

Sunrise or Sunset: Selective Comparison Learning for Subtle Attribute Recognition

Hong-Yu Zhou
zhouhy@lamda.nju.edu.cn

Bin-Bin Gao
gaobb@lamda.nju.edu.cn

Jianxin Wu
wujx2001@nju.edu.cn

National Key Laboratory for Novel
Software Technology,
Nanjing University,
Nanjing, China

Abstract

The difficulty of image recognition has gradually increased from general category recognition to fine-grained recognition and to the recognition of some subtle attributes such as temperature and geolocation. In this paper, we try to focus on the classification between sunrise and sunset and hope to give a hint about how to tell the difference in subtle attributes. Sunrise vs. sunset is a difficult recognition task, which is challenging even for humans. Towards understanding this new problem, we first collect a new dataset made up of over one hundred webcams from different places. Since existing algorithmic methods have poor accuracy, we propose a new pairwise learning strategy to learn features from selective pairs of images. Experiments show that our approach surpasses baseline methods by a large margin and achieves better results even compared with humans. We also apply our approach to existing subtle attribute recognition problems, such as temperature estimation, and achieve state-of-the-art results.

1 Introduction

Recognition has been one of the central tasks in computer vision and pattern recognition, and machine learning (especially deep learning) has been one of the key forces in developing various image recognition methods and systems. The success of deep ConvNet (CNN), e.g., AlexNet [1], has greatly advanced the state of the art in image recognition.

What is to be recognized in an image? The answer to this question has been constantly changing, which in consequence leads to different image recognition problems. The vision community has been recognizing properties of images with increasing difficulties: semantic image categories, fine-grained image categories and some physical attributes (such as the temperature [2]). Humans are good at recognizing semantic categories, while fine-grained recognition needs domain experts (e.g., bird specialists). For physical attributes, it is easy to tell whether a photo was taken at day or night while even an experienced traveler feels difficult to tell the exact place in the photo.

In this paper, these seemingly indistinguishable attributes (even difficult for human beings to correctly infer from an image) are classified as *subtle* attributes, and we argue that it is possible to recognize this kind of attribute from images. In Figure 1, we compare subtle

	Transient	Subtle
		
Is it taken at night?	✓	✗
Which month?	✗	✓
Where?	✗	✓
Does it feel warm?	✓	✗
What is the temperature?	✗	✓
Is it cloudy?	✓	✗
Morning or afternoon?	✗	✓
Which Season?	✓	✗

Figure 1: A comparison between subtle attributes and transient attributes [13]. Compared with transient attribute, subtle attribute can be more detailed.



(a) Monet's paintings

(b) Guess which is a sunrise?

Figure 2: Sunrise vs. Sunset: (a) Two famous paintings of Claude Monet. Left: "Impression, Sunrise"; Right: "Sunset in Venice". Notice the difference of color and luminance in these paintings. (b) Two photos taken by the same camera at the same location. One for sunrise and one for sunset. Guess which one is taken at sunrise? (answer below²) Best viewed in color.

attributes with transient attributes proposed by Laffont *et al.* [13], which suggests that subtle attribute recognition can be harder to some extent. For simplicity, we use *sunrise vs. sunset*, a very difficult classification task, as an example of subtle attribute recognition and build a "Sunrise or Sunset" (SoS) dataset.

For most people, we often marvel at the beauty of a sunrise or a sunset. In photography, the period of time witnessing sunrise or sunset is usually called the *golden hour* because sunrises and sunsets produce impressive photos. A probably less examined question is: What is the difference between sunrise and sunset when recorded in photos? In fact, many artists have noticed the difference and tried to recreate the difference with artistic expressions. For instance, as the founder of French Impressionist painting, Monet has composed many great paintings about sunrise and sunset by utilizing different color and luminance (cf. Figure 2).

Although it might be easy to distinguish sunrise from sunset in paintings (Figure 2(a)), it is difficult in real world photographs (Figure 2(b)). This issue (sunrise vs. sunset) has been discussed for many years, but we still lack definitive conclusions. Some atmospheric physicists have performed considerable research cataloging and documenting the colors at both time periods. They draw a conclusion that one cannot tell the difference from the color distribution [18]. However, it is possible that the transition of colors (e.g., the direction along which the luminance fades) might be helpful. There are also some other factors that might

²Answer: the photo in the right is taken at sunrise.

affect our decision, such as lighting and shading. Human activities are also useful clues. For example, if we observe in a photo lots of cars on the road, this observation may indicate that it is more likely to be a sunset.

