# Unlabeled Samples Generated by GAN Improve the Person Re-identification Baseline *in vitro*

Zhedong Zheng, Liang Zheng and Yi Yang
University of Technology Sydney
zdzheng12,liangzheng06,yee.i.yang@gmail.com

## Abstract

*The main contribution of this paper is a simple semi-supervised pipeline that only uses the original training set without collecting extra data. It is challenging in 1) how to obtain more training data only from the training set and 2) how to use the newly generated data. In this work, the generative adversarial network (GAN) is used to generate unlabeled samples. We propose the label smoothing regularization for outliers (LSRO). This method assigns a uniform label distribution to the unlabeled images, which regularizes the supervised model and improves the baseline.*

*We verify the proposed method on a practical problem: person re-identification (re-ID). This task aims to retrieve a query person from other cameras. We adopt the deep convolutional generative adversarial network (DCGAN) for sample generation, and a baseline convolutional neural network (CNN) for representation learning. Experiments show that adding the GAN-generated data effectively improves the discriminative ability of learned feature embedding. On three large-scale datasets, Market-1501, CUHK03 and DukeMTMC-reID, we obtain +4.37%, +1.6% and +2.46% improvement in rank-1 precision over the baseline CNN, respectively. We additionally apply the proposed method to fine-grained bird recognition and achieve a +0.6% improvement over a strong baseline.*

## 1. Introduction

Unsupervised learning can serve as an important auxiliary task to supervised tasks [12, 25, 9, 24]. In this work, we propose a semi-supervised pipeline that works on the original training set without an additional data collection process. First, the training set is expanded with unlabeled data using a GAN. Then our model minimizes the sum of the supervised and the unsupervised losses through a new regularization method. This method is evaluated with person re-ID, which aims to spot the target person in different cameras. This has been recently viewed as an image retrieval problem [46].

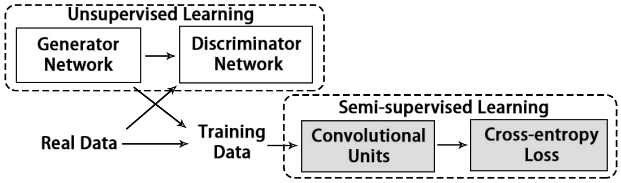


Figure 1. The pipeline of the proposed method. There are two components: a generative adversarial model [23] for unsupervised learning and a convolutional neural network for semi-supervised learning. “Real Data” represents the labeled data in the given training set; “Training data” includes both the “Real Data” and the generated unlabeled data. We aim to learn more discriminative embeddings with the “Training data”.

This paper addresses three challenges. First, current research in GANs typically considers the quality of the sample generation with and without semi-supervised learning *in vivo* [20, 28, 23, 5, 22, 37]. Yet a scientific problem remains unknown: moving the generated samples out of the box and using them in currently available learning frameworks. To this end, this work uses unlabeled data produced by the DCGAN model [23] in conjunction with the labeled training data. As shown in Fig. 1, our pipeline feeds the newly generated samples into another learning machine (i.e. a CNN). Therefore, we use the term “*in vitro*” to differentiate our method from [20, 28, 23, 5]; these methods perform semi-supervised learning in the discriminator of the GANs (*in vivo*).

Second, the challenge of performing semi-supervised learning using labeled and unlabeled data in CNN-based methods remains. Usually, the unsupervised data is used as a pre-training step before supervised learning [24, 9, 12]. Our method uses all the data simultaneously. In [21, 16, 20, 28], the unlabeled/weak-labeled real data are assigned labels according to pre-defined training classes, but our method assumes that the GAN generated data does not belong to any of the existing classes. The proposed LSRO method neither includes unsupervised pre-training nor label assignments for the known classes. We address semi-supervised learning from a new perspective. Since the unlabeled samples do not belong to any of the existing classes, they are assigned a uniform label distribution over the training classes. The network is trained not to predict a particular class for the generated data with high confidence.

Third, in person re-ID, data annotation is expensive, because one has to draw a pedestrian bounding box and assign an ID label to it. Recent progress in this field can be attributed to two factors: 1) the availability of large-scale re-ID datasets [45, 47, 40, 17] and 2) the learned embedding of pedestrians using a CNN [6, 8]. That being said, the number of images for each identity is still limited, as shown in Fig. 2. There are 17.2 images per identities in Market-1501 [45], 9.6 images in CUHK03 [17], and 23.5 images in DukeMTMC-reID [26] on average. So using additional data is non-trivial to avoid model overfitting. In the literature, pedestrian images used in training are usually provided by the training sets, without being expanded. So it is unknown if a larger training set with unlabeled images would bring any extra benefit. This observation inspired us to resort to the GAN samples to enlarge and enrich the training set. It also motivated us to employ the proposed regularization to implement a semi-supervised system.

In an attempt to overcome the above-mentioned challenges, this paper 1) adopts GAN in unlabeled data generation, 2) proposes the label smoothing regularization for outliers (LSRO) for unlabeled data integration, and 3) reports improvements over a CNN baseline on three person re-ID datasets. In more details, in the first step, we train DCGAN [23] on the original re-ID training set. We generate new pedestrian images by inputting 100-dim random vectors in which each entry falls within [-1, 1]. Some generated samples are shown in Fig. 3 and Fig. 5. In the second step, these unlabeled GAN-generated data are fed into the ResNet model [11]. The LSRO method regularizes the learning process by integrating the unlabeled data and, thus, reduces the risk of over-fitting. Finally, we evaluate the proposed method on person re-ID and show that the learned embeddings demonstrate a consistent improvement over the strong ResNet baseline.

