Learning Deep Feature Representations with Domain Guided Dropout for Person Re-identification

Tong Xiao Hongsheng Li Wanli Ouyang Xiaogang Wang
Department of Electronic Engineering, The Chinese University of Hong Kong
{xiaotong,hsli,wlouyang,xgwang}@ee.cuhk.edu.hk

Abstract

Learning generic and robust feature representations with data from multiple domains for the same problem is of great value, especially for the problems that have multiple datasets but none of them are large enough to provide abundant data variations. In this work, we present a pipeline for learning deep feature representations from multiple domains with Convolutional Neural Networks (CNNs). When training a CNN with data from all the domains, some neurons learn representations shared across several domains, while some others are effective only for a specific one. Based on this important observation, we propose a Domain Guided Dropout algorithm to improve the feature learning procedure. Experiments show the effectiveness of our pipeline and the proposed algorithm. Our methods on the person re-identification problem outperform state-of-the-art methods on multiple datasets by large margins.

1. Introduction

In computer vision, a *domain* often refers to a dataset where samples follow the same underlying data distribution. It is common that multiple datasets with different data distributions are proposed to target the same or similar problems. Multi-domain learning aims to solve the problem with datasets across different domains simultaneously by using all the data they provide. As deep learning arises in the recent years, learning good feature representations achieves great success in many research fields and real-world applications. The success of deep learning is driven by the emergence of large-scale training data, which makes multi-domain learning an interesting problem. Many studies [3, 10, 31] have shown that fine-tuning a deep model pre-trained on a large-scale dataset (*e.g.* ImageNet [8]) is effective for other related domains and tasks. However, in many specific areas, there is no such large-scale dataset for learning robust and generic feature representations. Nonetheless, different research groups have proposed many smaller datasets. It is necessary to develop an effective algorithm that jointly utilizes all of them to learn generic feature representations.

Figure 1. Examples of multiple person re-identification datasets. Each dataset has its own bias. Our goal is to learn generic feature representations that are effective on all of them simultaneously.

Another interesting aspect of multi-domain learning is that it enriches the data variety because of the domain discrepancies. Limited by various conditions, data collected by a research group might only include certain types of variations. Take the person re-identification [2, 23] problem as an example, pedestrian images are usually captured in different scenes (*e.g.*, campus, markets, and streets), as shown in Figure 1. Images in CUHK01 [21] and CUHK03 [23] are captured on campus, where many students wear backpacks. PRID [15] contains pedestrians in street views, where crosswalks appear frequently in the dataset. Images in VIPeR [13] suffer from significant resolution changes across different camera views. Each of such datasets is biased and contains only a subset of possible data variations, which is not sufficient for learning generic feature representations. Combining them together can diversify the training data, thus makes the learned features more robust.

In this paper, we present a pipeline for learning generic feature representations from multiple domains that are ef-

fective on all of them simultaneously. For concrete demonstration, we target the person re-identification problem, but the method itself would be generalized to other problems with datasets of multiple domains. As learning features from a large-scale classification dataset is proved to be effective [36], we first mix all the domains together and train a Convolutional Neural Network (CNN) to recognize person identities (IDs). The CNN model we designed consists of several BN-Inception [16, 37] modules, and its capacity well fits to the scale of the mixed dataset. This carefully designed CNN model provides us a fairly strong baseline, but the simple joint learning scheme does not take full advantages of the variations of multiple domains.

Intuitively, neurons that are effective for one domain could be useless for another domain because of the presence of domain biases. For example, only the i-LIDS dataset contains pedestrians with luggages, thus the neurons that capture luggage features are of no use when recognizing people from other domains.

Based on this observation, we propose *Domain Guided Dropout* — a simple yet effective method of muting non-related neurons for each domain. Different from the standard Dropout [14], which treats all the neurons equally, our method assigns each neuron a specific dropout rate for each domain according to its effectiveness on that domain. The proposed Domain Guided Dropout has two schemes, a deterministic scheme, and a stochastic scheme. After the baseline model is trained jointly with datasets of all the domains, we replace the standard Dropout with the deterministic Domain Guided Dropout and resume the training for several epochs. We observe that the proposed dropout scheme consistently improves the performance on all the domains after several epochs, especially on the smaller-scale ones. This step produces better generic feature representations that are effective on all the domains simultaneously. We further fine-tune the net with stochastic Domain Guided Dropout on each domain separately to obtain the best possible results.

The contribution of our work is three-fold. First, we present a pipeline for learning generic feature representations from multiple domains that perform well on all of them. This enables us to learn better features from multiple datasets for the same problem. Second, we propose Domain Guided Dropout to discard useless neurons for each domain, which improves the performance of the CNN. At last, our method outperforms state-of-the-arts on multiple person re-identification datasets by large margins. We observe that learning feature representations by utilizing data from multiple datasets improve the performance significantly, and the largest gain is 46% on the PRID dataset. Extensive experiments validate our proposed method and the internal mechanism of the method is studied in details.

