Cross domain Residual Transfer Learning for Person Re-identification

Furqan Khan, Francois Bremond

Cross domain Residual Transfer Learning for Person Re-identification

Furqan M. Khan Francois Bremond
INRIA Sophia Antipolis - Mediterranee
{furqan.khan|francois.bremond}@inria.fr

Abstract

This paper presents a novel way to transfer model weights from one domain to another using residual learning framework instead of direct fine-tuning. It also argues for ***hybrid*** *models that use learned (deep) features and statistical metric learning for multi-shot person re-identification when training sets are small. This is in contrast to popular end-to-end neural network based models or models that use hand-crafted features with adaptive matching models (neural nets or statistical metrics). Our experiments demonstrate that a hybrid model with residual transfer learning can yield significantly better re-identification performance than an end-to-end model when training set is small. On iLIDS-VID [42] and PRID [15] datasets, we achieve rank-1 recognition rates of* 89.8% *and* 95%*, respectively, which is a significant improvement over state-of-the-art.*

1. Introduction

Person re-identification (re-ID) refers to the retrieval task where the goal is to search for a given person (query) in disjoint camera views (gallery). Performance of appearance based person re-ID methods depend on the similarity metric and the feature descriptor used to build a person's appearance model from given image(s). Classical approaches [12, 9, 30, 55, 25, 32] use hand-crafted features with statistical metric learning [25, 44, 58, 34, 20, 35, 27, 38] to build re-ID models. The learned metric can be viewed as feature transformation that brings features of same person from different cameras closer under Euclidean distance. Recently, neural networks are used to combine appearance description and metric learning in end-to-end re-ID models. Conceptually, lower order layers in the network learn features which are often shared by different tasks. However, as we go deeper in the hierarchy, the layers focus on learning higher order task specific abstractions. In case of person re-ID, this higher order abstraction is equivalent of statistical metric learning. As these layers are closer to the objective function, they are more prone to overfitting a particular dataset than the lower order layers. Therefore, we have found that for small training sets, it is advantageous to use a *hybrid* approach, *i.e.* replace higher order abstraction layers (corresponding to a metric) with statistical metric learning, instead of learning an end-to-end deep re-ID model. This significantly reduces the number of model parameters. However, trivially dropping n higher order layers from the network before training (or fine-tuning) does not break conceptual division of low and high level abstraction layers in a network as the roles of layers are re-configured to obtain the balance between low-level and high-level features to accomplish desired task. In fact, dropping a number of layers decreases representational capacity of the model. This paper describes a framework to define and learn hybrid models for multi-shot person re-ID from small training sets.

Whereas collection of large re-ID datasets for training remains infeasible for practical applications, the impact of advances in deep learning on video-based re-ID systems stays limited as shown by their inferior performance in comparison to classical approaches on a number of public benchmarks (see supplemental material with [17] for exhaustive comparison). It is important to point out that unlike other tasks, such as object classification, where collection of large amount of data is considered a necessary evil, re-ID models are often trained for specific scenarios and camera setups. Therefore, in practical surveillance networks changes in the camera network are imminent due to camera failures, reconfiguration, *etc.*. Therefore, learned models need to be update frequently. Requiring recurrent data collection in large quantity is a significant limitation.

To address limited availability of data for training, [57, 11] transfer models learned using Imagenet [8] to target re-ID datasets, whereas [47] proposes domain guided dropout. We also take advantage of transfer learning; however, instead of directly fine-tuning parameters for the target domain, we take advantage of residual learning framework [13] to adapt models for desired task. Specifically, we judiciously add *bottleneck residual units* proposed in [13] to the model being transferred and selectively optimize the network in 4 stages with small amount of training data. We call this *Residual Transfer Learning* (RTL). In comparison to direct parameter fine-tuning, additional residual

units provide flexibility and increase model capacity. For instance, the residual units can have different parameters and configuration than original layers, thus increasing or decreasing trainable parameters (see Section 3 for further discussion). We use the adapted network to extract features to perform person re-ID. Our results show that for small datasets iLIDS-VID [42] and PRID [15], RTL gives better overall performance than direct fine-tuning. For large scale dataset, MARS [57], the performance approaches to that of direct parameter fine-tuning but requires less training data.

Another take away from this paper is the argument for use of hybrid re-ID models instead of end-to-end neural network models. For small datasets, this improves state-of-the-art from 79% to 89.8% in terms of rank-1 rate on iLIDS-VID and from 92.5% to 95.2% on PRID. This is also an improvement over baseline end-to-end models yielding 62.9% on iLIDS-VID and 88.1% on PRID. On much larger MARS dataset, the hybrid model performs similarly as the end-to-end baseline but require less training data.

2. Related Work

Based on number of images available to learn a signature, re-ID is categorized as either single-shot or multi-shot. Availability of only one image, makes single-shot re-ID extremely challenging to extract robust appearance information. A significant amount of literature is available that hand-crafts image features to extract appearance information [12, 9, 30, 55, 25, 32, 40, 3, 5, 26, 31, 2]. These methods have proven useful to varying degrees, however, overall their performance using a static similarity metric, such as Euclidean distance, has been generally quite low.

Consequently, over years the research has focused on improving similarity metrics by learning them using supervised techniques [25, 44, 58, 34, 20, 35, 27, 38, 7]. The goal of these methods is to learn a transformation of features so that distance between similar persons can be minimized while distance between different persons is maximized. Other ways to improve signature matching rely on learning discriminative dictionaries [18] or treating signature matching as sparse representation problem [52, 46]. However, the overall performance of matching is highly correlated with the quality of features used. More robust and discrminative signatures lead to higher performance.

Availability of multiple images per person provides an opportunity to build better signature models. Most approaches [9, 3, 23, 33, 45, 6, 54, 51, 19, 49, 60] treat images individually and use descriptor statistics (mean or max) or set based representation to build models, [43, 29] consider temporally adjacent images to form a spatio-temporal volume. However, how to best aggregate information over multiple images remains an open challenge as indicated by recent work [33, 45, 6, 19, 49, 60, 24, 50].

