976, 2016.
- 600 [25] Y. Gong, Q. Ke, M. Isard, S. Lazebnik, A Multi-View Embedding Space for Modeling Internet Images, Tags, and Their Semantics, *International Journal on Computer Vision* 106 (2) (2014) 210–233.
- [26] V. Ranjan, N. Rasiwasia, C. V. Jawahar, Multi-Label Cross-Modal Retrieval, in: *IEEE International Conference on Computer Vision (ICCV)*, 4094–4102, 2015.
- 605 [27] G. Andrew, R. Arora, K. Livescu, J. Bilmes, Deep Canonical Correlation Analysis, in: *International Conference on Machine Learning (ICML)*, 1247–1255, 2013.
- [28] F. Yan, K. Mikolajczyk, Deep Correlation for Matching Images and Text, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3441–3450, 2015.
- [29] W. Wang, R. Arora, K. Livescu, J. Bilmes, On Deep Multi-View Representation Learning, in: *International Conference on Machine Learning (ICML)*, 1083–1092, 2015.
- 610 [30] L. Ma, Z. Chen, L. Xu, Y. Yan, Multimodal deep learning for solar radio burst classification, *Pattern Recognition* 61 (2017) 573–582.
- [31] X. Bai, M. Yang, P. Lyu, Y. Xu, Integrating Scene Text and Visual Appearance for Fine-Grained Image Classification with Convolutional Neural Networks, *CoRR* abs/1704.04613.
- 615 [32] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, D. Parikh, VQA: Visual Question Answering, in: *IEEE International Conference on Computer Vision (ICCV)*, 2425–2433, 2015.
- [33] M. Malinowski, M. Rohrbach, M. Fritz, Ask Your Neurons: A Neural-Based Approach to Answering Questions About Images, in: *IEEE International Conference on Computer Vision (ICCV)*, 1–9, 2015.
- [34] H. Noh, P. Hongsuck Seo, B. Han, Image Question Answering Using Convolutional Neural Network With Dynamic Parameter Prediction, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 30–38, 2016.
- 620 [35] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, M. Rohrbach, Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding, in: *Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 457–468, 2016.
- 625 [36] Y. Gao, O. Beijbom, N. Zhang, T. Darrell, Compact Bilinear Pooling, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 317–326, 2016.
- [37] J. B. Tenenbaum, W. T. Freeman, Separating Style and Content with Bilinear Models, *Neural Computation* 12 (6) (2000) 1247–1283.
- [38] T.-Y. Lin, A. RoyChowdhury, S. Maji, Bilinear CNN Models for Fine-grained Visual Recognition, in: *IEEE International Conference on Computer Vision (ICCV)*, 1449–1457, 2015.
- 630 [39] X. He, Y. Peng, Fine-grained Image Classification via Combining Vision and Language, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7332–7340, 2017.
- [40] X. Zhang, F. Zhou, Y. Lin, S. Zhang, Embedding Label Structures for Fine-Grained Feature Representation, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1114–1123, 2016.
- 635 [41] N. Hussein, E. Gavves, A. W. Smeulders, Unified Embedding and Metric Learning for Zero-Exemplar Event Detection, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1096–1105, 2017.
- [42] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778, 2016.
- 640 [43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, *International Journal on Computer Vision* (2015) 1–42.



- [44] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, in: International Conference on Learning Representations (ICLR), 2015.
- [45] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed Representations of Words and Phrases and their Compositionality, in: Neural Information Processing Systems (NIPS), 3111–3119, 2013.
- [46] B. Klein, G. Lev, G. Sadeh, L. Wolf, Associating Neural Word Embeddings With Deep Image Representations Using Fisher Vectors, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4437–4446, 2015.
- [47] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, in: International Conference on Machine Learning (ICML), 448–456, 2015.
- [48] J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3431–3440, 2015.
- [49] S. Xie, Z. Tu, Holistically-Nested Edge Detection, in: IEEE International Conference on Computer Vision (ICCV), 1395–1403, 2015.
- [50] S. Yang, D. Ramanan, Multi-scale Recognition with DAG-CNNs, in: IEEE International Conference on Computer Vision (ICCV), 1215–1223, 2015.
- [51] Y. Liu, Y. Guo, M. S. Lew, On the Exploration of Convolutional Fusion Networks for Visual Recognition, in: International Conference on MultiMedia Modeling (MMM), 277–289, 2017.
