AL-DDCNN : A Distributed Crossing Semantic Gap learning for Person Re-identification

Keyang Cheng[1], Yongzhao Zhan[1], Man Qi[2]

1 School of ComputerScience & TelecommunicationsEngineering,JiangsuUniversity,Zhenjiang,Jiangsu212013,China

2 Department of Computing, Canterbury Christ Church University, Canterbury, UK

Abstract: Due to the often low video quality and high camera position, it is difficult to get clear human faces. Person re-identification across nonoverlapping camera views is a challenging computer vision task. In this paper, a novel approach called Attribute learning based on Distributed Deep Convolutional Neural Network (AL-DDCNN) model is proposed to deal with person re-identification task. It shows how mid-level "semantic attributes" can be generated for person description and further shows how this attribute-based description can be used in synergy with low-level feature descriptions to improve re-identification accuracy when author-topic model is employed to map category. Besides, considered the ability to operate on raw pixel input without the need to design special features, Deep Convolutional Neural Network is employed to generate features without supervision for attributes learning model. To overcome the model's weakness in computing seep, parallelized implementations such as distributed parameter manipulation and attributes learning are employed in AL-DDCNN model. Experiments show that the proposed approach achieve state-of-the-art recognition performance in the VIPeR data set and is with a good semantic explanation which cannot be given by other methods.

Keywords: Person Re-identification, Attribute learning, Deep Convolutional Neural Network, Distributed computing

1. Introduction

Person re-identification is the task of recognizing an individual by diverse scenes across different cameras. Monitored over space and time, a pedestrian will disappear from one camera, but perhaps will appear in another view with different viewing angle and lighting condition, which makes it become difficult to re-identify individuals in different cameras. Re-identification in large camera networks by human operators is a boring and inaccurate work. Manual matching operation will make significant re-identification accuracy gaps between different human operators [1]. Re-identification performance is good or not subjectively depended on individual operator's experience that makes it become difficult to transfer and also subject to operator bias [2]. There are many efforts to deal with theses problem and attempt to make an auto re-identification system[3], but despite extensive research, there are still difficult to auto re-identify pedestrian accurately. The most reason is that traditional features are insufficiently discriminate for cross-view pedestrian re-identification with varying view angle and lighting. Because of easily be obtained and measured by machine, low-level features, such as color, shape and texture [4], are typically employed to re-identify pedestrian. Ethnicity, gender and age are eager to be obtained, which would be most helpful to the task of pedestrian re-identification. But these high-level features are exceptionally difficult to get and measure reliably in surveillance video. Recently, a new mid-level attribute features are employed in image classification as a medium between low-level features and class. As far as person re-identification is concerned, mid-level attribute features,

such as hair style, shoe type or clothing style can be measured reasonably reliably with modern computer-vision techniques. In this paper, we will discuss how to obtain the mid-level attribute features from low-level features and how to use them into pedestrian re-identification.

2. Related Work

2.1 Attributes and their applications

Different from low-level features or high-level classes, attributes provide the mid-level description between images and their classes. There are various unsupervised (e.g. PCA or topic-models) or supervised (e.g. neural networks) modelling approaches which produce data-driven mid-level representations, which aim to project the data onto a basis set defined by the assumptions of the particular model (e.g. maximisation of variance, likelihood or sparsity). In contrast, attribute learning focuses on representing data instances by projecting them onto a basis set defined by domain-specific axes which are semantically meaningful to humans. Recent work shows that attributes are useful in a variety of settings. First, they are independently useful to describe familiar and unfamiliar things (e.g., the leopard is spotted and furry, whether or not we know to call it a leopard [5]), or to search through large image/video collections in semantic terms [6]. Second, they enable new zero-shot learning paradigms, where one can build an object model on the fly [7]. Third, they can serve as mid-level features to an object classification layer; having learned to predict the presence of each attribute, one can build supervised object models on top of those predictions [8,9,10]. Usually attribute-object associations are manually specified, but some work explores ways to obtain them automatically [11,12]. Notably, nearly all models using attributes for recognition learn them independently.

Visual attributes have received increasing interests in the past three years for classification problems ranging from image categorisation [13,14,15], person re-identification [16], to action and video event recognition[17]. Attributes are either user defined based on prior knowledge [14,16] or data driven or latent and discovered from data [17,18]. The former has clear semantic meaning and the latter not necessarily so. On the other hand, manually defined attributes may not be computable consistently nor discriminative sufficiently despite additional human annotation, from which data driven attributes do not suffer.

Throughout the research of person identification properties study in recent years, there are few to discuss the relationship between attributes and features, namely, how to determine what attribute is suitable to describe human with data driving. Recent years, deep learning is employed to select features. In this paper, we will discuss how to use deep learning to obtain suitable attributes.

2.2 Deep learning

In recent researches, with the steady advance of deep learning [19,20] and unsupervised feature learning [21], learnable features gain significant attentions. Specially, the Deep Convolutional Neural Network (DCNN) proposed by Krizhevsky et al. [22] achieved record-breaking results in ImageNet Large Scale Visual Recognition Challenge 2012. Afterwards, its specific network structure has been widely used in image classification and object detection [23,24]. In [25], Donahue et al. showed that features generated from a classifying CNN perform excellently in related vision tasks, implying that DCNN can be used as a generic feature extractor.

