Studying the Impact of the Full-Network Embedding on Multimodal Pipelines

Armand Vilalta [a,*], Dario Garcia-Gasulla [a], Ferran Parés [a], Eduard Ayguadé [a,b], Jesus Labarta [a,b], and Ulises Cortés [a,b],

[a] *Barcelona Supercomputing Center (BSC), Jordi Girona 1-3, 08034 Barcelona, Spain*
E-mails: armand.vilalta@bsc.es, dario.garcia@bsc.es, ferran.pares@bsc.es
[b] *Universitat Politècnica de Catalunya, Jordi Girona 1-3, 08034 Barcelona, Spain*
E-mail: ia@cs.upc.edu

Abstract. The current state-of-the-art for image annotation and image retrieval tasks is obtained through deep neural networks, which combine an image representation and a text representation into a shared embedding space. In this paper we evaluate the impact of using the Full-Network embedding in this setting, replacing the original image representation in four competitive multimodal embedding generation schemes. Unlike the one-layer image embeddings typically used by most approaches, the Full-Network embedding provides a multi-scale representation of images, which results in richer characterizations. To measure the influence of the Full-Network embedding, we evaluate its performance on three different datasets, and compare the results with the original embedding scheme, and with the rest of the state-of-the-art. Results for image annotation and image retrieval tasks indicate that the Full-Network embedding is consistently superior to the one-layer embedding. These results motivate the integration of the Full-Network embedding on any multimodal embedding generation scheme, something feasible thanks to the flexibility of the approach.

Keywords: Multimodal Embedding, Full-Network Embedding, Caption Retrieval, Image Retrieval, Deep Neural Network

1. Introduction

Image annotation (also known as caption retrieval) is the task of automatically associating an input image with a describing text. Image annotation methods are an emerging technology, enabling semantic image indexing, search applications and visual question answering [1] . The complementary task of associating an input text with a fitting image (known as image retrieval or image search) is also of relevance for the same sort of applications. Furthermore, many methods for caption generation are based on a joint embedding space. In this setting, retrieval is a natural way to asses the quality the joint embedding [2] before moving forward.

State-of-the-art image annotation methods are currently based on deep neural net representations, where an image embedding (*e.g.*, obtained from a convolutional neural network or CNN) and a text embedding (*e.g.*, obtained from a recurrent neural network or RNN) are combined into a unique multimodal embedding space. While several techniques for merging both spaces have been proposed [3–13], little effort has been made in finding the most appropriate image embeddings to be used in that process. In fact, most approaches simply use a one-layer CNN embedding [14, 15] and the only method proposed to increase the quality of the embedding rely on obtaining more data to allow for fine-tuning the CNN in the final stage of training [10]. In this paper we explore the impact of using a Full-Network embedding (FNE) [16] to generate the required image embedding, replacing the one-layer embedding. We do so by integrating the FNE into the multimodal embedding pipeline defined in [3], which is based in the use of a Gated Recurrent Units neural network (GRU) [17] for text encoding and CNN for image encoding. Unlike one-layer embeddings, the

[*] Corresponding author. E-mail: armand.vilalta@bsc.es.

FNE represents features of varying specificity in the context of the visual dataset, while also discretizes the features to regularize the space and alleviate the curse of dimensionality. These particularities result in a richer visual embedding space, which may be more reliably mapped to a common visual-textual embedding space. A specially interesting feature of FNE is that features are encoded using only 3 values with a semantic meaning. It makes FNE closer to linguistic representations based on concepts presence or absence than a regular real-valued embedding.

The generic pipeline defined by Kiros *et al.*[3] had been outperformed in image annotation and image search tasks by methods specifically targeting one of those tasks [6, 18]. However, more recent work by Vendrov *et al.*[9] and Faghri *et al.*[10], based on the same generic pipeline, has outperformed previous methods in both tasks. In this paper we extend our previous work *Full-Network Embedding in a Multimodal Embedding Pipeline* [19] introducing the improvements proposed by Vendrov *et al.*[9] and Faghri *et al.*[10]. We also report improvements in our implementation, which increase the performance of the original method [3] as well. Finally, we exhaustively test the main variations on a leveled playground, obtaining insights on the real impact on performance of each of them. Performance evaluation is done using three publicly available datasets: Flickr8k [20], Flickr30k [21] and MSCOCO [22]. Additionally, some hindrances found on Faghri *et al.*[10] are studied, and a methodology for solving them is proposed which also increases performance.

2. Related work

This paper builds upon the methodology described by Kiros *et al.*[3], which is in turn based on previous works in the area of Neural Machine Translation[23]. In their work, Kiros *et al.*[3] define a vectorized representation of an input text by using GRU RNNs. In this setting, each word in the text is codified into a vector embedding, vectors which are then fed one by one into the GRUs. Once the last word vector has been processed, the activations of the GRUs at the last time step conveys the representation of the whole input text in the multimodal embedding space. In parallel, images are processed through a Convolutional Neural Network (CNN) pre-trained on ImageNet [24], extracting the activations of the last fully connected layer to be used as a representation of the images. To solve the dimensionality matching between both representations (the output of the GRUs and the last fully-connected of the CNN) an affine transformation is applied on the image representation.

Following the same pipeline [3], Vendrov *et al.*[9] proposed an asymmetric order-embedding space. Its main hypothesis is that captions are actually abstractions of the images, such as an hypernym/hyponym relation. This relation is imposed using the order error similarity defined in Eq. (3), instead of cosine similarity in the same contrastive loss formulation used in previous work [3]. Another improvement for the same pipeline was proposed in [10] which instead of taking into account all the contrastive examples focus only in the hardest of them. This approach has also been applied to order embeddings successfully. The present work extends the FNE to these methods.

Also using the ranking loss as methodology keystone, the Embedding Network proposed in [25] introduce novel neighbourhood constraints in the form of additional loss penalties, achieving competitive performance. For the specific problem of image annotation, good results are obtained with the Word2VisualVec (W2VV) model [18]. This approach uses as a multimodal embedding space the same visual space where images are represented, involving a deeper text processing. These methods are very similar to the ones presented in this work thus are good candidates to benefit from same improvements (*e.g.*, FNE).

A different group of methods is based in the Canonical Correlation Analysis (CCA). A first successful approach in this direction is the Fisher Vector (FV) [6]. FV are computed with respect to the parameters of a Gaussian Mixture Model (GMM) and an Hybrid Gaussian-Laplacian Mixture Model (HGLMM). For both images and text, FV are build using deep neural network features; a VGG [26] CNN for images features, and a word2vec [27] for text features. A more recent approach [13], based on CCA methodology, introduce a novel bidirectional neural network architecture that project image and sentence to a maximally correlated space using the Euclidean loss instead of CCA. Since these methods rely on a CNN representation of the image the introduction of the FNE should be straightforward.