To summarize, our contributions are:

- the introduction of a semi-supervised pipeline that integrates GAN-generated images into the CNN learning machine *in vitro*;
- an LSRO method for semi-supervised learning. The integration of unlabeled data regularizes the CNN learning process. We show that the LSRO method is superior to the two available strategies for dealing with unlabeled data; and
- a demonstration that the proposed semi-supervised pipeline has a consistent improvement over the ResNet baseline on three person re-ID datasets and one fine-grained recognition dataset.

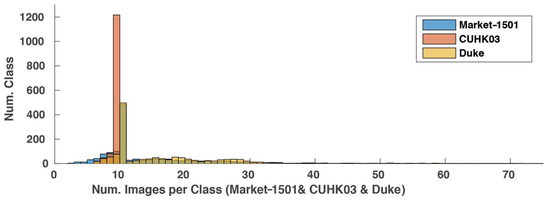


Figure 2. The image distribution per class in the dataset Market-1501 [45], CUHK03 [17] and DukeMTMC-reID [26]. We observe that all these datasets suffer from the limited images per class. Note that there are only a few classes with more than 20 images.

## 2. Related Work

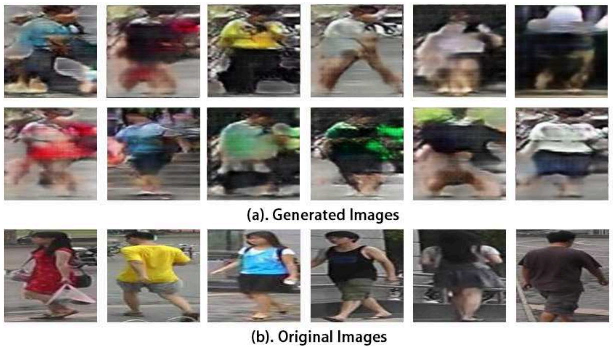
In this section, we will discuss the relevant works on GANs, semi-supervised learning and person re-ID.

### 2.1. Generative Adversarial Networks

The generative adversarial networks (GANs) learn two sub-networks: a generator and a discriminator. The discriminator reveals whether a sample is generated or real, while the generator produces samples to cheat the discriminator. The GANs are first proposed by Goodfellow *et al.* [10] to generate images and gain insights into neural networks. Then, DCGANs [23] provides some techniques to improve the stability of training. The discriminator of DCGAN can serve as a robust feature extractor. Salimans *et al.* [28] achieve a state-of-art result in semi-supervised classification and improves the visual quality of GANs. InfoGAN [5] learns interpretable representations by introducing latent codes. On the other hand, GANs also demonstrate potential in generating images for specific fields. Pathak *et al.* [22] propose an encoder-decoder method for image inpainting, where GANs are used as the image generator. Similarly, Yeh *et al.* [41] improve the inpainting performance by introducing two loss types. In [37], 3D object images are generated by a 3D-GAN. In this work, we do not focus on investigating more sophisticated sample generation methods. Instead, we use a basic GAN model [23] to generate unlabeled samples from the training data and show that these samples help improve discriminative learning.

### 2.2. Semi-supervised Learning

Semi-supervised learning is a sub-class of supervised learning taking unlabeled data into consideration, especially when the volume of annotated data is small. On the one


(a). Generated Images

(b). Original Images

Figure 3. Examples of GAN images and real images. (a) The top two rows show the pedestrian samples generated by DCGAN [23] trained on the Market-1501 training set [45]. (b) The bottom row shows the real samples in training set. Although the generated images in (a) can be easily recognized as fake images by a human, they still serve as an effective regularizer in our experiment.

hand, some research treats unsupervised learning as an auxiliary task to supervised learning. For example, in [12], Hinton *et al*. learn a stack of unsupervised restricted Boltzmann machines to pre-train the model. Ranzato *et al*. propose to reconstruct the input at every level of a network to get a compact representation [24]. In [25], the auxiliary task of ladder networks is to denoise representations at every level of the model. On the other hand, several works assign labels to the unlabeled data. Papandreou *et al*. [21] combine strong and weak labels in CNNs using an expectation-maximization (EM) process for image segmentation. In [16], Lee assigns a "pseudo label" to the unlabeled data in the class that has the maximum predicted probability. In [20, 28], the samples produced by the generator of the GAN are all taken as one class in the discriminator. Departing from previous semi-supervised works, we adopt a different regularization approach by assigning a uniform label distribution to the generated samples.

### 2.3. Person Re-identification

Recent progress in person re-ID mainly consists of advancing CNNs. Yi *et al*. [42] split a pedestrian image into three horizontal parts and respectively train three part-CNNs to extract features. Similarly, Cheng *et al*. [6] split the convolutional map into four parts and fuse the part features with the global feature. In [17], Li *et al*. add a new layer that multiplies the activation of two images in different horizontal stripes. They use this layer to explicitly allow patch matching in the CNN. Later, Ahmed *et al*. [2] improve the performance by proposing a new patch matching layer that compares the activation of two images in neighboring pixels. In addition, Varior *et al*. [31] combine the CNN with some gate functions, aiming to adaptively focus on the salient parts of input image pairs, this method is limited by computational inefficiency because the input should be image pairs.

