Sharing ConvNet Across Heterogeneous Tasks

Takumi Kobayashi(✉)

National Institute of Advanced Industrial Science and Technology,
Umezono 1-1-1, Tsukuba, Ibaraki, Japan
takumi.kobayashi@aist.go.jp

Abstract. Deep convolutional neural network (ConvNet) is one of the most promising approaches to produce state-of-the-art performance on image recognition. The ConvNet exhibits excellent performance on the task of the training target as well as favorable transferability to the other datasets/tasks. It, however, is still dependent on the characteristics of the training dataset and thus deteriorates performance on the other types of task, such as by transferring the ConvNet pre-trained on ImageNet from object classification to scene classification. In this paper, we propose a method to improve generalization performance of ConvNets. In the proposed method, the ConvNet layers are partially shared across heterogeneous tasks (datasets) in end-to-end learning, while the remaining layers are tailored to respective datasets. The method provides models of various generality and specialty by controlling the degree of shared layers, which are effectively trained by introducing the diversity into mini-batches. It is also applicable to fine-tuning the ConvNet especially on a smaller-scale dataset. The experimental results on image classification using ImageNet and Places-365 datasets show that our method improves performance on those datasets as well as provides the pre-trained ConvNet of higher generalization power with favorable transferability.

1 Introduction

Image recognition performance has been significantly improved by deep convolutional neural network (ConvNet) [1,2] in the framework of deep learning; it is applied with great success to such as object detection [3] and tracking [4]. The deep ConvNet stacks many convolution layers in order to extract image features of diverse levels and a huge number of parameters contained in those layers are trained in an end-to-end manner through back-propagation. The problem of over-fitting is remedied by leveraging large-scale annotated data [5,6] and some techniques such as rectified linear unit (ReLU) [7], DropOut [8] and BatchNormalization [9].

The so-trained Deep ConvNets exhibit excellent classification performance on the dataset/task of the training target, while being effectively transferable to the other datasets and tasks [10–12]. For example, the ConvNet pre-trained on ImageNet [5] can be applied as an image feature extractor to various image recognition tasks on which hand-crafted features [13,14] have effectively worked; the pre-trained (off-the-shelf) ConvNets produce state-of-the art performance

D. Liu et al. (Eds.): ICONIP 2017, Part II, LNCS 10635, pp. 343–353, 2017.
https://doi.org/10.1007/978-3-319-70096-0_36

on various datasets of even middle scale [10–12]. In the pre-trained ConvNets, however, we can find some dependency on the characteristics of the training dataset. As discussed in [10], the ConvNet pre-trained on ImageNet [5] works well for tasks related to object classification (ImageNet task), but it degrades performance on scene classification tasks which are far from the targets of ImageNet, and vice versa [6]. Thus, for effectively applying those ConvNets as feature extractors, it is required to carefully consider the type of target tasks in advance.

In this paper, we propose a method to improve generalization performance of ConvNets. The proposed method allows the ConvNet to be trained on heterogeneous datasets (tasks) in an end-to-end manner, while it has been usually learned on a single (homogeneous) dataset such as either of ImageNet [5] or Places-365 [6]. Our approach is close to the hybrid method in [6] which trains a single ConvNet on the union of those two datasets through simply concatenating their label sets. In contrast to [6], the proposed method deals with the label sets separately while sharing the network *partially* across heterogeneous datasets. Thus, it can provide various models of different generality and specialty by controlling the degree of the shared network components. Thereby, the method produces the ConvNet improving performance on the task of the training target as well as the one exhibiting better generalization performance with high transferability to (other) various tasks. The multitask learning (MTL) [15,16] is also related to our work in that the network components are shared across several datasets in training. The MTL, however, considers only the *related* (homogeneous) tasks, thereby deteriorating performance on heterogeneous ones. In this work, we effectively treat the heterogeneous tasks by taking into account their diversity in mini-batch construction. And, while it has not been clearly discussed how many network components should be shared, we thoroughly investigate the degree of the shared components in terms of classification performance. Furthermore, we also present an effective approach toward fine-tuning in our framework.

