Pattern Recognition Letters 63 (2015) 23–29

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Scene analysis by mid-level attribute learning using 2D LSTM networks and an application to web-image tagging[☆]

Wonmin Byeon[a,b,*], Marcus Liwicki[a], Thomas M. Breuel[a]

[a] University of Kaiserslautern, Gottlieb-Daimler-Str., Kaiserslautern 67663, Germany
[b] German Research Center for Artificial Intelligence (DFKI), Trippstadter Str., Kaiserslautern 67663, Germany

ARTICLE INFO

Article history:
Received 17 September 2014
Available online 11 June 2015

Keywords:
Recurrent neural network
LSTM
Mid-level attribute learning
Scene analysis
Web-image tagging

ABSTRACT

This paper describes an approach to scene analysis based on supervised training of 2D Long Short-Term Memory recurrent neural networks (LSTM networks). Unlike previous methods, our approach requires no manual construction of feature hierarchies or incorporation of other prior knowledge. Rather, like deep learning approaches using convolutional networks, our recognition networks are trained directly on raw pixel values. However, in contrast to convolutional neural networks, our approach uses 2D LSTM networks at all levels. Our networks yield per pixel mid-level classifications of input images; since training data for such applications is not available in large numbers, we describe an approach to generating artificial training data, and then evaluate the trained networks on real-world images. Our approach performed significantly better than others methods including Convolutional Neural Networks (ConvNet), yet using two orders of magnitude fewer parameters. We further show the experiment on a recently published dataset, outdoor scene attribute dataset for fair comparisons of scene attribute learning which had significant performance improvement (ca. 21%). Finally, our approach is successfully applied on a real-world application, automatic web-image tagging.

1. Introduction

Machine learning and deep neural networks have yielded much progress in recent years for tasks like classifying, tagging, and recognizing images [20,22]. Such methods have important applications in areas like web image search, personal image search, assistance devices for the blind, self-driving cars, and many more [2]. Commonly used databases (e.g., CIFAR, ImageNet [21,33]) in this research have often been low resolution compared to current digital images found on the web and tend to have target objects occupy a large fraction of the input image. As image recognition tasks become larger and more complex, issues of segmentation and texture classification become more important [36]. In traditional computer vision, these problems have been addressed using manually constructed feature descriptors, sometimes combined with simple classifiers, Markov Random Fields, and similar approaches [4,27].

For instance, one of the most popular methods, the Bag-of-visual-Words (BoVW) feature model, describes images as a set of local feature histograms, and classifies them using non-linear Support Vector Machines (SVM) [8,32]. There have been extended to multiple combinations of patch detectors and descriptors [23,38].

Recently, learning higher-level semantic contents (such as object parts and materials) beyond the plain visual cues (such as color and texture) had been achieved significant performance gains [10,11,29] in object recognition. Scene analysis is often thought of as a series of increasingly higher levels of abstraction, from raw pixel values to high-level whole attributes, object identifies, and categories. However, unlike object recognition, scene images cannot be identified as a single category; Each scene contains multiple salient attributes (for instance, sky, buildings, and ground). Therefore, inspired by the idea of learning semantic attribute, we introduce *mid-level attribute* to describe scene images. The mid-level attributes in the scene images are visual concepts, which are particularly closely related to the visual properties (such as sky, ocean, or sand). The learning process is similar to texture/material classification [7] which analyzes the regular textures, but the textures here are mid-level attributes, which are the parts of scenes and appear in complex scene images including many other elements.

One of the popular directions to learning visual properties is deep learning [25]. The approach creates a model to learn high-level abstractions from data. Recently, Convolutional Neural Networks (ConvNets) have been very successful in many image classification tasks like object recognition [22], video classification [19],

[☆] This paper has been recommended for acceptance by J. Laaksonen.
[*] Corresponding author. Tel.: +4917699013491.
E-mail addresses: wonmin.byeon@dfki.de (W. Byeon), liwicki@cs.uni-kl.de (M. Liwicki), tmb@cs.uni-kl.de (T.M. Breuel).

http://dx.doi.org/10.1016/j.patrec.2015.06.003
0167-8655/© 2015 Elsevier B.V. All rights reserved.

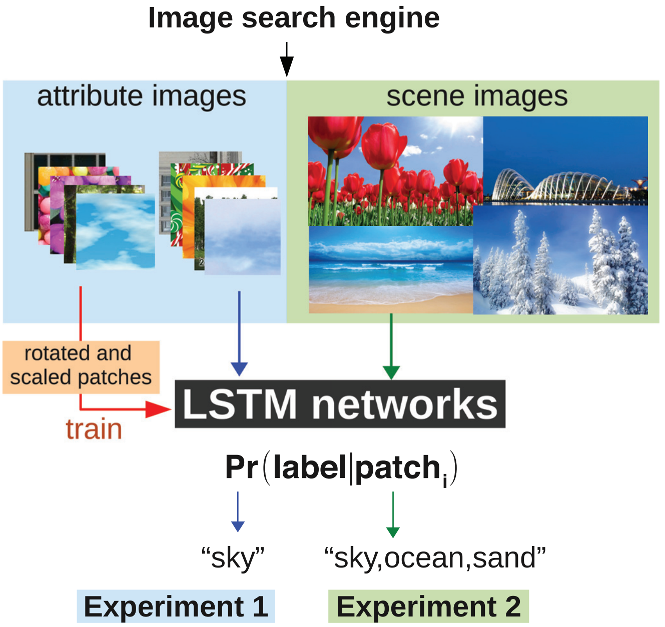


Fig. 1. An overview of our system. *Training: mid-level attribute learning (red).* 2D LSTM recurrent neural network model is trained on the mid-level visual attribute data which are collected from web image search engine. *Experiment 1: visual attribute classification (blue).* It is first tested on single attribute images. The images contain only a single attribute per image (e.g., "sky") *Experiment 2: natural scene analysis (green).* Real-world scene images are used for this experiment. These images include several attributes (e.g., "sky, ocean, and sand"). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

and face recognition [35], through a combination of different layers (like convolutional, pooling layers, etc) with very large numbers of parameters.

