DEIM Forum 2017 G6-1

Learning Subjective Adjectives from Images by Stacked Convolutional Auto-Encoders

Bei LIU[†], Makoto P. KATO[†], and Katsumi TANAKA[†]

† Department of Social Informatics, Kyoto University
Yoshida-honmach, Sakyo-ku, Kyoto
606–8501 Japan
E-mail: †{liubei,kato,tanaka}@dl.kuis.kyoto-u.ac.jp

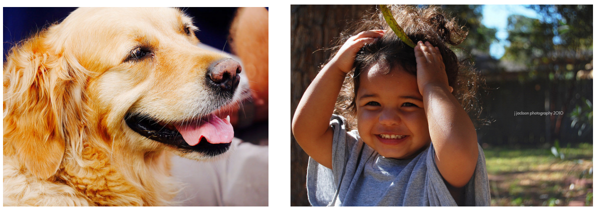
Abstract In this research, we tackle a problem of searching images based on "subjective adjective noun pair" queries and ranking images considering both "noun" and "subjective adjective". Conventional researches that focus on analyzing "subjective adjective" (e.g. sentiment) and "noun" (e.g. object) from images rely on supervised learning method which requires a large number of training data and the reliability of data will influence learning performance a lot. Moreover, in the context of image search, it is unrealistic to include all possible "subjective adjective noun pair" queries and give labels to their images. For this reason, we propose to find truly relevant images of "subjective adjective noun pair" queries by learning discriminative features from pair-wise difference between images with unsupervised deep learning method (pair-wise stacked convolutional auto-encoders) and then rank images based on their relevance to "subjective adjective" and "noun". We conduct experiments with flickr images to show the effectiveness of our approach.

Key words image search, unsupervised, deep learning

1. Introduction

Subjective adjectives refer to adjectives that express opinions and evaluations in natural language [16]. Recently, analysis of "subjective adjective" (e.g. sentiment) from visual contents has attracted considerable attentions [3] [4] [17]. Compared with object detection, scene categorization, textual analysis, or pure visual attribute analysis, subjective adjective analysis is more subjective and holistic, and it is related to broader and more abstract image analysis. The focus of this paper is image retrieval and in particular images of "subjective adjective noun pair" queries.

In contrast to ANP classification problems [3] [4] [12] which try to classify an image into adjective noun pair concepts, our purpose is to estimate relevance of an image to a given "subjective adjective noun pair" query. In order to achieve this goal, we need to compute the relevance between images and the query which is difficult to define and construct a mapping from completely visual contents to subjective adjectives and nouns. Narihira et al. [12] succeed in building a visual sentiment ontology from visual data and respects visual correlations along adjective and noun semantics with a factorized CNN model. However, their method depends on supervised learning which requires a large number of labeled images and the reliability of data will influence learning performance. In the context of image search, it is unrealistic to include all possible "subjective adjective noun pair" queries and give labels to their images. Intuitively, the discriminative features that are critical to relevance estimation vary on difference nouns in the context of the same subjective adjective. For example, in Figure 1, opening mouth and hanging tongue might a sign of "happy dog" while the rising radian of the mouth is enough to indicate a "happy" for the noun "girl".



(a) relevant image of "happy dog"[注1] (b) relevant image of "happy girl"[注2]

Figure 1 Two relevant images of the same "subjective adjective" but different "nouns".

Given a query, current image search engines defines an image's relevance to the query utilizing the image's contextual information and measure the image's similarity with other

[注1]：https://www.flickr.com/photos/151892120@N03/32723517222

[注2]：https://www.flickr.com/photos/12287146@N04/5250858198 result images (e.g. VisualRank). VisualRank [7] gives high weight to images that is most similar to other images. This approach might be enough for searching images of objects, since features of objects are much easier to be captured by existing feature extracting algorithm. However, with the query including "subjective adjective", the problem is not that simple. Firstly, many images are not the exact reflection of their surrounding texts that include "subjective adjective" and "noun", and this results in many irrelevant images, especially irrelevant to the "subjective adjective" query. Secondly, discriminative features that are responsible for the "subjective adjective" query is not easy to extract with existing feature extraction methods, since the way to compute those features differs from query to query. For an instance, color is an effective feature for "spicy chicken" while "size proportion is better to identify "soft pancake".