2 Sharing ConvNet

In [6,10], the generality or transferability of the ConvNets is improved by concatenating the ConvNets or the datasets. Let $\mathcal{F}$ indicate the ConvNet architecture, *e.g.*, AlexNet [1], with the parameters denoted by $\boldsymbol{\theta}$. Suppose we have D datasets $\{\mathcal{D}_d\}_{d=1}^{D}$, *e.g.*, $\{\mathcal{D}_d\}_{d=1}^{2} = \{\mathsf{ImageNet}, \mathsf{Places\text{-}365}\}$, each of which contains pairs of image I and its class label y. The ConvNet is usually trained on respective datasets by

$$\left\{ \min_{\boldsymbol{\theta}_d} \sum_{(I,y)\in\mathcal{D}_d} \mathtt{l}[y, \mathcal{F}(I;\boldsymbol{\theta}_d)] \right\}_{d=1}^{D} \Leftrightarrow \min_{\{\boldsymbol{\theta}_d\}_{d=1}^{D}} \sum_{d=1}^{D} \sum_{(I,y)\in\mathcal{D}_d} \mathtt{l}[y, \mathcal{F}(I;\boldsymbol{\theta}_d)], \quad (1)$$

where $\mathtt{l}$ indicates the cost function, usually cross-entropy classification loss, and $\boldsymbol{\theta}_d$ is the parameter set for the d-th dataset $\mathcal{D}_d$. In (1), the ConvNet of the parameter $\boldsymbol{\theta}_d$ is individually trained on the dataset $\mathcal{D}_d$. Then, the neuron activations


Soft
100
40
conv.
conv.

the task-oriented part without carefully considering the overlaps among the label sets; this is practically useful to free us from manually checking label contents. Through learning on various datasets of heterogeneous tasks, we can enhance the generality of the shared ConvNet, which facilitates classifying both objects (ImageNet) and scenes (Places-365), by extracting fundamental features shared across them.

The proposed model that shares the first l layers is trained as follows.

$$\min_{\boldsymbol{\theta}_0^{1:l},\{\hat{\boldsymbol{\theta}}_d^{l+1:L}\}_{d=1}^D} \sum_{d=1}^{D} \sum_{(I,y)\in\mathcal{D}_d} \mathtt{l}[y, \mathcal{F}(I;\boldsymbol{\theta}_d = \{\boldsymbol{\theta}_0^{1:l}, \hat{\boldsymbol{\theta}}_d^{l+1:L}\})], \tag{3}$$

where $\boldsymbol{\theta}_0^{1:l}$ indicates the shared parameters of up to the l-th layer and $\hat{\boldsymbol{\theta}}_d^{l+1:L}$ is the remaining parameter set which is specific to the d-th dataset. In other words, the ConvNet of $\mathcal{F}(I;\boldsymbol{\theta}_d = \{\boldsymbol{\theta}_0^{1:l}, \hat{\boldsymbol{\theta}}_d^{l+1:L}\})$ is trained on the d-th dataset. Note that the shared parameter $\boldsymbol{\theta}_0^{1:l}$ sees all the data while $\hat{\boldsymbol{\theta}}_d^{l+1:L}$ only looks at the data appearing in the d-th dataset $\mathcal{D}_d$. The degree of sharing ConvNet is controlled by the depth l at which the ConvNet branches (Fig. 1). This unified method (3) produces the separate model (1) by $l = 0$ and the hybrid model (2) by $l = L$. We conduct thorough experiments in Sect. 3 by gradually changing the depth l.

To properly learn the ConvNet (3) on heterogeneous datasets, we introduce the diversity into a mini-batch in training as follows. The same number of samples are drawn from respective datasets and packed into a mini-batch in order to fairly take into account the heterogeneous characteristics derived from the datasets at each updating step; for example, we sample 256 images from ImageNet and Places-365, respectively, and concatenate them to construct the mini-batch of 512 samples. Then, as shown in Fig. 1, each sample in the mini-batch is passed through the network differently according to which dataset it belongs to, and at the shared layers the derivatives for those (heterogeneous) samples are merged to update the network parameters $\{\boldsymbol{\theta}_d\}_{d=1}^D$ via mini-batched SGD. Thereby, the updating (derivative) is consistent throughout the end-to-end learning even on the heterogeneous datasets. In contrast, the MTL method [15] fills a mini-batch with *homogeneous* samples all of which are drawn from the randomly selected dataset. This produces consistent updates only when all the tasks are related, *i.e.*, the training datasets are homogeneous. In the case of heterogeneous datasets, however, the derivatives are inconsistent over the training steps since the characteristics of the mini-batches differ at every step according to what type of dataset is selected. This would hamper the learning, as empirically shown in Sect. 3. Note that our mini-batches merging derivatives across heterogeneous samples contribute to proper learning of ConvNet by effectively extracting the common updating information across the heterogeneous datasets.