Given the success of neural networks in other fields of computer vision, recent work on person re-ID also involve using neural networks [47, 33, 45, 51, 49, 60, 56, 1, 36, 41, 28]. However, small sized training sets make bigger and powerful networks difficult to train. Since collecting large-scale re-ID datasets, such as MARS [57], will continue to remain infeasible for practical applications, strategies need to be developed to leverage success of deep learning on small re-ID datasets. Thus, unlike most earlier work on re-ID using neural networks, which focuses on large datasets, this paper focuses on dealing with challenges of adopting bigger neural networks for small training sets.

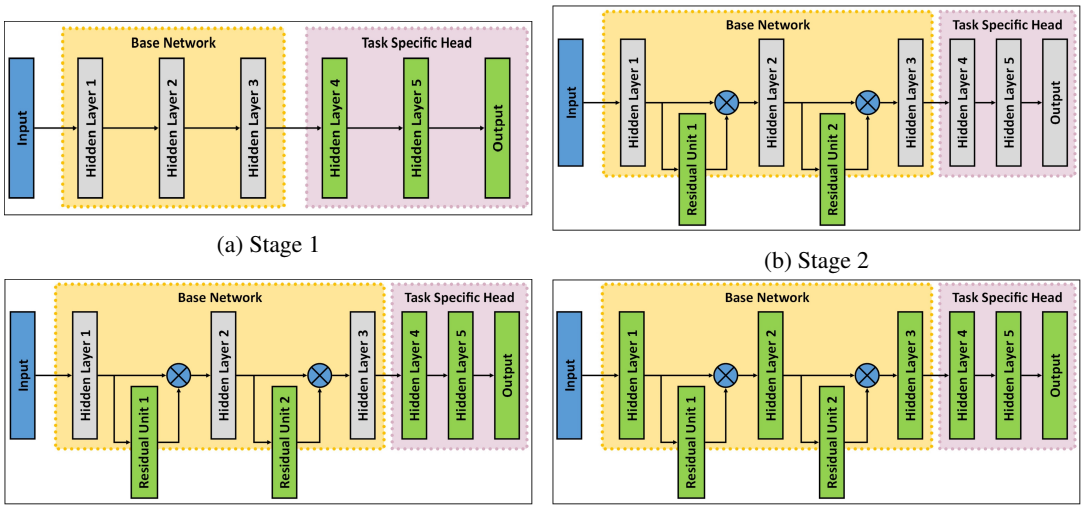
Similarly, Net2Net [4] focuses on *growing* a network's capacity via re-parameterization of the same function and initialization using *teacher* network to consume additional data/insights. Both "student" and "teacher" networks are however trained on the "same task". It also does not address the over-fitting issue faced in re-ID due to small train sets. Conversely, our emphasis is on better generalization with small train sets and network growth is only the means to the end. Thus, it addresses different problems at high level.

3. Residual Transfer Learning

One of the many reasons for the recent popularity of deep learning is the ease with which a model trained for a task/dataset can be used for another task/dataset. The most common strategy to transfer learning in neural networks is to initialize the network to be trained for the desired task with parameters that are learned for another task and then *fine-tune* the network parameters. Here, we present an alternative strategy for transfer learning using the concept of residual learning for ResNet by He *et.al.* [13].

In ResNet, each layer of the network estimates the residual between the input and the output signals. We pose transfer learning as a residual learning problem because the objective is to minimize the residual between the output of a pre-trained network and the desired output. This can be achieved by adding residual units for a number of layers to an existing model that needs to be transferred from one task to another. Each residual unit estimates the difference between the output of the layer and the output desired for the new task. An existing network can thus be made to perform a different task by adding and optimizing residual units.

One advantage of using residual learning for model transfer is that it allows more flexibility in terms of modeling the difference between two tasks through a number of residual units and their composition. One can add only one residual layer near the end of the network or in the beginning. The former may be suitable when the tasks are similar and we expect earlier layers to generate low error that can be compensated by the lone residual layer. On the other hand, the errors accumulate as we go deep in the network. Thus mitigating errors early may obviate the need to have residual layers deeper in the network. We find it more appealing



(a) Stage 1

(b) Stage 2

(c) Stage 3

(d) Stage 4 (optional)

Figure 1: Residual Transfer Learning in 4 stages. During each stage only the selected layers (shown in green) are trained. Residual Units are added to the network after first stage of RTL.

to add residual units for a number of layers in the middle part of the network to distribute the burden of error compensation on a specific layer. This way, each residual unit can be made simpler (have fewer parameters) in comparison to the original layer it is associated with. One such choice of residual units could be the residual units proposed in [13]. Having fewer parameters in the network makes it easier to get convergence and computationally faster to train.

3.1. Training

One challenge with addition of residual units to the network is their initialization. Ideally, the residual units should approximate the identity function in the beginning. However, as we intend to make residual units simpler and have fewer parameters, it is non-trivial to initialize parameters to approximate the identity function. The problem can be avoided by choosing a low enough learning rate when residual units are the only addition to the network. However, generally when a network is transferred from one task to another, the *head* - a number of layers towards the end of the network, such as a classification layer - is replaced with *task specific head*. The number of layers in the head depend on the high level difference between the tasks as higher order layers are more specialized for a particular task. Thus, residual units are not the only source of error in the output of the network. Therefore, we noted that when residual units are added to the network with a different network head, training loss is significantly higher in the beginning which pushes the network far away from pre-trained solution by trying to over compensate through residual units. To avoid this, we propose to train the network in 4 stages, with fourth stage being optional (Fig. 1).

Stage 1: In the first stage, we replace original head of the network with a task specific head and initialize it randomly. At this stage, we do not add any residual units to the network and train only the parameters of the replaced head of the network. Thus only the head layers are considered to contribute to the loss. This allows the network to learn noisy high level representation for the desired task and decrease the network loss without affecting lower order layers.

