If We Did Not Have ImageNet: Comparison of Fisher Encodings and Convolutional Neural Networks on Limited Training Data

Christian Hentschel(✉), Timur Pratama Wiradarma, and Harald Sack

Hasso Plattner Institute for Software Systems Engineering, Potsdam, Germany
{christian.hentschel,harald.sack}@hpi.de,
pratama.wiradarma@student.hpi.uni-potsdam.de

Abstract. This work aims to compare two competing approaches for image classification, namely Bag-of-Visual-Words (BoVW) and Convolutional Neural Networks (CNNs). Recent works have shown that CNNs (Convolutional Neural Networks) have surpassed hand-crafted feature extraction techniques in image classification problems. Their success is partly attributed to the fact that benchmarking initiatives such as ImageNet in a massive crowd sourcing effort gathered sufficient data necessary to train deep neural networks with a very large number of model parameters. Obviously, manually annotated training datasets on a similar scale cannot be provided in every classification scenario due to the massive amount of required resources and time. In this paper, we therefore analyze and compare the performance of BoVW- and CNN-based approaches for image classification as a function of the available training data. We show that CNNs benefit from growing datasets while BoVW-based classifiers outperform CNNs when only limited data is available. Evidence is given by experiments with gradually increasing training data and visualizations of the classification models.

1 Introduction

Recently, approaches for image classification based on the Bag-of-Visual-Words (BoVW) model as well as its more powerful successors (e.g. Vector of Locally Aggregated Descriptors, VLAD [1] and Fisher Vector encodings, FV [2]) have been significantly outperformed by approaches based on convolutional neural networks. The fact that BoVW encodings are largely based on handcrafted image descriptors was identified as a major drawback: Typically, a vector space representation of an image is computed by extracting local features (usually gradient based, e.g. SIFT [3]) at densely sampled image regions and summarizing these features into a global image descriptor (e.g. a histogram of vector quantized local features). Quantization of the local region descriptors (e.g. by using KMeans or Gaussian Mixture Models) is actually the only step where the features are adjusted to the training data. All other parameters (e.g. the number of bins

G. Bebis et al. (Eds.): ISVC 2015, Part II, LNCS 9475, pp. 400–409, 2015.
DOI: 10.1007/978-3-319-27863-6_37

and orientations in the SIFT gradient histogram) are kept fixed. Visual concept models are learned on top of these feature representations (typically linear and non-linear support vector machines are applied).

On the other hand, Convolutional Neural Networks combine several layers of non-linear feature extractors whose weights are trained directly on the image data at hand. Feature extraction and visual concept model training is performed in a single step of training one neural network. The large number of model parameters allows for a more fine-grained adjustment of the image features but also comes at the cost of increased training complexity: deep neural networks can only be reasonably trained on highly parallelized hardware (GPUs are exploited in most cases) and a large number of model parameter demands for large training data in order to avoid overfitting of the model. Especially the latter aspect represents a significant limitation: Assembly of (manually annotated) training data is considered a costly and time consuming process. On the other hand, BoVW-based approaches have shown reasonable classification accuracy even in scenarios with very little training data available.

In this paper, we therefore analyze the impact of varying training dataset size on the achieved classification performance using either BoVW and CNNs. By gradually increasing the number of available training data we are able to estimate a decision threshold based on which users can decide, which method to favor. Furthermore, we analyze the learned models by visualizing their classification accuracy in selected scenarios. The results give insights into the differences of the respective approaches in terms of adaptation to the training data.

This paper is structured as follows: In Sect. 2 we briefly review the related work. We describe the setup of our experiments, the employed BoVW descriptors as well as the architecture of the CNNs used in Sect. 3. Furthermore, we present the various training and test datasets used throughout our experiments. Section 4 provides a detailed analysis of the obtained results. Heat map visualizations computed for some of the trained models give further insights into how increased number of training images is used by the respective approach to learn the depicted concept. Finally, Sect. 5 concludes our paper and gives a short outlook to future work.