3 Experimental Results

We apply the proposed method (Sect. 2) to the AlexNet model [1] which is composed of the five convolution and three fully-connected layers ($L = 8$) as shown

in Fig. 1; hereafter, we follow the conventional naming of the layers, such as `conv1` for the first convolution layer. Note that since the batch normalization [9] is embedded in the ConvNet (Fig. 1), we do not apply DropOut [8]. All the networks are implemented by using MatConvNet toolbox [17].

3.1 Datasets

In this study, we train the ConvNets (3) on two large-scale datasets of ImageNet [5] for object classification and Places-365 [6] for scene classification. The ImageNet contains 1,329,405 training images of 1,000 object classes (ILSVRC2014) and the Places-365 is composed of 1,839,960 images sampled from 365 scene categories. For the hybrid model (2), we simply concatenate those two label sets into 1,365 class labels as in [6].

3.2 Mini-batch

For separately training ConvNets (1) (or (3) of $l = 0$), we apply the mini-batch of 256 samples on the respective datasets. On the other hand, as described in Sect. 2, we draw 256 samples from ImageNet and Places-365, respectively, to construct the (heterogeneous) mini-batch of 512 samples, in a fair manner with the training of the separate model (1). Note that the mini-batch of 512 samples is split into two mini-batches of 256 samples at the branch in our ConvNets (3), as shown in Fig. 1. Since the two datasets contain different numbers of images, *i.e.* ImageNet is smaller than Places-365, we pad ImageNet dataset with images randomly picked up from that dataset so that it has the same number of images as Places-365. Thereby, we can draw the same number of samples from those datasets in constructing the heterogeneous mini-batch.

The mini-batch is filled with images of 224×224 pixels cropped from the original ones with random flipping and jittering in terms of position and pixel values as in [1].

3.3 Learning

The ConvNets are trained by SGD in 20 epochs through decreasing learning rate constantly on log-scale from 10^{-1} to 10^{-4}; the learning rate is determined as $10^{-\frac{16+3t}{19}}$, $t \in \{1, \cdots, 20\}$ where t indicates the epoch. We use the learning parameter of 0.9 for momentum and 0.0005 for weight decay. This training scheme is applied to any ConvNets.

3.4 Performance on ImageNet and Places Datasets

We evaluate performance on the datasets used for training. According to the standard evaluation protocols in ImageNet [5] and Places-365 [6], we measured the top-5 classification error rates on a validation set by applying 10-crop testing procedure to test images [1]. In the hybrid model of (2) (or (3) with $l = 8$),

the last fully-connected layer is split so as to produce 1,000 class outputs on ImageNet and 365 on Places-365 after learning, which results in the same architecture as the model of $l = 7$. Note that the separate model (1) corresponds to the original AlexNet model.

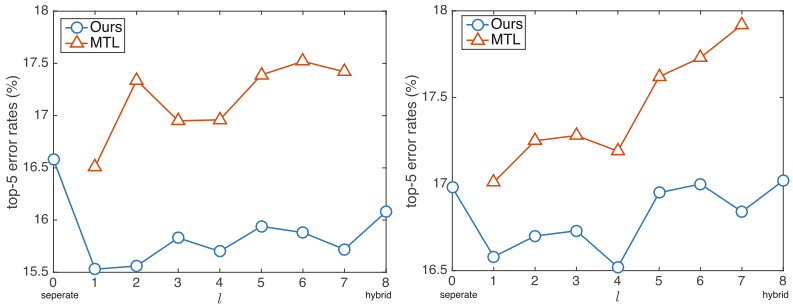
Figure 2 shows the performance results. Though the performances are slightly fluctuated due to only 1-shot evaluation, we can see that (1) the proposed model sharing a part of ConvNet improves performance being superior even to the hybrid model [6], and (2) the models sharing smaller part exhibit better performance; the best result is achieved by the model of $l = 1$. The hybrid model of $l = 8$ merges (concatenates) the label sets of ImageNet and Places-365 by force, and thus might take into account the label correlation wrongly, degrading performance, compared especially to our model of $l = 7$. Our method enjoys larger performance improvement on ImageNet than on Places-365 since the samples from Places-365 compensate the smaller-scale ImageNet by favorably exploiting the common characteristics across them. In contrast, the MTL method[1] [15] does not contribute to improvement but degrade the performance. The comparison between ours and the MTL highlights the effectiveness of our heterogeneous mini-batch construction for leveraging the heterogeneous datasets to improve performance. The MTL switches a dataset to produce mini-batches at each SGD step, leading to poor results especially as the shared components increases due to inconsistently updating the network at training steps. On the other hand, our approach makes the update consistent throughout the learning by merging the derivatives at each step to exploit the effective update information which is common across the heterogeneous samples. The heavily shared model of larger l imposes the same feature extractor on these heterogeneous tasks, which slightly deteriorates the performance compared to those of smaller l. Such shared model, however, would contribute to a general feature extractor as described in the next section.

