Person Attribute Recognition with a Jointly-trained Holistic CNN Model

Patrick Sudowe
Visual Computing Institute
RWTH Aachen University
sudowe@vision.rwth-aachen.de

Hannah Spitzer
Institute of Neuroscience and Medicine (INM1)
Research Centre Jülich
h.spitzer@fz-juelich.de

Bastian Leibe
Visual Computing Institute
RWTH Aachen University
leibe@vision.rwth-aachen.de

Abstract

This paper addresses the problem of human visual attribute recognition, i.e.*, the prediction of a fixed set of semantic attributes given an image of a person. Previous work often considered the different attributes independently from each other, without taking advantage of possible dependencies between them. In contrast, we propose a method to jointly train a CNN model for all attributes that can take advantage of those dependencies, considering as input only the image without additional external pose, part or context information. We report detailed experiments examining the contribution of individual aspects, which yields beneficial insights for other researchers. Our holistic CNN achieves superior performance on two publicly available attribute datasets improving on methods that additionally rely on pose-alignment or context. To support further evaluations, we present a novel dataset, based on realistic outdoor video sequences, that contains more than 27,000 pedestrians annotated with 10 attributes. Finally, we explore design options to embrace the N/A labels inherently present in this task.*

1. Introduction

We address the problem of person attribute recognition, where the input is an image of a person and the task is to make predictions for a set of attributes. In contrast to other recognition problems, attributes are based on a semantic proposition, with binary (*e.g.*, *is male? wears a tshirt? carries a bag in the left hand?*) or multinomial outcome (*e.g.*, *orientation - left, right, front, or back*). Such attribute predictions are interesting for a range of applications like image retrieval, querying databases by semantic propositions, tracking-by-detection, re-identification applications, and robotic applications that require semantic information of persons for interaction.

Attribute recognition approaches have to address three challenges: (i) Most attributes require fine-grained information, *i.e.*, they need to be decided based on small subregions of the input image. (ii) For most input images, some attributes are not decidable. If in the given image, like in Fig. 1, it is not possible to decide an attribute, because of occlusion, image boundaries, or any other reason, then the correct answer is to say "I cannot decide". We call this outcome the *N/A* label. (iii) In practice, many attributes of interest are dependent or correlated. For example, if a person is walking to the left, we often cannot decide whether she carries a bag in her right hand (*c.f*. Fig. 1).

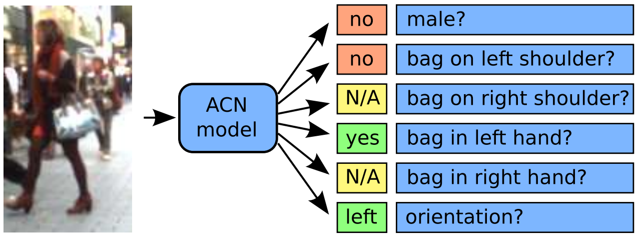

Figure 1. We propose a model that is jointly-trained, shares weights among attributes, and does not rely on additional external information like context or pose. Given an input image of a person, our ACN model predicts multiple attributes at once. Some attributes are labeled as not decidable (N/A) in the groundtruth.

This paper proposes a method to jointly train a monolithic CNN model for all attributes, allowing to share weights and thus effectively transfer or re-use knowledge among attributes. One problem we need to address is that most examples have at least one *N/A* label. Commonly, models are trained separately, so that this issue can be alleviated by filtering the training set [18, 19, 2, 22]. In contrast, we jointly train a model employing a loss function that handles the partially undefined targets vectors. Our model is much less complex than published methods relying on parts, pose, or context information [22, 23, 19]. Still, we show that our model outperforms all published methods in two public benchmarks.

Previous attribute recognition approaches often take a retrieval viewpoint, where only two prediction outcomes are possible. When presented with an *N/A* example, these models necessarily make a mistake. For some applications, such as robotics or tracking, it is preferable to only make decisions with high precision, while it is acceptable to defer a

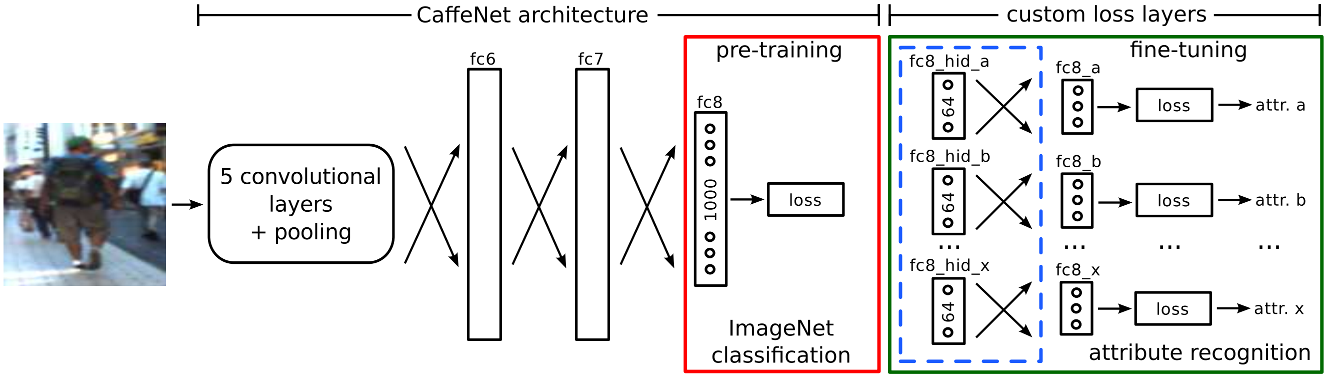

Figure 2. ACN Architecture. The underlying architecture is a CaffeNet that was pre-trained on ImageNet (red). For the attribute recognition task we replace the last fully-connected layer with one loss layer per attribute (green) using the loss described in Sec. 3. We show that introducing an additional hidden layer further improves the results (blue).

