Robust Person Re-identification by Modelling Feature Uncertainty

Citation for published version:
Yu, T, Li, D, Yang, Y, Hospedales, T & Xiang, T 2020, Robust Person Re-identification by Modelling Feature Uncertainty. in *2019 IEEE/CVF International Conference on Computer Vision (ICCV).* Institute of Electrical and Electronics Engineers (IEEE), pp. 552-561, International Conference on Computer Vision 2019, Seoul, Korea, Republic of, 27/10/19. https://doi.org/10.1109/ICCV.2019.00064

Digital Object Identifier (DOI):
10.1109/ICCV.2019.00064

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
2019 IEEE/CVF International Conference on Computer Vision (ICCV)

Robust Person Re-identification by Modelling Feature Uncertainty

Tianyuan Yu[1], Da Li[1,2], Yongxin Yang[1], Timothy Hospedales[2,3], Tao Xiang[1,2]

[1]University of Surrey [2]Samsung AI Centre, Cambridge [3]The University of Edinburgh

{tianyuan.yu, d.li, yongxin.yang, t.xiang}@surrey.ac.uk t.hospedales@ed.ac.uk

Abstract

We aim to learn deep person re-identification (ReID) models that are robust against noisy training data. Two types of noise are prevalent in practice: (1) label noise caused by human annotator errors and (2) data outliers caused by person detector errors or occlusion. Both types of noise pose serious problems for training ReID models, yet have been largely ignored so far. In this paper, we propose a novel deep network termed DistributionNet for robust ReID. Instead of representing each person image as a feature vector, DistributionNet models it as a Gaussian distribution with its variance representing the uncertainty of the extracted features. A carefully designed loss is formulated in DistributionNet to unevenly allocate uncertainty across training samples. Consequently, noisy samples are assigned large variance/uncertainty, which effectively alleviates their negative impacts on model fitting. Extensive experiments demonstrate that our model is more effective than alternative noise-robust deep models. The source code is available at: https://github.com/TianyuanYu/DistributionNet.

1. Introduction

Person re-identification (ReID) aims to match people across a camera network with non-overlapping camera views. When a person is captured by different cameras with different viewing conditions, his/her appearance often changes significantly. Meanwhile, there are many different people in public spaces wearing similar clothes, making distinguishing them difficult. Most recent ReID models therefore employ deep convolutional neural networks (CNNs) to learn a feature embedding space that is robust against the appearance changes as well as discriminative against impostors (different identities but of similar appearance) [2, 3, 7, 10, 18, 24, 27, 30, 33, 39]. As the performance of state-of-the-art ReID models on public benchmarks approaches saturation, more realistic real-world ReID challenges are being considered. For example, instead of using manually cropped person images, recent ReID benchmarks all provide person images produced by off-the-shelf person detectors. The open-world ReID problem has also started to attract attention [38, 19, 1], whereby a small gallery set is matched against a much larger probe set.

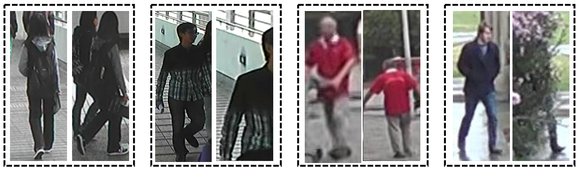


Figure 1: Outlying samples in existing ReID benchmarks. Each pair of images contain the same identity, with the left being an inlier and right an outlier.

However, one important ReID challenge has largely been ignored, that is, how to learn a robust ReID model with noisy training data. There are two types of data noise in practice. The first type is label noise, i.e., people assigned with the wrong identities. Label noise is caused by human errors [8]: matching people across camera views in the presence of impostors is a hard job even for humans who have short attention spans, and mistakes are made. More subtly, outliers provide a second source of noise. These are samples that have the correct identity labels, but are visually outlying due to either imperfect person detection or occlusion, as illustrated in Fig. 1. These outlying samples are found to be prevalent in existing benchmarks. Having both types of noisy samples in a training set inevitably has a detrimental effect on the learned feature embedding: Noisy samples are often far from inliers of the same class in the input (image) space. To minimise intra-class distance and pull the noisy samples close to their class centre, a ReID model often needs to sacrifice inter-class separability, leading to performance degradation (see Fig. 2(a) for an illustration).

In this paper, we propose a novel ReID model termed DistributionNet to deal with both types of noisy samples. With DistributionNet, each image is represented by a feature distribution, rather than a feature vector as in conventional deep models. Specifically, each image is now repre-

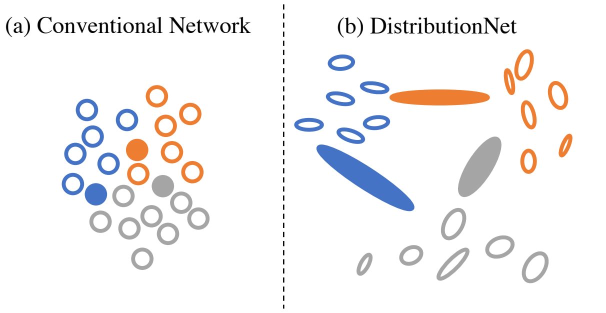


Figure 2: Illustrative comparison of the embedding space learned by (a) a conventional deep ReID model and (b) our DistributionNet in the presence of noisy training samples. Circle/ellipse colours denote class labels. Solid circles/ellipses indicate noisy samples.

sented as a Gaussian distribution. The mean of the distribution acts like the normal feature vector for ReID matching whilst the variance measures feature uncertainty. That is, given a noisy training sample, instead of forcing it to be closer to other inliers of the same class, DistributionNet computes a large variance, indicating that it is uncertain about what feature values should be assigned to the sample. Training samples of larger variances have less impact on the learned feature embedding space. This extra dimension thus allows the model to focus more on the clean inliers rather than overfitting to noisy samples, resulting in better class separability as illustrated in Fig. 2(b), and better generalisation to test data.

The inference and training of a deep CNN that represents an image as a feature distribution is nontrivial. In this work, we consider a feature vector as a random variable. Instead of delivering a point estimate using a CNN, we assume that the vector is drawn from a Gaussian distribution. From the distribution, we randomly sample feature vectors to compute training losses together with the distribution mean. This random sampling process prevents conventional end-to-end training. Therefore, a reparameterisation trick is introduced to enable our DistributionNet to be trainable using any off-the-shelf CNN optimiser. In addition to a supervised identity classification loss, we ensure that DistributionNet models uncertainty and allocates it appropriately by introducing losses to promote high net uncertainty. Together with the supervised loss, they help the model identify noisy training samples and discount them by assigning large variances.

