Learning Bag-of-Multi-Scale-Activations from Deep Convolutional Nets

Anonymous CVPR submission

Paper ID ****

Abstract

While Deep Convolutional Neural Network (CNN) encodes multi-scale features with their spatial locations, Bag-of-feature representations are more invariant to spatial translations. In this paper, we propose a CNN architecture transformation schema that extracts multi-scale CNN representations with spatial invariance. By directly chaining multiple average-pooled rectified linear units (ReLU) to the final layer of scoring function, we explicitly learn an augmented model from multi-scale activations. We term this transformation Bag-of-Multi-Scale-Activation (BoMSA) Augmentation. The BoMSA augmented model can be trained from scratch or built upon existing pre-trained models, which makes the augmentation more flexible. BoMSA augmentation is also model-independent and can be easily adapted to existing models. Experimental results show consistent improvements achieved by incorporating the BoMSA representation on various models in the literature.

1. Introduction

Deep CNN has shown its versatility in the tasks such object detection and recognition [16, 23, 24, 28, 18] in the field of computer vision. Trained with large number of instances, (such as ImageNet [3]), CNN is also an excel candidate for off-the-shelf feature extractions, results in outstanding performance in various recognition tasks [21]. In the meantime, a considerable amount of effort is spent on how to further improve the performance of CNN. On one hand, many are focus on techniques to efficiently and effective train a CNN. A good initialization need to be carefully selected [2] in the beginning. Data augmentation [16] is recommended to improve the model performance as well. Drop-out [13] and momentum [27] are also necessary to prevent over-fitting and obtain superior models. On the other hand, different model components and architectures are proposed. Rectified linear units (ReLU) [16] add non-linearity and enrich the model complexity. Different pooling method, such as Distance Transform Pooling [9], is adopted in [10], allowing local deformations, as in the widely-used deformable part-based model (DPM) [8]. [24] adopts a small 3×3 receptive field to deepen the model, while maintaining less parameters. A network in network [18] is proposed to enhance model discriminability for local patches within the receptive field. The award winning GoogLeNet [28] uses an *Inception* model that is based on the Hebbian principle [12], *i.e.*, neurons that fire together, wire together, the theoretical proof of which is provided by [1] under constraints.

One question to ponder: is there any model-independent potential that is yet to be discovered. Although allowing local deformation, CNN encodes the the spatial information of multi-scale features. Conversely, by ignoring the spatial location of features, bag-of-feature like techniques [17] achieves transformation invariance to some extend. As pointed out in [11], combining both type of features results in a better representation for recognition tasks with large variations, such as Scene classification [29, 20]. Traditionally, when CNN is used for feature extraction, the activation of the first fully-connected (FC) layer is usually considered as the feature [23, 11]. Using an off-the-shelf deep CNN model, [11] explicitly extracts image patches from three scales and computes the feed-forward CNN activations for each patch for feature extraction. This algorithm is cumbersome due to the computation of multiple patches for one image. Besides, it can only encode the multi-scale features to a certain extend limited by the burden of the post processing, *i.e.*, K-means and Principle Component Analysis (PCA). As a matter of fact, CNN feature is meant to encode multi-scale feature at each convolutional (Conv.) layer. Therefore, there is no need to extract multi-scale patches and only one feed-forward computation is sufficient to capture the activations of multi-scale features for one input image.

In this paper, we first conducted an empirical analysis on the feature discriminativeness of every unit of CNN model. By average-pooling at each layer, the output ignores global spatial information and results in a bag-of-feature representation. The bag-of-feature from ReLU is then found to carry the most performance gain at each layer. We have also observed a synergy of multi-scale activations results

(a) low-level feature is preferred

(b) high-level feature is preferred

Figure 1. Retrieval results using euclidean distance for both low- and high-level features on MIT67 [20]. *Green* (*Red*) box means correct (wrong) results. The correct label for wrong retrievals are provided. The retrieval results are displayed such that the left-most image has the closest distance to the query, and vice versa. This observation shows that both low- and high-level features are dispensable for a better representation.

