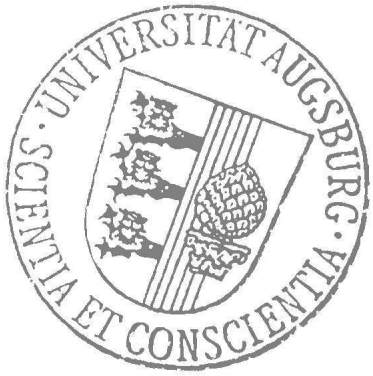
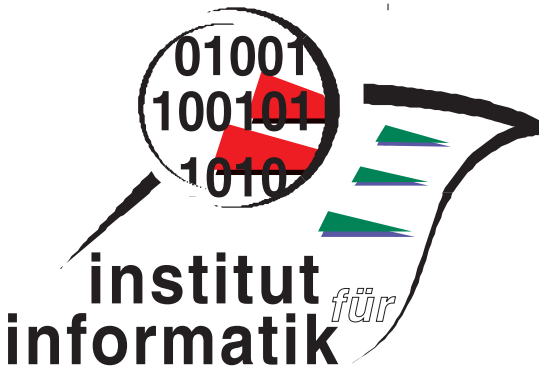
Universität Augsburg



Learning to Represent Multiple Object Classes on a Continuous Viewsphere

Johannes Schels, Joerg Liebelt
Rainer Lienhart

Report 2012-07 Juli 2012



Institut für Informatik
D-86135 Augsburg

Learning to Represent Multiple Object Classes on a Continuous Viewsphere

Johannes Schels, Joerg Liebelt
EADS Innovation Works
München, Germany
{johannes.schels,joerg.liebelt}@eads.net

Rainer Lienhart
University of Augsburg
Augsburg, Germany
lienhart@informatik.uni-augsburg.de

Abstract

Existing work on multi-class object detection usually does not cover the entire viewsphere of each class in a continuous way: object classes from different viewpoints are either discretized into a few sparse viewpoints [12], or treated as entirely separate object classes [20]. In the present work, we describe an approach to multi-class object detection which allows sharing parts between different viewpoints and several classes while also learning a dense representation for the entire viewsphere of each class. We describe three learning approaches with different part sharing strategies in order to reduce the computational complexity of the learnt representation. Our approach uses synthetic training data to achieve a dense viewsphere coverage which also allows to perform object class and 3D pose estimation on single images.

1. Introduction

It is estimated that humans are familiar with tens of thousands of different object classes [3]. In computer vision, a long-term objective is replicating this fundamental human ability. However, learning and recognizing multiple object classes from arbitrary viewpoints is still in its infancy. Several approaches address the problem of viewpoint-independent object class detection [14, 19, 22, 23, 25] or multi-class object detection [9, 12, 20, 24, 21, 26]. Most of these approaches consider these two problems in isolation, i.e. either a viewpoint-independent representation of an object class is built [14, 23, 25] or multiple object classes are trained from discrete viewpoints [12, 20, 26]. Based on a decomposition of each object class into parts, in the present work we propose and evaluate three novel learning strategies to represent multiple object classes on a continuous viewsphere: an independent, a joint, and a sequential learning strategy. Our experiments show that the sequential learning strategy achieves the best result with respect to 2D localization performance and flexibility during the training process and thus could be suitable for learning multiple object classes from arbitrary viewpoints on a larger scale. All our proposed learning strategies rely on a part-based object class detection approach, where a database of synthetic 3D object models serves as the only positive training source. The 2D localization performance of our learning strategies is evaluated on different testsets which consist of images from the 3D Object Category dataset [22] and the PASCAL VOC2006 dataset [7].