Our contributions are as follows: (1) For the first time, the problem of learning ReID models robust against both label noise and outlying samples is identified, and a unified solution is provided. (2) A novel deep ReID model termed DistributionNet is proposed which uniquely models each learned deep feature as a distribution to account for feature uncertainty and alleviate the impact of noisy samples. Extensive comparative evaluations demonstrate the superiority of the proposed model over existing models on four benchmarks including Market-1501 [36], DukeMTMC-ReID [23], CUHK01 [16], and CUHK03 [17]. We show that the proposed model is particularly effective given a large amount of label noise or under the more challenging open-world ReID setting.

2. Related Work

Deep Person ReID Models Existing deep ReID models [2, 3, 7, 10, 18, 24, 27, 30, 33, 39] typically adopt an off-the-shelf CNN architecture for deep feature learning, with recent methods mostly using ResNet [6]. To overcome misalignment and pose variations, several works use pose detectors to identify body parts on which part-specific features are learned [34, 25, 31, 32]. By contrast, [15, 18] learn spatial transformer networks to automatically localise body parts and [35] constructs a self-attention layer to produce spatial attention maps, which highlight potential body parts. Hard attention based on reinforcement learning is also attempted [13]. All these efforts have the potential to cope with some of the outlying samples caused by imperfect person detection shown in Fig. 1. However, none of the existing deep ReID models is able to provide a principled solution to identifying different outliers in order to reduce their impact. Furthermore, the label noise problem has never been addressed, to the best of our knowledge.

Most existing works treat ReID as a closed-world retrieval problem. That is, the gallery set and the probe set contain exactly the same set of person identities. In practice, however, there is little demand in matching everyone present in a public space. Instead, there is typically a watchlist containing a small number of targets, e.g., fugitives/crime suspects. Using the watchlist as the gallery, the probe set includes everyone observed by a camera network, thus being much bigger than the gallery set. This open-world ReID problem is first defined in [37]. Transfer learning frameworks for mining discriminant information from non-target people are proposed to solve the problem [37, 38]. More recently, [19] uses adversarial learning in a deep ReID model to synthesise impostors for the gallery, in order to make the model less prone to attack from real impostors in the probe set. Our DistributionNet does not require a Generative Adversarial Network (GAN) that is tricky to train; and is particularly effective in the open-world setting, beating [19] by a clear margin (see Sec. 4.3).

Robust Deep Learning with Label Noise Although the problem of label noise has never been considered in ReID, it has been studied extensively in machine learning [4]. Existing robust deep learning approaches can be grouped into two categories depending whether human supervision/verification of noise is required. In the first category, no such additional human noise annotation or pattern estimation is needed. These methods address label noise by either iterative label correction via bootstrapping [22], adding additional layers on top of a classification layer to estimate the noise pattern [26, 5], or loss correlation [21]. The second category of methods requires a subset of noisy data to be re-annotated (cleaned) by more reliable sources to verify which samples contain noise. This subset is then used as seed/reference set so that noisy samples in the full training set can be identified. The recently proposed CleanNet [14] learns the similarity between class- and query-embedding vectors, which is then used to detect noisy samples. MentorNet [9] on the other hand resorts to curriculum learning and knowledge distillation to focus on samples whose labels are more likely to be correct.

Compared to the existing robust deep learning approaches, our DistributionNet gains noise robustness by modelling feature uncertainty, which is a completely different approach. It is also more generally applicable. Specifically, it belongs to the first category and thus does not need any additional noise verification as in [14, 9]. Unlike [26, 5, 21], it does not assume any noise pattern. Importantly, it does not assume the noisy samples must be caused by label flipping (assigning to wrong labels); it can thus also handle the outlying samples in Fig. 1. Our experiments show that DistributionNet outperforms representative existing models from both categories (see Sec. 4.2 and Sec. 4.3).

Feature Distribution Modelling In most studies, the high-level representation of an image is modelled as a fixed-length feature vector. However, for other data types, it is possible to model an instance's feature as a distribution. E.g., [28] proposes to represent a *video*, consisting of multiple key frames, as a Gaussian distribution, where the mean/covariance is the empirical statistics of those frames, with each modelled as a vector. Besides, it is intuitive to model a *class centre* as a distribution, as a class typically has many members. Based on this motivation, [29] presents a reformulation of the widely used cross-entropy loss. On the other hand, in many generative models, a single image's feature is often modelled as a distribution, e.g., in variational autoencoders [12], for the ease of placing prior for unconditional image generation. This has been extended for metric learning by disentangling intra-class variance and invariance [20]. In our work, we do not build a generative model, but still model a single image's feature as a distribution. Importantly, we deal with the noisy sample robustness problem, a completely different problem to [28, 29, 20].

3. Methodology

Problem Definition Two types of noisy training samples in ReID are considered. The first is the ***outlier***, where poor person detection and/or severe occlusion mean that assigning to the image any identity label would be harmful for learning a ReID model. The second is ***label noise***, which can be divided into two sub-types. One is *random noise*, where a random person's identity is randomly assigned. And the other is *patterned noise*, where the wrongly assigned identity corresponds to a person of similar visual appearance. Given a training set containing an unknown amount of both or either type of noisy samples, the objective of our deep ReID model is to learn a feature embedding space where people of different identities are well separable. Note that we do not make any assumption on the noise type (label noise or outlier), or its percentage, nor do we require any additional annotation on a subset to enable estimation of the noise pattern. Given such a space, during testing, both gallery set images and a probe image will be represented in the space, where their distance is used to measure their visual similarity for matching.

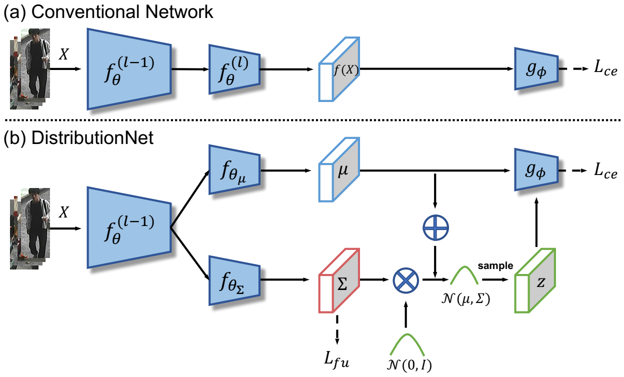


Figure 3: (a) A conventional ReID model trained for identity classification. (b) The proposed DistributionNet.

