International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.10 (2016), pp.119-130
http://dx.doi.org/10.14257/ijmue.2016.11.10.11

Unsupervised Feature Learning Assisted Visual Sentiment Analysis

Zuhe Li[1, 2], Yangyu Fan[1], Fengqin Wang[2] and Weihua Liu[1]

[1]School of Electronics and Information, Northwestern Polytechnical University, Xi'an 710072, China
[2]School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China
zuheli@126.com

Abstract

Visual sentiment analysis which aims to understand the emotion and sentiment in visual content has attracted more and more attention. In this paper, we propose a hybrid approach for visual sentiment concept classification with an unsupervised feature learning architecture called convolutional autoencoder. We first extract a representative set of unlabeled patches from the image dataset and discover useful features of these patches with sparse autoencoders. Then we use a convolutional neural network (CNN) to obtain feature activations on full images for sentiment concept classification. We also fine-tune the network with a progressive strategy in order to filter out noisy samples in the weakly labeled training data. Meanwhile, we use low-level visual features to classify visual sentiment concepts in a traditional manner. At last the classification results with unsupervised feature learning and that with traditional features are taken into account together with a fusion algorithm to make a final prediction. Extensive experiments on benchmark datasets reveal that the proposed approach can achieve better performance in visual sentiment analysis compared to its predecessors.

Keywords*: visual sentiment; deep learning; unsupervised feature learning; sparse autoencoder; convolutional neural network*

1. Introduction

With the rapid development of social media, online visual content such as images and videos is becoming an overwhelmingly dominant media type on the web [1]. Sentiment analysis which mostly concentrates on textual content [2] before has been gradually extended to visual content. As the famous saying goes, a picture is worth one thousand words. Visual content which depicts strong sentiment offers rich complementary information that influences the audience more effectively [3, 4]. As a study to understand the rich emotion and sentiment in visual content, visual sentiment analysis will greatly benefit behavior science and enable broad applications in many areas such as market prediction and brand monitoring [1, 4].

Even though content-based image classification which models generic visual concept has been widely studied in computer vision, limited efforts have been conducted to visual content sentiment analysis. Most existing publications on social media sentiment analysis adopt a conventional approach to establish a mapping of low-level features to affects directly [5-6]. This does not work very well because there is a big “affective gap” between the low-level features and the emotional content in images and videos [1]. And modeling visual sentiment like “amazing” and “shy” is still difficult as this kind of information is abstract and subjective [4].

To our knowledge, one of the most prominent researches in this field is made by a team of Columbia University. They have constructed a large-scale Visual Sentiment Ontology

(VSO) which consists of more than 1,200 Adjective Noun Pairs (ANP) based on psychological theories [1]. Each ANP is made of an emotion-related adjective and a noun corresponding to specific objects or scenes that have a feasibility of automatic detection [3]. What's more, they presented a visual concept detector library to detect the presence of 1,200 ANPs in visual content, called SentiBank, which establishes a novel mid-level features to bridge the affective gap [1]. This research opens a new way to visual sentiment analysis because it has partly solved the problem of affective gap.

In this era of big data, deep learning framework has been successfully applied to computer vision and produces the state-of-the-art performance on various tasks such as digit recognition [7-8] and image classification [9-10]. Coincidentally, there are about one million images in the dataset collected in [1] for visual sentiment analysis. This provides enough data for the training of deep learning algorithms and makes it feasible to analyze visual sentiment using deep learning algorithms. Recently, Chen *et al.* [4] and You *et al.* [11] have applied deep convolutional neural networks (CNNs) to visual sentiment analysis and achieved better performance. However, they trained large, deep convolutional neural networks in a supervised way and the classification result is not satisfying when the corresponding data of an ANP is insufficient. Inspired by the recent success of deep learning in visual sentiment analysis, we are interested in the feasibility of classifying visual sentiment concepts with unsupervised learning algorithms, which have been successfully used to extract generally useful features from visual content. These automatically learned features are different from artificial features like SIFT and are particularly suitable for applications with limited label information, such as pedestrian detection [12].

In this paper, we propose a visual sentiment concept classification framework with a convolutional autoencoder to discover useful structures of the input data in an unsupervised way. Since the labels of the large-scale dataset in [1] are machine generated, the training image data is weakly labeled and rather noisy. Thus, we attempt to leverage a progressive training strategy [11] to further fine-tune the neural network. Finally, we employ a late fusion algorithm to classify visual sentiment concepts by combining the classification result on this framework and that with traditional methods. Our evaluation results suggest that this strategy is effective for improving the ANP classification performance.

In the rest of this paper, we will start by discussing related work in Section 2 and move on to describe the architecture of the proposed framework in Section 3. Then we will describe unsupervised feature learning and the details of our approach in Section 4. We then present our experiments and the results in Section 5. Finally, we will conclude this paper and discuss future work in Section 6.