A CNN can be very discriminative by itself without explicit part-matching. Zheng *et al*. [46, 47] directly use a conventional fine-tuning approach (called the ID-discriminative embedding, or IDE) on the Market-1501 dataset [45] and its performance exceeds many other recent results. Wu *et al*. [39] combine the CNN embedding with hand-crafted features. In [48], Zheng *et al*. combine an identification model with a verification model and improve the fine-tuned CNN performance. In this work, we adopt the IDE model [46, 47] as a baseline, and show that the GAN samples and LSRO effectively improve its performance. Recently, Barbosa *et al*. [3] propose synthesizing human images through a photorealistic body generation software. These images are used to pre-train an IDE model before dataset-specific fine-tuning. Our method is different from [3] in both data generation and the training strategy.

## 3. Network Overview

In this section, we describe the pipeline of the proposed method. As shown in Fig. 1, the real data in the training set is used to train the GAN model. Then, the real training data and the newly generated samples are combined into training input for the CNN. In the following section, we will illustrate the structure of the two components, *i.e*., the GAN and the CNN, in detail. Note that, **our system does not make major changes to the network structures of the GAN or the CNN with one exception - the number of neurons in the last fully-connected layer in the CNN is modified according to the number of training classes.**

### 3.1. Generative Adversarial Network

Generative adversarial networks have two components: a generator and a discriminator. For the generator, we follow the settings in [23]. We start with a 100-dim random vector and enlarge it to $4 \times 4 \times 16$ using a linear function. To enlarge the tensor, five deconvolution functions are used with a kernel size of $5 \times 5$ and a stride of 2. Every deconvolution is followed by a rectified linear unit and batch normalization. Additionally, one optional deconvolutional layer with a kernel size of $5 \times 5$ and a stride of 1, and one *tanh* function are added to fine-tune the result. A sample that is $128 \times 128 \times 3$ in size can then be generated.

The input of the discriminator network includes the generated images and the real images in the training set. We use five convolutional layers to classify whether the generated image is fake. Similarly, the size of the convolutional filters is $5 \times 5$ and their stride is 2. We add a fully-connected layer to perform the binary classification (real or fake).

### 3.2. Convolutional Neural Network

The ResNet-50 [11] model is used in our experiment. We resize the generated images to $256 \times 256 \times 3$ using bilinear sampling. The generated images are mixed with the original training set as the input of the CNN. That is, the labeled and unlabeled data are simultaneously trained. These training images are shuffled. Following the conventional fine-tuning strategy [46], we use a model pre-trained on ImageNet [27]. We modify the last fully-connected layer to have $K$ neurons to predict the $K$-classes, where $K$ is the number of the classes in the original training set (as well as the merged new training set). Unlike [20, 28], we do not view the new samples as an extra class but assign a uniform label distribution over the existing classes. So the last fully-connected layer remains $K$-dimensional. The assigned label distribution of the generated images is discussed in the next section.

## 4. The Proposed Regularization Method

In this section, we first revisit the label smoothing regularization (LSR), which is used for fully-supervised learning. We then extend LSR to the scenario of unlabeled learning, yielding the proposed label smoothing regularization for outliers (LSRO) method.

### 4.1. Label Smoothing Regularization Revisit

LSR was proposed in the 1980s and recently rediscovered by Szegedy *et al.* [29]. In a nutshell, LSR assigns small values to the non-ground truth classes instead of $0$. This strategy discourages the network to be tuned towards the ground truth class and thus reduces the chances of over-fitting. LSR is proposed for use with the cross-entropy loss [29].

Formally, let $k \in \{1, 2, ..., K\}$ be the pre-defined classes of the training data, where $K$ is the number of classes. The cross-entropy loss can be formulated as:

$$l = -\sum_{k=1}^{K} \log{(p(k))} q(k), \tag{1}$$

where $p(k) \in [0, 1]$ is the predicted probability of the input belonging to class $k$, and can be outputted by CNN. It is derived from the softmax function which normalizes the output of the previous fully-connected layer. $q(k)$ is the ground truth distribution. Let y be the ground truth class label, $q(k)$ can be defined as:

$$q(k) = \begin{cases} 0 & k \neq y \\ 1 & k = y \end{cases}. \tag{2}$$

If we discard the $0$ terms in Eq. 1, the cross-entropy loss is equivalent to only considering the ground truth term in Eq. 3.

$$l = -\log{(p(y))}. \tag{3}$$

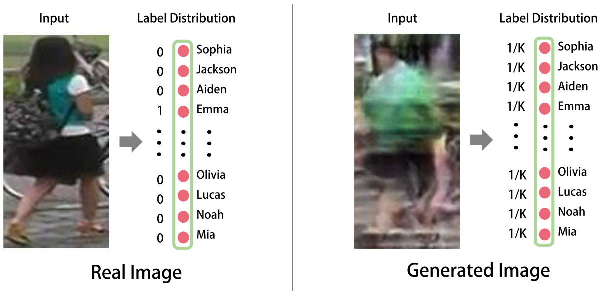


Figure 4. The label distributions of a real image and a GAN-generated image in our system. We use a classical label distribution (Eq. 2) for the real image (left). For the generated image (right), we employ the proposed LSRO label distribution (Eq. 6), *e.g.* a uniform distribution on every training class because the generated image is assumed to belong to none of the training classes. We employ a cross-entropy loss that combines the two types of label distributions as the optimization objective (Eq. 7).