On the SoS dataset we have built, the accuracy of sunrise vs. sunset recognition is around 50.0% (i.e., random guess) in a group of volunteers (who are graduate students). However, after some simple training and learning, the human accuracy is improved to 64.0%, much higher than the original performance. This fact supports that the sunrise vs. sunset distinction is (at least partially) viable. Commonly used image recognition methods (e.g., Bags of Visual Words [14] or CNN), however, are much worse in this task: their accuracies are between 51.7% and 56.3%. In other words, they are only slightly better than a random guess.

In this paper, we develop a new pairwise learning strategy inspired by the different performance of human’s sunrise vs. sunset recognition in various experimental setups, and then introduce a deep architecture with selective pairs of input images. Experimental results show that the proposed method (SoSNet) significantly outperforms other methods. To train and evaluate our model, we also introduce SoS, a benchmark database for discriminating sunrise from sunset images. In addition to solving the sunrise vs. sunset problem, the proposed approach is then applied to estimate the temperature given a photo, which achieved state-of-the-art experimental results too. These results suggest that our SoSNet has the potential to solve more subtle attribute recognition problems.

2 Related Work

The community has long been interested in general image recognition, i.e., recognizing objects or scenes in different semantic categories, e.g., distinguishing objects such as birds, dogs, bicycles and airplanes, or scenes such as cinema, clinic or beach. These tasks are quite easy for a human being, but not necessarily as easy for a computer. Along with the availability of larger and larger datasets (e.g., SUN [28], ImageNet [9]), various visual representations and learning methods have been proposed to deal with general image recognition, including manually designed visual features (e.g., SIFT [17], SURF [2]) and the recently popularized deep convolutional neural network learning methods. The gap between humans and computers on recognizing general objects and scenes is becoming smaller. For instance, He *et al.* [8] claimed that they surpass human-level performance on the ImageNet dataset for the first time. Several recent works [6] [29] also reported comparable results on the SUN 397 scene recognition dataset [28].

Fine-grained image recognition is a more difficult recognition task, in which the objects in different categories look alike each other. Fine-grained objects in different categories only differ in some details. The CUB200-2011 dataset [26] has 200 categories, one for each bird species. Fine-grained details are required to distinguish birds in different categories. Fine-grained recognition requires fine-grained visual representation, e.g., by finding the parts of a bird and extracting features from them rather than extracting a visual representation from the whole bird globally. Part-based methods [32], [16] are widely used in fine-grained recognition. Zhang *et al.* [32] proposed a part-based R-CNN model by utilizing part detectors. Lin *et al.* [16] employed a two-stream deep convolutional model to form a bilinear model which is able to produce more discriminative features.

Recently, researchers also move to recognize some more difficult image attributes. Glasner *et al.* [5] proposed a scene-specific temperature prediction algorithm that could turn a camera into a crude temperature sensor. They also made an investigation about the rela-

tion between certain regions of the image and the temperature. Weyand *et al.* [27] proposed PlaNet, a deep model to use and integrate multiple visual cues to solve the photo geolocation problem. Their deep model outperforms previous approaches and even attains superhuman levels of accuracy in some cases.

Learning from pairwise input images is natural in many computer vision tasks, such as face verification [8], person re-identification [10], domain adaptation [22] and image similarity learning [61]. A siamese (two streams sharing the same parameters) network is the standard solution in these researches. Parikh and Grauman [19] proposed that binary attributes are an artificially restrictive way to describe images and then employed a rank function to describe some relative visual properties. Yu and Grauman [50] used the local learning approach to resolve the problem of fine-grained visual comparisons.

The proposed subtle attribute recognition can be seen as a fine-grained version of traditional attribute recognition and is more difficult than fine-grained image classification. We also build a SoSNet which modifies the siamese network with a ranking loss to learn representations from selective comparison.

3 Building the SoS Dataset

In this section we propose to build a SoS dataset to help us study the problem. We divide the collecting process into several stages, and individual stages are discussed in detail in the following paragraphs.