2. Related Work

So far, researchers have achieved much promising progress in text-based sentiment analysis [2, 13]. However, people are more likely to express emotions with visual content such as images and videos in the context of social media. There are huge amounts of visual data available in the modern network. This adds additional challenges to sentiment analysis.

Sentiment analysis based on visual content has been much less studied compared with textual content. Still, there are also several recent works on sentiment analysis based on images and videos. Siersdorfer *et al.* [14] applied machine learning techniques to predict the sentiment of images using the bag-of-visual words representation and the color distribution of images. Considering the difficulty of mapping low-level visual features to sentiment, Borth *et al.* [1, 3] and Yuan *et al.* [15] employed attributes or entities as mid-level features to analyze visual sentiment. As mentioned in Section 1, Borth *et al.* [1, 3] designed a large-scale visual sentiment ontology based on Adjective-Noun Pairs corresponding to different emotions. They crawled images from Flickr using these ANPs and trained 1200 ANP detectors with low-level features of images in each ANP. Then they used the responses of these classifiers as mid-level features to predict sentiment in visual content. This version of classifiers of the ANPs is denoted as SentiBank 1.1[4]. Similarly, Yuan *et al.* [15] employed 102 scene attributes as mid-level features. Chen *et al.* [16] further improved the model in [1] in an object-based manner by decomposing the problem into object localization and sentiment concept modeling. This version of SentiBank with object-based localization is called SentiBank 1.5R [4].

Recently, researchers have started to apply deep learning algorithms to this challenging task. Chen *et al.* [4] proposed a visual sentiment concept classification method based on the deep CNNs which show great performance improvement on image classification. Both annotation accuracy and retrieval performance of the newly trained model SentiBank 2.0 are significantly improved compared to its predecessors. You *et al.* [11] also proposed a suitable CNN framework for visual sentiment analysis and adopted a progressive strategy to fine-tune the network.

To our knowledge, there have not been any attempts to apply unsupervised feature learning algorithms such as sparse autoencoders to visual sentiment analysis. However, some researchers have successfully introduced unsupervised feature learning into other areas such as satellite imagery classification [17, 18]. Inspired by these successes, we intend to study the feasibility of unsupervised feature learning algorithms in visual sentiment analysis.

3. Overall Architecture

Here we describe the overall architecture of the proposed framework for visual sentiment concept (ANP) classification. As shown in Figure 1, we present a parallel scheme to train an ANP detector using both artificial low-level features and features with unsupervised learning. To avoid overfitting, we don’t adopt the early fusion method to merge and normalize these two kinds of features into a single vector. Instead, we use a late fusion algorithm to achieve a final result by the fusion of the two detector scores after their respective classification.

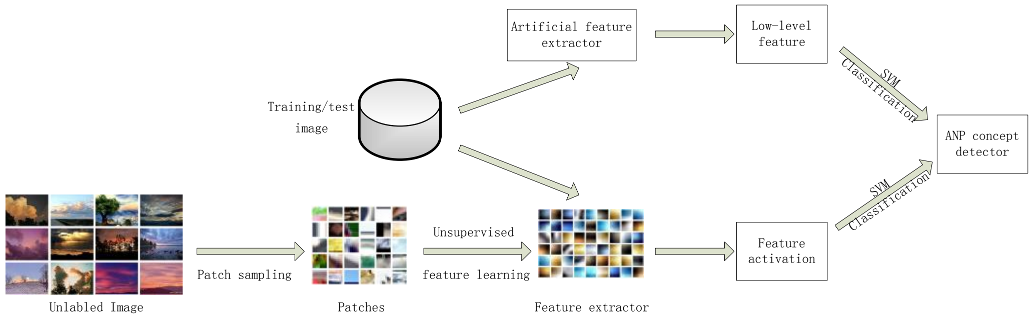

Figure 1. Overall Architecture of the Hybrid Framework for ANP Classification

For classification with artificial features, we directly use the detector library (SentiBank 1.1) released in [1]. The visual features they used include Color Histogram in RGB color space, GIST descriptor, Local Binary Pattern (LBP) descriptor, a Bag-of-Words quantized descriptor and a 2000 dimensional attribute useful for characterizing abstract ANPs. Linear SVMs are employed to train these ANP detectors to ensure high efficiency.