decision. This motivates us to move on to explicitly modeling the *N/A* label, resulting in an N+1 model (Sec. 6).

In detail, this paper makes the following contributions: (i) We propose a method to jointly train a recognition model for all attributes. The resulting approach outperforms all published methods on two public benchmarks, to the best of our knowledge. (ii) We investigate the effect of modeling the *N/A* class and evaluate different options for handling it during training. (iii) We introduce a novel benchmark *PARSE-27k*, which is a larger well-aligned attribute dataset, and use it to perform extensive evaluation to examine the performance relevant factors of our model.

The remainder of this paper is structured as follows: The next section discusses related work. Sec. 3 details our training procedure and highlights some key concepts. Sec. 4 describes the datasets used in this work. Sec. 5 will stick to the common evaluation protocol, based on *mAP*. First, we will focus on our new dataset *PARSE-27k* and examine individual aspects of our model. Secondly, we will show the efficacy of our method by evaluating on two recent benchmarks. Finally, Sec. 6 takes the N+1-viewpoint and investigates how one can leverage *N/A* examples.

2. Related Work

Attribute Recognition. There are several interpretations of attributes, we follow the interpretation as a categorical (*i.e.*, binary or multinomial) predicate [18, 19, 2, 22, 23]. Other interpretations include relative attributes defined over a continuous scale [15] or discriminative attributes considered by Farhadi *et al.* [6]. The attribute task has been described in the context of many applications, such as improving object recognition [6] and image retrieval by attribute queries [20]. This work aims at semantic person attributes.

One main characteristic of the attribute recognition problem is its fine localization, *i.e.* only a small sub-region of the input is discriminative for the decision in many cases. This has motivated work by Sharma *et al.* [18] who learn spatially-localized features similar to spatial pyramid models. With a similar objective, Bourdev *et al.* [2] aim at pose-alignment with their poselets framework. Following the same motivation, Sharma *et al.* [19] propose to tackle the attribute task with a collection of localized discriminative templates. We compare to those methods in Sec. 5.

CNNs and Transfer Learning. In recent years, Convolutional Neural Networks (CNN) have gained huge popularity. Particularly, the success of Krizvhesky *et al.* [12] in the ILSVRC-2012 classification task, has fostered a line of work applying similar models to tasks like object recognition or detection [4, 16, 3, 14]. The work of Donahue *et al.* [4] and Razavian *et al.* [16] shows that CNNs learn excellent feature representations, which can be directly leveraged for other tasks. Due to the large number of parameters, CNNs are prone to overfit on smaller datasets. This can be alleviated to some extent by pre-training the weights on a large-scale task, followed by training on the target task. This procedure, known as *fine-tuning*, leverages auxiliary tasks and transfers feature representations [3, 8, 14]. A recent study by Yosinksi *et al.* [21] shows that by transferring models between tasks the generalization on the target task can be significantly improved. The recent work of Branson *et al.* [3] tackles fine-grained object recognition using a transfer-learning approach similar to ours.

Most similar to our approach are Zhang *et al.* [22] and Razavian *et al.* [16], both are based on CNNs. Razavian *et al.* use OverFeat as a feature extractor and train SVM models, achieving competitive results on the Berkeley Attributes of People dataset [17]. In contrast to their work, we only perform end-to-end training to adapt the CNN's weights. Zhang *et al.* combine deep features with pose-alignment in a multi-level pipeline, training multiple sub-models combined by final SVM models (one per attribute) [22].

3. Our Method: ACN

Most current approaches train separate models for the different attributes [2, 22, 16, 19, 18]. In contrast, we propose to jointly train a CNN for all attributes in one model — the Attributes Convolutional Net (ACN) — based on the proposition that it is desirable to share parameters among attributes (Fig. 2). We show that if performed with careful attention to detail, this significantly improves performance.

A network can comprise multiple loss layers to target distinct concepts (like attributes). For every training example forwarded through the network, one then computes the losses and backpropagates the sum of their gradients.

One particular challenge we address is that most training examples contain at least one *N/A* label. Many approaches exclude the *N/A* labels prior to training, but this is only feasible if one trains separate models for each attribute. Another option would be to introduce an *N/A* class. But the *N/A* label only expresses that the attribute's proposition is not decidable, while it certainly has an actual value in reality. We take the view that the *N/A* label is *not an actual state* of the attribute (hence not its own class), but it only expresses the "undecidability" due to the limited observer. More specifically, a person in reality either carries a bag in her hand or not. If the observer cannot make a decision, it is natural to say that there is simply no information available. Hence, a natural choice would be to set the gradient to 0, muting its influence in training. We will see that this is consistent with the KL-loss. Our models have one loss for each attribute, which are all accumulated during backpropagation. Even if some labels of a particular example are *N/A*, the other labels will still help further the training process.