Stage 1. Downloading images from websites. Modern image databases are often collected via image search engines and image hosting websites (e.g., Flickr). We cannot collect images through these channels because we cannot be sure whether those photos have been postprocessed, and their labels might be incorrect. For instance, many pictures in Flickr are labeled with both sunrise and sunset, and most of them are likely edited. We downloaded images from the AMOS (Archive of Many Outdoor Scenes) dataset [11], which contains images from nearly 30,000 webcams. For convenience, we first downloaded data between 2010 and 2016 from over 1,000 webcams and then extracted 4 images per month for each webcam.

Stage 2. Improving data quality. To get exact camera locations, we only used those cameras whose uploaders attached fairly precise locations along with them. Besides, some cameras were broken while significant parts of them provided low resolution images. We solved this problem by manually getting rid of broken cameras: once the webcam has one low-quality image, we delete the camera at once.

Stage 3. Extracting locations of cameras and calculating local sunrise/sunset time. Note our intention is to retrieve images taken at correct *local* sunrise and sunset time in order to guarantee the correctness of the whole dataset. We used IP address of each webcam to get its location, and directly employed the algorithm published by the Nautical Almanac Office to calculate local sunrise/sunset time.³ Images captured at sunrise/sunset time were split into two classes.

Finally, we obtain 12,970 images from 128 webcams over 30 countries with good quality. Note that sunrise and sunset images occupy exactly 50% of the images in our dataset, respectively. Two sets of tasks are defined for SoS: an *easy task* and a *difficult task*. We randomly split the images into 10,448 training and 2,522 testing images in the *easy task*. For

³http://williams.best.vwh.net/sunrise_sunset_algorithm.htm

Table 1: Accuracy of humans and fine-tuned VGG-16 (pretrained on ImageNet) on the SoS dataset (one random image as input). mAcc means the average accuracy(%).

Method	Task	Sunrise	Sunset	mAcc
FT-VGG-16	Easy	79.6	80.0	79.8
FT-VGG-16	Hard	53.3	52.9	53.1
Human	Hard	64.3	63.7	64.0

the *difficult task*, we use images from 104 webcams (10,448 images) for training, and the images from the rest 24 cameras for testing.

4 Selective Comparison Learning

We first study the performance of existing deep models and humans on the SoS dataset to assess its difficulty. The experiments that involve human also give us some hints on how to design a deep learning model to distinguish sunrise from sunset.

4.1 Difficulty of the SoS dataset

We first fine-tune the VGG-16 CNN model for the easy and difficult tasks, whose results are reported in Table 1. These results suggest that the fine-tuned VGG-16 CNN model has good accuracy in the easy task. In the difficult task, the accuracy (53.1%) is close to that of a random guess. The comparison between the easy and the difficult tasks suggests that *a vanilla convolutional neural network might pay attention to the pixel intensities rather than the subtle attributes*. Hence, we need a different approach for the sunrise vs. sunset classification problem.

We also informally studied how good humans are for the difficult task. 20 volunteers achieve roughly 52.0% accuracy in this task when tested without preparation. After they read the answers on Quora for sunrise vs. sunset techniques, 100 images were shown to the volunteers who are required to give their predictions. In this case, the mean accuracy goes up to 64.0%.

A few interesting observations emerge in Table 1. Compared with hard task, VGG-16 performs much better on the easy task because the model might have memorized historical data in each webcam. On the other side, even when presented with randomly chosen single image, humans can outperform the VGG-16 model by a large margin (64.0% vs. 53.1%), which suggests that humans have the ability to learn to distinguish subtle attributes to some extent. Unless specified otherwise, the rest experiments are performed on the hard task which is a more realistic problem.

The poor performance of VGG-16 show that a single model might not be able to focus on the right parts in the input image. During human tests, we are surprised to find that given a pair of input images, especially when volunteers have been told that the pair has similar properties (e.g., the same place), they achieve much better performance. To help the recognition process, we studied the influence of different constraints, e.g., if the paired images are captured at the same day or the same place. We give 5 different constraints in Table 2 and their test results.