This paper describes an approach to learning mid-level attribute labels similar to deep learning and ConvNets in that it learns attribute labels directly from raw pixel values without intervening manual feature extraction or other pre-processing steps. In our experiments, we compare deep learning with an architecture consisting of a single 2D Long Short-Term Memory recurrent neural networks (LSTM networks) [14,17], which we will see outperforms the deep ConvNets in this task. The LSTM networks have found to give better performance on a number of tasks: image segmentation [15], off-line handwriting recognition [16], texture classification [7] and segmentation [6].

In this work, our networks are evaluated on natural scene images containing a mix of mid-level attributes collected from web-searches. This experiment further demonstrates the real-world application, i.e., automatic web-image tagging, which shows that our model is directly applicable to a more realistic scenario. We also apply our model to public scene attribute dataset (SceneAtt) and report the significant improvement over the baseline by Wang et al. [39] and ConvNets.

2. System description

In this section, we will briefly explain the 2D LSTM networks, then the training process (mid-level attribute learning using the LSTM networks) and two kinds of experiments: a large-scale attribute classification (Experiment 1), and unknown web scene image analysis (Experiment 2). An overview of our system is illustrated in Fig. 1.

2.1. 2D LSTM recurrent neural networks

The full network architecture we use is illustrated in Fig. 2. The network is divided into three parts: one input layer, one hidden layer (with four directional LSTM subnets), and one output layer. Note that this is a fairly shallow network compared to many deep learning approaches. The main procedure is as follows:

Input layer: As in other deep learning approaches, the 2D LSTM networks were given raw RGB pixel values. Although, the 2D LSTM networks could have operated directly over the entire image, we choose to divide into blocks. We compared sliding window and non-overlapping block approaches and found that the latter performed just as well, but faster. This process localizes features first before collapsing global information in the networks.

2D hidden layer: We use the standard 2D LSTM architecture with peepholes as described in [16]. Let us briefly summarize the structure and operation of this architecture. LSTM subnets in the hidden layer internally scan all surrounding contexts, and this process is iterated along all the image blocks, which contribute to the final decision. All states of LSTM (Input (i), Forget (f), Cell (c), Output (o) states, and Net-output (h)) are computed iteratively at $t = 1$ to T (T is the number of block) for all directions D ($2^{\text{dim}} = 2^2 = 4$: top-left (tl), top-right (tr), bottom-left (bl), and bottom-right (br)). The LSTM model uses

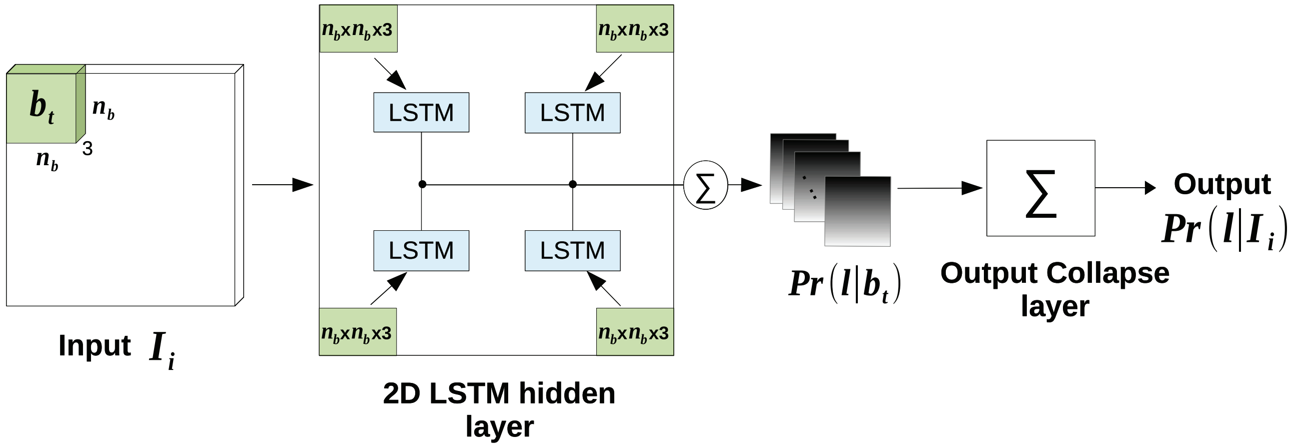


Fig. 2. 2D LSTM network architecture. First, raw RGB values at the block b_t ($3 \times n_b \times n_b$) is sent to the network. 2D LSTM hidden layer includes four LSTM subnets with two recurrent connections. The recurrent connections access to each dimension, and each hidden subnet accumulates the information of each direction (left to right, top to bottom, right to left, and bottom to top). Thus, it keeps the all surrounding context and processes it with the current pixel. All outputs of LSTM hidden layer ($Pr(l|b_t)$, $t = \{1 \ldots T\}$, T is the number of blocks per image) are collapsed, then the network outputs the class probabilities for each image ($Pr(l|I_i)$ for the ith image).