In general, there are three different strategies for learning to represent multiple object classes: first, an independent learning which trains each class separately from all other classes. Second, a joint learning [26] which trains all classes simultaneously, and third, a sequential learning which trains one object class after the other [20]. In [26] multiple classes are trained jointly based on boosted decision stumps to find common features. A variation of [26] is proposed in [20] which enables the sequential addition of a new class without retraining the previously learnt classes. In the context of learning object classes from a small number of training samples [1, 2, 9, 17, 24] sequential learning is also termed *knowledge transfer* or *one-shot learning*. In [9] the priors of probabilistic models are adapted by a few training samples to represent new classes and in [1] a template from a previously trained class is used to regularize the training of a novel object class. [2] replaces features from known classes with ones from a new but similar class. [17] uses prior information about a novel class in order to assist a feature selection process. [24] proposes a shape-based model which enables full or partial knowledge transfer. All mentioned approaches have in common that they either learn the classes from just a few discrete viewpoints [9, 24] or they perform knowledge transfer within visually very similar classes, as in [1, 2, 17]. In contrast, in this paper we propose three novel learning strategies to represent multiple, potentially dissimilar object classes on a continuous viewsphere. To this purpose, we rely on the approach of [23] where a continuous object class representation is learnt based on a database of synthetic 3D models. Specifically, we extend their approach and propose three novel strategies to represent multiple object classes on a

continuous viewsphere. Our work is related to [12] where a hierarchial framework is used to propose and compare different types of multi-class learning strategies. In contrast to our work, [12] restricts possible synergies among the object classes to a few discrete viewpoints.

In the remainder of this paper in Section 2 we first describe the approach of [23], then in Section 3 we propose different learning strategies which are evaluated in Section 4, and we conclude the paper with an outlook on future work in Section 5.

2. The Viewsphere Model

For the proposed learning strategies we rely on the synthetically trained part-based model of [23] which we briefly summarize in this section. Further details are given in [23]. For simplification we term this model the viewsphere model. The following training steps are necessary to build a viewsphere model for a specific object class c:

Training Data: The viewsphere model derives its positive training images exclusively from a database of synthetic 3D models and its negative images from the VOC2006 dataset [7]. Each 3D model is rendered from many viewpoints which cover the entire viewsphere densely. This rendering is performed once in front of a black background which we term the pure training images I^c_{pure}, and once in front of randomly selected images from the negative dataset which we term the validation images I^c_{val}.

Generating a Pool of Parts: HOG-features [6] of different cell layouts are computed densely on each pure training image. Affinity propagation [13] is applied to all features of each HOG cell layout, collected from the pure training images. A standard bootstrapping procedure is used to train a linear SVM, based on the features assigned to a cluster. Finally, we obtain a pool P^c of parts where each part is represented by a linear SVM classifier.

Selecting the Most Informative Parts: The pool P^c contains a large number of non-informative or redundant parts, due to symmetries and self-similarity. In this training step, a subset of N^c object parts is selected by ranking the informativeness of each part w.r.t. a positive and a negative image set with an entropy-based measure [27] as follows: as we intend to separate an entire object class from the background, the pure training images I^c_{pure} are chosen as the positive image set and the negative training examples from the VOC2006 dataset are chosen as the negative image set. Subsequently, the optimal detection threshold of each object part from the pool P^c is determined by maximizing the mutual information [5] of the occurrence of a part in the positive and negative image set as follows: an indication function p of an object $part$ in association with a detection threshold θ is defined as a binary variable

$$p(I,\theta) = \begin{cases} 1, \text{ if } s_{max}(I, part) \geq \theta \\ 0, \text{ otherwise} \end{cases} . \quad (1)$$

Here s_{max} is the maximum score of the object part classifier (i.e. the linear SVM) in an image I. In addition, a binary class variable K is defined where $K(I) = 1$ if the image I belongs to the positive set of images and 0 otherwise. Between these two binary variables the mutual information $MI(p(\theta); K)$ is defined as

$$MI(p(\theta); K) = H(K) - H(K \mid p(\theta)) \quad (2)$$

with $H(x)$[1] and $H(x \mid y)$[2] being the marginal and the conditional entropy. The optimal detection threshold θ^{opt} for each object part can be determined from

$$\theta^{opt} = \underset{\theta}{\operatorname{argmax}}[MI(p(\theta); K)] \quad (3)$$

which results in the maximal mutual information MI^{max}. After the optimal detection threshold for each object part has been determined an optimal subset of N^c parts from the pool P^c can be selected iteratively (for further details see [23]). Such a subset contains a maximum of information regarding the chosen sets of positive and negative images.