So, minimizing the cross-entropy loss is equivalent to maximizing the predicted probability of the ground-truth class. In [29], the label smoothing regularization (LSR) is introduced to take the distribution of the non-ground truth classes into account. The network is thus encouraged not to be too confident towards the ground truth. In [29], the label distribution $q_{LSR}(k)$ is written as:

$$q_{LSR}(k) = \begin{cases} \frac{\varepsilon}{K} & k \neq y \\ 1 - \varepsilon + \frac{\varepsilon}{K} & k = y \end{cases}, \tag{4}$$

where $\varepsilon \in [0, 1]$ is a hyperparameter. If $\varepsilon$ is zero, Eq. 4 reduces to Eq. 2. If $\varepsilon$ is too large, the model may fail to predict the ground truth label. So in most cases, $\varepsilon$ is set to $0.1$. Szegedy *et al.* assume that the non-ground truth classes take on a uniform label distribution. Considering Eq. 1 and Eq. 4, the cross-entropy loss evolves to:

$$l_{LSR} = -(1 - \varepsilon) \log{(p(y))} - \frac{\varepsilon}{K} \sum_{k=1}^{K} \log{(p(k))}. \tag{5}$$

Compared with Eq. 3, Eq. 5 pays additional attention to the other classes, rather than only the ground truth class. In this paper, we do not employ LSR on the IDE baseline because it yields a slightly lower performance than using Eq. 2 (see Section 5.3). We re-introduce LSR because it inspires us in designing the LSRO method.

### 4.2. Label Smoothing Regularization for Outliers

The label smoothing regularization for outliers (LSRO) is used to incorporate the unlabeled images in the network. This extends LSR from the supervised domain to leverage unsupervised data generated by the GAN.

In LSRO, we propose a virtual label distribution for the unlabeled images. We set the virtual label distribution to be uniform over all classes, due to two inspirations. 1) We assume that the generated samples do not belong to any predefined classes. 2) LSR assumes a uniform distribution over the all classes to address over-fitting. During testing, we expect that the maximum class probability of a generated image will be low, *i.e.*, the network will fail to predict a particular class with high confidence. Formally, for a generated image, its class label distribution, $q_{LSRO}(k)$, is defined as:

$$q_{LSRO}(k) = \frac{1}{K}. \tag{6}$$

We call Eq. 6 the label smoothing regularization for outliers (LSRO).

The one-hot distribution defined in Eq. 2 will still be used for the loss computation for the real images in the training set. Combining Eq. 2, Eq. 6 and Eq. 1, we can re-write the cross-entropy loss as:

$$l_{LSRO} = -(1-Z)\log(p(y)) - \frac{Z}{K}\sum_{k=1}^{K}\log(p(k)). \tag{7}$$

For a real training image, $Z = 0$. For a generated training image, $Z = 1$. So our system actually has two types of losses, one for real images and one for generated images.

**Advantage of LSRO.** Using LSRO, we can deal with more training images (outliers) that are located near the real training images in the sample space, and introduce more color, lighting and pose variances to regularize the model. For instance, if we only have one green-clothed identity in the training set, the network may be misled into considering that the color green is a discriminative feature, and this limits the discriminative ability of the model. By adding generated training samples, such as an unlabeled green-clothed person, the classifier will be penalized if it makes the wrong prediction towards the labeled green-clothed person. In this manner, we encourage the network to find more underlying causes and to be less prone to over-fitting. We only use the GAN trained on the original training set to produce outlier images. It would be interesting to further evaluate whether real-world unlabeled images are able to achieve a similar effect (see Table 4).

**Competing methods.** We compare LSRO with two alternative methods. Details of both methods are available in existing literature [20, 28, 16]; breif descriptions follow.

- **All in one.** Using [20, 28], a new class label is created, *i.e.*, $K + 1$, and every generated sample is assigned to this class. CNN training follows in Section 5.2.
- **Pseudo label.** Using [16], during network training, each incoming GAN-image is passed forward through the current network and is assigned a pseudo label by taking the maximum value of the probability prediction vector ($p(k)$ in Eq. 1). This GAN-image can be thus trained in the network with this pseudo label. During training, the pseudo label is assigned *dynamically*, so that the same GAN-image may receive different pseudo labels each time it is fed into the network. In our experiments, we begin feeding GAN images and assigning them pseudo labels after 20 epochs. We also set a global weight to the softmax loss of 0.1 to the GAN and 1 to the real images.

Our experimental results show that the two methods also work on the GAN images and that LSRO is superior to "All in one" and "Pseudo label". Explanations are provided in the Section 5.3.

## 5. Experiment

We mainly evaluate the proposed method using the Market-1501 [45] dataset, because it is a large scale and has a fixed training/testing split. We also report results on the CUHK03 dataset [17], **but due to the computational cost of 20 training/testing splits, we only use the GAN images generated from the Market-1501 dataset**. In addition, we evaluate our method on a recently released pedestrian dataset DukeMTMC-reID [26] and a fine-grained recognition dataset CUB-200-2011 [34].