the standard equations [16]

$$i_t = f_1\left(W_i \cdot a_t + \sum_d \left(H_i^d \cdot h_{t-1}^d + C_i^d \cdot c_{t-1}^d\right) + b_i\right) \qquad \text{[Input]}$$

$$f_t^{d'} = f_1\left(W_f \cdot a_t + \sum_d \left(H_f^d \cdot h_{t-1}^d\right) + C_f^{d'} \cdot c_{t-1}^{d'} + b_f^{d'}\right) \qquad \text{[Forget]}$$

$$\widetilde{c}_t = f_2\left(W_{\widetilde{c}_t} \cdot a_t + \sum_d \left(H_{\widetilde{c}_t}^d \cdot h_{t-1}^d\right) + b_{\widetilde{c}_t}\right)$$

$$c_t = \sum_d \left(f_t^d \circledcirc c_{t-1}^d\right) + i_t \circledcirc \widetilde{c}_t \qquad \text{[Cell]}$$

$$o_t = f_1\left(W_o \cdot a_t + \sum_d \left(H_o^d \cdot h_{t-1}^d\right) + C_o \cdot c_t + b_o\right) \qquad \text{[Output]}$$

$$h_t = o_t \circledcirc f_2(c_t) \qquad \text{[Net-output]}$$

$$y_t = W_y \cdot h_t + b_y$$

where W, H, and C terms denote weight matrices for input to gates, recurrent connections, and cell to gates, respectively. d indicates the recurrent connections to x and y axes, i.e., $d' \in d = \{x, y\}$. f_1 and f_2 are the logistic sigmoid and hyperbolic tangent activation function, respectively. $(\circledcirc)$ is the element-wise product and $(\cdot)$ is matrix multiplication. At the end, y_t, the output activation vector at t is obtained and sent to the collapse layer.

Collapse layer: In order to contribute all blocks to a final prediction of an image, all activations are collapsed and send them to the output layer. The operation is simply the sum over all the inputs and directions, i.e., $M = \sum_{d=\{tl,tr,bl,br\}} \sum_{t=1}^{T} y_t^r$ where y_t^r is the final activation at block t and direction r, and M is the collapsed activation.

Output layer: Finally, M is fed to a Softmax layer, which will output the class probabilities for image i: $Pr(l|I_i) = \text{Softmax}(M)$.

2.2. Mid-level attribute learning

Training: an LSTM network model takes mid-level attribute images containing a single attribute per-image (like sky, ocean, or building; shown in Fig. 1, left).

In order to increase the number of training samples, we augmented the training data with rotated, scaled, and shifted version of the input as follows. First, randomly sized patches are selected in a random position. They are then rotated and scaled-up or down into the size of $n \times n$. This process is repeated multiple times for each image. It allows us to generate randomly scaled and rotated as well as the same sized patches which can be easily applied to the network without prior knowledge of the image resolution and condition. Thus, we show that our model can capture the variation of each attribute under limited number of training samples and keep the input dimension constant to retain one optimal model for different data.

2.3. Visual attribute classification

After training, models are first evaluated on single-attribute image similar to the training data (Experiment 1; Fig. 1, left). During the training phase, a number of patches are randomly extracted from a single image and passed through the network. The networks are expected to output conditional probabilities of all labels given each patch j: $Pr(\text{label} \,|\, \text{patch}_j)$. They are then integrated into one attribute label. Due to the way data is generated, images contain irrelevant (noise, clutter, or watermark) and mislabeled regions. We use the following smoothing process to correct this. The score of each label is first averaged by the number of patches then the best label is determined, such that the averaged score is high. Its equation is as follows:

$$\underset{\text{label}}{\arg\max} \frac{1}{\#\,\text{patches}} \sum_{j=1}^{\#\,\text{patches}} Pr\,(\text{label} \,|\, \text{patch}_j) \qquad (1)$$

To evaluate the performance, per-image accuracy is measured using the integrated score outlined above:

$$\text{Acc}_{\text{per-image}} = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} 1 & \arg\max_l \frac{1}{n}\sum_{j=1}^{n} Pr\,(l \,|\, p_j) \;=\; l_t, \\ 0 & \text{otherwise} \end{cases}$$

where n is the number of patches per image and N is the number of images. l and p indicate the label and patch, and l_t is the true label of the image.

2.4. Natural scene analysis

Although training was performed on specially constructed and augmented data, we use natural images containing a mix of mid-level features (Experiment 2; Fig. 1, right) obtained from the web-searches for testing and evaluation. Like for the visual attribute classification task, multi-patch inputs are sent to the network model, and each patch is classified according to the mid-level attribute it belongs to. The natural images obtained from the web-searches contain a wide range of scales, resolutions, rotation, and clutter. The following experiments are intended to demonstrate that our method works on such images and can directly applicable to more realistic applications. To achieve scene analysis of these complex natural images, we need additional mechanisms to apply the model trains in the previous section to images containing multiple mid-level attributes. The two primary methods for this are: probabilistic patch pruning and top-k rank pruning.