The classification framework with unsupervised feature learning consists of four parts: patch sampling, unsupervised feature learning, feature extraction and classification. First, image patches with size of $n \times n$ are sampled randomly from the image dataset of each ANP. Each $n \times n$ patch has three channels(R, G and B) and can be combined into one long vector in R^N of the intensity values, where $N=n \times n \times 3$. The data set is then fed to a sparse autoencoder to learn K feature extractors with unsupervised learning. These automatically learned feature extractors are different from the manually designed feature extractors mentioned above. Then we can use these feature extractors to learn feature activations from the training and test images with convolution and pooling operation. Finally, we also employ a Linear SVM to predict the presence of an ANP. In the training process, we fine-tune the model with a progressive strategy to reduce the impact of the noisy training data. The process of unsupervised feature learning will be described in detail in the next section.

In the output layer, we have two detector scores using different approaches. Here we propose a simple fusion algorithm to choose the detector score which is less close to 0.5 as the final result. Let $s \in S = \{s_1, s_2\}$ be the classification scores of an ANP using the two approaches, the result with a larger value of $|s - 0.5|$ is selected.

4. Unsupervised Feature Learning and Progressive Fine-Tuning

In this section, we will describe the details of the sparse autoencoder-based unsupervised feature learning algorithm and the progressive fine-tuning in training process. An unsupervised feature learning algorithm can learn features from the image patches sampled from the unlabeled data and discover the features of a whole image with convolution and pooling.

4.1. Sparse Autoencoder

As an unsupervised learning algorithm to discover useful structures of the input data, an autoencoder is a symmetrical neural network whose target values are equal to the inputs [19, 20]. In an autoencoder, an input vector $x^{(i)} \in R^N$ is fed to the network and the outputs of the hidden layer are as follows:

$$a^{(i)} = f(z^{(i)}) = f(W_1 x^{(i)} + b_1) \tag{1}$$

where $x^{(i)}$ is the i-th training sample, W_1 is the weight matrix , b_1 is the bias vector and $f(\square)$ is the activation function. Here we will choose the sigmoid function as the activation function:

$$f(z) = \frac{1}{1 + \exp(-z)} \tag{2}$$

In our study, we have rescaled the pixel values from the range [0, 255] to [0, 1] by dividing the data by 255.Thus we also need to constrain the outputs to be in the range [0, 1] with the sigmoid function as follows:

$$\hat{x}^{(i)} = f(\hat{z}^{(i)}) = f(W_2^T a^{(i)} + b_2) \tag{3}$$

where $\hat{x}^{(i)}$ is the i-th reconstructed output value, W_2 is the decoding weight matrix and b_2 is the decoding bias.

Like other neural networks, a sparse autoencoder learns feature extractors in the dataset by minimizing the cost function. The cost function we adopted here includes three parts: an error term, a regularization term (weight decay term) and a sparsity penalty term [18,20-23]. The total cost function is expressed as follows:

$$J\left(W,b\right)=\left[\frac{1}{2m}\sum_{i=1}^{m}\left\|x^{(i)}-\hat{x}^{(i)}\right\|^{2}\right]+\frac{\lambda}{2}\left\|W\right\|^{2}+\beta\sum_{j=1}^{K}KL(\rho\|\hat{\rho}_{j}) \tag{4}$$

where m is the number of training data, λ is the weight decay parameter, β is the weight of the sparsity penalty term, ρ is the sparsity parameter, $\hat{\rho}_j$ is the average activation of the *j*-th hidden unit, *K* is the number of features in the hidden layer and $KL(\square)$ is the Kullback-Leibler (KL) divergence.

The sparsity penalty term is used to force $\hat{\rho}_j$ to be approximately equal to ρ which is a small value close to zero. The KL-divergence [24] which is a standard function measuring the difference of two distributions is given by

$$KL(\rho\|\hat{\rho}_{j})=\rho\log\frac{\rho}{\hat{\rho}_{j}}+\left(1-\rho\right)\log\frac{1-\rho}{1-\hat{\rho}_{j}} \tag{5}$$

We use the back propagation algorithm [18, 25] to train the model by minimizing *J(W, b)* as a function of *W* and *b*. And we adopt the limited Broyden-Fletcher-Goldfarb-Shanno(L-BFGS) [26] algorithm to carry out the minimization process. Let $\hat{\rho}$ be a vector of the average activation of each hidden unit $\hat{\rho}_j$, the learning rule for W_2, W_1, b_2 and b_1 is described as follows:

$$\Delta W_{2}=(\hat{x}^{(i)}-x^{(i)})\cdot f^{'}(\hat{z}^{(i)})\cdot(a^{(i)})^{T} \tag{6}$$

$$\Delta W_{1}=(W_{2}^{T}\cdot(\hat{x}^{(i)}-x^{(i)})\cdot f^{'}(\hat{z}^{(i)})+\beta(-\frac{\rho}{\hat{\rho}}+\frac{1-\rho}{1-\hat{\rho}}))\cdot f^{'}(z^{(i)})\cdot(x^{(i)})^{T} \tag{7}$$

$$\Delta b_{2}=(\hat{x}^{(i)}-x^{(i)})\cdot f^{'}(\hat{z}^{(i)}) \tag{8}$$

$$\Delta b_{1}=(W_{2}^{T}\cdot(\hat{x}^{(i)}-x^{(i)})\cdot f^{'}(\hat{z}^{(i)})+\beta(-\frac{\rho}{\hat{\rho}}+\frac{1-\rho}{1-\hat{\rho}}))\cdot f^{'}(z^{(i)}) \tag{9}$$