In the field of pedestrian detection, many feature learning and deep learning methods have been introduced recently. In [26,27,28], Sermanet et al. proposed a two layers convolutional model and layers were pre-trained by convolutional sparse coding. In [29], Ouyang et al. conducted Restricted Boltzmann Machine (RBM) in modeling mutual visibility relationship for occlusion handling. And in

[29] authors further cooperated with Convolutional Neural Network, and proposed a joint deep learning framework that jointly consider four key components in pedestrian detection: feature extraction, deformation model, occlusion model and classifier. In [30], Convolutional Neural Network has been successfully applied in pedestrian detection, where the used network structure have only 2 layers. In contrast, Krizhevsky's CNN [22] that has 7 layers is much deeper.

Nowadays, distributed computing system such as Hadoop, Spark are employed in many real time system of large scale data. Distributed Deep Convolutional Neural Network (DDCNN) has already become a research focus.

In this paper, we will discuss how to employ Distributed Deep Convolutional Neural Network(DDCNN) to discover attributes and how to use DDCNN to set attribute model and target classifier. At last, we will show the experiment results of our Attribute Learning based on Distributed Deep Convolutional Neural Network (AL-DDCNN) model with milti-GPU Parallel integrate.

3. Method

3.1 Attribute obtain

As in most popular person re-identification methods, the feature representation approach described in the previous subsection reflects low-level visual features[32]. However, identifying people is a high-level task. There is a semantic gap between low-level feature representation and high-level task , which are recognized as middle-level description of people. Therefore, embedding attribute layers into the identification provides a possible way to bridge the semantic gap.

In the literature of computer vision, attributes are obtained by two approaches. In the work of Yamaguchi et al. [33], the attributes are crawled from the fashion website. Although such a data acquisition method can provide plenty of attributes, these attributes are not suitable for person re-identification. For example, in [33], jacket and coat are both annotated, but they are high-level clothes concepts and difficult to distinguish in low-quality surveillance videos. For the person re-identification task, the attributes should be visually separable in the surveillance video scenario.

Besides mining Web data, another approach to obtain attributes is manual annotation. Layne et al. [34] annotated 15 binary-valued and utilized them in person re-identification. In this work, there define 11 kinds, to describe the appearance of people.

But even if we can afford to ask domain experts to provide a list of attributes most descriptive of the images we wish to categorize, there is no guarantee that those attributes will be sufficiently separable in the image feature space—a necessary condition if they are intended to serve as the mid-level cues for recognition. On the other hand, even though we have abundant machine learning tools to discover discriminative splits in image feature space that together carve out each object of interest, there is no guarantee that any such features will happen to correspond to human-nameable traits—a desirable condition if we are to leverage the transfer, description, and other attractive aspects mentioned above.

Figure 1 shows an overview of our approach to mine both human understandable and discriminative attributes based on data driving. At each iteration t, we actively determine an attribute hypothesis (a hyper plane in the visual feature space) that helps discriminate among classes that are most confused given the current collection of attributes A_t. We then estimate the probability that the hypothesis is nameable, using a learned model of nameability that is continually augmented by any hypotheses accepted (i.e. named) by the human in the loop. If it appears unnameable, we discard it and

loop back to select the next potential attribute hypothesis. If it appears nameable, the system creates a visualization of the attribute using a subset of training images, presents the images to the annotator, and requests an attribute label. The annotator may either accept and name the hypothesis, or reject it. If it is accepted, we append this new named attribute a_j to our discovered vocabulary, $A_{t+1} = [A_t ; a_j]$, retrain the higher level classifier accordingly, and update our nameability model. If it is rejected, the system loops back to generate a new attribute hypothesis. Thus, only those attributes that are named by the user are added to the pool and can be used for recognition. The discovery loop terminates once human resources are exhausted, or when a desired number of named attributes have been collected.

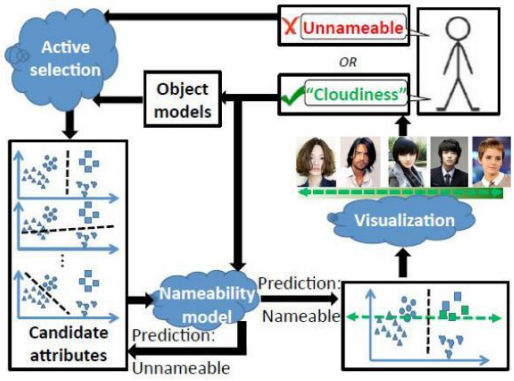
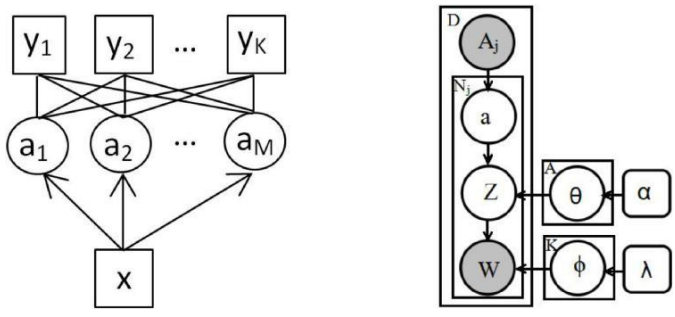


Figure 1. Overview of our attribute obtainment approach

3.2 Attribute Model and Target Classifier

Traditional attribute model is shown as Figure 2.a. Attributes are the media layer between samples and classes. The attribute model we employed in our method is the Author-Topic (AT) model (Figure 2.b) . The AT model is originally designed to model the interests of authors from a given document corpus[35]. In this paper, we extend the AT model to describe the distribution of image features related to attributes. Indeed, authors of a document and attributes of an image category have many similarities, which allow us to analogize the latter to the former: a document can have multiple authors and an object category can have multiple attributes; an author can write multiple documents and an attribute can be presented in multiple object categories. Nevertheless, there is also noticeable difference between them: each document can have a distinct list of authors, while all images within an object category share a common list of Attributes.