Although it is not guaranteed that truly relevant images account for the most among result images of "subjective adjective noun pair" query, we assume that there are more truly relevant images in the result images than in images of only "noun" query. Thus, we propose to compare two result image sets with such assumption: discriminative features that help add the relevance to "subjective adjective" are similar for one object ("noun") in certain dimensions. Note that we focus on finding differences that are important for "subjective adjective" since we consider it as a more difficult problem than "noun", although our approach can also apply to noun's difference.

Suppose we have two image sets, one of "subjective adjective noun" and one of "noun". They are two candidate images of relevant images and irrelevant images. Our purpose is first to find truly relevant images and truly irrelevant images and then using these images to extract useful features that can represent the input "subjective adjective noun" query. To better find discriminative features that are especially effective to "subjective adjective" of the "noun", we compare these two image sets in a pairwise way. Without any labels that indicate whether differences of an image pair contains discriminative features, we propose to use unsupervised method to find truly relevant and truly irrelevant images first and then estimate relevance of an image to a "subjective adjective noun" query.

Performance of feature representation is limited by the representation power of handcrafted features if we extract features like SIFT, HOG first and then learn image similarity, and neural network has shown its advantages in many researches [8] [10] [1]. In this research, we proposed a stacked pairwise convolutional auto-encoder by borrowing the idea of convolutional auto-encoder (CAE) from Masci et al. [11]. Ideally, our approach will be able to learn representative features that can present discriminative difference between two image sets. These discriminative features are then used to estimate images' relevance to the "subjective adjective noun" query.

Two contributions of this paper are briefly described:

（1） We proposed to learn truly relevant training dataset of "subjective adjective noun" query by comparing images of to pseudo-relevant image set and pseudo-irrelevant image set,

（2） We proposed to find discriminative features to represent differences of images by unsupervised deep learning method (stacked convolutional auto-encoders)

We conduct experiments with images from flickr to evaluate the effectiveness of our approach.

2. Preminaries

2.1 Auto-Encoder

Encoder-decoder paradigm is used in many unsupervised feature learning methods, such as Predictability Minimization Layers [14], Restricted Boltzmann Machines (RBMs) [5] and auto-encoders [6].

Here we briefly specify the auto-encoder (AE) framework and its terminology.

*Encoder:*a deterministic function f_θ mapps an input vector $\mathbf{x} \in \mathbb{R}^d$ into hidden representation $\mathbf{y} \in \mathbb{R}^{d'}$: $\mathbf{y} = f_\theta(\mathbf{x}) = \sigma(\mathbf{W}\mathbf{x}+\mathbf{b})$ with parameters $\theta = \{\mathbf{W}, \mathbf{b}\}$, where $\mathbf{W}$ is a $d' \times d$ weight matrix and b is an offset vector of dimensionality d'.

*Decoder:*the resulting hidden representation $\mathbf{y}$ is then mapped back to a reconstructed d-dimensional vector $\mathbf{z}$: $\mathbf{z} = f_{\theta'}(\mathbf{y}) = \sigma(\mathbf{W}'\mathbf{y} + \mathbf{b}')$ with $\theta' = \{\mathbf{W}', \mathbf{b}'\}$. The two parameter sets are usually constrained to have a tied weights between $\mathbf{W}$ and $\mathbf{W}'$: $\mathbf{W}' = \mathbf{W}^{\mathbf{T}}$.

The parameters are optimized to to minimize an appropriate cost function (e.g. measure square error) over the training set.

2.2 Denoising Auto-Encoder

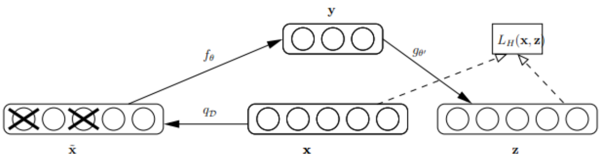


Figure 2 The denoising auto-encoder architecture.