Probabilistic patch pruning: The random multi-patches may contain noise or unrelated contents (such as a logo, watermark, or ambiguous attribute texture). We first prune these patches to avoid confusion on the final decision. This pruning rule is based on the posterior class probability $Pr\,(l \,|\, p)$ of the label l given the patch p and the threshold T_p: $\max_l Pr\,(l \,|\, p) > T_p \; (0 < T_p \leq 1)$.

Top-k rank pruning: The basic idea is that the highly probable visual patterns, which tend to recur frequently are the main semantic regions of the scene. For the remaining patches after probabilistic patch pruning, the ranking score is computed from the output integration (Eq. (1)), and the potential attributes are then ranked based on their ranking score. Since the number of semantic regions (the number of k) is uncertain, we cannot easily define an optimal k. To handle this problem, top-k ranked list passes the threshold T_r: $\frac{1}{S}\sum_{s=1}^{S} Pr\,(l \,|\, p_s) > T_r \; (0 < T_r \leq 1)$, $S > N_p$, where S is the number of patches with a corresponding label after the patch pruning. The label is also rejected if the final number of patches of the label is less than the integer value N_p. Thus, it prunes unreliable class labels from a statistical observation (the frequencies of the highly probable patches of the label) on the image.

3. Experiments

3.1. Web-image dataset

We used Google's image search[1] for collecting both the attribute and the scene images. For these experiments, we chose twelve common mid-level categories, some frequent and generic in outdoor scenes (building, flower, forest, grass, snow, ocean, sand, sky, and gravel), and narrow (stucco candy, and meat). When constructing our dataset, we aim to achieve the following properties: (1) plenty of diverse samples are created, (2) the uncontrolled raw web-data containing noise and errors are directly applied without manual annotation or pre-selection, (3) randomly collected scene images (unseen and unknown data) represent the real-world data and the understanding of them is only by the visual attribute information, and

[1] http://images.google.com.

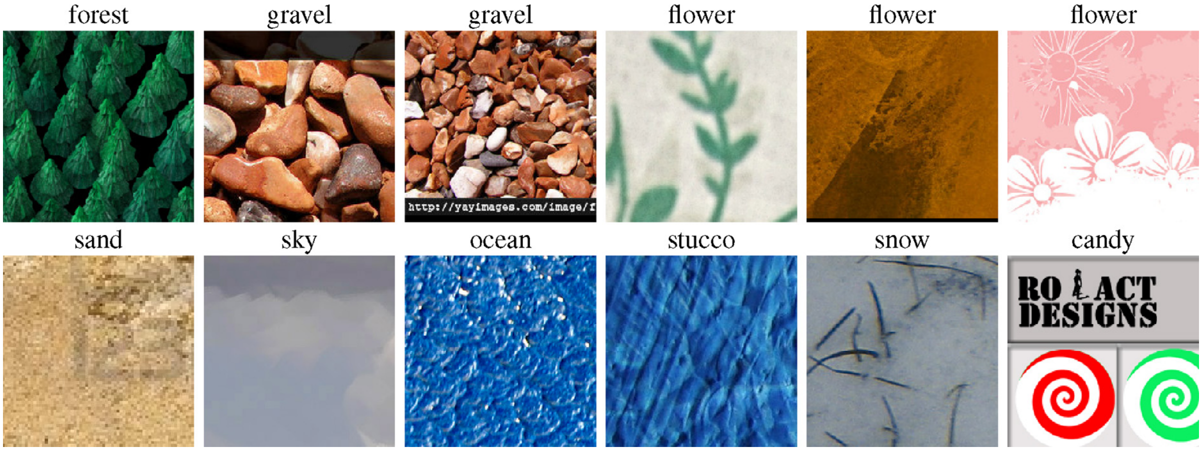


Fig. 3. The difficulties of web-image dataset. The images contain errors (wrong labels) or noise (logos, watermark, or irrelevant objects).

(4) web-images retrieved from a web image search are used directly for tagging, which show the more realistic scenario.

Attribute data: Ideal training data for our task would consist of manually segmented and labeled natural images, as for example, found in the databases from [13,39]. However, little training data of that form is available. Therefore, we created training data by the web-search querying the attribute's texture (e.g., sky texture). We performed no manual correction of validation, and the training data contains label noise and other artifacts (see Fig. 3). In the experiments, we show that the LSTM networks are able to train on such noisy data, can be immediately applicable on the web. Around 600 attribute images per class were generated and were split by assigning 60% to the training set, 10% to the validation set, and 30% to the test set.

Scene data: Natural scene images contain a mix of mid-level attributes. We randomly collected the images from the web-search, and the only data cleaning we performed was to discard duplicated images in the results. This means that for each category, a wide range of representations of that category may occur. For example, we observed images that represented the category "stucco" as a building or a person applying the stucco on a building. In addition, web-images contain watermarks or logos or low-resolution image which were included in our scene data. Finally, 540 scene images with multiple classes in each image were used for estimating scene attributes.

3.2. Experimental setup

We perform two separate experiments to evaluate and compare the performance of our approach. The first experiment evaluates the quality of mid-level attribute learning using the single attribute data. The second part considers a more realistic scenario; The attribute regions in a scene image are analyzed using the learned model, and the best tags are extracted using the pruning rules.

For training, 200 patches from an input image were randomly sampled with the size between 50 × 50 and 80 × 80. The constant number of patches on the different image resolution are used to show the robustness of the multi-patch based approach under the variety of image resolutions. The patch was then rotated at angles of 0°–360° and rescaled to 64 × 64. Both scale and rotation are with 1 pixel or 1° level increment.