(a) (b)

Figure 2. Attribute Model. The left is traditional attribute model. The right is author-topic attribute model.

The AT model is a generative model. In this model, an image x_j has a list of attributes, denoted by A_j. An attribute l in A_j is modeled by a discrete distribution of K topics, which parameterized by a

K-dim vector $_l$ $(_{l1},\ldots,_{lK})$ with topic k receiving weight $_{lk}$. The topic k is modeled by a discrete distribution of w codewords in the lexicon, which is parameterized by a W-dim vector $(_{l\,k1},\ldots,_{kW})$ with codeword v receiving weight $_{kv}$. Symmetric Dirichlet priors are placed on and , with $_l \sim$ Dirichlet( ), and $_k \sim$ Dirichlet( ), where andare hyper parameters that affect the sparsity of these distributions. The generative process is outlined in Algorithm 1.

Algorithm 1. The generative process of the Author-Topic model

---

1: given the attribute list A_j and the desired number of visual words in image x_j, N_j

2: for i = 1 to N_j do

3: conditioning on A_j, choose an attribute $a_{ji} \sim$Uniform(A_j)

4: conditioning on a_{ji}, choose a topic $z_{ji} \sim$Discrete($_{a\,ji}$), where $_l$ defines the distribution of topics for attribute a l

5: conditioning on z_{ji}, choose a visual word $w_{ji} \sim$ Discrete($_{z\,ji}$), where $_k$ defines the distribution of visual words for topic z k

6: end for

Given a training corpus, the goal of inference in an AT model is to identify the values of and . In [36], Rosen-Zvi et al. presented a collapsed block Gibbs sampling method. The "collapse" means that the parameters and are analytically integrated out, and the "block" means that we draw the pair of (a_{ji}, z_{ji}) together. The pair of (a_{ji}, z_{ji}) is drawn according to the following conditional distribution:

$$p(a_{ji} \quad l, z_{ji} \quad k \mid w_{ji} \quad v, \ ) \quad \frac{/K \ N_l^k{}_{,\backslash ji}}{{}_k^K{}_1 N_l^k{}_{,\backslash \ ji}} \ \frac{/W \ C_k^v{}_{,\backslash ji}}{{}_v^W{}_1 C_k^v{}_{,\backslash \ ji}} \quad (1)$$

where $\{A_j, z_{\backslash ji}, a_{\backslash ji}, w_{\backslash ji}, , \}$, the subscript ji represents the i-th visual word in image x_j, a_{ji} l and z_{ji} k represent the assignments of current visual word to attribute l and topic k respectively, w_{ji} v represents the observation that the current visual word is the v-th codeword in the lexicon, $z_{\backslash ji}$ and $a_{\backslash ji}$ represent all topic and attribute assignments in the training corpus excluding the current visual word, $N^k_{l,\backslash ji}$ is the total number of visual words that are assigned to attribute l and topic k, excluding w_{ji}, and $C^v_{k,\backslash ji}$ is the total number of visual words with value v that are assigned to topic k, excluding w_{ji}.

To run the Gibbs sampling algorithm, we first initialize a and z with random assignments. In each Gibbs sampling iteration, we draw samples of a_{ji} and z_{ji} for all visual words in the training corpus according to the distribution in Equation (1) in a randomly permuted order of i and j. The samples of a and z are recorded after the burn-in period. In experiments, we observe 200 iterations are sufficient for the sampler to be stable. The posterior means of and can then be estimated

using the recorded samples as follows:

$$\hat{}_{lk} \quad \frac{/K\ N_l^k}{{}_k^K{}_1 N_l^k} \quad , \quad \hat{}_{kv} \quad \frac{/W\ C_k^v}{W_{v\,1} C_k^v} \qquad (2)$$

where N_l^k and C_k^v are defined in a similar fashion as in Equation (1), but without excluding the instance indexed by ji.

If the attribute list is unique in each category, an AT model can be used to classify a image by the maximum likelihood criterion. Suppose we have learned $_l$ for every $l\ 1,...,A$ from the source categories, we can then use them in classifying an image of a target category using the approximate likelihood

$$p(w_t \mid y\ m, A_m, D \quad {}^{train}{}_{)kw_{ti}} \quad {}^{N_{tK}\ \hat{}}_{i\,1\ k\,1} \quad (\frac{1}{|A_m|} \quad {}^{\hat{}}_{lk\ l\ A_m})_{kw_{ti}} \quad {}^{N_{tK}\ \hat{}}_{i\,1\ k\,1} \quad \sim_{mk} \qquad (3)$$

where A_m is the attribute list associated to a target category $y\ m$, $|A_m|$ is the length of A_m. In the above equations, we have constructed a pseudo weight for the category-specified topic distribution

$$\sim \quad \frac{1}{|A|\ l\ a}$$

$_{mk}$ before we see the real training examples of the category. So our approach can be used to predict unseen categories, namely shot learning problem. Given a threshold value, the probability in Equation (3) can also used to rank images for query.

3.3 Deep Convolutional Neural Network & Feature Extraction

In order to make the data driving attribute model work for classify, Deep Convolutional Neural Network (DCNN) is employed in the feature extraction stage. Following the network architecture proposed by Krizhevsky et al. [22], we used the RCNN package [22] which utilize the Caffe [31] to implement DCNN.