4.2. Feature Extraction

To learn features from large-size images, we can use a convolutional network to reduce the computational cost without training on full images. The architecture of the framework for feature extraction is depicted in Figure 2. It is a network with convolutional and pooling layers to extract features from large-size images in training and test dataset with the features learned from small patches by sparse autoencoders.

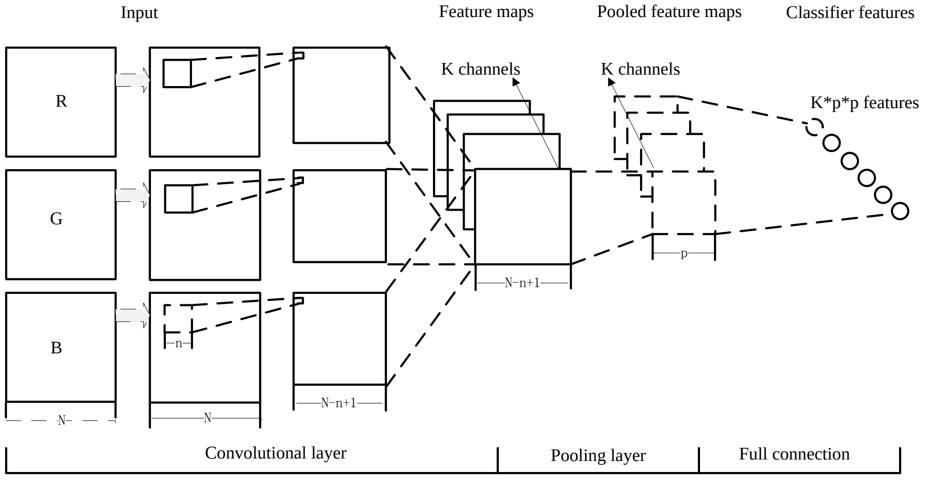

Figure 2. The Architecture of the Framework for Feature Extraction

4.2.1. Convolutional Layer

Having learned features over $n \times n$ patches, we convolve them with a larger image to obtain different feature activations at each location on the whole image. For computational efficiency, we perform a 2D convolution in each color channel separately without combining the intensities of three color channels into one vector. And the calculated values of every color channel should be summed up after separate convolution. Concretely, given an image of $N \times N$ pixels with three color channels and K feature extractors learned from $n \times n$ patches by sparse autoencoders, we can obtain a $(N\text{-}n+1)\times(N\text{-}n+1)$ array of convolved features with K channels.

4.2.2. Pooling Layer

A Pooling layer in a convolutional network combines the outputs of neuron clusters in the previous layer to reduce the resolution of feature maps and achieve spatial invariance [9, 10, 27]. After pooling operation, computational cost is reduced and over-fitting can be avoided. Thus a pooling operation aggregates the feature activations within a region R_j to generate a pooled feature s_j as follows:

$$s_j = pool\left(a_i\right) \forall i \in R_j \tag{10}$$

where a_i is a feature activation with the index of i in a region and R_j denotes a pooling region j. The pooling region can overlap each other in varying sizes [27].

Though several pooling methods have been proposed, average pooling and max pooling are still the most common methods. In this paper we use a max-pooling strategy without overlapping.

4.3. Progressive Fine-Tuning

As mentioned in Section 1, the images of the dataset for visual sentiment analysis in [1] are automatically crawled from social media and flickr tags are used directly as pseudo labels of ANPs [1].This might incur either false positive, *i.e.* a sample is labeled with an ANP but it was the exact opposite, or false negative, *i.e.* a sample is not labeled by an ANP but shows the ANP. An Amazon Mechanical Turk (AMT) experiment conducted by the authors in [1] has shown that false positive is not a big problem. But solving the false negative issue is difficult since it needs to label all the samples for all ANPs. Though they tried to minimize the probability of false negative by randomly sampling positive samples of other ANPs as the negative set for each ANP, this problem persists because there are similar ANPs with different names such as “magical forest” and “amazing trees”. So our main work is to reduce the impact of false negative.