2. Related Work

In recent years, training deep neural networks with multiple domains has been explored. Feature representations learned by Convolutional Neural Networks have shown their effectiveness in a wide range of visual recognition tasks [6, 12, 17, 20, 27, 44]. Long *et al*. [28] incorporated the multiple kernel variant of Maximum Mean Discrepancy (MMD) objective for regularizing the training of neural networks. Ganin *et al*. [11] proposed to reduce the distribution mismatch between the source and target domains by reversing the gradients of the domain classification loss, which is also utilized by [38] with a softlabel matching loss to transfer task information. Most of these methods aim at finding a common feature space that is domain invariant. However, our approach allows the representation to have disjoint components that are domain specific, while also learning a shared representation.

As deep neural networks usually contain millions of parameters, it is of great importance to reduce the parameter space by adding regularizations to the weights. The quality of the regularization method would significantly affect both the discriminative power and generalization ability of the trained networks. Dropout [14] is one of the most widely used regularization method in training deep neural networks, which significantly improves the performance of the deep model [20]. During the network training process, Dropout randomly sets neuron responses to zero with a probability of 0.5. Thus a training batch updates only a subset of all the neurons at each time, which avoids co-adaptation of the learned feature representations.

While the standard Dropout algorithm treats all the neurons equally with a fixed probability, Ba *et al*. [4] proposed an adaptive dropout scheme by learning a binary belief network to predict the dropout probability for each neuron. In practice, they use the response of each neuron to compute the dropout probability for itself. Our approach significantly differs from this method, as we propose to train a CNN from multiple domains, and utilize the domain information to guide the dropout procedure.

We target the person re-identification (Re-ID) problem in this work, which is very challenging and draws much attention in recent years [22, 24, 26, 39, 42, 45]. Existing Re-ID methods mainly address the problem from two aspects: finding more powerful feature representations and learning better metrics. Zhao *et al*. [46, 47, 48] proposed to combine SIFT features with color histogram as features. In deep learning literature, Li *et al*. [23] and Ahmed *et al*. [1] designed CNN models specifically to the Re-ID task and achieved good performance on large-scale datasets. They trained the network with pairs of pedestrian images and adopted the verification loss function. Ding *et al*. [9] utilized triplet samples for training features that maximize relative distance between the pair of same person and the pair

of different people in the triplets. Apart from the feature learning methods, a large number of metric learning algorithms [7, 19, 29, 32, 40, 41] have also been proposed to solve the Re-ID problem from a complementary perspective. Some recent works addressed the problem of mismatch between traditional Re-ID and real application scenarios. Liao *et al*. [25] proposed a database for open-set Re-ID. Zheng *et al*. [49] treated Re-ID as an image search problem and introduced a large-scale dataset. Xu *et al*. [43] raised the problem of searching a person inside whole images rather than cropped bounding boxes.

3. Method

Our proposed pipeline for learning CNN features from multiple domains consists of several stages. As shown in Figure 2, we first mix the data and labels from all the domains together, and train a carefully designed CNN from scratch on the joint dataset with a single softmax loss. This pretraining step produces a strong baseline model that works on all the domains simultaneously. Next, for each domain, we perform the forward pass on all its samples and compute for each neuron its average impact on the objective function. Then we replace the standard Dropout layer with the proposed Domain Guided Dropout layer, and continue to train the CNN model for several more epochs. With the guidance of which neurons being effective for each domain, the CNN learns more discriminative features for all of them. At last, if we want to obtain feature representations for a specific domain, the CNN could be further fine-tuned on it, again with the Domain Guided Dropout to improve the performance. In this section, we detail these stages, and compare our design choices with other alternatives.

3.1. Problem formulation

Although the pipeline itself is not limited to any specific scope, we target the person re-identification problem for concrete demonstration. The problem can be formulated as follows. Suppose we have D domains, each of which consists of N_i images of M_i different people. Let $\{(x_i^{(j)}, y_i^{(j)})_{j=1}^{N_i}\}_{i=1}^{D}$ denote all training samples, where $x_i^{(j)}$ is the j-th image of the i-th domain, and $y_i^{(j)} \in \{1, 2, \ldots, M_i\}$ is the identity of the corresponding person. Our goal is to learn a generic feature extractor $g(\cdot)$ that has similar outputs for images of the same person and dissimilar outputs for different people. During the test phase, given a probe pedestrian image and a set of gallery images, we use $g(\cdot)$ to extract features from all of them, and rank the gallery images according to their Euclidean distances to the probe image in the feature space. For the training phase, there are several frameworks that use pairwise [1, 23] or triplet [33] inputs for learning feature embeddings. In our approach, we train a CNN to recognize the identity of each person, which is also adopted in the face verification work [36].

3.2. Joint learning objective and the CNN structure

When mixing all the D domains together, a straightforward solution is to employ a multi-task objective function, *i.e*., learning D softmax classifiers $f_1, f_2, \ldots, f_D$ and a shared features extractor g that minimize

$$\arg\min_{f_1, f_2, \ldots, f_D, g} \sum_{i=1}^{D} \sum_{j=1}^{N_i} \mathcal{L}\left(f_i(g(x_i^{(j)})), y_i^{(j)}\right), \qquad (1)$$

where $\mathcal{L}$ is the softmax loss function that equals to the cross-entropy between the predicted probability vector and the ground truth.