2 Related Work

In the first two years of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [4] the leading participating teams all used Bag-of-Visual-Words derived approaches such as Fisher Vector (or the closely related Super Vector) encodings. In 2012, the authors in [5] proposed an approach based on Deep Convolutional Neural Networks that outperformed the BoVW competitors by a large margin. The neural network architecture presented has more than 60 million parameters which made training on a GPU a necessity (training on two GTX 580 GPUs took between 5 to 6 days) and which makes the approach prone to overfitting – only attenuated by the large number of training images available in ILSVRC (1.2 million images were manually assigned to 1,000 categories).

By 2014 almost all participating teams had adopted CNN-based approaches. One straightforward way of improving the performance of deep neural networks is by increasing their size – the winning team in the 2014 ILSVRC used a CNN with up to 144 million parameters [6] – which, however, likewise increases the risk of overfitting. Hence, several efforts have focused on exploring approaches that work in low training data scenarios as well.

One promising idea are CNN models *pre*-trained on a larger dataset (e.g., ILSVRC) and *fine-tuned* on the new target outputs [7–9]. Another approach uses the pen-ultimate layer of the pre-trained CNN as a powerful feature descriptor and then applies machine learning (e.g. linear Support Vector Machines) to train the target models (e.g. [10,11]). However, both approaches rely on the assumption that the data used to train the initial CNN exhibits features similar to the data that is actually supposed to be classified. In [9], Wei et al. compare both methods to a Fisher Vector based implementation and report the superior performance of the CNN approaches in a multi label classification experiment conducted on the PASCAL VOC-2007 dataset [12]. Similarly, the authors in [7] conclude that using CNNs as feature descriptors (pre-trained on ILSVRC2012 data) and SVMs as linear predictors outperform Improved Fisher Encodings when tested on the VOC and Caltech [13] datasets. Both datasets – PASCAL VOC and Caltech – show real world objects and scenes and should be considered as visually similar to the ILSVRC dataset (with some images from the Caltech datasets being also present in the ILSVRC 2012 dataset).

Using a CNN without pre-training immediately on a comparatively small dataset such as the Caltech datasets, leads to significantly worse results than BoVW-based classifiers as reported in [11], which underlines the above mentioned necessity of large datasets for CNN training. Furthermore, this makes BoVW like implementations a competitive candidate for scenarios with low amounts of training data.

In this paper, we therefore analyze the impact of incrementing training set sizes on the classification performance of FV and CNN based approaches for image classification. We directly train both approaches on the datasets, i.e. without relying on pre-trained CNN models, in order to avoid a bias induced by dataset similarities. Thus, the reported results are valid even in scenarios where the data to be classified differs strongly from the ImageNet datasets typically used to pre-train CNNs. Our assumption is that FV are better candidates when limited training data is at hand.

3 Experimental Setup

In this section we will detail the two image representation and training approaches that we compared in our experiments: linear Support Vector Machines trained on (improved) Fisher Vector encodings as well as Convolutional Neural Networks. Furthermore, we describe the dataset employed and the different experiments conducted.

3.1 Improved Fisher Encodings and Linear Predictors

In [14] the authors compared different local feature encodings in a large scale experiment and conclude the superior performance of FV encodings which we therefore adopted in our experiments as well. Consistent with the FV implementations proposed in [2], our approach starts by extracting SIFT descriptors [3] at a dense grid with a stride of 4 pixels at 7 different scales. We use the implementation provided by [15] which uses triangular feature reweighting (as opposed to Gaussian feature weighting proposed by Lowe). Following [2] we decorrelate and reduce the original feature dimensions from $d = 128$ to $d = 80$ by means of Principal Component Analysis (PCA). We further enhance the local descriptors by spatially extending the features with the (normalized) sampling point's coordinates, yielding a $d = 82$ dimensional local descriptor

A FV encoding is then obtained by first computing the Gaussian Mixture Model (GMM) with $k = 256$ components on a random subset of $n = 256,000$ local descriptors equally selected from all training images. Subsequently, each local descriptor of an image is soft-quantized using the obtained mixtures and first and second order statistics between the descriptor and its Gaussian cluster are accumulated. Finally, the *improved* version of FV (IFV, as suggested by the authors in [2]) applies signed square-rooting to the individual components of the encoding followed by a $\| \cdot \|_2$ normalization.