Attention-based models have also proved their advantages in this task. Dual Attention Networks (DANs) [11] currently holds the best results on Flickr30K dataset. DANs exploits two attention mechanisms to estimate the similarity between images and sentences by focusing on their shared semantics. In the same line, selective multimodal Long Short-Term Memory network (sm-LSTM) [12] includes a multimodal context-modulated attention scheme at each time-step that can selectively attend to a pair of instances of image and sentence, by predicting pairwise instance-aware saliency maps for image and sentence. Attention-based methods rely on representations of parts of the image (and text) while FNE obtains a compact representation of the whole image at the cost of losing the spatial information. Application of the FNE to those techniques is far from being immediate, requiring important modifications in the FNE schema to be possible.

3. Methods

The multimodal embedding generator pipeline of [3] represents images and textual captions within the same space. The pipeline is composed by two main elements, one which generates image embeddings and another one which generates text embeddings. In this work we replace the original image embedding generator by the FNE, resulting in the architecture shown in Figure 1. Next we describe these components in further detail.

3.1. Full-network Embedding

The FNE generates a vector representation of an input image by processing it through a pre-trained CNN, and extracting the neural activations of all convolutional and fully-connected layers. After the initial feature extraction process, the FNE performs a dimensionality reduction step for convolutional activations, by applying a spatial average pooling on each convolutional filter. After the spatial pooling, every feature (from both convolutional and fully-connected layers) is standardized through the z-values, which are computed over the whole image train set. This standardization process puts the value of the each feature in the context of the dataset. At this point, the meaning of a single feature value is the degree with which the feature value is atypically high (if positive) or atypically low (if negative) in the context of the dataset. Zero marks the typical behavior.

The last step of the FNE is a feature discretization process. The previously standardized embedding is usually of large dimensionality (*e.g.*, 12,416 features for VGG16) which entails problems related with the curse of dimensionality. A common approach to address this issue would be to apply some dimensionality reduction methods (*e.g.*, PCA) [28, 29]. The FNE uses a different approach, reducing expressiveness through the discretization of features, while keeping the dimensionality. Specifically, the FNE discretization maps the feature values to the $\{-1, 0, 1\}$ domain, where -1 indicates an unusually low value (*i.e.*, the feature is significant by its absence for an image in the context of the dataset), 0 indicates that the feature has an average value (*i.e.*, the feature is not significant) and 1 indicates an uncommonly high activation (*i.e.*, the feature is significant by its presence for an image in the context of the dataset). The mapping of standardized values into these three categories is done through the definition of two constant thresholds. The optimal values of these thresholds can be found empirically for a labeled dataset [30]. Instead, we use threshold values shown to perform consistently across several domains [16].

3.2. Multimodal embedding

In our approach, we integrate the FNE with the multimodal embedding pipeline of Kiros *et al.*[3]. To do so we obtain the FNE image representation instead of the output of the last layer of a CNN, as the original model does. The encoder architecture processing the text is used as in the original pipeline, using a GRUs recurrent neural network to encode the sentences. Each word in the sentence is first encoded in a one-hot vector using a dictionary containing all the words in train and validation sets. Next, it is encoded through a trainable linear embedding into a word embedding of lower dimensionality. Finally, the sequence of words (embeddings) in the sentence is fed to a GRU and the final state of the hidden units is the sentence embedding. To combine both embeddings, Kiros *et al.*[3] use an affine transformation on the image representation (in our case, the FNE) analogous to a fully connected neural network layer (with identity activation function). We simplified it by removing the bias term, resulting in a linear transformation as in [9]. This linear transformation is trained simultaneously with the GRUs and the word embedding. The elements of the multimodal pipeline that are tuned during the training phase of the model are shown in orange in Figure 1 (notice the image embedding is not fitted).

In simple terms, the pipeline training procedure consist on the optimization of the pairwise ranking loss between the correct image-caption pair and a random pair. Assuming that a correct pair of elements should

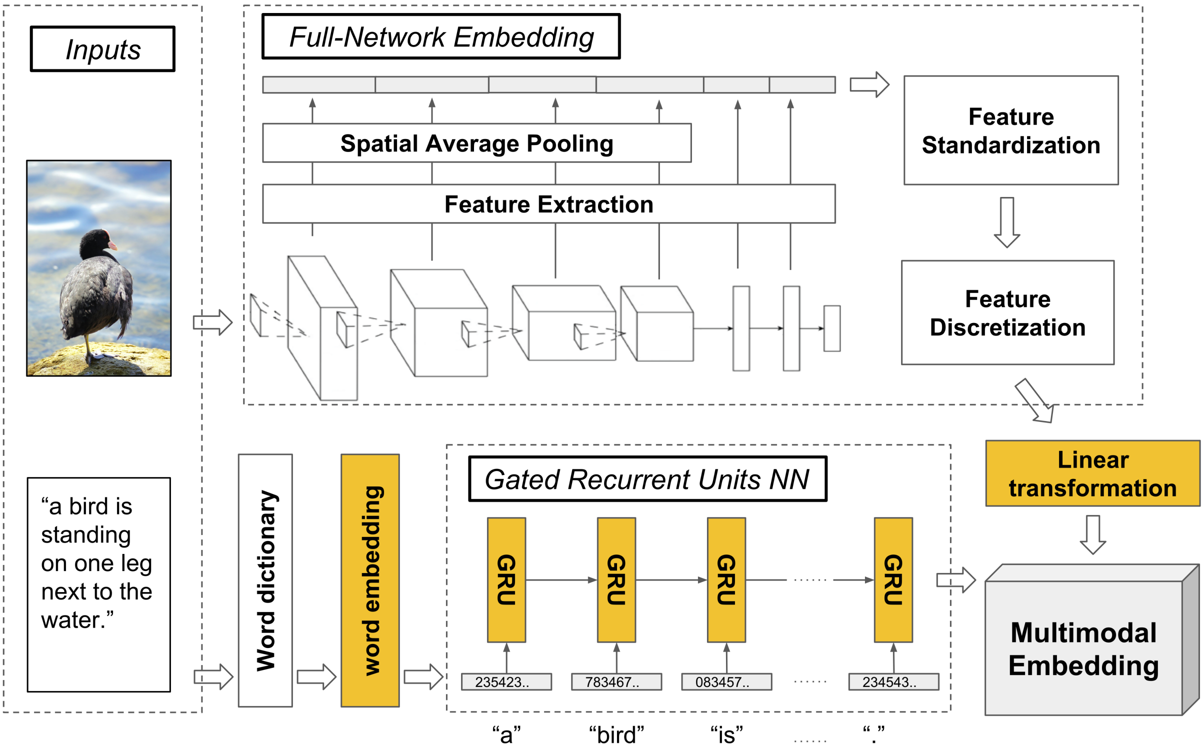

Fig. 1. Overview of the proposed multimodal embedding generation pipeline with the integrated full-network embedding. Elements colored in orange are components modified during the neural network training phase. During testing, only one of the inputs is provided.

