What Image Classifiers Really See – Visualizing Bag-of-Visual Words Models

Christian Hentschel and Harald Sack

Hasso Plattner Institute for Software Systems Engineering,
Potsdam, Germany
{christian.hentschel,harald.sack}@hpi.de

Abstract. Bag-of-Visual-Words (BoVW) features which quantize and count local gradient distributions in images similar to counting words in texts have proven to be powerful image representations. In combination with supervised machine learning approaches, models for nearly every visual concept can be learned. BoVW feature extraction, however, is performed by cascading multiple stages of local feature detection and extraction, vector quantization and nearest neighbor assignment that makes interpretation of the obtained image features and thus the overall classification results very difficult. In this work, we present an approach for providing an intuitive heat map-like visualization of the influence each image pixel has on the overall classification result. We compare three different classifiers (AdaBoost, Random Forest and linear SVM) that were trained on the Caltech-101 benchmark dataset based on their individual classification performance and the generated model visualizations. The obtained visualizations not only allow for intuitive interpretation of the classification results but also help to identify sources of misclassification due to badly chosen training examples.

1 Introduction

Given a set of images and a set of concepts, the task of visual concept detection is to automatically assign one or more concepts to each of the images solely based on the visual content. An approach commonly used to solve this task is the Bag-of-Visual-Words (BoVW) model [11], where images are represented as a frequency distribution of a set of visual words (i.e. a visual vocabulary). This extends an idea from text classification where a document is described by a vector of individual word frequencies (Bag-of-Words). Image classification is then considered a learning problem of separating histograms corresponding to images of one class from those of another.

A visual word is usually described by means of local histograms of gradients (e.g. SIFT, Scale Invariant Feature Transform [9]) extracted either at specific regions of interest (keypoints) or at dense grid points. The visual vocabulary is generated by processing all local features of the training data using vector quantization approaches such as k-means and Gaussian Mixtures [12]. By assigning the local SIFT features of each image to the most similar vocabulary

X. He et al. (Eds.): MMM 2015, Part I, LNCS 8935, pp. 95–104, 2015.

vector (e.g. through nearest neighbor search), a histogram of visual word vector frequencies is generated per image. This frequency distribution is referred to as *Bag-of-Visual-Words* and provides a global image descriptor. Learning a visual concept model usually optimizes a weight vector that emphasizes different visual words depending on the classification task – very similar to learning the importance of individual words for a specific text document class. Although this approach often provides highly accurate classification results, analysis of the derived models tends to be difficult.

While in text classification each Bag-of-Words dimension corresponds to a linguistic term that carries an explicit meaning, the Bag-of-*Visual*-Words approach usually is considered a black box. This is due to the fact that visual words are much harder to interpret for a human. Each vocabulary word is a prototype for a number of local SIFT features and each SIFT feature represents local gradient distributions at a specific image region. For many image classification scenarios, e.g. computer-aided diagnosis in medical imaging, it is however crucial to provide information about how a decision is made rather than being confined to giving positive or negative classification results only.

In this work we present an approach to visualize the impact of image regions on the classification result by superposing the images with a heat map-like graphical representation of the learned visual word weights. The obtained visualization provides an intuitive way to interpret trained visual concept models by simply analyzing the image regions that contribute most (and least) to the overall result. Sources of misclassification are made explicit such as ill-chosen training examples that exhibit specific characteristics not representative for the actual concept to be classified. We present heat maps for three different classifiers, namely linear SVMs, AdaBoost and Random Forests each trained and tested on the Caltech-101 benchmark dataset [4]. Furthermore, we compare the performance of the classifiers and show that the average precision of all classifiers is comparable. This allows us to make general propositions based on the visualization.

This paper is structured as follows: Section 2 briefly reviews the related work. In Section 3 we compare the aforementioned classifiers and present the proposed approach for model visualization. Section 4 discusses the obtained results and gives an outlook to future work.