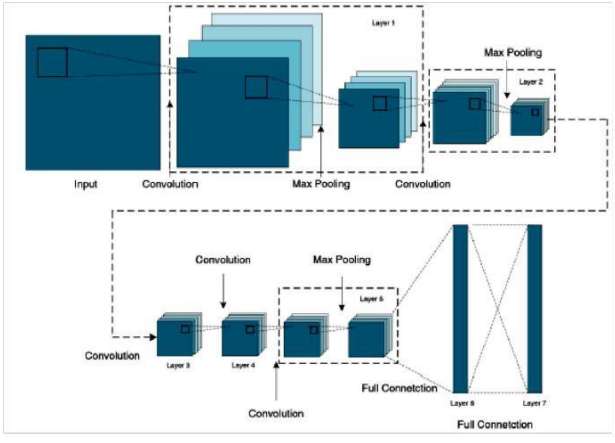


Figure 3. Architecture of DCNN

The architecture of used DCNN is presented in Figure 3, which has 7 layers. Notice that the DCNN requires input images of 128×48 pixels size, so first we simply warp candidate windows to the required size. In the first layer, the images are filtered with 96 kernels of size 11×11×3 pixels with a stride of 4 pixel, then max-pooling is applied in 3×3 grid. The second layer has the same pipeline as first layer, with 256 kernels of size 5×5×48, and max-pooling in 3×3 grid. Afterwards, there are two convolution layers without pooling, which both contains 384 kernels. In the fifth layer, again, the output of previous layer is first convoluted with 256 kernels then applied spatial max-pooling in 3×3 pixel grid. The last two layers of the network are fully connected layer, which both contains 4096 nodes respectively. The DCNN eventually output features of 4096 dimensions from the last layer. The activation function used in the convolution and full connected layer is Rectified Linear function f(x) = max(0, x). For more details about network parameters and training protocol, we refer reader to [19].

The DCNN is employed to unsupervised learn features from original image. This is the base of attributes target recognition. The overview of attribute learning based DCNN is shown as Figure 4.

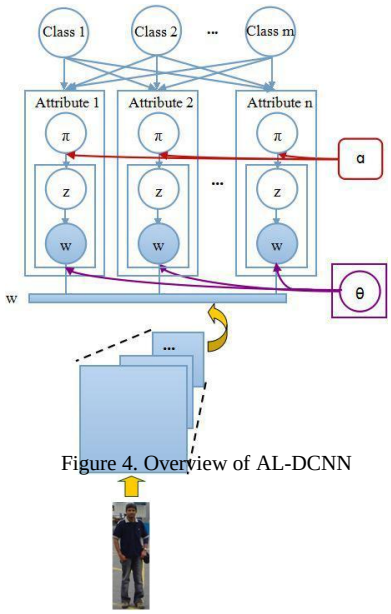


Figure 4. Overview of AL-DCNN

Figure 4. Overview of attribute learning based DCNN

3.4 Parallelized implementations of attribute learning based on DCNN

DCNN have been shown to excel at classification tasks, and its ability to operate on raw pixel input without the need to design special features is very appealing. However, it is notoriously slow at inference time. So we employ parallelized implementations to speed up recognition. The key idea of our model's parallelized implementations is distributed parameter manipulation and attributes learning. As far as distributed parameter manipulation concerned, instead direct accessing to the model parameters, the coordinator issues commands drawn from a small set of operations that can be performed by each parameter server shard independently, with the results being stored locally on the same shard. Additional information, e.g the history cache for the optimization algorithm, is also stored on the parameter server shard on which it was computed[37]. This allows running large models without incurring the overhead of sending all the parameters and gradients to a single central server. In the parallelized implementations, data is distributed to many machines and the results are sent back to a central parameter server. Many such methods wait for the slowest machine, and therefore do not scale well to large shared clusters. To account for this problem, we employ the following load balancing scheme: The coordinator assigns each of the N model replicas a small portion of work, much smaller

than 1/Nth of the total size of a batch, and assigns replicas new portions whenever they are free. With this approach, faster model replicas do more work than slower replicas. To further manage slow model replicas at the end of a batch, the coordinator schedules multiple copies of the outstanding portions and uses the result from whichever model replica finishes first. This scheme is similar to the use of "backup tasks" in the MapReduce framework [38]. Prefetching of data, along with supporting data affinity by assigning sequential portions of data to the same worker makes data access a non-issue.

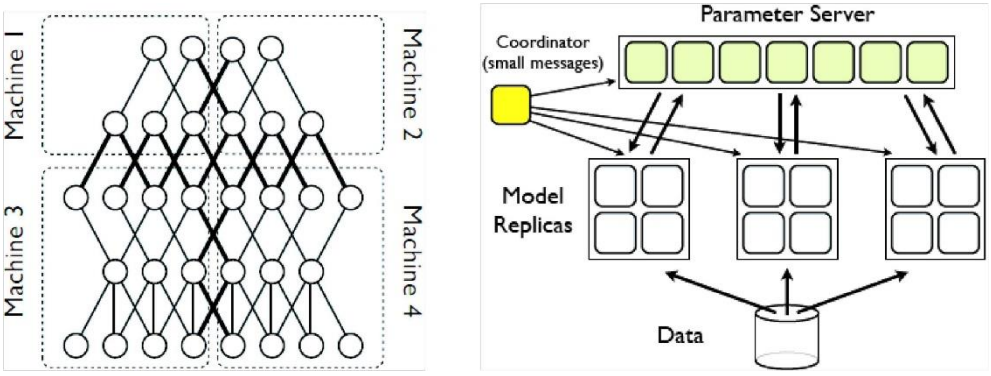


Figure 5. Distributed parameter manipulation of AL-DCNN

Another parallelized implementation idea of our model is distributed attributes learning. The attributes learning model is partitioned across several machines (Figure 6), so that the learning task of different attribute is assigned to different machine. The framework automatically parallelizes computation in each machine using all available cores, and manages communication, synchronization and information transfer between machines during both training and inference.