Feature Distribution Modelling As shown in Fig. 3(a), when a conventional deep neural network model is applied to person ReID, there are normally two modules: feature extractor $f_\theta(\cdot)$ and person identity classifier $g_\phi(\cdot)$. Given the i-th input image $\boldsymbol{X}^{(i)}$ and its one-hot encoding label $\boldsymbol{y}^{(i)}$, the model is usually trained by minimising the cross-entropy loss[1] between $\boldsymbol{y}^{(i)}$ and $g_\phi(f_\theta(\boldsymbol{X}^{(i)}))$. In the test stage, the output of $f_\theta(\cdot)$, i.e., the so-called feature vector, is used for distance calculation.

In contrast, our DistributionNet (see Fig. 3(b)) models the variance of the feature vector produced by a feature embedding network, as a measure of uncertainty. As a result, what our neural network delivers is no longer a feature vector (as a fixed point estimation) but a distribution over that vector, parameterised by mean and variance. Specifically, we propose to *explicitly* model the feature distribution of an *individual image* as Gaussian. From the probabilistic perspective, this means that we think of the feature vector as a random variable. That is, we assume the feature vector of

[1] Note that other training objectives such as triplet ranking can also be used in a deep ReID model, though identity classification has dominated recent models. They can be added readily in our framework.

the i-th image $f_\theta(\boldsymbol{X}^{(i)})$ is drawn from a Gaussian distribution parametrised by a mean vector $\boldsymbol{\mu}^{(i)}$ and a covariance matrix $\boldsymbol{\Sigma}^{(i)}$, which are produced by a neural network.

Network Architecture As shown in Fig. 3(b), in order to generate $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ from given input images $\boldsymbol{X}$, we split the network at the penultimate feature extraction layer and build two separate branches for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ respectively. Concretely, assume the conventional feature extraction module $f_\theta(\boldsymbol{X})$ can be decomposed as $f_\theta(\boldsymbol{X}) = f_\theta^{(l)}(f_\theta^{(l-1)}(\boldsymbol{X}))$. At the layer indexed by $(l-1)$, i.e., $f_\theta^{(l-1)}$, we drop its successive layer, i.e., $f_\theta^{(l)}$, and link it to two newly introduced layers: $f_{\theta_\mu}(f_\theta^{(l-1)}(\boldsymbol{X})) \rightarrow \boldsymbol{\mu}$ and $f_{\theta_\Sigma}(f_\theta^{(l-1)}(\boldsymbol{X})) \rightarrow \boldsymbol{\Sigma}$. We can think of f_{θ_μ} as a drop-in replacement of $f_\theta^{(l)}$ and f_{θ_Σ} produces a measure of uncertainty of $f_\theta^{(l)}$. Modelling a full covariance matrix is prohibitively expensive, so we constrain it to be diagonal. Therefore, f_{θ_Σ} produces a vector of the same size as f_{θ_μ}.

Classification Loss Conventionally, the output of $f_\theta(\boldsymbol{X})$ will be fed into a classifier $g_\phi(\cdot)$, and the cross-entropy loss $\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \sum_{i=1}^{c} \boldsymbol{y}_i \log \hat{\boldsymbol{y}}_i$, where c is the class number, will be computed as

$$L_{ce} = \ell(g_\phi(f_\theta(\boldsymbol{X})), \boldsymbol{y}). \tag{1}$$

With DistributionNet, Eq. 1 becomes

$$L_{ce} = \ell(g_\phi(\boldsymbol{\mu}), \boldsymbol{y}) + \lambda(\frac{1}{N}\sum_{j=1}^{N} \ell(g_\phi(\boldsymbol{z}^{(j)}), \boldsymbol{y})). \tag{2}$$

where $\boldsymbol{z}^{(j)} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Thus, we feed two kinds of inputs to the classifier: (i) the mean vector $\boldsymbol{\mu}$, which serves as a direct replacement of $f_\theta(\boldsymbol{X})$ and (ii) N random samples drawn from the Gaussian parametrised by $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. λ is the weight for the sampled feature vectors and is set to 0.1. Without the sampling part, Eq. 2 is reduced to Eq. 1.

Note that large variance leads to drastically different features $\boldsymbol{z}^{(j)}$ with the same label as $\boldsymbol{X}$. This leads to a large loss in the second term of Eq. 2. Therefore, in optimisation, the classification loss has an incentive to reduce the variance of the training samples. Indeed, as training progresses, the variance $\boldsymbol{\Sigma}$ always decreases with this loss alone. Thus training Eq. 2 alone will eventually revert DistributionNet back to the conventional model. So we add another loss to ensure that the variance is maintained.

Feature Uncertainty Loss To prevent the trivial solution of variance decreasing to zero, we add a feature uncertainty loss to encourage the model to maintain the uncertainty level about the training samples as a whole. To this end, we first use entropy to measure the uncertainty level of an individual training sample given its variance $\boldsymbol{\Sigma}$. The entropy of any multivariate Gaussian distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is:

$$q = \frac{1}{2}\log(\det(2\pi e\boldsymbol{\Sigma})). \tag{3}$$

Large variance leads to large entropy. Recall that our learned feature distribution is a diagonal multivariate normal distribution. So the above equation for the i-th input image is equivalent to

$$\begin{aligned} \boldsymbol{q}^{(i)} &= \frac{1}{2}\log\prod_{k}^{m}(2\pi e * \mathrm{diag}(\boldsymbol{\Sigma}^{(i)})_k) \\ &= \frac{1}{2}\sum_{k}^{m}\log(2\pi e * \mathrm{diag}(\boldsymbol{\Sigma}^{(i)})_k) \\ &= \frac{m}{2}(\log 2\pi + 1) + \frac{1}{2}\sum_{k}^{m}\log(\mathrm{diag}(\boldsymbol{\Sigma}^{(i)})_k) \end{aligned} \tag{4}$$

where $\mathrm{diag}(\cdot)$ means the diagonal vector of the input matrix, m is the total feature dimension, and k indexes each dimension. The feature uncertainty loss is then formulated as,

$$L_{fu} = \max(0, \gamma - \sum_{i=1}^{n}\boldsymbol{q}^{(i)}), \tag{5}$$

where n is mini-batch size, i indexes images in the batch, and γ is a margin to bound of the total uncertainty. Clearly, with L_{fu}, the model prefers to maintain the total uncertainty level/variance of the training samples. As the clean samples always have smaller variance caused by the classification loss, variance of noisy samples are expected to be larger to hold the total uncertainty of all samples.