Usually, visual concept models are trained based on these global feature representations using Support Vector Machines ([7,14]). Our implementation learns a linear SVM per image class (using a *one-vs-rest* pattern) by minimizing the hinge loss function. While in theory the regularization C hyperparameter should be optimized using cross validation, we fix it to $C = 10$ in order to reduce training time. Empirical results in small test scenarios have shown no significant disadvantage incurred from this simplification, however, clearly this leaves room for future improvements.

3.2 Convolutional Neural Networks

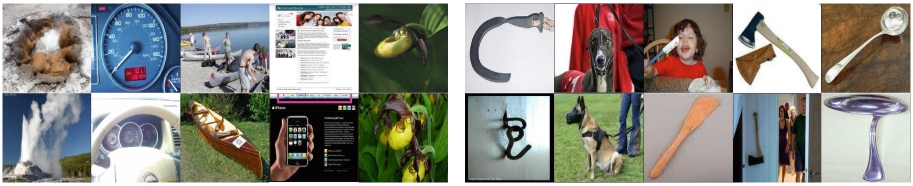
The CNN-based classifiers follow the architecture as proposed by Krizhevsky, et al. in [5] with some minor modifications. These modifications address the sequence of pooling and normalization layers (compared to the original model we flip the order, i.e. pooling is applied before normalization) and mainly help to speed up the forward run without sacrificing the accuracy. The remainder of the architecture is left unchanged: The network consists of five convolutional layers activated by a Rectified Linear Unit (ReLU) and followed by a max pooling layer (applied to $1^{st}, 2^{nd}$ and 5^{th} convolutional layer). Local Response Normalization is applied after the 1^{st} and 2^{nd} pooling layer. Layers 6 to 8 are fully connected layers and a softmax layer computes a probability for each target class.

Our implementation uses the Caffe framework [16]. Different from [5], we train the model on a Tesla K20X GPU with 6GB of memory (instead of using two independent GPUs with less memory). Following Krizhevsky, et al., every image is resized to 256×256 pixels and the center crop (224×224) is used as input image for the model. Additionally, mean subtraction – obtained by averaging the pixel values from all training images – is carried out for each input image. No further data augmentation is applied in this experiment since it was reported to contribute only slightly to the results.

3.3 Dataset

Since our primary goal is to analyze the impact of trainingset size on the two competing approaches – IFV and CNN – we had to make sure, to provide enough training data for CNN to be able to show its true power. We therefore opted for the ILSVRC 2012 training and validation datasets, which likewise provides comparability to other experiments.

In order to reduce the overall training time, we decided to limit our experiments to train and test models for only 10 out of the entire 1,000 classes provided in the ILSVRC. Considering the mean error from the top 5 predictions from all submissions to the 2012 ILSVRC[1], we took the 5 best and worst performing classes respectively yielding a total number of 12,424 training and 500 test images. Figure 1 shows example images for each class.



(a) Best performing classes

(b) Worst performing classes

Fig. 1. Example images for best and worst performing classes according to the 2012 ILSVRC submissions. From left to right: (a) *geyser*, *odometer*, *canoe*, *website*, *yellow lady's slipper*; (b) *hook/claw*, *muzzle*, *spatula*, *hatchet*, *ladle*.

While keeping the test data fixed, we conducted 7 individual training runs by selecting between 5 % and 100 % of the original data, uniformly distributed over all classes. In order to test the impact of additional negative data, in two further tests, we added training images from 90 and 190 classes randomly selected from the remaining categories provided in the ILSVRC 2012 dataset. Since our linear predictors in the IFV approach were trained in a one-vs.-rest fashion, we simply added the additional images to the negative sets. In the CNN-based training scenario, however, in order to avoid problems arising from imbalanced datasets, we actually trained a total of 100 and 200 classes. Test results were always evaluated based on the achieved scores for the initial set of 10 classes. Just like in the original 10-class scenario, we generated uniformly sampled subsets of the

[1] See http://image-net.org/challenges/LSVRC/2012/ilsvrc2012.pdf for more information.

100- and 200-class training sets as well. Table 1 provides and overview over the respective number of classes and available training images. In total, we trained and tested models based on 7 × 3 = 21 ILSVRC 2012 subsets.

