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 channeling 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 shows noticeable 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[], recognition[], segmentation[], etc. 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 [16]. In the meantime, a considerable amount of effort is given to how to further improve the performance of CNN. On one hand, many are focus on techniques to efficiently and effective train a CNN. First, a good initialization need to be carefully selected [2]. Data augmentation [12] is recommended to improve the model performance as well. Drop-out [10] and momentum [19] are also necessary to prevent over-fitting and result in superior models. On the other hand, different model components and architectures are proposed. A rectified linear unit (ReLU) [12] adds non-linearity and enriches the model complexity. Different pooling method, such as Distance Transform Pooling [6], is adopted in [7], allowing local deformations, as in the widely-used deformable part-based model (DPM) [5]. [18] adopts a small 3×3 receptive field to deepen the model, while maintaining less parameters. A network in network [13] is proposed to enhance model discriminability for local patches within the receptive field. The award winning GoogLeNet [20] uses an *Inception* model that is based on the Hebbian principle [9], *i.e.*, neurons that fire together, wire together, the theoretical proof of which is provided by [1] under constraints.

We ask ourselves, is there any model-independent hidden potential that has not been reveal yet.

¡talk about vanishing gradient problem in deep model¿

¡spatial-aware deep model vs. bag-of-feature model: talks about multi-scale [Gong14]¿

In this paper, present a new model, link ReLU layer for

Fig. CNN filter visualization at different scale

2. Related Work

3. Technical Approach

We seek to discover a model-independent feature representation that can be efficiently computed. This representation should conform the property of *multi-scale* and *orderless* which outperform single-scale representation in scene classification tasks.

3.1. Model

In this work, we consider two models in the literature, namely Caffe and Deep19. Caffe [11] is a well-known deep learning framework. It has a pre-trained model which follows the landmark "AlexNet" [12] trained with millions of data in the ImageNet dataset [3]. This model has 6 conv. layers and 2 fully-connected (FC) layers. Deep19 [18] 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 [17].

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 pro-

vide the analysis of the aforementioned Caffe and Deep19 models as exemplars in the rest of the paper.

3.2. Multi-scale Orderless CNN Activation

We consider the marginal activation of a rectified linear unit (ReLU) as the *orderless* feature response at its corresponding scale. As described in [14, 12], ReLU models a neuron's output r as a mapping of its input x with the non-saturating nonlinearity $r(x) = \max(0, x)$.

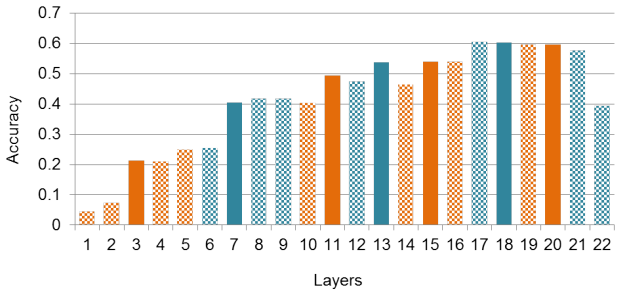
Concretely, to compute the proposed representation, let α_i of dimension $w \times h \times k$ be the output of ReLU at i-th layer ($w = h$ in most CNN models). Each $w \times h$ is the convolutional response map of one of the k filters. It carries the spatial activation information at the current scale. Thus, the *orderless* response, f_i, at each scale can be computed by sum-pooling the first two dimension of α_i, resulting in a k-dimensional vector, *i.e.*,

$$f_i = \sum_w \sum_h \alpha_i \tag{1}$$

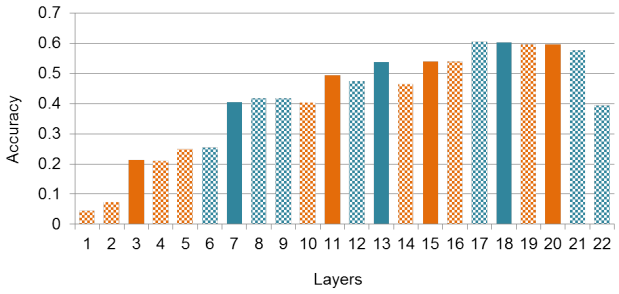
We are motivated to choose the output of the ReLU layer due to the nonlinearity it introduces to the CNN architecture, which boosts the discriminative power of the model. To further provide statistical evidences for the argument, we conduct classification experiments on MIT Indoor Scene (MIT67) dataset [15] using both Caffe and Deep19 models. We consider the orderless response, f_i, from each layer as different features; one-vs-all linear SVMs are then trained and the classification accuracy is reported in Fig. 1. The detailed experimental setups for both Caffe and Deep19 models are similar to the ones described in Sec. 4. As seen in Fig. 1, in general, the discriminativeness of the model increases as the layer becomes higher. More importantly, we observe significant improvement resulting from each ReLU layer. These observations not only inform us that the CNN model owes its performance boost to the nonlinearity of the ReLU layer, but also motivate us to leverage multiple ReLU layer responses to enrich the discriminative ability of the model.