In order to make the trained representation robust to partial destruction of the input, Vincent et al. [15] proposed denoising auto-encoders by introducing a corrupted version of the input. As showed in Figure 2, during the encoder and decoder process, the representation $\mathbf{y}$ is trained on the corrupted input $\tilde{\mathbf{x}}$ instead of original input $\mathbf{x}$, while the cost function is measured between the reconstructed $\mathbf{z}$ and the uncorrupted input $\mathbf{x}$.

2.3 Convolutional Auto-Encoder

To deal with 2D image structure with auto-encoder and reduce redundancy in the parameters brought by global features, convolutional auto-encoder (CAE) is proposed [11]. The weights are shared among all locations in one feature map of a channel and the reconstruction is a linear combination of basic image patches based on the latent code.

For the input x of k-th feature map ($0 < k \leqslant H$, H is the number of latent feature maps), the representation is computed as $\mathrm{y}^{\mathrm{k}} = \sigma(\mathrm{x} * \mathrm{W}^{\mathrm{k}} + \mathrm{b}^{\mathrm{k}})$. Here σ is an activation function and $*$ denotes the 2D convolution. The bias b^{k} is broadcasted to the whole map. The reconstruction is obtained with: $\mathrm{z} = \sigma(\sum_{k \in H} \mathrm{y}^{\mathrm{k}} * \tilde{\mathrm{W}}^{\mathrm{k}} + c)$. $\tilde{\mathrm{W}}^{\mathrm{k}}$ denotes the flip operation over both dimensions of the weights.

Mean squared error (MSE) between the input x and reconstructed z is used to measure the cost function that is be minimized. As in the standard neural networks, the backpropagation algorithm is applied to compute the gradient of the cost function with respect to the parameters. A max-pooling layer is used to obtain translation-invariant representation.

2.4 Stacked Auto-Encoder

Deep networks can be trained by building several auto-encoders in a layer-wise way [2]. The representation of the n-th layer is used as the input for the next (n + 1)-th layer and the (n + 1)-th layer is trained after the n-th has been trained. This pair-wise greedy procedure has shown significantly better generalization on a number of tasks [9].

3. Approach

The final goal of this research is to estimate the relevance of an image to a given "subjective adjective noun" query. An image is relevant to a "subjective adjective noun" query when the content of the image includes exactly the "noun" (e.g. sky, cat, people) in the query and we can feel the quality of the "subjective adjective" (e.g. blue, cute, happy) from the image as well.

Intuitively, the problem of measuring the relevance of an image to an object ("noun") is similar to object recognition problem. However, when object and subjective adjective are combined to be measured, the problem becomes complicated. As we have explained in the first section, features that make an image relevant to a subjective adjective and noun will depends on both the adjective and the noun. Traditional training ways that try to learn useful features with supervised methods require a large number of images labeled with ground truth subjective adjectives and nouns. However this is unrealistic for image searching problems. Thus, we propose to apply unsupervised approach to learn truly relevant training dataset first, and then use the dataset to estimate relevance of an image to the query.

Most search engines return images based on the relevance between the query and image's contextual contents while not all images are exactly corresponding to their surrounding contents. As a result, it is not guaranteed that most resulting images are relevant images in consider of a "subjective adjective noun" query, and we call these result images as **pseudo-relevant images**. We can also find images from existing searching engines with the query that includes the "noun" keyword and excludes the "subjective adjective" keyword, and these result images are denoted as **pseudo-irrelevant images**. By applying unsupervised learning method to pseudo-relevant images, such as traditional stacked convolutional auto-encoders, it is able to learn representative feature for both "subjective adjective" and "noun". However, features of "noun" are usually more significant than features of "subjective adjective". As a result, we propose to compare pseudo-relevant images with pseudo-irrelevant images to better learn discriminative features for "subjective adjective" of a certain "noun". Our assumption is that:

[Assumption 1] Discriminative features that help add the quality of "subjective adjective" are similar for one object ("noun") in certain dimensions.

With this assumption, we can know that differences that represent discriminative features are similar while differences of other features are not similar. As a result, we can learn the discriminative features of image pairs with some unsupervised method.