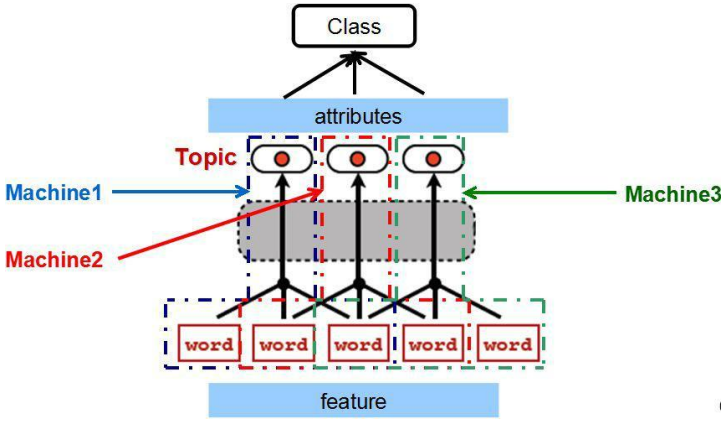


Figure 6. Distributed attributes learning of AL-DCNN

4. Experiments

4.1 Datasets and feature selection

We selected two challenging datasets with which to validate our model, the VIPeR dataset introduced by Gray et al.[39]. VIPeR is comprised of 632 pedestrian image pairs from two cameras with different viewpoint, pose and lighting conditions. The images are uniformly scaled to 128×48 pixel size. We follow [39,40] in considering Cam B as the gallery set and Cam A as the probe set. Performance is evaluated by matching each test image in Cam A against the Cam B gallery. We follow [40] in randomly selecting one image for each pedestrian to build a gallery set, while the others form the probe set. This is repeated 10 times and the results averaged. We split the data set into a training and a test set.

In the feature selection stage, we use DCNN to obtain features without supervised learning. Figure 7. shows activation of DCNN for an input example. Each panel shows the convolutional layer, the normalization and pooling, then the 1x1 convolutional one and finally the fully-connected one.

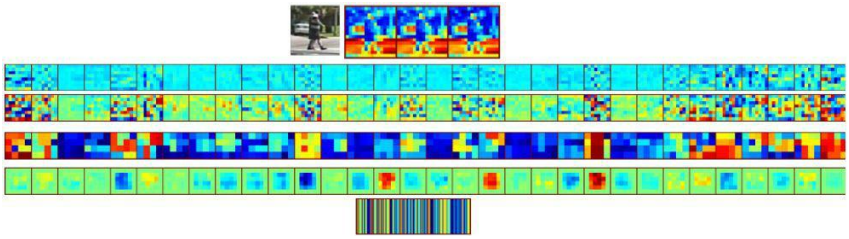


Figure 7 Activation of DCNN for an input example

4.2 Attribute Visualizing & Obtaining

In order to display the attribute hypothesis to the annotator, we wish to convey the difference in the images that lie on either side of the hyperplane, while ensuring that within the constraints of finite data, we show only the changes induced along the direction orthogonal to that hyperplane. To do this, we first consider the range from the hyperplane within which 95% of the training data falls, in order to disregard potential outlier instances. We divide this range into 15 equidistant bins, and select three images per bin that are closest to the median along all other dimensions, yielding a 3×15 collage. Figure 7 shows an example of attribute visualizing and naming. The attributes mined by our approach are shown in Figure 8 . Individuals in the training data set were labeled with these attributes.

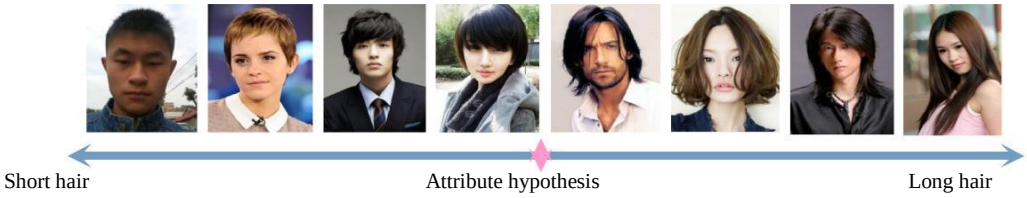


Figure 7. visualize an attribute hypothesis

Figure 8. Obtained attributes

4.3 Attribute prediction

To make attribute-to-class mapping come true, the prediction of the attributes for a specific image is critical procedure. At the beginning of the process, the responses of test persons were averaged to determine the real-valued association strength between attributes and the picture. Binary sample-attribute matrices are obtained by thresholding at the overall matrix mean. Some examples are shown in Figure 9. We also investigated the role of individual attribute with respect to prediction accuracy. We use images from test set labeled attributes by human to test the quality of individual attribute predictors. The results shown in Figure 10 can give us an observation regarding the contribution of each attribute. Some attributes are more difficult to predict than others. This is expected, since, intuitively, ability to identify a specific attribute is related to its ubiquitousness. Better defined attributes are easier to identify than more fuzzy ones.