Stage 2: In the second stage, we add residual units to the network and initialize them randomly. Then we freeze all other layers, including in network head, and optimize the parameters of added residual units. As the head and other layers are fixed, residual units are considered as the source of loss. As we start with a reasonably low loss value, residual units are not forced to over compensate for the loss.

Stage 3: In the third stage, we train the network by learning parameters of both added residual units and network head. Thus allowing both the lower and higher order representations to adjust to the specific task.

Stage 4 (Optional): We noticed in our experiments on different datasets that the loss function generally gets low enough by the end of third stage. However, if needed, the whole network is trained to further improve performance.

4. RTL for Person Re-Identification

We use RTL to transfer a model trained on Imagenet [8] for object classification to perform person re-ID. We considered different popular network choices such as AlexNet [21], ResNet [13] and VGG [37]. We chose to use 16-layer VGG model due to its superior performance in comparison to AlexNet and overlooked ResNet for its extreme depth because our target datasets are small and do not warrant such a deep model for higher performance.

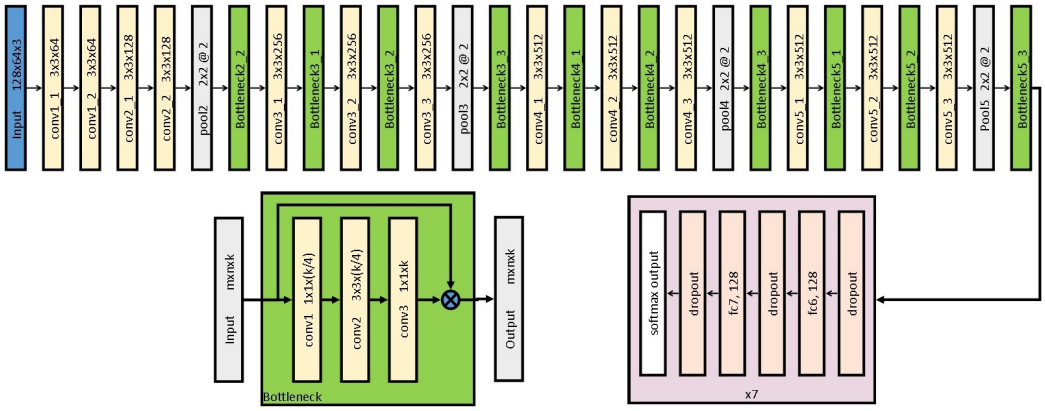


Figure 2: Final network architecture used for person re-ID. 10 new bottleneck layers are added to 16-layer VGG model. For each layer, the name of the box indicates the type of operation applied and its parameters, except for input and output, for which dimensions of data is shown. For convolutional layers, parameters are size of kernel $\times$ number of filters. For pooling layers, max-pool is used with kernel size of 2×2 @ stride of 2. Each bottleneck layer implement a residual unit with 3 convolutional layers and a shortcut. ReLU is applied to output of each layer, including bottleneck. Output of $Bottleneck5_3$ is sliced into 7 parts to learn 7 part models simultaneously without parameter sharing.

4.1. Task specific base model modifications

16-layer VGG model consists of 5 groups of convolutional layers (*conv1-conv5*) followed by two fully connected layers (*fc6, fc7*) and a classification layer. Output of each convolutional group is passed through a max-pool layer (*pool1-pool5*) before input to the next group. We transfer the model trained for 1000-way object classification on Imagenet with a receptive field of 224×224 pixels. Considering that object classification and person re-ID are significantly different tasks at high-level we remove fully connected layers *fc6* and *fc7* along with the output layer. The also allows us to change the receptive field of the network to a more suitable 128×64 (height $\times$ width) for person re-ID. Due to a smaller receptive field, we removed the first max-pool layer (*pool1*) after *conv1* group. This produces 512 filter maps of 7×3 at the output of max-pool layer *pool5*.

4.2. Task specific head

Neural networks for person re-ID can be trained for Identity Discriminative Embedding (IDE) [57], as a Siamese Network [33], or using a Triple Loss [14]. In practice, IDE often performs well even though Triplet loss allows improved performance. We train our network for IDE due to its popularity in the person re-ID community, which allows for wider comparison with published results.

When training a network for IDE, the objective is to predict the identity of a person by learning a *n*-way classifier. At test time, appearance model of a person is built using features extracted by the *embedding* layer. We learn multiple embedding models for a person, one for each local region or part. We achieve this by splitting the output of *pool5* layer in 7 ways, one for each horizontal stripe of the feature map. For each part feature map, we use two fully connected layers of size 128 (*partx/fc6,partx/fc7*), interleaved with dropout layers, before using *n*-way classification layer. We train all the part models simultaneously by summing the loss for each part. This way for each part a shared low-level feature representation (*conv* layers) and a part specific representation (*fc* layers) are learned. Even though local modeling of appearance is common in literature, local models are often learned independently. Simultaneous training of part models makes better use of limited amount of data by enforcing latent representation to be generic across parts.

4.3. Residual units and their placement

As we go deeper down the neural network hierarchy, the layers learn more abstract and task specific features. We expect lower convolutional groups, *conv1* and *conv2*, to consist of generic feature detectors. Consequently, we chose to surround each layer in groups 3 through 5 with residual units. When a convolutional layer is followed by a max-pool layer, residual unit is placed after the max-pool layer instead of after the convolutional layer.

We define each residual unit similar to [13]. Specifically, each residual unit consists of 3 convolutional layers. First layer uses $1/4$ times the number of filters of the input layer and 1×1 kernel with stride of 1. Second layer uses the same number of filters as the first layer but uses 3×3 kernel and stride of 1. Final layer again uses 1×1 kernel and stride of 1 but has the same number of filters as the input to the residual unit. We apply ReLU to the output of each layer in the residual unit. Finally, the output of the residual unit is added with the input and ReLU is applied. Our final network architecture is summarized in Fig. 2.