We use $P = \{p1, p2, p3...\}, |P| = m$ to denote top m search result images of "subjective adjective noun" query and $Q = \{q1, q2, q3...\}, |Q| = m$ to denote top m search result images of "noun" query. Here P denotes pseudo-relevant image set and Q denotes pseudo-irrelevant image set.

The main approach consists of four parts:

(1) Make image pairs from pseudo-relevant image set and pseudo-irrelevant image set,

(2) Learn discriminative feature to represent differences between truly relevant images and truly irrelevant images from pseudo-relevant image set and pseudo-irrelevant image set,

(3) Use the learnt discriminative features to cluster image pairs and find images pairs that include truly relevant image and truly irrelevant image,

(4) Utilize the discriminative features and truly relevant images to learn a function that can measure the relevance of an image to the "subjective adjective noun" query.

In this paper, we will have a detailed explanation of how we make image pairs, learn discriminative features of subjective adjective from images with pair-wise stacked convo-

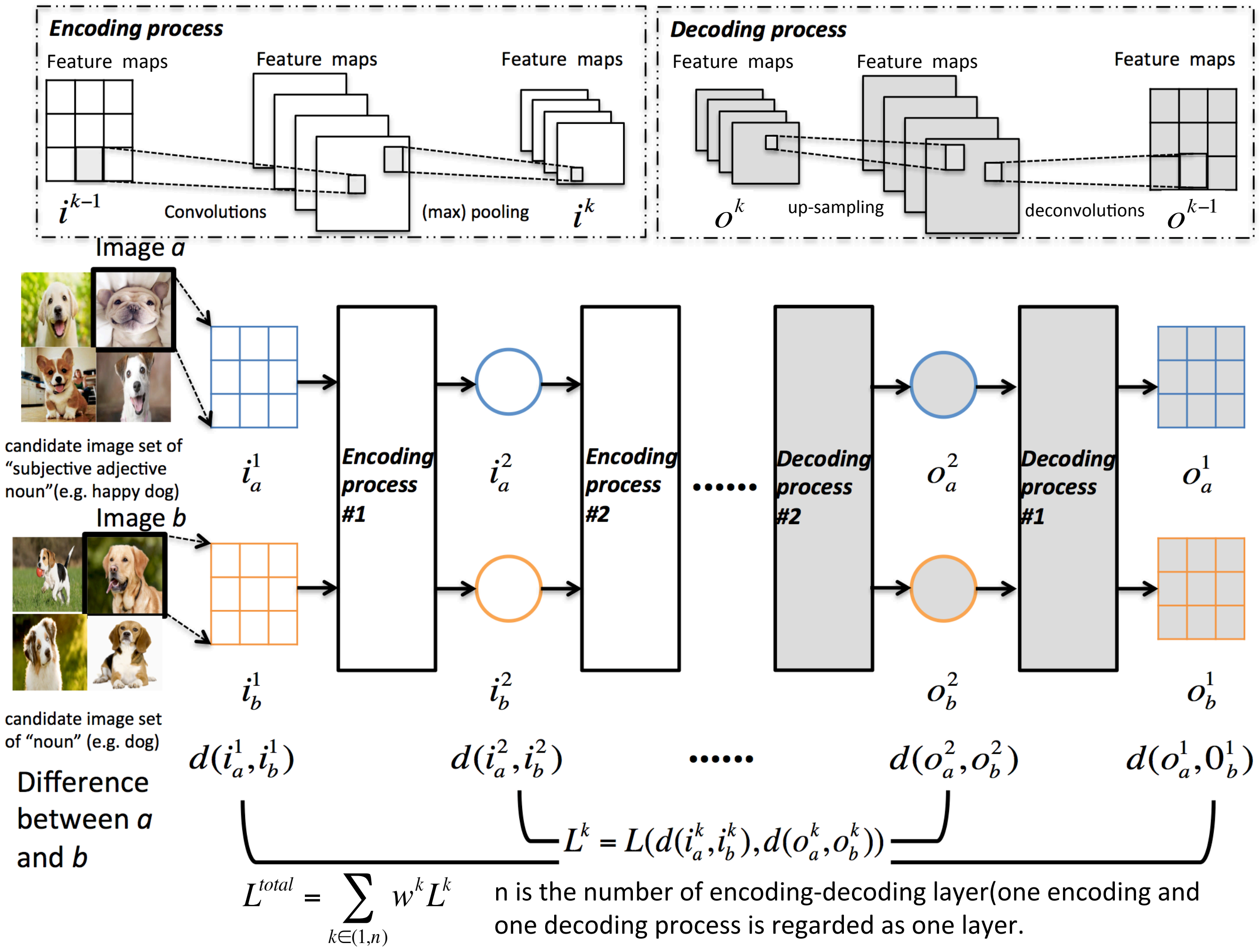


Figure 3 The framework of pair-wise stacked convolutional auto-encoders architecture.

lutional auto-encoders, and then learn truly relevant images with discriminative features.

3. 1 Image Pair Construction

In order to decrease the side effects of many noisy differences between two images, we first conduct a simple image pair selection from the pseudo-relevant image set and the pseudo-irrelevant image set. Since the discriminative features we aim to find are more about subjective adjectives, objects play a less importance in the difference between a truly relevant image and a truly irrelevant image.

We utilize state-of-art object detection networks faster R-CNN [13] to detect objects in images of two dataset and construct our image pairs to include two images with similar objects in each image set.

3. 2 Pair-Wise Stacked Convolutional Auto-Encoders

As we has explained in the above section, we can apply unsupervised learning method to learn discriminative features that represent differences between truly relevant images and truly irrelevant images. To better learn discriminative features for "subjective adjective" of a "noun", we modify the architecture of stacked convolutional auto-encoders to make it suitable to learn representative differences between pseudo-relevant image set and pseudo-irrelevant image set. Figure 3 shows the architecture of our pair-wise stacked convolutional auto-encoders.