Modeling a Dense Grid of Spatial Part Layouts: A spatial layout model which describes the spatial occurrence of a small subset of M^c object parts ($M^c \subseteq N^c$) for each defined viewpoint on the entire viewsphere is built to provide initial object hypotheses. The occurrence of the object parts is modeled on the pure training images I^c_{pure} by a mixture of Gaussian distributions [4] and the resulting spatial layout models can be efficiently evaluated [10].

Learning the Global Object Class Appearance: The spatial layout models for all defined viewpoints on the viewsphere allow generating a set of object hypotheses. In order to rank these hypotheses in a consistent way, all generated hypotheses on the validation images I^c_{val} are resized to the training scale and converted into a spatial pyramid representation [15]. Based on this spatial pyramid representation which encodes the detection scores of all N^c selected parts a non-linear SVM with an intersection kernel [16] is trained to describe the entire object class c.

3. Multi-Class and Multi-View Learning Strategies

In this section, we propose three novel learning strategies, based on the training steps of the viewsphere model

[1] $H(x) = -\sum_x p(x) \log(p(x))$

[2] $H(x \mid y) = -\sum_{x,y} p(x,y) \log(p(x \mid y))$

of Section 2, to represent C object classes on a continuous viewsphere: an independent, a joint, and a sequential learning strategy. By relying on the part-based representation of the viewsphere model, we follow common multi-class approaches which also decompose each object class into parts [2, 21, 26]. Pseudo-code for all three learning strategies is given in Table 1.

3.1. Independent Learning

First, we propose an independent learning of C object classes (see Table 1 (top)) as a standard and base strategy to represent multiple object classes: based on the pure training images I^c_{pure} and the validation images I^c_{val} each object class c is trained independently from all other classes. For each object class c a pool P^c_I of independent and class-specific parts is generated and N^c_I parts from P^c_I are selected. Based on a subset of M^c_I parts ($M^c_I \subseteq N^c_I$) a dense grid of class-specific spatial layout models is established and the global appearance of the N^c_I selected parts on the validation images I^c_{val} is learnt for each class.

Training each object class independently from all other classes comes with the advantage that a new object class can easily be added without retraining the previously learnt object classes [12]. However, parts are not shared among object classes which implies that the computational complexity of the overall representation grows linearly with the number of object classes, as shown in [26].

3.2. Joint Learning

The second learning strategy is a joint learning of C object classes (see Table 1 (center)): based on the pure training images of all object classes $I_{pure} = \{I^1_{pure}, \ldots, I^C_{pure}\}$ a joint pool P_J of object parts is generated and N_J object parts, which cover all object classes at once, are selected from P_J. Subsequently, a dense grid of spatial layout models is established for each object class c by using the pure training images I^c_{pure} and a subset of M_J parts ($M_J \subseteq N_J$). Finally, the global appearance of all object classes is jointly learnt into one non-linear SVM with an intersection kernel (cf. Section 2) using the N_J selected object parts to encode the validation images of all object classes $I_{val} = \{I^1_{val}, \ldots, I^C_{val}\}$ with a spatial pyramid representation.

The properties of the joint learning strategy are opposed to the properties of the independent learning strategy: for joint learning, adding a new object class to an already existing multi-class representation is not possible without retraining all previously trained object classes from scratch. As shown in [26], a joint learning of multiple object classes normally reduces the computational complexity of the overall representation, by finding common object parts that can be shared across different object classes. In the following section, we introduce a sequential learning strategy which combines the advantages of both the independent and the joint learning strategy [12].