3.3. The Feature Selection from Activation Layer

Since different CNN layer models image patterns at various scales, the first question to ask is that "Does combining activations at all scales help to extract a better feature"? To address this question, we carried out another experiments on the MIT67 data [15]. In the literature when CNN is used for feature extraction, the response of the first fully connected layer is usually selected. Our previous analysis in Fig. 1 also demonstrates the discriminative power of high-level CNN layer. Thus, in this experiment, we start from the f of the last ReLU layer and greedily concatenate the marginal responses from previous ReLU layer. The rest of the experimental setups are the same as in Sec. 3.2 and Sec. 4, *i.e.*, 67-way one-vs-all linear SVM classifiers are trained every time we add one more layer. Since we keep on concatenating the fs to our feature, the feature dimension is monotonically increasing. As the results shown in Fig. 2, 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. 2(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 *corner* or *edge*.



(a) Caffe

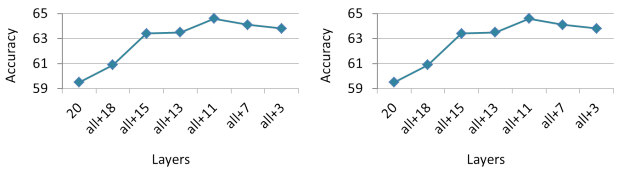


(b) Deep19

Figure 1. The classification accuracy on MIT67 [15] using the marginal activation of each layer. The color alternation indicates change of convolutional 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 convolutional layers.

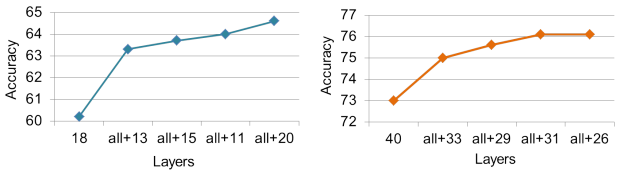
A more systematic way to select the layers to incorporate as feature is feature forward selection, *i.e.*, greedily add the layer in the ReLU pool which results in the best performance boost in an iterative manner. As seen in Fig. 3(a), the optimal results of this greedy approach is congruent with the previous results in Fig. 2(a), which rejects the low-level features. Similarly for the Deep19 model in Fig. 3(b), although the feature pool is large (18 ReLU layers), the optimal greedy results only selects the mid- or high-level filter responses.

By incorporating marginal responses from multiple lay-



(a) Caffe (b) Deep19

Figure 2. The performance trend when incorporating more f from the ReLU layer. The keyword "all" means previous fs been included as feature. A performance drop is observed when incorporating low level marginal responses.



(a) Caffe (b) Deep19

Figure 3. The performance trend when using forward selection to incorporate fs from the ReLU layer.

ers, we have achieved *multi-scale orderless* feature extraction. We term this feature "MOFeat". We should point out that for the latter analysis and experiments on other datasets, we opt to use the same feature selected by the forward selection algorithm on MIT67 data. An consistent performance boost demonstrates the data-independence of the proposed feature representation.

3.3.1 Discussions

Though inspired by [8], our computation of the *multi-scale orderless* feature is completely different from them. In [8], local patches at three scales are first extracted (roughly 50 patches per image in their setting); each patch is then feed to the entire 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 [8]:

1. [8] computes multiple feed-forward on a single image due to multiple patches. The computation of convolution is the bottleneck for this procedure. In our implementation, only one feed-forward computation is needed for one image and the activation is available at each scale.

2. [8] uses K-means clustering algorithm in the VLAD pooling, which can result in inconsistent result due to random initialization.

3. [8] 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).

4. the feature extraction procedure in [8] takes more than 20 seconds for one test image while ours only take less than 1 second.