Table 1. Training sets generated by taking the top and least performing classes from the ILSVRC 2012 dataset (10 classes) and subsampling the obtained train images. Further sets are obtained by adding additional 90 and 190 randomly selected classes.

No. Classes	No. of images per subset						
	5 %	10 %	20 %	40 %	60 %	80 %	100 %
10 classes	622	1,242	2,485	4,969	7,455	9,939	12,424
100 classes	6404	12,808	25,615	51,230	76,846	102,461	128,076
200 classes	12,866	25,728	51,456	102,911	154,368	205,823	257,279

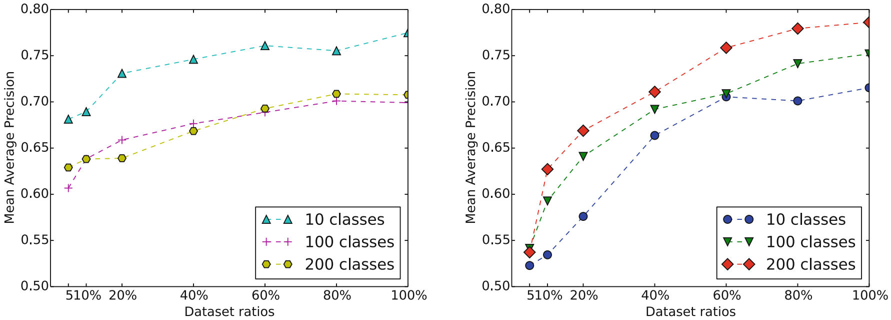
4 Analysis

Based on the trained models and the separated test set we have computed Average Precision (AP) scores for all 10 test classes and each individual run. Figure 2 shows the mean AP scores for both approaches, IFV and CNNs.

The plots show that adding more *positive* samples (i.e. going from 5 % to 100 % of the individual subsets) increases the performance for both approaches – most significant improvements occurring between 5 % to 40 %. However, incrementing the number of training samples from 80 % to 100 % contributes little to the achieved MAP score (i.e., less than 1 % MAP increase) for both models. On the other hand, while IFV based models seem to saturate at around 80 % of the entire dataset sizes (MAP even drops slightly), the CNN models seem to continue growing if going beyond 100 %.

Interestingly, when increasing the number of *negative* samples (i.e. going from 10 to 200 classes) we observe a clear drop in the achieved accuracy for the IFV based model (MAP score dropping from 76 % to 71 %) whereas CNNs benefit from the increased number of (negative) examples. In fact, the best performance by the IFV model (MAP = 76.5 %) is achieved when using the initial set of 10 classes for training whereas the best CNN-based model reaches an MAP score of 78.6 % when using 100 % of the data of the 200 class scenario. One reason may be that IFV models do not learn enough features to be able to separate classes on a more fine-grained level. When analyzing the individual per class AP scores (see Fig. 3) we observe that the best performing classes can be mostly predicted correctly by both approaches whereas the worst performing classes are equally hard to capture for CNNs as well as IFV.

Considering our initial hypothesis, we observe that our assumption of IFV outperforming CNNs in low training set scenarios holds. Especially when considering the individual per class AP scores of the best performing classes we see that IFV-based models achieve high accuracies even when provided with as little



(a) IFV (b) CNN

Fig. 2. A comparison of IFV and CNN MAP scores on the different ILSVRC subsets.

