Ph.D. DISSERTATION

Learning Local Matching with Hard Sample Mining for Person Re-identification

영상 기반 동일인 판별을 위한 부분 정합 학습

BY

YUMIN SUH

FEBRUARY 2019

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Learning Local Matching with Hard Sample Mining for Person Re-identification

영상 기반 동일인 판별을 위한 부분 정합 학습

BY

YUMIN SUH

FEBRUARY 2019

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Learning Local Matching with Hard Sample Mining for Person Re-identification

영상 기반 동일인 판별을 위한 부분 정합 학습

지도교수 이 경 무

이 논문을 공학박사 학위논문으로 제출함

2019년 2월

서울대학교 대학원

전기 정보 공학부

서 유 민

서유민의 공학박사 학위 논문을 인준함

2019년 2월

위 원 장: ______________________________
부위원장: ______________________________
위 원: ______________________________
위 원: ______________________________
위 원: ______________________________

Abstract

Person re-identification is a problem of identifying the same individuals among the persons captured from different cameras. It is a challenging problem because the same person captured from non-overlapping cameras usually shows dramatic appearance change due to the viewpoint, pose, and illumination changes. Since it is an essential tool for many surveillance applications, various research directions have been explored; however, it is far from being solved.

The goal of this thesis is to solve person re-identification problem under the surveillance system. In particular, we focus on two critical components: designing 1) a better image representation model using human poses and 2) a better training method using hard sample mining. First, we propose a part-aligned representation model which represents an image as the bilinear pooling between appearance and part maps. Since the image similarity is independently calculated from the locations of body parts, it addresses the body part misalignment issue and effectively distinguishes different people by discriminating fine-grained local differences. Second, we propose a stochastic hard sample mining method that exploits class information to generate diverse and hard examples to use for training. It efficiently explores the training samples while avoiding stuck in a small subset of hard samples, thereby effectively training the model. Finally, we propose an integrated system that combines the two approaches, which is benefited from both components. Experimental results show that the proposed method works robustly on five datasets with diverse conditions and its potential extension to the more general conditions.

keywords: Deep metric learning, Person re-identification, Image retrieval, Hard sample mining

student number: 2014-30305

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Person re-identification is the problem to check and identify if a person seen from one camera occurs in the videos taken from different cameras. Due to its broad usage, it has been a very hot topic in computer vision for recent few years [75, 120, 114, 36, 115, 122, 21, 76, 77, 117]. It serves as a fundamental tool for various surveillance and security applications such as person search, person count, and multi-target tracking. However, distinguishing same person among a bunch of similar-looking candidates based on the only subtle differences under a varying viewpoint/pose/illumination is very challenging. Numerous approaches exist while focusing on different issues and topics with various tools including metric learning techniques [110, 111, 59, 41, 33, 39, 30], probabilistic patch matching algorithms [7, 6, 69], and graph matching [79, 78, 1]. To solve the problem, we believe it is essential to design an effective strategy to learn local matching and perform partwise appearance comparison. It motivates our study on learning part-based image representation for person re-identification.

Our system for the person re-identification is illustrated in Figure 1.1. We follow the two-step approach to first detect every person occurring in the videos and then identify person by comparing the query image and the detected candidates. Here, the key challenge is to learn a metric or an image embedding function that the similarity between images reflects the semantic part-wise similarity between persons. Therefore,

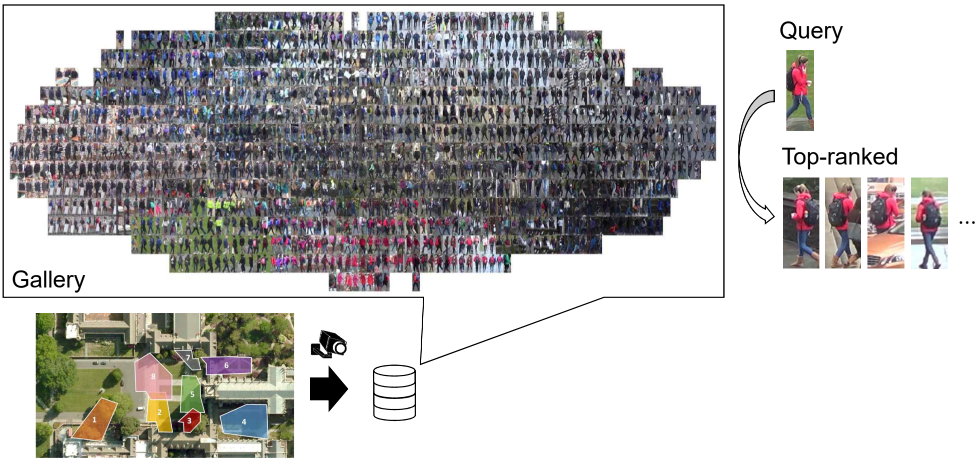


Figure 1.1: Person re-identification system. For a given query, images with the smallest distances are retrieved.

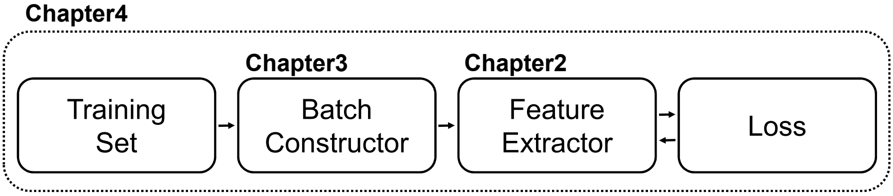


Figure 1.2: Overview of the training procedure

we formulate the person re-identification as a deep metric learning problem, which is to learn an embedding function that maps a detected bounding box to a metric space. We want the distances between the representations of similar persons to be small while distances between embeddings of dissimilar persons to be large. To this end, we explore two critical directions for better image representation learning; enhancing the embedding network and the batch constructor used for training. The overall procedure for training the image representation and the thesis organization is shown in Figure 1.2. In the following subsections, we summarize our approach and contributions.

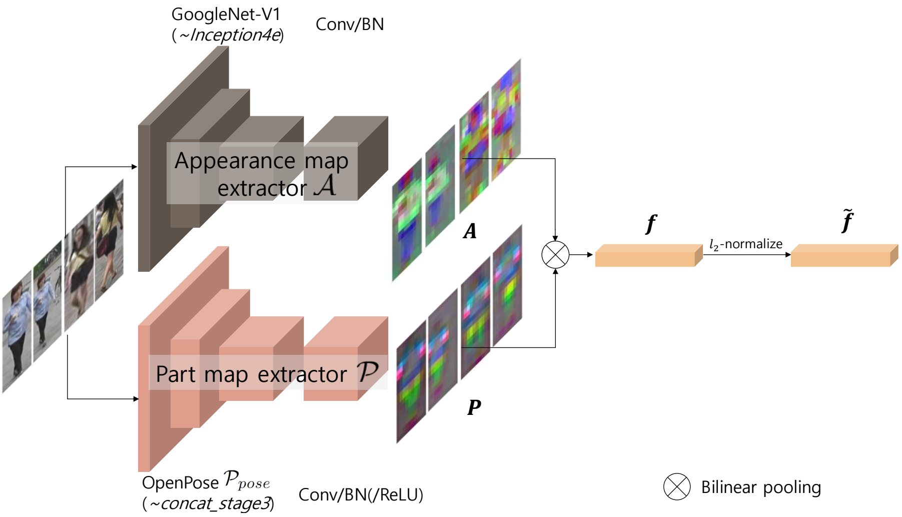


Figure 1.3: Network architecture of the proposed part-aligned feature extractor

1.1 Part-Aligned Bilinear Representations

In Chapter 2, part-aligned bilinear representations is proposed [80]. Comparing the appearance of corresponding body parts is essential for person re-identification. As body parts are frequently misaligned between the detected human boxes, an image representation that can handle this misalignment is required. In this paper, we propose a network that learns a part-aligned representation for person re-identification. Our model consists of a two-stream network, which generates appearance and body part feature maps respectively, and a bilinear-pooling layer that fuses two feature maps to an image descriptor. We show that it results in a compact descriptor, where the image matching similarity is equivalent to an aggregation of the local appearance similarities of the corresponding body parts. Since the image similarity does not depend on the relative positions of parts, our approach significantly reduces the part misalignment problem. Training the network does not require any part annotation on the person re-identification dataset. Instead, we simply initialize the part sub-stream using a pre-trained sub-network of an existing pose estimation network and train the whole network to minimize the re-identification loss. We validate the effectiveness

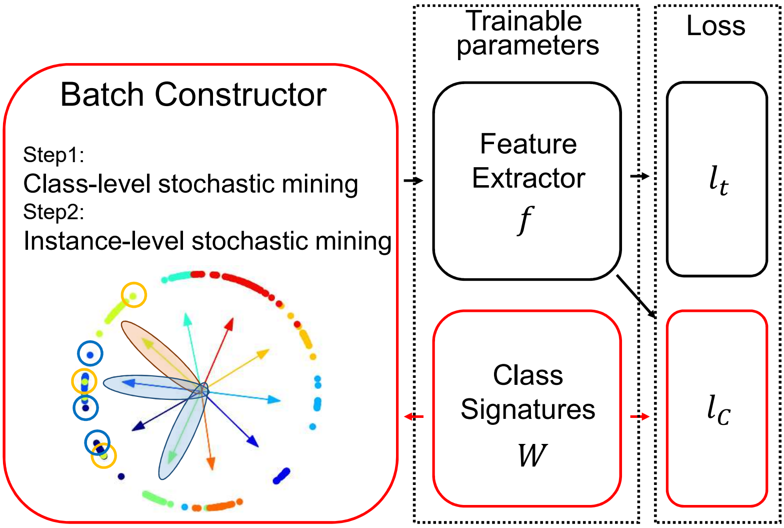


Figure 1.4: Overview of the training process with the proposed hard sample mining

of our approach by demonstrating its superiority over the state-of-the-art methods on the standard benchmark datasets including Market-1501, CUHK03, CUHK01 and DukeMTMC, and standard video dataset MARS.

1.2 Stochastic Class-Based Hard Sample Mining

In Chapter 3, stochastic class-based hard sample mining method is introduced. Performance of deep metric learning depends heavily on the capability of mining hard negative examples during training. However, many metric learning algorithms often require intractable computational cost due to frequent feature computations and nearest neighbor searches in a large-scale dataset. As a result, existing approaches often suffer from trade-off between training speed and prediction accuracy. To alleviate this limitation, we propose a two-step approach. For a given anchor instance, it first selects a few candidate hard negative classes based on the class-to-sample distances and then performs a refined search in an instance-level only from the selected classes. As most of the classes are pruned at the first step, it is much more efficient than exhaustive search while effectively mining a large number of hard examples. We handle the imperfect class-level pruning due to the intra-class variation by stochastically mining candidate classes and hard instances. Since the proposed method can be applied to generic objects beyond the person, we perform experiments on both object retrieval and person re-identification datasets. Our experiment shows that the proposed technique finds hard negative samples effectively and improves image retrieval accuracy substantially in both image retrieval and person re-identification datasets; it achieves the state-of-the-art performance on the standard benchmark including CUB-200-2011, CARS-196, In-shop retrieval, and Stanford online products datasets.

1.3 Integrated System for Person Re-identification

In Chapter 4, we propose an integrated person re-identification system by combining the two approaches, part-aligned image representation (Chapter 2) and hard sample mining technique (Chapter 3). In addition, we propose a hard positive sample mining technique to further enhance the performance in the person re-identification datasets. In the experiments, we show that the proposed method consistently improves the accuracy in two most popular person re-identification datasets: Market-1501 and DukeMTMC.

The thesis is concluded in Chapter 5 with a summary of contributions of the thesis and suggestion for the future research directions.

Chapter 2

Part-Aligned Bilinear Represenatations

2.1 Introduction

The goal of person re-identification is to identify the same person across videos captured from different cameras. It is a fundamental visual recognition problem in video surveillance with various applications [84]. It is challenging because the camera views are usually disjoint, the temporal transition time between cameras varies considerably, and the lighting conditions/person poses differ across cameras in real-world scenarios.

Body part misalignment (i.e., the problem that body parts are spatially misaligned across person images) is one of the key challenges in person re-identification. Figure 2.1 shows some examples. This problem causes conventional strip/grid-based representations [38, 2, 112, 107, 13, 88] to be unreliable as they implicitly assume that every person appears in a similar pose within a tightly surrounded bounding box. Thus, a body part-aligned representation, which can ease the representation comparison and avoid the need for complex comparison techniques, should be designed.

To resolve this problem, recent approaches have attempted to localize body parts explicitly and combine the representations over them [75, 120, 114, 36, 115]. For example, the body parts are represented by the pre-defined (or refined [75]) bounding boxes estimated from the state-of-the-art pose estimators [120, 75, 5, 114]. This

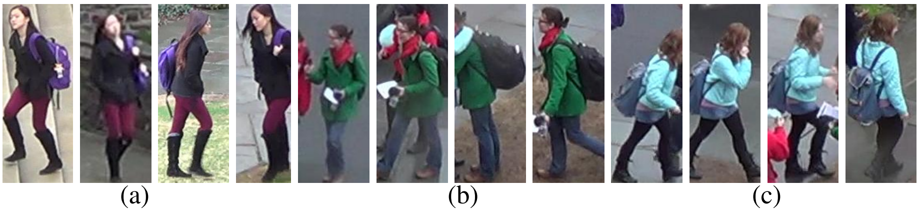


Figure 2.1: (a, b) As a person appears in different poses/viewpoints in different cameras, and (c) human detections are imperfect, the corresponding body parts are usually not spatially aligned across the human detections, causing person re-identification to be challenging.

scheme requires highly-accurate pose estimation. Unfortunately, state-of-the-art pose estimation solutions are still not perfect. Also, these schemes are bounding box-based and lack fine-grained part localization within the boxes. To mitigate the problems, we propose to encode human poses by feature maps rather than by bounding boxes. Recently, Zhao et al. [115] represented body parts through confidence maps, which are estimated using attention techniques. The method has a lack of guidance on body part locations during the training, thereby failing to attend to certain body regions consistently.

In this paper, we propose a part-aligned representation for person re-identification. Our approach learns to represent the human poses as part maps and combine them directly with the appearance maps to compute part-aligned representations. More precisely, our model consists of a two-stream network and an aggregation module. 1) Each stream separately generates appearance and body part maps. 2) The aggregation module first generates the part-aligned feature maps by computing the bilinear mapping of the appearance and part descriptors at each location, and then spatially averages the local part-aligned descriptors. The resulting image matching similarity is equivalent to an aggregation of the local appearance similarities of the corresponding body parts. Since the similarity does not depend on the relative positions of parts, the misalignment problem is reduced.

Training the network does not require any body part annotations on the person re-identification dataset. Instead, we simply initialize the part map generation stream using the pre-trained weights, which are trained from a standard pose estimation dataset. Surprisingly, although our approach only optimizes the re-identification loss function, the resulting two-stream network successfully separates appearance and part information into each stream, thereby generating the appearance and part maps from each of them, respectively. In particular, the part maps adapt from the original form to further differentiate informative body parts for person re-identification. Through extensive experiments, we verify that our approach consistently improves the accuracy of the baseline and achieves competitive/superior performance over standard image datasets, Market-1501, CUHK03, CUHK01 and DukeMTMC, and one standard video dataset, MARS.

2.2 Related Work

The early solutions of person re-identification mainly relied on hand-crafted features [51, 40, 25, 54], metric learning techniques [110, 111, 59, 41, 33, 39, 30], and probabilistic patch matching algorithms [7, 6, 69] to handle resolution/light/view/pose changes. Recently, attributes [76, 77, 117], transfer learning [60, 70], re-ranking [122, 21], partial person matching [124], and human-in-the-loop learning [53, 93], have also been studied. More can be found in the survey [123]. In the following, we review recent spatial-partition-based and part-aligned representations, matching techniques, and some works using bilinear pooling.