Independent Learning of C object classes:
for $c := 1$ **to** C
- generate a pool P^c_I of parts from the pure training images I^c_{pure}
- select the N^c_I most informative parts from P^c_I with an entropy-based measure
- model a grid of spatial layout models based on the pure training images I^c_{pure}
- learn the global appearance based on a spatial pyramid representation of all N^c_I parts on the validation images I^c_{val}

end

Joint Learning of C object classes:
- generate a common pool P_J of parts from the pure training images $I_{pure} = \{I^1_{pure}, \ldots, I^C_{pure}\}$
- select the N_J most informative parts from P_J with an entropy-based measure

for $c := 1$ **to** C
- model a grid of spatial layout models based on the pure training images I^c_{pure}

end
- learn the common global appearance based on a spatial pyramid representation of all N_J parts on the validation images $I_{val} = \{I^1_{val}, \ldots, I^C_{val}\}$

Sequential Learning of C object classes:
- generate an initial pool P^1_S of parts from the pure training images I^1_{pure}

for $c := 2$ **to** C
- perform *knowledge transfer* from P^{c-1}_S to the object class c in oder to determine the remaining training images I^c_{remain} which are not covered by the pool P^{c-1}_S
- generate a pool $\overline{P}^c_S$ of parts with the remaining training images I^c_{remain}
- merge part pools $P^c_S = P^{c-1}_S \cup \overline{P}^c_S$

end
- select N_S parts from P^C_S

for $c := 1$ **to** C
- model a grid of spatial layout models based on the pure training images I^c_{pure}

end
- learn the common global appearance based on a spatial pyramid representation of all N_S parts on the validation images $I_{val} = \{I^1_{val}, \ldots, I^C_{val}\}$

Table 1. Three different learning strategies based on the training steps of the viewsphere model of Section 2: an independent (top), a joint (center) and a sequential (bottom) learning strategy of C object classes. For a single object class ($C = 1$) all learning strategies reduce to the viewsphere model. The term *knowledge transfer* stems from the machine learning literature [8] and is described w.r.t. this work in Section 3.3.

3.3. Sequential Learning

In this work, the knowledge of an object class can be defined as the appearance and the spatial arrangement of the selected object parts (see Section 2). By learning one object class after the other, we are able to perform *knowledge transfer* [8] from previously trained object classes to novel classes. By finding common knowledge across different object classes we reduce the computational complexity of the overall representation. In contrast to joint learning, it is possible to learn a new object class without retraining the previously learnt object classes. In the following paragraph, a knowledge transfer algorithm is proposed to transfer the knowledge which is captured by the appearance of the parts.

Knowledge Transfer: Starting point of the knowledge transfer algorithm is a pool P_S^{c-1} of previously trained object parts and the pure training images I_{pure}^c of a novel object class c. We intend to transfer knowledge from P_S^{c-1} to c in order to reduce the number of the pure training images I_{pure}^c and to determine the training images I_{remain}^c of the novel object class ($I_{remain}^c \subseteq I_{pure}^c$) which are not yet covered by the pool P_S^{c-1}. We term these images I_{remain}^c, the remaining training images. To this purpose, we calculate for each part from the pool P_S^{c-1} a joint mutual information MI_{joint} as follows

$$MI_{joint} = \frac{1}{C_{pre}} MI_{all}^{max} + (1 - \frac{1}{C_{pre}}) MI_{novel}^{max} \geq \alpha. \quad (4)$$

Here MI_{all}^{max} is the maximal mutual information on all pure training images, i.e. the positive image set consists of both the pure training images of all previously trained object classes and the pure training images of the novel object class. MI_{novel}^{max} is the maximal mutual information on the pure training images of the novel class. See Equations (1-3) for calculating the maximal mutual information in conjunction with the optimal detection threshold. C_{pre} is the number of previously trained object classes and α is a threshold which we term the information threshold. The joint mutual information of Equation 4 takes into account that with an increasing number of previously trained classes an object part is less likely to contain knowledge of all classes simultaneously. However, with an increasing number of pre-trained classes Equation 4 requires that a part must provide at least knowledge of the novel object class. Finally, parts which contain information above the information threshold α are preserved. We term these parts the transferable object parts. For each transferable object part it is possible to determine its visibility in a pure training image of the novel class. To this purpose, we use the corresponding indication function (cf. Equation 1) of MI_{all}^{max} to determine if the maximum detection score of the corresponding SVM classifier is above the optimal detection threshold. We require that at least L^3 transferable object parts are visible in an image to remove this image from I_{pure}^c and finally we determine the remaining training images I_{remain}^c of a novel class c which are not yet covered by the pool P_S^{c-1}.