### 5.1. Person Re-id Datasets

**Market-1501** is a large-scale person re-ID dataset collected from six cameras. It contains 19,732 images for testing and 12,936 images for training. The images are automatically detected by the deformable part model (DPM) [7], so the misalignment is common, and the dataset is close to realistic settings. There are 751 identities in the training set and 750 identities in the testing set. There is an average of 17.2 training identity images in the set. We use all the 12,936 detected images from the training set to train the GAN model.

**CUHK03** contains 14,097 images of 1,467 identities. Each identity is captured by two cameras on the CUHK campus. This dataset contains two image sets. One is annotated by hand-drawn bounding boxes, and the other is produced by the DPM detector [7]. We use the detected set in this paper. There is an average of 9.6 training identity images in the set. We report the averaged result after training/testing 20 times. We use the **single shot** setting.

**DukeMTMC-reID** is a subset of the newly-released multi-target, multi-camera pedestrian tracking dataset [26]. The original dataset contains eight 85-minute high-resolution videos from eight different cameras. Hand-drawn pedestrian bounding boxes are available. In this work, we use a subset of [26] for image-based re-ID, in the format of the Market-1501 dataset [45]. We crop pedestrian images from the videos every 120 frames, yielding 36,411 total bounding boxes with IDs annotated by [26]. The DukeMTMC-reID dataset for re-ID has 1,812 identities from eight cameras. There are 1,404 identities appearing in more than two cameras and 408 identities (distractor ID) who appear in only one camera. We randomly select 702 IDs as the training set and the remaining 702 IDs as the testing set. In the testing set, we pick one query image for each ID in each camera and put the remaining images in the gallery. As a result, we get 16,522 training images with 702 identities, 2,228 query images of the other 702 identities and 17,661 gallery images. The evaluation protocol is available on our website [1]. Some example re-ID results from the DukeMTMC-reID are shown in Fig. 6.

## 5.2. Implementation Details

**CNN re-ID baseline.** We adopt the CNN re-ID baseline used in [46, 47]. Specifically, the Matconvnet [33] package is used. During training, We use the ResNet-50 model [11] and modify the fully-connected layer to have 751 and 1,367 neurons for Market-1501 and CUHK03, respectively. All the images are resized to 256 × 256 before being randomly cropped into 224 × 224 with random horizontal flipping. We insert a dropout layer before the final convolutional layer and set the dropout rate to 0.5 for CUHK03 and 0.75 for Market-1501 and DukeMTMC-reID, respectively. We use stochastic gradient descent with momentum 0.9. The learning rate of the convolution layers is set to 0.002 and decay to 0.0002 after 40 epochs and we stop training after the 50th epochs. During testing, we extract the 2,048-dim CNN embedding in the last convolutional layer for an input image with a size of 224 × 224. The similarity between two images is calculated by a cosine distance before ranking.

**GAN training and testing.** We use Tensorflow [1] and the DCGAN package[2] to train the GAN model using the provided data in the original training set without preprocessing (*e.g.*, foreground detection). All the images are resized to 128 × 128 and randomly flipped before training. We use Adam [13] with the parameters $\beta_1 = 0.5, \beta_2 = 0.99$. We stop training after 30 epochs. During GAN testing, we input a 100-dim random vector in GAN, and the value of each entry ranges in [-1, 1]. The outputted image is resized to 256 × 256 and then used in CNN training (with LSRO). More GAN images are shown in Fig. 5.

## 5.3. Evaluation

**The ResNet baseline.** Using the training/testing procedure described in Section 5.2, we report the baseline performance of ResNet in Table 1, Table 5 and Table 3. The rank-1 accuracy is 73.69%, 71.5% and 60.28% on Market-1501, CUHK03 and DukeMTMC-reID respectively. Our baseline results are on par with the those reported in [46, 48]. Note that the baseline alone exceeds many previous works [18, 32, 43].

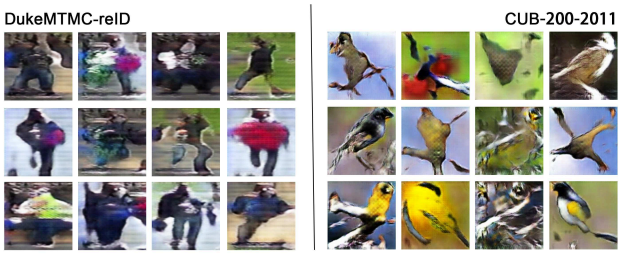


Figure 5. The newly generated images from a DCGAN model trained on DukeMTMC-reID and CUB-200-2011. Through LSRO, they are added to the training sets of DukeMTMC-reID and CUB-200-2011 to regularize the CNN model.

**The GAN images improve the baseline.** As shown in Table 2, when we add $24,000$ GAN images to the CNN training, our method significantly improves the re-ID performance on Market-1501. We observe improvement of +4.37% (from 73.69% to 78.06%) and +4.75% (from 51.48% to 56.23%) in rank-1 accuracy and mAP, respectively. On CUHK03, we observe improvements of +1.6%, +1.2%, +0.8%, and +1.6% in rank-1, 5, 10 accuracy and mAP, respectively. The improvement on CUHK03 is relatively small compared to that of Market-1501, because the DCGAN model is trained on Market-1501 and the generated images share a more similar distribution with Market-1501 than CUHK03. We also observe improvements of +2.46% and +2.14% in rank-1 and mAP, respectively, on the strong ResNet baseline in the DukeMTMC-reID dataset. These results indicate that the unlabeled images generated by the GAN effectively yield improvements over the baseline using the LSRO method.