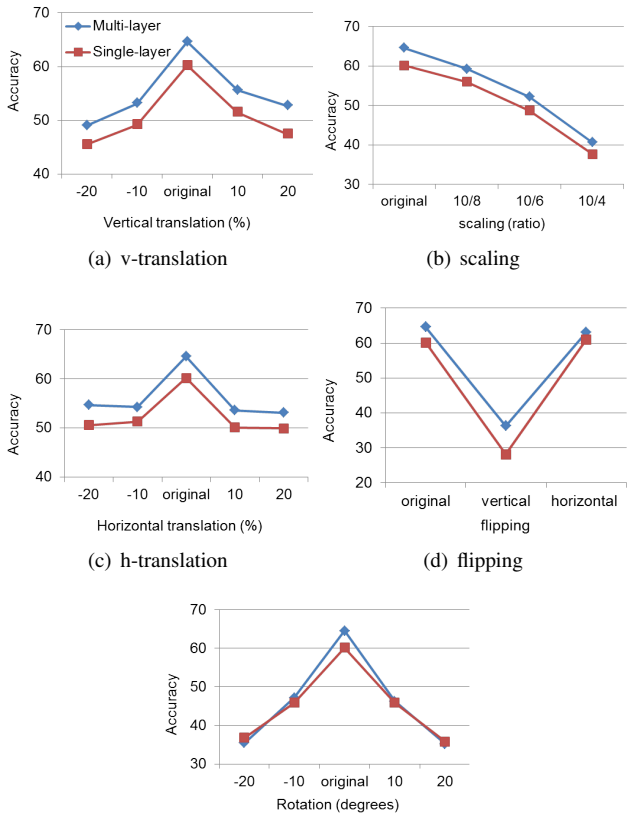
Overall, the implementation of *multi-scale orderless* feature in [8] is not only inefficient but also prone to overfitting due to a number of hyper-parameter tuning. On the contrary, our implementation is easy to compute and little parameter tuning is needed.

3.4. The Analysis of Invariance

Besides the insight on visualizing deep CNN features, [22] 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, [8] improve the invariance analysis by experiments on the SUN397 scene database [21]. Unfortunately, only the results on 4 categories are provided in their paper. The performance for the entire dataset is still unknown

To cope with the aforementioned gap and analyze the invariance of our representation, we carried out experiments on the entire MIT Indoor dataset [15] and report the average performance. We intend to compare the tolerance of the proposed multi-scale MOFeat compared with single-scale CNN features. Concretely, the single-scale CNN feature in comparison is the activation of CNN model which achieves best classification performance (not necessarily the output of the first FC layer). 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 [8]. After applying the transformations to test images, features are extracted for the proposed MOFeat and single-scale feature. The classification is conducted using the trained SVMs, and the accuracy is shown in Fig. 4. Each data point is the performance of the entire test data. As seen in Fg. 4, the transformations almost always hurt the performance except for horizontal flip. This observation is similar to the one in [8]. We also see that multi-scale feature, MOFeat, consistently out-performs the single-scale feature under every degree of all transformations. As far as the invariance is concerned, we observe the same level of performance decrease, if not more, for MOFeat compared with single-scale. This means that both features have the same level of tolerance to transformations. This immediately suggest that "training jittering", *i.e.*, including transformations in the training data, will improve

the performance. Since "training jittering" is conducted during the training of CNN model for classification [18], it is interesting that the observation of our invariance analysis shows that when using CNN for feature extraction, it is beneficial to perform another round of "training jittering" to train the classifiers.



(a) v-translation (b) scaling

(c) h-translation (d) flipping

(e) rotation

Figure 4. The classification accuracy of various transformations on test images in MIT67 data [15]. The legends shown in Fig. 4(a) is the same in the other four figures.

4. Experimental Results

In this section, we carried out experiments on benchmark dataset, SUN397 [21], MIT67 [15], and Scene15 [4] to demonstrate the consistency of our approach. We should stress that our MOFeat representation is learned on MIT67 and applied to all the experiments.

5. Experiment Setup

We follow the standard image pre-processing steps in the literature [12, 11, 18]. Since both Caffe [11] and Deep19 models [18] 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.

consistent performance improvement on all scene dataset

1. SUN397
2. MIT67
3. Scene15

6. 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

[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, 4

[4] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 4

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

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

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

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

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

[10] 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

[11] 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. 1, 4

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

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

[14] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 2

[15] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009. 2, 3, 4

[16] 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

[17] 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. 1

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

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

[20] 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

[21] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 3, 4

[22] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. 3