5. Appearance Model for Person Re-ID

Given the network trained using RTL, we extract features for each image of a person from layers *partx/fc7*, *pool5* and sub-sampled *pool4*, to build corresponding appearance models RTL_{fc7}, RTL_{pool5}, and RTL_{pool4}. More precisely for RTL_{pool4}, we pass the output of *pool4* through another max-pool layer with same parameters, giving us the same sized feature maps as *pool5*. Each model RTL_f, $f \in \{fc7, pool5, pool4\}$ consists of 8 sub-models: one for each of the 7 parts and another representing full body using their concatenation. For $f \in \{pool5, pool4\}$, the parts correspond to 7 horizontal stripes of respective feature maps.

Multiple images per person allow numerous ways to aggregate appearance information of a person. For example, [19] uses GMMs, [50] and [33] use RNNs, and [29] uses motion based spatio-temporal pooling. However, to emphasize on the quality of learned features and minimize impact of aggregation mechanism, we use average pooling of image features for iLIDS-VID and PRID, and max pooling for MARS. Difference in pooling between small and large datasets is based on the results of [57]. We normalize descriptors to have unit norm following the pooling operation.

To perform ranking, we compute similarity between appearance models of persons p and q, by summing the similarity scores between all of their corresponding sub-models.

$$S(p,q) = \sum_{i=1:8} s_{\mathcal{M}}(x_i^p, x_i^q) \tag{1}$$

where x_i^p, x_i^q are the i^{th} sub-models of $x \in \{RTL_{fc7}, RTL_{pool5}, RTL_{pool4}\}$ for persons p and q, and $\mathcal{M}$ is the metric. For experiments, we use two variants of similarity functions. A simple version uses:

$$s_{\mathcal{M}}(x_i^p, x_i^q) = -d_{\mathcal{M}}(x_i^p, x_i^q) \tag{2}$$

where $d_{\mathcal{M}}$ is the distance under metric $\mathcal{M}$. Another version uses gallery based exponential kernel proposed by [19] to normalize the similarity between each model in $(0, 1)$ range using scores for the query set given a gallery item $p \in \boldsymbol{G}$.

$$s_{\mathcal{M}}(x_i^p, x_i^q) = \exp\left(-3\frac{d_{\mathcal{M}}(x_i^p, x_i^q) - \min_{q\in\boldsymbol{Q}} d_{\mathcal{M}}(x_i^p, x_i^q)}{\max_{q\in Q} d_{\mathcal{M}}(x_i^p, x_i^q) - \min_{q\in\boldsymbol{Q}} d_{\mathcal{M}}(x_i^p, x_i^q)}\right) \tag{3}$$

where, $\boldsymbol{G}$ and $\boldsymbol{Q}$ are gallery and query sets. The constant 3 spreads the similarity scores between $(0.05, 1)$ range.

6. Evaluation

As single camera pedestrian detection and tracking has become robust over recent years, we focused on evaluation on three popular multi-shot re-ID datasets: iLIDS-VID [42], PRID [15] and MARS [57]. iLIDS-VID contain 300 persons recorded from 2 cameras. Each person has only one track per camera. PRID has a total of 749 persons, however, only 200 of them have been recorded from two cameras with at most one track per camera. To be consistent with earlier work, we used only 178 persons appearing in both cameras with at least 21 frames. We follow common evaluation practice of [42, 38, 17, 57, 51] to use half of the persons from both cameras for training and half for test, and report results for 10 random trials. Both iLIDS-VID and PRID are considerably small for deep learning. Contrarily, MARS has more than $20K$ tracks of 1261 persons from 6 cameras. Following experimental setup of [57], we use 625 people for training and the remaining for test.

6.1. Hyper-parameters

We tuned hyper parameters, such as learning rate schedule and placement of residual units based on performance on iLIDS-VID. The parameters were then fixed for other datasets and experiments. We used stochastic gradient descent for training with momentum of 0.9 and batch size of MK, where $M = 64$ persons and $K = 2$ images per person were used based on GPU memory size. In each batch, we emphasized on finding images from different cameras for each person, unless the person appeared only in one camera. The order of persons was shuffled after each person was selected once. The learning rate for training stages 1 and 2 was set as 0.1. It is then lowered by an order of magnitude (0.01 and 0.001) for each subsequent stage.

The number of iterations per stage depended on whether we were training for large (MARS) or small dataset (iLIDS-VID or PRID). Let N be the minimum number of batches needed to process an epoch (entire number of images in training set). Average number of training images per trial for PRID and iLIDS-VID are $\sim 25K$ and for MARS is $\sim 500K$. Thus given the batch size, we needed $N = 200$ iterations for PRID and iLIDS-VID, and $N = 4000$ iterations for MARS to process one epoch. Given N, we train the network for $\{10N, 10N, 20N, 20N\}$ iterations during first through fourth stage on small dataset and $\{N, N, 2N, 2N\}$ iterations correspondingly on large dataset. Finally, note that since we randomly sample data for each person, it is not guaranteed that all images would be used for training during an epoch, so in effect we train for as many images in the training set and not all the images during an epoch.