The neural network may get stuck at a bad local optimum in the training process because of the noise in the training set. To reduce the effect of noisy training images, You *et al.* [11] proposed a progressive method to fine-tune the CNN for visual sentiment by selecting a subset of the training images progressively. Inspired by their work, we employ a strategy to fine-tune the convolutional autoencoder we proposed. Figure 3 shows the overall flow of the progressive fine-tuning.

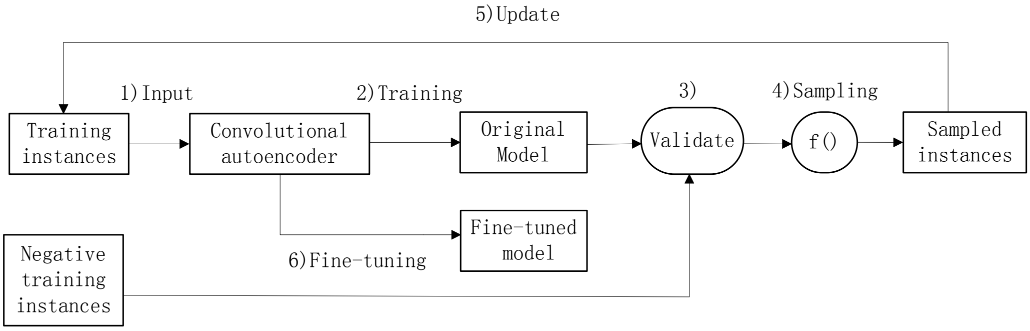

Figure 3. Overall Flow of the Progressive Fine-Tuning Strategy

We first train the convolutional autoencoder with noisy training images of each ANP. Then we use the trained model to validate the training images themselves and filter the training set by removing the negative samples with high detector scores. In the sampling process, we update the negative training subset of an ANP with a function *f()* which is a probabilistic sampling algorithm aiming to remove the negative instances having high detector scores with high probability. Concretely, let *s* be the detector score of an image in the negative subset of an ANP, we choose to remove the training instance with a probability of *p* given by Equation 7.

$$p = \max(0, \frac{\exp(s-0.5)-1}{\exp(0.5)-1}) \tag{11}$$

When the detector score of one negative training instance is smaller than 0.5, we will keep this negative training sample in the training set. Otherwise, the larger the detector score becomes, the larger the probability of this sample being excluded from the training subset.

Next, we further fine-tune the model using these sampled instances which are potentially cleaner than before. Finally, we choose the fine-tuned model as the final model for visual sentiment analysis.

5. Experiments

We first trained detectors for each ANP based on the hybrid model we proposed and evaluated the new classification model by both annotation accuracy and retrieval performance. Furthermore, we used the proposed model to predict sentiment on a benchmark and compared its performance with conventional methods.

5.1. ANP Detectors Training

To train new ANP detectors with our framework, we used the Flickr images released in [1] and [3]. The database consists of a set of Flickr images with Creative Common (CC) licenses organized by 1553 ANPs to train and test 1200 ANP detectors in SentiBank. To directly utilize the results obtained in [1] with a fusion algorithm, we selected the images corresponding to the same 1200 ANPs chose in [1] to train our model. For testing we selected 20 images from the subset of each ANP randomly. For training we sampled the remaining positive images of each ANP and twice as many negative images by randomly sampling positive samples of other ANPs.

We first resized all images of the dataset to 256×256 without maintaining the aspect ratio and rescaled the pixel values from the range [0, 255] to [0, 1]. For unsupervised feature learning, we extracted 100,000 small 11×11 patches from the dataset to learn 400 features using a sparse autoencoder with training parameters λ =3e-3, β =5 and ρ =0.03. For feature extraction, we constructed a CNN including a convolutional layer with stride s=1 and a max-pooling layer over non-overlapping regions of size 20×20.For each ANP, we utilized a Linear SVM as classification model to obtain an ANP detector. And the model was fine-tuned with the progressive strategy described in Section 4.3 to filter the noisy training data. The detector score was finally combined with the score acquired from SentiBank 1.1 using late fusion.

5.2. Annotation Accuracy

To evaluate the annotation accuracy of the proposed model, we adopt a method employed in [4] by measuring the percentage of test samples that have corresponding pseudo labels in top detected ANP concepts, called top-k accuracy. We evaluate the annotation accuracy on the 1200 ANPs mentioned in Section 5.1 by computing top-1, 5, 10 accuracies of each and all ANPs. We also compare the accuracies among fine-tuned hybrid model we proposed, hybrid model without fine-tuning, and SentiBank 1.1 [1]. For the reason that the 1200 ANP subsets of SentiBank 2.0 [4] have not been released to the public, we will make a comparison with it using retrieval performance in Section 5.3. The overall accuracies are shown in Table 1 and the top-10 accuracy per ANP is shown in the form of curves in Figure 4.