However, since different person re-identification datasets usually have totally different identities, it is also safe to merge all $M = \sum_{i=1}^{D} M_i$ people together and relabel them with new IDs $y' \in \{1, 2, \ldots, M\}$. For the merged dataset, we can define a single-task objective function, *i.e*., learning one softmax classifier f and the features extractor g that minimize

$$\arg\min_{f, g} \sum_{i=1}^{D} \sum_{j=1}^{N_i} \mathcal{L}\left((f \circ g)(x_i^{(j)}), y'^{(j)}_i\right). \qquad (2)$$

Compared with the multi-task formulation, this single-task learning scheme forces the network to simultaneously distinguish people from all domains. The feature representations capture two types of information: domain biases (*e.g*., background clutter, lighting, *etc*.) as well as person appearance and attributes. If the data distributions of two domains differ a lot, it would be easy to separate the persons of the two domains by observing only the domain biases. However, when these biases are not significant enough, the network is required to learn discriminative person-related features to make the decisions. Thus the single-task objective fits better to our setting and is chosen for this work.

Since pedestrian images are usually quite small and are not of square-shapes, it is not appropriate to directly use the ImageNet pretrained CNN models, which are trained with object images of high resolution and abundant details. Thus we propose to design a network structure that well fits our problem scale. Inspired by [16, 35], we build a CNN with three preceding 3×3 convolutional layers followed by six Inception modules and two fully connected layers. Detailed structures are listed in Table 1. The Batch Normalization (BN) layers are employed before each ReLU layer, which accelerate the convergence process and avoid manually tweaking the initialization of weights and biases. For training the CNN from scratch, we randomly dropout 50% neurons of the fc7 layer. The initial learning rate is set to 0.1 and is decreased by 4% for every 4 epochs until it reaches 0.0005. The learning rate is then fixed at this value for a few more epochs until convergence.

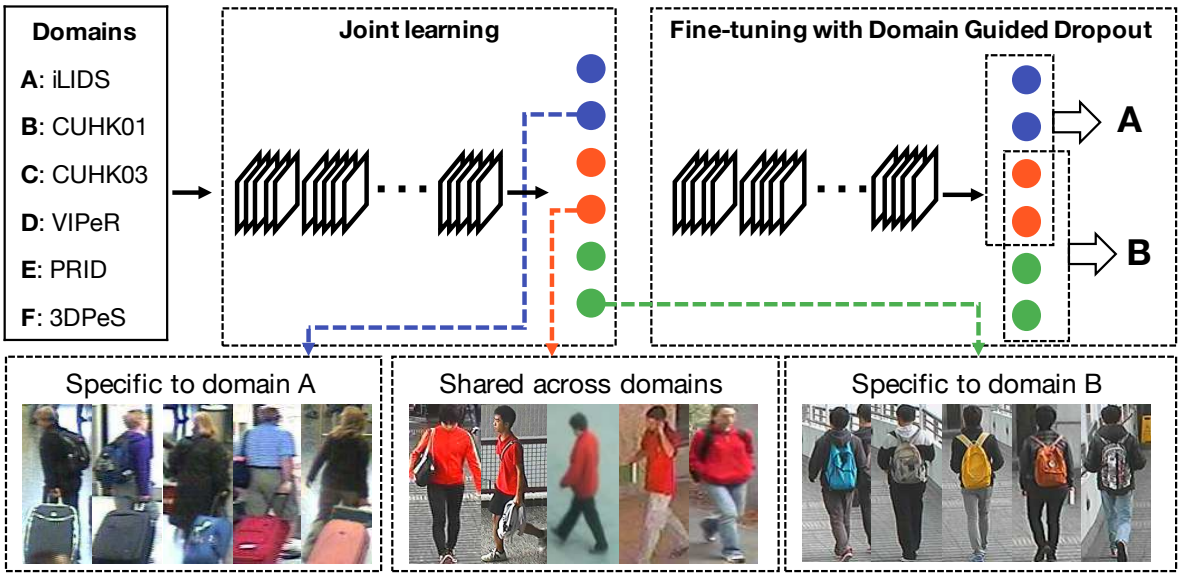

Figure 2. Overview of our pipeline. For the person re-identification problem, we first train a CNN jointly on all six domains. Then we analyze the effectiveness of each neuron on each domain. For example, some may capture the luggages that only appear in domain A, while some others may capture the red clothes shared across different domains. We propose a Domain Guided Dropout algorithm to discard useless neurons for each domain during the training process, which drives the CNN to learn better feature representations on all the domains simultaneously.