	Hat	Short hair	Long hair	Longs single color	Longs multi-color	Shorts	Long sleeve	No sleeve	Backpack
	No	Yes	No	No	Yes	No	No	Yes	Yes
	No	Yes	No	No	Yes	No	Yes	No	Yes
	No	Yes	No	No	Yes	Yes	No	No	Yes
	No	Yes	No	No	Yes	No	No	No	No
	No	No	Yes	No	Yes	Yes	No	No	Yes
	Yes	No	No	No	Yes	Yes	No	No	No
	No	No	Yes	No	Yes	No	Yes	No	No
	No	Yes	No	No	Yes	No	Yes	No	No
	No	No	Yes	Yes	No	No	Yes	No	No

Figure 9. Examples with binary sample-attributes matrices

To enable attribute-to-class mapping, the accuracy of the attribute prediction for a specific image is important. Figure 9 illustrates some classes and their corresponding attributes automatically learned with our system. We also investigated the role of individual attribute with respect to classification accuracy. First, figure 10 gives the test accuracy of our attribute learner. It can be seen that some attributes are more difficult to predict than others.

This is as expected, since, intuitively, ability to identify a specific attribute is related to its ubiquitousness. Better defined attributes are easier to be identified. Secondly, Figure 11 visualizes the attribute-class matrix which reveals the correlation between each attribute and each category.

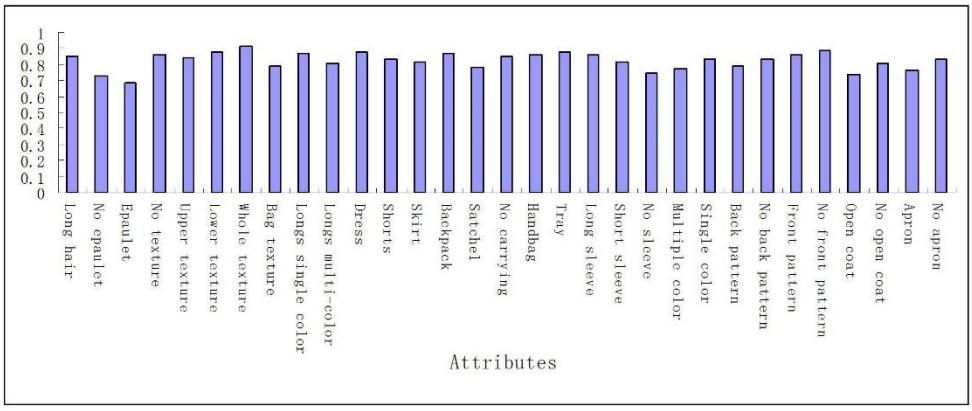


Figure 10. Quality of individual attribute predictors

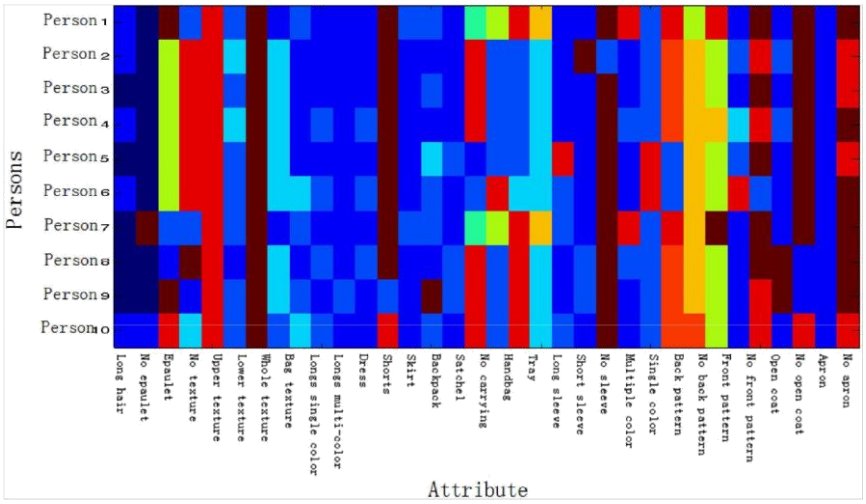


Fig. 11. Visualization ofattribute–class matrix.

4.3 Person Re-identification matching comparison

Following the classification and evaluation protocol discussing above, we randomly sample half of the dataset, i.e., 316 image pairs, for training, and the remaining for test. In the first round, images from CAM A are used as probe and those from CAM B as gallery. Each probe image is matched with every gallery image, and the correctly matched rank is obtained. Rank-k recognition rate is the expectation of the matches at rank k, and the CMC curve is the cumulated values of recognition rate at all ranks. After this round, the probe and gallery are switched. We take the average of the two rounds of CMC curves as the result of one trial. 10 trials of evaluation are repeated to achieve stable statistics, and the average result is reported.