Reparameterisation Trick When using the random sample $\boldsymbol{z}$ to train g_ϕ, a problem arises: the error will not propagate back to the preceding layers due to the nature of it being a random sample. In order to make those layers benefit from the random samples as well, we use a reparameterisation trick during sampling. Concretely, instead of drawing a sample from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ directly, we first draw a sample $\boldsymbol{\varepsilon}$ from a standard Gaussian with zero mean and unit covariance, i.e., $\boldsymbol{\varepsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and then we get the sample by computing $\boldsymbol{\mu} + \boldsymbol{\varepsilon}\boldsymbol{\Sigma}$. By doing so, we split the random part and the trainable part in the sample, so the gradient can be passed through the trainable part.

Discussion With the losses, DistributionNet exhibits two behaviours: (1) It gives large variances to noisy samples (either with wrong labels or outlying) and small variances to clean inliers. (2) Training samples with larger variances contribute less to learning the feature embedding space. By combining the two, the model in effect focuses mostly on the clean inliers to learn a better embedding space for ReID. Next, we explain why DistributionNet behaves in these two ways.

Why large variances for noisy samples? First, we need to understand what the supervised classification loss L_{ce} wants: we mentioned earlier that samples with large variances will lead to large loss values of L_{ce}; it is also noted that samples with wrong labels or outlying also have the same effect because they are normally far away from the class centres and the clean inliers. Second, we explained that with the feature uncertainty loss L_{fu}, the model cannot simply satisfy L_{ce} by reducing the variance of every sample to zero – the overall variance/uncertainty level has to be maintained. So who will get the large variance? Now the decision is clear: reducing variances of noisy samples would still lead to large L_{ce}, whilst reducing those of clean inliers will have a direct impact of reducing L_{ce}; the model therefore allocates large variance to noisy samples.

Why samples with larger variance contribute less for model training? The reason is intuitive – if an image embedding has a large variance, when it is sampled, the outcome $\boldsymbol{z}$ will be far away from its original point (the mean vector $\boldsymbol{\mu}$) but with the same class label. So when several diverse $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, ..., \boldsymbol{z}^{(N)}$ and $\boldsymbol{\mu}$ are fed to the classifier, it is likely that their gradients will cancel each other out. On the other hand, when a sample has a small variance, all $\boldsymbol{z}^{(j)}$ will be close to $\boldsymbol{\mu}$; feeding these to the classifier gives consistent gradients thus reinforcing the importance of the sample. The variance/uncertainty thus provides a mechanism for DistributionNet to give more/less importance to different training samples. Since noisy samples are given large variance, their contribution to model training is reduced, resulting in a better feature embedding space (see Fig. 4 for an example).

4. Experiments

4.1. Baselines and Implementation Details

We adopt ResNet50 [6] as our backbone feature extractor as it is used by most recent ReID models. DistributionNet is compared against four baselines. Unless otherwise stated, for a fair comparison, ResNet50 with the same hyperparameters is used as the backbone in all baselines, and the training steps are the same as our DistributionNet. **ResNet-baseline** is the main baseline. Our model introduces a few more parameters in order to predict the variance. To ensure a fair comparison, we add a parallel layer to the penultimate (feature) layer in the baseline. The input of the added layer is the same as the feature layer, and its output is element-wise added along with the output of the feature layer to get the final feature, which is then fed to a fully connected layer for computing the classification loss. This way, our DistributionNet and the ResNet-baseline have an identical number of parameters. **Bootstrap_hard** and **Bootstrap_soft** are introduced in [22]. Both iteratively use the model-predicted labels to refine the original labels that are potentially corrupted by noise. They are as generally applicable as ours because no assumption on the noise distribution is made. **CleanNet** [14] achieves state-of-the-art results on a number of visual recognition tasks. Different from all other compared methods, CleanNet requires a subset of the training set to be 'cleaned' manually, that is, verified regarding which images contain noise. This thus gives CleanNet an unfair advantage over other compared models. In our experiments, 10% of the training set is used as a clean reference set to train CleanNet using the author-provided code. After training, 20% of the whole training set deemed most likely to be noisy are removed before the final ReID model is trained on the remainder. This is again different from all other models which do not require explicit noisy sample removal.

The training has two steps: (1) Using a ResNet50 pretrained on ImageNet, we fine-tune it using the given ReID training set (for training identity classification) for $60,000$ steps with a batch size of 32. The ADAM optimiser [11] is used with learning rate 3.5×10^{-3} and the default momentum terms: $\beta_1 = 0.9$ and $\beta_2 = 0.999$. (2) We initialise the parameters of DistributionNet with the trained model in step (1), and only train the last ResNet50 block unit and variance generating modules for another $20,000$ steps with the same batch size but lower learning rate 5×10^{-4}. Weight of feature uncertainty loss is 0.001.

4.2. Experiments with Noisy Labels

Existing ReID benchmarks contain many outlying samples (see Fig. 1), yet the identity labels are clean. To simulate real-world large-scale ReID datasets (annotated by real imperfect workers), we additionally introduce label noise in this experiment.

Datasets and Settings Four large-scale ReID datasets are used, including Market-1501 [36], DukeMTMC-ReID [23], CUHK01 [16], and CUHK03 [17]. We adopt the standard training/test splits provided by the datasets (see Supplementary Material). Note that, following these standard splits, the testing gallery set and probe set contain the same number of identities, i.e., this is a closed-world setting. Two types of noise are considered. For random noise, a certain percentage of training images are randomly selected, and their identity labels are then randomly assigned to wrong ones. For patterned noise, we use a ResNet50 trained on the clean data to obtain the feature of each training sample and search for the most visually similar samples using Euclidean distance. Then for the randomly selected training samples, their identity labels are assigned to that of the most similar sample that has a different identity. For both noise types, three percentages are considered: 10%, 20%, and 50%. For each noise percentage, due to the randomness in sample selection and label assignment, 5 runs are carried out. The final result reported is the average of the 5 runs.