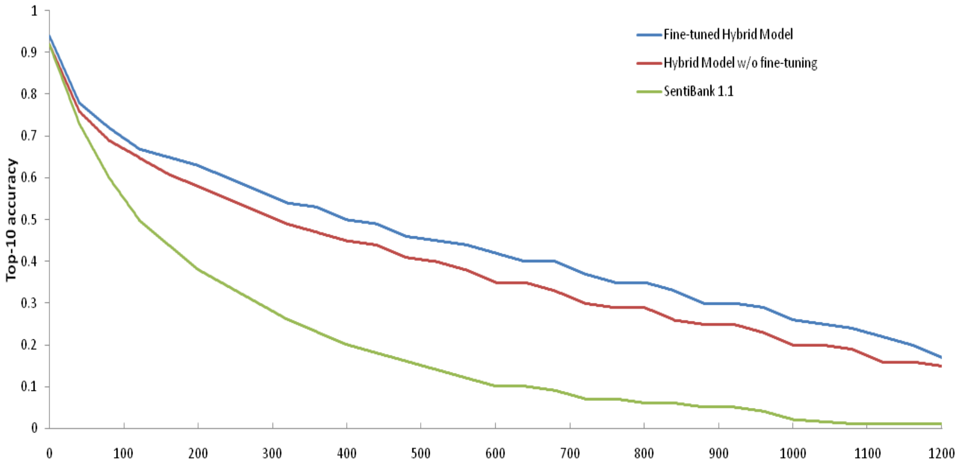

Figure 4. The Curves of Ranked Top-10 Accuracy Per ANP of Different Approaches

As shown in Table 1 and Figure 4, the unsupervised feature learning assisted model greatly improves the performance of the approach with low-level features only. It achieves a 320% increment on top-1 accuracy, 160% on top-5 accuracy, and 110% on top-10 accuracy respectively. And progressive fine-tuned model outperforms that without fine-tuning with as much as 7~18% gain on top-k accuracy. It shows that the progressive fine-tuning strategy can partly solve the problem of noisy labels in the dataset. The SVM based models are more suitable for retrieval than annotation since they train binary classifiers rather than multi-label classifiers. Thus the retrieval performance of our model will be evaluated in the next section.

Table 1. Comparison of the Annotation Accuracy of Different Models

Models	Top-1	Top-5	Top-10
SentiBank 1.1	3.08%	11.60%	19.02%
Hybrid Model w/o fine-tuning	10.87%	27.06%	37.13%

Fine-tuned Hybrid Model	12.92%	30.04%	39.93%

5.3. Retrieval Performance

To make a comparison with the new DeepSentiBank[4] and SentiBank 1.5R[16] mentioned in Section 1 and 2,we also built a subset as the authors of [4] and [16] did to test the retrieval performance. Six frequently tagged nouns named “car”, “dog”, “dress”, “face”, “flower” and “food” were selected to form a large set of 135 ANPs with diverse adjectives. 20 positive images and 40 negative images were manually annotated to form the test set for each ANP. We applied our fine-tuned model and SentiBank 1.1 to the test set we constructed and ranked the test images according to the estimated probability of the ANP. Then we used the average precision (AP) at top 20 on the ranking result to evaluate the retrieval performance. The values of mean AP for each and all nouns are presented in Figure 5.

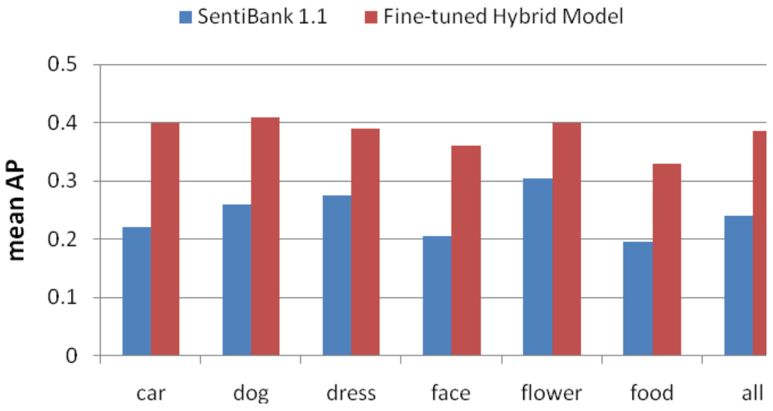

Figure 5. The Mean AP for Each and All Noun Categories for the Subset of 135 ANPs

As the test sets and the trained ANP detectors in [4] and [16] haven’t been made available to the community, the subset we built ourselves may be slightly different from them. However, the comparison result of retrieval performance of SentiBank 1.1, 1.5R and Deep SentiBank has been reported in [4]. So we make a comparison of various models in an indirect way. Concretely, we compute the degree of performance improvement of our model compared to SentiBank 1.1 on the subset we built. Then we compare this result with corresponding performance improvement of Deep SentiBank and SentiBank 1.5R obtained in [4] compared to SentiBank 1.1. In this way, we take the SentiBank 1.1 as a benchmark to compare our model with others.