Table 1 Comparison results on VIPeR

Method	r=1	r=5	r=10	r=20
DF	12.00	22.00	34.00	43.00
ELF	12.00	31.00	41.00	58.00
bLDFV	22.34	47.00	60.04	71.00
SDALF	19.87	38.89	49.37	65.73
SDC-knn	26.54	40.03	47.89	54.76
SDC-ocsvm	26.29	46.57	58.84	72.72
eSDC-knn	26.31	46.61	58.86	72.77
eSDC-ocsvm	26.74	50.70	62.37	76.36
AL-DDCNN	**26.76**	**52.64**	**63.45**	**77.27**

Since DF, ELF, bLDFV, SDALF, SDC-knn, SDC-ocsvm, eSDC-knn and eSDC-ocsvm[41,42] have published their results on the VIPeR dataset, they are used for comparison. The splitting assignments in these approaches are used in our experiments. Figure 12 report the comparison results. It is observed that our approach outperform all these benchmarking approaches. In particular, matching rate is around 27% at rank 1 and is around 77% at rank 20 for our AL-DDCNN. Figure 13 shows examples of the query results of our proposed method on the VIPeR database. Probe images from one view are shown in the left-most column and the top18 query results are sorted from left to right. The correct matches are indicated by the red boxes. The right-most column shows the true matches. The bottom row shows a failed attempt. The correct match failed to show up in the top18

queries.This is because the appearance of this individual was radically altered in the different views. The recognition accuracy of our proposed AL-DDCNN compared with the state-of-the-art works on VIPeR with different number of searching classes are summarized in Table 1. From these figures, it is clear that AL-DDCNN gives the best results. Although our method does not provide a significant gain compared to the state-of-the-art method, it gives a good semantic explanation which cannot be given by other methods. Besides, our method can be used to deal with "zero-shot" scenario in which a visual probe is unavailable but re-identification can still be performed with user-provided semantic attribute description.

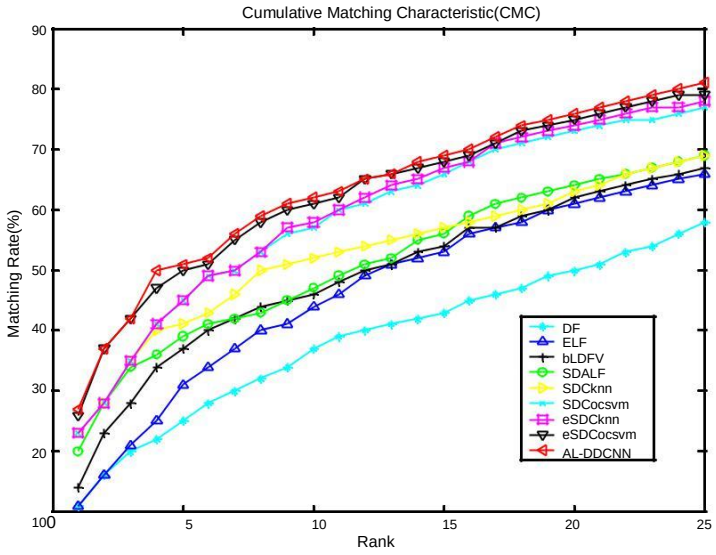


Figure 12. Rate of cumulative matching characteristic

Figure 13. Example of query results using AL-DCNN

4.5 Increase speed of Model parallelism

To explore the performance of our model parallelism, we measured the mean time for training the number of partitions (machines) used in our model. In Figure 14 we quantify the impact of parallelizing across N machines by reporting the average training speed-up: the ratio of the time taken using only a single machine to the time taken using N.

Figure 14 shows that the moderately sized model e.g. our 34 attributes model, runs fastest on 8 machines, computing 2.2 faster than using a single machine. Partitioning the model on more than 8 machines actually slows training, as network overhead starts to dominate in the fully-connected network structure and there is less work for each machine to perform with more partitions. In contrast, the model with more parameters or attributes will benefit more from the use of additional machines than do model with fewer parameters.

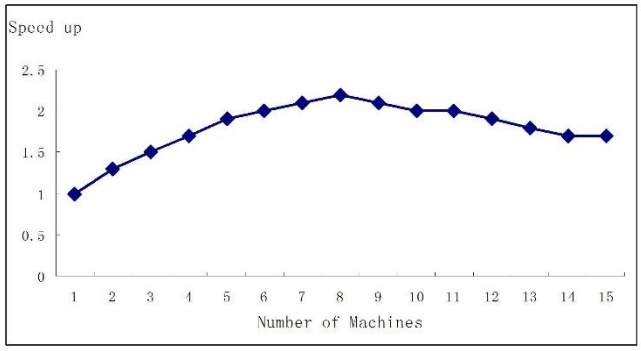


Figure 14. Increase speed of model parallelism with different number of machines

5. Conclusion

In this paper, we describe how to use middle-level attributes to assist person re-identification. Instead of training for the recognition of a specific category of person directly based on the manually designed features such as SIFT and HoG, a series of visual attributes are extracted from a given set of images, which consider both human understandable and discriminative demand. As the media layer between samples and classes, attributes are play key role in our novel approach. To generate features without supervision for the attributes learning model, Deep Convolutional Neural Network is employed to generate features. In addition, parallelized implementations such as distributed parameter manipulation and attributes learning are employed to make the model speed up. Experiments in dataset VIPeR show that the proposed approach achieve state-of-the-art recognition performance and is with a good semantic explanation.

Acknowledgment

This research is supported by the national science foundation of China (NFSC) No.61170126, the science foundation of Jiangsu province No.BK20150527 and the science foundation of Zhenjiang city No.SH2014017

References

[1]Gray D, Tao H. Viewpoint invariant pedestrian recognition with an ensemble of localized features[M]//Computer Vision–ECCV 2008. Springer Berlin Heidelberg, 2008: 262-275.

[2]Prosser B, Zheng W S, Gong S, et al. Person Re-Identification by Support Vector Ranking[C]//BMVC. 2010, 2(5): 6

[3]Zheng W S, Gong S, Xiang T. Person re-identification by probabilistic relative distance comparison[C]//Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011: 649-656.