As we can see from Table 2, humans perform gradually better as more constraints are added. The pure pair gets the worst result because there is too much noise. While the best

Table 2: Give a pair of input images, the performance of humans on the SoS dataset. Note that each volunteer is shown 5 groups of paired images corresponding to 5 different settings, and each group contains 50 pairs. **SS** is another restriction on each pair which requires that one image is sunrise, the other is sunset.

Pair Constraint	SS	Sunrise	Sunset	mAcc
Random pair	w/o	65.3	64.9	65.1
Random pair	w	67.3	66.9	67.1
The same day	w	67.8	66.8	67.3
The same location	w	70.7	70.0	70.3
The same location and day	w	72.4	72.4	72.3

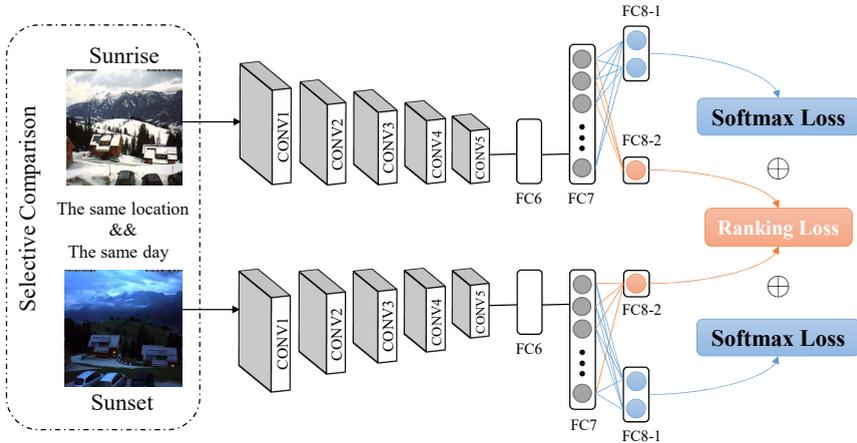


Figure 3: SoSNet for learning representations and prediction. This model uses the selective pairs to learn visual representations for subtle attribute recognition.

result requires that both photos contain sunrise and sunset and should be taken at the same location and the same day. This phenomenon helps us to design an effective model to learn good representations.

4.2 Architecture and loss function

Inspired by the observation, we design a two-stream network which uses a selective pair of images as its input. The two streams share the same parameters, and we denote the computation in each stream as $f_{\theta}(\cdot)$ (in which θ denotes the parameters in the two-stream model). More details about the architecture are shown in Figure 3.

Let us denote the two images in the pair as X_R and X_S , for sunrise and sunset images taken by the same camera in the same day, respectively. Specifically, we use $f_{\theta_1}(\cdot)$ and $f_{\theta_2}(\cdot)$ to represent the output of layer FC8-1 and layer FC8-2 in Figure 3, respectively. The final loss function can be regarded as a combination of softmax loss and ranking loss,

$$\ell(X_R^n, X_S^n) = \frac{1}{N} \sum_{n=1}^N (\ell_{softmax}(f_{\theta_1}(X_R^n), y_R^n) + \ell_{softmax}(f_{\theta_1}(X_S^n), y_S^n) + \lambda \ell_{ranking}(f_{\theta_2}(X_R^n), f_{\theta_2}(X_S^n))), \quad (1)$$

in which

$$\ell_{\text{ranking}}(f_{\theta_2}(X_R^n), f_{\theta_2}(X_S^n)) = \frac{1}{1 + \exp(f_{\theta_2}(X_R^n) - f_{\theta_2}(X_S^n))}, \quad (2)$$

where ℓ_{softmax} is softmax loss, N is the number of pairs, $\{y_R^n, y_S^n\}$ ($n = 1, 2, \dots, N$) represent image-level labels and λ is a balancing parameter. This loss function is differentiable with respect to its parameters.

During training, we use stochastic gradient descent (SGD) to minimize the average loss of all training pairs. Because we minimize the average loss between same-location same-day pairs, the pair of images X_R and X_S must be similar to each other in terms of the contents in them. Minimizing the ranking loss amounts to maximizing the difference between the features extracted from the pair of images using the same nonlinear mapping $f_{\theta}(\cdot)$. Hence, $f_{\theta}(\cdot)$ is forced to learn those subtle differences between the pair of images. At the same time, the softmax loss keeps the prediction correct when the input is a single image. This pairwise learning strategy discourages $f_{\theta}(\cdot)$ to capture patterns such as texture, scene or object (which vary a lot in different images).