As shown in Figure 5, our hybrid model outperforms SentiBank 1.1 by 60.4% for all noun categories. It is reported in [4] that Deep SentiBank outperforms SentiBank 1.1 by 62.3% and SentiBank 1.5R outperforms SentiBank 1.1 by 49.0%. This indicates that our model can achieve almost the same performance compared to the newly trained Deep SentiBank. And our model could be further improved if we use the special visual features described in [1] such as facial features and aesthetics related features. Moreover, the unsupervised learning framework could be further optimized with many skills which have shown good performance in other applications.

5.4. Sentiment Prediction

We further evaluated the performance improvement of the proposed model for sentiment prediction on a benchmark containing 603 photo tweets with a set of 21 topics covering human, social, event, people, location and technology [1]. The dataset was collected via the PeopleBrowsr API and ground truths for the collected image tweets were obtained by Amazon Mechanic Turk annotation. In this sense, we intend not only to detect the ANP concepts reflected in images but also to explain the sentiment with a prediction label. For each image, SentiBank provides a 1,200 dimension ANP response as a mid-level representation. In this paper, we employ linear classifiers such as Linear SVM and Logistic Regression to construct a mapping between the ANP response and sentiment prediction.

We used the hybrid framework proposed above to obtain ANP response in images and utilized both Linear SVM and Logistic Regression to train the sentiment prediction model. Prediction accuracy with this model and that obtained with SentiBank 1.1 and low-level features in [1] is given in Table 2. The hybrid model achieves significant performance improvement (about 8% relatively compared to SentiBank 1.1). As is found in [1], the logistic regression model performs a little better than Linear SVM.

Table 2. Visual-based Tweet Sentiment Prediction Accuracy

Models	Linear SVM	Logistic Regr.
Low-level Features[1]	0.55	0.57
SentiBank 1.1[1]	0.67	0.70
Fine-tuned Hybrid Model	0.73	0.75

6. Conclusion

In this paper, we present a hybrid framework to introduce autoencoder-based unsupervised feature learning algorithms into visual sentiment concept classification. To deal with the pseudo labeled training data which contains noisy images, we utilize a progressive fine-tuning strategy to further optimize the network by filtering potentially false negative samples. Experimental results show the hybrid model performs better in both annotation and retrieval compared to previous classification model with only low-level features. Our work indicates the feasibility of applying unsupervised feature learning to visual sentiment analysis. And it also shows the possibility of the combination of novel deep learning algorithms and traditional technologies in computer vision. In the future, we will incorporate other special visual features into our model and further improve the performance by leveraging the newly proposed techniques in deep learning.

Acknowledgements

This work was supported by the Science and Technology Innovation Engineering Program for Shaanxi Provincial Key Laboratories under Grant 2013SZS15-K02.

References

[1] D. Borth, R. Ji, T. Chen, T. Breuel, and S. F. Chang, “Large-scale visual sentiment ontology and detectors using adjective noun pairs”, Proceedings of the 21st ACM international conference on Multimedia, Barcelona: Spain, **(2013)**.

[2] B. Pang and L. Lee, “Opinion Mining and Sentiment Analysis”, Information Retrieval, vol. 2, no. 1-2, **(2008)**, pp. 1-135.

[3] D. Borth, T. Chen, R. Ji, and S. F. Chang, “SentiBank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content”, Proceedings of the 21st ACM international conference on Multimedia, Barcelona: Spain, **(2013)**.

[4] T. Chen, D. Borth, T. Darrell, and S. F. Chang, “DeepSentiBank: Visual Sentiment Concept Classification with Deep Convolutional Neural Networks”, arXiv preprint arXiv: 1410.8586, **(2014)**.

[5] J. Machajdik and A. Hanbury, “Affective Image Classification using Features inspired by Psychology and Art Theory”, Proceedings of the international conference on Multimedia, Florence: Italy, **(2010)**.

[6] J. Jia, S. Wu, X. Wang, P. Hu, L. Cai, and J. Tang, "Can we Understand van Gogh's Mood?Learning to infer Affects from Images in Social Networks", Proceedings of the 20th ACM international conference on Multimedia, Nara: Japan, **(2012)**.

[7] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition", Neural Computation, vol. 1, no. 4, **(1989)**, pp. 541-551.

[8] G. E. Hinton, S. Osindero and Y.-W Teh, "A fast learning algorithm for deep belief nets", Neural Computation, vol. 18, no. 7, **(2006)**, pp. 1527-1554.