name	patch size/ stride	output size	#1×1	#3×3 reduce	#3×3	double #3×3 reduce	double #3×3	pool+proj
input		$3 \times 144 \times 56$						
conv1 – conv3	$3 \times 3/2$	$32 \times 144 \times 56$						
pool3	$2 \times 2/2$	$32 \times 72 \times 28$						
inception (4a)		$256 \times 72 \times 28$	32	32	32	32	32	avg + 32
inception (4b)	stride 2	$384 \times 72 \times 28$	32	32	32	32	32	max + pass through
inception (5a)		$512 \times 36 \times 14$	64	64	64	64	64	avg + 64
inception (5b)	stride 2	$768 \times 36 \times 14$	64	64	64	64	64	max + pass through
inception (6a)		$1024 \times 36 \times 14$	128	128	128	128	128	avg + 128
inception (6b)	stride 2	$1536 \times 36 \times 14$	128	128	128	128	128	max + pass through
fc7		256						
fc8		M						

Table 1. The structure of our proposed CNN for person re-identification

3.3. Domain Guided Dropout

Given the CNN model pretrained by using the mixed dataset, we identify for each domain which neurons are effective. For each domain sample, we define the impact of a particular neuron on this sample as the gain of the loss function when we remove the neuron. Specifically, let $g(x) \in \mathbb{R}^d$ denote the d-dimensional CNN feature vector of an image x. The impact score of the i-th ($i \in \{1, 2, \ldots, d\}$) neuron on this image sample is defined as

$$s_i = \mathcal{L}(g(x)_{\setminus i}) - \mathcal{L}(g(x)), \tag{3}$$

where $g(x)_{\setminus i}$ is the feature vector after we setting the i-th neuron response to zero. For each domain $\mathcal{D}$, we then take the expectation of s_i over all its samples to obtain the averaged impact score $\bar{s}_i = \mathrm{E}_{x \in \mathcal{D}}[s_i]$. We visualize the neuron impact scores between several pairs of domains in Figure 3. It clearly shows that the two sets of impact scores have little correlation, indicating that the effective neurons for different domains are not the same.

A naive computation of all the impact values requires $O(d|\mathcal{D}|)$ network forward passes, which is quite expensive if d is large. Therefore, we follow [34] to accelerate the process by using approximate Taylor's expansion of $\mathcal{L}(g(x))$ to

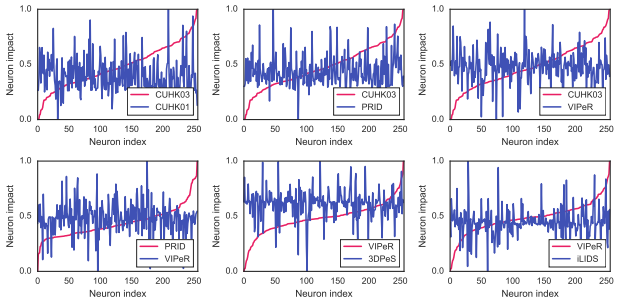

Figure 3. The neuron impact scores between several pairs of domains. For each pair of domains (A, B), the neurons are sorted w.r.t. their impact scores on domain A (red curves). Their impact scores on domain B are shown in blue. The two curves have little correlation, which indicates that different domains have different effective neurons.

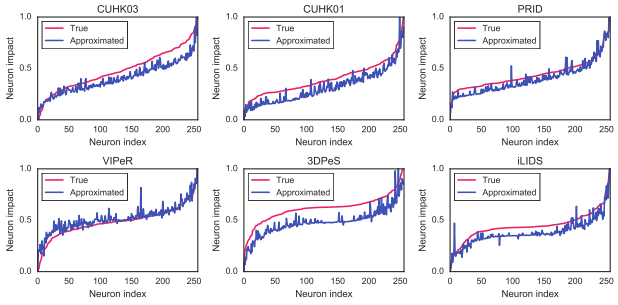

Figure 4. Comparison of the true (Eq. (3)) and the approximated (Eq. (4)) neuron impact scores

the second order

$$s_i \approx -\frac{\partial \mathcal{L}}{\partial g(x)_i} g(x)_i + \frac{1}{2} \frac{\partial^2 \mathcal{L}}{\partial g(x)_i^2} g(x)_i^2. \tag{4}$$

We study the quality of this approximation empirically, and observe that it is more accurate for higher-level layers close to the loss function. Here we show in Figure 4 the difference between the approximation and its true values for the neurons of the fc7 layer.

After obtaining all the $\bar{s}_i$, we continue to train the CNN model, but with these impact scores as guidance to dropout different neurons for different domains during the training process. For all the samples belonging to a particular domain, we generate a binary mask m for the neurons according to their impact scores s, and then elementwisely multiply m with the neuron responses. Two schemes are proposed on how to generate the mask m. The first one is deterministic, which discards all the neurons having non-positive impact scores

$$m_i = \begin{cases} 1 & \text{if } s_i > 0 \\ 0 & \text{if } s_i \leq 0 \end{cases} \tag{5}$$

The other one is stochastic, where m_i is drawn from a Bernoulli distribution with probability

$$p(m_i = 1) = \frac{1}{1 + e^{-s_i/T}}. \tag{6}$$

Here we use the sigmoid function to map a impact score to $(0, 1)$, and T is the temperature that controls how significantly the scores s would affect the probabilities. When $T \to 0$, it is equivalent to the deterministic scheme; when $T \to \infty$, it falls back to the standard Dropout with a ratio of 0.5. We study the effect of T empirically in Section 4.3.