be closer in the multimodal space than a random pair, the loss $\mathcal{L}_S$ can be formally defined as follows:

$$\mathcal{L}_S = \sum_{\mathbf{I}} \sum_{k} \max(0, \alpha - \mathcal{S}(\mathbf{i},\mathbf{c}) + \mathcal{S}(\mathbf{i},\mathbf{c}_k)) \quad (1)$$
$$+ \sum_{\mathbf{C}} \sum_{k} \max(0, \alpha - \mathcal{S}(\mathbf{i},\mathbf{c}) + \mathcal{S}(\mathbf{c},\mathbf{i}_k))$$

Where $\mathbf{i}$ is an image vector, $\mathbf{c}$ is its correct caption vector, and $\mathbf{i}_k$ and $\mathbf{c}_k$ are sets of random images and captions respectively. The operator $\mathcal{S}(\bullet,\bullet)$ stands for a similarity metric. This formulation includes a margin term α to avoid pulling the image and caption closer once their distance is smaller than the margin. This makes the optimization focus on pulling together distant pairs instead of improving the ones that are already close.

The similarity metric proposed in [3] is the cosine similarity defined in Eq. (2). In our case, since all embeddings $(\mathbf{c}, \mathbf{i})$ are already normalized to have unit norm, we use the dot product of the vectors as similarity.

$$\mathcal{S}_{COS}(\mathbf{c},\mathbf{i}) = \frac{\mathbf{c} \cdot \mathbf{i}}{\parallel \mathbf{c} \parallel \cdot \parallel \mathbf{i} \parallel} \quad (2)$$

3.3. Multimodal Order Embedding

Using the same general schema, Vendrov *et al.*[9] proposed an asymmetric order embedding space. Their main hypothesis is that captions are actually abstractions of the images, including information such as the hypernym/hyponym relation. In the resulting shared embedding space, an image corresponds to a caption if the value of all components of the image embedding have higher values than the components of the caption embedding ($i_k > c_k \forall i_k \in \mathbf{i}, c_k \in \mathbf{c}$). This relation is imposed during training, using the order error similarity defined in Eq. (3) instead of the cosine similarity in the same contrastive loss formulation defined in Eq. (1).

$$\mathcal{S}_{OE}(\mathbf{c},\mathbf{i}) = - \parallel \max(0, \mathbf{c} - \mathbf{i}) \parallel^2 \quad (3)$$

Notice that since image and caption embeddings are normalized to have unit L2-norm, both lay on an hyper-sphere centered on its coordinate origin, thus a perfect order-embedding will not be achieved unless they are exactly the same vector which is extremely unlikely to happen.

3.4. Maximum error loss

A recent contribution to the field [10] proposes to compute the loss focusing only on the worst contrasting example (*i.e.*, the closest but wrong) instead of taking into account all the examples. To achieve it, Eq. (1) is modified substituting the sum over all contrasting examples for the maximum contrasting example, as shown in Eq. (4).

$$\begin{aligned}\mathcal{L}_M = \sum_{\mathbf{I}} \max_k \{\max(0, \alpha - \mathcal{S}(\mathbf{i},\mathbf{c}) + \mathcal{S}(\mathbf{i},\mathbf{c}_k))\} \quad (4)\\ + \sum_{\mathbf{C}} \max_k \{\max(0, \alpha - \mathcal{S}(\mathbf{i},\mathbf{c}) + \mathcal{S}(\mathbf{c},\mathbf{i}_k))\}\end{aligned}$$

3.5. Curriculum learning

Faghri *et al.*[10] reported problems in training beginning using their proposed Maximum of Hinge Loss (MH). They indicate that a rough form of curriculum learning [31] could be applied, but do not develop or experiment it further as in their preliminary experiments it obtained worse performance than the proposed method. Our experiments showed these difficulties and also an unstable behaviour with respect to hyper-parameter selection which cause the models not to train in a reasonable number of epochs.

We define a sort of curriculum learning approach to combine the benefits of the sum loss $\mathcal{L}_S$ and the max loss $\mathcal{L}_M$. The basic idea is to train using one method until there is no improvement in the validation set. Next take this pre-trained model and train it again using a different method. Several of those training steps can be concatenated.

We propose to train the model using the sum of errors loss $\mathcal{L}_S$, obtain the best performing model and, in a second step, train it using the maximum error loss $\mathcal{L}_M$. Notice that different hyper-parameters may be used in each training phase as long as the dimensionalities of the embeddings are not changed.

We performed preliminary experiments using this methodology to apply a learning rate reduction. We obtained small increases in the performance of some algorithms but we kept these results out of the paper to avoid shadowing more relevant points.

4. Experiments

In this section we evaluate the impact of using the FNE in a multimodal pipeline for both image annotation and image retrieval tasks. We extend our previous work [19] introducing the FNE in different multimodal pipelines. To properly measure the relevance of the FNE, we compare the results obtained with those of the original multimodal pipelines (*i.e.*, without the FNE). Given the discrepancies in the experimental setup of the different contributions, we define baselines by keeping as much of the original setup as possible while leveling the playground (*i.e.*, same training and test sets, same text preprocessing, same source CNN, same data augmentation, *etc.*).

We identify the different combinations of embedding and multimodal pipeline with a notation in the form of EMB-PIPE. EMB denotes the embedding being either FNE (for the full network embedding) or FC7 (for the baselines using the last CNN layer, `fc7`). PIPE denotes the multimodal pipeline used, one of SH, MH, SOE, MOE, PH, POE. The details of each pipeline and the hyper-parameters used in the experiments can be found in Section 4.2.

4.1. Datasets

In our experiments we use three different and publicly available datasets:

The **Flickr8K** dataset [20] contains 8,000 hand-selected images from Flickr, depicting actions and events. Five correct captions are provided for each image. Following the provided splits, 6,000 images are used for train, 1,000 are used for validation and 1,000 more are kept for testing.