Regular spatial-partition based representations. The approaches in this stream of research represent an image as a combination of local descriptors, where each local descriptor represents a spatial partition such as grid cell [38, 2, 112] and horizontal stripe [107, 13, 88]. They work well under a strict assumption that the location of each body part is consistent across images. This assumption is often violated under realistic conditions, thereby causing the methods to fail. An extreme case is that no spatial partition is used and a global representation is computed over the whole image [59, 101, 119, 100, 9, 102].

Body part-aligned representations. Body part and pose detection results have been exploited for person re-identification to handle the body part misalignment problem [15, 105, 4, 18, 97, 14]. Recently, these ideas have been re-studied using deep learning techniques. Most approaches [120, 75, 114] represent an image as a combination of body part descriptors, where a dozen of pre-defined body parts are detected using the off-the-shelf pose estimator (possibly an additional RoI refinement step). They usually crop bounding boxes around the detected body parts and compute the representations over the cropped boxes. In contrast, we propose part-map-based representations, which is different from the previously used box-based representations [120, 75, 114].

Tang et al [84] also introduced part maps for person re-identification to solve the multi-people tracking problem. They used part maps to augment appearances as another feature, rather than to generate part-aligned representations, which is different from our method. Some works [49, 115] proposed the use of attention maps, which are expected to attend to informative body parts. They often fail to produce reliable attentions as the attention maps are estimated from the appearance maps; guidance from body part locations is lacking, resulting in a limited performance.

Matching. The simple similarity functions [107, 88, 13], e.g., cosine similarity or Euclidean distance, have been adapted, for part-aligned representations, such as our approach, or under an assumption that the representations are body part/pose aligned. Various schemes [92, 2, 38, 112] were designed to eliminate the influence from body part misalignment for spatial partition-based representations. For instance, a matching sub-network was proposed to conduct convolution and max-pooling operations, over the differences [2] or the concatenation [38, 112] of grid-based representation of a pair of person images. Varior et al. [87] proposed the use of matching maps in the intermediate features to guide feature extraction in the later layers through a gated

CNN.

Bilinear pooling. Bilinear pooling is a scheme to aggregate two different types of feature maps by using the outer product at each location and spatial pooling them to obtain a global descriptor. This strategy has been widely adopted in fine-grained recognition [43, 19, 31] and showed promising performance. For person re-identification, Ustinova et al. [85] adopted a bilinear pooling to aggregate two different appearance maps; this method does not generate part-aligned representations and leads to poor performance. Our approach uses a bilinear pooling to aggregate appearance and part maps to compute part-aligned representations.

2.3 Our Approach

The proposed model consists of a two-stream network and an aggregation module. It receives an image $\mathbf{I}$ as an input and outputs a part-aligned feature representation $\tilde{\mathbf{f}}$ as illustrated in Figure 2.2. The two-stream network contains two separate sub-networks, the appearance map extractor $\mathcal{A}$ and the part map extractor $\mathcal{P}$, which extract the appearance map $\mathbf{A}$ and part map $\mathbf{P}$, respectively. The two types of maps are aggregated through bilinear pooling to generate the part-aligned feature $\mathbf{f}$, which is subsequently normalized to generate the final feature vector $\tilde{\mathbf{f}}$.

2.3.1 Two-Stream Network

Appearance map extractor. We feed an input image $\mathbf{I}$ into the appearance map extractor $\mathcal{A}$, thereby outputting the appearance map $\mathbf{A}$:

$$\mathbf{A} = \mathcal{A}(\mathbf{I}). \tag{2.1}$$

$\mathbf{A} \in \mathbb{R}^{h \times w \times c_A}$ is a feature map of size $h \times w$, where each location is described by c_A-dimensional local appearance descriptor. We use the sub-network of GoogLeNet [83] to form and initialize $\mathcal{A}$.

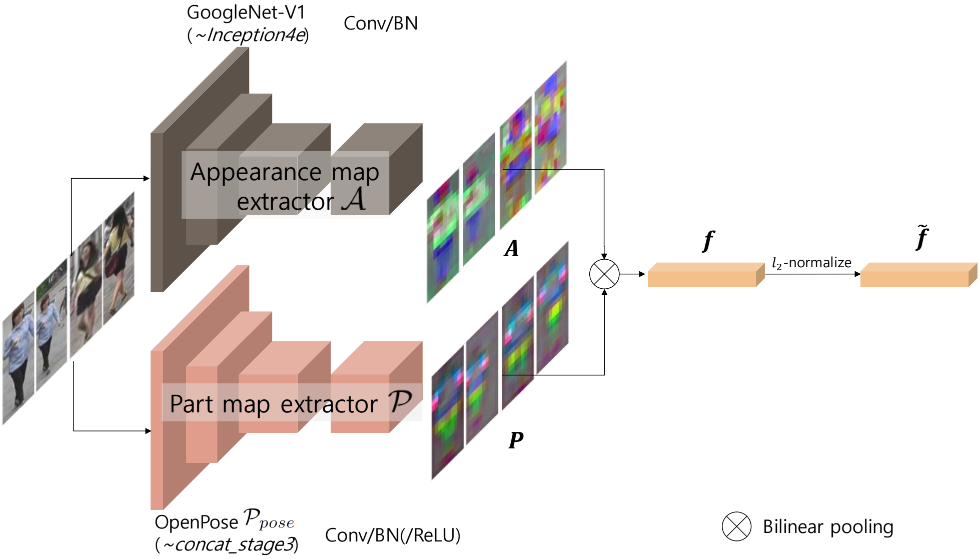


Figure 2.2: Overview of the proposed model. The model consists of a two-stream network and an aggregator (bilinear pooling). For a given image $\mathbf{I}$, the appearance and part map extractors, $\mathcal{A}$ and $\mathcal{P}$, generate the appearance and part maps, $\mathbf{A}$ and $\mathbf{P}$, respectively. The aggregator performs bilinear pooling over $\mathbf{A}$ and $\mathbf{P}$ and generates a feature vector $\mathbf{f}$. Finally, the feature vector is l_2-normalized, resulting in a final part-aligned representation $\tilde{\mathbf{f}}$. Conv and BN denote the convolution and batch normalization layers, respectively.

Part map extractor. The part map extractor $\mathcal{P}$ receives an input image $\mathbf{I}$ and outputs the part map $\mathbf{P}$:

$$\mathbf{P} = \mathcal{P}(\mathbf{I}). \tag{2.2}$$

$\mathbf{P} \in \mathbb{R}^{h \times w \times c_P}$ is a feature map of size $h \times w$, where each location is described by a c_P-dimensional local part descriptor. Considering the rapid progress in pose estimation, we use the sub-network of the pose estimation network, OpenPose [5], to form and initialize $\mathcal{P}$. We denote the sub-network of the OpenPose as $\mathcal{P}_{pose}$.

2.3.2 Bilinear Pooling

Let $\mathbf{a}_{xy}$ be the appearance descriptor at the position (x, y) from the appearance map $\mathbf{A}$, and $\mathbf{p}_{xy}$ be the part descriptor at the position (x, y) from the part map $\mathbf{P}$. We perform bilinear pooling over $\mathbf{A}$ and $\mathbf{P}$ to compute the part-aligned representation $\mathbf{f}$. There are two steps, bilinear transformation and spatial global pooling, which are mathematically given as follows:

$$\mathbf{f} = \text{pooling}_{xy}\{\mathbf{f}_{xy}\} = \frac{1}{S}\sum_{xy} \mathbf{f}_{xy}, \qquad \mathbf{f}_{xy} = \text{vec}(\mathbf{a}_{xy} \otimes \mathbf{p}_{xy}), \tag{2.3}$$

where S is the spatial size. The pooling operation we use here is average-pooling. vec(.) transforms a matrix to a vector, and $\otimes$ represents the outer product of two vectors, with the output being a matrix. The part-aligned feature $\mathbf{f}$ is then normalized to generate the final feature vector $\tilde{\mathbf{f}}$ as follows:

$$\tilde{\mathbf{f}} = \frac{\mathbf{f}}{\|\mathbf{f}\|_2}. \tag{2.4}$$

Considering the normalization, we denote the normalized part-aligned representation as $\tilde{\mathbf{f}}_{xy} = \text{vec}(\tilde{\mathbf{a}}_{xy} \otimes \tilde{\mathbf{p}}_{xy})$, where $\tilde{\mathbf{a}}_{xy} = \frac{\mathbf{a}_{xy}}{\sqrt{\|\mathbf{f}\|_2}}$ and $\tilde{\mathbf{p}}_{xy} = \frac{\mathbf{p}_{xy}}{\sqrt{\|\mathbf{f}\|_2}}$. Therefore, $\tilde{\mathbf{f}} = \frac{1}{S}\sum_{xy} \tilde{\mathbf{f}}_{xy}$.

Part-aligned interpretation. We can decompose $\mathbf{a} \otimes \mathbf{p}$[1] into c_P components:

$$\mathrm{vec}(\mathbf{a} \otimes \mathbf{p}) = [(p_1\mathbf{a})^\top \ (p_2\mathbf{a})^\top \ \ldots (p_{c_P}\mathbf{a})^\top]^\top, \tag{2.5}$$

where each sub-vector $p_i\mathbf{a}$ corresponds to a i-th part channel. For example, if $p_{knee} = 1$ on knee and 0 otherwise, then $p_{knee}\mathbf{a}$ becomes $\mathbf{a}$ only on the knee and $\mathbf{0}$ otherwise. Thus, we call $\mathrm{vec}(\mathbf{a}\otimes\mathbf{p})$ as part-aligned representation. In general, each channel c does not necessarily correspond to a certain body part. However, the part-aligned representation remains valid as $\mathbf{p}$ encodes the body part information. Section 2.4 describes this interpretation in detail.

2.3.3 Loss

To train the network, we utilize the widely-used triplet loss function. Let $\mathbf{I}_q$, $\mathbf{I}_p$ and $\mathbf{I}_n$ denote the query, positive and negative images, respectively. Then, $(\mathbf{I}_q, \mathbf{I}_p)$ is a pair of images of the same person, and $(\mathbf{I}_q, \mathbf{I}_n)$ is that of different persons. Let $\tilde{\mathbf{f}}_q$, $\tilde{\mathbf{f}}_p$, and $\tilde{\mathbf{f}}_n$ indicate their representations. The triplet loss function is formulated as

$$\ell_{\text{triplet}}(\tilde{\mathbf{f}}_q, \tilde{\mathbf{f}}_p, \tilde{\mathbf{f}}_n) = \max(m + \mathrm{sim}(\tilde{\mathbf{f}}_q, \tilde{\mathbf{f}}_n) - \mathrm{sim}(\tilde{\mathbf{f}}_q, \tilde{\mathbf{f}}_p), 0), \tag{2.6}$$

where m denotes a margin and $\mathrm{sim}(\mathbf{x}, \mathbf{y}) =< \mathbf{x}, \mathbf{y} >$. The margin is empirically set as $m = 0.2$. The overall loss function is written as follows.

$$\mathcal{L} = \frac{1}{|\mathcal{T}|} \sum\nolimits_{(\mathbf{I}_q, \mathbf{I}_p, \mathbf{I}_n) \in \mathcal{T}} \ell_{\text{triplet}}(\tilde{\mathbf{f}}_q, \tilde{\mathbf{f}}_p, \tilde{\mathbf{f}}_n), \tag{2.7}$$

where $\mathcal{T}$ is the set of all triplets, $\{(\mathbf{I}_q, \mathbf{I}_p, \mathbf{I}_n)\}$.

2.4 Analysis

2.4.1 Part-Aware Image Similarity

We show that under the proposed part-aligned representation in Eqs.(2.3) and (2.4), the similarity between two images is equivalent to the aggregation of local appearance

[1] We drop the subscript xy for presentation clarification

similarities between the corresponding body parts. The similarity between two images can be represented as the sum of local similarities between every pair of locations as follows.

$$\begin{aligned}\mathrm{sim}_I(\mathbf{I},\mathbf{I}') =< \tilde{\mathbf{f}}, \tilde{\mathbf{f}}' > &= \frac{1}{S^2} < \sum_{xy} \tilde{\mathbf{f}}_{xy}, \sum_{x'y'} \tilde{\mathbf{f}}'_{x'y'} > \\ &= \frac{1}{S^2} \sum_{xy} \sum_{x'y'} < \tilde{\mathbf{f}}_{xy}, \tilde{\mathbf{f}}'_{x'y'} > \\ &= \frac{1}{S^2} \sum_{xy} \sum_{x'y'} \mathrm{sim}(\tilde{\mathbf{f}}_{xy}, \tilde{\mathbf{f}}'_{x'y'}), \end{aligned} \tag{2.8}$$

where $\mathrm{sim}_I(,)$ measures the similarity between images. Here, the local similarity is computed by an inner product:

$$\begin{aligned}\mathrm{sim}(\tilde{\mathbf{f}}_{xy}, \tilde{\mathbf{f}}'_{x'y'}) &=< \mathrm{vec}(\tilde{\mathbf{a}}_{xy} \otimes \tilde{\mathbf{p}}_{xy}), \mathrm{vec}(\tilde{\mathbf{a}}'_{x'y'} \otimes \tilde{\mathbf{p}}'_{x'y'}) > \\ &=< \tilde{\mathbf{a}}_{xy}, \tilde{\mathbf{a}}'_{x'y'} >< \tilde{\mathbf{p}}_{xy}, \tilde{\mathbf{p}}'_{x'y'} > \\ &= \mathrm{sim}(\tilde{\mathbf{a}}_{xy}, \tilde{\mathbf{a}}'_{x'y'})\, \mathrm{sim}(\tilde{\mathbf{p}}_{xy}, \tilde{\mathbf{p}}'_{x'y'}). \end{aligned} \tag{2.9}$$

This local similarity can be interpreted as the appearance similarity weighted by the body part similarity or vice versa. Thus, from Eqs(2.8) and (2.9), the similarity between two images is computed as the average of local appearance similarities weighted by the body part similarities at the corresponding positions:

$$\mathrm{sim}_I(\mathbf{I},\mathbf{I}') = \frac{1}{S^2} \sum_{xyx'y'} \mathrm{sim}(\tilde{\mathbf{a}}_{xy}, \tilde{\mathbf{a}}'_{x'y'})\, \mathrm{sim}(\tilde{\mathbf{p}}_{xy}, \tilde{\mathbf{p}}'_{x'y'}).$$

As a result, the image similarity does not depend on the relative positions of parts in images, and therefore the misalignment problem is reduced. To make the local part similarity to be always non-negative and therefore the sign of the local similarity depends only on the sign of the local appearance similarity, we can also restrict the part descriptors $\mathbf{p}_{xy}$ to be element-wise non-negative by adding a ReLU layer after the part map extractor $\mathcal{P}$ as shown in Figure 2.2. As this variant results in similar accuracy to the original one, we used the model without the ReLU layer for all the experiments. See supplementary material for more details.

2.4.2 Relationship to the Baseline Models

Consider a baseline approach that only uses the appearance maps and spatial global pooling for image representation. Then, the image similarity is computed as $\mathrm{sim}_I(\mathbf{I}, \mathbf{I}') = \frac{1}{S^2}\sum_{xyx'y'} \mathrm{sim}(\tilde{\mathbf{a}}_{xy}, \tilde{\mathbf{a}}'_{x'y'})$. Unlike our model, this approach cannot reflect part similarity. Consider another model based on the box-based representation, which represents an image as a concatenation of K body part descriptors, where k-th body part is represented as the average-pooled appearance feature within the corresponding bounding box. This model is equivalent to our model when $\mathbf{p}_{xy}$ is defined as $\mathbf{p}_{xy} = [\delta[(x, y) \in R_1], \cdots, \delta[(x, y) \in R_K]]$, where R_k is the region within the k-th part bounding box and $\delta[\cdot]$ is an indicator function, i.e., $\delta[x] = 1$ if x is true otherwise 0. Because our model contains these baselines as special cases and is trained to optimize the re-identification loss, it is guaranteed to perform better than them.