Pseudo-code for the sequential learning strategy is given in Table 1 (bottom): based on the pure training images of the first class I_{pure}^1 an initial pool P_S^1 of object parts is generated and the following procedure is sequentially performed on the remaining classes ($2 \leq c \leq C$): knowledge transfer is performed from P_S^{c-1} to the novel object class c in oder to determine the remaining training images I_{remain}^c. The remaining training images I_{remain}^c are used to generate a pool $\overline{P}_S^c$ of parts and subsequently the pools P_S^{c-1} and $\overline{P}_S^c$ are merged to P_S^c. Finally, N_S parts are selected from the final pool P_S^C. For each object class c a dense grid of spatial layout models is established, by using the pure training images I_{pure}^c and a subset of M_S parts ($M_S \subseteq N_S$). Just as for the joint learning strategy, the global appearance of all object classes is jointly learnt into one non-linear SVM with an intersection kernel, by using the validation images of all classes $I_{val} = \{I_{val}^1, \ldots, I_{val}^C\}$ and the N_S selected object parts.

4. Experimental Results

In this section, we outline the experimental results which we achieve with the proposed learning strategies. First, the 2D localization performance of the different learning strategies is evaluated with the detection quality criterion suggested by [7]. In addition, an object class estimation approach for the sequential and joint learning strategy is briefly explained and evaluated.

4.1. Multi-Class Datasets and Training Setup

The 2D localization performance of the proposed learning strategies is evaluated on three different testsets which consist of images from the 3D Object Category dataset [22] and the PASCAL VOC2006 dataset [7]. Specifically, we utilize the following multi-view testsets:

- **Bicycle-Car-Dataset**: This testset contains 192 images from the 3D Object Category dataset, showing two bicycle and two car instances from 48 different viewpoints.

- **Bicycle-Motorbike-Dataset**: This testset contains 96 images from the 3D Object Category dataset, showing two bicycle instances from 48 different viewpoints. In addition, we use the first 96 images from the VOC2006 motorbike testset which show only one motorbike (not labeled as 'truncated' or 'difficult').

[3] We set $L = 3$ since this value performed best in our experiments.

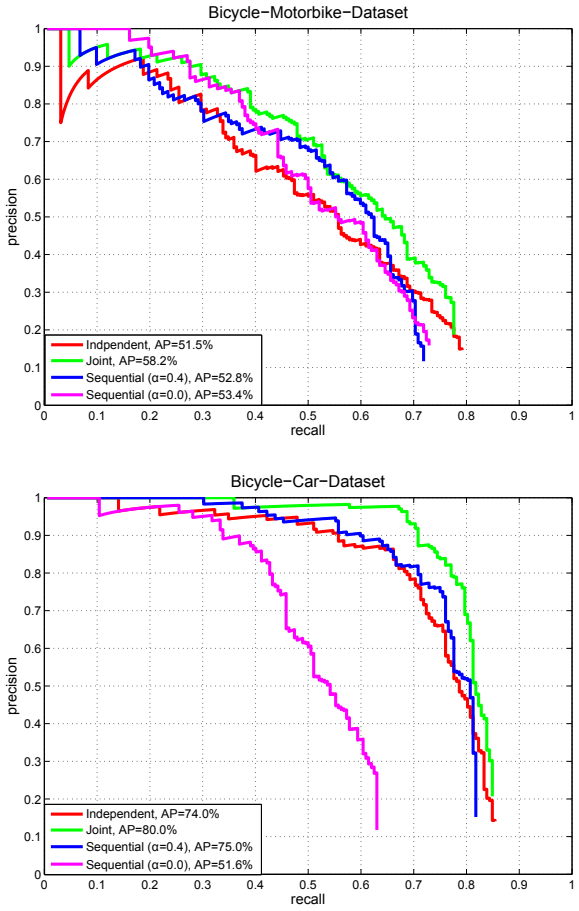


Figure 1. Precision/Recall curves for our multi-class and multi-view learning strategies on the Bicycle-Motorbike-Dataset (top) and the Bicycle-Car-Dataset (bottom).