The **Flickr30K** dataset [21] is an extension of Flickr8K and includes it. It contains 31,783 photographs of everyday activities, events and scenes. Five correct captions are provided for each image. In our experiments 29,000 images are used for training, 1,014 conform the validation set and 1,000 are kept for test. These splits are the same ones used in [3, 32].

The **MSCOCO** dataset [22] includes images of everyday scenes containing common objects in their natural context. For captioning, 82,783 images and 413,915 captions are available for training, while 40,504 images and 202,520 captions are available for validation. Captions from the test set are not publicly available. Previous contributions consider using a subset of the validation set for validation and the rest for test. In most cases, such subsets are composed by either 1,000 or 5,000 images per set, with their corresponding 5 captions per image. In our experiments we only consider the 1k test set to simplify results presentation. Some previous work extend the training set by adding the images and captions in the original validation set that are not used for validation or test [9, 10]. This split raises the number of training images to 113,287, consequently increasing the performance of algorithms [10]. We did not consider using this extended training set since the effect of the quantity of training data is already seen on the performance obtained for the 3 different datasets (which have different sizes).

4.2. Experimental Setup

We will experiment the impact of the FNE on the methods proposed in [3, 9, 10], and on the curriculum learning methodology proposed in Section 3.5. The methods are named following the convention of [10]. Notice all losses are based on a Hinge Loss:

- Sum of Hinge Loss (**SH**). Uses the sum loss $\mathcal{L}_S$ with cosine similarity s_{COS}.
- Maximum of Hinge Loss (**MH**). Uses the max loss $\mathcal{L}_M$ with cosine similarity s_{COS}.
- Sum of Order Embedding Loss (**SOE**). Uses the sum loss $\mathcal{L}_S$ with order embedding similarity s_{OE}.
- Maximum of Order Embedding Loss (**MOE**). Uses the max loss $\mathcal{L}_M$ with cosine similarity s_{COS}.
- Pre-trained Hinge Loss (**PH**). Use curriculum learning. Pre-train using the sum loss $\mathcal{L}_S$ and fine-tune using the max loss $\mathcal{L}_M$ using always cosine similarity s_{COS}.
- Pre-trained Order Embedding Loss (**POE**). Use curriculum learning. Pre-train using the sum loss $\mathcal{L}_S$ and fine-tune using the max loss $\mathcal{L}_M$ using always order embedding similarity s_{OE}.

The details of the hyper-parameters used in the experiments for each method can be found in Table 1.

4.3. Implementation Details

The devil is in the details. To facilitate the reproducibility and interpretability of our work, we provide in this section all the details regarding our implementation.

4.3.1. Training

During a training epoch all images are presented with one caption chosen randomly from the five captions available. This approach differs from the usual of presenting all 5 captions per image each epoch [3, 19]. If all 5 image-caption pairs are included in the dataset it implies that more than one correct image-caption pairs can be included in the same random batch. Since the method uses all image-caption combinations in the batch as contrastive examples, a real correct pair could be wrongly considered incorrect during loss computation, leading to a noisy labels hindrance. The approach used remove this possibility. On the other hand it is now possible that not all captions are used for training. It is easy to check that the probability of actually not using all of the captions during the whole training is in the order of 10^{-8} for our setups. Practically, this approach implies that to achieve a similar training it requires 5 times the number of epochs. On the other side, it reduces to almost 1/5 the memory requirements.

The models are trained until a maximum number of epochs is reached and the best performing model on the validation set is chosen (*i.e.*, early stopping). In the case of baseline experiments the maximum number of epochs is set to 200 for all methods. In MH experiments on Flicker8k and Flicker30k we raise the maximum number of epochs to 400 as we observed results kept improving after 200 epochs.

On all our experiments (both the FC7 and the FNE) the batch size is of 128 image-caption pairs. Within the same batch, every possible alternative image-caption pair is used as contrasting example (*i.e.*, we sum over 127 contrasting examples or we choose the worst example out of 127). We use gradient clipping with a threshold of 2. We use ADAM [33] as optimization algorithm.

4.3.2. Caption processing

The caption sentences are word-tokenized using the Natural Language Toolkit (NLTK) for Python [34]. We did not remove punctuation marks as in [10, 19] in contrast to [9]. Also in contrast to previous work [3, 19] we do not remove long sentences from the training split. We did not observe a significant impact on performance with this reduction of the text pre-processing.

The choice of the word embedding size and the number of GRUs has been analyzed to obtain a range of suitable parameters to test in the validation set. Previous contributions [3, 9, 10] set the word embedding dimensionality to 300. In our preliminary experiments we found that higher dimensionalities help to obtain

Table 1
Hyper-parameter configuration for the experiments

	Model	SH	SH-bl	MH	MH-bl	SOE	SOE-bl	MOE	MOE-bl	PH	POE
	Loss	sum	sum	max	max	sum	sum	max	max	sum-max[b]	sum-max[b]
	Similarity	cos	cos	cos	cos	order	order	order	order	cos	order
Embed. dim.	f8k	1536	2048	1024	1024	1024	1024	1536	1024	1536	1024
	f30k	1536	2048	1536	1024	1024	1024	1536	1024	1536	1024
	coco	1536	2048	2048	1024	1536	1024	2048	1024	1536	2048
	Word embed. dim.	1024	1000[a]	1024	300	1024	300	1536	300	1024	1024
	Learning rate	0.0002	0.0002[a]	0.0002	0.0002	0.001	0.001	0.001	0.0002	0.0002	0.001 - 0.0001[b]
	Margin	0.2	0.2	0.2	0.2	0.05	0.05	0.05	0.2	0.2	0.05
	Absolute value embed.	✗	✗	✗	✗	✓	✓	✓	✗	✗	✓

[a] For MSCOCO Word embedding dimensionality is 2000 and Learning rate is 0.00025. [b] First training - second training parameters

better results. We also found that very different dimensionalities between the word embedding and the multimodal embedding (*i.e.*, 300D - 2048D) slow down the convergence speed during training. For word embeddings from 1024D to 1536D the performance is good for all methods.

Similarly we found that multimodal embedding dimensionalities (*i.e.*, number of GRU units) between 1024 and 2048 gives good results for all methods. Previous methods use to adopt 1024 as the dimensionality of the multimodal embedding space [9, 10], or even as little as 300 [3].