2.4.3 Decomposition of Appearance and Part Maps

At the beginning of the training, the two streams of the network mainly represent the appearance and part maps because the appearance map extractor $\mathcal{A}$ and the part map extractor $\mathcal{P}$ are initialized using GoogleNet [82] pre-trained on ImageNet [66] and OpenPose [5] model pre-trained on COCO [42], respectively. During training, we do not set any constraints on the two streams, i.e., no annotations for the body parts, but only optimize the re-identification loss. Surprisingly, the trained two-stream network maintains to decompose the appearance and part information into two streams: one stream corresponds to the appearance maps and the other corresponds to the body part maps.

We visualize the distribution of the learned local appearance and part descriptors using t-SNE [52] as shown in Figures 2.5 (a) and (b). Figure 2.5 (a) shows that the appearance descriptors are clustered depending on the appearance while being independent on the parts that they come from. For example, the red/yellow box shows that the red/black-colored patches are closely embedded, respectively. By contrast, Figure 2.5

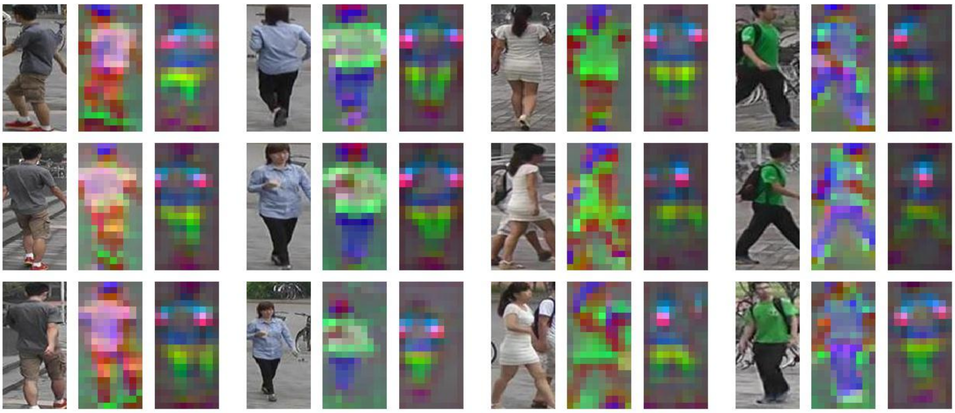


Figure 2.3: Visualization of the appearance maps $\mathbf{A}$ and part maps $\mathbf{P}$ obtained from the proposed method. For a given input image (left), appearance (center) and part (right) maps encode the appearance and body parts, respectively. For both appearance and part maps, the same color implies that the descriptors are similar, whereas different colors indicate that the descriptors are different. The appearance maps share similar color patterns among the images from the same person, which means that the patterns of appearance descriptors are similar as well. In the part maps, the color differs depending on the location of the body parts where the descriptors came from. (Best viewed in color)

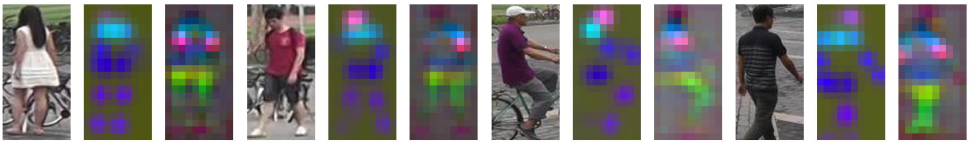
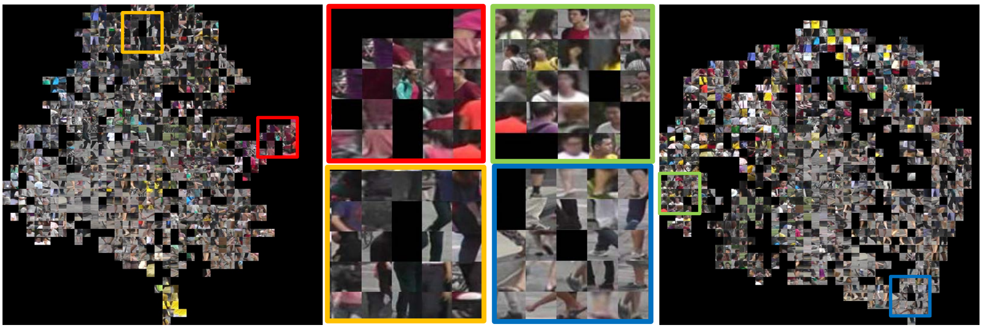


Figure 2.4: Comparing the body part descriptors. For a given image (left), the conventional joint-based (center) and the proposed (right) descriptors are visualized. (Best viewed in color)



(a) Appearance features

(b) Part features

Figure 2.5: The t-SNE visualization of the normalized local appearance and part descriptors on the Market-1501 dataset. It illustrates that our two-stream network decomposes the appearance and part information into two streams successfully. (a) Appearance descriptors are clustered roughly by colors, independently from the body parts where they came from. (b) Part descriptors are clustered by body parts where they came from, regardless of the colors. (Best viewed on a monitor when zoomed in)

(b) illustrates that the local part embedding maps the similar body parts into close regions regardless of color. For example, the green/blue box shows that the features from the head/lower leg are clustered, respectively. In addition, physically adjacent body parts, such as head–shoulder and shoulder–torso, are also closely embedded.

To understand how the learned appearance/part descriptors are used in person re-identification, we visualize the appearance maps $\mathbf{A}$ and the part maps $\mathbf{P}$ following the visualization used in SIFTFlow [46], as shown in Figure 2.3. [2] For a given input image (left), the appearance (center) and part (right) maps encode the appearance and body parts, respectively. The figure shows how the appearance maps differentiate different persons while being invariant for each person. By contrast, the part maps encode the body parts independently from their appearance. In particular, a certain body part is represented by a similar color across images, which confirms our observation in Figure 2.5 that the part features from physically adjacent regions are closely embedded.

Our approach learns the optimal part descriptor for person re-identification, rather than relying on the pre-defined body parts. Figure 2.4 qualitatively compares the conventional body part descriptor and the one learned by our approach. [3] In the previous works on human pose estimation [96, 5, 57], human poses are represented as a collection of pre-defined key body joint locations. It corresponds to a part descriptor which one-hot encodes the key body joints depending on the existence of a certain body joint at the location, e.g, $p_{knee} = 1$ on knee and 0 otherwise. Compared to the baseline, ours smoothly maps the body parts. In other words, the colors are continuous over the whole body in ours, which implies that the adjacent body parts are mapped closely. By contrast, the baseline not always maps adjacent body parts maps closely. For example, the upper leg between the hip and knee is more close to the background descriptors than to ankle or knee descriptors. This smooth mapping makes our method to work robustly against the pose estimation error because the descriptors do not change rapidly

[2] we project the c_A(or c_P)-dimensional local descriptor vector onto the 3D RGB space, by mapping the top three principal components of the descriptor to the principal components of RGB.

[3] We used the visualization method proposed in SIFTFlow [46]

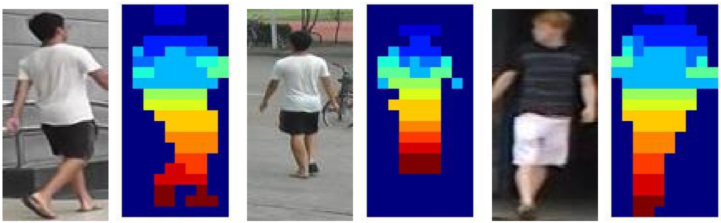
along the body parts and therefore are insensitive to the error in estimation. In addition, the part descriptors adopt to distinguish the informative parts more finely. For example, the mapped color varies sharply from elbow to shoulder and differentiates the detailed regions. Based on these properties, the learned part descriptors better support the person re-identification task and improve the accuracy.

2.4.4 Part-Alignment Effects on Reducing Misalignment Issue

Consider a similarity matrix $S_{feat}(\mathbf{I}, \mathbf{I}') \in \mathbb{R}^{hw \times h'w'}$, whose $(x + wy, x' + w'y')$-th element is $\mathrm{sim}(\tilde{\mathbf{f}}_{xy}, \tilde{\mathbf{f}}'_{x'y'})$. $S_{part}(\mathbf{I}, \mathbf{I}')$ and $S_{app}(\mathbf{I}, \mathbf{I}')$ are constructed similarly from $\mathrm{sim}(\tilde{\mathbf{p}}_{xy}, \tilde{\mathbf{p}}'_{x'y'})$ and $\mathrm{sim}(\tilde{\mathbf{a}}_{xy}, \tilde{\mathbf{a}}'_{x'y'})$. From Eq.8 and 9, $\mathrm{sim}(\mathbf{I}, \mathbf{I}')$ is the average of all the elements of $S_{feat}(\mathbf{I}, \mathbf{I}')$, and $S_{feat}(\mathbf{I}, \mathbf{I}') = S_{app}(\mathbf{I}, \mathbf{I}') \odot S_{part}(\mathbf{I}, \mathbf{I}')$, where $\odot$ denotes the element-wise product.

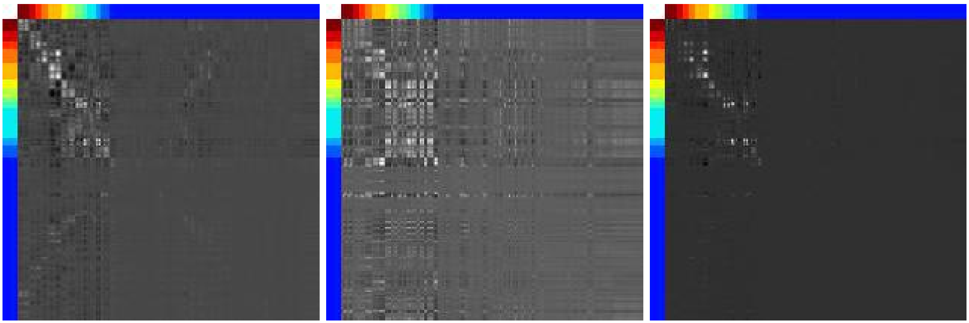
To demonstrate part-alignment effect, we visualize similarity matrices (Fig. 2.7 and 2.8) for two example image pairs, $(\mathbf{I}_a, \mathbf{I}_p)$ and $(\mathbf{I}_a, \mathbf{I}_n)$, shown in Fig. 2.6. In Fig. 2.7 and 2.8, the order of rows/columns is re-arranged for better visualization (color bars represent the corresponding parts shown in Fig. 2.6). Fig. 2.7 and 2.8 shows that local part similarity $\mathrm{sim}(\tilde{\mathbf{p}}_{xy}, \tilde{\mathbf{p}}'_{x'y'})$ is approximately 0 for almost every location pair $((x, y), (x', y'))$ and activates positively/negatively (bright/dark) only when (x, y) and (x', y') belong to the same/negatively-related body parts.

From Eq.9, $\mathrm{sim}(\tilde{\mathbf{f}}_{xy}, \tilde{\mathbf{f}}'_{x'y'})$ activates only when both part and appearance similarities, $\mathrm{sim}(\tilde{\mathbf{p}}_{xy}, \tilde{\mathbf{p}}'_{x'y'})$ and $\mathrm{sim}(\tilde{\mathbf{a}}_{xy}, \tilde{\mathbf{a}}'_{x'y'})$, activate. Thus, from Eq. 8, $\mathrm{sim}(\mathbf{I}, \mathbf{I}') \approx \sum_{((x,y),(x',y')) \in \mathcal{R}} \mathrm{sim}(\tilde{\mathbf{a}}_{xy}, \tilde{\mathbf{a}}'_{x'y'}) \, \mathrm{sim}(\tilde{\mathbf{p}}_{xy}, \tilde{\mathbf{p}}'_{x'y'})$, where $\mathcal{R}$ is a set of every location pair $((x, y), (x', y'))$ that has non-zero $\mathrm{sim}(\tilde{\mathbf{p}}_{xy}, \tilde{\mathbf{p}}'_{x'y'})$, i.e. (x, y) and (x', y') belong to same/negatively-related body parts. Since the image similarity is represented as the sum of appearance similarities between corresponding parts minus the sum of appearance similarities between negatively-related parts, regardless of their locations in images, the misalignment problem is reduced.



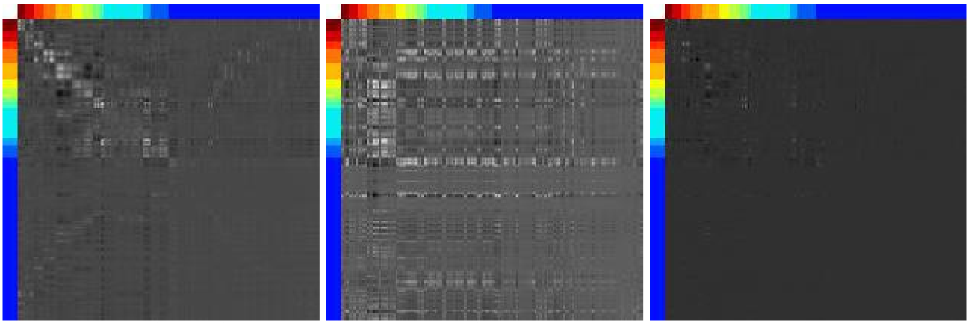
(a) $\mathbf{I}_a$ (b) $\mathbf{I}_p$ (c) $\mathbf{I}_n$

Figure 2.6: Image and its part color encoding for visualization in Fig. 2.7 and 2.8



$S_{part}(\mathbf{I}_a, \mathbf{I}_p)$ $S_{app}(\mathbf{I}_a, \mathbf{I}_p)$ $S_{feat}(\mathbf{I}_a, \mathbf{I}_p)$

Figure 2.7: Visualization of the similarity matrices for pairs of images shown in Fig 2.6



$S_{part}(\mathbf{I}_a, \mathbf{I}_n)$ $S_{app}(\mathbf{I}_a, \mathbf{I}_n)$ $S_{feat}(\mathbf{I}_a, \mathbf{I}_n)$

Figure 2.8: Visualization of the similarity matrices for pairs of images shown in Fig 2.6

2.5 Implementation Details

Network architecture. We use a sub-network of the first version of GoogLeNet [82] as the appearance map extractor $\mathcal{A}$, from the image input of size 160×80 to the output of *inception4e*, which is followed by a 1×1 convolution layer and a batch normalization layer to reduce the dimension to 512 (Figure 2.2). Moreover, we optionally adopt dilation filters in the layers from the *inception4a* to the final layer, resulting in 20×10 response maps. Figure 2.2 illustrates the architecture of the part map extractor $\mathcal{P}$. We use a sub-network of the OpenPose network [5], from the image input to the output of stage2 (i.e., *concat_stage3*) to extract 185 pose heat maps, which is followed by a 3×3 convolution layer and a batch normalization layer, thereby outputting 128 part maps. We adopt the compact bilinear pooling [19] to aggregate the two feature maps into a 512-dimensional vector $\mathbf{f}$.

Compact bilinear pooling. The bilinear transformation over the 512-dimensional appearance vector and the 128-dimensional part vector results in an extremely high dimensional vector, which consumes large computational cost and memory. To resolve this issue, we use the tensor sketch approach [61] to compute a compact representation as in [19]. The key idea of the tensor sketch approach is that the original inner product, on which the Euclidean distance is based, between two high-dimensional vectors can be approximated as an inner product of the dimension-reduced vectors, which are random projections of the original vectors. Details can be found in [61].

Network training. The appearance map extractor $\mathcal{A}$ and part map extractor $\mathcal{P}$ are fine-tuned from the network pre-trained on ImageNet [66] and COCO [42], respectively. The added layers are initialized following [24]. We use the stochastic gradient descent algorithm. The initial learning rate, weight decay, and the momentum are set to 0.01, 2×10^{-4}, and 0.9, respectively. The learning rate is decreased by a factor of 5 after every $20,000$ iterations. All the networks are trained for $75,000$ iterations.