in a better discriminative representation. These observations tie closely to the vanishing gradient [2] issue in the deep CNN literature. During the training of a CNN, the top layer can be easily saturated. The gradients in the back-propagation algorithm will not effective reach the lower levels, resulting in a model that focuses more high-level features. Thus, we proposed a model augmentation schema, BoMSA, that chains the ReLU at each layer to the ultimate loss function. This augmentation approach is not confined by model variations and can be applied to existing pre-trained models or re-training the model from the ground up. By explicitly linking the lower layers to the decision layer, we gear the training towards the bag-of-activations from all layers. Thus, the augmented model is designed to be robust to spatial translations while maintaining its discriminative power in the multi-scale fashion. We apply our augmentation approach on various models (*e.g.*, Caffe [14], Deep19 [24], etc.) in the literature and test them on various scene recognition benchmark datasets, such as SUN397 [29], MIT67 [20], Scene15 [7]. A consistent performance improvement is observed of our multi-scale representation over the best discriminative single-scale feature.

The rest of the paper is organized as follows: we motivate and describe our model augmentation approach in Sec. 2. An in-depth analysis is provided in Sec. 3. Systematic experimental results are included in Sec. 4.

2. Approach

We seek to discover a model-independent augmentation approach that enable the model with multi-scale feature representation. We first provide our motivation based on empirical studies on the discrimintiveness of individual feature at various scales as well as multi-scale features. We then describe our model augmentation technique.

In this work, we consider two models in the literature, namely Caffe and Deep19. Caffe [14] is a well-known deep learning framework. It has a pre-trained model which follows the landmark "AlexNet" [16] trained with millions of data in the ImageNet dataset [3]. This model has 6 conv. layers and 2 fully-connected (FC) layers. Deep19 [24] uses very small 3×3 receptive fields for the entire net and the deepest model has 19 layers (16 conv. and 3 FC layers). This model registered the state-of-the-art performance in ILSVRC-2014 classification challenge [22].

We intend to demonstrate the consistency and model-independence of our observations and approach. We have

conducted experiments on several CNN models and observed similar patterns. Due to limited space, we only provide the analysis of the aforementioned Caffe and Deep19 models as exemplars in the rest of the paper.

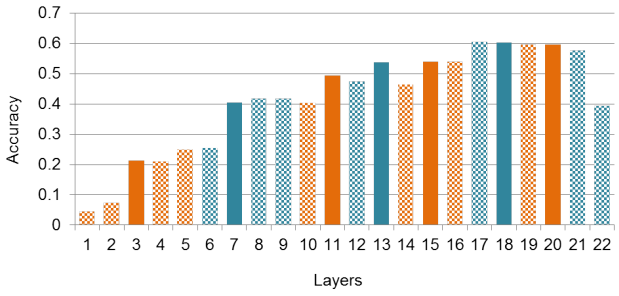
2.1. Motivation: Classification Performance of Individual CNN Activation at Different Scale

We consider the average-pooled activation of each computation unit in a CNN as the feature at its corresponding scale. We conduct classification experiments on MIT Indoor Scene (MIT67) dataset [20] using both Caffe and Deep19 models. 67-way one-vs-all linear SVM classifiers are then trained from a 10-fold cross validation, and the classification accuracy is reported in Fig. 2. The detailed parameter option for both Caffe and Deep19 models are similar to the ones described in Sec. 4. As seen in Fig. 2, in general, the discriminativeness of the model increases as the layer becomes higher. More importantly, we observe significant improvement resulting from each Rectified Linear Units (ReLU). As described in [19, 16], ReLU models a neuron's output r as a mapping of its input x with the non-saturating nonlinearity $r(x) = \max(0, x)$. We are then motivated to choose the output of the ReLU layer due to this nonlinearity it introduces to the CNN architecture, which boosts the discriminative power of the model.

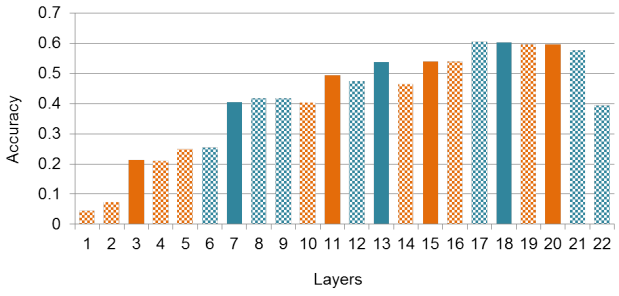
2.2. Motivation: Classification Performance of Multi-scale CNN Activation

Since different CNN layers correspond to image feature at various scales, seen in Fig. 1, we wonder whether combining activations at multiple scales help to extract a better feature. To address this question, we carried out another experiments on the MIT67 data [20]. We start from the average-pooled response of the last ReLU layer and greedily concatenate the ones from previous ReLU layers. The reason we start from the last ReLU layer is that, in the literature when CNN is used for feature extraction, the response of the first fully connected layer is usually selected [21, 11]. Besides, our previous analysis in Fig. 2 also demonstrates better discriminative power of high-level CNN layer.