- **Bicycle-Car-Motorbike-Dataset**: This testset contains the 192 testimages from the Bicycle-Car-Dataset and the 96 motorbike images from the Bicycle-Motorbike-Dataset.

Our proposed learning strategies rely on training steps of the viewsphere model where a database of synthetic 3D models serves as training source. To this purpose, we use 25 car models, 8 bicycle models and 13 motorbike models which are available from the distributors turbosquid.com and doschdesign.com. Azimuth is sampled from $0°$ to $360°$ in $5°$ steps and elevation is sampled from $0°$ to $20°$ in $5°$ steps to define a dense grid of viewpoints on the viewsphere. In order to make sure that the training images for the different learning strategies are identical, this viewpoint setup is used to generate once for each class the pure training images and the validation images. For a fair comparison of the three different learning strategies, two possibilities exist: either we keep the 2D detection performance constant and compare the computational complexity (which is measured by the number of object parts to detect) of the overall representation or we keep the computational complexity constant and compare the 2D detection performance. In our case, we choose to keep the computational complexity for the different learning strategies constant and compare the detection performance. For all experiments, the following settings are used: $M_I^c = \frac{M_J}{C} = \frac{M_S}{C} = 10$ parts for modeling a dense grid of spatial layout models and $N_I^c = \frac{N_J}{C} = \frac{N_S}{C} = 25$ parts for learning the global appearance where C is the number of classes.

Our proposed learning strategies rely on the viewsphere model of Section 2. In order to obtain a baseline performance of the viewsphere model, its 2D detection performance is compared with the current state-of-the-art detector of [11], using their pre-trained object class models provided as part of *voc-release3*. To this purpose, we use the 3D Object Category bicycle dataset and follow the test protocol of [18]. The Precision/Recall curves are shown in Figure 3 (right). With 74.4% the viewsphere model outperforms the approach of [11] with 71.2%, despite being trained on synthetically generated images.

4.2. Two Object Classes

We apply our different learning strategies to two visually very similar classes (i.e. Bicycle-Motorbike-Dataset) and two dissimilar classes (i.e. Bicycle-Car-Dataset). Figure 1 shows the corresponding Precision/Recall curves. We observe for both cases that the joint learning strategy (green curves) outperforms the independent learning strategy (red curves) and the sequential learning strategy (blue and magenta curves) due to a higher precision. In order to assess the influence of the transferable object parts (see Section 3.3) the sequential learning from the bicycle to the motorbike class and from the bicycle to the car class is performed for two different information thresholds α (cf. Equation 4). For $\alpha = 0.0$ (magenta curves) all parts from the previously trained bicycle class are considered as transferable object parts with the result that for both cases (bicycle to car and bicycle to motorbike) the set of remaining training images I_{remain} is an empty set and consequently no further object parts for the novel classes (i.e. motorbike or car) are generated. As a result, for similar object classes (i.e. bicycle to motorbike) the detection result for the sequential learning (53.4%) is still on par with the independent learning (51.5%) and worse than the joint learning (58.2%). For dissimilar object classes (i.e. bicycle to car) the detection result for the sequential learning (51.6%) is worse than both the independent learning (74.0%) and the joint learning (80.0%). With an increased information threshold of $\alpha = 0.4$ (blue curves) the situation is different. For dissimilar object classes (i.e. bicycle to car) none of the bicycle parts are considered as transferable object parts which results in a non-empty set for the remaining training

Figure 2. Examples for transferable object parts: bicycle to motorbike (left) and bicycle to car (right).

images I_{remain} and consequently additional car parts are generated. The increased detection performance (75.0%) is now on par with the independent learning (74.0%). For similar object classes (i.e. bicycle to motorbike) three of the bicycle parts are still considered as transferable object parts. This results in a non-empty set for the remaining training images I_{remain}, additionally generated motorbike parts and a detection result (52.8%) which is on par with the result of the independent learning (51.5%). This shows that in both cases (i.e. for similar and dissimilar object classes) an information threshold of $\alpha = 0.4$ for the sequential learning achieves the best trade-off between transfering knowledge from previously trained classes to novel classes and generating additional knowledge from novel classes, and consequently results in a detection performance which is on par with or better than the detection performance of the independent learning. Therefore, for subsequent tests the information threshold is set to $\alpha = 0.4$. Examples for transferable object parts on both datasets are shown in Figure 2.