Results The comparative results of random noise on the four datasets are shown in Tab. 1 and Tab. 2. Patterned noise results are shown in Tab. 3 and Tab. 4. The following observations can be made: (1) Our DistributionNet achieves the best results among all compared models. In most cases, the

Dataset		Market-1501				DukeMTMC-ReID			
noise		mAP	Rank1	Rank5	Rank10	mAP	Rank1	Rank5	Rank10
10%	B	55.50	79.39	91.87	94.96	42.60	63.78	78.82	83.55
	H	57.28	80.79	92.20	94.97	42.62	64.54	78.73	83.53
	S	55.37	79.77	91.69	94.91	41.84	62.79	78.14	83.53
	C	59.14	81.41	91.99	94.82	47.88	68.09	80.61	**85.77**
	D	**61.47**	**82.31**	**93.13**	**95.76**	**47.99**	**68.61**	**81.87**	**86.09**
20%	B	45.36	71.68	87.79	91.68	34.94	56.73	73.75	79.58
	H	46.03	72.71	87.31	91.33	34.23	55.66	71.63	78.73
	S	45.49	71.46	86.89	91.42	34.13	55.52	71.27	77.56
	C	44.22	71.40	87.28	91.93	33.98	55.07	71.72	76.48
	D	**53.40**	**77.03**	**90.60**	**94.01**	**40.87**	**62.39**	**77.38**	**82.49**
50%	B	28.01	55.14	75.75	82.56	18.83	37.47	54.67	61.69
	H	28.22	54.87	76.20	83.11	19.88	38.87	56.69	63.91
	S	27.78	55.17	74.98	82.42	19.27	37.70	55.79	63.02
	C	26.08	52.73	72.89	79.98	19.01	38.96	55.52	62.21
	D	**35.14**	**61.08**	**81.05**	**87.07**	**25.82**	**45.98**	**63.91**	**70.87**

Table 1: Results on Market-1501 and DukeMTMC-ReID with random noise. Model abbreviations: **B**: ResNet-Baseline, **H**: Bootstrap_hard [22], **S**: Bootstrap_soft [22], **C**: CleanNet [14], **D**: DistributionNet.

Dataset		CUHK01				CUHK03			
noise		mAP	Rank1	Rank5	Rank10	mAP	Rank1	Rank5	Rank10
10%	B	52.02	84.61	93.68	95.40	24.57	25.17	42.43	52.17
	H	55.35	87.22	92.78	95.88	23.99	24.00	42.43	51.36
	S	54.53	87.42	94.23	96.91	24.48	25.14	41.93	51.86
	C	48.69	81.44	91.55	94.02	23.28	23.14	40.21	50.71
	D	**63.36**	**89.85**	**96.25**	**98.02**	**31.80**	**32.29**	**51.81**	**61.67**
20%	B	43.22	76.91	87.56	90.72	16.45	16.25	31.50	40.54
	H	45.62	80.00	90.52	93.20	16.65	16.79	32.00	40.79
	S	48.94	83.09	91.96	94.02	16.50	17.57	31.71	41.93
	C	40.15	75.88	86.60	91.13	12.92	12.14	25.93	34.00
	D	**58.07**	**87.30**	**94.60**	**96.58**	**24.20**	**24.33**	**43.09**	**53.14**
50%	B	35.55	71.22	83.30	86.84	6.44	6.05	14.37	20.67
	H	34.77	69.07	82.06	86.60	7.22	6.79	16.86	23.21
	S	35.65	70.31	82.68	86.19	6.48	5.71	14.57	20.86
	C	34.44	70.10	81.86	85.15	5.17	4.86	11.50	16.07
	D	**44.83**	**78.89**	**89.32**	**92.08**	**10.61**	**10.14**	**21.77**	**29.86**

Table 2: Results on CUHK01 and CUHK03 with random noise.

Dataset		Market-1501				DukeMTMC-ReID			
noise		mAP	Rank1	Rank5	Rank10	mAP	Rank1	Rank5	Rank10
10%	B	25.87	51.46	70.21	76.81	18.26	36.10	51.87	58.62
	H	25.76	5108	70.11	77.06	18.74	36.57	51.93	58.59
	S	25.50	50.47	69.43	76.32	18.01	35.68	51.23	58.03
	C	26.64	52.47	70.90	77.41	18.80	36.28	51.71	58.22
	D	**27.04**	**52.40**	**71.20**	**77.67**	**20.74**	**37.69**	**53.21**	**60.01**
20%	B	23.49	48.44	67.85	74.67	16.96	33.71	49.15	56.01
	H	23.40	48.25	67.34	74.40	16.93	33.76	49.38	56.40
	S	24.08	49.51	68.77	75.62	16.83	33.13	49.21	56.05
	C	24.28	49.54	68.35	75.33	17.03	33.79	48.81	55.61
	D	**24.41**	**49.25**	**68.31**	**75.42**	**18.49**	**34.48**	**50.22**	**57.08**
50%	B	20.74	44.04	64.32	72.27	14.17	29.59	45.50	52.45
	H	19.87	42.87	63.37	71.40	14.32	30.00	45.58	52.55
	S	20.72	43.86	64.49	72.55	13.65	28.35	44.44	51.82
	C	19.90	43.01	63.13	71.34	13.70	28.46	44.25	51.19
	D	**21.42**	**44.84**	**64.77**	**72.72**	**15.95**	**30.75**	**46.95**	**53.58**

Table 3: Results on Market-1501 and DukeMTMC-ReID with patterned noise.

Dataset		CUHK01				CUHK03			
noise		mAP	Rank1	Rank5	Rank10	mAP	Rank1	Rank5	Rank10
10%	B	42.87	80.33	90.52	93.40	10.10	9.82	19.16	25.98
	H	44.66	81.73	90.97	93.86	9.31	8.60	17.93	24.66
	S	44.20	80.95	90.10	93.03	9.65	9.39	19.57	25.29
	C	39.70	78.14	87.88	90.72	8.17	7.64	16.00	21.39
	D	**52.72**	**86.60**	**94.35**	**96.41**	**10.87**	**10.48**	**20.11**	**26.81**
20%	B	41.86	79.38	89.40	92.49	8.71	8.17	17.36	23.44
	H	43.46	81.07	90.47	93.07	8.75	8.36	17.46	23.36
	S	42.97	80.41	90.43	93.07	8.42	7.77	16.54	23.07
	C	37.59	75.42	85.85	89.61	7.89	7.13	15.67	21.17
	D	**49.85**	**83.30**	**92.41**	**95.42**	**9.46**	**8.81**	**18.77**	**25.32**
50%	B	38.67	76.37	87.67	90.85	7.57	7.21	15.47	20.93
	H	39.30	77.49	87.84	91.38	7.67	6.98	15.80	21.06
	S	38.98	76.91	87.26	90.47	7.76	7.21	16.24	22.21
	C	36.21	73.28	84.37	88.45	6.30	5.72	12.96	18.13
	D	**45.43**	**81.11**	**90.23**	**93.40**	**8.18**	**7.61**	**16.50**	**22.74**

Table 4: Results on CUHK01 and CUHK03 with patterned noise.