The rest of the experimental setups are the same as in Sec. 2.1, *i.e.*, a set of 67-way one-vs-all linear SVM classifiers are trained every time we add one more layer. Since we keep on concatenating features at lower level, the feature dimension is monotonically increasing. As the results shown in Fig. 3, the performance increase in the beginning by adding mid-level features to enrich the model discriminativeness. However, including some low level responses to the feature actually hurts the performance, *i.e.*, layer 7 and 3 in Fig. 3(a). We believe it demonstrates the benefit of incorporating more discriminative mid-level and high-level features of CNN (*i.e.*, *parts* and *objects*), but not necessarily the low-level features, such as *edge* or *textures*.

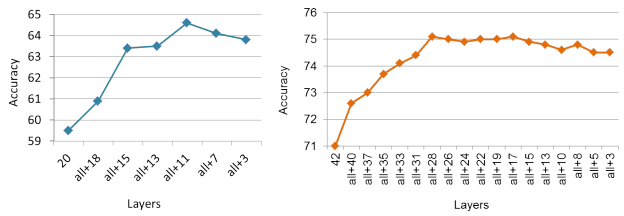


(a) Caffe



(b) Deep19

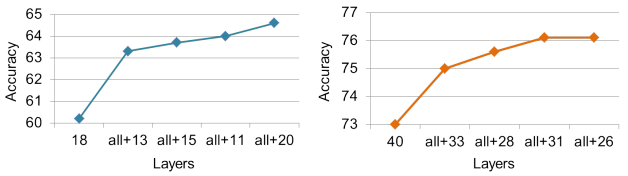
Figure 2. The classification accuracy on MIT67 [20] using the marginal activation of each layer. The color alternation indicates change of Conv. layers. The solid color fill represents the output of the ReLU layer. There are 7 ReLU layer for the Caffe model and 18 for Deep19. A significant performance improvement is observed at each ReLU layer compared with its previous layer, especially at the early Conv. layers.



(a) Caffe (b) Deep19

Figure 3. The performance trend when incorporating more lower-level features. The keyword "all" means the previously added average-pooled ReLUs. A performance drop is observed when incorporating more low level features.

A more systematic way for select the correct scale to incorporate as feature is feature forward selection, *i.e.*, greedily add the layer in the ReLU pool which results in the best performance boost in an iterative manner. As seen in Fig. 4(a), the optimal results of this greedy approach is congruent with the previous results in Fig. 3(a), which rejects the low-level features. Similarly for the Deep19 model in Fig. 4(b), although the feature pool is large (18 ReLU layers), the optimal greedy results includes only the mid- or high-level features.



(a) Caffe (b) Deep19

Figure 4. The performance trend when using forward selection to incorporate fs from the ReLU layer.

We should point out that for the latter analysis and experiments on other datasets, we opt to use scales selected by the forward selection algorithm on MIT67 data. An consistent performance boost demonstrates the data-independence of our analysis and observations.

2.3. Model

Based on the analysis in Sec. 2.1 and Sec. 2.2, we propose a novel deep CNN architecture, Bag-of-Multi-Scale-Activation (BoMSA), shown in Fig. 5. A typical CNN layer consists of four layers, *i.e.*, Conv., ReLU, contrast normalization (Norm), Max-pool layers (with the Norm and Max-pool layers being optional). This augmentation schema converts the traditional CNN *chain* model [16, 24] to a complex model by linking each ReLU to an average-pooled layer and FC layer thereafter. Before feeding into the Soft-max layer, an *Add* layer is introduced to combine the responses from all FC layers.