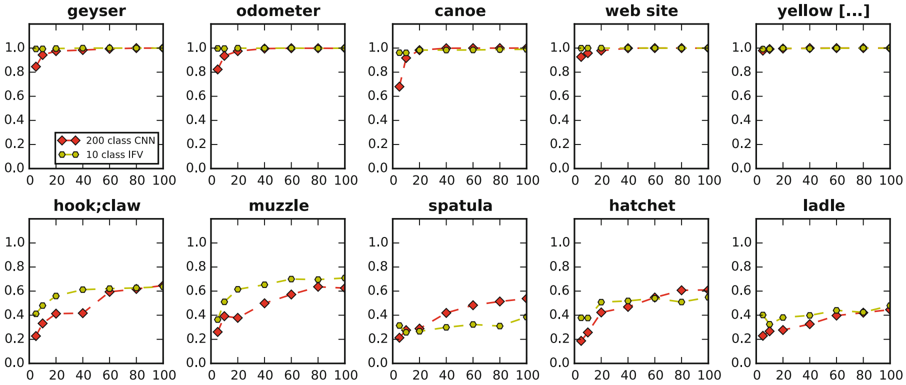
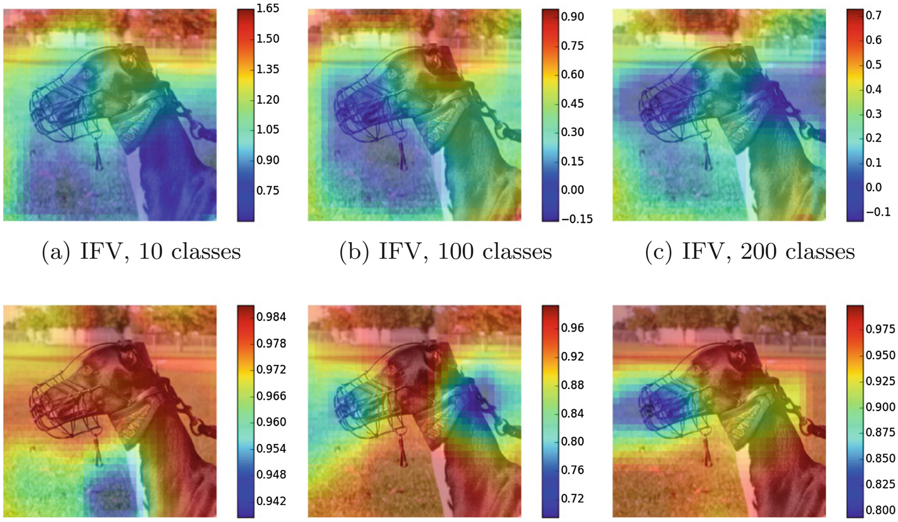


Fig. 3. Comparison of per class Average Precision scores obtained by the best performing IFV and CNN models.

as 5 %–20 % of the original data. Similarly, IFV achieves better results for most of the least performing classes, whereas CNNs need up to 60 % of the training data to catch up. This makes IFV a valid candidate when the effort to manually label large amounts of training images cannot be taken.

4.1 Model Visualization

In order to better understand what the individual models learn and how they evolve over different number of training images, we computed heat map visualizations. The method we use follows the one presented in [11], and works by stepwise occluding parts of a test image prior to classification. By that, we are able to visualize, which regions of the image have the highest impact on the overall classification score. A sliding window of 64 × 64 pixels is moved over the image pane partially setting the occluded pixel values to 0. The models trained



(a) IFV, 10 classes (b) IFV, 100 classes (c) IFV, 200 classes

(d) CNN, 10 classes (e) CNN, 100 classes (f) CNN, 200 classes

Fig. 4. An test image taken from the *muzzle* class superposed by heat maps depicting the impact of individual image regions. Blue color denotes higher impact. All heat maps are based on models trained with the maximum number of available images when using 10, 100 and 200 classes of the 2012 ILSVRC dataset (Color figure online).

in Sect. 3 are used to compute a prediction score for the true class label of that test image – each partially occluded image will give a different score. Finally, all scores are aggregated to compute the heat map.

Figure 4 presents heat maps computed using IFV and CNN models trained on 10, 100 and 200 classes (using the entire sets, i.e. 100 % of the data as we have shown that both approaches benefit from increased number of positive samples). The figure shows an image from the class *muzzle* – one of the hardest classes to be correctly predicted – superposed with the computed heat maps. Colors represent the achieved classification score when obstructing the respective region. A blue color here denotes a lower score meaning that the region is more important for the overall classification than a region superposed with a red color (denoting a higher classification score).

When comparing the heat maps of IFV and CNN based models, one can clearly see, that with increasing number of training examples, the CNN focuses more on the region of the object to be classified (*muzzle*): while in the smallest example (10 classes) the grassy region in the lower part of the image is considered important, in the 200 class setting, the CNN almost perfectly concentrates on the region depicting the *muzzle*. In contrast, the IFV-based approach right from the beginning covers the object (which might explain the higher accuracy in the 10 class scenario), however, likewise large parts of the background as well.