4.3.3. Image processing

For generating the image embedding we use the classical VGG16 CNN architecture [26] pretrained for ImageNet [24] as source model. This architecture is composed by 16 convolutional layers combined with pooling layers, followed by two fully connected layers and a final softmax output layer. Using only the activations of the last fully connected layer before the softmax (`fc7`) the dimensionality of the image embedding is 4096. When using the FNE, features from different layers are combined in an image embedding space of 12,416 dimensions.

To obtain a better representation of the image, the full network embedding resizes the image to 256x256 pixels and extracts 5 crops of 224x224 pixels (one from each corner and the center). Mirroring these 5 crops horizontally we obtain a total of 10 crops which are processed through the CNN independently. The activations collected from each of these 10 crops are averaged to obtain a single representation of the image before further processing. For the baseline we use the same process before L2-normalization. Although this process is common for data augmentation notice that we are not actually doing data augmentation since the number of training samples does not increase.

4.4. Evaluation metrics

To evaluate the image annotation and image retrieval tasks we use the following metrics:

- **Recall@K** (R@K) is the fraction of images for which a correct caption is ranked within the top-K retrieved results (and vice-versa for sentences). Results are provided for R@1, R@5 and R@10.
- **Median rank** (Med *r*) of the highest ranked ground truth result.

To obtain a comparable performance metric per model, we use the sum of the recalls on both tasks. This has been done before in [19] and in [10], the latter using only R@1 and R@10. We only use the score obtained on the validation set to select the best performing model for early stopping and hyper-parameter selection.

5. Results

Table 2 shows the results of the proposed full network embedding on the Flickr8k dataset, for both image annotation and image retrieval tasks. The top part of the table includes the current state-of-the-art (SotA) results as published. The second part summarize the results published by the original contributions this work is based on. Following parts contain the results produced by us for each of the models defined in Section 4.2. Each of these blocks contain two pairs of results. The first pair corresponds to the results while using a configuration of hyper-parameters as close as possible to the original (*i.e.*, baseline or -bl), while

Table 2

Results obtained for the Flickr8 dataset. R@K is Recall@K (high is good). Med *r* is Median rank (low is good). Best results for each FC7 - FNE comparison are shown in underline. Best results for SotA and our experiments are shown in **bold**

Model		Image Annotation				Image Retrieval			
		R@1	R@5	R@10	Med *r*	R@1	R@5	R@10	Med *r*
FV	[6]	21.2	50.0	64.8	5	**31.0**	**59.3**	**73.7**	**4**
m-CNN	[7]	24.8	53.7	67.1	5	20.3	47.6	61.7	5
Bi-LSTM	[35]	29.3	58.2	69.6	**3**	19.7	47.0	60.6	5
W2VV	[18]	33.6	62.0	**75.3**	**3**	-	-	-	-
2WayNet	[13]	**43.4**	**63.2**	-	-	29.3	49.7	-	-
UVS	[3]	18.0	40.9	55.0	8	12.5	37.0	51.5	10
FC7-SH-bl[a]		21.0	45.7	60.4	7	14.0	35.8	48.6	11
FNE-SH-bl[a]		23.3	50.8	66.8	5	15.0	38.2	51.6	10
FC7-SH		22.4	49.8	62.9	6	16.6	41.2	54.3	8
FNE-SH		25.0	50.8	64.3	5	18.6	44.9	58.0	7
FC7-MH-bl [b]		22.6	48.6	61.9	6	17.7	42.7	54.9	8
FNE-MH-bl [b]		24.2	52.0	65.2	5	19.4	44.3	57.3	7
FC7-MH		23.0	49.0	63.3	6	18.5	43.2	56.1	8
FNE-MH		**27.3**	**56.8**	**69.3**	**4**	**21.2**	**47.1**	**59.7**	**6**
FC7-SOE-bl		20.6	45.4	58.0	7	15.4	38.8	52.7	9
FNE-SOE-bl		21.5	48.5	60.7	6	16.2	40.7	53.8	9
FC7-SOE		21.2	48.1	61.7	6	17.8	43.6	56.5	8
FNE-SOE		24.0	52.4	63.9	5	18.7	44.2	57.7	7
FC7-MOE-bl		22.6	48.2	62.3	6	16.9	41.5	54.2	9
FNE-MOE-bl[b]		0.1	0.3	0.3	2,476	0.1	0.6	1.0	499
FC7-MOE		21.5	46.1	60.0	7	15.6	39.0	51.9	9
FNE-MOE		25.5	55.5	67.8	**4**	18.7	44.4	58.4	7
FC7-PH		22.9	48.8	62.5	6	17.1	41.7	54.6	8
FNE-PH		26.3	55.7	68.5	**4**	20.5	45.8	58.1	7
FC7-POE		21.0	48.3	62.0	6	16.9	41.7	55.3	8
FNE-POE		26.2	53.6	65.8	5	19.7	45.6	58.4	7

[a] Results from [19]. [b] Trained for 400 epochs.

the second pair corresponds to the results while using the best configuration we found for the FNE. Within each pair, the first experiment uses the FC7 embedding and the second uses the FNE, keeping all hyper-parameters unchanged. Best results for each pair are underlined. Tables 4.4 and 4 are analogous for the Flickr30k and MSCOCO datasets. Additional results of the UVS model were made publicly available later on by the original authors [36]. We include these for the MSCOCO dataset, which was not evaluated in the original paper [3].

First, let us consider the effect of all modifications in the pipeline (detailed in Sections 3 and 4.3) compared to our previous work [19]. In the first block of experiments we can compare the results from [19] (hyper-parameters are already optimized for FNE) with the ones obtained in this work for the same model. The modifications are We can see a significant improvement in results obtained using both the FC7 and the FNE image embeddings. Results are now very close to the ones obtained by other methods, dimming the benefits of the proposed variants. These results validate the improvements made in the pipeline.