6.2. Effectiveness of hybrid modeling

Performing re-ID using appearance model RTL_{fc7} with Euclidean distance corresponds to using an end-to-end re-ID model with integrated metric layers. The conceptual division of the network into feature and metric layers is subjective. We consider two conceptual divisions: First, we consider layers up to *pool5* as feature layers and the rest as metric layers. Second, we divide the network at layer *pool4*. In each case, we replaced higher layers with XQDA [25]

	iLIDS-VID			PRID			MARS			
Model	r=1	r=5	r=10	r=1	r=5	r=10	r=1	r=5	r=10	mAP
$B_7 \equiv RTL_{fc7} + Eucl$	54.0	77.2	84.6	83.1	96.3	98.4	66.9	82.7	86.8	48.2
$H_7 \equiv RTL_{fc7} + XQDA$	56.4	81.7	89.0	82.7	96.0	97.8	67.9	83.4	87.3	49.6
$B_5 \equiv RTL_{pool5} + Eucl$	51.2	75.1	82.7	79.8	94.6	98.2	65.5	80.2	84.2	42.8
$H_5 \equiv RTL_{pool5} + XQDA$	57.8	82.1	89.1	82.4	95.2	98.0	66.3	82.1	86.7	46.6
$B_4 \equiv RTL_{pool4} + Eucl$	46.0	71.1	78.0	79.7	92.4	95.8	41.4	56.1	62.0	18.3
$H_4 \equiv RTL_{pool4} + XQDA$	78.7	94.0	96.9	92.1	98.1	98.8	58.3	71.3	76.5	34.8

Table 1: Performance comparison of hybrid models with end-to-end and respective baselines with Euclidean distance using recognition rates in percent at ranks $r \in \{1, 5, 10\}$ for different datasets. mAP in case of MARS is also reported.

	iLIDS-VID						PRID						MARS					
Stage	B_7	B_5	B_4	H_7	H_5	H_4	B_7	B_5	B_4	H_7	H_5	H_4	B_7	B_5	B_4	H_7	H_5	H_4
1	34	18	32	46	42	69	75	55	63	74	62	82	35	21	19	42	27	30
2	48	41	42	55	56	76	83	73	75	83	79	91	54	49	32	57	56	49
3	53	50	45	57	58	77	85	77	77	83	82	92	66	64	39	66	65	55
4	54	51	46	56	58	79	83	80	79	83	82	92	67	65	41	68	66	58

Table 2: Performance of re-ID models after each stage of RTL as measured by rank-1 rate in percent.

to get hybrid models $H_5 \equiv RTL_{pool5} + XQDA$ and $H_4 \equiv RTL_{pool4} + XQDA$. We compare hybrid models with their respective baselines using Euclidean distance and RTL_{fc7} with and without XQDA in Table 1. Results are for 4 stage training **without** using exponential similarity kernel.

Table 1 shows that there is progression in performance of base models (B_4, B_5, B_7) as we go deeper in the network to build appearance models. However, the performance of hybrid models (H_4, H_5, H_7) regresses for small datasets as the number of metric layers decreases. The hybrid model H_4 performs significantly better than the baseline model B_7 for both small datasets. The performance gap is enhanced on difficult iLIDS-VID which requires better generalization of the model due to higher occlusions. The obvious explanation is that the number of parameters in XQDA are many orders smaller than the deep metric model represented by higher layers in the network. For scarce amount of data, XQDA is able to generalize better. It is interesting to note that H_5 performs similar to B_7 on both dataset sizes.

6.3. Performance at each stage of RTL

To study the improvement brought by each stage of RTL, we evaluated models at the completion of each stage. Table 2 shows rank-1 rates of different models textbfwithout using exponential kernel. Since, hidden layers are not trained during stage 1, performance of RTL_{pool4} and RTL_{pool5} for stage 1 is the baseline performance of these models. Note that, across feature models, the major improvement in performance comes during the second stage of training when added residual units are trained. Whereas on larger dataset, MARS, both stage 2 and stage 3 contribute significantly in improving performance. As expected, stage 4 does not further improve performance by much.

6.4. Effect of training set size

We trained multiple models by randomly selecting 100, 200, 300, 400, and 500 persons from MARS and test them on entire test set of 636 persons. Average performance of models across 5 random trials is reported in Table 3. Results show that both H_5 and H_4 perform considerably better than B_7 and H_7 when only 100 persons are used for training and marginally better when 200 persons are used. On the other side, performance of H_5 is similar to B_7 and H_7 over larger train sets. A hybrid model provides reliable results in both situations, whereas an end-to-end model does not.

Note that XQDA is not effective with RTL_{fc7} irrespective of training set size. This is because low training error is achieved by over-fitting, making XQDA redundant. Higher layers suffer more from over-fitting than the lower layers. This is evident from higher rank-1 rate of H_4 over H_5 for small train sets. Also, for 100 person train set, both H_5 and H_7 are marginally poor than B_5 and B_7 respectively.

6.5. Comparison of features

Performance of multi-shot re-ID methods depends on image features, aggregation method and similarity metric. To compare different features, we aggregate features of a person using mean-pool operation over image set, hence isolating the effect of representation. In addition to popular hand-crafted features, we also compare other IDE features in conjunction with various network architectures. The other IDE models use traditional fine-tuning based model transfer from Imagenet to target set. Thus the difference in performance of IDE-VGG [17] and RTL_{fc7} can be attributed to the difference in model transfer technique. A summary of results is presented in Table 4 and Table 5.

Under Euclidean distance, RTL gives the best perfor-

	rank-1						mAP					
Model↓ / T→	100	200	300	400	500	625	100	200	300	400	500	625
B_7	43.6	51.7	60.5	66.0	67.2	67.0	23.6	30.9	39.5	44.9	46.7	48.2
B_5	49.3	53.3	60.5	64.2	64.3	65.5	25.4	30.2	36.2	39.8	40.9	42.8
B_4	38.0	40.2	43.0	43.0	42.8	41.4	15.1	17.0	18.4	18.3	18.4	18.3
H_7	42.7	50.6	60.7	66.0	67.7	67.9	23.2	30.6	39.6	45.7	47.6	49.6
H_5	48.3	52.8	60.5	65.2	65.6	66.3	25.7	30.9	38.4	43.4	44.6	46.6
H_4	49.5	53.9	56.8	57.6	58.6	58.3	25.9	30.5	32.2	32.9	34.3	34.8

Table 3: Performance of learned model on MARS test set with respect to size of training set. For each re-ID model, average of rank-1 rates and mAP over 5 random trials are reported for training set size $T \in \{100, 200, 300, 400, 500, 625\}$.