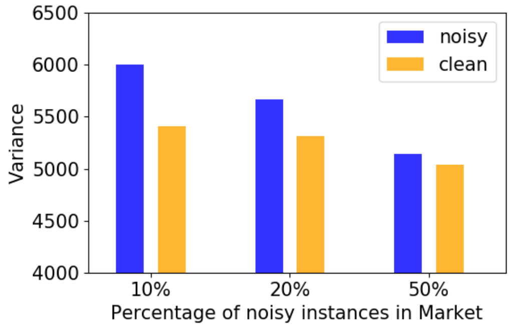
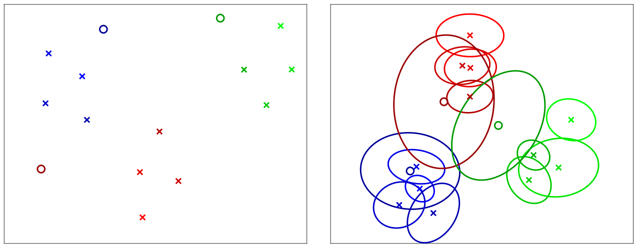


Figure 4: Comparison between variance ($\bar{\Sigma}$) of clean and label-noise data generated by DistributionNet.

margins over the baselines are significant. (2) Our model vs. ResNet-Baseline shows that modelling feature uncertainty brings clear and consistent improvements. (3) As expected, patterned noise is harder and the performance of every method on each dataset is lower than that with random noise. However, DistributionNet still obtains the best results. (4) The improvement margin over the baselines increases when the noise level is raised from 10% to 20%; however, when 50% noise is added, all models struggle and the gap becomes smaller. In practice, 50% noise is extreme, and around 10 to 20% noise levels are more realistic. (5) Among the three compared noise-robust deep models, even with additional annotation, CleanNet does not have a clear advantage over the much simpler Bootstrap_soft/hard, and sometimes even fails to beat the ResNet-Baseline.

How Noise Robustness is Achieved by DistributionNet As explained in Sec. 3, DistributionNet gains its robustness by allocating large variances to noisy samples, which subsequently reduces their impact on model training, leading to a better embedding space where different identity classes become more separable. Fig. 4 shows the average variance inferred for clean and noisy data in Market-1501. We can see that, indeed the variance of noisy data is larger than that of clean data on average. Why having large variance/uncertainty for noisy samples helps? As explained in Fig. 2 and Sec. 3, we assume that by introducing feature uncertainty modelling, DistributionNet is able to explain away noisy training samples by assigning them with large variances, rather than distorting the embedding space and sacrificing class separability, or overfitting to the training set. Fig. 5 compares the ResNet-baseline and DistributionNet embeddings for Market-1501. It is noted that the noisy (circle) data points are given larger variance by Distribution-



(a) ResNet-baseline (b) DistributionNet

Figure 5: t-SNE visualisation of feature distributions (ellipses). Images with noisy (circles) or clean (cross) labels are shown with different colours. 15 images from 3 identities are randomly selected from Market-1501 with 20% random label noise.

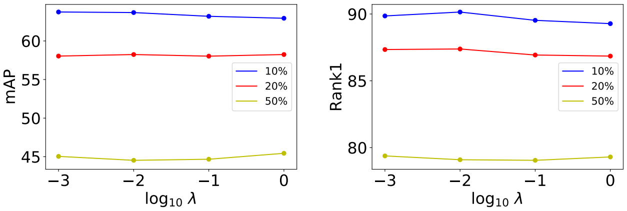


Figure 6: Evaluation with different values of λ in Eq. 2.

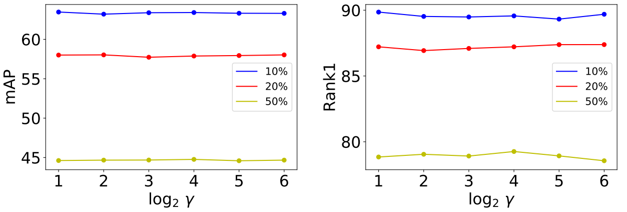
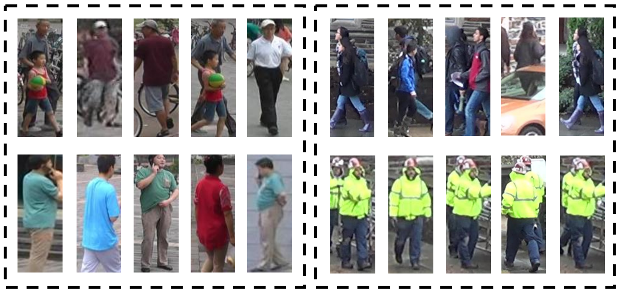


Figure 7: Evaluation with different values of γ in Eq. 5.

Net. Furthermore, different person identities become more separable with DistributionNet, which explains its superior performance.

Hyperparameter Analysis We first analyse the sensitivities of our model to two important hyperparameters, i.e., the weight of loss λ in Eq. 2 and the margin γ in Eq. 5. We use CUHK01 with random noise to carry out the analysis. By default, we vary the value of one parameter and keep the others fixed. For simplicity, we only illustrate mAP and Rank1 values for different hyperparameters.

In Fig. 6 and Fig. 7, we compare different values of λ in Eq. 2 and γ in Eq. 5 under the conditions of different noisy samples respectively. It is clearly shown that, our approach is impacted just marginally and significantly improves the baseline at all values of λ and γ. Therefore, it is safe to make the conclusion that our model is insensitive to λ and γ.



(a) Market-1501 (b) DukeMTMC-ReID

Figure 8: Examples of person images with the highest (first row) and lowest (second row) feature variance.