Suppose we have one image pair $(a, b), a \in P, b \in Q$. The encoding and decoding process is similar to the convolutional auto-encoder explained in Section 2. 3. Both images a and b are passed through the network. The whole network consists of two main parts, encoding part in the first half and decoding part in the second half. The encoding part includes several encoding process. And the decoding part has the same number of decoding processes with each one corresponding to the encoding process respectively. As we can see from Figure 3, suppose we have three processes in the encoding part: *Encoding process # 1*, *Encoding process #2*, and *Encoding process #3*. The output of lower process serves as the input of next process. In the decoding part, we have three decoding processes in a stacked way. In other words, the first decoding process is corresponding to the most upper encoding process and the last decoding process is corresponding to the first encoding process.

The upper part of Figure 3 explains the detailed workflow of each encoding process and decoding process. Each encoder consist of a convolutional layer to map the input to several feature maps with different kernel (convolutional matrix) and a max-pooling layer for spatial down-sampling. The decoder includes an up-sampling layer and a deconvolutional layer. In traditional auto-encoder systems, the output of the decoding process is compared with the input of the encoding process and the training process is repeated to make them as similar as possible. As a result, the representative features are learnt to reconstruct the input image as well as possible.

In our pair-wise auto-encoder, we are supposed to find the representative differences between sets of image pairs. Thus, instead of computing cost function with reconstructed output and original input, we compare differences of the reconstructed output with differences of original input. As we can see from Figure 3, for two images a and b in the image pair, the input of the k-th encoding process are denoted as i_a^k and i_a^k respectively. We use $d(i_a^k, i_b^k)$ to define the difference of these two images' input before passing them into the k-th encoding process through our network:

$$d(i_a^k, i_b^k) = i_a^k - i_b^k.$$

Similarly, we have o_a^k and o_b^k to represent the feature representation (reconstructed output) after k-th decoding process for image a and image b through the network, and their difference are denoted as $d(o_a^k, o_b^k)$. We then define the squared-error loss between the two differences in k-th encoding process and k-th decoding process:

$$L^k(a, b) = MSE(d(i_a^k, i_b^k), d(o_a^k, o_b^k)).$$

And the total loss is weighted sum of mean square errors in all corresponding encoding-decoding processes:

$$L^{total} = \sum_{k \in (1,n)} w^k L^k,$$

where w^k are weighting factors of the contribution of each process to the total loss. The backpropagation algorithm is applied to compute the gradient of the error function with respect to the parameters.