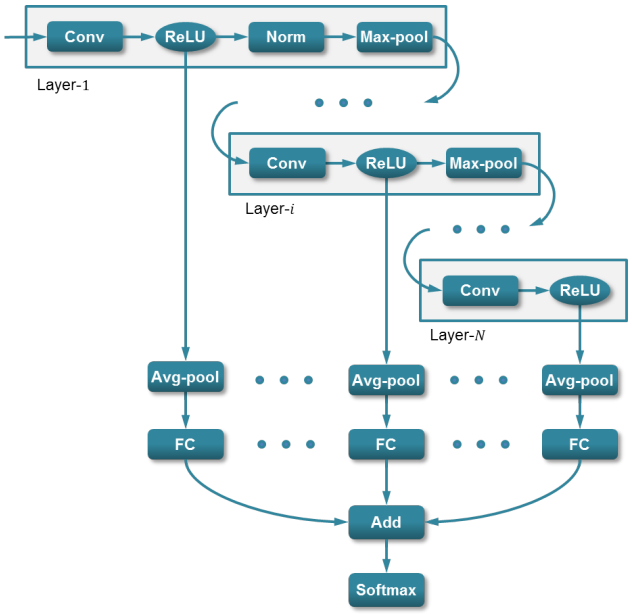


Figure 5. The architecture of the proposed generalized model augmentation technique. The traditional *chain* model is represented by the connections from Layer-1 to Layer-N.

2.4. Training

Let $\mathbf{w}_1, ...\mathbf{w}_K$ be the CNN model parameters at $1, .., K$-th layer, training data be $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, where $\mathbf{x}^{(i)}$ is the i-th input image and $\mathbf{y}^{(i)}$ is the indicator vector of the class of $\mathbf{x}^{(i)}$. Then we intend to solve the following optimization problem

$$\arg\min_{\mathbf{w}_1,...\mathbf{w}_K} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f(\mathbf{x}^{(i)}; \mathbf{w}_1, ..., \mathbf{w}_K), \mathbf{y}^{(i)}) \quad (1)$$

We adopt the stochastic gradient descent to minimize the objective function. For a traditional *chain* model, the derivative of the objective function computed by chain rule results in the well-known back-propagation algorithm. In our proposed complex model, care need to be taken in the feed-forward for Add layer and back-propagation for all the ReLU layers. In the feed-forward step, the Add unit simply adds multiple inputs from all ReLU layers. In the back-prop step for i-th ReLU layer (seen in Fig. 6), let α_i be its input, $\beta_i^{(j)}$ be the output for its j-th branching; z is the final output of the Softmax layer. Thus, the gradient of z with respect to i-th ReLU layer can be computed as

$$\frac{\partial z}{\partial \alpha_i} = \sum_{j=1}^{C} \frac{\partial z}{\partial \beta_i^{(j)}} \frac{\partial \beta_i^{(j)}}{\partial \alpha_i} \quad (2)$$

where $C = 2$ for our model. Each partial derivative component, $\frac{\partial z}{\partial \beta_i^{(j)}} \frac{\partial \beta_i^{(j)}}{\partial \alpha_i}$, can be computed in the typical back-prop fashion.

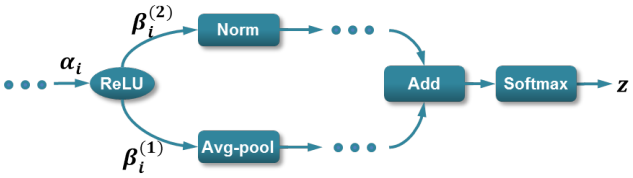


Figure 6. Visualization of the parameter setup at i-th ReLU.

3. Discussions

3.1. off-the-shelf Model Instantiation

We first point out an interesting instantiation of our model, linking back to the analysis in 2.2. We could switch the softmax loss to hinge loss and freeze the back-prop at the FC layer. In this way, we prevent altering the chain-part of our model and only allow training on the augmented part our model, *i.e.*, we learn an linear combination of the FC layers represents multi-scale ReLU activations. Thus, this is equivalent of training a SVM classifier based on the concatenation of the average-pooled activations at all ReLUs.

As a result, we could extract multi-scale features using pre-trained CNN models such as the ones in [16, 24] and use off-the-shelf SVM solver such as [6] to train the augmented part our model.

3.2. Relationship to the literature

Our work is inspired by [11]. In [11], local patches at three scales are first extracted (roughly 50 patches per image in their setting); each patch is then feed to an off-the-shelf CNN and the activations of the first fully connected layer (4096-dimension) are used for post processing, *i.e.*, K-means + VLAD pooling.