Now, let us focus on the differences between a model and the same model using the FNE image embedding. This is the most important contribution of this paper, as it introduces the FNE on several multimodal embedding pipelines. We can see through the tables of results that every method on every dataset obtains better results when using the FNE embedding when compared to the FC7. Moreover, even with the original hyper-parameter configuration (sub-optimal for FNE) the FNE obtains better results on all tests. The only exception is FNE-MOE-bl where training problems occur with the original configuration (in Section 6 we analyze this issue). Even in this case, results using

Table 3

Results obtained for the Flickr30 dataset. R@K is Recall@K (high is good). Med *r* is Median rank (low is good). Best results for each FC7 - FNE comparison are shown in underline. Best results for SotA and our experiments are shown in **bold**

Model		Image Annotation				Image Retrieval			
		R@1	R@5	R@10	Med *r*	R@1	R@5	R@10	Med *r*
FV	[6]	25.0	52.7	66.0	5	35.0	62.0	73.8	3
m-CNN	[7]	33.6	64.1	74.9	3	26.2	56.3	69.6	4
Bi-LSTM	[35]	28.1	53.1	64.2	4	19.6	43.8	55.8	7
W2VV	[18]	39.7	67.0	76.7	2	-	-	-	-
sm-LSTM[a]	[12]	42.4	67.5	79.9	2	28.2	57.0	68.4	4
2WayNet	[13]	49.8	67.5	-	-	36.0	55.6	-	-
DAN (VGG)	[11]	41.4	73.5	82.5	2	31.8	61.7	72.5	3
DAN (ResNet)	[11]	**55.0**	**81.8**	**89.0**	**1**	39.4	**69.2**	79.1	**2**
EN	[25]	43.2	71.6	79.8	-	31.7	61.3	72.4	-
UVS	[3]	23.0	50.7	62.9	5	16.8	42.0	56.5	8
VSE++(1C)	[10]	31.9	-	68.0	4	23.1	-	60.7	6
VSE++(ResNet)[b]	[10]	52.9	-	87.2	1	**39.6**	-	**79.5**	**2**
FC7-SH-bl[c]		30.4	58.0	69.5	4	18.9	44.6	57.0	7
FNE-SH-bl[c]		30.4	61.8	73.2	3	22.1	47.6	59.8	6
FC7-SH		32.4	60.9	72.6	3	24.1	51.1	64.1	5
FNE-SH		36.4	64.6	75.7	3	25.5	53.8	65.7	5
FC7-MH-bl [d]		29.5	59.9	70.8	4	23.0	48.9	60.4	6
FNE-MH-bl [d]		34.7	63.1	75.6	3	25.1	52.3	64.7	5
FC7-MH		33.6	59.4	69.3	3	23.6	50.0	61.8	5
FNE-MH		**37.7**	66.6	**78.6**	**2**	27.8	56.0	67.1	**4**
FC7-SOE-bl		31.6	60.0	72.4	3	24.0	52.1	64.1	5
FNE-SOE-bl		33.7	63.8	75.3	3	26.0	55.1	67.7	**4**
FC7-SOE		30.2	59.4	70.4	4	23.8	50.5	62.7	5
FNE-SOE		35.5	63.4	75.3	3	26.8	56.1	67.5	**4**
FC7-MOE-bl		31.1	56.2	67.8	4	20.8	47.1	58.2	7
FNE-MOE-bl		0.1	0.4	0.4	2,461	0.1	0.5	0.9	498
FC7-MOE		31.9	61.3	72.7	3	23.8	50.2	61.5	5
FNE-MOE		35.3	65.0	77.1	3	27.3	55.2	68.0	**4**
FC7-PH		31.8	60.1	73.6	3	24.0	51.8	63.3	5
FNE-PH		36.6	63.9	75.0	3	25.9	54.3	66.2	4
FC7-POE		31.4	60.9	72.3	3	24.5	51.3	63.7	5
FNE-POE		37.2	**67.1**	77.9	**2**	**28.1**	**57.8**	**69.1**	**4**

[a] Single model. [b] CNN fine-tuned. [c] Results from [19]. [d] Trained for 400 epochs.

an appropriate hyper-parameter selection are superior to those of the baseline (FC7-MOE-bl). Considering all experiments on MSCOCO dataset (including baselines), the average increase in recall using the FNE embedding is 3.7%.

Beyond the impact of the FNE, performing a consistent comparison between different multimodal approaches is difficult since different authors make different choices in the settings of their experiments (and sometimes fail to detail them thoroughly). Particularly, large differences arise depending on the data used for training and testing. This is specially significant when experimenting on MSCOCO dataset as we have seen in Section 4.1. Similarly, data augmentation techniques, a common approach in most SotA contributions, can give a boost on performance. In our experiments we did our best to avoid such differences or to explicit them clearly when they are unavoidable. In this context, the results we provide are as comparable as possible. Its important to keep in mind all these considerations, when comparing the results we report with the ones from other publications.

Weighting the results of the family of methods based on [3] with the state-of-the-art we see that its rela-

Table 4

Results obtained for the MSCOCO dataset. R@K is Recall@K (high is good). Med *r* is Median rank (low is good). Best results for each FC7 - FNE comparison are shown in underline. Best results for SotA and our experiments are shown in **bold**

Model		Image Annotation				Image Retrieval			
		R@1	R@5	R@10	Med *r*	R@1	R@5	R@10	Med *r*
FV	[6]	25.1	59.8	76.6	4	39.4	67.9	80.9	2
m-CNN	[7]	42.8	73.1	84.1	2	32.6	68.6	82.8	3
sm-LSTM [a]	[12]	52.4	81.7	90.8	1	38.6	73.4	84.6	2
2WayNet	[13]	55.8	75.2	-	-	39.7	63.3	-	-
EN	[25]	54.9	**84.0**	92.2	-	43.3	**76.4**	87.5	-
UVS [b]	[3]	43.4	75.7	85.8	2	31.0	66.7	79.9	3
Order [c]	[9]	46.7	-	88.9	2	37.9	-	85.9	2
VSE++(1C)	[10]	43.6	-	85.8	2	33.7	-	81.0	3
VSE++(ResNet)[c,d]	[10]	**64.6**	-	**95.7**	**1**	**52.0**	-	**92.0**	**1**
Order++[c]	[10]	53.0	-	91.9	1	42.3	-	88.1	2
FC7-SH-bl[e]		41.2	72.8	85.1	2	26.2	58.6	73.9	4
FNE-SH-bl[e]		47.3	76.8	85.8	2	31.4	65.4	78.7	3
FC7-SH		44.0	77.0	86.0	2	33.6	68.8	81.1	3
FNE-SH		**50.6**	80.0	88.4	**1**	36.7	71.3	82.7	**2**
FC7-MH-bl		43.8	74.7	84.5	2	32.8	67.5	80.5	3
FNE-MH-bl		49.6	78.9	89.5	2	37.5	72.1	83.6	**2**
FC7-MH		44.6	75.8	85.7	2	34.1	68.2	80.7	3
FNE-MH		50.2	80.5	**90.5**	**1**	37.2	71.9	83.0	2
FC7-SOE-bl		41.5	74.4	86.0	2	33.8	69.0	82.6	3
FNE-SOE-bl		47.1	78.5	89.6	2	36.8	71.6	84.2	**2**
FC7-SOE		44.3	74.8	84.4	2	34.9	69.2	81.9	3
FNE-SOE		46.7	79.8	88.9	2	36.4	72.8	84.7	**2**
FC7-MOE-bl		40.7	75.3	85.9	2	32.2	66.4	78.3	3
FNE-MOE-bl		0.1	0.3	0.4	2,472	0.1	0.5	0.9	499
FC7-MOE		43.9	75.4	84.9	2	34.2	68.0	81.2	3
FNE-MOE		47.1	79.6	88.3	2	36.6	71.7	83.3	**2**
FC7-PH		45.3	75.0	85.5	2	33.8	68.4	81.0	3
FNE-PH		**50.6**	80.0	88.4	**1**	36.7	71.3	82.7	**2**
FC7-POE		45.6	75.9	86.6	2	35.2	69.7	83.1	**2**
FNE-POE		48.2	**81.5**	89.7	2	**38.8**	**73.5**	**85.0**	**2**