3.5 Transferability

Next, we evaluate the transferability of the above pre-trained ConvNets by applying them to the other datasets than ImageNet and Places-365. The pre-trained ConvNets are tested on various datasets which are categorized into four types in terms of classification targets (Table 1); VOC2007 [18] and Caltech256 [19] for *object* classification, Indoor67 [20], Scene15 [21] and SUN397 [22] for *scene* classification, and Bird200 [23], Flower102 [24] and Pet37 [25] for *fine-grained* classification, and Event8 [26], Action40 [27] and FMD [28] for the *others*.

The image features are extracted by applying the pre-trained Convnet in a convolution manner to a rescaled image of which the minimum side has 256 pixels, and then are max-pooled over the image region. The neuron activations at the intermediate layer are employed to produce holistic image feature vector of fixed

[1] In training by the MTL method, for fair comparison, we use the same number of samples as in ours by padding ImageNet dataset.



(a) ImageNet (b) Places-365

Fig. 2. Classification error rate (%) on a validation set of ImageNet/Places-365. The top-5 error rates are measured by applying 10-crop testing procedure [1]. The ConvNets (3) are trained by the MTL approach [15] and ours.

Table 1. Details of the datasets used for evaluating transferability. This table shows the number of training samples, test samples and class categories from the top row to the bottom.

	Object		Scene			
	VOC2007	Caltech256	Indoor67	Scene15	SUN397	
Training samples	5011	15360	5360	1500	19850	
Test samples	4952	9984	1340	2985	19850	
Categories	20 objects	256 objects	67 scenes	15 scenes	397 scenes	
	Fine-grained			Others		
	Bird200	Flower102	Pet37	Event8	Action40	FMD
Training samples	5994	2040	3680	560	4000	500
Test samples	5794	6149	3669	480	5532	500
Categories	200 species	102 species	37 species	8 sports	40 actions	10 materials

dimensionality. As shown in Fig. 3, the ConvNet pre-trained on ImageNet/Places-365 exhibits dependency on the types of the training datasets. For achieving general features, as in [10], we exploit the layers of `fc7`[2] both on ImageNet and Places-365 (see Fig. 1b) and concatenate them into the 8,192-dimensional feature vector for the models of $l = 0, \cdots, 6$. On the other hand, we concatenate `fc6` and `fc7` to produce 8,192-dimensional features for $l = 7, 8$ since the layers of `fc6` and `fc7` are both shared in those models (see Fig. 1c, d). The features are finally classified by linear SVM [29] and the classification accuracy is measured according to the standard protocol provided in the respective datasets; on Caltech256, we draw 60 training samples on each class, and for the details, refer to the respective papers.

The performance results are shown in Table 2. By combining two type of pre-trained ConvNets for objects (ImageNet) and scenes (Places-365), the

[2] `fc7` outperforms `fc6` as shown in Fig. 3.

ConvNet features exhibit favorable transferability on various kinds of tasks including both object and scene classifications. The heavily shared models of larger l are superior to those of smaller l, which contrasts to Table 2. By sharing larger part of ConvNet across the heterogeneous datasets, the pre-trained ConvNet achieves better generalization power by exploiting common (general) features. Especially, the model of $l = 7$ produces favorable performance on the tasks of *fine-grained* and *others*. Comparing $l = 7$ with $l = 8$ (hybrid), one can see that splitting `fc8` layer is more effective than concatenating label sets for enhancing generalization performance.