4.3. Three Object Classes

The Precision/Recall curves on the Bicycle-Car-Motorbike-Dataset are shown in Figure 3 (left). In this case, the joint learning (green curve) clearly outperforms the independent learning (red curve) due to a higher precision. We observe that the order in which the classes are learnt during the sequential learning affects the detection performance. However, both detection results (68.4% and 74.1%) of the sequential learning (blue and magenta curve) significantly outperform the independent learning (56.5%) and perform better or on par with the joint learning (69.8%). In addition, with the sequential learning strategy it is possible to learn a novel object class without retraining the appearance of the previously trained object classes.

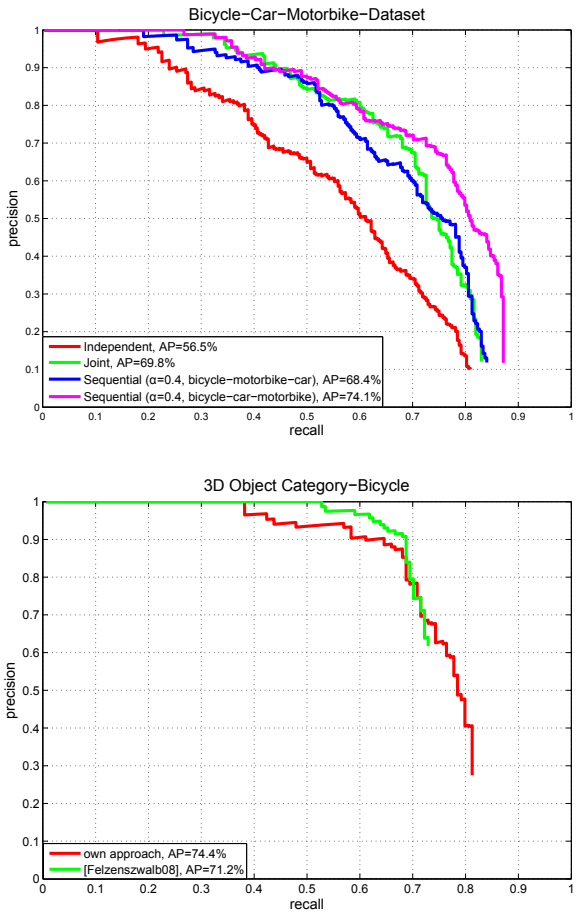


Figure 3. Precision/Recall curves for our multi-class and multi-view learning strategies on the Bicycle-Car-Motorbike-Dataset (top). On the 3D Object Category dataset bicycle a single-class detector (viewsphere model) is compared with a state-of-the-art detector (bottom).

4.4. Object Class and Pose Estimation

An additional advantage of our part representation for multiple object classes resides in the spatial co-occurrence of parts which can be used for both object class and 3D pose estimation from single images [14, 18, 23, 25]. The following experiment shows that this advantage is retained even when the parts are shared over several object classes. Based on the sequential (or joint) learning strategy it is possible to estimate the object class for a predicted bounding box. To this purpose, it is necessary to adapt the selection criteria in Section 2: for each object class we draw a new subset of N_S (or N_J) parts from the final pool P_S^C (or P_J) which contains a maximum amount of information about a specific object class. The pure training images of a specific class serve as positive set and the pure training images of the remaining object classes serve as negative set. With the selected subsets of class-specific object parts the

common global appearance for each object class is learnt, as described in Section 2. Finally, a predicted bounding box obtains the class label from the corresponding spatial pyramid classifier with the highest classification score. Figure 4 (left) shows the confusion matrix which we observe, when classifying all positive detections of the sequential learning strategy (learning order: bicycle-car-motorbike) on the Bicycle-Car-Motorbike-Dataset. The matrix shows that confusion is more pronounced between bicycles and motorbikes but we still obtain an average classification accuracy (AA) of 93.7%. Based on our part representation for multiple object classes it is also possible to estimate the 3D pose for a predicted bounding box by using the 3D pose estimation approach of [23]. Figure 4 (right) shows some successful results of the full detection process with 2D localization and 3D pose estimation on the Bicycle-Car-Motorbike-Dataset.