Our approach has several advantages over [11]:

1. [11] computes multiple feed-forward on patches generated from a single image. Computing convolution is the bottleneck for this procedure. In our approach, our model is trained to adapt to multi-scale features. Thus, during test phase, only one feed-forward computation is needed for one image. As matter of fact, extracting feature of one image in [11] takes more than 20 seconds while ours only take less than 1 second for the augmented Caffe model.

2. [11] uses K-means clustering algorithm in the VLAD pooling, which can generate inconsistent results due to random initialization.

3. [11] conducts two steps of PCA (first PCA to reduce 4096-dimensional activation to 500-dimension, and second PCA to reduce $50,000$-dimensional VLAD pooled feature to 4096-dimension). This can also be one bottleneck of their approach.

3.3. The Analysis of Invariance

Besides the insight on visualizing deep CNN features, [30] provides transformation invariance analysis of their model on several images. This is done by comparing the feature vector distance between the original and transformed images. Realizing that several examples do not necessarily represents the overall statistics, [11] improve the invariance analysis by experiments on the SUN397 scene database [29]. Unfortunately, only the results on 4 categories are provided in their paper. The performance for the entire dataset is still unknown.

To close the aforementioned gap and analyze the invariance of our representation, we carried out experiments on the entire MIT Indoor dataset [20] and report the average performance. We intend to compare the tolerance to transformation of the our multi-scale feature compared with a single-scale CNN feature. Concretely, the single-scale CNN feature in comparison is the activation of a CNN model which achieves best classification performance (not necessarily the output of the first FC layer). For a fair comparison, the multi-scale features are extracted by concatenating the output of all sum-pooling layers. We first trained 67-way one-vs-all linear SVM classifiers for all 67 classes using features extracted from the original images. During test phase, four transformations are considered for the analysis, *i.e.*, translation, scaling, flipping, and rotation. Fig. **??** illustrates all the transformations and their corresponding parameters, which are similar to the ones in [11].

After applying the transformations to test images, both multi- and single-scale features are extracted. The classification is conducted using the trained SVMs, and the accuracy is shown in Fig. 7. Each data point represents the performance on the entire test data. As seen in Fig. 7, the transformations almost always hurt the performance except for horizontal flip. This observation is similar to the one in [11]. We also see that multi-scale feature consistently out-performs the single-scale feature under every degree of all transformations. As far as the invariance is concerned, we observe the similar level of performance decrease. This means that both features have the similar level of tolerance to transformations. This immediately suggest that "training jittering", *i.e.*, including transformations in the training data, will improve the performance.

4. Experimental Results

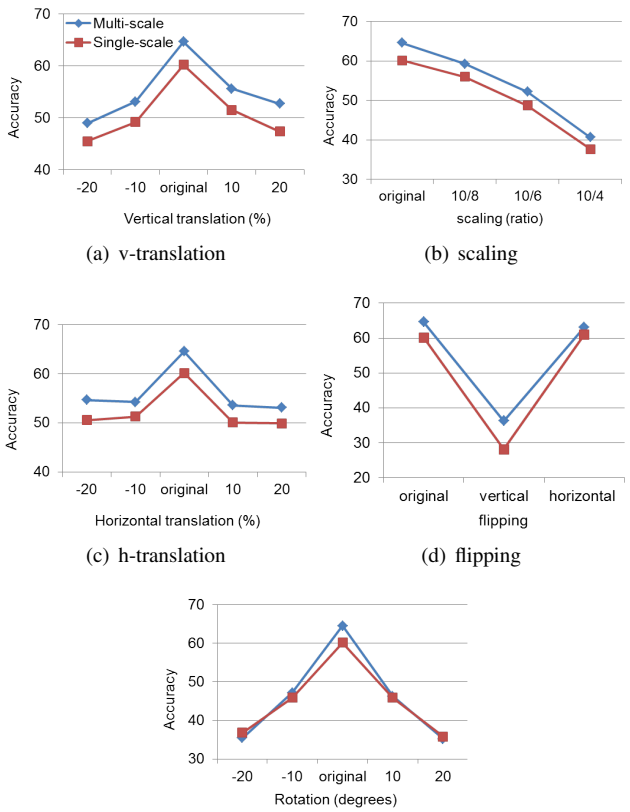
In this section, we conduct experiments on benchmark datasets, SUN397 [29], MIT67 [20], and Scene15 [7], and show consistent performance increase using our approach.