KL Loss. We minimize the Kullback-Leibler divergence of two discrete distributions, our predictions Q and a binary attribute's state in reality P:

$$\mathrm{KL}(P||Q) = \sum_i^N P(x_i)\; log \frac{P(x_i)}{Q(x_i)}$$

where our target P for example x_i is specified as follows:

$$P(x_i = \text{yes}) = \ell; P(x_i = \text{no}) = 1 - \ell,$$

with $\ell \in \{0, 1\}$ corresponding to the groundtruth annotation. The empirical risk minimization of the KL-divergence is equivalent to maximum likelihood estimation [1, p.56]. During backpropagation this definition will naturally yield a gradient of 0 for any *N/A* example. Yet, for valid labels it yields gradients equivalent to log-likelihood. This generalizes to the multinomial case using softmax.

Important Aspects. We would like to point out two important aspects at this point: (1) While it is common to fit a CNN for multiple classes, our attribute recognition problem is not an N-way classification like in the ImageNet challenge. Our target space is rather the cross-product of each attribute's individual target space, additionally complicated by the *N/A* labels. So it can be seen as a structured prediction problem not a multi-class problem. (2) Some other approaches try to capture attribute correlations a-posteriori (like [2] by fitting a kernel SVM on the initial models). In contrast to this, our model shares a large part of its parameters across all target attributes, and only uses a single training step. This enables us to capture correlations implicitly.

Implementation. Our work is based on the CAFFE-framework [10]. We start with the CAFFENET reference implementation of the SUPERVISION-CNN due to Krizhevsky *et al.* [12], which is trained on the 1.2M images from the ImageNet ILSVRC-2012 1000-way classification task[1].

As mentioned above, we employ a two-stage training. Fig. 2 shows our network's architecture, including the losses of both stages. We replace the original loss layers (red box) with additional fully-connected layers and our loss (green box). For each of the attributes in the target task, we add a group of additional classification and loss layers. Observe that the layout of the first layers remains completely unchanged. Then we use stochastic gradient descent to optimize both the newly introduced weights and the pre-trained weights. In this way, the "knowledge" obtained in training on the auxiliary task is transferred, *i.e.*, we use the pre-trained weights as an initialization of the network [3, 8, 21]. Our training does update the weights in the first layers, but at a smaller learning rate $(1/10)$ than our newly added weights. This is in contrast to the work of Oquab *et al.* [14] who report to keep the first layers fixed during training for the target task. We find that adapting the weights yields consistently better results on our task. Here, a significantly reduced learning rate appears to be key.

Details of the Model. We now cover the details – aiming to enable other researchers to easily build upon our results. We preprocess all examples by warping the input image to 256×256 pixels. During training, we make use of several data augmentation techniques: We resize the original input's bounding box in multiple scales to add more training examples with varying degrees of background. We include horizontally mirrored duplicates during training. For attributes that are not invariant to mirroring, we adapt the labels accordingly. As another data augmentation technique we employ *PCA jittering* as proposed by Krizhevsky *et al.* [12]. Here, we compute the PCA of the covariance matrix of all RGB values in the training set, yielding three eigenvalues e_i and corresponding eigenvectors v_i. Then in each training iteration we sample a value $\alpha \sim \mathcal{N}(0, 0.1)$ and jitter every pixel p in the example image with $\hat{p} = p + \sum_i \alpha \cdot e_i \cdot v_i$. Sec. 5 investigates the effect of these data augmentation techniques.

For training, we randomly crop 227×227 sub-windows from the input image. At test time, we deterministically take sub-crops and average over the individual predictions (c.f. [12]). We find that this yields a small yet consistent

[1] http://caffe.berkeleyvision.org/model_zoo.html

improvement over a single crop at the center of the window.

For training, we use a step-wise reduction of the learning-rate, which implies a factor 10 decrease every $20{,}000$ updates (with a batch-size of 64 examples). We train until we see a plateau on the validation loss that is unchanged by a further decrease in learning-rate.

We found it important to consider the regularization through *weight decay*. Since we are dealing with relatively small datasets, this regularization helps even though other techniques like *drop out* were already used additionally, as in the original SUPERVISION architecture. In practice, we used a *weight decay* of 0.005, unless stated otherwise.

3.1. Forced Choice vs. Reject Option

Most work on the attribute recognition task evaluates the models in such a way that the *N/A* label is discarded from the test set, *i.e.*, the model's answers are excluded from the evaluation. The reason for this is that the average precision (AP) is commonly used for evaluation, which is a measure originating from retrieval tasks. Depending on the application scenario this may not be appropriate. If one is interested in retrieving all images of a collection where persons are wearing sunglasses, then AP is an appropriate measure. In contrast, applications in robotics, tracking-by-detection, and intelligent vehicles need to rely on the predictions and would be better off not acting on uncertain outcomes. The models trained in Sec. 5 follow a *forced choice principle*: Despite the natural presence of *N/A* targets in the ground-truth, the models, trained for N classes, cannot predict *N/A* labels (class N+1).