**The impact of using different numbers of GAN images during training.** We evaluate how the number of GAN images affects the re-ID performance. Since the unlabelled data is easy to obtain, we expect the method would learn more general knowledge as the number of unlabelled images increases. The experimental results on Market-1501 are shown in Table 2. We note that the number of real training images in Market-1501 is 12,936. Two observations are made.

First, the addition of different numbers of GAN images consistently improves the baseline. Adding approximately 3×GAN images compared to the real training set still has a +2.38% improvement to rank-1 accuracy.

Second, the peak performance is achieved when 2×GAN images are added. When too few GAN sample are incorporated into the system, the regularization ability of the LSRO is inadequate. In contrast, when too many GAN samples

[1] https://github.com/layumi/DukeMTMC-reID_evaluation
[2] https://github.com/carpedm20/DCGAN-tensorflow

| method | Single Query | | Multi. Query | |
|---|---|---|---|---|
| | rank-1 | mAP | rank-1 | mAP |
| BoW+kissme [45] | 44.42 | 20.76 | - | - |
| MR CNN [30] | 45.58 | 26.11 | 56.59 | 32.26 |
| FisherNet [38] | 48.15 | 29.94 | - | - |
| SL [4] | 51.90 | 26.35 | - | - |
| S-LSTM [32] | - | - | 61.6 | 35.3 |
| DNS [43] | 55.43 | 29.87 | 71.56 | 46.03 |
| Gate Reid [31] | 65.88 | 39.55 | 76.04 | 48.45 |
| SOMAnet [3]* | 73.87 | 47.89 | 81.29 | 56.98 |
| Verif.-Identif. [48]* | 79.51 | 59.87 | 85.84 | 70.33 |
| DeepTransfer [8]* | 83.7 | 65.5 | **89.6** | 73.8 |
| Basel. [46, 48]* | 73.69 | 51.48 | 81.47 | 63.95 |
| Basel. + LSRO | 78.06 | 56.23 | 85.12 | 68.52 |
| Verif-Identif. + LSRO | **83.97** | **66.07** | 88.42 | **76.10** |

Table 1. Comparison of the state-of-the-art methods reported on the Market-1501 dataset. We also provide results of the fine-tuned ResNet baseline. Rank-1 precision (%) and mAP (%) are listed. * the respective paper is on ArXiv but not published.

| # GAN Img. | LSRO | | All in one | | Pseudo label | |
|---|---|---|---|---|---|---|
| | rank-1 | mAP | rank-1 | mAP | rank-1 | mAP |
| 0 (basel.) | 73.69 | 51.48 | 73.69 | 51.48 | 73.69 | 51.48 |
| 12,000 | 76.81 | 55.32 | 75.33 | 52.82 | 76.07 | 53.56 |
| 18,000 | 77.26 | 55.55 | 77.20 | 55.04 | 76.34 | 53.45 |
| 24,000 | **78.06** | **56.23** | 76.63 | 55.12 | 75.80 | 53.03 |
| 30,000 | 77.38 | 55.48 | 75.95 | 55.18 | 75.21 | 52.65 |
| 36,000 | 76.07 | 54.59 | 76.87 | 55.47 | 74.67 | 52.38 |

Table 2. Comparison of LSRO, "All in one", and "Pseudo label" under different numbers of GAN-generated images on Market-1501. We show that LSRO is superior to the other two methods whose best performance is highlighted in blue and red, respectively. Rank-1 accuracy (%) and mAP (%) are shown.

| method | rank-1 | mAP |
|---|---|---|
| BoW+kissme [45] | 25.13 | 12.17 |
| LOMO+XQDA [18] | 30.75 | 17.04 |
| Basel. [46, 48] | 65.22 | 44.99 |
| Basel. + LSRO | **67.68** | **47.13** |

Table 3. Comparison of the baseline on DukeMTMC-reID. Rank-1 accuracy (%) and mAP (%) are shown.

| Unsup. Data | rank-1 | mAP |
|---|---|---|
| 0 (basel.) | 73.69 | 51.48 |
| CUHK03-Real-12000 | 75.65 | 53.25 |
| Market-1501-GAN-12000 | **76.81** | **55.32** |

Table 4. We add the 12,000 real pedestrian images in CUHK03 as outliers to Market-1501. We find the model trained on the generated samples slightly out-performs the model trained on CUHK03 real data. Rank-1 accuracy (%) and mAP (%) are shown.

| method | rank-1 | rank-5 | rank-10 | mAP |
|---|---|---|---|---|
| KISSME [14] | 11.7 | 33.3 | 48.0 | - |
| DeepReID [17] | 19.9 | 49.3 | 64.7 | - |
| BoW+HS [45] | 24.3 | - | - | - |
| LOMO+XQDA [18] | 46.3 | 78.9 | 88.6 | - |
| SI-CI [36] | 52.2 | 84.3 | 94.8 | - |
| DNS [43] | 54.7 | 80.1 | 88.3 | - |
| SOMAnet [3]* | 72.4 | 92.1 | 95.8 | - |
| Verif-Identif. [48]* | 83.4 | 97.1 | 98.7 | 86.4 |
| DeepTransfer [8]* | 84.1 | - | - | - |
| Basel. [46, 48]* | 71.5 | 91.5 | 95.9 | 75.8 |
| Basel.+LSRO | 73.1 | 92.7 | 96.7 | 77.4 |
| Verif-Identif. + LSRO | **84.6** | **97.6** | **98.9** | **87.4** |

Table 5. Comparison of the state-of-the-art reports on the CUHK03 dataset. We list the fine-tuned ResNet baseline as well. The mAP (%) and rank1 (%) precision are presented. * the respective paper is on ArXiv but not published.

are present, the learning machine tends to converge towards assigning uniform prediction probabilities to all the training samples, which is not desirable. Therefore, a trade-off is recommended to avoid poor regularization and over-fitting of uniform label distributions.