4.3 Implementation details

We implemented the network in MatConvNet [24]. Both streams are initialized based on VGG-16 (pretrained on ImageNet but the last fully connected layer is removed). Then, we add three fully connected layers: FC7 ($1 \times 1 \times 4096 \times 256$), FC8-1 ($1 \times 1 \times 256 \times 2$) and FC8-2 ($1 \times 1 \times 256 \times 1$) to form a single stream. The balancing parameter λ is set to 1. The batch size is 16 pairs, which means each batch actually contains 32 images. We did not use dropout or batch normalization during the training. The learning rate is 10^{-3} and gradually reduces to 10^{-5} using logspace in 20 epochs. In the test stage, we use one stream to predict the label of each input image.

5 Experiments

5.1 Experimental setup

In this section, we compare the accuracy of different methods on the difficult task of the SoS dataset, which is a subtle attribute recognition task. We compare the proposed approach with several baselines, ranging from classical image classification methods to recent deep convolutional models

To demonstrate the effectiveness of SoSNet, we firstly consider two related methods as baselines: **Single-stream ConvNet** and **SoSNet with random pair (SoSNet-rand)**. The former one uses AlexNet [25], Network in Network (NiN) [26], VGG-16 [27] net, and recent ResNet [9]. All models are pretrained on ImageNet. The latter baseline replaces the selective pair in SoSNet with random pair and hence make use of contrast loss [28] instead of ranking loss. In addition, we also compare SoSNet with the following baseline methods:

- **Hand-crafted features + SVM.** The first family of baseline methods uses the SIFT features [29] implemented in VLFeat [23], which are computed at seven scales with a factor $\sqrt{2}$ between successive scales, with a stride of 4 pixels. An input image returns a set of SIFT feature vectors. We use three encoding methods to turn the set of SIFT features into one vector, including Fisher Vectors (FV) [20], Bags of Visual Words (BOVW) [24] and Vector of Locally Aggregated Descriptors (VLAD) [20]. Experiments labeled with

Table 3: Evaluation of different methods on the hard task. All encoding methods are based on SIFT and followed by a linear SVM.

	$FV(s.p.)$	FV	$VLAD(aug.)$	$BOVW(aug.)$	$AlexNet$	NiN	$VGG-16$	$ResNet-101$	$SiameseNet$	$SoSNet-rand$	$SoSNet$
sunrise	54.1	53.4	50.6	56.6	52.2	52.1	53.3	53.8	54.2	58.6	70.9
sunset	52.1	54.0	52.8	56.0	52.0	52.5	52.9	53.2	54.8	59.0	71.6
mAcc	53.1	53.7	51.7	56.3	52.1	52.3	53.1	53.5	54.5	58.8	71.2

“aug.” encode spatial information; “s.p.” use a spatial pyramid with 1×1 and 3×3 subdivisions. After encoding feature vectors, a linear SVM is used as a classifier.

- **Siamese features + SVM (SiameseNet).** Siamese networks have the ability to learn good features from training a similarity metric from data. Here, we implement the contrast loss function [2] used in person re-identification [3] based on VGG-16 as comparison. Let X_i and X_j be two training images, c_i and c_j be the labels for X_i and X_j , respectively. $f_\theta(\cdot)$ is the deep model parameterized by θ , which maps one input image to an output vector. Let $v_i = f_\theta(X_i)$, $v_j = f_\theta(X_j)$. We consider the squared Euclidean distance in the loss,

$$\ell(X_i, X_j) = \begin{cases} \|v_i - v_j\|_2^2 & \text{if } c_i = c_j \\ \max(1 - \|v_i - v_j\|_2^2, 0) & \text{otherwise} \end{cases} \quad (3)$$

In this model, the dimensionality of v_i and v_j is 256. We extract the last layer from siamese network as features and use a linear SVM for classification.

5.2 Experimental results

The sunrise vs. sunset recognition results of these baselines and the proposed method are reported in Table 3. In Table 3, the proposed method achieves an accuracy of 71.5%, outperforming other baselines by at least 17%.

A few interesting points can be observed from Table 3.