We follow [115] to sample a mini-batch of samples at each iteration and use all the possible triplets within each mini-batch. The gradients are computed using the acceleration trick presented in [115]. In each iteration, we sample a mini-batch of 180 images, e.g., there are on average 18 identities with each containing 10 images. In total, there are approximately $10^2 \cdot (180 - 10) \cdot 18 \approx 3 \times 10^5$ triplets in each iteration.

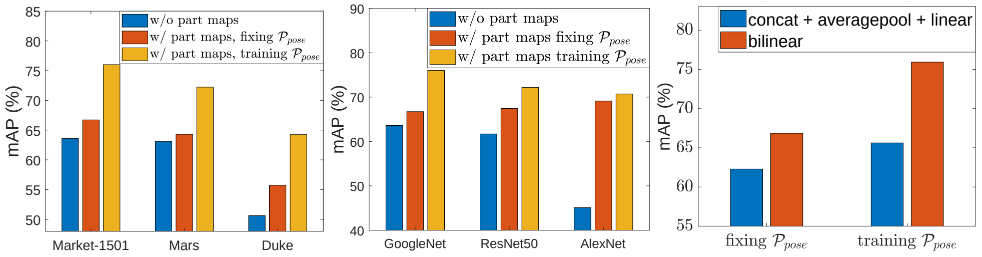
2.6 Experiments

2.6.1 Datasets

Market-1501 [121] This dataset is one of the largest benchmark datasets for person re-identification. Six cameras are used: five high-resolution cameras and one low-resolution camera. There are $32,668$ DPM-detected pedestrian image boxes of $1,501$ identities: 750 identities are utilized for training and the remaining 751 identities are used for testing. There are $3,368$ query images and $19,732$ gallery images with $2,793$ distractors.

CUHK03 [38] This dataset consists of $13,164$ images of $1,360$ people captured by six cameras. Each identity appears in two disjoint camera views (i.e., 4.8 images in each view on average). We divided the train/test set following the previous work [38]. For each test identity, two images are randomly sampled as the probe and gallery images and the average accuracy over 20 trials is reported as the final result.

CUHK01 [37] This dataset comprises 3884 images of 971 people captured in two disjoint camera views. Two images are captured for each person from each of the two cameras (i.e., a total of four images). Experiments are performed under two evaluation settings [2], using 100 and 486 test IDs. Following the previous works [2, 9, 13, 115], we fine-tuned the model from the one learned from the CUHK03 training set for the experiments with 486 test IDs.



(a) (b) (c)

Figure 2.9: (a) Comparison of different pooling methods on the appearance maps. (c) Comparing models, with and without part maps, on different datasets. (d) Comparing models, with and without part maps, on different architectures of the appearance map extractor. If not specified, the results are reported on Market-1501. (b) Comparison of different methods to aggregate the appearance and part maps.

DukeMTMC [64] This dataset is originally proposed for video-based person tracking and re-identification. We use the fixed train/test split and evaluation setting following [45].It includes 16,522 training images of 702 identities, 2,228 query images of 702 identities and 17,661 galley images.

MARS [119] This dataset is proposed for video-based person re-identification. It consists of 1261 different pedestrians captured by at least two cameras. There are 509,914 bounding boxes and 8,298 tracklets from 625 identities for training and 681,089 bounding boxes and 12,180 tracklets from 636 identities for testing.

2.6.2 Evaluation Metrics

We use both the cumulative matching characteristics (CMC) and mean average precision (mAP) to evaluate the accuracy. The CMC score measures the quality of identifying the correct match at each rank. For multiple ground truth matches, CMC cannot measure how well all the images are ranked. Therefore, we report the mAP scores for Market-1501, DukeMTMC, and MARS where more than one ground truth images are in the gallery.

2.6.3 Comparison with the Baselines

We compare the proposed method with the baselines in three aspects. In this section, when not specified, all the experiments are performed on the Market-1501 dataset, all the models do not use dilation, and $\mathcal{P}_{pose}$ is trained together with the other parameters.

Effect of part maps We compare our method with a baseline that does not explicitly use body parts. As a baseline network, we use the appearance map extractor of Eq.(2.1), which is followed by a global spatial average pooling and a fully connected layer, thereby outputting the 512-dimensional image descriptor. Figures 2.9 (a) and (b) compare the proposed method with the baseline, while varying the training strategy: *fixing* and *training* $\mathcal{P}_{pose}$. *Fixing* $\mathcal{P}_{pose}$ initializes $\mathcal{P}_{pose}$ using the pre-trained weights [5, 42] and fixes the weight through the training. *Training* $\mathcal{P}_{pose}$ also initializes $\mathcal{P}_{pose}$ in the same way, but fine-tunes the network using the loss of Eq.(2.7) during training. Figure 2.9 (a) illustrates the accuracy comparison on three datasets, Market-1501, MARS, and Duke. It shows that using part maps consistently improves the accuracy on all the three datasets from the baseline. In addition, training $\mathcal{P}_{pose}$ largely improves the accuracy than fixing $\mathcal{P}_{pose}$. It implies that the part descriptors are adopted to better serve the person re-identification task. Figure 2.9 (b) shows the accuracy comparison while varying the appearance sub-network architecture. Similarly, the baseline accuracy is improved when part maps are introduced and further improved when $\mathcal{P}_{pose}$ is fine-tuned during training.

Effect of bilinear pooling Figure 2.9 (c) compares the proposed method (*bilinear*) to the baseline with a different aggregator. For the given appearance and part maps, *concat+averagepool+linear* generates a feature vector by concatenating two feature maps, spatially average pooling, and feeding through a fully connected layer, resulting in a 512-dimensional vector. The result shows that bilinear pooling consistently achieves higher accuracy than the baseline, for both cases when $\mathcal{P}_{pose}$ is fixed/trained.

Comparison with previous pose-based methods Finally, we compare our method with three previous works [120, 114, 75], which use human pose estimation, on Market-1501. For a fair comparison, we use the reduced CPM(R-CPM [$\sim$3M param]) utilized in [75]as $\mathcal{P}_{pose}$. The complexity of the R-CPM is lower than the standard FCN ($\sim$6M param) used in [114] and CPM ($\sim$30M param) used in [120]. As the appearance network, [114] used GoogLeNet and [120] used ResNet50. [75] used 13 inception modules, whereas we use 7. Table 2.2 shows the comparison. In comparison with the method adopted by [120, 114, 75], the proposed method (Inception V1, R-CPM) achieves an increase of 4% and 9% for rank@1 accuracy and mAP, respectively. It shows that our method effectively uses the part information compared with the previous approaches.

2.6.4 Comparison with State-of-the-Art Methods

Market-1501 Table 2.2 and 2.3 show the comparison over two query schemes, single query and multi-query. Single query takes one image from each person whereas multi-query takes multiple images. For the multi-query setting, one descriptor is obtained from multiple images by averaging the feature from each image. Our approach achieves the best accuracy in terms of both mAP and rank@K for both single and multi-query. We also provide the result after re-ranking [128], which further boosts accuracy. In addition, we conduct the experiment over an expanded dataset with additional $500K$ images [121]. Following the standard evaluation protocol [27], we report the results over four different gallery sets, $19,732$, $119,732$, $219,732$, and $519,732$, using two evaluation metrics (i.e., rank-1 accuracy and mAP). Table 2.4 reports the results. The proposed method outperforms all the other methods.

CUHK03 We report the results with two person boxes: manually labeled and detected. Table 2.5 presents the comparison with existing solutions. In the case of detected boxes, the state-of-the-art accuracy is achieved. With manual bounding boxes,

Table 2.1: Accuracy comparison on CUHK01

	100 test IDs				486 test IDs			
Rank	1	5	10	20	1	5	10	20
Shi et al. [107]	69.4	90.8	96.0	-	-	-	-	-
SIR-CIR [92]	71.8	91.6	96.0	98.0	-	-	-	-
Zhang et al. [112]	89.6	97.8	98.9	99.7	76.5	94.2	**97.5**	-
Zhao et al. [115]	88.5	**98.4**	**99.6**	**99.9**	74.7	92.6	96.2	98.4
Geng et al. [23]	**93.2**	-	-	-	77.0	-	-	-
Chen et al. [12]	-	-	-	-	74.5	91.2	94.8	97.1
Ustinova et al. [85] (Bilinear)	-	-	-	-	52.9	78.1	86.3	92.6
Zhao et al. [114] (Pose)	-	-	-	-	79.9	**94.4**	97.1	98.6
Part-aligned	90.4	97.1	98.1	98.9	**80.7**	**94.4**	**97.3**	**98.6**

Table 2.2: Accuracy comparison on Market-1501 with single query

	Sinlge Query				
Rank	1	5	10	20	mAP
Varior et al. 2016 [87]	61.6	-	-	-	35.3
Zhong et al. 2017 [128]	77.1	-	-	-	63.6
Zhao et al. 2017 [115]	80.9	91.7	94.7	96.6	63.4
Sun et al. 2017 [81]	82.3	92.3	95.2	-	62.1
Geng et al. 2016 [23]	83.7	-	-	-	65.5
Lin et al. 2017 [45]	84.3	93.2	95.2	97.0	64.7
Bai et al. 2017 [3]	82.2	-	-	-	68.8
Chen et al. 2017 [12]	72.3	88.2	91.9	95.0	-
Hermans et al. 2017 [27]	84.9	94.2	-	-	69.1
+ re-ranking	86.7	93.4	-	-	81.1
Zhang et al. 2017 [113]	87.7	-	-	-	68.8
Zhong et al. 2017 [129]	87.1	-	-	-	71.3
+ re-ranking	89.1	-	-	-	83.9
Chen et al. 2017 [11] (MobileNet)	90.0	-	-	-	70.6
Chen et al. 2017 [11] (Inception-V3)	88.6	-	-	-	72.6
Ustinova et al. 2017 [85] (Bilinear)	66.4	85.0	90.2	-	41.2
Zheng et al. 2017 [120] (Pose)	79.3	90.8	94.4	96.5	56.0
Zhao et al. 2017 [114] (Pose)	76.9	91.5	94.6	96.7	-
Su et al. 2017 [75] (Pose)	84.1	92.7	94.9	96.8	65.4
Part-aligned (Inception-V1, R-CPM)	88.8	95.6	97.3	98.6	74.5
Part-aligned (Inception-V1, OpenPose)	**90.2**	**96.1**	**97.4**	**98.4**	**76.0**
+ dilation	**91.7**	**96.9**	**98.1**	**98.9**	**79.6**
+ re-ranking	**93.4**	**96.4**	**97.4**	**98.2**	**89.9**

Table 2.3: Accuracy comparison on Market-1501

	Multi Query				
Rank	1	5	10	20	mAP
Geng et al. 2016 [23]	89.6	-	-	-	73.8
Bai et al. 2017 [3]	88.2	-	-	-	76.2
Hermans et al. 2017 [27]	90.5	96.3	-	-	76.4
+ re-ranking	91.8	95.8	-	-	87.2
Zhang et al. 2017 [113]	91.7	-	-	-	77.1
Part-aligned (Inception-V1, R-CPM)	92.9	97.3	98.4	99.1	81.7
Part-aligned (Inception-V1, OpenPose)	93.2	97.5	98.4	99.1	82.7
+ dilation	**94.0**	**98.0**	**98.8**	**99.3**	**85.2**
+ re-ranking	**95.4**	**97.5**	**98.2**	**98.9**	**93.1**

Table 2.4: Accuracy comparison on Market-1501+500k

		Gallery size			
	metric	19732	119732	219732	519732
Zheng et al. 2017 [125]	rank-1	79.5	73.8	71.5	68.3
	mAP	59.9	52.3	49.1	45.2
Linet al. 2017 [45]	rank-1	84.0	79.9	78.2	75.4
	mAP	62.8	56.5	53.6	49.8
Hermans et al. 2017 [27]	rank-1	84.9	79.7	77.9	74.7
	mAP	69.1	61.9	58.7	53.6
Part-aligned (Inception V1, OpenPose)	rank-1	**91.7**	**88.3**	**86.6**	**84.1**
	mAP	**79.6**	**74.2**	**71.5**	**67.2**

Table 2.5: Accuracy comparison on CUHK03

	Detected				Manual			
Rank	1	5	10	20	1	5	10	20
Shi et al. [107]	52.1	84.0	92.0	96.8	61.3	88.5	96.0	99.0
SIR-CIR [92]	52.2	-	-	-	-	-	-	-
Varior et al. [87]	68.1	88.1	94.6	98.8	-	-	-	-
Bai et al. [3]	72.7	92.4	96.1	-	76.6	94.6	98.0	-
Zhang et al. [112]	-	-	-	-	80.2	97.7	99.2	99.8
Sun et al. [81]	81.8	95.2	97.2	-	-	-	-	-
Zhao et al. [115]	81.6	97.3	98.4	**99.5**	85.4	97.6	99.4	**99.9**
Geng et al. [23]	84.1	-	-	-	85.4	-	-	-
Chen et al. [12]	87.5	97.4	98.7	**99.5**	-	-	-	-
Ustinova et al. [85] (Bilinear)	63.7	89.2	94.7	97.5	69.7	93.4	98.9	99.4
Zheng et al. [120] (Pose)	67.1	92.2	96.6	98.1	-	-	-	-
Zhao et al. [114] (Pose)	-	-	-	-	88.5	97.8	98.6	99.2
Su et al. [75] (Pose)	78.3	94.8	97.2	98.4	88.7	98.6	99.2	99.7
Part-aligned	**88.0**	**97.6**	**98.6**	**99.0**	**91.5**	**99.0**	**99.5**	**99.9**

Table 2.6: Accuracy comparison on DukeMTMC

Rank	1	5	10	20	mAP
Zheng et al. [127]	67.7	-	-	-	47.1
Tong et al. [103]	68.1	-	-	-	-
Lin et al. [45]	70.7	-	-	-	51.9
Schumann et al. [68]	72.6	-	-	-	52.0
Sun et al. [81]	76.7	86.4	89.9	-	56.8
Chen et al. [11] (MobileNet)	77.6	-	-	-	58.6
Chen et al. [11] (Inception-V3)	79.2	-	-	-	60.6
Zhun et al. [129]	79.3	-	-	-	62.4
+ re-ranking	84.0	-	-	-	78.3
Part-aligned (Inception V1, OpenPose)	82.1	90.2	92.7	95.0	64.2
+ dilation	**84.4**	**92.2**	**93.8**	**95.7**	**69.3**
+ re-ranking	**88.3**	**93.1**	**95.0**	**96.1**	**83.9**

Table 2.7: Accuracy comparison on MARS

Rank	1	5	10	20	mAP
Xu et al. [104] (Video)	44	70	74	81	-
McLaughlin et al. [55] (Video)	45	65	71	78	27.9
Zheng et al. [119] (Video)	68.3	82.6	-	89.4	49.3
Liu et al. [47] (Video)	68.3	81.4	-	90.6	52.9
Zhou et al. [130]	70.6	90.0	-	97.6	50.7
Li et al. [36]	71.8	86.6	-	93.1	56.1
+ re-ranking	83.0	93.7	-	97.6	66.4
Liu et al. [50]	73.7	84.9	-	91.6	51.7
Hermans et al. [27]	79.8	91.4	-	-	67.7
+ re-ranking	81.2	90.8	-	-	77.4
Part-aligned (Inception V1, OpenPose)	83.0	92.8	95	96.8	72.2
+ dilation	**83.1**	**94.2**	**95.8**	–	**74.8**

our method also achieves the best accuracy.

CUHK01 We compare the results with two evaluation settings (i.e., 100 and 486 test IDs) in Table 2.1. For 486 test IDs, the proposed method shows the best result. For 100 test IDs, our method achieves the second best result, following [23]. Note that [23] fine-tuned the model which is learned from the CUHK03+Market1501, whereas we trained the model using 871 training IDs of the CUHK01 dataset, following the settings in previous works [2, 9, 13, 115].