For testing on scene images, 200 patches were contributed for final top-k prediction, and the threshold of the pruning rules, T_p, T_r, and N_p were selected empirically to 0.6, 0.4, and 10, respectively. We kept the same parameters for all experiments below.

Pre-training on ImageNet for the network initialization is often effective [12], but in these experiments, it was not used for either LSTM networks or ConvNets.

LSTM networks: For LSTM network training, we used the RNNLIB library.[2] For all experiments, the size of input block, hidden size, and learning rate were fixed in 5, 15, 1e−4, respectively.

Baseline experiments: We compare our performance to the various existing approaches: feature-based and filter based methods. Following a standard dense-SIFT and PHOW feature (a variant of dense-SIFT descriptors, extracted at multiple scales) [5], multiple encoding schemes, including bag-of-visual-words (BoVW) and Fisher Vector (FV) [31,34], were compared. We extracted the dense feature in each 3 steps, discretized them with k-means and Kd-tree (BoVW) or Gaussian mixture model (FV) vector quantization after PCA projection, and accumulated words into histograms with spatial pyramid encoding. 80 dimensions, 1024 words and 64 words were used for PCA, BoVW, and FV, respectively. The open library called VLFeat [37] has been used for all feature extraction and classification methods.

For the comparison with filter-based approaches, we used the best low-level features reported in [3]: co-occurrence and Gabor with color chromatic features [1,3,9]. Eight co-occurrence matrices corresponding to one-pixel displacements along the following eight directions: $\{0^\circ, 45^\circ, \ldots, 315^\circ\}$. Five statistical features (contrast, correlation, energy, entropy, and homogeneity) were extracted in each direction, and averaged for rotation invariance. These features were normalized between 0 and 1. For Gabor filter, a bank of filters with the following parameters was used in the experiment: number of frequencies = 4, number of orientations = 6, maximum frequency = 0.327, frequency ratio = half-octave. All parameters were set based on the work from [3]. All features were extracted in HSV color space and SVM with Chi Squared kernel of period 2 was used as a classifier. All above parameters were optimized empirically.

Convolutional Neural Networks (ConvNet): Convolutional neural networks were implemented using the Caffe library [18]. Hyper parameter search was carried out based on the work of Krizhevsky [22]. The optimal structure identified by this process was five convolutional and two fully-connected layers with half the size of dimensions of Krizhevsky's network architecture. We also observed that using fewer than five layers resulted in significant decreases in recognition performance, e.g., 95.79% with five convolutional and two fully-connected layers, and 90.09% with four convolutional and one fully-connected layers.

3.3. Results

The accuracy comparisons of all experiments (visual attribute classification and scene analysis) are summarized in Table 1.

[2] http://sourceforge.net/projects/rnnl/.

Table 1
Accuracy comparisons of all experiments (the best score shown in bold). In order to compare the performance, all of our experiments have been following the same experimental setup. Visual attribute classification (single attribute classification): The best accuracy of our approach lead to superior performance compared to other common approaches. (No. test samples = 2352, 95% confidence interval = ±0.20). *Natural scene analysis:* multi-attribute classification; since we cannot directly make one decision of a classifier for scene analysis (multiple visual attributes), top-k ranked list was used to decide the high probable visual attribute classes of a scene. The visual attributes were listed based on its ranking score in descending order, and top-3 and top-5 accuracies were considered to compare the performance of our approach with other methods (no. test samples = 540, 95% confidence interval = ±0.42).

Method	Visual attribute	Scene analysis			# weights for NNs
	accuracy (%)	Top-1 (%)	Top-3 (%)	Top-5 (%)	
C-PHOW-BoVW, SVM [5]	59.86	8.56	31.91	48.84	–
C-PHOW-FV, SVM [34]	68.28	8.38	35.12	53.30	–
C-DSIFT-FV, SVM	72.66	9.09	32.98	50.62	–
C-Gabor, SVM [3]	62.12	42.34	70.88	81.03	–
C-Co-occurrence, SVM [3]	75.21	46.17	74.52	82.38	–
ConvNet [22]	95.79	56.81	79.85	90.21	38,802,300
LSTM networks	**97.32**	**71.43**	**84.23**	**94.25**	**100,272**
C-: HSV color space					

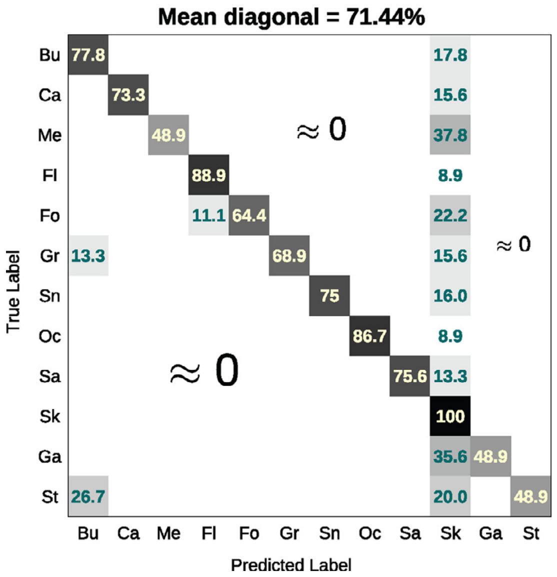


Fig. 4. Confusion table with top-1 predicted semantic attribute on scene images. The column is the top-1 predicted semantic attribute and the row is the ground truth label (labels from top to bottom (column) and left to right (row): building (Bu), candy (Ca), meat (Me), flower (Fl), forest (Fo), grass (Gr), snow (Sn), ocean (Oc), sand (Sa), sky (Sk), gravel (Ga), stucco (St)).