We apply the Domain Guided Dropout to the fc7 neurons and resume the training process. The network's learning rate policy is changed to decay polynomially from 0.01 with the power parameter set to 0.5. The whole network is trained for 10 more epochs.

During the test stage, for the deterministic scheme, the neurons are also discarded if their impacts are no greater than zero. While for the stochastic scheme, we keep all the neuron responses but scale the i-th one with $1/(1+e^{-s_i/T})$.

4. Experiments

We conducted experiments on several popular person re-identification datasets. In this section, we first detail the characteristics of each dataset and the test protocols we followed in Section 4.1. Then we compare the results of our approach with state-of-the-arts, showing the effectiveness of our multi-domain deep learning pipeline in Section 4.2. Section 4.3 analyzes the Domain Guided Dropout module through a series of experiments, and discusses its properties based on the results. At last, we present some figures that help us understand the underlying mechanisms. The code is publicly available on GitHub[1].

4.1. Datasets and protocols

There exist many challenging person re-identification datasets. In our experiments, we chose seven of them to cover a wide range of domain varieties. CUHK03 [23] is one of the most largest published person re-identification datasets, it consists of five different pairs of camera views, and has more than 14,000 images of 1467 pedestrians. CUHK01 [21] is also captured on the same campus with CUHK03, but only has two camera views and 1552 images in total. PRID [15] extracts pedestrian images from recorded trajectory video frames. It has two camera views, each contains 385 and 749 identities, respectively. But only 200 of them appear in both views. Shinpuhkan [18] is another large-scale dataset with more than 22,000 images. The highlight of this dataset is that it contains only 24 individuals, but all of them are captured with 16 cameras, which provides rich information on intra-personal variations.

[1] https://github.com/Cysu/person_reid

Dataset	#ID	#Trn. images	#Val. images	#Prb. ID	#Gal. ID
CUHK03 [23]	1467	21012	5252	100	100
CUHK01 [21]	971	1552	388	485	485
PRID [15]	385	2997	749	100	649
VIPeR [13]	632	506	126	316	316
3DPeS [5]	193	420	104	96	96
i-LIDS [50]	119	194	48	60	60
Shinpuhkan [18]	24	18004	4500		

Table 2. Statistics of the datasets and evaluation protocols

The remaining three datasets are relatively quite small. VIPeR [13] is one of the most challenging dataset, since it has 632 people but with various poses, viewpoints, image resolutions, and lighting conditions. 3DPeS [5] has 193 identities but the number of images for each person is not fixed. iLIDS [50] captures 119 individuals by surveillance cameras in an airport, and thus consists of large occlusions due to luggages and other passengers.

Since Shinpuhkan dataset has only 24 people, it cannot be used for testing the performance of re-identification systems. Thus we only use it in the training phase. For the other datasets, we mainly follow the settings in [32] to generate the test probe and gallery sets. But our training set has two differences with theirs. First, both the manually cropped and automatically detected images in CUHK03 were used. Second, we sampled 10 images from the video frames of the training identities in PRID. We also randomly drew roughly 20% of all these images for validation. Notice that both the training and validation identities have no overlap with the test ones. The statistics of all the datasets and evaluation protocols are summarized in Table 2. In our experiments, we employed the commonly used CMC [30] top-1 accuracy to evaluate all the methods.

4.2. Comparison with state-of-the-art methods

We compare the results of our approach with those by state-of-the-art ones on all the six test datasets. For the 3DPeS and iLIDS datasets, the best previous method are [41] and [9], respectively. While for the other four datasets, the best results are reported by [32]. Both methods are built upon hand-crafted features, and exploit a ranking ensemble of kernel-based metrics to boost the performance. However, our method relies on the learned CNN features and uses the Euclidean distance directly as the metric, which stresses the quality of the learned features representation rather than the metrics.

In order to validate our approach, we first obtain a baseline by training the CNN individually on each domain. Then we merge all the domains jointly with a single-task learning objective (JSTL) and train the CNN from scratch using all these domains. Next, we improve the learned CNN with the proposed deterministic Domain Guided Dropout (JSTL+DGD). Notice that this step provides a single model working on all the domains simultaneously. To show our best possible results, we further fine-tune the CNN separately on each domain with the stochastic Domain Guided Dropout (FT-JSTL+DGD). We also adopt a baseline method by fine-tuning from the JSTL model on each domain with standard dropout (FT-JSTL) for comparison. The results are summarized in Table 3.