DukeMTMC We follow the setting in [45] to conduct the experiments. Table 2.6 reports the results. The proposed method achieves the best result for both with and without re-ranking.

MARS We also evaluate our method on one video-based person re-identification dataset [119]. We use our approach to extract the representation for each frame and aggregate the representations of all the frames using temporal average pooling, which shows similar accuracy to other aggregation schemes (RNN and LSTM). Table 4.3 presents the comparison with the competing methods. Our method shows the highest accuracy over both image-based and video-based approaches.

2.7 Summary

We propose a new method for person re-identification. The key factors that contribute to the superior performance of our approach are as follows. (1) We adopt part maps where parts are not pre-defined but learned specially for person re-identification. They are learned to minimize the re-identification loss with the guidance of the pre-trained pose estimation model. (2) The part map representation provides a fine-grained/robust differentiation of the body part depending on their usefulness for re-identification. (3) We use part-aligned representations to handle the body part misalignment problem.

The resulting approach achieves superior/competitive person re-identification performances on the standard image and video benchmark datasets.

Chapter 3

Stochastic Class-Based Hard Sample Mining

3.1 Introduction

Deep metric learning is a fundamental problem applicable to various tasks in computer vision including image retrieval [56, 72, 73, 26, 99], person re-identification [116, 28], face recognition [67, 90], and many others. The goal of deep metric learning is to approximate a feature embedding function that maps data—images in our domain—onto a common feature space. After learning, visually similar images are supposed to be clustered while the ones with heterogeneous contents are expected to be located far from each other. To meet this requirement, one can consider a triplet loss [67], which is defined on all the triplets of images in the training set. The triplet loss penalizes the cases that the distances between the images in the same classes are larger than the ones between images with different labels.

One key challenge of using the triplet loss is lack of efficient methods to identify hard negative examples for training, which is partly because embedding functions are changing continuously during training and most of the triplets easily satisfy the desired constraints [72, 26, 90, 71, 17, 118]. A naïve implementation of a metric learning algorithm based on triplet loss requires a forward propagation of the whole training dataset through feature extractor and distance computation between every pair of ex-

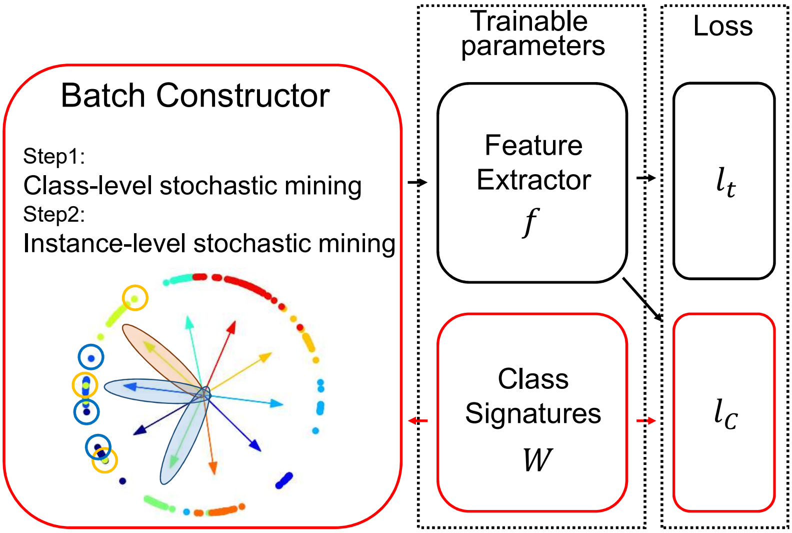


Figure 3.1: Overview of the proposed training process

amples in each iteration, which is computationally infeasible in a large-scale datasets. Therefore, most of the existing works focused on the efficient mining of the hard examples [26, 71, 90, 118].

In this paper, we argue that diversifying the training examples is also critical for high performance because it increases the number of training samples seen during the training. Existing works focused on efficient mining of hard triplets but from only a certain difficulty level (hard or semi-hard) measured with a heuristic criterion [67, 28, 26]. To balance between the diversity and hardness, our strategy is to construct a set of candidate triplet pools in different difficulty levels and compose each minibatch by sampling from one of them. Stochastically iterate between the multiple difficulty levels during training explicitly enlarges the range of difficulty and diversify examples, while keeping them hard enough for efficient training.

To this end, we propose a stochastic hard example mining technique, which models the relations across training images using surrogate relations in a coarse level. Specifically, we identify nearest neighbor classes from a set of stochastically sampled instances in an anchor class, and draw hard examples from the classes only. Since it is much more efficient than exhaustive search, it allows us to change embedding functions adaptively and update image representations in every iteration. To this end, we learn class signatures, which track the change of the embedding function, and find the hard negative classes based on them in an online manner during training with minor additional computational cost.

In sum, our contributions are as follows. First, we provide an observation that diversifying the hard triplets during training increases the accuracy. Second, we propose an efficient and effective batch construction algorithm using the class-level pruning and instance-level refined search for hard examples.

Our experiment shows that the proposed hard class mining technique improves accuracy in image retrieval tasks compared to several baseline methods on the standard datasets including CARS-196 [34], CUB-200-2011 [89], in-shop retrieval [35] and Stanford online products [74]. In addition, we adopt a compact bilinear pooling [20] to exploit the local features, which further enhances the representation power. When combined with the local-aware model, our method outperforms the state-of-the-art methods in the standard datasets.

3.2 Related Works

Hard negative mining is widely adopted to speed up convergence and enhance the discriminative power of the learned embeddings in deep metric learning [67, 28, 16, 72], especially for the triplet loss. Among them, our approach is mostly related to the ones which exploit the class label information for the mining.

There are works using precomputed relationships of neighboring classes [65, 90, 94] under the assumption that neighbor classes for a given class do not change during the training despite of the changes of the embedding function. However, that assumption does not hold in general since the goal of metric learning is to change the distance between samples and those change naturally results in different neighborhood relationship.

To resolve this problem, online update of neighboring classes have been explored [72, 71, 63, 22]. Sohn *et.al.* [72] first represents each class by its random sample and greedily find the hard classes which violate triplet conditions. Smirnov *et.al.* [71] finds the hardest negative class of another given class, by finding highest false prediction probability. Although they consider the change of embeddings, their sampling rely on greedy local search, which may limit diversity in minibatch construction especially when the number of class increases. Rippel *et.al.* [63] estimates sub-clusters within each class by performing k-means clustering over the embeddings. They exploit the cluster centers for hard negative mining, by using the cluster-to-cluster distances. However, since they use k-means clustering, they need iterative forward propagation of the entire data, and therefore, the computational issue remains. Ge *et.al.* [22] periodically calculated the center of each class by averaging the member features.

Movshovitz-Attias *et.al.* [56] and Wen *et.al.* [98] are related to ours in a sense that class representatives are jointly trained with the feature extractor. However, their purpose is different from ours that they do not use class proxies for the hard negative mining. Recently, Harwood *et.al.* [26] proposed to efficiently approximate the neighborhood relationship in the entire training set, however, they still have $O(N^2)$ complexity with the dataset of size N, to periodically scan the entire dataset and search nearest neighbors, which is not scalable to the dataset size.

Online semi-hard negative mining assigns more weights on the hard negatives within each minibatch. It has proved to improve the discriminative power of the learned embeddings [67, 28, 65]. Hermans *et.al.* [28] reported that using a hard negative subset to calculate the loss resulted in higher accuracy than using the whole samples within the minibatch, in person re-identification problem. Yu *et.al.* [108] proposes a point-to-set triplet loss which is based on the point-to-set distance which weighs hard samples more. Wu *et.al.* [99] resampled training examples within each minibatch during training with contrastive loss, in a way that resulting pairwise distances are uniformly distributed. Yuan *et.al.* [109] proposed a hard-aware deeply cascaded (HDC) network to exploit hard negatives depending on their difficulty level. Since all of them focuses on local search within a given minibatch, they can be complemented by global hard sample mining which focuses on the construction of minibatch itself. In this paper, we also use the semi-hard negative mining within each minibatch as a baseline.

Generation-based approach avoids costly mining process by learning to generate hard examples [8]. It aims at making diverse hard examples, while ensuring them to not contradict the real relationships. For example, a fake negative example closer to an anchor more effectively improves training, however, if it becomes closer than a real positive example then it may rather harm the accuracy. Therefore, generation needs careful balancing between hardness and correctness. To this end, existing works generate fake samples that preserve the original labels by reducing the l2-distance with the random sample from that class [17] or by training to be well classified [118]. Since most of the class pairs are easily distinguishable, they need to combine hard class mining process to train more powerful generator. It means the problem of mining still remains: the lack of diversity and heavy computational cost. In other words, they are not contradictory to the proposed hard class mining. Rather, they can be used together to complement each other.

In recent few years, a variety of metric learning losses have been designed to improve the most basic and popular contrastive loss and triplet loss by considering relationship between more than three examples or using different distances other than Euclidean [74, 86, 10, 72, 73, 95, 56, 63, 48, 91, 22, 44]. [74] proposed lifted structured loss. The loss takes into account a positive pair and all the associated negative pair together. Ustinova *et.al.* [86] proposed histogram loss. Their loss computes the histogram of positive and negative distances, and then penalizes the probability of pairs to be in a wrong order. Chen *et.al.* [10] showed quadruplets can improve the performance further than triplets. Sohn *et.al.* [72] proposed N-pair loss which also generalized triplet loss by allowing to consider all negative example within a batch. Song *et.al.* [73] proposed the clustering loss which also consider all the examples in a minibatch to optimize clustering score. Wang *et.al.* [95] exploited angular relationships to achieve scale invariance and use higher-order information. Since every metric loss is defined based on the relative distances between images, the loss is largely affected by the samples which the local loss is calculated from. In other words, not only contrastive/triplet loss but also structured losses have potential benefit from proper minibatch construction methods.

3.3 Deep Metric Learning with Triplet Loss

3.3.1 Triplet Loss

Our goal is to learn a function f that embeds an image $\mathbf{I}$ to a feature vector $\mathbf{x}$ in a space with a known metric, *e.g.* , Euclidean space with Euclidean distance. The function f is often called a feature extractor. The desired condition of the learned function f is that distances between the representations of similar images are small while distances between embeddings of dissimilar images are large. The notion of similarity is typically defined by semantic relations, which is often derived from the class labels. A pair of images with same label are considered to be similar and a pair of images with different labels are dissimilar. We call them as positive and negative pairs, respectively.

Let $\mathcal{X}$ be a training dataset. For a given triplet of samples, $(\mathbf{x}_a, y_a), (\mathbf{x}_p, y_a), (\mathbf{x}_n, y_n) \in \mathcal{X}$, which consists of an anchor $\mathbf{x}_a$ and a positive sample $\mathbf{x}_p$ with label y_a, and a negative sample $\mathbf{x}_n$ with label y_n, the triplet loss penalizes the case that the distance from an anchor to a positive sample is not sufficiently smaller than the distance to a negative one, which is formally given by

$$l(t) = \max(0, d(\mathbf{x}_a, \mathbf{x}_p) - d(\mathbf{x}_a, \mathbf{x}_n) + m), \tag{3.1}$$

$$l_T(\mathcal{X}) = \frac{1}{\sum_{t \in \mathcal{T}} w(t)} \sum_{t \in \mathcal{T}} w(t) l(t), \tag{3.2}$$

where $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2$, $\mathcal{T}$ denotes a set of all possible triplets constructed with elements in $\mathcal{X}$, $w(t)$ denotes an importance of the triplet t, and m denotes a margin for the difference between distances to positive and negative pairs. When every triplet has a same weight, *i.e.* $w(t) = 1$, Eq.(3.2) becomes a conventional triplet loss $l_T(\mathcal{X}) = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} l(t)$. Based on the observation that weighing more on the semi-hard triplets enhances the performance [28], we use following binary weight in all the experiments:

$$w(t) = \begin{cases} 1, & \text{if} \quad l(t) > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

In our experiments, Eq.(3.3) consistently improved the accruacy from the baseline which used the uniform weights.

3.3.2 Efficient Learning with Triplet Loss

To facilitate deep metric learning based on triplet loss, it is critical to construct minibatches containing many hard triplet examples while diversifying examples over iterations. Our main idea is to learn a signature vector for each class to reduce the computational overhead for hard triplet search. Intuitively, if two classes are located closely in an embedding space, the instances in a class are likely to be hard negatives with respect to the other. We first search for nearest neighbor classes from an anchor class based on distances from samples in the anchor class to the rest of classes, where class signature is employed to represent the classes. After reducing the number of candidate instances, we seek for nearest neighbors in an instance-level, only among the examples in the identified classes. We perform both class- and instance-level search in a stochastic manner to increase diversity of samples that belong to a minibatch. The next section describes the details of our minibatch construction algorithm.

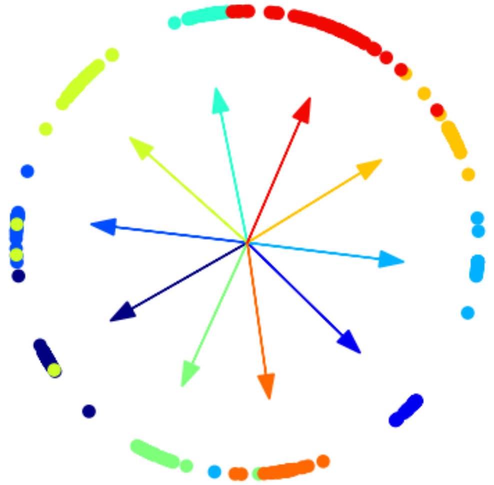


Figure 3.2: An example of class signitures and embedded instances in the MNIST dataset. Each circle and arrow denotes an instance and a class signiture trained by the proposed method. Classes are color-coded. (best viewed in color)

3.4 Batch Construction for Metric Learning

3.4.1 Neighbor Class Mining by Class Signatures

Given an anchor class, we first aim to find nearest neighbor classes based on their signatures, denoted by $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots \mathbf{w}_C\}$, which are optimized by minimizing dissimilarity to instances within the corresponding classes.

Let us denote a training example as $(\mathbf{x}, y_\mathbf{x}) \in \mathcal{X}$ with ℓ_2 normalized feature vector $\mathbf{x}$ and its label $y_\mathbf{x}$. Since instance features lie on a unit hypersphere, we can use the cosine similarity measure to compare the representations of instances and class signatures by constraining the class signatures to have a unit norm, *i.e.* , $\|\mathbf{w}_c\|_2 = 1$ for all c.

For a given instance $(\mathbf{x}, y_\mathbf{x})$, a sample-to-class similarity function $S(\mathbf{w}_c, \mathbf{x})$ should have a large value if $y_\mathbf{x} = c$ and a small one otherwise. Hence, to find nearest neighbor classes based on sample-to-class similarity, we define the following loss function:

random variable $\mathbf{x}$

$$l_C(\mathcal{W}, \mathcal{X}) = -\frac{1}{N} \sum_{(\mathbf{x}, y_\mathbf{x}) \in \mathcal{X}} \log\left(\mathbb{P}[\mathbf{x}; \mathcal{W}]\right) \tag{3.4}$$

$$= -\frac{1}{N} \sum_{(\mathbf{x}, y_\mathbf{x}) \in \mathcal{X}} \log\left(\frac{\exp(S(\mathbf{w}_{y_\mathbf{x}}, \mathbf{x}))}{\sum_c \exp(S(\mathbf{w}_c, \mathbf{x}))}\right)$$

$$= -\frac{1}{N} \sum_{(\mathbf{x}, y_\mathbf{x}) \in \mathcal{X}} \log\left(\frac{\exp(\cos\theta_{y_\mathbf{x}})}{\sum_c \exp(\cos\theta_c))}\right), \tag{3.5}$$

where $\theta_c = \angle(\mathbf{w}_c, \mathbf{x})$. It can be also interpreted as the log likelihood of $\mathbf{x}$ to be the class $y_\mathbf{x}$. Ideally, $\theta_{y_\mathbf{x}} = 0$ and $\theta_c = \pi/2$ for $c \neq y_\mathbf{x}$. Figure 3.2 shows an example distribution of the instances and the corresponding class signatures trained in the MNIST dataset.