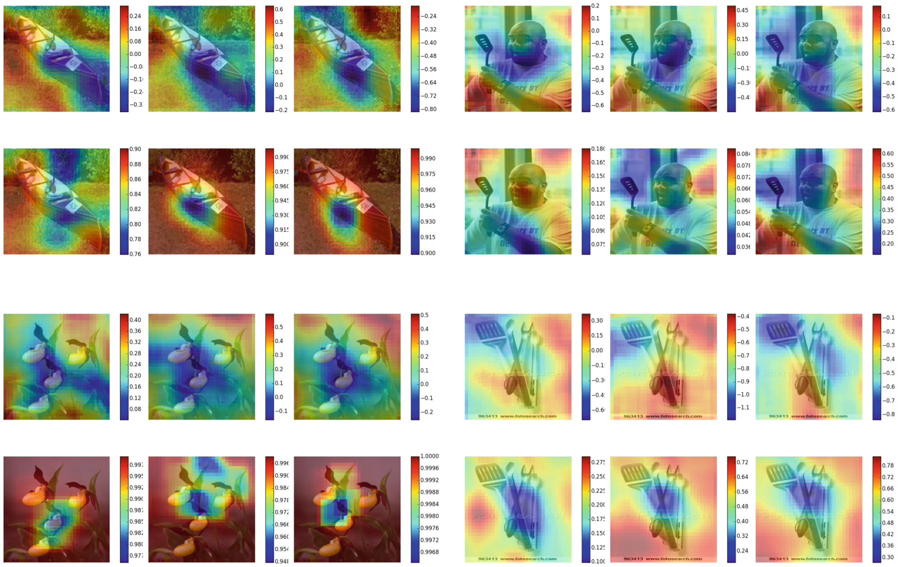


Fig. 5. Heat maps computed for some of the easier classes (*canoe* [top left]) and *yellow lady's slipper* [bottom left]) show that CNNs focus better on the depicted object when adding more (negative) data. Example images taken from one of the harder classes (*spatula* [right]) convey that both approaches have difficulties in locating meaningful regions. IFV models (top rows) and CNN model (bottom rows) have been trained on 10, 100 and 200 classes datasets (Color figure online).

While this changes slightly with increased number of classes, the IFV-based approach never reaches the precision of the CNN. Similar observations can be made when analyzing heat maps from other classes (see Fig. 5).

5 Conclusions

In this paper we evaluated the impact of growing trainingset sizes on the classification performance of Convolutional Neural Networks and Improved Fisher Vector-based image predictors. In line with our initial hypothesis, we have shown, that while CNNs largely benefit from bigger datasets, IFV is a competitive candidate when limited amounts of training data are available. Furthermore, we have presented that CNNs may use negative images to learn better feature representations. On the other hand, the precision of IFV-based models suffer from the increased diversity. Computed heat map visualizations underline our findings.

Future work will target the comparison of CNNs and IFV with feature representations obtained from CNNs pre-trained on large datasets. We aim to explore whether these representations can generalize well to significantly smaller and visually completely different datasets and how these representations compare to BoVW-like representations.

References

1. Jegou, H., Douze, M., Schmid, C., Perez, P.: Aggregating local descriptors into a compact image representation. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2010)
2. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)
3. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2. IEEE (1999)
4. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems 25. Curran Associates, Inc. (2012)
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recoginition. In: International Conference on Learning Representations (ICLR) (2015)
7. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets (2014). CoRR abs/1405.3531
8. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA (2014)
9. Wei, Y., Xia, W., Huang, J., Ni, B., Dong, J., Zhao, Y., Yan, S.: CNN: single-label to multi-label (2014). CoRR abs/1406.5726
10. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition (2014). CoRR abs/1403.6382
11. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks (2013). CoRR abs/1311.2901
12. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010)
13. Fei-Fei, L., Fergus, R.P.: One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. **28**, 594–611 (2006)
14. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods (2011)
15. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008). http://www.vlfeat.org/
16. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding (2014). arXiv preprint arXiv:1408.5093