Visual attribute classification: The performance shows that our approach outperforms on complex attribute data. As pointed out by Mikolajczyk and Schmid [28], many popular methods were limited to the specific types of texture features. For instance, PHOW-BoW performed well only on repetitive-textures and some structured-textures (e.g., gravel and building). In addition, Co-occurrence or Gabor feature discriminates well on limited color-texture image datasets [3], but suffers from various attribute types and its diversity. Especially with low-textured categories, whichever feature detector or descriptor was chosen, it failed in providing sufficient evidence of textural information. However, the results from the 2D LSTM networks looked promising under various attribute types (including the low-textured attribute) and transformations (including huge distortions) without hand designed features.

Automatic web-image tagging: We examined the quality of scene analysis with top-k ranked list. It predicts the top-k most relevant attributes — k is the maximum attributes (tags) in the scene to be predicted, and the ranking score indicates the highly probable attributes. The confusion table (k as one) is shown in Fig. 4; The major portion of keywords was predicted as top-1. Our scene image includes multiple attributes and the major portion is not always the keyword — the weakness of the web image search engine. Therefore, a considerable number of images in each class were predicted as sky, and it shows that our system can potentially improve the image search engine. Especially, stucco was predicted as building in some images, since stucco could be a part of the building depending on scale. In the case of stucco, which is the part of a building, building-like images were retrieved from the query stucco, but our top-1 results predicted them as building (see the last middle column in Fig. 5; keyword: stucco, tagging result: building, sky, and gravel).

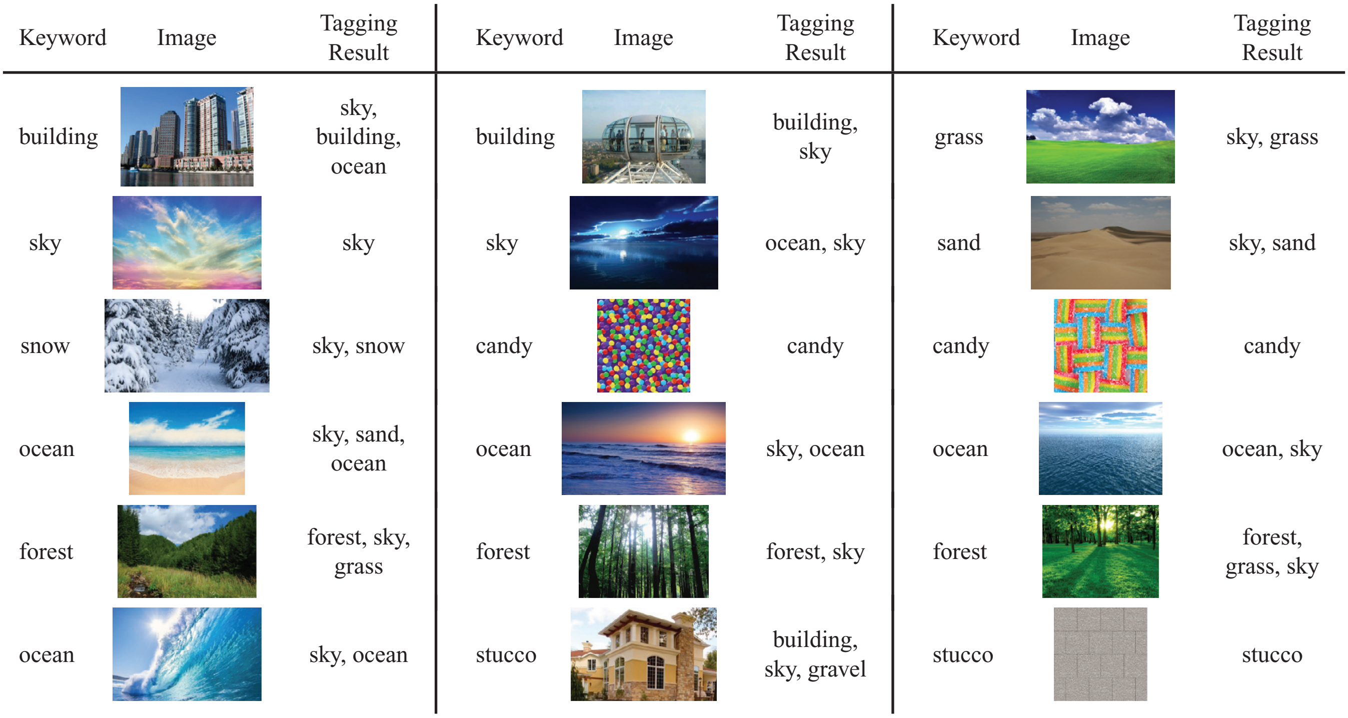
The LSTM networks outperformed all other approaches. From our observation, around one to five semantic classes are contained in a scene image (mostly up to three). Therefore, top-5 list is most likely able to provide all semantic parts. For performance evaluation, we tested with top-1, top-3, and top-5 results to compare with its keyword (the keyword was considered as a ground-truth label). In addition, we presented multi-tagging results of each image. Fig. 5 shows examples of multi-tagging results, which were correctly predicted by using their associated top-ranked list.