4.1. Experiment Setup

We should first stress that our augmentation is applied to the ReLU layers selected based on MIT67 dataset, as shown in Fig. 4. The learned structure for both Caffe and Deep19 is then applied to all the experiments in this Section, demonstrating the data-independence of our approach.

We follow the standard image pre-processing steps in the literature [16, 14, 24]. Since both Caffe [14] and Deep19 models [24] have a fixed input size of 224×224, we first resize the image such that the smaller side matches 224, and then crop an 224×224 patch from the center. The mean RGB value of each model learned on ImageNet [3] is then subtracted.

We show results of using our BoMSA augmented model for feature extraction. Thus, the output of all the average-pooled ReLU activations are concatenated as our multi-scale representation. For a fair comparison, we compare the multi-scale feature to the best performing single-scale feature. All features are $l2$-normalized. Then the off-the-shelf linear SVM solver [6] is used to train 67-way one-vs-all classifiers with no parameter tuning.



(a) v-translation (b) scaling

(c) h-translation (d) flipping

(e) rotation

Figure 7. The classification accuracy of various transformations on test images in MIT67 data [20]. The legends shown in Fig. 7(a) is the same in the other four figures.

Approach	Accuracy(%)
Deep19-BoMSA	**55.5**
Deep19-single	51.9
Caffe-BoMSA	46.6
Caffe-single	43.5
MOP-CNN [11]	52.0
Places [**?**]	54.3
DeCaf [5]	40.9
FV [26]	47.2
Baseline [29]	38.0

Table 1. Classification results on SUN397

4.2. SUN397

SUN397 [29] is a large scene recognition dataset with 397 categories, each of which includes more than 100 images. The total number of images exceeds 100k. The average classification accuracy is usually report from a 10-fold cross validation. The split file is provided in [29] with 50 images for training and the rest being test for each fold.

¡Improvement over pre-trained models¿ ¡Achieves best performance¿ ¡talk about Places, using different data -¿ model structure and better domain specific data are both important¿

Approach	Accuracy(%)
Deep19-BoMSA	**76.1**
Deep19-single	70.8
Caffe-BoMSA	64.6
Caffe-single	59.5
MOP-CNN [11]	68.9
Places [**?**]	68.2
Mid-level [4]	64.0
FV+BoP [15]	63.2
Disc. Patch [25]	49.4
SPM [17]	34.4

Table 2. Classification results on MIT67

4.3. MIT67

MIT67 refers to the MIT Indoor [20] scene classification for 67 categories. Indoor scenes depends on highly variable features to describe them. There are cases that can be well characterized by high-level spatial geometry (*e.g.* church and cloister) and low-level textures (*e.g.* wine celler and library). Our multi-scale model is designed to encode feature at all levels, and thus, enable us to learn both high- and low-level statistics of scene categories. The training/test split is made available and there are 80/20 training/test images for each class.

4.4. Scene15

5. Conclusion

References

[1] S. Arora, A. Bhaskara, R. Ge, and T. Ma. Provable bounds for learning some deep representations. *CoRR*, abs/1310.6343, 2013. 1

[2] Y. Bengio and X. Glorot. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AISTATS 2010*, 2010. 1, 2

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 2, 5

[4] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *ECCV*. 2013. 6

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, 2013. 6

[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 2008. 5

[7] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 2, 5

[8] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 1

[9] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 2004. 1

[10] R. B. Girshick, F. N. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. *CoRR*, abs/1409.5403, 2014. 1

[11] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*, 2014. 1, 3, 5, 6

[12] D. Hebb. *The Organization of behaviour*. New York: Wiley & Sons, 1949. 1

[13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. 1

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 2, 5

[15] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, 2013. 6

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012. 1, 2, 3, 4, 5

[17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1, 6

[18] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 1

[19] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 3

[20] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 1, 2, 3, 5, 6

[21] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014. 1, 3

[22] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. 2

[23] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013. 1

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014. 1, 2, 4, 5

[25] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 6

[26] J. Snchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 2013. 6

[27] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013. 1

[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 1

[29] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 1, 2, 5, 6

[30] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. 5