5. Conclusion

In this paper, we propose three novel learning strategies to recognize multiple object classes from arbitrary viewpoints. The learning strategies rely on the part-based approach of [23], where a database of synthetic 3D models serves as training source. We show that a sequential learning achieves the best result with respect to flexibility during the training process and recognition performance. Future work will focus on extending the proposed sequential learning strategy to train multiple object classes from arbitrary viewpoints on a larger scale.

References

[1] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *IEEE International Conference on Computer Vision*, 2011.

[2] E. Bart and S. Ullman. Cross-generalization: learning novel classes from a single example by feature replacement. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[3] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.

[4] C. A. Bouman. Cluster: An unsupervised algorithm for modeling gaussian mixtures. 1997.

[5] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. J. Wiley, 1991.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[7] M. Everingham, A. Zisserman, C. K. I. Williams, and L. V. Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf, 2006.

[8] L. Fei-Fei. Knowledge transfer in learning to recognize visual objects classes. *International Conference on Development and Learning*, 2006.

[9] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:594 – 611, 2006.

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61, 2005.

[11] P. F. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[12] S. Fidler, M. Boben, and A. Leonardis. Evaluating multi-class learning strategies in a hierarchical framework for object detection. In *Advances in Neural Information Processing Systems*, 2009.

[13] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[14] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *IEEE International Conference on Computer Vision*, 2011.

[15] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *Journal of Machine Learning Research*, 2007.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[17] K. Levi, M. Fink, and Y. Weiss. Learning from a small number of training examples by exploiting object categories. In *Workshop on Learning in Computer Vision*, 2004.

[18] J. Liebelt and C. Schmid. Multi-view object class detection with a 3D geometric model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[19] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3D feature maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[20] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

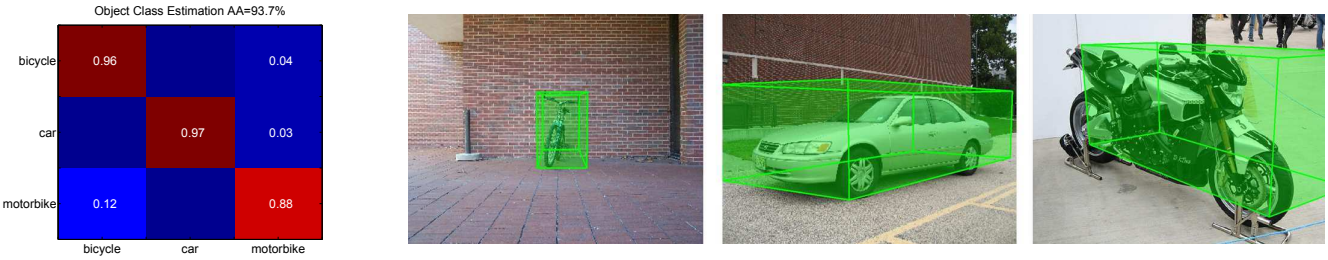


Figure 4. Confusion matrix (rows: groundtruth, columns: estimates) for the object class estimation (left). With our proposed multi-class learning strategies it is possible to predict an approximate 3D pose for a bounding box (right).

[21] N. Razavi, J. Gall, and L. V. Gool. Scalable multi-class object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[22] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *IEEE International Conference on Computer Vision*, 2007.

[23] J. Schels, J. Liebelt, and R. Lienhart. Learning an object class representation on a continuous viewsphere. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[24] M. Stark, M. Goesele, and B. Schiele. A shape-based object class model for knowledge transfer. In *IEEE International Conference on Computer Vision*, 2009.

[25] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *IEEE International Conference on Computer Vision*, 2009.

[26] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):854–869, 2007.

[27] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *IEEE International Conference on Computer Vision*, 2003.