BOVW has higher accuracy than single-stream ConvNets. Different from general or fine-grained image recognition, the bag of visual words models have achieved higher accuracy than single-stream ConvNets when recognizing sunrise/sunset. The drawbacks of the bag of visual words model, including small receptive field and low representational power when compared with CNN, might make it less overfit by the pixel values as the CNN.

SoSNet-rand performs better than SiameseNet. This phenomenon tells us that learning representations and predictions simultaneously (an end-to-end manner) might somehow facilitate the recognition process.

SoSNet exceeds SoSNet-rand by 10 points. Since the difference is only the pair constraint, the success suggests that selective comparison might be more useful than random pair in subtle attribute recognition.

Table 4: Temperature estimation results for each scene in Glasner’s dataset.

	R^2 (the higher the better)/ $RMSE$ (the lower the better)									
	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)
Local Regression [9]	0.67/6.85	0.65/7.24	0.70/6.03	0.59/4.53	0.76/5.77	0.38/3.19	0.50/7.63	0.77/5.09	0.10/3.68	0.59/7.77
LR Temporal Win. [9]	0.61/7.52	0.69/6.86	0.72/5.82	0.64/4.23	0.79/5.39	0.53/2.77	0.54/7.35	0.76/5.22	0.11/3.67	0.58/7.85
Global Ridge Reg. [9]	0.00/18.16	0.78/5.74	0.00/35.02	0.00/11.37	0.00/43.51	0.10/3.84	0.74/5.54	0.00/13.86	0.23/3.41	0.46/8.91
CNN [9]	0.49/8.55	0.79/5.59	0.71/5.96	0.24/6.17	0.61/7.30	0.48/2.90	0.39/8.48	0.79/4.88	0.43/2.93	0.66/7.12
Transient Attrib. [9]	0.36/9.60	0.70/6.69	0.58/7.20	0.55/4.75	0.68/6.62	0.21/3.59	0.58/7.03	0.65/6.31	0.16/3.56	0.67/7.00
FC6 [9]	0.52/8.28	0.80/5.46	0.61/6.89	0.56/4.72	0.80/5.30	0.21/3.60	0.54/7.34	0.79/4.90	0.06/3.78	0.59/7.80
Pool4 [9]	0.58/7.79	0.84/4.87	0.79/5.03	0.60/4.45	0.87/4.22	0.40/3.14	0.63/6.61	0.80/4.72	0.52/2.70	0.76/6.01
Ours	0.73/6.26	0.89/4.57	0.83/4.92	0.70/3.80	0.90/3.98	0.58/2.53	0.80/5.20	0.86/3.95	0.55/2.48	0.78/5.81

6 Ambient Temperature Prediction

Glasner *et al.* [9] first proposed to use simple scene specific temperature prediction algorithms to turn a camera into a crude temperature sensor. The data were also collected from the AMOS dataset [10], which mainly contains 10 different webcams. From each webcam they extracted one image every day at 11:00 am local time over a period of two consecutive years. The first year images were used for training, while the second year for testing. In this section, we follow the same experimental settings as those in *et al.* [9]. Ten cameras are referred to as scenes (a)-(j).

6.1 Baseline methods and evaluation metric

Glasner *et al.* [9] described 5 different estimation methods: Local Regression (LR), Local Regression with a Temporal Window (LRTW), Global Regularized Regression (GRR), CNN and Transient Image Attributes (TA). The first three use simple pixel intensities as features while the last two use more sophisticated global image features.

There are two protocols to evaluate the performance of aforementioned algorithms. The coefficient of determination (R^2) and Root Mean Squared Error (RMSE). Glasner *et al.* [9] used R^2 to compare results for different scenes, while RMSE provides an intuitive interpretation of the estimate’s quality. More details of these two protocols can be found in [9].

6.2 Temperature prediction with selective comparison

Unlike the experiments in sunrise/sunset, *the selective pairs are restricted to photos taken at the same place*. And we replace the softmax loss with square loss to predict the temperature. We train an independent SoSNet for each scene. The learning rate is 10^{-3} and gradually reduces to 10^{-5} using logspace in 10 epochs. Then we follow the same instructions in [15] which extracts Pool4 as features for a linear ν -SVR using LIBSVM. For fairness, we set the SVR parameters as $C = 100$, $\nu = 0.5$ and $g = 1$ for all our experiments which are the same as those in [9]. More details can be found in [9] and [15].