Method	CUHK03	CUHK01	PRID
Best	62.1 [32]	53.4 [32]	17.9 [32]
Individually	72.6	34.4	37.0
JSTL	72.0	62.1	59.0
JSTL+DGD	72.5	63.0	60.0
FT-JSTL	74.8	66.2	57.0
FT-JSTL+DGD	**75.3**	**66.6**	**64.0**
Method	VIPeR	3DPeS	iLIDS
Best	**45.9** [32]	54.2 [41]	52.1 [9]
Individually	12.3	31.1	27.5
JSTL	35.4	44.5	56.9
JSTL+DGD	37.7	45.6	59.6
FT-JSTL	37.7	54.0	61.1
FT-JSTL+DGD	38.6	**56.0**	**64.6**

Table 3. CMC top-1 accuracies of different methods

CNN structure. We first evaluate the effectiveness of the proposed CNN structure. When the network is trained only with the CUHK03 dataset, which is large enough for training CNN from scratch, we improve the state-of-the-art result by more than 10% to 72.6% (row 2 of Table 3). Compared with the previous best deep learning method [1], whose result is 54.7%, our method achieves a gain of 18% in the performance. A two-stream network is used in [1] to compute the verification loss given a pair of images, while we opt for learning a single CNN through an ID classification task and directly computing Euclidean distance based on the features. When the training set is large enough, this classification objective makes the CNN much easier to train. The CMC curves of different methods on the CUHK03 dataset are shown in Figure 5. However, when the dataset is quite small, it would be insufficient to learn such a large capacity network from scratch, which is demonstrated in Table 3 by the results of training the CNN only on each of the VIPeR, 3DPeS, and iLIDS datasets.

Joint learning. To overcome the scale issue of small datasets, we propose to merge all the datasets jointly as a single-task learning (JSTL) problem. In row three of Table 3, we can see the performance increase on most of the datasets. This indicates learning from multiple domains

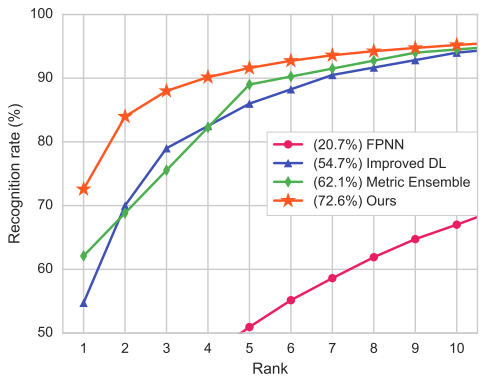

Figure 5. CMC curves of different methods on CUHK03 dataset

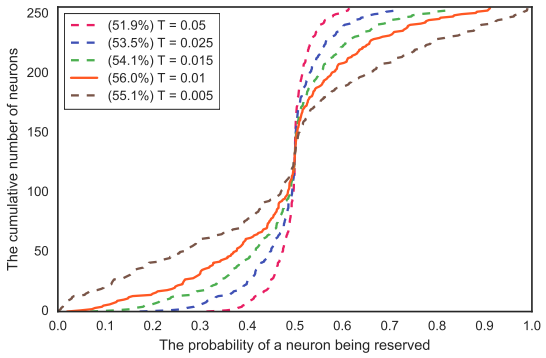

Figure 6. The cumulative number of neurons to be reserved under certain probabilities. Different temperature T settings and corresponding CMC top-1 accuracies are shown in the legend.

jointly is very effective to produce generic feature representations for all the domains. An interesting phenomenon is that the performance on CUHK03 decreases slightly. We hypothesize that when combining different datasets together without special treatment, the larger domains would leverage their information to help the learning on the others, which makes the features more robust on different datasets but less discriminative on the larger ones themselves. Note that we do not balance the data from multiple sources in a mini-batch, as it would give more weights on smaller datasets, which leads to severe overfitting.

Domain Guided Dropout. The fourth row of Table 3 shows the effectiveness of applying the proposed Domain Guided Dropout (DGD) to the JSTL scheme. Based on the JSTL pretrained model, we compute the neuron impact scores of the fc7 layer on different domains, replace the standard Dropout layer with the proposed deterministic Domain Guided Dropout layer, and continue to train the network for several epochs. Although the original JSTL model has already converged to a local minimum, utilizing Domain Guided Dropout consistently improves the performance on all the domains by 0.5%-2.7%. This indicates that it is effective to regularize the network specifically for different domains, which maximizes the discriminative power of the CNN on all the domains simultaneously.

At last, to achieve the best possible performance of our model on each domain, we fine-tune the previous JSTL+DGD model on each of them individually with stochastic Domain Guided Dropout. This step adapts the CNN to the specific domain biases and sacrifices the generalization ability to other domains. As a result, the final CMC top-1 accuracies are increased by several percents, as listed in the last row of Table 3. On the other hand, comparing with FT+JSTL, the results are improved by 3% on average, which indicates that JSTL+DGD provides better generic features. Note that FT+JSTL on PRID results in even worse performance than JSTL. Such overfitting problem is resolved by applying DGD.

4.3. Effectiveness of Domain Guided Dropout

After evaluating the overall performance of our pipeline, we also investigate in details the effects of the proposed Domain Guided Dropout module in this subsection.