We can conclude that (1) the less shared ConvNet of $l = 1$ is effective for improving performance on the task of training target (Fig. 2), and (2) the heavily shared ConvNet of $l = 7$ provides a general feature extractor with better transferability (Table 2).

Table 2. Classification accuracies (%) by the pre-trained ConvNets on various datasets.

Dataset	Separate $l = 0$	$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$	$l = 7$	Hybrid $l = 8$
VOC2007	79.97	80.50	80.20	80.52	80.59	80.38	80.15	79.96	80.05
Caltech256	74.27	74.85	75.07	74.39	74.88	74.19	74.78	74.59	74.91
avg. (*object*)	77.12	77.67	77.64	77.46	**77.73**	77.29	77.46	77.28	77.48
Indoor67	74.82	75.56	76.53	76.19	75.40	75.84	75.77	75.38	75.56
Scene15	93.11	93.39	93.47	93.35	93.65	93.31	93.05	93.03	93.06
SUN397	60.63	61.13	61.32	61.05	61.19	60.48	60.13	59.22	59.01
avg. (*scene*)	76.19	76.69	**77.11**	76.86	76.75	76.54	76.32	75.88	75.88
Bird200	63.35	62.82	62.87	62.68	62.34	63.05	63.45	65.29	64.72
Flower102	90.07	90.18	90.79	90.36	90.22	90.32	91.01	90.20	90.60
Pet37	81.92	82.33	82.41	82.10	81.62	81.55	81.66	82.37	81.68
avg. (*fine-grained*)	78.44	78.44	78.69	78.38	78.06	78.31	78.71	**79.29**	79.00
Event8	96.04	95.90	96.04	95.97	96.32	96.11	96.60	96.32	96.11
Action40	62.60	64.21	64.05	64.37	64.70	63.69	63.34	64.67	63.61
FMD	72.85	73.27	73.62	72.19	73.73	73.65	75.13	74.37	72.40
avg. (*others*)	77.16	77.80	77.90	77.51	78.25	77.81	78.35	**78.45**	77.37

3.6 Fine-Tuning

Fine-tuning is employed to further adapt the pre-trained ConvNet to the target dataset, though requiring tedious learning parameter tuning. We fine-tune the pre-trained ConvNet by decreasing the learning rate from 10^{-3} to 10^{-6} over 40 epochs ($10^{-\frac{114+3t}{39}}$, $t \in \{1, \cdots, 40\}$), with the mini-batch of 128 samples. Note that the ConvNet is initialized as the optimized parameter values in Sect. 3.4 except for the last `fc8` layer which is randomly initialized.

Based on the results in Table 2, we apply the model of $l = 7$ to the tasks other than *object* and *scene* classifications which are the targets in the pre-training. The performance results are shown in Table 3. By fine-tuning the model,

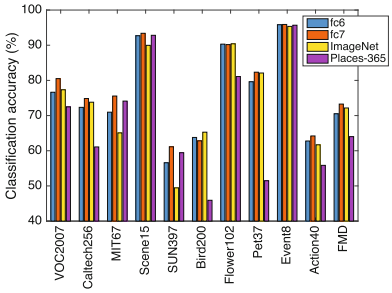


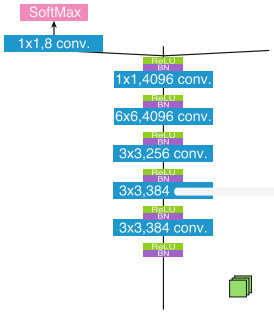
Fig. 3. Performance comparison for `fc6`, `fc7` in the model of $l = 1$ and `fc7` in ConvNets pre-trained on ImageNet/Places-365. The `fc7` features exhibit superior performance to `fc6`, and ImageNet-ConvNet works only on the ImageNet-related tasks, excluding scene classification.

Table 3. Classification accuracies (%) of fine-tuned ConvNet of $l = 7$ pre-trained on both ImageNet and Places-365 in Sect. 3.4.