In the following, we describe three approaches to *N/A*-label-prediction. Naturally, the mAP-based evaluation is not useful for this approach, because it ignores the predictions on *N/A* ground-truth examples, and the scores would not reflect the performance on the target task. So Sec. 6 reports results based on the *balanced error rate* (BER) for *PARSE-27k*.

Reject Region. A first straight-forward way to predict *N/A* labels is to define a region by a threshold δ. Considering a two class model, we observe activations in the range $[0, 1]$. Motivated by the uncertainty encoded by a Bernoulli variable, we predict *N/A* for $0.5 - \delta \leq a \leq 0.5 + \delta$. We fit δ on the validation set optimizing with respect to the *BER* on the N+1 class problem (*c.f.* Sec. 6). Any two class model that outputs a continuous score can be extended into an N+1 predictor using this strategy.

Softmax. As a second baseline approach, we train with a standard softmax loss function with N+1 outputs, *i.e.*, we simply add one output for the *N/A* targets. We will show that this is surprisingly effective, even though one could argue that the *N/A* labels do not form a class of their own.

Hierarchical Softmax. The softmax model assumes that the *N/A* targets form their own class. From a philosophical point of view this is doubtful. There are many reasons why a visual attribute might not be decidable, like occlusion of the relevant regions. However, the attribute will in reality still have one of the distinct states, either true or false. This motivates the *hierarchical softmax* approach, instead of the direct *softmax* model. One can think of the N+1 class prediction as a two step procedure. Consider two random variables A, B. Let $p(A)$ denote whether the attribute is decidable, i.e. *N/A* *vs.* not *N/A*. Further, let $p(B)$ denote the probability of the attribute being true. Then one can naturally assume the following factorization of the joint: $p(A, B) = p(B|A) \cdot p(A)$. From the network perspective this allows to use different parameters for both losses. We create a network with two loss layers per attribute, one a logistic loss for A and one a softmax loss for B. The final predictions are obtained by multiplying the probability estimates of both $p(A)$ and $p(B|A)$, *i.e.*, the two network outputs for a given attribute.

4. Datasets

HATDB. The Database of Human Attributes originally published by Sharma *et al.* [18] contains labels for 27 binary attributes (covering age, gender, appearance, and pose). The images have been taken from Flickr and show a considerable variance in resolution. Persons shown in the dataset appear in many different poses, like sitting or standing, and are depicted in different crops (*i.e.*, upper-body, head-only, full-body). The dataset proposes a train-val-test split with $3{,}500$, $3{,}500$, and $2{,}344$ examples, respectively.

Berkeley - Attributes of People. This dataset was originally compiled by Bourdev *et al.* [2]. It comprises $4{,}013$ training and $4{,}022$ test examples, which are labeled with 9 binary attributes (MALE, LONG_HAIR, GLASSES, HAT, T-SHIRT, LONG_SLEEVES, SHORTS, JEANS, LONG_PANTS). Several authors have evaluated their methods on this dataset, including recent CNN-based approaches [16, 22]. Since there are many results to compare, this dataset lends itself as a testbed. The dataset contains examples from various sources, at various resolutions. The examples feature a large variance in pose and resolution. Additionally, some pictures show only parts, like only the upper-body, whereas others show the full body of a person. This, in combination with the rather small training set, renders it challenging to train good models.

4.1. New Dataset – PARSE-27k

We created a dataset named *Pedestrian Attribute Recognition on Sequences* containing 27k annotated examples (*PARSE-27k*). The previously described datasets for human attribute recognition are both relatively small and contain very general image collections, including upper-body

binary attributes	train			val			test		
	pos	neg	N/A	pos	neg	N/A	pos	neg	N/A
male	5822	6465	1454	2876	3314	428	3109	2913	633
standing	1078	11712	951	638	5758	222	5660	563	432
hasBagOnShoulderLeft (bsl)	1873	6436	5432	1094	3614	1910	897	3851	1907
hasBagOnShoulderRight (bsr)	2064	6240	5437	990	3722	1906	1044	3739	1872
hasBagInHandLeft (bhl)	1684	6409	5648	770	3349	2499	737	3317	2601
hasBagInHandRight (bhr)	1676	6492	5573	983	3272	2363	811	3307	2537
hasBackpack (backp)	381	9989	3371	159	5166	1293	322	5279	1054
isPushing (pushing)	332	12789	620	194	6225	199	333	6043	279

ori4	left	front	right	back	N/A
train	984	5359	964	6343	91
val	450	2828	456	2876	8
test	349	2630	445	3205	26

ori8	left	f-l	front	f-r	right	b-r	back	b-l	N/A
train	526	1553	3278	1015	531	1032	4709	1082	15
val	226	823	1600	682	232	481	1994	561	19
test	203	758	1562	490	261	560	2321	465	35

Table 1. Frequencies of labels for the 10 attributes in the *PARSE-27k* dataset.