2 Related Work

While the BoVW approach for image classification has been extensively studied, considerably few works actually addresses the visualization of the trained classification models. A large body of literature focuses on the visualization of distinct regions used for local feature extraction (e.g. [2,7]). Typically, these regions are highlighted within the image by small dots or ellipses. Although this approach helps to understand which image parts were used for feature extraction it does not help to estimate to what extent each region actually influences the classification result.

Other approaches try to visualize the learned models by showing examples for visual words. In [8] the image regions assigned to the 100 vocabulary words

with maximal inter-class discrimination are presented. While this gives a notion of which parts of an image contribute to distinguish one class from another, it does not show the relative importance of each image region. The authors in [14] likewise provide binary decision scores only by using bicolored dots to visualize the decision of "important" and "not important" keypoints according to the trained model. Nevertheless, both approaches show that matching local image regions and classification importances helps to gain understanding of whether a learning algorithm identified a reasonable association between features and category.

A method that tries to identify the regions in an image that, if removed, would cause the image to be wrongly classified is presented in [14]. Although this approach provides an intuitive idea of which pixels in an image are actually important for a *true positive* classification result, it does not help to identify regions that lead to a *false negative* result, which sometimes can be even more important.

When aiming at visualizing the significance of specific image regions, heat maps have been successfully applied in the past. Being intuitive while at the same time allowing for an immediate estimation to what extent each pixel in the underlying image contributes to the conveyed information they have been proven to be powerful representations. Heat maps usually superpose the original image and use color temperature to mark important (red) and less important (blue) regions. Heat maps are often used to visualize recordings of eye trackers measuring human visual attention [3]. Quite similarly, our goal is to visualize the importance of specific image pixels, however, w.r.t. to "attention" payed by a trained visual model rather than a human. In the field of image classification, heat maps have already been used to highlight patches in images selected as important by a random forest classifier [15]. However, since the patch size used is rather large, the approach offers only a very coarse estimation of region-to-classification-result importance. In this paper, the size of each region to be assigned an importance score is only limited by the size of the support region used to extract local features.

The authors in [13] propose a semantic point detector, which is based on individual feature weights of BoVW vectors learned by a linear SVM. Although not focusing on the visualization, the authors present an approach for highlighting semantically important regions using heat maps. Similarly, we pursue heat map-like visualizations for linear SVM classifiers. Additionally, we compare these to visualizations obtained from AdaBoost and Random Forest classifiers and thus enable comparison of these different models.

3 Visualization of BoVW Models

We have computed BoVW concept models for the 101 classes of the Caltech-101 benchmark dataset [4]. SIFT features are extracted at a dense grid of $s = 6$ pixels and at a fixed scale of $\sigma = 1.0$ and k-means clustering is used to quantize the SIFT features to $k = 100$ vocabulary vectors. Thus, each image is described by a 100-dimensional histogram of visual words (see [6] for implementation details).

3.1 Model Training

Kernel-based Support Vector Machines (SVM) are widely used in BoVW classification scenarios and χ^2-kernels have shown to provide good results for histogram comparisons [16]. However, due to the kernel trick, creating a direct mapping between the individual BoVW dimensions (i.e. the visual words) and the learned model weights is infeasible. Thus, visualizing the impact of individual dimensions on the overall classification result is likewise not possible. Therefore, kernel SVM results are reported only as baseline reference and in comparison to the performance of classifiers that allow for direct inference of individual feature importances.

Linear SVMs compute a linear hyperplane to best separate positive from negative examples in the original feature space. The trained model consists of a bias and a weight vector – an unknown sample is classified by computing the dot product between the weight vector and the sample's feature vector (plus the bias). While the classification results usually tend to be inferior to the results of non-linear SVMs, the linear model allows for an immediate interpretation of the weight vector dimensions as feature importance scores.

Random Forests [1] and AdaBoost [5] are usually based on decision trees where each node correlates to a specific feature in the training set. These methods typically select features based on their capability of solving the classification problem beginning with the most perfect split, e.g. by computing the decrease in entropy of the obtained class separation. We use the *mean decrease in impurity* over all decision trees in an ensemble as direct indicator for feature importance.