	iLIDS-VID		PRID2011	
Feature	Eucl.	XQDA	Eucl.	XQDA
HistLBP[48]	3.1	27.7	10.1	36.3
LDFV[30]	5.5	36.9	19.0	50.2
gBiCov[31]	5.1	2.8	7.1	4.4
SDC[55]	8.3	26.2	13.6	44.6
LOMO[25]	6.8	49.6	44.5	80.1
WHOS[26]	12.9	47.9	31.1	65.2
GOG[32]	23.2	63.5	60.3	84.6
IDE-CaffeNet[57]	43.6	53.0	61.1	72.8
IDE-VGG[17]	37.4	53.5	75.7	83.7
IDE-ResNet[17]	42.0	61.0	74.7	84.4
RTL_{fc7}	**54.0**	56.4	**83.1**	82.7
RTL_{pool5}	51.2	57.8	79.8	82.4
RTL_{pool4}	46.0	**78.7**	78.7	**92.1**

Table 4: Performance comparison of appearance descriptors using rank-1 recognition rate. For multiple images of a person, average of image-wise descriptors is used for all methods. Best results in each column are colored in red.

mance for both iLIDS-VID and PRID datasets. This indicates that using RTL for model transfer is better than direct fine-tuning of network parameters. We, however, noticed that the gain obtained by using XQDA for IDE-VGG is relatively higher than for RTL_{fc7}, nevertheless final performance of RTL_{fc7} is better on iLIDS-VID by 3% and lower on PRID by 1%. However, both these models perform poorly in comparison to hand-crafted GOG [32] feature when using XQDA. This is due to difficulty in training deeper models with small data. In comparison, RTL_{pool4} + XQDA performs 15% and 8% points better than top performing GOG on iLIDS-VID and PRID, respectively.

On MARS, RTL leads to better performance under Euclidean distance than popular hand crafted features and IDE models based on CaffeNet and ResNet by at least 5% points. Given that ResNet is a considerably deeper model than VGG, we claim that the superior performance based on VGG is a result of improved training procedure. Our model, however, gives 2.6% points lower performance than IDE-ResNet when combined with XQDA.

	MARS	
Feature	Eucl.	XQDA
HistLBP[48]	3.1	18.6
gBiCov[31]	5.1	9.2
LOMO[25]	6.8	30.7
IDE-CaffeNet[57]	60.0	65.3
IDE-ResNet[59]	62.7	**70.5**
RTL_{fc7}	**67.0**	67.9
RTL_{pool5}	65.5	66.3
RTL_{pool4}	41.4	58.3

Table 5: Performance comparison of appearance descriptors using rank-1 recognition rate. For multiple images of a person, average of image-wise descriptors is used for all methods. Best results in each column are colored in red.

6.6. Comparison with state-of-the-art

Table 6 and Table 7 give comparison of contemporary methods on iLIDS-VID and PRID, respectively. For this comparison, we report results of our method with and without application of exponential similarity kernel (Eq(3)). Our approach significantly improves state-of-the-art on both datasets. On more challenging iLIDS-VID, it outperforms previous best method of [19] by 10%, which uses hand crafted features and GMM based feature aggregation. Note that on both datasets, earlier neural network based methods do not compete favorably against hand crafted features. On iLIDS-VID, best neural network based ASTPN [49] achieves 62% rank-1 rate, whereas our method achieves 89.8% and 78.7%, with and without using exponential kernel, respectively. Similarly, on PRID, the best neural network based JST-RNN [60] achieves 79.4% rank-1 rate, in comparison to 95.2% and 92.1% achieved by our approach with and without using exponential kernel, respectively. One reason for inferior performance of most neural network based methods on these small sized datasets is the use of shallow networks such as in RCNN [33]. Contrarily, deeper models such as used in [57] fail to learn generalized embedding due to paucity of training data. Our method achieves higher performance due to the fact that we are able to better train a deeper network using less amount of data.

Method	r=1	r=5	r=10
PaMM [6]	30.3	56.3	70.3
DSVR [43]	39.5	61.1	71.7
MTL-LORAE [38]	43.0	60.1	70.3
STFV3D+KISS [29]	43.8	71.7	83.7
SI2DL [61]	48.7	81.1	89.2
LOMO+XQDA [25]	53.0	78.5	86.9
TAPR [54]	55.0	87.5	93.8
Zhang *et.al.* [10]	55.3	85.0	91.7
TDL [51]	56.3	87.6	95.6
GOG+SRID+KISS [17]	75.7	90.1	93.6
PAM [19]	79.5	95.1	97.6
AFDA [24]	37.5	62.7	73.0
DRCN [45]	46.1	76.8	89.7
RFA-Net+RSVM [50]	49.3	76.8	85.3
IDE+XQDA [57]	53.0	81.4	-
JST-RNN [60]	55.2	86.5	-
RCNN [33]	58.0	84.0	91.0
ASTPN [49]	62	86	94
RTL_{fc7}+XQDA	56.4	81.7	89.0
RTL_{pool5}+XQDA	57.8	82.1	89.1
RTL_{pool4}+XQDA	78.7	94.0	96.9
RTL_{fc7}+XQDA+Exp	62.9	85.1	91.5
RTL_{pool5}+XQDA+Exp	65.0	87.5	93.5
RTL_{pool4}+XQDA+Exp	**89.8**	**98.3**	**99.5**

Table 6: Recognition rate at rank=$\{1,5,10\}$ of different methods on iLIDS-VID, grouped based on use of neural networks

Finally, we compare our method with available results on MARS in Table 8. Due to large training set, best performing methods on MARS use neural networks. The best method, TriNet [14], fine-tunes parameters of ResNet-50 using triplet loss to learn embedding instead of using classification loss. The performance gap between our approach and TriNet can be attributed to using both deeper network and different loss function. It will be interesting to see performance of RTL in conjunction with triplet loss. With the exception of TriNet, most other methods give similar rank-1 rate; however, some yield better mAP. Note that JST-RNN [60] and other IDE models do not perform well on small datasets. Hence, our model has wider applicability.