[a] Single model. [b] Results provided on [36]. [c] Extra training data from validation set.
[d] CNN fine-tuned. [e] Results from [19].

tive performance increases with dataset size (larger datasets lead to more competitive performances of these methods). Since the methods tested are more data-driven (*i.e.*, fewer assumptions are made apriori), it is to be expected that they can benefit more from the increase of available data. These results are congruent with the ones in [10] where the experiments using more data obtain state-of-the-art results.

Considering the methods tested in our consistent experimental setup, we see that FNE-MH tend to obtain the best results on image annotation while FNE-POE is usually superior in image retrieval tasks. With these results we can not consider one method clearly superior to the other except in the smallest Flicker8K dataset, where FNE-MH is clearly superior. Nevertheless the differences between the best versions of each method remain quite small. In experiments on MSCOCO, the recall gap between the best and the worst method (for each task separately) is on average 2.1%.

The proposed methodology of curriculum learning increases the already good performance of the original FC7-MOE [10] and the FNE-MOE 1.7% on average on MSCOCO. On the other hand, on methods based on the cosine similarity $\mathcal{S}_{COS}$, the second training step (using max loss $\mathcal{L}_M$) add very little improvement on the sum loss $\mathcal{L}_S$ results. Final results of FC7-PH and

Table 5

Hyper-parameter configuration and results for the experiments on MOE training behaviour. Success indicates the number of times that experiment succeeded in starting training (*i.e.*, score > 10) over total repetitions

Model	L.rate	Margin	Abs. val.	Success
FC7-MOE-bl	0.0002	0.2	✗	5/5
FC7-MOE-bl-abs	0.0002	0.2	✓	0/5
FC7-MOE-abs	0.0001	0.05	✓	0/5
FNE-MOE-bl	0.0002	0.2	✗	0/5
FNE-MOE-bl-abs	0.0002	0.2	✓	0/5
FNE-MOE-abs	0.0001	0.05	✓	4/5

FNE-PH are in general inferior to those achieved by single training using max loss $\mathcal{L}_M$ (FC7-MH, FNE-MH).

6. Experiments on MOE training behaviour

When training models using the maximum order embedding (MOE and MOE-bl), we observed instability issues. For some configurations of the hyper-parameters, the model does not start learning, even after extending the number of epochs significantly. To obtain some insights on that behaviour we trained the exactly same model 5 times with different random initializations. The configurations tested are shown in Table 5. The combinations of learning rate, margin and absolute value are taken from the original works of [9, 10].

The rest of the hyper-parameters are kept the same for all experiments. The dimensionality of the word embedding is 300 and multimodal embedding has 1024 dimensions. The maximum number of epochs is 200. We run all the tests on Flickr8K to minimize computational cost, although we observed this behaviour in Flickr30K and MSCOCO too.

To evaluate these experiments we count the number of times the algorithm succeeded in starting training out of 5 tests. We consider it does not train if validation and test scores are below 10 (regular scores are higher than 200). The results obtained are shown in Table 5.

Results, quite surprisingly, do not point to a single variable as the cause of the problem. For the FC7 embedding it did not train when absolute value was used, independently of the learning rate and margin. The experiment with the same configuration that worked well with FC7 does not train with FNE. On the other hand, the original configuration from [9] (but using max loss) successfully trained on FNE embedding, but this behaviour is not fully robust since it failed once.

These experiments show that the instability training does not depend on the embedding mainly, but on hyper-parameter selection and parameter initialization. While these experiments help to shed light into the problem, further work is required to truly understand the cause.

The proposed curriculum learning methodology (see Section 3.5) effectively solve this problem as it initializes the network using the more robust sum loss. None of the experiments we did using the proposed curriculum learning methodology for different hyper-parameters configurations failed to start training.

7. Conclusions

For the multimodal pipeline of Kiros *et al.*[3] and other methods based on it [9, 10], using the Full-Network image embedding results in consistently higher performances than using a one-layer image embedding. These results suggest that the visual representation provided by the FNE is superior to the current standard for the construction of most multimodal embeddings. Indeed, the impact FNE has on performance is significantly superior to the improvement resultant of applying the main contributions from [9] and [10].

When compared to the current state-of-the-art, the results obtained by the studied variants using FNE are below results reported through other methods. This difference is often the result of using a larger amount of data for training. Indeed, results from [10] indicate that models based on the pipeline of [3] can obtain state-of-the-art results when using enough data.

Another issue we tackled was the instability of MOE models. Depending on the random initialization of the weights, the exactly same model may start training or not. The proposed curriculum learning method of pre-training using a sum of losses effectively alleviate this hindering while increasing performance significantly.

Finally, let us remark that the FNE is straightforward compatible with most multimodal pipelines based on CNN embeddings. If the boost in performance we demonstrate here for the variants proposed by [3, 9, 10] translates to other methods, the introduction of the FNE on methods which currently hold the state-of-the-art results would likely define a new *best method*. The integration of FNE with methods not so clearly compatible (*e.g.*, DAN) remains as future work.

Since the FNE is compatible with most multimodal pipelines based on CNN embeddings it should be straightforward to introduce it. If the boost in performance obtained by the FNE on the methods studied [3, 9, 10] translates to other methods, such combination would be likely to define new state-of-the-art results on both tasks.

References

[1] M. Malinowski, M. Rohrbach and M. Fritz, Ask your neurons: A neural-based approach to answering questions about images, in: *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE Computer Society, 2015, pp. 1–9.

[2] M. Hodosh, P. Young and J. Hockenmaier, Framing image description as a ranking task: Data, models and evaluation metrics, *Journal of Artificial Intelligence Research* **47** (2013), 853–899.