4.3. Experiments without Noisy Labels

In this set of experiments, no label noise is added to the four benchmarks. However, as mentioned earlier, there are still numerous noisy training samples caused by imperfect person detectors (partial body or large proportions of background) and occlusion (by static objects or other people in the scene). Their negative effect on the learned feature space, though not as severe as the noisy labels, should also be dealt with for effective ReID feature learning. Experiments are carried out under both the conventional closed-world setting and more practical open-world setting.

4.3.1 Closed-world ReID

Under this setting, the testing gallery and probe set contain the same number of identities; in other words, a probe image would always have a correct match in the gallery. This setting has been adopted by the majority of the published ReID work and it was also used in the noisy label experiments reported earlier. Tab. 6 compares our DistributionNet with three baselines used in the label noise experiments. Note that CleanNet [14] cannot be compared here because it requires a reference set where noisy samples are manually identified. Which image is an outlying sample is subjective; obtaining such a reference set is thus not straightforward. From Tab. 6, we can see that, again modelling feature uncertainty using DistributionNet brings a clear improvement (D vs. B). The margin ranges from 1.98% mAP for DukeMTMC-ReID to 9.64% for CUHK01. Interestingly both Bootstrap_hard and Bootstrap_soft can bring a moderate amount of improvement over the ResNet-baseline. This is despite that they assign outlying samples (identified as those whose predicted labels disagree with the original (correct) labels) to other labels. Overall these results show that our model is capable of dealing with both noise-labels and outliers. We show some examples of outliers (those with the largest variance) in Fig. 8. It is clear that, they are mostly caused by either poor person detection or occlusion. In contrast, images with the smallest variance mostly contain people of distinct clothing and were produced by perfect person

Dataset	Market-1501						CUHK01						CUHK03					
FTR	0.1%	1%	5%	10%	20%	30%	0.1%	1%	5%	10%	20%	30%	0.1%	1%	5%	10%	20%	30%
Set Verification																		
B	43.03	81.21	95.15	95.15	95.76	96.36	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	38.10	80.95	95.24	**100.00**	**100.00**	**100.00**
APN	43.85	82.31	**96.92**	**98.46**	99.23	**100.00**	55.56	55.56	55.56	66.67	77.78	77.78	**66.67**	78.57	92.86	95.24	95.24	95.24
H	50.30	86.67	96.36	96.97	**100.00**	**100.00**	88.89	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	38.10	76.19	**97.62**	**100.00**	**100.00**	**100.00**
S	38.18	**89.09**	95.76	96.97	99.39	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	50.00	73.81	95.24	95.24	97.62	**100.00**
D	**55.76**	87.88	95.76	96.97	98.18	98.18	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	54.76	**83.33**	95.24	97.62	**100.00**	**100.00**
Individual Verification																		
B	76.40	94.33	99.20	99.33	99.73	99.87	61.11	77.78	**88.89**	**94.44**	**100.00**	**100.00**	77.38	**94.05**	96.43	97.62	98.81	98.81
APN	84.00	96.72	98.69	99.58	99.58	99.58	44.44	61.11	77.78	77.78	83.33	88.89	79.54	**94.05**	95.24	95.24	97.15	97.15
H	81.58	97.54	99.47	99.87	**100.00**	**100.00**	**66.67**	77.78	**88.89**	**94.44**	94.44	94.44	79.76	88.10	96.43	98.81	**100.00**	**100.00**
S	80.86	97.19	**100.00**	**100.00**	**100.00**	**100.00**	61.11	72.22	**88.89**	**94.44**	94.44	**100.00**	77.38	90.48	**97.62**	97.62	97.62	98.81
D	**86.29**	**97.77**	**100.00**	**100.00**	**100.00**	**100.00**	61.11	**88.89**	**88.89**	**94.44**	**100.00**	**100.00**	**86.90**	**94.05**	**97.62**	**98.81**	98.81	98.81

Table 5: Open-world person ReID results. The numbers are TTR (%) against different FTR(%) values. Model abbreviations: **B**: Resnet-Baseline, **H**: Bootstrap_hard [22], **S**: Bootstrap_soft [22], **APN**: [19], **D**: DistributionNet.

	mAP	Rank1	Rank5	Rank10	mAP	Rank1	Rank5	Rank10
	Market-1501				DukeMTMC-ReID			
B	67.66	86.57	**95.77**	96.72	54.17	73.61	84.69	88.55
H	69.09	86.98	94.61	**96.90**	54.98	72.89	84.29	88.51
S	67.85	86.44	94.88	96.81	53.68	73.16	84.25	87.39
D	**70.82**	**87.26**	94.74	96.73	**55.98**	**74.73**	**85.05**	**88.82**
	CUHK01				CUHK03			
B	60.70	88.66	95.46	97.32	34.11	34.93	52.00	63.07
H	62.62	89.48	96.70	97.53	38.15	38.86	59.07	**68.86**
S	61.03	88.66	95.88	97.32	38.20	38.79	**59.93**	68.36
D	**70.70**	**94.23**	**97.53**	**98.56**	**38.47**	**39.36**	58.93	67.93

Table 6: Closed-world person ReID results. Model abbreviations: **B**: Resnet-Baseline, **H**: Bootstrap_hard [22], **S**: Bootstrap_soft [22], **D**: DistributionNet.

detection with no occlusion – the model is thus most confident about the computed features for them.

4.3.2 Open-world ReID

Settings In the open-world ReID setting, a small number of identities are used to form the targets, and the test gallery set only contains images of these target identities. For direct comparison with [19], we follow exactly the same setting: Market-1501, CUHK01, and CUHK03 are used with the same splits as in [19] (see Supplementary Material for details). For each dataset, two images of the targets form the gallery list. Half of the remaining images of targets and all the images of non-targets in training set form the training set. The other half of the targets' images and images of non-targets in the test set are used as the probe list. The key challenge for this setting is that the probe set contains many impostors which need to be rejected. We use true target rate (TTR) and false target rate (FTR) as evaluation metrics for both set-based and individual-based verification tasks as in [37, 38, 19]. The definition of these metrics and tasks are given in Supplementary Material.

Baselines The three baselines used in the closed-world setting are again compared here. In addition, we add the state-of-the-art open-world ReID model APN in [19]. The results reported in [19] were obtained under exactly the same setting[2]. In addition, their model also has a ResNet50 backbone. The comparison is thus fair.