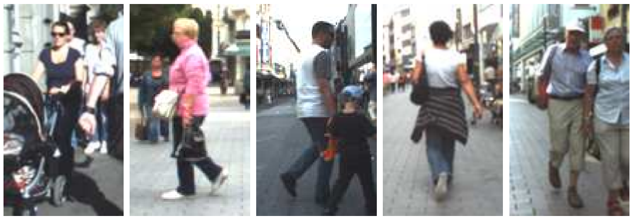

Figure 3. Examples from our new *PARSE-27k* dataset — We aim at a large well-aligned yet diverse dataset of pedestrians in realistic scenarios.

or face shots. In addition the datasets follow a random split routine, which leads to a few closely related training and test examples. Recently, the *CPR* dataset [9] with a similar size but only four attributes has been published.

PARSE-27k is based on 8 video sequences of varying length taken by a moving camera in a city environment. Every 15th frame of the sequences was processed by the DPM pedestrian detector [7]. The obtained bounding boxes were manually annotated with 10 attribute labels. The choice of attributes is motivated by a robotics/automotive application scenario and includes two orientation labels with 4 and 8 discretizations, and several *binary* attributes such as *is male?* and *has bag on left shoulder?* (Tab. 1). All attributes additionally may be labeled with an *N/A* value, so the *binary* attributes have 2+1 possible labels. Fig. 3 shows some example images of our new dataset. Note, that the other two datasets described previously also include *N/A* labels (0 in the ground-truth). This is not specific to our dataset, but rather induced by the definition of the attribute recognition task. If the underlying proposal is not decidable, one cannot sensibly give a valid ground-truth label.

PARSE-27k has a careful train (50%), val (25%) and test (25%) split. This means that we have split only along sequence boundaries. Additionally, sequences taken on the same day are either in train-val or test. This avoids highly similar examples across splits. Further, *PARSE-27k* has less variance with respect to pose and crop, since it only contains crops of pedestrian bounding boxes obtained by a pedestrian detector. By both increasing the dataset size and reducing this variance, we hope to improve model quality. We will make *PARSE-27k* available to the research community.

5. Experimental Evaluation

In this section, we adopt the retrieval viewpoint that is prevalent in the literature. This allows us to evaluate our models using the common *mean average precision* (*mAP*), which yields a fair comparison to the state-of-the-art. All models in this section are trained to yield an N-class answer, using the KL-loss detailed in Sec. 3. We call this a *forced choice* model, because the models can only choose one of the N labels. This is the commonly used interpretation in the literature for evaluating attribute prediction models [18, 2, 23, 16]. In contrast to this are *reject option* models, which are capable of rejecting the decision (N+1 : *N/A*) if none of the N labels appears appropriate.

Throughout this paper, we use the definition of *AP* by Everingham *et al.* in the context of the VOC object detection challenge [5]. In short, this definition averages over 11 points in regular intervals of the precision-recall curves. However, evaluation routines used by Zhang *et al.* [22] differ slightly from this, leading to slightly different scores. To allow a meaningful comparison, we adopt their publicly available evaluation routine[2] only for experiments on the Berkeley Attributes of People dataset.

5.1. Experiments on PARSE-27k Dataset

We begin our experimental evaluation by reporting results on our *PARSE-27k* dataset. We show the positive effects of jointly training all attributes in one CNN model as compared to separate models. Further, we investigate the effects of several aspects of our proposed pipeline, in order to assess their individual contributions, such as the effects of two data augmentation techniques. Next, we explore design variants such as an additional hidden layer for each

[2] https://github.com/facebook/pose-aligned-deep-networks

Attribute	mAP	male	standing	bsl	bsr	bhl	bhr	backp	pushing	ori4	ori8
ACN - jitter - aug	53.2	88.6	44.0	48.2	57.6	44.9	54.8	35.6	47.2	83.8	68.2
ACN - jitter	60.7	90.1	48.0	67.3	72.0	**65.3**	65.7	36.2	44.1	86.8	71.9
separate + aug + jitter	53.9	89.7	48.1	48.1	66.1	52.9	57.5	28.8	39.6	87.2	**73.8**
binary + aug + jitter	58.7	89.5	40.9	63.7	69.9	59.7	63.5	38.1	43.9	-	-
ACN	62.4	90.4	51.3	67.6	72.7	65.2	65.7	**38.8**	47.4	86.9	72.9
ACN + SVM	60.9	**90.5**	50.6	64.0	69.8	63.5	66.3	37.4	45.0	-	-
ACN + Hidden	**63.6**	90.1	**58.3**	**69.6**	**73.2**	64.6	**69.2**	35.8	**48.3**	**87.9**	**73.8**

Table 2. Detailed performance on the *PARSE-27k* dataset in terms of AP. For the orientations, accuracy scores are given. *ACN* is trained on all attributes, *binary* on the binary attributes, and *separate* summarizes models trained on each attribute separately. The results show that *ACN* outperforms both the *separate* and the *binary* baseline. Results further improve by adding an additional hidden layer.

attribute or training SVMs on the activations of the network which have been proposed in the literature.