[4]Prosser B, Gong S, Xiang T. Multi-camera matching under illumination change over time[C]//Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications-M2SFA2 2008.

[5]Fu Y, Hospedales T M, Xiang T, et al. Attribute learning for understanding unstructured social activity[M]//Computer Vision–ECCV 2012. Springer Berlin Heidelberg, 2012: 530-543

[6]Lampert C H, Nickisch H, Harmeling S. Learning to detect unseen object classes by between-class attribute transfer[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 951-958

[7]Liu J, Kuipers B, Savarese S. Recognizing human actions by attributes[C]//Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011: 3337-3344

[8]Farhadi A, Endres I, Hoiem D, et al. Describing objects by their attributes[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 1778-1785.

[9]Wang G, Forsyth D. Joint learning of visual attributes, object classes and visual saliency[C]//Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009: 537-544

[10]Kumar N, Belhumeur P, Nayar S. FaceTracer: A search engine for large collections of images with faces[M]//Computer Vision–ECCV 2008. Springer Berlin Heidelberg, 2008: 340-353.

[11]Ferrari V, Zisserman A. Learning visual attributes[C]//Advances in Neural Information Processing Systems. 2007: 433-440.

[12]Kumar N, Berg A C, Belhumeur P N, et al. Attribute and simile classifiers for face verification[C]//Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009: 365-372.

[13]Chen K, Gong S, Xiang T, et al. Cumulative attribute space for age and crowd density estimation[C]//Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013: 2467-2474.

[14]Lampert C H, Nickisch H, Harmeling S. Learning to detect unseen object classes by between-class attribute transfer[C]//Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 951-958. [15]Parikh D, Grauman K. Relative attributes[C]//Computer Vision (ICCV), 2011 IEEE International Conference on. IEEE, 2011: 503-510.

[16]Layne R, Hospedales T M, Gong S, et al. Person Re-identification by Attributes[C]//BMVC. 2012, 2(3): 8

[17]Fu Y, Hospedales T M, Xiang T, et al. Attribute learning for understanding unstructured social activity[M]//Computer Vision–ECCV 2012. Springer Berlin Heidelberg, 2012: 530-543

[18]Liu J, Kuipers B, Savarese S. Recognizing human actions by attributes[C]//Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011: 3337-3344.

[19]Chen X, Wei P, Ke W, et al. Pedestrian Detection with Deep Convolutional Neural Network[C]//Computer Vision-ACCV 2014 Workshops. Springer International Publishing, 2014: 354-365.

[20]Bengio Y. Learning deep architectures for AI[J]. Foundations and trends® in Machine Learning, 2009, 2(1): 1-127.

[21]Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013, 35(8): 1798-1828.

[22]Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[25]Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on. IEEE, 2014: 580-587.

[26]Sermanet P, Eigen D, Zhang X, et al. Overfeat: Integrated recognition, localization and detection using convolutional networks[J]. arXiv preprint arXiv:1312.6229, 2013.

[27]Zeiler M D, Fergus R. Visualizing and understanding convolutional networks[M]//Computer Vision–ECCV 2014. Springer International Publishing, 2014: 818-833.

[28]Sermanet P, Kavukcuoglu K, Chintala S, et al. Pedestrian detection with unsupervised multi-stagefeature

learning[C]//Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013: 3626-3633
[29]Ouyang W, Zeng X, Wang X. Modeling mutual visibility relationship in pedestrian detection[C]//Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013: 3222-3229.
[30]Ouyang W, Wang X. Joint deep learning for pedestrian detection[C]//Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, 2013: 2056-2063.
[31]Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the ACM International Conference on Multimedia. ACM, 2014: 675-678
[32]Li A, Liu L, Yan S. Person Re-identification by Attribute-Assisted Clothes Appearance[M]//Person Re-Identification. Springer London, 2014: 119-138
[33]Yamaguchi K, Kiapour M H, Ortiz L E, et al. Parsing clothing in fashion photographs[C]//Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012: 3570-3577
[34]Layne R, Hospedales T M, Gong S. Towards person identification and re-identification with attributes[C]//Computer Vision–ECCV 2012. Workshops and Demonstrations. Springer Berlin Heidelberg, 2012: 402-412.
[35]Yu X, Aloimonos Y. Attribute-based transfer learning for object categorization with zero/one training example[M]//Computer Vision–ECCV 2010. Springer Berlin Heidelberg, 2010: 127-140.
[36]Rosen-Zvi M, Chemudugunta C, Griffiths T, et al. Learning author-topic models from text corpora[J]. ACM Transactions on Information Systems (TOIS), 2010, 28(1): 4
[37]Dean J, Corrado G, Monga R, et al. Large scale distributed deep networks[C]//Advances in Neural Information Processing Systems. 2012: 1223-1231
[38]Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113
[39]Gray D, Tao H. Viewpoint invariant pedestrian recognition with an ensemble of localized features[M]//Computer Vision–ECCV 2008. Springer Berlin Heidelberg, 2008: 262-275
[40] Ma B, Su Y, Jurie F. Local descriptors encoded by fisher vectors for person re-identification[C]//Computer Vision–ECCV 2012. Workshops and Demonstrations. Springer Berlin Heidelberg, 2012: 413-422
[41]arenzena M, Bazzani L, Perina A, et al. Person re-identification by symmetry-driven accumulation of local features[C]//Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010: 2360-2367
[42]Zhao R, Ouyang W, Wang X. Unsupervised salience learning for person re-identification[C]//Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013: 3586-3593