Our implementation uses the scikit-learn library that provides SVM Solvers and Ensemble methods [10]. We train binary classifiers for each of the 101 concept classes in the Caltech-101 dataset using the 'background' data class as negative samples. Training is performed on 50% of the dataset images while the remainder is used for validation purposes of the obtained models. Different model parameters for each classifier are optimized using a nested cross-validation[1].

As a first step, we have computed the mean average precision (MAP) each classifier achieves on all 101 concept classes. Table 1 compares the different classification models. As expected, the performance achieved by the linear SVM model is the worst (MAP is 4–13% lower than for other models). The χ^2-kernel model on the other hand outperforms all other classifiers and underlines the superiority of this approach. However, AdaBoost and Random Forests show competitive results falling only 6–9% behind. Both ensemble methods show almost identical performance which seems logical considering they both follow a very similar approach and differ only in the way predictions of individual weak learners are aggregated. As a next step we want to visualize and compare the individual visual word importance for each of the trained classification models.

[1] The model parameters optimized are: the number of decision trees (AdaBoost, Random Forests), the maximum depth of each tree (AdaBoost), the regularization parameter (both SVM models). The kernel-SVM width parameter is set to the average χ^2-distance of all training examples[16].

Table 1. Classifier performance on Caltech-101 dataset. Mean average precision (MAP) scores are reported.

classifier	MAP
AdaBoost	0.615
Random Forest	0.593
linear SVM	0.549
χ^2-kernel SVM	0.678

3.2 Model Visualization

As discussed in Section 2 we have decided for a heat map-like representation to visualize each pixel's importance score. This requires to map the learned importance scores of visual words to pixels in the image plane. Since each feature of a BoVW-vector corresponds to a visual word in the vocabulary and the value of each feature is generated by binning local SIFT descriptors to the most similar visual words we can extend the learned importance scores to the respective SIFT descriptors. As the support regions of neighboring descriptors overlap (by default SIFT uses a support region of 16×16 pixels and our dense sampling step size is set to $s = 6$ pixels), the importance score of each pixel is set to be the maximum of all corresponding importances since our intention was to visualize the *most important* visual words.

Figure 1 shows the obtained visualization[2] for an example of the category "airplanes", correctly classified by our AdaBoost model with a confidence score of $c = 0.995$. For reasons of clarity we limit the visualized pixel contributions to the most important visual words, i.e. only the upper quartile of the importance scores obtained per visual word are shown. Darker areas mark more important regions.

Similarly, we have computed the visualizations for Random Forests and linear SVM (see Fig. 2 for a comparison of all three models). Again, we restrict the highlighted regions to the upper quartile of the most important visual words. The visualization of the SVM model differs in that since SVMs produce negative as well as positive weights, we visualize them using different colors (blue for negative weights, red for positive) and select the upper quartile of the most important positive weights as well as the most important negative weights.

Comparison of all three visualizations confirms the closeness of AdaBoost and Random Forests. Both ensemble models produce almost identical heat maps that differ mainly in the absolute values of the respective importance scores. Surprisingly, the visualization of the trained SVM model is also very similar to those of AdaBoost and Random Forests which could explain why the classification performance of SVM and ensemble methods are comparable. Please note that

[2] All figures are best viewed in color and magnification.

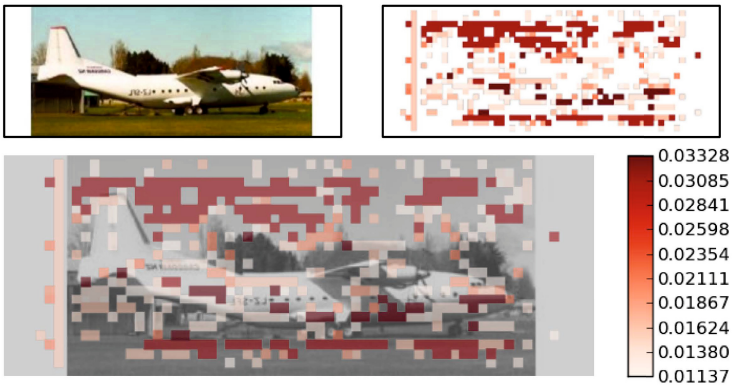


Fig. 1. Visualization of feature importances of the AdaBoost classifier trained for the category "airplanes". Top left: original image. Top right: heat map of the upper quartile of the learned feature importances. Bottom: Original superposed by the semi-transparent heat map.