For a given $\mathcal{W}$, we can approximate the similarity between two samples, $(\mathbf{x}, y)$ and $(\mathbf{x}', y')$, by the corresponding class-to-class similarity or class-to-sample similarity .

$$\mathbb{E}[S(\mathbf{x}, \mathbf{x}')] \approx \mathbb{E}[S(\mathbf{x}, \mathbf{w}_{y'})] \approx S(\mathbf{w}_y, \mathbf{w}_{y'}). \tag{3.6}$$

Assuming that the angle between a class signature $\mathbf{w}_c$ and the expectation of member intances is bounded by ϵ_c, following inequalities holds from the triangular inequallity:

$$\angle(\mathbf{w}_y, \mathbf{w}_{y'}) - 2\epsilon \leq \mathbb{E}[\angle(\mathbf{x}, \mathbf{x}')] \leq \angle(\mathbf{w}_y, \mathbf{w}_{y'}) + 2\epsilon \tag{3.7}$$

$$\mathbb{E}[\angle(\mathbf{x}, \mathbf{w}_{y'})] - \epsilon \leq \mathbb{E}[\angle(\mathbf{x}, \mathbf{x}')] \leq \mathbb{E}[\angle(\mathbf{x}, \mathbf{w}_{y'})] + \epsilon, \tag{3.8}$$

where $\epsilon = \max_i \epsilon_i$. Finally, from Eq.(3.6), (3.7) and (3.8), we get the following relationship between sample-to-sample distance and the corresponding class-to-class / class-to-sample distance when $\epsilon \ll 1$ (Proof in supplementary material):

$$S(\mathbf{w}_y, \mathbf{w}_{y'}) - 2\epsilon \leq \mathbb{E}[S(\mathbf{x}, \mathbf{x}')] \leq S(\mathbf{w}_y, \mathbf{w}_{y'}) + 2\epsilon, \tag{3.9}$$

$$\mathbb{E}[S(\mathbf{x}, \mathbf{w}_{y'})] - \epsilon \leq \mathbb{E}[S(\mathbf{x}, \mathbf{x}')] \leq \mathbb{E}[S(\mathbf{x}, \mathbf{w}_{y'})] + \epsilon. \tag{3.10}$$

Though we cannot guarantee that ϵ is small enough in practice, we empirically confirmed that the class-to-class similarity and the sample-to-sample similarity are highly correlated as shown in Figure 3.3 Since most of the class pairs have small similarities, it is useful for pruning the classes with small similarity values by the rank.

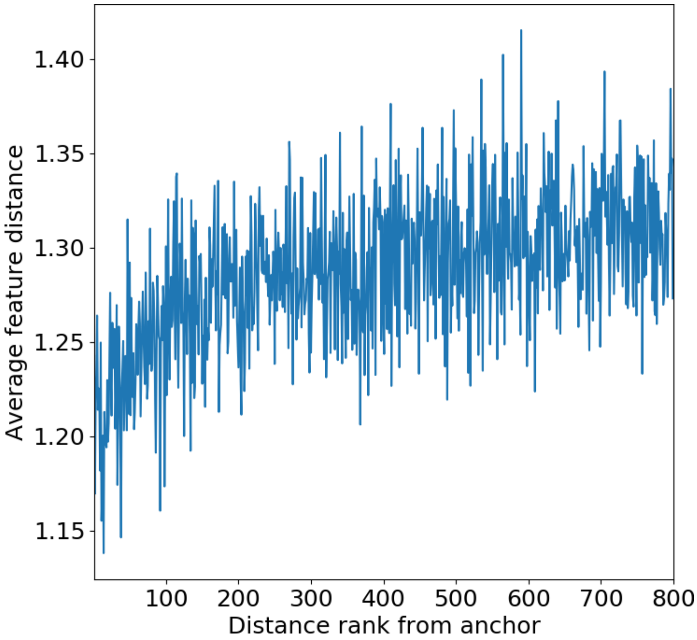
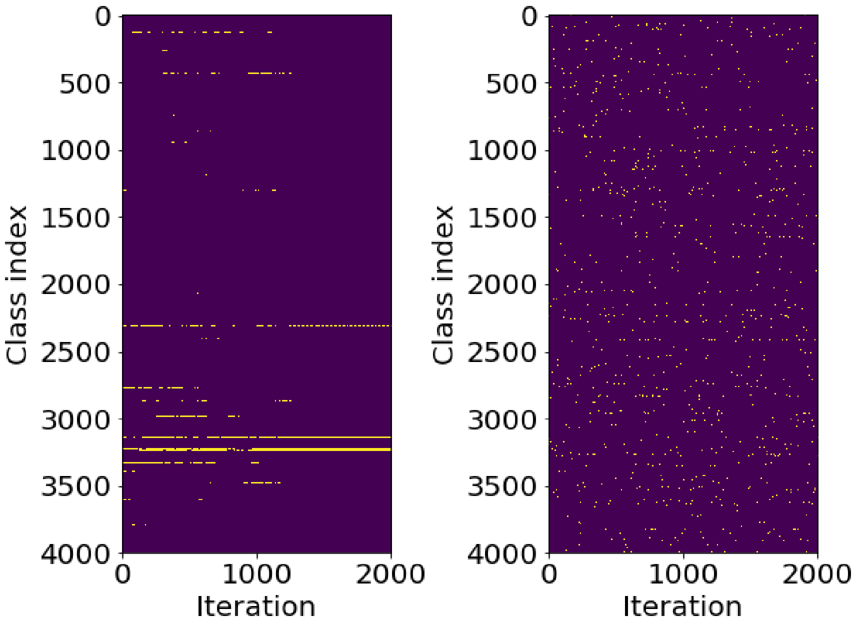


Figure 3.3: The average sample-to-sample distance between classes ranked by the class-to-class distances in In-shop retrieval dataset.

3.4.2 Batch Construction

In this section, we first review a baseline framework [116] for batch construction and loss calculation. Then, we improve the baseline by introducing a class-level hard sample mining (Alg. 1). Since a class-level neighbor search becomes relatively ineffective when intra-class variation is large, we propose a stochastic hard sample mining method, which performs a refined search in an instance-level while reducing the computational cost using class-level pruning (Alg. 3).

Baseline protocol [116] We adopt the approach in Zhao *et.al.* [116] as the baseline batch construction protocol. At each iteration, it constructs a minibatch by first randomly sampling K classes and then randomly sampling η images per each class, resulting in a minibatch of size $M = K\eta$. When calculating the loss, every possible triplet composable from the minibatch is used. This approach is popularly used [116, 72, 28]



(a) (b)

Figure 3.4: For a given anchor class c_a, the change of chosen negative classes in each minibatch is illustrated in the In-shop retrieval dataset. The x-axis and y-axis correspond to the training iteration and class index, respectively. At iteration t, the corresponding column illustrates all the negative classes used to construct a minibatch, i.e. 1 if it is chosen and 0 otherwise. It shows that the combination of classes of seen triplet varies over the iteration in Alg. 3 (b), while they are mostly fixed in Alg. 1 (a).

for its simplicity and high performance, where [72] is a special case when $K = M/2$ and $\eta = 2$.

Improved baseline by class-level hard sample mining We improve the baseline protocol by composing each minibatch with the instances randomly sampled from an anchor class and its $(K-1)$-nearest classes. It increases the expected number of hard triplets composable from each minibatch. The overall training process is summarized in Alg. 1. At each iteration, an anchor class c_a is randomly sampled. Then, its $(K-1)$-

Algorithm 1 Improved baseline with class-level mining

```
**Parameters** $K, \eta$
for $t$ = 1 : $T$ do
    Random sample anchor class $c_a$
    $\mathcal{B} \leftarrow$ Sample $\eta$ instances from $\{\mathbf{x} | y_{\mathbf{x}} = c_a\}$
    Get $\mathcal{N}$ by Eq.(3.11)
    for $c \in \mathcal{N}$ do
        $\mathcal{B}_c \leftarrow$ Sample $\eta$ instances from $\{\mathbf{x} | y_{\mathbf{x}} = c\}$
        $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{B}_c$
    end for
    Perform one iteration of training to minimize the loss Eq.(3.20) using mini-batch $\mathcal{B}$
end for
```

nearest classes $\mathcal{N}$ are chosen as follows

$$\begin{aligned} \underset{\mathcal{N}\subset\mathcal{C}}{\text{maximize}} \quad & \sum_{\mathbf{c}\in\mathcal{N}} S(\mathbf{w}_c, \mathbf{w}_{c_a}) \\ \text{s.t.} \quad & c_a \notin \mathcal{N},\ |\mathcal{N}| = K-1, \end{aligned} \tag{3.11}$$

where $\mathcal{C}$ is a set of class labels. Finally, η instances are randomly sampled from each of the selected classes and combined to construct a minibatch $\mathcal{B}$.

Though the class-level hard sample mining is efficient in terms of time, it has two limitations. First, due to the intra-class variation, approximating the sample-to-sample distance by class-to-class distance is not always accurate. Figure 3.3 shows the average sample-to-sample distance between classes, which it is ranked by the class-to-class distance calculated based on the class signatures. It implies that considering only few nearest neighbors is not enough to mine all the hard triplets. Also, an instance pair from the nearest classes may not form a hard negative as they may lie far enough. This motivates an instance-level refined search, which can be done more efficiently after a rough class-level pruning. Second, for a given anchor class, its neighbor classes converge to a certain subset of the classes as the training proceeds. Figure 3.4 (a) shows the change of chosen negative classes in each minibatch, when class c_a is used as the anchor class. It shows that the combinations of the classes in each triplet is fixed to a small subset of all the possible ones. It implies that the feature extractor sees non-diverse examples during the training. To resolve these problems, we propose a stochastic batch construction method.

Stochastic hard sample mining We first define a similarity between $\mathcal{A}$ and $\mathbf{b} \in \mathcal{B}$ as follows:

$$S_g(\mathcal{A}, \mathbf{b}) = \max_{\mathbf{a}\in\mathcal{A}} S(\mathbf{a}, \mathbf{b}), \tag{3.12}$$

where $\mathcal{A}$ and $\mathcal{B}$ are sets of vectors and $S(\mathbf{a}, \mathbf{b})$ denotes a similarity between two vectors, $\mathbf{a}$ and $\mathbf{b}$. Given a vector $\mathbf{b} \in \mathcal{B}$, $S_g(\cdot,\cdot)$ is the maximum of the similarities to the elements in $\mathcal{A}$. Based on this notation, Alg. 4 solves the following optimization

Algorithm 2 Select k elements from $\mathcal{B}$ with the largest similarity to an element in $\mathcal{A}$

```
    Input $\mathcal{A}, \mathcal{B}, k$
    Output $\mathcal{N}$
1: Sort $\mathcal{S} = \{S(\mathbf{a}, \mathbf{b})\}_{\mathbf{a} \in \mathcal{A}, \mathbf{b} \in \mathcal{B}}$ in the ascending order
2: $\mathcal{N} \leftarrow$ Unique top-$k$ elements of $\mathcal{B}$ in $\mathcal{S}$
```

problem:

$$\mathcal{N} = g(\mathcal{A}, \mathcal{B}, k) = \operatorname*{arg\,max}_{\substack{\mathcal{B}' \subset \mathcal{B} \\ |\mathcal{B}'| = k}} \sum_{\mathbf{b} \in \mathcal{B}'} S_g(\mathcal{A}, \mathbf{b}), \tag{3.13}$$

In a nutshell, it selects a subset of $\mathcal{B}$ with size k that maximizes the sum of similarity between $\mathcal{A}$ and its elements.

Now, we propose a method that uses both class- and instance-level stochastic hard sample mining to facilitate visiting diverse examples during the training, while maintaining the hardness for efficiency. At every iteration, we first sample a random anchor class c_a and its corresponding η instances to form $\mathcal{B}_{c_a}$. Different from Alg. 1, we search for a pool of classes, which is α-times larger than the original $(K-1)$ nearest neighbor classes, where α is randomly chosen from $\{3, 4, 5\}$. In particular, we first search for nearest neighbor classes from an anchor class based on distances from samples in the anchor class to the rest of classes, where class signature is employed to represent the classes. It can be formulated as following:

$$\mathcal{P}_c = g(\mathcal{B}_{c_a}, \mathcal{W} \backslash \{\mathbf{w}_{c_a}\}, \alpha(K-1)), \tag{3.14}$$

where $\mathcal{B}_{c_a}$ denotes the set of instances sampled from the anchor class c_a. Here, $g(\mathcal{A}, \mathcal{B}, k)$ is a function that takes sets of vectors $\mathcal{A}$ and $\mathcal{B}$, and k as inputs and outputs k elements from $\mathcal{B}$ with the largest similarity to an element in $\mathcal{A}$ (Alg. 4). We use $\mathcal{P}_c$ as class candidates for the refined search, thereby reducing the number of candidate instances. For a given $\mathcal{P}_c$, to further diversify the training examples, we randomly sample instances among the instance pool, $\mathcal{P}_s$, which is β-times larger than the number of samples

Algorithm 3 Stochastic hard sample mining

```
Parameters $K, \eta, \mathcal{A}, \beta$
for $t$ = 1 : $T$ do
    $\alpha \leftarrow$ Random sample from $\mathcal{A}$
    Random sample an anchor class $c_a$
    $\mathcal{B}_{c_a} \leftarrow$ Sample $\eta$ instances from $\{\mathbf{x}|y_{\mathbf{x}} = c_a\}$
    $\mathcal{B} \leftarrow \mathcal{B}_{c_a}$
    $\mathcal{P}_c \leftarrow g(\mathcal{B}_{c_a}, \mathcal{W}\backslash\{\mathbf{w}_{c_a}\}, \alpha(K-1))$
    $\mathcal{P}_s \leftarrow g(\mathcal{B}_{c_a}, \{\mathbf{x}|y_{\mathbf{x}} = c, c \in \mathcal{P}_c\}, \beta(K-1)\eta)$
    $\mathcal{B}_a \leftarrow$ Random sample $(K-1)\eta$ elements from $\mathcal{P}_s$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{B}_a$
    Perform one iteration of training to minimize the loss Eq.(3.20) using mini-batch $\mathcal{B}$
end for
```

$(K-1)\eta$:

$$\mathcal{P}_s = g(\mathcal{B}_{c_a}, \{\mathbf{x}|y_{\mathbf{x}} = c, c \in \mathcal{P}_c\}, \beta(K-1)\eta). \tag{3.15}$$

In our framework, restricting the search space to the examples in the classes in $\mathcal{P}_c$ significantly reduces the computational cost of hard sample mining. The overall training process is summarized in Alg. 3. stochastic

3.4.3 Scalable Extension to the Number of Classes

Training the class signatures of $|\mathcal{C}|d$ parameters is problematic when there are an extremely large number of classes. To address this problem, we propose a scalable extension of the proposed method by modifying the followings: 1) Class signatures 2) Signature loss 3) Nearest class search for each anchor instance.

Class signatures To make the number of trainable parameters from the class signatures not proportional to the number of classes $|\mathcal{C}|$, our idea is to define a small dictionary $\mathcal{F}$ which consists of J vectors,

$$\mathcal{F} = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_J\}, \tag{3.16}$$

and represent each class as a sum of L different vectors from $\mathcal{F}$. It can represent ${}_JC_L$ different vectors, therefore, we can cover $|\mathcal{C}|$ classes with $J \ll |\mathcal{C}|$. In particular, for each class c, we randomly select L elements from $\{1, 2, \cdots, J\}$ as

$$\mathcal{I}_c = \{i_1^c, i_2^c, \cdots, i_L^c\}. \tag{3.17}$$

Now, the class signature $\mathbf{w}_c$ is represented a function of $\mathcal{B}$ as

$$\mathbf{w}_c(\mathcal{B}) = \sum_{i \in \mathcal{I}_c} \mathbf{b}_i. \tag{3.18}$$

Since the number of parameters of class signatures is reduced to Jd, it is scalable to the number of classes.