Temperature T. As the temperature T significantly affects the behavior and performance of the stochastic Domain Guided Dropout scheme, we first study the effects of this hyperparameter. From the theoretical analysis we know that the stochastic Domain Guided Dropout falls back to the standard Dropout (ratio equals to 0.5) when $T \to \infty$, and to the deterministic scheme when $T \to 0$. However, it is still unclear how to set it properly in real applications. Therefore, we provide some empirical results of tuning the temperature T. We use the 3DPeS dataset as an example, and fine-tune the JSTL+DGD model on it with different values of T. For each temperature, all the fc7 neurons have certain probabilities to be reserved according to Eq (6). We count the histogram of the neurons with respect to their probabilities to be reserved, and plot the cumulative distribution function in Figure 6. We can see that the best performance can be achieved when T is in a certain range that makes $\max_i p(m_i = 1) \approx 0.9$. This phenomenon indicates that a good T should assign the most effective neuron a high enough probability (0.9) to be reserved. We set T according to this empirical observation when using the stochastic Domain Guided Dropout scheme in our experiments.

Deterministic vs. stochastic. The next question is whether the deterministic and stochastic Domain Guided Dropout have similar behaviors, or one outperforms the other in certain pipeline stages. We compare these two strategies within the JSTL+DGD and FT-JSTL+DGD stages in our pipeline. Their gains on the CMC performance for each domain under different settings are shown in Figure 7 as the blue and green bars, respectively.

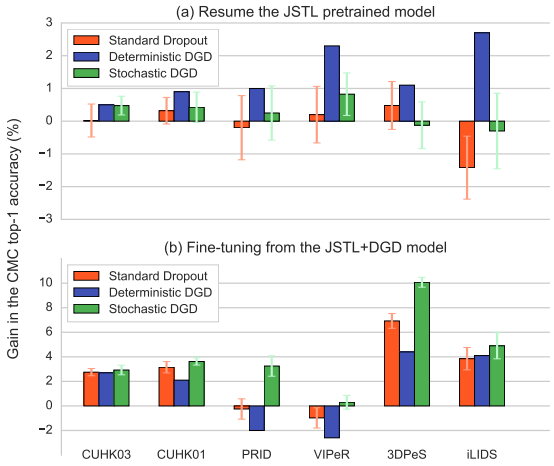

Figure 7. Comparison of different Dropout schemes

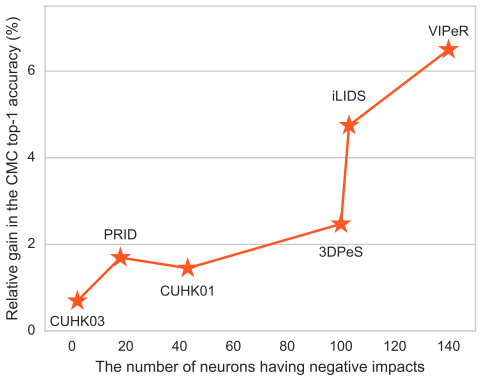

Figure 8. Relative performance gain with respect to the number of neurons having negative impact scores on specific domain in the deterministic Guided Dropout scheme

From Figure 7(a) we can see that when feeding the network with the data from all the domains, deterministic Domain Guided Dropout is better in general. This is because the objective here is to learn generic representations that are robust for different domains. The deterministic scheme strictly constrains that data from each domain are used to update only a specific subset of neurons. Thus it eliminates the potential confusion due to the discrepancies between different domains. On the contrary, when fine-tuning the CNN with the data only from one specific domain, the domain discrepancy no longer exists. All the inputs follow the same underlying distribution, so we can use stochastic Domain Guided Dropout to update all the neurons with proper guidance to determine the dropout rate for each of them, as shown in Figure 7(b). As a conclusion, the deterministic DGD is more effective when it is used to train the CNN jointly with all the domains, while the stochastic DGD is superior when fine-tuning the net separately on each domain.

Standard Dropout vs. Domain Guided Dropout. At last, we compare the proposed Domain Guided Dropout with the standard Dropout under different scenarios. The results are summarized in Figure 7. First, when resuming the training of the JSTL pretrained model, we applied the deterministic Domain Guided Dropout. From Figure 7(a) we can see that since the model is already converged, continue to use standard Dropout scheme cannot further improve the performance. The performance would rather jitter insignificantly or decrease on particular domains due to overfitting. However, by using the deterministic Domain Guided Dropout scheme, the performance improves consistently on all the domains, especially for the small-scale ones. On the other hand, by comparing the orange and the green bars in Figure 7(b), we can validate the effectiveness of the stochastic Domain Guided Dropout when fine-tuning the CNN model. This is because we utilize the domain information to regularize the network better, which keeps the CNN in the right track when training data is not enough.

We further investigate how does the deterministic Domain Guided Dropout change the network behavior by evaluating the relative performance gain on each domain with respect to the number of neurons having negative impact scores on that domain. As shown in Figure 8, smaller datasets tend to have more useless neurons to be dropped out, meanwhile the performance would be increased more significantly. This again indicates that we should not treat all the domains equally when using all their data, but rather regularize the CNN properly for each of them.

5. Conclusion

In this paper, we raise the question of learning generic and robust CNN feature representations from multiple domains. An effective pipeline is presented, and a Domain Guided Dropout algorithm is proposed to improve the feature learning process. We conduct extensive experiments on multiple person re-identification datasets to validate our method and investigate the internal mechanisms in details. Moreover, our results outperform state-of-the-art ones by large margin on most of the datasets, which demonstrates the effectiveness of the proposed method.