We report the performance of our approach and seven other algorithms on ten different scenes in Table 4. Our method achieves the best results on all 10 scenes. It is worth noting that our approach outperforms the traditional CNN by a large margin.

7 Conclusion

In this paper, we have proposed to study a new type of image recognition problem: to recognize subtle attributes. We built a SoS dataset, and our experiments showed that both humans and existing computer vision and machine learning methods have poor accuracy on this dataset. We proposed a model named SoSNet that learns discriminative features using selective pairs as inputs while is still able to make predictions. The same SoSNet also achieved state-of-the-art results in temperature prediction from an image.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61422203.

References

- [1] Ejaz Ahmed, Michael Jones, and Tim K. Marks. An improved deep learning architecture for person re-identification. In *CVPR*, pages 3908–3916, 2015.
- [2] Herbert Bay, Tinne Tuytelaars, and Van Gool Luc. SURF: Speeded up robust features. In *ECCV*, pages 404–417, 2006.
- [3] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages 539–546, 2015.
- [4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.
- [5] Daniel Glasner, Pascal Fua, Todd Zickler, and Lihi Zelnik-Manor. Hot or Not: Exploring correlations between appearance and temperature. In *ICCV*, pages 3997–4005, 2015.
- [6] Sheng Guo, Weilin Huang, and Yu Qiao. Locally-supervised deep hybrid model for scene recognition. In *arXiv preprint arXiv:1601.07576*, 2016.
- [7] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, pages 1735–1742, 2006.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *ICCV*, pages 1026–1034, 2015.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, pages 580–587, 2016.
- [10] Nathan Jacobs, Nathaniel Roman, and Robert Pless. Consistent temporal variations in many outdoor scenes. In *CVPR*, pages 1–6, 2007.
- [11] Herve Jegou, Matthijs Douze, Cordelia Schmid, and Patrick Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010.

- [12] Alex Krizhevsky, Ilya Sutskever, and G. E Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.
- [13] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient Attributes for High-Level Understanding and Editing of Outdoor Scenes. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 33(4), 2014.
- [14] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.
- [15] Min Lin, Qiang Chen, and Shuicheng Yan. Network in Network. In *ICLR*, 2014.
- [16] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *ICCV*, pages 1449–1457, 2015.
- [17] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [18] David K. Lynch and William C. Livingston. *Color and light in nature*. Cambridge university press, 2001.
- [19] Devi Parikh and Kristen Grauman. Relative attributes. In *ICCV*, pages 503–510, 2011.
- [20] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, pages 1–8, 2007.
- [21] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015.
- [22] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous Deep Transfer Across Domains and Tasks. In *ICCV*, pages 4068–4076, 2015.
- [23] Andrea Vedaldi and Brian Fulkerson. VLFeat – An open and portable library of computer vision algorithms. In *ACM MM*, pages 1469–1472, 2010.
- [24] Andrea Vedaldi and Karel Lenc. MatConvNet – Convolutional neural networks for MATLAB. In *ACM MM*, pages 689–692, 2015.
- [25] Anna Volokitin, Radu Timofte, and Luc Van Gool. Deep Features or Not: Temperature and Time Prediction in Outdoor Scenes. In *CVPRW*, 2016.
- [26] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. In *Tech. Report CNS-TR-2011-001*, 2011.
- [27] Tobias Weyand, Ilya Kostrikov, and James Philbin. PlaNet - Photo Geolocation with Convolutional Neural Networks. In *ECCV*, pages 37–55, 2016.
- [28] Jian X. Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, pages 3485–3492, 2010.
- [29] Guosen Xie, Xuyao Zhang, Shuicheng Yan, and Chenglin Liu. Hybrid CNN and dictionary-based models for scene recognition and domain adaptation. *TCSVT*, 2015.

- [30] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, pages 192–199, 2014.
- [31] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, pages 4353–4361, 2015.
- [32] Ning Zhang, Jeff Donahue, Ross Girshick, and Trevor Darrell. Part-based R-CNNs for fine-grained category detection. In *ECCV*, pages 834–849, 2014.