**GAN images vs. real images in training.** To further evaluate the proposed method, we replace the GAN images with the real images from the CUHK03 dataset which are viewed as unlabeled in the experiment. Since CUHK03 only 14,097 images, we randomly select 12,000 for the fair comparison.

Experimental results are shown in Table 4. We compare the results obtained using the 12,000 CUHK03 images and the 12,000 GAN images. We find the real data from CUHK03 also assists in the regularization and improves the performance. But the model trained with GAN-generated data is sightly better. In fact, although the images generated from DCGAN are visually imperfect (see Fig. 3), they still possess similar regularization ability as the real images.

**Comparison with the two competing methods.** We compare the LSRO method with the "All in one" and "Pseudo label" methods implied in [20, 28] and [16], respectively. The experimental results on Market-1501 are summarized in Table 2.

We first observe that both strategies yield improvement over the baseline. The "All in one" method treats all the unlabeled samples as a new class, which forces the network to make "careful" predictions for the existing $K$ classes. The "Pseudo label" method gradually labels the new data, and thus introduces more variance to the network.

Nevertheless, we find that LSRO exceeds both strategies by approximately +1% $\sim$ +2%. We speculate the reason is that the "All in one" method makes a coarse label estimation, while the "Pseudo label" originally assumes that all the unlabeled data belongs to the existing classes [16] which is

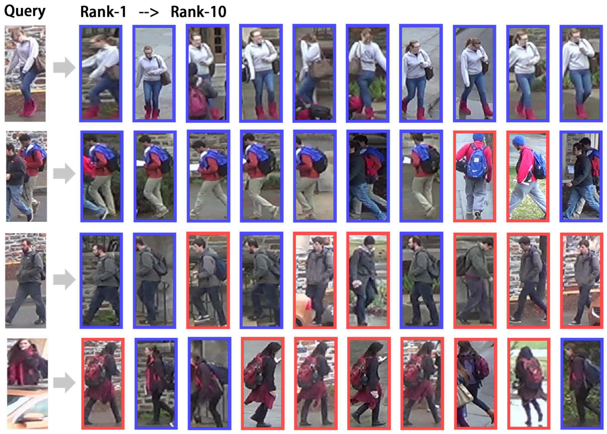


Figure 6. Sample retrieval results on DukeMTMC-reID using the proposed method. The images in the first column are the query images. The retrieved images are sorted according to the similarity scores from left to right. The correct matches are in the blue rectangles, and the false matching images are in the red rectangles. DukeMTMC-reID is challenging because it contains pedestrians with occlusions and similar appearance.

not true in person re-ID. While these two methods still use the one-hot label distribution, the LSRO method makes a less stronger assumption (label smoothing) towards the labels of the GAN images. These reasons may explain why LSRO has a superior performance.

**Comparison with the state-of-the-art methods.** We compare our method with the state-of-the-art methods on Market-1501 and CUHK03, listed in Table 1 and Table 5, respectively. On the Market-1501, we achieve **rank-1 accuracy = 78.06%, mAP = 56.23%** when using the single query mode, which is the best result compared to the published papers, and the second best among all the available results including ArXiv papers. On the CUHK03, we arrive at **rank-1 accuracy = 73.1%, mAP = 77.4%** which is also very competitive. The previous best result is produced by combining the identification and the verification losses [8, 48]. We further investigate whether the LSRO could work on this two-stream model. We fine-tuned the publicly available model [3] in [48] with LSRO and achieve state-of-the-art accuracy **rank-1 accuracy = 83.97%, mAP = 66.07%** on Market-1501. On CUHK03, we also observe a state-of-the art performance **rank-1 accuracy = 84.6%, mAP = 87.4%**. We, therefore, show that the LSRO method is complementary to previous methods due to the regularization of the GAN data.

### 5.4. Fine-grained Recognition

Fine-grained classification also faces the problem of a lack of training data and annotations. To further test the effectiveness of our method, we provide results on the CUB-200-2011 dataset [34]. This dataset contains 200 bird classes with 29.97 training images per class on average. Bounding boxes are used in both training and testing. We do not use part annotations. In our implementation, the ResNet baseline has a recognition accuracy of 82.6%, which is slightly higher than the 82.3% reported in [19]. This is the baseline we will compare our method with.

| method | model | annotation | top-1 |
|---|---|---|---|
| Zhang *et al.* [44] | AlexNet | 2×part | 76.7 |
| Zhang *et al.* [44] | VGGNet | 2×part | 81.6 |
| Liu *et al.* [19] | ResNet-50 | attribute | 82.9 |
| Wang *et al.* [35] | 3×VGGNet | × | 83.0 |
| Basel. [19] | ResNet-50 | × | 82.6 |
| Basel.+LSRO | ResNet-50 | × | 83.2 |
| Basel.+LSRO | 2×ResNet-50 | × | **84.4** |

Table 6. We show the recognition accuracy (%) on CUB-200-2011. The proposed method has a 0.6% improvement over the competitive baseline. The two-model ensemble shows a competitive result.