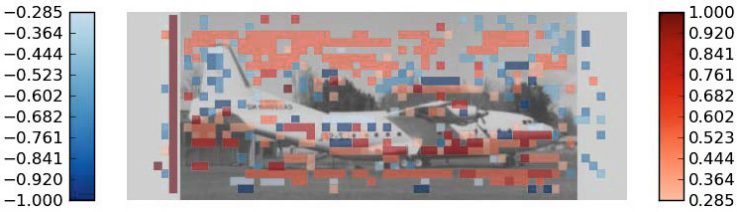
regions which have been assigned a high *negative* importance weight (color coded in blue) have likewise been selected by the ensemble methods as important.

When analyzing the visualizations with regard to the ability of explaining the classification results, the figures immediately convey that considerably few important regions actually coincide with pixels that belong to the airplane object. All three models assess the sky above as well as the grassy ground below the airplane as important features for classification of airplanes. While this seems reasonable ("airplanes are objects surrounded by sky"), it likewise means that other images with large sky-like areas will have a high chance of being falsely classified as "airplanes" as well (e.g. "birds").

A second aspect that also becomes immediately apparent due to the visualization is that most photos of airplanes carry a more or less dominant white frame that surrounds the actual photo (in fact only 108 out of 800 images in the Caltech-101 dataset annotated as "airplanes" are not surrounded by a more or less prominent white border). While all three classification models have correctly learned this specificity by selecting border pixels among the most important (upper quartile) features it represents most likely an unwanted characteristic and classifying photos of airplanes that do not exhibit a white border will most likely show inferior classification results. In order to validate this assumption, we have split the testing data into those images having a border ($|I_{border}| = 341$) and those that do not ($|I_{\neg border}| = 59$), applied the models (trained on both types) and computed the average precision scores separately. As expected, scores dropped significantly by up to 24% (see Table 2).

This artifactual cue of the Caltech-101 dataset is revealed in other categories as well [16]. Figure 3 (left) shows an example visualization of the AdaBoost classifier for a correctly classified image (confidence score $c = 0.85$) in the category "trilobite". All images of this category exhibit a "corner" artifact resulting from

(a) Original Image



(b) linear SVM

(c) Random Forests

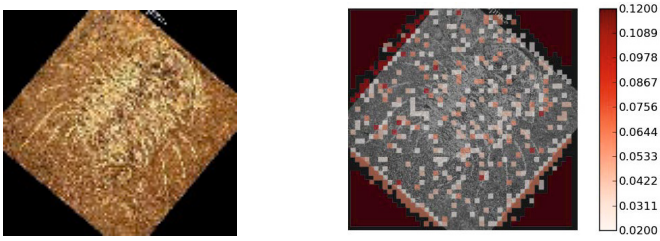
(d) AdaBoost

Fig. 2. Visualization of feature importances of three different classification models

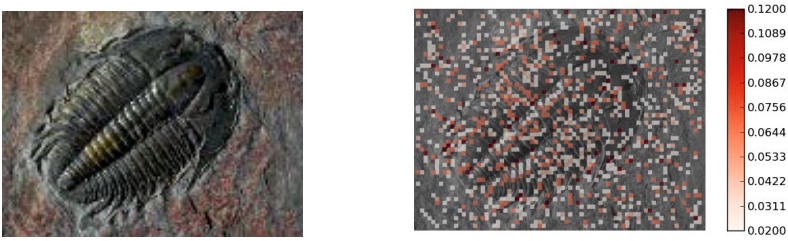
Table 2. Performance of classifiers applied to images of the "airplanes" category with and without white border

classifier	AP I_{border}	AP $I_{\neg border}$
AdaBoost	0.99	0.90
Random Forest	0.98	0.83
linear SVM	0.97	0.73

artificial image rotation leaving a black background – a strong cue that all classifiers pick up as a highly important feature. When applied on an example image of a trilobite not showing this artifact (taken from Google images), AdaBoost fails as expected (Fig. 3, right, $c = 0.577$, Random Forests behave similarly). The linear SVM classifier correctly classifies the image, however with a rather low confidence value.