We train our models with a learning rate of 0.0001. The model without data augmentation requires a weight decay of 0.05 to regulate the overfitting. When including data augmentation, the higher variance in the dataset reduces the need for regularization, and we empirically find a weight decay of 0.0005 to work well. Unless otherwise stated, we train our models with all augmentation techniques presented in Sec. 3, namely *random cropping*, *mirroring*, *scaling* with three different scales, and *PCA jittering*. For this experiment, we train our models on 10 attributes. Additionally, we give accuracy scores for the orientation attributes, as the AP is not defined for multinomial attributes. Note, that we cannot give meaningful performance figures for other methods. This would require re-training of the models. Even where code is available, this is bound to lead to inferior performance and misleading comparisons. We do provide meaningful comparisons, using the originally reported scores, for benchmark datasets in Sec. 5.2 and 5.3.

Effect of Joint Training. Tab. 2 shows our results on the *PARSE-27k* dataset. Our jointly trained *ACN* yields an mAP of 62.4. If we consider models of the same architecture trained on each attribute separately (*separate*), the mAP is 53.9. Combining only the binary attributes into an *ACN* model improves performance to 58.7 mAP. This indicates that sharing weights among the individual attributes is beneficial and that both the binary and the orientation attributes contribute to the performance improvement. The binary attributes' appearances are highly dependent on the orientation. The *ACN* model leverages this indirectly by learning to predict the orientation. Note, this is an additional output of the network, not an input to the binary models. Hence the two orientation attributes, which are not even considered in the mAP, contribute almost half of the performance increase gained by joint training.

These results clearly show that it is advantageous to train a combined model. While these results are in line with the literature, it was important to point this out, because previously many models have been trained separately [2, 18, 22]. The attributes can indirectly influence each other by adapting the weights in the lower layers. Thus, the inclusion of the orientation as target attribute especially helps the performance of those attributes which are sensitive to the orientation (*e.g. has bag in hand*).

Effect of Data Augmentation. To investigate the impact of the data augmentation techniques, we trained two *ACN*s, one without data augmentation (*ACN* - jitter - aug) and one with data augmentation but without PCA jittering (*ACN* - jitter). The data augmentation techniques *random cropping*, *mirroring*, and *scaling* increase the mAP by 7 points. Additionally, the *PCA jittering* yields another 2 mAP points. Data augmentation and especially the PCA jittering result in very well regularized models that are not prone to overfit. When training without data augmentation, the training parameters need to be adapted due to heavy overfitting, which cannot be alleviated by adapting the weight decay.

Hidden Layers for each Attribute. It is possible to include an additional fully-connected layer for each attribute as indicated in Fig. 2. This corresponds to learning a multi-layer-perceptron with one hidden layer for each attribute separately, allowing them to do complex adaptions based upon the shared weights. Empirically, 64 hidden nodes for each attribute are optimal, as the performance of models with less and more hidden nodes decreases. The introduction of these hidden layers further increases the performance of our model by 1.2 mAP points, yielding the best performing model on *PARSE-27k* with an mAP of 63.6. In the following we call this architecture *ACNH*.

SVM vs. FCL. Several authors have proposed to train SVM models on top of the activations of a CNN [16, 22, 8]. Razavian *et al.* [16] followed this approach as they were using the Overfeat architecture as a fixed black box and separately trained SVM models using its activations as features. In contrast to this, we train our models end-to-end. So the obvious question is: Does training an SVM on top of deep features yield additional benefit? In order to investigate this, we trained a linear SVM on the activations of the penultimate layer *after fine-tuning*, and optimized its regularization parameter on the validation split. This procedure is similar to Girshick *et al.* [8]. Our results show that the fine-

	mAP
DSR [18]	53.8
SPM [13, 19]	55.5
EPM [19]	58.7
EPM + context[19]	59.7
ACNH	**66.1**
ACNH 5-ensemble	**66.2**

Table 3. Results on HATDB. We report results of SPM and EPM as published in [19]. Our *ACNH* outperforms all published results. For a detailed break-up of the results see supplementary material.

tuned fully connected layer (FCL) predictions outperform the SVM models by 1.5 mAP points (*ACN vs. ACN* + SVM in Tab. 2). This indicates that there is no advantage to training an SVM on top of CNN activation features. We believe it is worthwhile to point this out, because the SVM approach is popular in the literature.

5.2. Experiments on HATDB

We next turn to the publicly available HATDB (*c.f.* Sec. 4). We follow the training procedure detailed in Sec. 3. Some of the attributes in this dataset would form a multinomial attribute if combined. However, they are separately annotated and have been treated individually by related work [18, 19]. Hence, we follow the same view and train models to predict 27 separate binary attributes. The dataset proposes a train-val-test split, which we follow. Training continues until there is no further improvement in validation loss. In our experiments, we set the learning rate to 0.001 and the weight decay to 0.005, which gave best results.

We use the setup as explained in Sec. 3. We do not use the PCA jittering based data augmentation, as a quick experiment showed worse performance. This is likely related to dataset size, which is so small that the additional noise overlays the signal. We still use the other data augmentation techniques. Due to the stochastic nature of the CNN training it is a common final tweak to build an ensemble of several independently trained models. We report both *ACNH*, a single model, and *ACNH 5-ensemble*, an ensemble of 5 models trained with identical learning parameters.