Results Comparative results are shown in Tab. 5. We observe that: (1) Overall, our DistributionNet outperforms all baselines under both set and individual verification tasks. The improvement is particularly large at smaller FTR values which are more important in practice. (2) It is impressive that our model can beat the state-of-the-art APN model under most settings, sometimes by significant margins. Note that APN takes a two-stepped approach and in the first step a GAN model is trained to synthesise more training samples. Our one-step model is simpler yet more effective thanks to its ability to model feature uncertainty. (3) Compared with Tab. 6, it is apparent that the advantages of DistributionNet over the baselines are more pronounced under the more challenging yet realistic open-set setting. This is expected – when different person identities become inseparable in the learned feature space using the baselines, its negative impact under the open-set setting is more tangible. For instance, if even a single gallery identity gets overlapped with other identities in the probe, this will result in a large drop in the matching performance using the two metrics.

5. Conclusion

In this work, for the first time, we addressed both the noisy label and outlying sample problems in learning a deep ReID model. A unified solution to cope with both types of noisy samples was proposed. The key idea was to model feature uncertainty explicitly by modelling each feature as a distribution. The resulting DistributionNet is able to mitigate the negative impact of the noisy samples by assigning large variance to them. Extensive experiments were conducted to validate the effectiveness of DistributionNet. It was shown to outperform a number of state-of-the-art competitors in various settings.

[2] Note that the latest ArXiv version of this paper reported higher results than their conference proceeding version. The higher/newer results are thus included in our comparison.

References

[1] Brais Cancela, Timothy M Hospedales, and Shaogang Gong. Open-world person re-identification by multi-label assignment inference. In *BMVC*, 2014. 1

[2] Xiaobin Chang, Timothy M Hospedales, and Tao Xiang. Multi-level factorisation net for person re-identification. In *CVPR*, 2018. 1, 2

[3] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *CVPR*, 2017. 1, 2

[4] Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2014. 2

[5] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017. 3

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 5

[7] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737, 2017. 1, 2

[8] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *ACM SIGKDD workshop on human computation*. ACM, 2010. 1

[9] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *ICML*, 2018. 3

[10] Mahdi M. Kalayeh, Emrah Basaran, Muhittin Gökmen, Mustafa E. Kamasak, and Mubarak Shah. Human semantic parsing for person re-identification. In *CVPR*, 2018. 1, 2

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014. 5

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2013. 3

[13] X. Lan, H. Wang, S. Gong, and X. Zhu. Deep reinforcement learning attention selection for person re-identification. In *BMVC*, 2017. 2

[14] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *ICCV*, 2017. 3, 5, 6, 7

[15] Dangwei Li, Xiaotang Chen, Zhang Zhang, and Kaiqi Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *CVPR*, 2017. 2

[16] Wei Li, Rui Zhao, and Xiaogang Wang. Human reidentification with transferred metric learning. In *ACCV*, 2012. 2, 5

[17] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014. 2, 5

[18] Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In *CVPR*, 2018. 1, 2

[19] Xiang Li, Ancong Wu, and Wei-Shi Zheng. Adversarial open-world person re-identification. In *ECCV*, 2018. 1, 2, 8

[20] Xudong Lin, Yueqi Duan, Qiyuan Dong, Jiwen Lu, and Jie Zhou. Deep variational metric learning. In *ECCV*, 2018. 3

[21] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017. 3

[22] Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *CoRR*, abs/1412.6596, 2014. 3, 5, 6, 8

[23] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *ECCV workshop on Benchmarking Multi-Target Tracking*, 2016. 2, 5

[24] Chunfeng Song, Yan Huang, Wanli Ouyang, and Liang Wang. Mask-guided contrastive attention model for person re-identification. In *CVPR*, 2018. 1, 2

[25] Chi Su, Jianing Li, Shiliang Zhang, Junliang Xing, Wen Gao, and Qi Tian. Pose-driven deep convolutional model for person re-identification. In *ICCV*, 2017. 2

[26] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. In *ICLR*, 2015. 3

[27] Maoqing Tian, Shuai Yi, Hongsheng Li, Shihua Li, Xuesen Zhang, Jianping Shi, Junjie Yan, and Xiaogang Wang. Eliminating background-bias for robust person re-identification. In *CVPR*, 2018. 1, 2

[28] Christos Tzelepis, Vasileios Mezaris, and Ioannis Patras. Linear maximum margin classifier for learning from uncertain data. *CoRR*, abs/1504.03892, 2015. 3

[29] Weitao Wan, Yuanyi Zhong, Tianpeng Li, and Jiansheng Chen. Rethinking feature distribution for loss functions in image classification. *CoRR*, abs/1803.02988, 2018. 3

[30] Yan Wang, Lequn Wang, Yurong You, Xu Zou, Vincent Chen, Serena Li, Gao Huang, Bharath Hariharan, and Kilian Q Weinberger. Resource aware person re-identification across multiple resolutions. In *CVPR*, 2018. 1, 2

[31] Longhui Wei, Shiliang Zhang, Hantao Yao, Wen Gao, and Qi Tian. Glad: global-local-alignment descriptor for pedestrian retrieval. In *ACM MM*, 2017. 2

[32] Jing Xu, Rui Zhao, Feng Zhu, Huaming Wang, and Wanli Ouyang. Attention-aware compositional network for person re-identification. In *CVPR*, 2018. 2

[33] Qian Yu, Xiaobin Chang, Yi-Zhe Song, Tao Xiang, and Timothy M. Hospedales. The devil is in the middle: Exploiting mid-level representations for cross-domain instance matching. *CoRR*, abs/1711.08106, 2017. 1, 2

[34] Haiyu Zhao, Maoqing Tian, Shuyang Sun, Jing Shao, Junjie Yan, Shuai Yi, Xiaogang Wang, and Xiaoou Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*, 2017. 2

[35] Liming Zhao, Xi Li, Yueting Zhuang, and Jingdong Wang. Deeply-learned part-aligned representations for person re-identification. In *ICCV*, 2017. 2

[36] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015. 2, 5

[37] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Transfer re-identification: From person to set-based verification. In *CVPR*, 2012. 2, 8

[38] Wei-Shi Zheng, Shaogang Gong, and Tao Xiang. Towards open-world person re-identification by one-shot group-based verification. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):591–606, 2016. 1, 2, 8

[39] Zhedong Zheng, Liang Zheng, and Yi Yang. A discriminatively learned cnn embedding for person reidentification. *TOMM*, 14, 2017. 1, 2