Using the same pipeline in Fig. 1, we train DCGAN on the 5,994 training images with the bounding box, and then we combine the real images with the generated images (see Fig. 5) to train the CNN. During testing, we adopt the standard 10-crop testing [15], which uses 256 × 256 images as input and the averaged prediction as the classification result. As shown in Table 6, the strong baseline alone is superior to some recent methods, and the proposed method further yields an improvement of +0.6% (from 82.6% to 83.2%). We also combine the two models generated by our method with a different initialization to form an ensemble. This leads to an **84.4%** recognition accuracy. In [19], Liu *et al.* report an 85.5% recognition accuracy with a five-model ensemble using parts and a global scene. We do not include this result because extra annotations are used. We focus on the regularization ability of the GAN, but not on producing a state-of-the-art result.

## 6. Conclusion

In this paper, we propose an "*in vitro*" usage of the GANs for discriminative learning, *i.e.*, person re-identification. Using a baseline DCGAN model [23], we show that the imperfect GAN images effectively demonstrate their regularization ability when trained with a ResNet baseline network. Through the proposed LSRO method, we mix the unlabeled GAN images with the labeled real training images for simultaneous semi-supervised learning. Albeit simple, we demonstrate consistent performance improvement over the re-ID and fine-grained recognition baseline systems, which sheds light on the practical use of GAN-generated data.

In the future, we will continue to investigate on whether

[3] https://github.com/layumi/2016_person_re-ID

GAN images of better visual quality yield superior results when integrated into supervised learning. This paper provides some baseline evaluations using the imperfect GAN images and the future investigation would be intriguing.

## References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, 2016.

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

[3] I. B. Barbosa, M. Cristani, B. Caputo, A. Rognhaugen, and T. Theoharis. Looking beyond appearances: Synthetic training data for deep cnns in re-identification. *arXiv:1701.03153*, 2017.

[4] D. Chen, Z. Yuan, B. Chen, and N. Zheng. Similarity learning with spatial constraints for person re-identification. In *CVPR*, 2016.

[5] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

[6] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng. Person reidentification by multi-channel parts-based cnn with improved triplet loss function. In *CVPR*, 2016.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[8] M. Geng, Y. Wang, T. Xiang, and Y. Tian. Deep transfer learning for person re-identification. *arXiv:1611.05244*, 2016.

[9] I. Goodfellow, M. Mirza, A. Courville, and Y. Bengio. Multiprediction deep boltzmann machines. In *NIPS*, 2013.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[12] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[13] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

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

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

[16] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop*, 2013.

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

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

[19] X. Liu, J. Wang, S. Wen, E. Ding, and Y. Lin. Localizing by describing: Attribute-guided attention localization for fine-grained recognition. *arXiv:1605.06217*, 2016.

[20] A. Odena. Semi-supervised learning with generative adversarial networks. *arXiv:1606.01583*, 2016.

[21] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *ICCV*, 2015.

[22] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[23] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2016.

[24] M. Ranzato and M. Szummer. Semi-supervised learning of compact document representations with deep networks. In *ICML*.

[25] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *NIPS*, 2015.

[26] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV Workshop*, 2016.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[28] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

[30] E. Ustinova, Y. Ganin, and V. Lempitsky. Multiregion bilinear convolutional neural networks for person re-identification. *arXiv:1512.05300*, 2015.

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

[32] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang. A siamese long short-term memory architecture for human re-identification. In *ECCV*, 2016.

[33] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. In *ACMMM*, 2015.

[34] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[35] D. Wang, Z. Shen, J. Shao, W. Zhang, X. Xue, and Z. Zhang. Multiple granularity descriptors for fine-grained categorization. In *ICCV*, 2015.

[36] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang. Joint learning of single-image and cross-image representations for person reidentification. In *CVPR*, 2016.

[37] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, 2016.

[38] L. Wu, C. Shen, and A. van den Hengel. Deep linear discriminant analysis on fisher networks: A hybrid architecture for person re-identification. *Pattern Recognition*, 2016.

[39] S. Wu, Y.-C. Chen, X. Li, A.-C. Wu, J.-J. You, and W.-S. Zheng. An enhanced deep feature representation for person re-identification. In *WACV*, 2016.

[40] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang. End-to-end deep learning for person search. *arXiv:1604.01850*, 2016.

[41] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with perceptual and contextual losses. *arXiv:1607.07539*, 2016.

[42] D. Yi, Z. Lei, and S. Z. Li. Deep metric learning for practical person re-identification. *arXiv:1407.4979*, 2014.

[43] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. *arXiv:1603.02139*, 2016.

[44] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based r-cnns for fine-grained category detection. In *ECCV*, 2014.

[45] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.

[46] L. Zheng, Y. Yang, and A. G. Hauptmann. Person re-identification: Past, present and future. *arXiv:1610.02984*, 2016.

[47] L. Zheng, H. Zhang, S. Sun, M. Chandraker, and Q. Tian. Person re-identification in the wild. *arXiv:1604.02531*, 2016.

[48] Z. Zheng, L. Zheng, and Y. Yang. A discriminatively learned cnn embedding for person re-identification. *arXiv:1611.05666*, 2016.