Results & Discussion. Sharma *et al.* [18] propose their *Discriminative Spatial Representation (DSR)* approach, which learns fine-localized features from data. The *Expanded Parts Model (EPM)* [19] learns a set of discriminative templates and corresponding locations particularly aiming at classification of fine-localized attributes. Further, their *EPM+context* model even includes additional context information, which reportedly yields an additional improvement. Our *ACNH* model does not rely on modeling parts or context explicitly. In contrast, we use a single holistic model trained end-to-end (in the sense that the whole input window is processed at once, instead of individual submodels for separate parts). Tab. 3 shows that our approach outperforms, to the best of our knowledge, all published results on this dataset, improving to 66.1 mAP compared to the previous best at 59.7 mAP. This indicates that our CNN model is able to capture the finely-localized information. Overall, these are promising results that show the efficacy of our approach.

5.3. Experiments on Berkeley Attributes of People

One might argue that while the baselines we compared to in the previous section also learn representations from data, they are not based on CNNs. Hence, we next turn to the Berkeley Attributes of People dataset [2], because several CNN-based methods have been published on this dataset. It is particularly interesting to investigate how our model performs relative to elaborate methods such as PANDA [22]. As we will show, our approach is able to outperform all published results on this dataset, although it does not rely on any additional pose or context information. We use the same training setup as for the training on HATDB.

Results & Discussion. Tab. 4 compares to all published results we are aware of. Our *ACNH* model outperforms all previous results, and *ACNH 5-ensemble* yields an additional improvement. This final improvement comes at the extra cost of training and evaluating five models. Note, that this still involves less computational effort than the runner-up (PANDA), with its CNN-per-poselet computation.

Note that our method performs very well even for attributes like LONG_HAIR, where one would expect localized or aligned models to be in advantage. PANDA apparently benefits from its alignment for the GLASSES attribute, where it outperforms our method as well as Razavian *et al.* [16], the latter also being a single global model. It is also remarkable that for T-SHIRT both our model and Razavian *et al.* outperform PANDA by a large margin. Overall our model performs best with 80.02 mAP.

Our training pipeline learns all attributes at once, without intermediate steps. This is less complex than the PANDA approach [22], which exploits poselet activations. Comparing to the other CNN-based approaches, namely Razavian *et al.* and PANDA, our approach achieves better results despite its holistic model. An interesting question for future work is whether a pose-aligned model could again improve results when combined with our *ACN* model.

6. Evaluation – Reject Option

The previous section focused on models from the retrieval viewpoint and thus effectively excluded the *N/A* labels. However, as motivated in Sec. 3.1, for many applications the retrieval viewpoint does not make sense. A model which does well in retrieving all males from a person database, does not necessarily perform well in deciding on one particular person that a robot encounters. The com-

Attribute	mAP	male	long hair	glasses	hat	tshirt	longsleeves	shorts	jeans	long pants
Poselets [2]	65.18	82.4	72.5	55.6	60.1	51.2	74.2	45.5	54.7	90.3
DPD [23]	69.88	83.7	70.0	38.1	73.4	49.8	78.1	64.1	78.1	93.5
Joo *et al.* [11]	70.7	88.0	80.1	56.0	**75.4**	53.5	75.2	47.6	69.3	91.1
Razavian *et al.* [16]	73.0	84.8	71.0	42.5	66.9	57.7	84.0	79.1	75.7	95.3
PANDA [22]	78.98	**91.7**	**82.7**	**70.0**	74.2	49.8	86.0	79.1	81.0	96.4
ACNH	**79.71**	87.64	80.72	49.34	74.54	62.61	87.90	86.69	**90.02**	97.95
ACNH 5-ensemble	**80.02**	87.83	81.49	48.75	75.32	**64.07**	**88.06**	**87.08**	89.51	**98.05**

Table 4. Berkeley Attributes of People Dataset - Our *ACNH* models outperform all published methods in terms of mAP , while our approach is trained end-to-end and does not rely on any additional external information (such as pose).

Attribute	mBER	male	standing	bsl	bsr	bhl	bhr	backp	pushing	ori4	ori8
Reject Region	51.7	42.3	52.4	48.3	51.0	53.3	52.4	56.4	57.6	34.5	38.5
Softmax N+1	45.5	41.8	54.4	37.0	39.9	40.1	40.6	49.9	60.4	35.3	38.1
Softmax N+1 + Hidden	**43.9**	39.9	**51.4**	**36.5**	**38.1**	39.8	40.0	48.7	56.5	34.6	37.5
Hier. Softmax + Hidden	**43.9**	**39.6**	51.5	37.2	38.8	39.5	**39.9**	**48.0**	**56.4**	**34.1**	38.1

Table 5. Evaluation of the models trained with a reject option on the *PARSE-27k* dataset in terms of BER (lower values are better). The mean BER is taken over the binary attributes. Softmax N+1 + Hidden is the best model with the smallest BER.

mon evaluation procedure, based on *average precision*, disregards test examples labeled *N/A*, *i.e*., the models' performances on these are not reflected at all. To avoid wrong predictions, a *reject option* is helpful. If one allows the model such an outcome as the N+1 option, then it is natural to include the *N/A* examples in the test set.