The representative features that learnt from the convolutional auto-encoder can be used as the input of next auto-encoder process. After the weights are fine-tuned with backpropagation, the top level activations can be used as the feature representation for further supervised learning.

3.3 From Pseudo-Relevant Images to Truly Relevant Images

With the pair-wise stacked convolutional auto-encoders, we will learn feature representations for only "subjective adjective" of a certain "noun". With these significant features, we can do clustering to the training image pairs. The similarity between these image pairs are computed by Euclidean metric between their differences' feature representations. We apply K-means algorithm for clustering. The largest cluster are treated as image pairs that include a truly relevant image and a truly irrelevant image.

4. Experiment

Table 1 The queries we used in the experiment. (Ratio A: ratio of truly relevant images in pseudo-relevant images, Ratio B: ratio of truly relevant images in pseudo-irrelevant images)

Query	Ratio A	Ratio B
happy dog	0.785	0.18
tiny flower	0.688	0.3
clear sky	0.565	0.2
ancient city	0.865	0.2
falling snow	0.735	0.25
warm water	0.425	0.075
happy kids	0.83	0.3
dry flower	0.81	0.055
fluffy clouds	0.899	0.675
fresh flowers	0.78	0.6

In the experiment, because of the restricted images crawling from the search engines (not allowed to crawl or a very limited number of permission), we decided to use existing dataset that used in [3]. One advantage of using this dataset is that with the labels for each images, we do not need to spend extra cost to evaluate whether an image is relevant to a query or not in the evaluation phase. The images in the dataset are from Flickr and the dataset include 1553 ANPs (Adjective Noun Pairs) with their images. In order to make our dataset, we clustered all the ANPs based on nouns. The we selected ten queries (ANPs as called in their research) with nouns that have many adjectives in the cluster. We also considered the number of images for the queries to make sure that each query have more than 1000 images. Table 1 lists all the queries we used in the experiment.

To better simulate the ratio of truly relevant images in the pseudo-relevant image set and pseudo-irrelevant image set as in the real search engines, we conducted a survey of these ten queries in web image search engines (Google and Flickr). For each query, we surveyed the ratio of truly relevant images to the query in the top 200 result images with

Table 2 The size of input and output data in each layer of the network.

Process	Layer	Type	Input	Filter	Output
Encoding Process #1	Layer 1	Conv	100 × 100 × 3	5 × 5 × 3 × 20	96 × 96 × 20
	Layer 2	Pool (max)	96 × 96 × 20	2 × 2	48 × 48 × 20
Encoding Process #2	Layer 3	Conv	48 × 48 × 20	5 × 5 × 20 × 20	44 × 44 × 20
	Layer 4	Pool (max)	44 × 44 × 20	2 × 2	22 × 22 × 20
Decoding Process #2	Layer 5	Up-sampling	22 × 22 × 20	2 × 2	44 × 44 × 20
	Layer 6	Deconv	44 × 44 × 20	5 × 5 × 20 × 20	48 × 48 × 20
Decoding Process #1	Layer 5	Up-sampling	48 × 48 × 20	2 × 2	96 × 96 × 20
	Layer 6	Deconv	96 × 96 × 20	5 × 5 × 20 × 3	100 × 100 × 3

both pseudo-relevant images (results of the subjective adjective noun query) and pseudo-irrelevant images (results of the noun query). We took the average number as the simulation ratio as showed in Table 1. For each query, images of the query are treated as truly relevant images and images of other ANPs with the same noun are treated as truly irrelevant images (we also filtered some ANPs that have very similar adjectives, such as "excited kids" for "happy kids"). The pseudo-relevant image dataset and pseudo-irrelevant image dataset were constructed by adding images from the truly relevant images and truly irrelevant images with their numbers fit the ratio we surveyed in real image search engines.