3.4. Experiments on outdoor scene attributes dataset

We also evaluated our approach on the public scene dataset (SceneAtt) proposed by Wang et al. [39]. We selected this dataset since this study is the most similar to our work. As reported in [39], most of the public scene datasets focused on the specific objects, humans or the functional activities, in contrast, our goal is to analyze the all visible contents of the scene. Furthermore, this dataset contains precise text descriptions with weak labels (not precise) which make the experiment more complex and realistic.

The dataset was collected from LMO [26], SUN attribute dataset [30], Google images, and Flickr. It consists of 1226 images of 256 × 256 pixels and 30 noun + adjective attribute pairs. The dataset was split into 645 images for training and the rest for testing. ConvNet and the LSTM networks were trained on 635 training images with the same parameter setting as our web-scene image analysis. Note that, we did not use the same model as the previous experiments. This experiment was intended as a harder test case because of more attributes, different descriptions, and small training samples. Thus, a new LSTM model was trained and evaluated using the same training and test data as Wang's work [39].

The mean average precision (MAP) is reported in Table 2. We compared the methods reported by Wang et al. [39] and our trained model using ConvNet and the LSTM networks. The best method [39], HST-att learns the spatial layout and attribute association by scene's



Keyword	Image	Tagging Result	Keyword	Image	Tagging Result	Keyword	Image	Tagging Result
building		sky, building, ocean	building		building, sky	grass		sky, grass
sky		sky	sky		ocean, sky	sand		sky, sand
snow		sky, snow	candy		candy	candy		candy
ocean		sky, sand, ocean	ocean		sky, ocean	ocean		ocean, sky
forest		forest, sky, grass	forest		forest, sky	forest		forest, grass, sky
ocean		sky, ocean	stucco		building, sky, gravel	stucco		stucco

Fig. 5. The results of automatic web-image tagging. The left side of each image indicates the keyword of image, and the right side of the image shows our tagging results. Top-3 after the patch pruning was resulted as tags of each image (the order of the list shows higher rank). As can be seen from tagging result, relevant attributes were well-detected in each scene. The system can even improve the retrieval system. For instance, the wrongly retrieved image from a web-search engine (e.g., the left bottom image — keywords: stucco, correct semantic attributes: building, sky) can be corrected by our tagging system.

Table 2
The comparison of Mean Average Precision (mAP) on SceneAtt dataset.

Method	MAP (%)
eKernel + SVM [41]	64.48
BoW + SPM [24]	53.11
HST − geo [40]	51.67
HST − att [39]	67.58
ConvNet [22]	63.24
LSTM networks	**88.59**

appearance model. It then finds the most probable parse tree of the adjective and noun description. To train ConvNet, the same architecture as in the previous experiments was used without pre-training. Using the simple LSTM networks, MAP reached about 21% higher than HST-att and 25% higher than ConvNet. The performance of ConvNet can further be improved by using the pre-trained model on ImageNet as mentioned in Section 3.2, since the training data is scarce.

4. Conclusion

Our experimental results show that neural network approaches work well for mid-level attribute recognition compared to non-neural network methods, and that among neural networks, 2D LSTM approaches outperform deep convolutional neural networks. Our approach performs end-to-end mid-level attributes learning using 2D LSTM recurrent networks. The network is shallow and requires fewer parameters compared to many deep learning approaches. Moreover, the networks take raw RGB values without any task-specific features and pre-/post-processing from noisy data and can adapt well to a full range of situations in natural images. Thus, the learned attributes model was successfully applied to the two different types of image data: single attribute and natural scene data — randomly collected from the web. We compared the performance with feature-based and filter-based methods as well as ConvNet, and showed the robustness of our approach under complex natural scene images. We further evaluated our approach on a publicly available dataset (outdoor scene attribute dataset) and showed the significant performance gain. We at the end demonstrated the feasibility of a real-world application: automatic web-image tagging.

These experiments show the effectiveness and generality of our approach; our model can be easily combined with other applications, e.g., visual classification, segmentation (for object or scene), or scene parsing. Another possible direction can be to improve the quality of the image search by re-ranking the images, since our scene analysis system has resulted in better visually intuitive tags of scenes than the results of the image search engine. As shown in Fig. 5, the native performance of Google's image search from the keyword is not always accurate and satisfying. Our ranking score and corresponding label indicate the importance of the semantics of the scene which can be directly employed to improve image search engines.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patrec.2015.06.003.

References

[1] V. Arvis, C. Debain, M. Berducat, A. Benassi, Generalization of the cooccurrence matrix for colour images: application to colour texture classification, Image Anal. Stereol. 23 (1) (2011) 63–72.
[2] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.
[3] F. Bianconi, R. Harvey, P. Southam, A. Fernndez, Theoretical and experimental comparison of different approaches for color texture classification, J. Electron. Imaging 20 (4) (2011) 043006-1–043006-17.
[4] A. Blake, P. Kohli, C. Rother, Markov Random Fields for Vision and Image Processing, Mit Press, 2011.
[5] A. Bosch, A. Zisserman, X. Muoz, Image classification using random forests and ferns, in: IEEE 11th International Conference on Computer Vision, ICCV, 2007, pp. 1–8.