Algorithm 4 Select k elements from $\mathcal{B}$ with the largest similarity to an element in $\mathcal{A}$

```
    Input $\mathcal{A}, \mathcal{B} = \{\mathcal{B}_{\mathbf{a}}\}_{\mathbf{a}\in\mathcal{A}}, k$
    Output $\mathcal{N}$
Sort $\mathcal{S} = \cup_{\mathbf{a}\in\mathcal{A}}\{S(\mathbf{a}, \mathbf{b})\}_{\mathbf{a}\in\mathcal{A},\mathbf{b}\in\mathcal{B}_{\mathbf{a}}}$ in the ascending order
$\mathcal{N} \leftarrow$ Unique top-$k$ elements of $\cup_{\mathbf{a}\in\mathcal{A}}\mathcal{B}_{\mathbf{a}}$ in $\mathcal{S}$
```

Signature loss Within a minibatch $\mathcal{B}$, we approximate the original loss of Eq.(5) as

$$l_C(\mathcal{F}, \mathcal{X}) = -\frac{1}{|\mathcal{B}|} \sum_{(\mathbf{x}, y_{\mathbf{x}})\in\mathcal{B}} \log\left(\frac{\exp(S(\mathbf{w}_{y_{\mathbf{x}}}(\mathcal{F}), \mathbf{x}))}{\sum_{c\in\mathcal{C}_{\mathcal{B}}} \exp(S(\mathbf{w}_c(\mathcal{F}), \mathbf{x}))}\right), \tag{3.19}$$

where $\mathcal{C}_{\mathcal{B}}$ denotes the set of classes occurring in a minibatch $\mathcal{B}$. Note that it is calculated over a $|\mathcal{C}_{\mathcal{B}}|$ classes, rather than over the original $|\mathcal{C}|$ classes. Since the batch size $|\mathcal{B}|$ is much smaller than the number of classes in usual and the number of different classes occurring in a batch is not larger than a batchsize, *i.e.* , $|\mathcal{C}_{\mathcal{B}}| < |\mathcal{B}| \ll |\mathcal{C}|$, calculation of Eq.(3.19) is scalable to the number of classes.

Nearest class search In Alg. 3, for each anchor instance from class c, its nearest $\alpha K(\eta - 1)$ classes are found as a candidate class pool $\mathcal{P}_c$. To make it scalable, we first sample M classes from $\mathcal{C}\backslash\{c\}$ to construct a set of random candidate classes $\mathcal{P}_r$. Then, the nearest $\alpha K(\eta - 1)$ classes are searched only from $\mathcal{P}_r$.

Finally, we combine the three modifications to modify the proposed method to be scalable to the number of classes. The algorithm is summarized in Alg. 5.

Algorithm 5 Stochastic hard sample mining

```
Parameters $K, \eta, \mathcal{A}, \beta$
for $t = 1 : T$ do
    $\alpha \leftarrow$ Random sample from $\mathcal{A}$
    Random sample an anchor class $c_a$
    $\mathcal{B}_{c_a} \leftarrow$ Sample $\eta$ instances from $\{\mathbf{x} | y_{\mathbf{x}} = c_a\}$
    $\mathcal{B} \leftarrow \mathcal{B}_{c_a}$
    for $\mathbf{x} \in \mathcal{B}$ do
        $\mathcal{P}_r(\mathbf{x}) \leftarrow$ Random sample $K$ classes from $\mathcal{C} \backslash \{y_{\mathbf{x}}\}$
    end for
    $\mathcal{P}_c \leftarrow h(\mathcal{B}_{c_a}, \{\mathcal{P}_r(\mathbf{x})\}_{\mathbf{x} \in \mathcal{B}}, \alpha(K-1))$
    $\mathcal{P}_s \leftarrow g(\mathcal{B}_{c_a}, \{\mathbf{x} | y_{\mathbf{x}} = c, c \in \mathcal{P}_c\}, \beta(K-1)\eta)$
    $\mathcal{B}_a \leftarrow$ Random sample $(K-1)\eta$ elements from $\mathcal{P}_s$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{B}_a$
    Perform one iteration of training to minimize the loss Eq.(3.19) using mini-batch $\mathcal{B}$
end for
```

3.5 Loss

We jointly train the parameters of feature extractor f and the class signatures $\mathcal{W}$ to minimize both triplet loss and the class signature loss:

$$l(\mathcal{W}, \mathcal{X}) = l_T(\mathcal{X}) + l_C(\mathcal{W}, \mathcal{X}). \tag{3.20}$$

Note that the gradient from the class signature loss l_C back-propagates to the feature extractor. It is known from the existing works that joint triplet and classification loss improves the accuracy from each of them [26, 126]. we tested both with and without back-propagation of the gradient from l_C to the feature extractor f in the experiments.

3.6 Feature Extractor

As a baseline feature extractor f, we closely follow the previous works [26, 56]. We use Inception_v1 [82] from input to the last pooling layer, followed by one batch normalization layer, one fully connected layer, and the final l_2-normalization layer.

Based on it, we design a local-aware model by applying two changes, to enhance the discriminative power of the feature embeddings. First, we found that simply increasing the input image resolution improved the accuracy in the retrieval benchmark datasets. In Table 3.1, R@1 increases as the input resolution grows from 224×224 to 336×336. Compared to the baseline, this change does not increase the number of parameters while taking about 2.25 times more computation in FLOPs. One possible hypothesis for this improvement is that higher resolution image preserves more details than the lower counterpart.

Second, to further reflect the local details to the feature embeddings, we replace the last average pooling to a second-order pooling [20]. For an input feature map $\mathbf{G}$, the second-order pooling is formulated as

$$\text{Pooling}(\mathbf{f}_{xy}) = \frac{1}{S}\sum_{xy} \mathbf{f}_{xy} = \frac{1}{S}\sum_{xy} \text{vec}(\mathbf{g}_{xy} \otimes \mathbf{g}_{xy}), \tag{3.21}$$

Table 3.1: R@1 (%) in CARS-196 and CUB-200-2011 dataset for different feature extractors and different the input sizes

	Method	224×224	336×336
CARS-196	Inception_v1	83.6	89.7
	Local	86.9	91.3
CUB-200-2011	Inception_v1	55.1	60.9
	Local	58.1	65.2

where $\mathbf{g}_{xy}$ denotes a feature vector in $\mathbf{G}$ at position (x, y), $\otimes$ denotes the outer product and $\mathrm{vec}(\cdot)$ vectorizes the input. To enlarge the resolution of the input feature map to the second-order pooling layer, we drop the layers from Inception_v1 *5a* block. More specifically, we use the network from the input to the Inception_v1 *4e* block, followed by a 1×1 convolution (512-dim) and a batch normalization layer to extract the feature map $\mathbf{G}$. Then, the second-order pooling is performed over the extracted feature map followed by l_2-normalization. Table 3.1 shows that the local model based on the second-order pooling (Local) consistently improves the accuracy of the baseline (Inception_v1) which uses average pooling.

3.7 Experiments

3.7.1 Datasets

We evaluated our method on two benchmarks in person re-identification (Market-1501 and DukeMTMC) and four popular benchmarks in image retrieval.

CARS-196 [34] This dataset consists of $16,183$ images of 196 different classes of cars. We used the first 98 classes for training ($8,052$ images) and the other 98 classes for testing ($8,131$ images), following the previous work [74]. For each test identity, two images are randomly sampled as the probe and gallery images and the average accuracy over 20 trials is reported as the final result.

CUB-200-2011 [89] This dataset contains images of 200 different bird species. We used the first 100 classes for training ($5,864$ images) and the other 100 classes for testing ($5,924$ images), following the previous work [74].

In-shop retrieval [35] This dataset consists of $11,735$ classes of clothing items with $54,642$ images. We used $3,997$ classes for training ($25,882$ images) and other $3,985$ classes for testing ($28,760$ images), following the previous work [35]. In the test set, $3,985$ classes with $14,218$ images are used as queries and the remaining $3,985$ classes with $12,612$ images are used as the retrieval database.

Stanford onilne products (SOP) [74] This dataset has $120,053$ product images of $22,634$ classes. $11,318$ classes with $59,551$ images are used for training and $11,316$ classes with $60,499$ images are used for testing.

3.7.2 Implementation Details

We first normalize the images to 256×256 and then perform standard random crops to 224×224 and horizontal flipping for data augmentation. In the baseline batch construction proptocol, the number of triplets seen in each batch is calculated as $O(K\eta^2(K-1)\eta) = O(K^2\eta^3) = O(M^2\eta)$. It indicates that for a fixed batch size, increasing the number of positive samples per each class increases the number of composible triplets. On the other hand, in the extream case when $\eta = M/2$, triplets consists of a limited composition of classes. Therefore, for a batch size $M = 60$, we choose $\eta = 10$ for the small datasets (CARS-196 and CUB-200-2011), which has enough number of samples per class, and $\eta = 5$ for the larger datasets (In-shop retrieval and Stanford Online Products), which has 5 examples per each class.

Network architecture For the feature extractor, we initialize the parameters with GoogLeNet [82], which was pretrained on the ImageNet ILSVRC dataset [66], and randomly initialize an added fully connected layer. We fix the feature dimension to 512 for all the experiments.

Optimization We use stochastic gradient descent for optimization with batchsize 60. The initial learning rate, weight decay, and the momentum are set to 0.001, 5×10^{-3}, and 0.9, respectively. The learning rate is decreased by a factor of 5 after every 200 epochs.

Compact bilinear pooling The bilinear transformation over the two 512-dimensional feature vectors result in an extremely high dimensional vector, which consumes large computational cost and memory. To resolve this issue, we use the tensor sketch approach [62] to compute a compact representation as in [20]. The key idea of the tensor sketching is to reduce dimensionality of two high-dimensional vectors and compute their inner-product efficiently, which is performed implicitly in a unified framework. Refer to [62] for the details.

3.7.3 Evaluation Metrics

For the evaluation metric, we use Recall@K (R@K). For Recall@K, all the embeddings of test samples are first extracted. And then, for each sample, K nearest neighbors are retrieved from the remaining test set. If retrieved images include at least one sample from the same class, it is considered to be correct. The Recall@K metric measures the number of correct sample over entire sample. For the distance measure, Euclidean distance is used, which is equivalent to the cosine distance in our case, because the feature are l_2-normalized.

Table 3.2: Recall@K (%) comparison with the baseline

		CARS-196			
K		1	2	4	8
Smartmining [26]		64.7	76.2	84.2	90.2
Inception_v1	Baseline [116]	78.2	86.0	90.9	94.2
	Class Mining (Alg. 1)	81.3	87.8	92.6	95.6
	Stochastic Mining (Alg. 3, *var1*)	81.3	88.3	92.3	95.5
	Stochastic Mining (Alg. 3, *var2*)	82.5	89.2	93.4	96.2
	Stochastic Mining (Alg. 3)	83.4	89.9	93.9	96.5

Table 3.3: Recall@K (%) comparison with the baseline

		CUB-200-2011			
K		1	2	4	8
Smartmining [26]		49.8	62.3	74.1	83.3
Inception_v1	Baseline [116]	52.4	64.4	74.9	84.2
	Class Mining (Alg. 1)	52.9	64.8	75.6	84.1
	Stochastic Mining (Alg. 3, *var1*)	54.1	66.3	76.7	84.8
	Stochastic Mining (Alg. 3, *var2*)	55.1	66.4	76.2	84.8
	Stochastic Mining (Alg. 3)	56.0	68.3	78.2	86.3

Table 3.4: Recall@K (%) comparison with the baseline

		In-shop retrieval			
	K	1	10	20	30
Inception_v1	Baseline [116]	86.4	96.5	97.9	98.4
	Class Mining (Alg. 1)	88.0	96.7	97.8	98.3
	Stochastic Mining (Alg. 3, *var1*)	87.3	96.3	97.4	97.9
	Stochastic Mining (Alg. 3, *var2*)	89.0	97.3	98.1	98.6
	Stochastic Mining (Alg. 3)	90.8	98.0	98.6	98.9

Table 3.5: Recall@K (%) comparison with the baseline

		Stanford online products			
	K	1	10	10^2	10^3
Inception_v1	Baseline [116]	67.8	84.0	93.2	97.9
	Class Mining (Alg. 1)	70.6	84.9	93.1	97.7
	Stochastic Mining (Alg. 3, *var1*)	68.7	82.4	90.8	96.2
	Stochastic Mining (Alg. 3, *var2*)	72.1	85.9	93.3	97.6
	Stochastic Mining (Alg. 3)	75.2	87.5	93.7	97.4

3.7.4 Effect of the Stochastic Hard Example Mining

Image retrieval datasets We evaluate the proposed stochastic hard example mining method in two aspects: hardness and diveristy of the mined triplets. Figure 3.8 shows the comparison of the number of triplets of non-zero loss, occuring in each minibatch during the training. Compared to the random sampling [116], class-level mining (Alg. 1) and the stochastic mining (Alg. 3) result in far more triplets that have non-zero loss. Comparing the class-mining and the stochastic mining, the latter results in about 2-5 times more triplets with non-zero loss thanks to the instance-level refinement.

Figure 3.4 compares the diversity of the mined triplets between class-mining and the stochastic mining. For a given anchor class c_a, the change of chosen negative classes in each minibatch is illustrated. The x-axis and y-axis correspond to the training iteration and class index, respectively. At iteration t, the corresponding column illustrates all the negative classes used to construct a minibatch, i.e. 1 if it is chosen and 0 otherwise. It shows that the combination of classes of seen triplet varies over the iteration in Alg. 3 (b), while they are mostly fixed in Alg. 1 (a). It verifies that the proposed method effectively diversify the examples shown during the training.

To evaluate the effect of the proposed stochastic hard example mining, we compare our method (Alg. 3) with a baseline protocol [116], class mining (Alg. 1), and two variants of ours. Table 3.2–3.9 show the accuracy comparision on four popular datasets: CARS-196, CUB-200-2011, In-shop retrieval, and SOP. It is known that training the feature extractor with joint loss of triplet and classification enhances the accuracy from the baseline which uses only triplet loss. To discriminate the effect of the hard sample mining and the addition of loss (Eq. 3.4), we show both results, only with mining (*var2*) and the full model. In *var2*, the class signature loss does not back-propagate to the feature extractor. Another variant (*var1*) replaces the proposed class signature with the class-wise average of features, extracted from the Inception_v1 pretrained on the ImageNet dataset. Table 3.2–3.9 show that the proposed method consistently improves

Table 3.6: Recall@K (%) comparison with the baseline

	CARS-196			
K	1	2	4	8
Smartmining [26]	64.7	76.2	84.2	90.2
Baseline [116]	78.2	86.0	90.9	94.2
Scalable Mining (Alg. 5)	**80.4**	**87.7**	**92.4**	**95.6**

the accuracy compared to the random sampling baseline in all four datasets.

Figure 3.5–3.7 compares the minibatches constructed using different batch construction methods on the In-shop retrieval dataset. For a given anchor and its sampled instances in the first row, the remaining images show the selected negative examples. Compared to the random sampling baseline (Figure 3.5), which selects diverse and easily distinguishable negative examples, the hard class mining (Figure 3.6) samples classes that are more likely to be confused with the anchor class. For example, random sampling contains pants as a negative class, which is easily discriminated and does not contribute to the training with zero loss, while hard class mining selects only tops which mostly consist of the upper body images. Finally, the proposed hard negative mining (Figure 3.7) chooses more similar images in an instance level. Each negative instances are selected as the nearest neighbor of an anchor instance with color and type of the clothes, thereby generating a large number of hard triplets to be used for training.