(a) True positive



(b) False negative

Fig. 3. Visualizations of the AdaBoost model trained on images of the category "trilobite". *Top:* Image taken from Caltech-101 with a prominent rotation artifact. *Bottom:* Image without artifact classified negative by the same model.

4 Summary

In this paper we have presented an approach for an intuitive visualization of different Bag-of-Visual-Words models. We have trained three different classifiers,

linear SVM, Random Forests and AdaBoost, and compared the performance of these classifiers based on the Caltech-101 benchmark dataset. The visualization we propose uses a heat map-like representation of the importance scores of visual words as learned by a classifier. By providing examples from two different categories we have shown the effectiveness of our visualization. In both cases, deficits in the models' ability to generalize from the training examples as well as peculiarities within the Caltech-101 training material became immediately apparent by looking at a single testing instance enabling the user to understand *how* a decision is made.

Future work will on the one hand focus on the BoVW feature representations. Often, a multi-scale approach is taken, that extracts SIFT features at various scales in order to obtain scale-invariant descriptors. It will be interesting to see to what extent different scales will be reflected in the visualizations. Furthermore, we intend to correlate heat maps with saliency maps generated by human eye movement data. Possible similarities between image classifiers and the human perception might help to further improve image classification.

References

1. Breiman, L.: Random forests. Machine Learning (2001)
2. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C., Maupertuis, D.: Visual Categorization with Bags of Keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV (2004)
3. Cutrell, E., Guan, Z.: What are you looking for?: an eye-tracking study of information usage in web search. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems
4. Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop (2004)
5. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. 55(1) (August 1997)
6. Hentschel, C., Gerke, S., Mbanya, E.: Classifying images at scene level: Comparing global and local descriptors. In: Detyniecki, M., García-Serrano, A., Nürnberger, A., Stober, S. (eds.) AMR 2011. LNCS, vol. 7836, pp. 72–82. Springer, Heidelberg (2013)
7. Jiang, Y.G., Yang, J., Ngo, C.W., Hauptmann, A.G.: Representations of Keypoint-Based Semantic Concept Detection: A Comprehensive Study. IEEE Transactions on Multimedia 12(1) (2010)
8. Jurie, F., Triggs, B.: Creating efficient codebooks for visual recognition. In: IEEE International Conference on Computer Vision, ICCV 2005 (2005)
9. Lowe, D.G.: Object recognition from local scale-invariant features. In: IEEE International Conference on Computer Vision, ICCV 1999 (1999)
10. Pedregosa, F., Varoquaux, G., Gramfort, et al.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12 (2011)
11. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: IEEE International Conference on Computer Vision, ICCV 2003 (2003)

12. Snoek, C.G.M., Worring, M.: Concept-Based Video Retrieval. Foundations and Trends in Information Retrieval 2(4) (2009)
13. Yang, K., Zhang, L., Wang, M., Zhang, H.: Semantic point detector. In: Proceedings of the 19th ACM ..., pp. 1209–1212 (2011), http://dl.acm.org/citation.cfm?id=2071976
14. Yang, L.Y.L., Jin, R.J.R., Sukthankar, R., Jurie, F.: Unifying discriminative visual codebook generation with classifier training for object category recognition. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition (2008)
15. Yao, B., Khosla, A., Fei-Fei, L.: Combining randomization and discrimination for fine-grained image categorization. In: CVPR 2011 (2011)
16. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. International Journal of Computer Vision 73(2) (2006)