The models considered in this section are designed to incorporate the *reject option*. Thus, evaluation with the AP score is not appropriate. Due to the imbalanced label frequencies, the accuracy also is not suitable as an evaluation score. Instead, we propose to use the *balanced error rate* (BER), defined as the mean of the per-class errors. Let K be the number of classes, and $\mathbf{C}$ be the confusion matrix such that the row $\mathbf{C}_{j*}$ holds the predictions for groundtruth class j, then

$$\text{BER} = \tfrac{1}{K}\textstyle\sum_i^K \left(1 - \frac{\mathbf{C}_{ii}}{\sum_j \mathbf{C}_{ij}}\right).$$

We evaluate the mean BER (mBER) only on the binary attributes, as the ranges of the orientation BERs differ and would distort the mean.

Results & Discussion. Tab. 5 shows the performance in terms of BER for the three methods proposed in Sec. 3.1 (lower values are better). One could argue that a well-performing approach in terms of AP could easily be transformed into an N+1 classifier by introducing a reject region thresholding the Bernoulli probabilities. We fit such a reject region on the best-performing model from Sec. 5. However, it can be seen that this *Reject Region* approach is clearly inferior to the softmax approach, which is a classifier on the N+1 classification problem. Thus, it is not easily possible to transform the binary models to N+1 models. One reason might be the model's over-confidence, which does not properly reflect the uncertainty appropriately. This motivates the use of models, a priori designed to predict *N/A* labels. In this work, we discuss two rather simple approaches: *Softmax N+1* and *Hierarchical Softmax*. Similar to the results in Sec. 5, the additional hidden layer improves performance also for the *Softmax N+1* and *Hierarchical Softmax* approaches. While the *Hierarchical Softmax* allows a fine-grained control of the *N/A*-vs-all component, it does not show a quantitative improvement over *Softmax N+1* in our experiments. Hence, we suggest to use the latter for predictions with *reject option*. The similarity in BER score for the two models is related to the fact that BER weighs errors equally. It is an unbiased measure that can serve as a benchmark for more elaborate methods. For future work, it will be interesting to reflect the uncertainty more appropriately.

7. Conclusion

We have proposed a method to jointly train a CNN model for multiple attributes that naturally handles *N/A* labels in the ground-truth. Our model (Sec. 5) achieves better results than previous best methods on two public benchmarks. At the same time, it is less complex, *i.e*., does not rely on multi-level pipelines or external information. Additionally, we reported results on the new *PARSE-27k* dataset, enabling other researchers to build on our dataset and compare to our method. Secondly, we have pointed out that the common *mAP* evaluation for attributes is only suitable for some target applications. Sec. 6 proposed an alternative evaluation scheme, which also reflects prediction quality on the *N/A* labels. We hope that this, in combination with our new dataset, will serve as a starting point for future research.

Acknowledgments: This work was funded, in parts, by ERC Starting Grant project CV-SUPER (ERC-2012-StG-307432).

References

[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., 6th edition, 2007.

[2] L. Bourdev, S. Maji, and J. Malik. Describing People: A Poselet-Based Approach to Attribute Classification. In *ICCV*, 2011.

[3] S. Branson, G. Van Horn, S. Belongie, and P. Perona. Bird Species Categorization Using Pose Normalized Deep Convolutional Nets. In *BMVC*, 2014.

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. In *ICML*, 2013.

[5] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 88(14), 2010.

[6] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing Objects by their Attributes. In *CVPR*, 2009.

[7] P. Felzenszwalb, B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *PAMI*, 2010.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[9] D. Hall and P. Perona. Fine-Grained Classification of Pedestrians in Video: Benchmark and State of the Art. In *CVPR*, 2015.

[10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. In *arXiv preprint arXiv:1408.5093*, 2014.

[11] J. Joo, S. Wang, and S.-C. Zhu. Human Attribute Recognition by Rich Appearance Dictionary. In *ICCV*, 2013.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *CVPR*, 2006.

[14] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *CVPR*, 2014.

[15] D. Parikh and K. Grauman. Relative Attributes. In *ICCV*, 2011.

[16] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN Features off-the-shelf: an Astounding Baseline for Recognition. In *CVPR Workshop*, 2014.

[17] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. In *ICLR*, 2014.

[18] G. Sharma and F. Jurie. Learning Discriminative Representation for Image Classification. In *BMVC*, 2011.

[19] G. Sharma, F. Jurie, and C. Schmid. Expanded Parts Model for Human Attribute and Action Recognition in Still Images. In *CVPR*, 2013.

[20] B. Siddiquie, R. S. Feris, and L. S. Davis. Image Ranking and Retrieval based on Multi-Attribute Queries. In *CVPR*, 2011.

[21] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How Transferable are Features in Deep Neural Networks. In *NIPS*, 2014.

[22] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. PANDA: Pose Aligned Networks for Deep Attribute Modeling. In *CVPR*, 2014.

[23] N. Zhang, F. I. R. Farrell, and T. Darrell. Deformable Part Descriptors for Fine-grained Recognition and Attribute Prediction. In *ICCV*, 2013.