Acknowledgements

This work is partially supported by SenseTime Group Limited, the General Research Fund sponsored by the Research Grants Council of Hong Kong (Project Nos. CUHK14206114, CUHK14205615, CUHK417011, CUHK14207814, CUHK14203015), and National Natural Science Foundation of China (NSFC, NO.61371192).

References

[1] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *CVPR*, 2015. 2, 3, 6

[2] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand. Domain-adversarial neural networks. *arXiv:1412.4446*, 2014. 1

[3] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. *arXiv:1406.5774*, 2014. 1

[4] J. Ba and B. Frey. Adaptive dropout for training deep neural networks. In *NIPS*, 2013. 2

[5] D. Baltieri, R. Vezzani, and R. Cucchiara. 3dpes: 3d people dataset for surveillance and forensics. In *Proc. of the ACM workshop on Human gesture and behavior understanding*, 2011. 6

[6] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In *CVPR*, 2016. 2

[7] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *ICML*, 2007. 3

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[9] S. Ding, L. Lin, G. Wang, and H. Chao. Deep feature learning with relative distance comparison for person re-identification. *Pattern Recognition*, 2015. 2, 6

[10] 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. *arXiv:1310.1531*, 2013. 1

[11] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 2

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[13] D. Gray, S. Brennan, and H. Tao. Evaluating appearance models for recognition, reacquisition, and tracking. In *PETS*, 2007. 1, 6

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *NIPS*, 2012. 2

[15] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof. Person re-identification by descriptive and discriminative classification. In *SCIA*, 2011. 1, 5, 6

[16] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, 2015. 2, 3

[17] K. Kang, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. In *CVPR*, 2016. 2

[18] Y. Kawanishi, Y. Wu, M. Mukunoki, and M. Minoh. Shinpuhkan2014: A multi-camera pedestrian dataset for tracking people across multiple cameras. In *20th Korea-Japan Joint Workshop on Frontiers of Computer Vision*, 2014. 5, 6

[19] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, 2012. 3

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[21] W. Li and X. Wang. Locally aligned feature transforms across views. In *CVPR*, 2013. 1, 5, 6

[22] W. Li, Y. Wu, M. Mukunoki, Y. Kuang, and M. Minoh. Locality based discriminative measure for multiple-shot human re-identification. *Neurocomputing*, 2015. 2

[23] W. Li, R. Zhao, T. Xiao, and X. Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014. 1, 2, 3, 5, 6

[24] X. Li, W.-S. Zheng, X. Wang, T. Xiang, and S. Gong. Multi-scale learning for low-resolution person re-identification. In *ICCV*, 2015. 2

[25] S. Liao, Z. Mo, Y. Hu, and S. Z. Li. Open-set person re-identification. *arXiv:1408.0872*, 2014. 3

[26] C. Liu, S. Gong, and C. C. Loy. On-the-fly feature importance mining for person re-identification. *Pattern Recognition*, 2014. 2

[27] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv:1411.4038*, 2014. 2

[28] M. Long and J. Wang. Learning transferable features with deep adaptation networks. *arXiv:1502.02791*, 2015. 2

[29] B. McFee and G. R. Lanckriet. Metric learning to rank. In *ICML*, 2010. 3

[30] H. Moon and P. J. Phillips. Computational and performance aspects of pca-based face-recognition algorithms. *Perception*, 2001. 6

[31] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014. 1

[32] S. Paisitkriangkrai, C. Shen, and A. v. d. Hengel. Learning to rank in person re-identification with metric ensembles. *arXiv:1503.01543*, 2015. 3, 6

[33] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015. 3

[34] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR*, 2014. 4

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 3

[36] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, 2014. 2, 3

[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[38] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, 2015. 2

[39] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu. Shape and appearance context modeling. In *ICCV*, 2007. 2

[40] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2005. 3

[41] F. Xiong, M. Gou, O. Camps, and M. Sznaier. Person re-identification using kernel-based metric learning methods. In *ECCV*, 2014. 3, 6

[42] Y. Xu, L. Lin, W.-S. Zheng, and X. Liu. Human re-identification by matching compositional template with cluster sampling. In *ICCV*, 2013. 2

[43] Y. Xu, B. Ma, R. Huang, and L. Lin. Person search in a scene by jointly modeling people commonness and person uniqueness. In *ACM Multimedia*, 2014. 3

[44] W. Yang, W. Ouyang, H. Li, and X. Wang. End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation. In *CVPR*, 2016. 2

[45] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *TIP*, 2015. 2

[46] R. Zhao, W. Ouyang, and X. Wang. Person re-identification by salience matching. In *ICCV*, 2013. 2

[47] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *CVPR*, 2013. 2

[48] R. Zhao, W. Ouyang, and X. Wang. Learning mid-level filters for person re-identification. In *CVPR*, 2014. 2

[49] L. Zheng, L. Shen, L. Tian, S. Wang, J. Bu, and Q. Tian. Person re-identification meets image search. *arXiv:1502.02171*, 2015. 3

[50] W.-S. Zheng, S. Gong, and T. Xiang. Associating groups of people. In *BMVC*, 2009. 6