7. Conclusion

When using identity loss and large amount of training data, RTL gives comparable performance to direct fine-tuning of network parameters. However, the performance difference between two transfer learning approaches is considerably in favor of RTL when training sets are small. The reason is that when using RTL only a few parameters are modified to compensate for the residual error of the network. Still, the higher order layers of the network are prone to over-fitting. Therefore, we propose using hybrid models

Method	r=1	r=5	r=10
ColorLBP [16]+RSVM	34.3	56.0	65.5
DSVR [43]	40.0	71.1	84.5
PaMM [6]	45.0	72.0	85.0
TDL [51]	56.7	80.0	87.6
STFV3D+KISS [29]	64.1	87.3	89.9
TAPR [54]	68.6	94.6	97.4
Zhang *et.al.* [10]	72.8	92.0	95.1
SI2DL [61]	76.7	95.6	96.7
GOG+SRID+KISS [17]	91.5	97.8	98.8
PAM [19]	92.5	**99.3**	**100.0**
AFDA [24]	43.0	72.7	84.6
RFA-Net+RSVM [50]	58.2	85.8	93.4
RCNN [33]	70.0	90.0	95.0
ASTPN [49]	77	95	99
IDE+XQDA [57]	77.3	93.5	-
JST-RNN [60]	79.4	94.4	-
RTL_{fc7}+XQDA	82.7	96.0	97.8
RTL_{pool5}+XQDA	82.4	95.2	98.0
RTL_{pool4}+XQDA	92.1	98.1	98.8
RTL_{fc7}+XQDA+Exp	88.1	97.2	98.5
RTL_{pool5}+XQDA+Exp	89.3	97.5	99.4
RTL_{pool4}+XQDA+Exp	**95.2**	98.9	99.6

Table 7: Recognition rate at rank=$\{1,5,10\}$ of different methods on PRID, grouped based on use of neural networks

Method	mAP	r=1	r=5	r=20
Zheng *et.al* [53]	-	55.5	70.2	80.2
IDE-CaffeNet [57]	42.4	60.0	77.9	87.9
IDE-ResNet [59]	44.1	62.7	-	-
IDE-CaffeNet+XQDA [57]	47.6	65.3	82.0	89.0
CDSC [39]	-	68.2	-	-
JST-RNN [60]	50.7	70.6	90.0	97.6
IDE-ResNet+XQDA [59]	55.1	70.5	-	-
LatentParts [22]	56.1	71.8	86.6	93.1
LuNet [14]	60.5	75.6	89.7	-
TriNet [14]	67.7	79.8	91.4	-
RTL_{fc7}+Euclidean	48.2	67.0	82.7	90.6
RTL_{pool5}+Euclidean	42.8	65.5	80.2	88.1
RTL_{pool4}+Euclidean	18.3	41.4	56.1	68.1
RTL_{fc7}+XQDA	49.6	67.9	83.4	90.7
RTL_{pool5}+XQDA	46.6	66.3	82.1	89.7
RTL_{pool4}+XQDA	34.8	58.3	71.3	80.8

Table 8: Comparison of different methods on MARS

where higher order domain specific layers are replaced with statistical metric learning. We demonstrate that the hybrid model performs significantly better on small datasets and gives comparable performance on large datasets. The ability of the model to generalize well from small amount of data is important for practical applications because frequent data collection in large amount for training is inconvenient.

References

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

[2] S. Bak, R. Kumar, and F. Bremond. Brownian descriptor: a rich meta-feature for appearance matching. In *WACV*, 2014. 2

[3] L. Bazzani, M. Cristani, A. Perina, M. Farenzena, and V. Murino. Multiple-shot person re-identification by hpe signature. In *ICPR*, 2010. 2

[4] T. Chen, I. Goodfellow, and J. Shlens. Net2net: Accelerating learning via knowledge transfer. In *International Conference on Learning Representations*, 2016. 2

[5] D. S. Cheng, M. Cristani, M. Stoppa, L. Bazzani, and V. Murnio. Custom pictorial structures for re-identification. In *BMVC*, 2011. 2

[6] Y.-J. Cho and K.-J. Yoon. Improving person re-identification via pose-aware multi-shot matching. In *CVPR*, 2016. 2, 8

[7] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information theoretic metric learning. In *ICML*, 2007. 2

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 1, 3

[9] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*, 2010. 1, 2

[10] C. Gao, J. Wang, L. Liu, J.-G. Yu, and N. Sang. Learning bidirectional temporal cues for video-based person re-identification. In *TCSVT*, 2017. 8

[11] M. Geng, Y. Wang, T. Xiang, and Y. Tian. Deep Transfer Learning for Person Re-identification. *CoRR*, 2016. 1

[12] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, 2008. 1, 2

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1, 2, 3, 4

[14] A. Hermans, L. Beyer, and B. Leibe. In Defense of the Triplet Loss for Person Re-Identification. *CoRR*, 2017. 4, 8

[15] M. Hirzer, C. Beleznai, P. M. Roth, and H. Bischof. Person re-identification by descriptive and discriminative classification. In *Image Analysis*, pages 91–102. Springer, 2011. 1, 2, 5

[16] M. Hirzer, P. Roth, M. Köstinger, and H. Bischof. Relaxed pairwise learned metric for person re-identification. In *ECCV*, 2012. 8

[17] S. Karanam, M. Gou, Z. Wu, A. Rates-Borras, O. Camps, and R. J. Radke. A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets. *CoRR*, 2016. 1, 5, 6, 7, 8

[18] S. Karanam, Y. Li, and R. J. Radke. Person re-identification with discriminatively trained viewpoint invariant dictionaries. In *ICCV*, 2015. 2