[9] D. C. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification", Proceedings of International Joint Conference on Artificial Intelligence, Barcelona: Spain, **(2011)**.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Image net classification with deep convolutional neural networks", Proceedings of Advances in neural information processing systems, Lake Tahoe: USA, **(2012)**.

[11] Q. You, J. Luo, H. Jin, and J. Yang, "Robust Image Sentiment Analysis using Progressively Trained and Domain Transferred Deep Networks", Proceedings of The Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI), Austin: USA, **(2015)**.

[12] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning", Proceedings of Computer Vision and Pattern Recognition, Portland: USA, **(2013)**.

[13] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text", Journal of the American Society for Information Science and Technology, vol. 61, no. 12, **(2010)**, pp. 2544-2558.

[14] S. Siersdorfer, E. Minack, F. Deng, and J. Hare, "Analyzing and predicting sentiment of images on the social web", Proceedings of the international conference on Multimedia, Florence: Italy, **(2010)**.

[15] J. Yuan, S. Mcdonough, Q. You, and J. Luo, "Sentribute: image sentiment analysis from a mid-level perspective", Proceedings of the Second International Workshop on Issues of Sentiment Discovery and Opinion Mining, Chicago: USA, **(2013)**.

[16] T. Chen, F. X. Yu, J. Chen, Y. Cui, Y. Y. Chen, and S. F. Chang, "Object-based visual sentiment concept analysis and application", Proceedings of the ACM International Conference on Multimedia, Orlando: USA, **(2014)**.

[17] A. M. Cheriyadat, "Unsupervised feature learning for aerial scene classification", IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 1, **(2014)**, pp. 439-451.

[18] F. Zhang, B. Du, and L. Zhang, "Saliency-Guided Unsupervised Feature Learning for Scene Classification", IEEE Transactions on Geoscience and Remote Sensing, vol. 53, no. 4, **(2015)**, pp. 2175-2184.

[19] H. C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, **(2013)**, pp. 1930-1943.

[20] A.Y. Ng, "Sparse autoencoder", CS294A Lecture notes, **(2011)**, pp. 72.

[21] M. A. Ranzato, Y. Boureau, and Y. LeCun, "Sparse feature learning for deep belief networks", Proceedings of Advances in neural information processing systems, Vancouver: Canada, **(2008)**.

[22] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders", Proceedings of the 25th international conference on Machine learning, Helsinki: Finland, **(2008)**.

[23] H. Lee, C. Ekanadham, and A. Ng, "Sparse deep belief net model for visual area V2", Proceedings of Advances in neural information processing systems, Vancouver: Canada, **(2008)**.

[24] S. Kullback and R. A. Leibler, "On information and sufficiency", The Annals of Mathematical Statistics, vol. 22, no. 1, **(1951)**, pp. 79-86.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors", Nature, vol. 323, no. 6088, **(1986)**, pp. 533-536.

[26] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization", Mathematical Programming, vol. 45, no. 3, **(1989)**, pp. 503–528.

[27] Z. Li, Y. Fan, and W. Liu, "The effect of whitening transformation on pooling operations in convolutional autoencoders", EURASIP Journal on Advances in Signal Processing, vol. 2015, no. 1, **(2015)**, pp. 1-11.

Authors

Zuhe Li, received the B.S. degree in electronic information science and technology from Zhengzhou University of Light Industry, Zhengzhou, China, in 2004, and the M.S. degree in communication and information system from Huazhong University of Science and Technology, Wuhan, China, in 2008.He is currently pursuing the Ph.D. degree at the Northwestern Polytechnical University, Xi’an, China. His major research interests include Computer Vision and Machine Learning.

Yangyu Fan, received the B.S. degree and M.S. degree from Shaanxi University of Science & Technology, Xi’an, China, in 1982 and 1992 respectively, and the Ph.D. degree from Northwestern Polytechnical University, Xi’an, China, in 1999. He is the director of the Laboratory of Multimedia and Virtual Reality of School of Electronics and Information, Northwestern Polytechnical University. His research interests include Computer Vision, Virtual Reality and Signal Processing.

Fengqin Wang, received the B.S. degree from Zhengzhou University, Zhengzhou, China, in 2002, and the M.S. degree and Ph.D. degree from Northwestern Polytechnical University, Xi’an, China, in 2005 and 2010 respectively. Her research interests include Video Coding, Virtual Reality and Signal Processing.

Weihua Liu, received the B.S. degree from Xi’an Technological University, Xi’an, China, in 2009. He is currently pursuing the Ph.D. degree at Northwestern Polytechnical University, Xi’an, China. His major research interests include Computer Vision, Human gesture recognition and Machine Learning.