- [52] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, T. Mikolov, DeViSE: A Deep Visual-Semantic Embedding Model, in: Neural Information Processing Systems (NIPS), 2121–2129, 2013.
- [53] L. van der Maaten, G. Hinton, Visualizing Data Using t-SNE, *Journal of Machine Learning Research* 9 (2008) 2579–2605.
- [54] N. Pham, R. Pagh, Fast and Scalable Polynomial Kernels via Explicit Feature Maps, in: ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 239–247, 2013.
- [55] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The Caltech-UCSD Birds-200-2011 Dataset, Tech. Rep. CNS-TR-2011-001, 2011.
- [56] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: ECCV, 740–755, 2014.
- [57] C. Rashtchian, P. Young, M. Hodosh, J. Hockenmaier, Collecting Image Annotations Using Amazon’s Mechanical Turk, in: Proceedings of the NAACL HLT Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk, 139–147, 2010.
- [58] M.-E. Nilsback, A. Zisserman, Automated Flower Classification over a Large Number of Classes, in: Indian Conference on Computer Vision, Graphics and Image Processing, 722–729, 2008.
- [59] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, in: ACM International Conference on Multimedia (MM), 675–678, 2014.
- [60] R. Socher, A. Karpathy, Q. Le, C. Manning, A. Ng, Grounded Compositional Semantics for Finding and Describing Images with Sentences, *Transactions of the Association for Computational Linguistics* 2 (2014) 207–218.
- [61] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in: Neural Information Processing Systems (NIPS), 1097–1105, 2012.
- [62] S. Rastegar, M. Soleymani, H. R. Rabiee, S. Mohsen Shojaei, MDL-CW: A Multimodal Deep Learning Framework With Cross Weights, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2601–2609, 2016.
- [63] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, D. Forsyth, Every Picture Tells a Story: Generating Sentences from Images, in: European Conference on Computer Vision (ECCV), 15–29, 2010.
- [64] G. Lev, G. Sadeh, B. Klein, L. Wolf, RNN Fisher Vectors for Action Recognition and Image Annotation, in: European Conference on Computer Vision (ECCV), 833–850, 2016.

- 690 [65] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, S. Carlsson, Factors of Transferability for a Generic ConvNet Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (9) (2016) 1790–1802.
- [66] N. Zhang, J. Donahue, R. Girshick, T. Darrell, Part-based RCNN for Fine-grained Detection, in: *European Conference on Computer Vision (ECCV)*, 834–849, 2014.
- 695 [67] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, Z. Zhang, The Application of Two-Level Attention Models in Deep Convolutional Neural Network for Fine-Grained Image Classification, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 842–850, 2015.
- [68] D. Lin, X. Shen, C. Lu, J. Jia, Deep LAC: Deep Localization, Alignment and Classification for Fine-Grained Recognition, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1666–1674, 2015.
- 700 [69] M. Simon, E. Rodner, Neural Activation Constellations: Unsupervised Part Model Discovery With Convolutional Networks, in: *IEEE International Conference on Computer Vision (ICCV)*, 1143–1151, 2015.
- [70] X. Zhang, H. Xiong, W. Zhou, W. Lin, Q. Tian, Picking Deep Filter Responses for Fine-Grained Image Recognition, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1134–1142, 2016.
- [71] Q. Qian, R. Jin, S. Zhu, Y. Lin, Fine-Grained Visual Categorization via Multi-Stage Metric Learning, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3716–3724, 2015.
- 705 [72] Y. Guo, Y. Liu, S. Lao, E. M. Bakker, L. Bai, M. S. Lew, Bag of Surrogate Parts Feature for Visual Recognition, *IEEE Transactions on Multimedia* .
- [73] L. Xie, J. Wang, W. Lin, B. Zhang, Q. Tian, Towards Reversal-Invariant Image Representation, *International Journal on Computer Vision* 123 (2) (2017) 226–250.
- 710 [74] S. Cai, L. Zhang, W. Zuo, X. Feng, A Probabilistic Collaborative Representation Based Approach for Pattern Classification, in: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2950–2959, 2016.
- [75] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, Z. Zhang, Multiple Granularity Descriptors for Fine-Grained Categorization, in: *IEEE International Conference on Computer Vision (ICCV)*, 2399–2406, 2015.