[19] F. M. Khan and F. Bremond. Multi-shot person re-identification using part appearance mixture. In *WACV*, 2017. 2, 5, 7, 8

[20] M. Köstinger, M. Hirzer, P. Wohlhart, P. M. Roth, and H. Bischof. Large scale metric learning from equivalence constraints. In *CVPR*, 2012. 1, 2

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, 2012. 3

[22] D. Li, X. Chen, Z. Zhang, and K. Huang. Learning Deep Context-aware Features over Body and Latent Parts for Person Re-identification. In *CVPR*, 2017. 8

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

[24] Y. Li, Z. Wu, S. Karanam, and R. Radke. Multi-shot human re-identification using adaptive fisher discriminant analysis. In *BMVC*, 2015. 2, 8

[25] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, 2015. 1, 2, 5, 7, 8

[26] G. Lisanti, I. Masi, A. Bagdanov, and A. D. Bimbo. Person re-identification by iterative re-weighted sparse ranking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2014. 2, 7

[27] G. Lisanti, I. Masi, and A. D. Bimbo. Matching people across camera views using kernel canonical correlation analysis. In *ICDSC*, 2014. 1, 2

[28] H. Liu, J. Feng, Z. Jie, K. Jayashree, B. Zhao, M. Qi, J. Jiang, and S. Yan. Neural person search machines. In *ICCV*, 2017. 2

[29] K. Liu, W. Zhang, and R. Huang. A spatio-temporal appearance representation for video-based pedestrian re-identification. In *ICCV*, 2015. 2, 5, 8

[30] B. Ma, Y. Su, and F. Jurie. Local descriptors encoded by fisher descriptors for person re-identification. In *ECCV Workshops*, 2012. 1, 2, 7

[31] B. Ma, Y. Su, and F. Jurie. Covariance descriptor based on bio-inspired features for person re-identification and face verification. *Image and Vision Computing*, 2014. 2, 7

[32] T. Matsukawa, T. Okabe, E. Suzuki, and Y. Sato. Hierarchical gaussian descriptor for person re-identification. In *CVPR*, 2016. 1, 2, 7

[33] N. McLaughlin, J. M. del Rincon, and P. Miller. Recurrent convolutional network for video-based person re-identification. In *CVPR*, 2016. 2, 4, 5, 7, 8

[34] A. Mignon and F. Jurie. Pcca: A new approach for distance learning from sparse pairwise constraints. In *CVPR*, 2012. 1, 2

[35] S. Pedagadi, J. Orwell, S. Velastin, and B. Boghossian. Local fisher discriminant analysis for pedestrian re-identification. In *CVPR*, 2013. 1, 2

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

[37] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 3

[38] C. Su, F. Yang, S. Zhang, Q. Tian, L. S. Davis, and W. Gao. Multi-task learning with low rank attribute embedding for person re-identification. In *CVPR*, 2015. 1, 2, 5, 8

[39] Y. T. Tesfaye, E. Zemene, A. Prati, M. Pelillo, and M. Shah. Multi-Target Tracking in Multiple Non-Overlapping Cameras using Constrained Dominant Sets. *CoRR*, 2017. 8

[40] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV*, 2006. 2

[41] R. R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *ECCV*, 2016. 2

[42] T. Wang, S. Gong, X. Zhu, and S. Wang. Person re-identification by video ranking. In *ECCV*, 2014. 1, 2, 5

[43] T. Wang, S. Gong, X. Zhu, and S. Wang. Person re-identification by discriminative selection in video ranking. *T-PAMI*, 2016. 2, 8

[44] K. Q. Weinberger and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *JMLR*, 2009. 1, 2

[45] L. Wu, C. Shen, and A. Hengel. Deep recurrent convolutional networks for video-based person re-identification: An end-to-end approach. In *arXiv*, 2016. 2, 8

[46] Y. Wu, M. Minoh, M. Mukunoki, W. Li, and S. Lao. Collaborative sparse approximation for mulitple-shot across-camera person re-identification. In *AVSS*, 2014. 2

[47] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016. 1, 2

[48] F. Xiong, M. Gou, O. Camps, and M. Sznaier. Person re-identification using kernel-based metric learning methods. In *ECCV*, 2014. 7

[49] S. Xu, Y. Cheng, K. Gu, Y. Yang, S. Chang, and P. Zhou. Jointly attentive spatial-temporal pooling networks for video-based person re-identification. In *ICCV*, 2017. 2, 7, 8

[50] Y. Yan, B. Ni, Z. Song, C. Ma, Y. Yan, and X. Yang. Person re-identification via recurrent feature aggregation. In *ECCV*, 2016. 2, 5, 8

[51] J. You, A. Wu, X. Li, and W. Zheng. Top-push video-based person re-identification. In *CVPR*, 2016. 2, 5, 8

[52] L. Zhang, M. Yang, and X. Feng. Sparse representation or collaborative representation: which helps face recognition? In *ICCV*, 2011. 2

[53] W. Zhang, S. Hu, and K. Liu. Learning compact appearance representation for video-based person re-identification. *CoRR*, 2017. 8

[54] W. Zhang, X. Yu, and X. He. Temporally aligned pooling representation for video-based person re-identification. In *ICIP*, 2016. 2, 8

[55] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 1, 2, 7

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

[57] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian. Mars: A video benchmark for large-scale person re-identification. In *ECCV*, 2016. 1, 2, 4, 5, 7, 8

[58] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*, 2011. 1, 2

[59] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking Person Re-identification with k-reciprocal Encoding. In *CVPR*, 2017. 7, 8

[60] Z. Zhou, Y. Huang, W. Wang, L. Wang, and T. Tan. See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification. In *CVPR*, 2017. 2, 7, 8

[61] X. Zhu, X.-Y. Jing, F. Wu, and H. Feng. Video-based person re-identification by simultaneously learning intra-video and inter-video distance metrics. In *IJCAI*, 2016. 8