[6] W. Byeon, T.M. Breuel, Supervised texture segmentation using 2d LSTM networks, in: IEEE International Conference on Image Processing, ICIP, 2014, pp. 4373–4377.

[7] W. Byeon, M. Liwicki, T. Breuel, Texture classification using 2d LSTM networks, in: 22nd International Conference on Pattern Recognition, ICPR, 2014, pp. 1144–1149.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on statistical learning in computer vision, ECCV, vol. 1, 2004, pp. 1–2.

[9] A. Drimbarean, P.F. Whelan, Experiments in colour texture analysis, Pattern Recognit. Lett. 22 (10) (2001) 1161–1167.

[10] A. Farhadi, I. Endres, D. Hoiem, D. Forsyth, Describing objects by their attributes, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2009, pp. 1778–1785.

[11] V. Ferrari, A. Zisserman, Learning visual attributes, in: Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, 3–6 December 2007, Vancouver, British Columbia, Canada, 2007.

[12] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2014, pp. 580–587.

[13] S. Gould, R. Fulton, D. Koller, Decomposing a scene into geometric and semantically consistent regions, in: IEEE 12th International Conference on Computer Vision, 2009, pp. 1–8.

[14] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, 385, Springer, 2012.

[15] A. Graves, S. Fernández, J. Schmidhuber, Multi-dimensional recurrent neural networks, in: Artificial Neural Networks, ICANN, Springer, 2007, pp. 549–558.

[16] A. Graves, J. Schmidhuber, Offline handwriting recognition with multidimensional recurrent neural networks, in: Advances in Neural Information Processing Systems, 2008, pp. 545–552.

[17] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[18] Y. Jia, Caffe: an open source convolutional architecture for fast feature embedding, 2013 http://caffe.berkeleyvision.org/.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: CVPR, 2014.

[20] K. Kavukcuoglu, P. Sermanet, Y. lan Boureau, K. Gregor, M. Mathieu, Y.L. Cun, Learning convolutional feature hierarchies for visual recognition, in: J. Lafferty, C. Williams, J. Shawe-taylor, R. Zemel, A. Culotta (Eds.), Advances in Neural Information Processing Systems, vol. 23, 2010, pp. 1090–1098.

[21] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images, Computer Science Department, University of Toronto, Tech. Report 1 (4) (2009) 7.

[22] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: NIPS, vol. 1, 2012, p. 4.

[23] Ladick, P. Sturgess, K. Alahari, C. Russell, P. Torr, What, where and how many? combining object detectors and CRFS, in: K. Daniilidis, P. Maragos, N. Paragios (Eds.), Computer Vision ECCV 2010, Lecture Notes in Computer Science, vol. 6314, Springer, Berlin, Heidelberg, 2010, pp. 424–437.

[24] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2006, pp. 2169–2178.

[25] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[26] C. Liu, J. Yuen, A. Torralba, Nonparametric scene parsing: label transfer via dense scene alignment, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2009, pp. 1972–1979.

[27] L. Liu, P.W. Fieguth, Texture classification from random features, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 574–586.

[28] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 27 (10) (2005) 1615–1630.

[29] R. Mittelman, H. Lee, B. Kuipers, S. Savarese, Weakly supervised learning of mid-level features with beta-Bernoulli process restricted Boltzmann machines, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2013, pp. 476–483.

[30] G. Patterson, J. Hays, Sun attribute database: discovering, annotating, and recognizing scene attributes, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2012, pp. 2751–2758.

[31] F. Perronnin, C. Dance, Fisher kernels on visual vocabularies for image categorization, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2007, pp. 1–8.

[32] P.M. Roth, M. Winter, Survey of appearance-based methods for object recognition, Institute for Computer Graphics and Vision, Graz University of Technology, Austria, Technical Report ICGTR0108 (ICG-TR-01/08) (2008).

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. (IJCV) (2015) 1–42, doi:10.1007/s11263-015-0816-y.

[34] J. Sánchez, F. Perronnin, High-dimensional signature compression for large-scale image classification, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2011, pp. 1665–1672.

[35] Y. Sun, Y. Chen, X. Wang, X. Tang, Deep learning face representation by joint identification-verification, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, vol. 27, Curran Associates, Inc., 2014, pp. 1988–1996.

[36] M. Varma, A. Zisserman, Classifying images of materials: achieving viewpoint and illumination independence, in: Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark, 3, Springer-Verlag, 2002, pp. 255–271.

[37] A. Vedaldi, B. Fulkerson, Vlfeat: an open and portable library of computer vision algorithms, in: Proceedings of the International Conference on Multimedia, ACM, 2010, pp. 1469–1472.

[38] A. Vedaldi, V. Gulshan, M. Varma, A. Zisserman, Multiple kernels for object detection, in: IEEE 12th International Conference on Computer Vision, 2009, pp. 606–613.

[39] S. Wang, J. Joo, Y. Wang, S.-C. Zhu, Weakly supervised learning for attribute localization in outdoor scenes, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2013, pp. 3111–3118.

[40] S. Wang, Y. Wang, S.-C. Zhu, Hierarchical space tiling for scene modeling, in: Computer Vision–ACCV 2012, Springer, 2013, pp. 796–810.

[41] J. Xiao, J. Hays, K. Ehinger, A. Oliva, A. Torralba, Sun database: large-scale scene recognition from abbey to zoo, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2010, pp. 3485–3492.