		Original	Fine-tuned
Fine-grained	Bird200	65.29	65.59
	Flower102	90.20	91.53
	Pet37	82.37	80.24
Others	Event8	96.32	96.60
	Action40	64.67	65.00
	FMD	74.37	76.56

Table 4. Classification accuracies (%) of the pre-trained ConvNets of $l = 7$ which is fine-tuned by our method. All the three datasets are used in our fine-tuning.

	Others		
	Event8	Action40	FMD
Original	96.32	64.67	74.37
Standard fine-tuning	96.60	65.00	76.56
Our fine-tuning	96.81	65.13	78.03



(Table 1). Through our fine-tuning, the ConvNet can see larger number of training samples and effectively exploit common characteristics across the multiple datasets to improve performance on the small-scale dataset.

4 Conclusion

In this paper, we have proposed a method to train a ConvNet on heterogeneous tasks (datasets) for improving performance. In the proposed method, the ConvNet layers are partially shared across the different datasets in the end-to-end learning to enhance generalization power, while the remaining layers are tailored to respective tasks (datasets). By controlling the degree of shared network layers, the method provides various types of ConvNet of different generality. To properly learn the ConvNet on the heterogeneous datasets, we construct a mini-batch so as to fairly contain heterogeneous samples, producing consistent updates (derivatives) throughout the training. The experimental results on ImageNet and Places-365 datasets show that the ConvNet sharing less layers favorably improves performance on those dataset, and that of heavily shared layers exhibits better generalization performance with favorable transferability. We have also demonstrated that the proposed method is applicable to fine-tuning the ConvNet especially on small-scale datasets. Our future works include to apply the method to various ConvNets.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
2. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
3. Ouyang, W., Wang, X., Zeng, X., Qiu, S., Luo, P., Tian, Y., Li, H., Yang, S., Wang, Z., Loy, C., Tang, X.: Deepid-net: deformable deep convolutional neural networks for object detection. In: CVPR, pp. 2403–2412 (2015)
4. Bertinetto, L., Valmadre, J., Henriques, J., Vedaldi, A., Torr, P.: Fully-convolutional siamese networks for object tracking arXiv:1606.09549 (2016)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: CVPR, pp. 248–255 (2009)
6. Zhou, B., Khosla, A., Lapedriza, A., Torralba, A., Oliva, A.: Places: an image database for deep scene understanding. arXiv:1610.02055 (2016)
7. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: ICML, pp. 807–814 (2010)
8. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
9. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. J. Mach. Learn. Res. **37**, 448–456 (2015)

10. Azizpour, H., Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: Factors of transferability for a generic convnet representation. IEEE Trans. Pattern Anal. Mach. Intell. **38**(9), 1790–1802 (2016)
11. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: CVPR, pp. 1717–1724 (2014)
12. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: CVPR Workshop, pp. 512–519 (2014)
13. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: ECCV Workshop, pp. 1–22 (2004)
14. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)
15. Collobert, R., Weston, J.: A unified architecture for natural language processing. In: ICML, pp. 160–167 (2008)
16. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: NIPS, pp. 568–576 (2014)
17. Vedaldi, A., Lenc, K.: MatConvNet - convolutional neural networks for matlab. In: ACM MM (2015)
18. The PASCAL Visual Object Classes Challenge 2007 (VOC 2007). http://www.pascal-network.org/challenges/VOC/voc2007/index.html
19. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset. Technical report 7694, Caltech (2007)
20. Quattoni, A., Torralba, A.: Recognizing indoor scenes. In: CVPR, pp. 413–420 (2009)
21. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: CVPR, pp. 2169–2178 (2006)
22. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from Abbey to zoo. In: CVPR (2010)
23. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD birds-200-2011 dataset. Technical report CNS-TR-2011-001, California Institute of Technology (2011)
24. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian Conference on Computer Vision, Graphics and Image Processing (2008)
25. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.V.: Cats and dogs. In: CVPR, pp. 3498–3505 (2012)
26. Li, L.J., Fei-Fei, L.: What, where and who? classifying events by scene and object recognition. In: ICCV (2007)
27. Yao, B., Jiang, X., Khosla, A., Lin, A., Guibas, L., Fei-Fei, L.: Human action recognition by learning bases of action attributes and parts. In: ICCV (2011)
28. Sharan, L., Rosenholtz, R., Adelson, E.: Material perception: What can you see in a brief glance? J. Vis. **9**(8), 784 (2009)
29. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)