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

[4] R. Kiros, R. Salakhutdinov and R. Zemel, Multimodal neural language models, in: *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 595–603.

[5] A. Karpathy, A. Joulin and F.F.F. Li, Deep fragment embeddings for bidirectional image sentence mapping, in: *Advances in neural information processing systems*, 2014, pp. 1889–1897.

[6] B. Klein, G. Lev, G. Sadeh and L. Wolf, Associating neural word embeddings with deep image representations using fisher vectors, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4437–4446.

[7] L. Ma, Z. Lu, L. Shang and H. Li, Multimodal convolutional neural networks for matching image and sentence, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2623–2631.

[8] Q. Sun, S. Lee and D. Batra, Bidirectional Beam Search: Forward-Backward Inference in Neural Sequence Models for Fill-in-the-Blank Image Captioning, *arXiv preprint arXiv:1705.08759* (2017).

[9] I. Vendrov, R. Kiros, S. Fidler and R. Urtasun, Order-embeddings of images and language, *arXiv preprint arXiv:1511.06361* (2015).

[10] F. Faghri, D.J. Fleet, J.R. Kiros and S. Fidler, VSE++: Improving Visual-Semantic Embeddings with Hard Negatives, *arXiv preprint arXiv:1707.05612* (2017).

[11] H. Nam, J.-W. Ha and J. Kim, Dual Attention Networks for Multimodal Reasoning and Matching, *arXiv preprint arXiv:1611.00471* (2016).

[12] Y. Huang, W. Wang and L. Wang, Instance-aware Image and Sentence Matching with Selective Multimodal LSTM, *arXiv preprint arXiv:1611.05588* (2016).

[13] A. Eisenschtat and L. Wolf, Linking Image and Text with 2-Way Nets, *arXiv preprint arXiv:1608.07973* (2016).

[14] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng and T. Darrell, DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition., in: *Icml*, Vol. 32, 2014, pp. 647–655.

[15] A. Sharif Razavian, H. Azizpour, J. Sullivan and S. Carlsson, CNN features off-the-shelf: an astounding baseline for recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 806–813.

[16] D. Garcia-Gasulla, A. Vilalta, F. Parés, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés and T. Suzumura, An Out-of-the-box Full-network Embedding for Convolutional Neural Networks, *arXiv preprint arXiv:1705.07706* (2017).

[17] K. Cho, B. Van Merriënboer, D. Bahdanau and Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, *arXiv preprint arXiv:1409.1259* (2014).

[18] J. Dong, X. Li and C.G. Snoek, Word2VisualVec: Image and video to sentence matching by visual feature prediction, *CoRR, abs/1604.06838* (2016).

[19] A. Vilalta, D. Garcia-Gasulla, F. Parés, E. Ayguadé, J. Labarta, U. Cortés and T. Suzumura, Full-network embedding in a multimodal embedding pipeline, *arXiv preprint arXiv:1707.09872* (2017). http://www.aclweb.org/anthology/W17-7304.

[20] C. Rashtchian, P. Young, M. Hodosh and J. Hockenmaier, Collecting image annotations using Amazon's Mechanical Turk, in: *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, Association for Computational Linguistics, 2010, pp. 139–147.

[21] P. Young, A. Lai, M. Hodosh and J. Hockenmaier, From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions, *Transactions of the Association for Computational Linguistics* **2** (2014), 67–78.

[22] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C.L. Zitnick and P. Dollar, Microsoft coco: Common objects in context, *arXiv preprint arXiv:1405.0312* (2014).

[23] I. Sutskever, O. Vinyals and Q.V. Le, Sequence to sequence learning with neural networks, in: *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., Imagenet large scale visual recognition challenge, *International Journal of Computer Vision* **115**(3) (2015), 211–252.

[25] L. Wang, Y. Li, J. Huang and S. Lazebnik, Learning two-branch neural networks for image-text matching tasks, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018).

[26] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* (2014).

[27] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, in: *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[28] A. Mousavian and J. Kosecka, Deep convolutional features for image based retrieval and scene categorization, *arXiv preprint arXiv:1509.06033* (2015).

[29] H. Azizpour, A.S. Razavian, J. Sullivan, A. Maki and S. Carlsson, Factors of transferability for a generic convnet representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(9) (2016), 1790–1802.

[30] D. Garcia-Gasulla, F. Parés, A. Vilalta, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés and T. Suzumura, On the Behavior

of Convolutional Nets for Feature Extraction, *arXiv preprint arXiv:1703.01127* (2017).

[31] Y. Bengio, J. Louradour, R. Collobert and J. Weston, Curriculum learning, in: *Proceedings of the 26th annual international conference on machine learning*, ACM, 2009, pp. 41–48.

[32] A. Karpathy and L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3128–3137.

[33] D. Kingma and J. Ba, Adam: A method for stochastic optimization, *arXiv preprint arXiv:1412.6980* (2014).

[34] S. Bird, NLTK: the natural language toolkit, in: *Proceedings of the COLING/ACL on Interactive presentation sessions*, Association for Computational Linguistics, 2006, pp. 69–72.

[35] C. Wang, H. Yang, C. Bartz and C. Meinel, Image captioning with deep bidirectional LSTMs, in: *Proceedings of the 2016 ACM on Multimedia Conference*, ACM, 2016, pp. 988–997.

[36] R. Kiros, visual-semantic-embedding. https://github.com/ryankiros/visual-semantic-embedding.

[37] H. Hanke and D. Knees, A phase-field damage model based on evolving microstructure, *Asymptotic Analysis* **101** (2017), 149–180.

[38] E. Lefever, A hybrid approach to domain-independent taxonomy learning, *Applied Ontology* **11**(3) (2016), 255–278.

[39] P.S. Meltzer, A. Kallioniemi and J.M. Trent, Chromosome alterations in human solid tumors, in: *The Genetic Basis of Human Cancer*, B. Vogelstein and K.W. Kinzler, eds, McGraw-Hill, New York, 2002, pp. 93–113.

[40] P.R. Murray, K.S. Rosenthal, G.S. Kobayashi and M.A. Pfaller, *Medical Microbiology*, 4th edn, Mosby, St. Louis, 2002.

[41] E. Wilson, Active vibration analysis of thin-walled beams, PhD thesis, University of Virginia, 1991.

[42] J. Pennington, R. Socher and C.D. Manning, GloVe: Global Vectors for Word Representation, in: *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. http://www.aclweb.org/anthology/D14-1162.

[43] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki and S. Carlsson, From generic to specific deep representations for visual recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 36–45.