In our experiment, we had two encoding processes and two decoding processes. We set the batch size of image as 100 × 100. Size of input and output in each layer is shown in Table 2 as well as the shape of each layer. For example, the first convolutional layer used 20 kernels to filter the 100 × 100 × 3 input images of the size 5 × 5 × 3. The output of the first convolutional layer is processed after pooling and then taken as input of the second convolutional layer (also the input of the second encoding process).

5. Result and Discussion

Table 3 Result of our approach and the precision of top 200 in image search engines for two queries.

Query	Precision@200*	Accuracy of ours
happy dog	0.565	0.617
clear sky	0.785	0.802

* the mean precision of top 200 in image search engines (Google image and Flickr)

Table 3 shows comparison of our approach and the mean precision of top 200 in image search engines (Google image and Flickr) for two queries. We can see that our approach could slightly outperform the current image search engine when the query is a "subjective adjective noun" query.

We consider much more space for improvement in this research. In the future, we will take consideration of similar subjective adjectives when getting pseudo-relevant images and pseudo-irrelevant images. Images in the dataset we use are not really truly relevant images and truly irrelevant images. In that case, we consider to make our own dataset that can perfectly match our research goal, such as sequence of images for the same objects. Parameter is a very important factor to influence the performance of deep neural network and we will need more trials to adjust them to make better performance. Moreover, we will try to get visual representation of the learnt features to have a better and intuitive understanding of what we have learnt with the network.

6. Conclusion

In this paper, we propose to solve the problem of estimating relevance of images to "subjective adjective noun" queries by first learning trustful relevant images with unsupervised deep convolutional auto-encoders and then learn to measure the relevance. We propose pair-wise stacked convolutional auto-encoders to find discriminative features that can represent differences between relevant images and irrelevant images. We show our conducted experiment and the result is compared with precision of some image search engines. Finally we make a discussion according to the result and we list some future plans.

7. Acknowledgement

This work was supported in part by Grants-in-Aid for Scientific Research (Nos. 15H01718, and 26700009) from MEXT of Japan.

References

[1] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2425–2433 (2015)

[2] Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H., et al.: Greedy layer-wise training of deep networks. Advances in neural information processing systems 19, 153 (2007)

[3] Borth, D., Ji, R., Chen, T., Breuel, T., Chang, S.F.: Large-scale visual sentiment ontology and detectors using adjective noun pairs. In: Proceedings of the 21st ACM international conference on Multimedia. pp. 223–232. ACM (2013)

[4] Chen, T., Borth, D., Darrell, T., Chang, S.F.:

Deepsentibank: Visual sentiment concept classification with deep convolutional neural networks. arXiv preprint arXiv:1410.8586 (2014)

[5] Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural computation 14(8), 1771–1800 (2002)

[6] Huang, F.J., Boureau, Y.L., LeCun, Y., et al.: Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: 2007 IEEE conference on computer vision and pattern recognition. pp. 1–8. IEEE (2007)

[7] Jing, Y., Baluja, S.: Visualrank: Applying pagerank to large-scale image search. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(11), 1877–1890 (2008)

[8] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

[9] Larochelle, H., Erhan, D., Courville, A., Bergstra, J., Bengio, Y.: An empirical evaluation of deep architectures on problems with many factors of variation. In: Proceedings of the 24th international conference on Machine learning. pp. 473–480. ACM (2007)

[10] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431–3440 (2015)

[11] Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: International Conference on Artificial Neural Networks. pp. 52–59. Springer (2011)

[12] Narihira, T., Borth, D., Yu, S.X., Ni, K., Darrell, T.: Mapping images to sentiment adjective noun pairs with factorized neural nets. arXiv preprint arXiv:1511.06838 (2015)

[13] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)

[14] Schmidhuber, J., Eldracher, M., Foltin, B.: Semilinear predictability minimization produces well-known feature detectors. Neural Computation 8(4), 773–786 (1996)

[15] Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. pp. 1096–1103. ACM (2008)

[16] Wiebe, J.: Learning subjective adjectives from corpora. In: AAAI/IAAI. pp. 735–740 (2000)

[17] You, Q., Luo, J., Jin, H., Yang, J.: Robust image sentiment analysis using progressively trained and domain transferred deep networks. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence. pp. 381–388. AAAI'15, AAAI Press (2015)