The evaluation of the scalable extension is shown in Table. 3.6-3.8. It shows that the proposed scalable extension (*Scalable Mining*) consistently improves the result from the random sampling baseline on the four popular datasets for retrieval.

Person re-identification datasets We also evaluate out method on two popular person re-identification datasets: Market-1501 and DukeMTMC. In our experiments, backpropagating the class signature loss L_C to the feature extractor f lowered the accu-

Table 3.7: Recall@K (%) comparison with the baseline

	CUB-200-2011			
K	1	2	4	8
Smartmining [26]	49.8	62.3	74.1	83.3
Baseline [116]	52.4	64.4	74.9	84.2
Scalable Mining (Alg. 5)	**54.5**	**66.2**	**77.2**	**85.4**

Table 3.8: Recall@K (%) comparison with the baseline

	In-shop retrieval			
K	1	10	20	30
Baseline [116]	86.4	96.5	97.9	98.4
Scalable Mining (Alg. 5)	**88.2**	**97.2**	**98.1**	**98.5**

Table 3.9: Recall@K (%) comparison with the baseline

	Stanford online products			
K	1	10	10^2	10^3
Baseline [116]	67.8	84.0	93.2	97.9
Scalable Mining (Alg. 5)	**70.6**	**85.6**	**93.5**	–

Table 3.10: Recall@K (%) comparison with the baseline

	Market-1501				DukeMTMC			
K	1	5	10	20	1	5	10	20
Baseline [116]	83.4	92.9	95.5	97.1	72.8	84.4	88.9	91.6
Stochastic Mining (Alg. 3, *var2*)	83.7	92.8	95.4	97.3	73.2	85.6	89.0	91.5

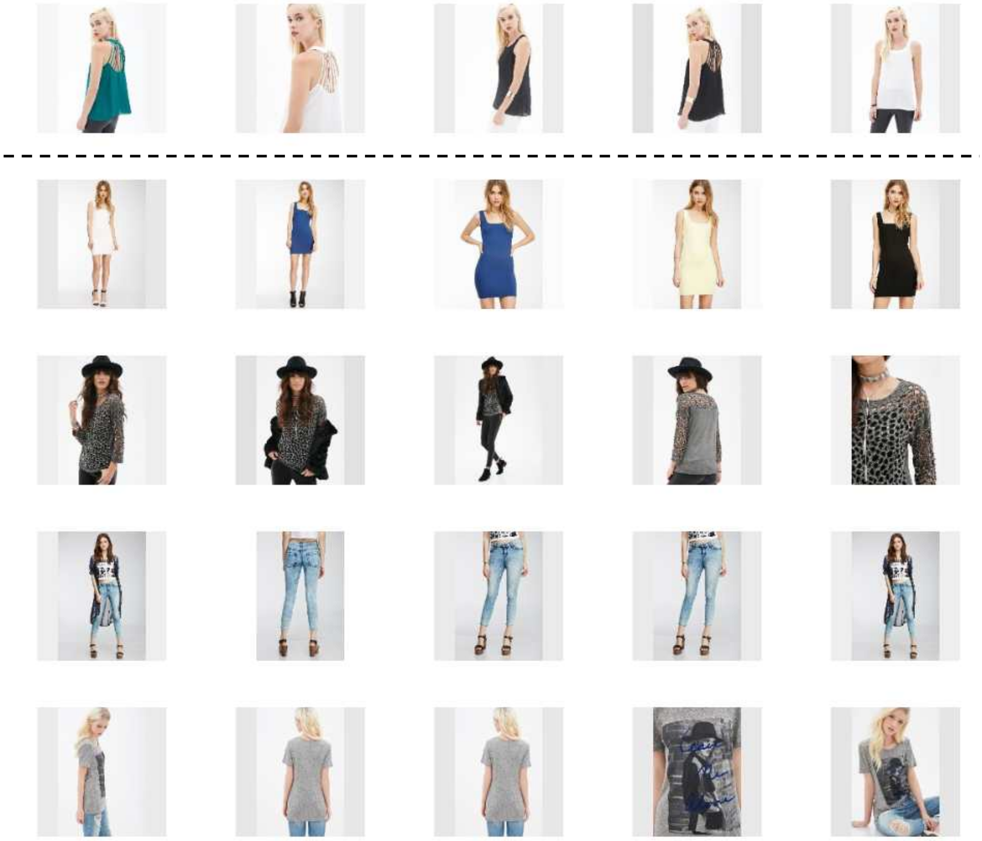


Figure 3.5: A minibatch example from the In-shop retrieval dataset constructed using the random sampling. For a given anchor class and its instances in the first row, selected negative instances are shown in the remaining rows.

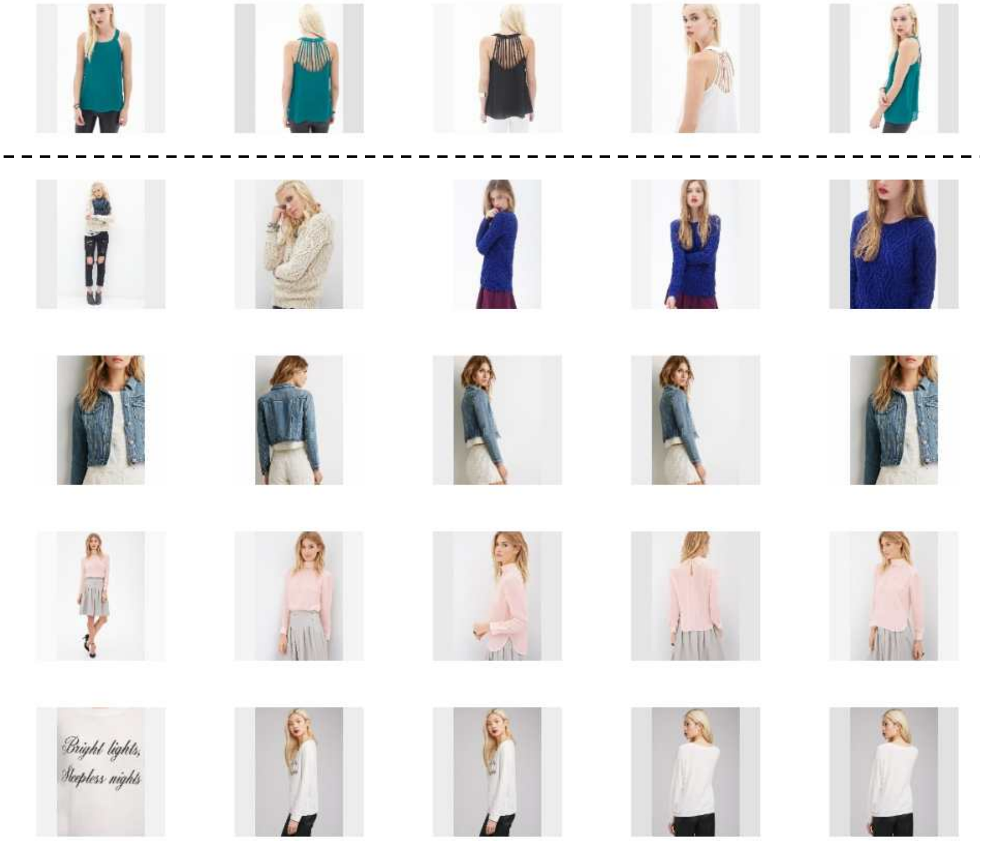


Figure 3.6: A minibatch example from the In-shop retrieval dataset constructed using the hard class mining (Alg. 1). For a given anchor class and its instances in the first row, selected negative instances are shown in the remaining rows.

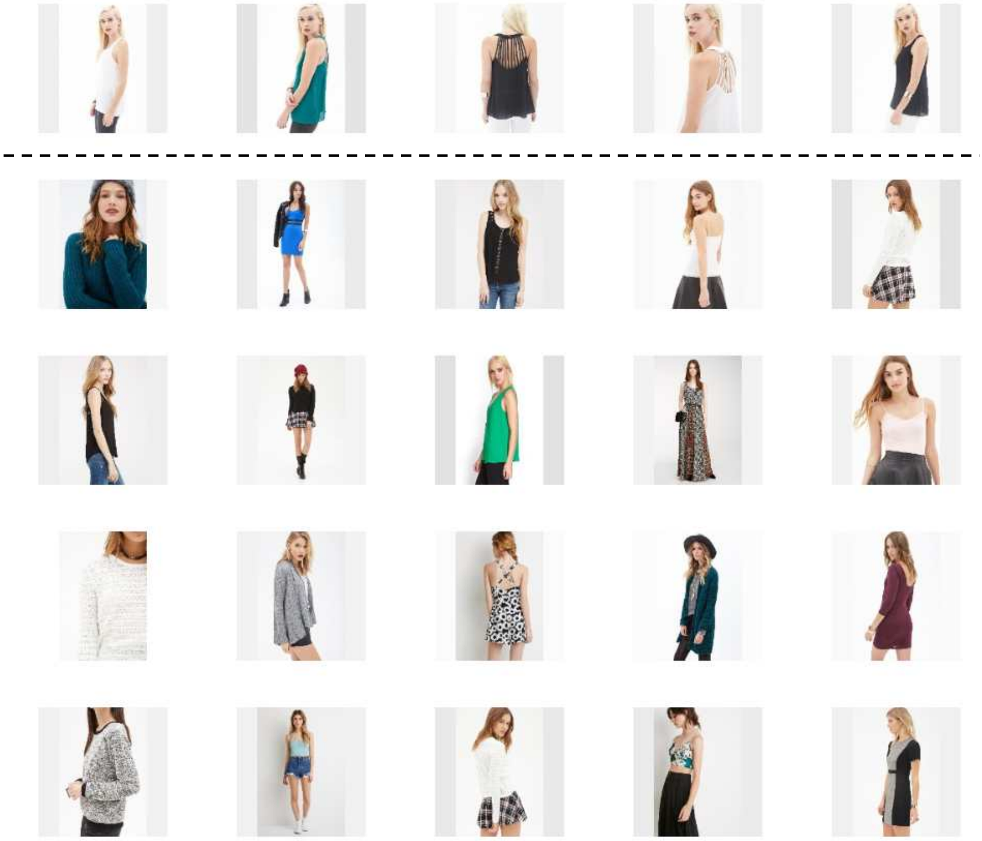


Figure 3.7: A minibatch example from the In-shop retrieval dataset constructed using the hard negative mining (Alg. 3). For a given anchor class and its instances in the first row, selected negative instances are shown in the remaining rows.

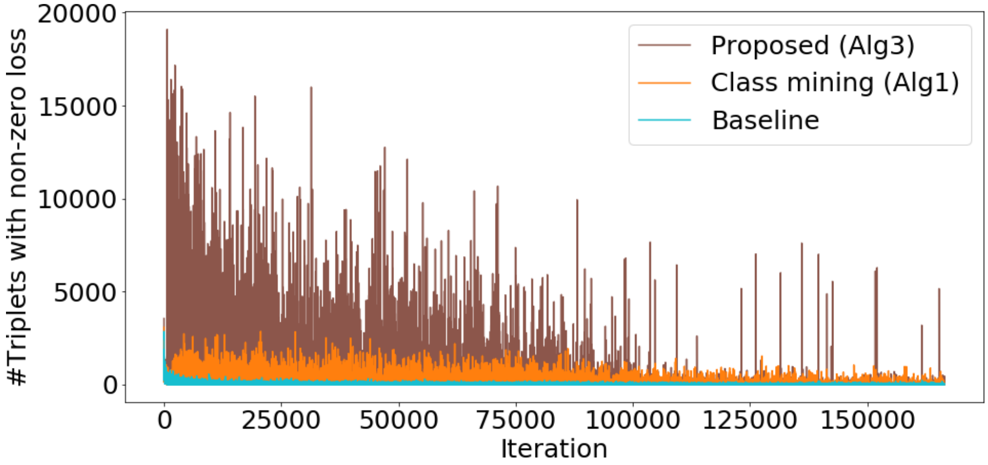


Figure 3.8: The comparison of the number of triplets of non-zero loss in each minbatch during the training.

racy for the person re-identification task. Therefore, we used *var2* for the person re-identifciation task. The results are shown in Table 3.10. It shows that the propsoed method improves the accuracy on both datasets.

3.7.5 Comparison with the Existing Methods on Image Retrieval Datasets

We compare our method to the state-of-the-art methods in Table.3.11–3.14. Since the backbone network architecture affects the retrieval accuracy, we show the architecture in the parenthesis. Overall, when compared to the previous works which use the same network architecture (Incetpion_v1), our method achieves the best accuracy in all datasets (underlined), except for the SOP, where we achieved the second best. When compared to the existing hard sample mining method [26], ours achieve higher accuracy. Note that SmartMining only provided their result in small datasets. We also report the accuracy of proposed mining method applied to the local-aware model (+local). In every dataset, our method outperforms the previous state-of-the-art with comparable compuational cost.

Table 3.11: Accuracy comparison on CARS-196

Method	R@1	R@2	R@4	R@8
Lifted [74] (Inception_v1)	53.0	66.7	76.0	84.3
Facility [73] (Inception_v1)	58.1	70.6	80.3	87.8
SmartMining [26] (Inception_v1)	64.7	76.2	84.2	90.2
N-pair [72] (Inception_v1)	71.1	79.7	86.5	91.6
Angular [95] (Inception_v1)	71.4	81.4	87.5	92.1
Proxy NCA [56] (Inception_v1)	73.2	82.4	86.4	88.7
HDC [109] (Inception_v1 + ensemble)	73.7	83.2	89.5	93.8
DAML [17] (Inception_v1)	75.1	83.8	89.7	93.5
HTG [118] (Inception_v1 + att)	76.5	84.7	90.4	94.0
Margin [99] (ResNet-50)	79.6	86.5	91.9	95.1
HTL [22] (BN-Inception)	81.4	88.0	92.7	95.7
A-Bier [58] (Inception_v1 + ensemble)	82.0	89.0	93.2	96.1
ABE [32] (Inception_v1 + ensemble)	85.2	90.5	93.9	96.1
DREML [106] (Inception_v3 + ensemble)	84.2	89.4	93.2	95.5
DREML [106] (ResNet-18 + ensemble)	86.0	91.7	95.0	97.2
Hard negative mining (Inception_v1)	83.4	89.9	93.9	96.5
Hard negative mining + Local	**91.7**	**95.3**	**97.3**	**98.4**

Table 3.12: Accuracy comparison on CUB-200-2011

Method	R@1	R@2	R@4	R@8
SmartMining [26] (Inception_v1)	49.8	62.3	74.1	83.3
Proxy NCA [56] (Inception_v1)	49.2	61.9	67.9	72.4
N-pair [72] (Inception_v1)	51.9	64.3	74.9	83.2
DAML [17] (Inception_v1)	52.7	65.4	75.5	84.3
HDC [109] (Inception_v1 + ensemble)	53.6	65.7	77.0	85.6
Angular [95] (Inception_v1)	54.7	66.3	76.0	83.9
HTL [22] (BN-Inception)	57.1	68.8	78.7	86.5
A-Bier [58] (Inception_v1 + ensemble)	57.5	68.7	78.3	86.2
HTG [118] (Inception_v1 + att)	59.5	71.8	81.3	88.2
ABE [32] (Inception_v1 + ensemble)	60.6	71.5	79.8	87.4
Margin [99] (ResNet-50)	63.6	74.4	83.1	90.0
DREML [106] (inception_v3 + ensemble)	58.9	69.6	78.4	85.6
DREML [106] (ResNet-18 + ensemble)	63.9	75.0	83.1	89.7
Hard negative mining (Inception_v1)	56.0	68.3	78.2	86.3
Hard negative mining + Local	**66.2**	**76.3**	**84.1**	**90.1**

Table 3.13: Accuracy comparison on In-shop retrieval

Method	R@1	R@10	R@20	R@30
HDC [109] (Inception_v1 + ensemble)	62.1	84.9	89.0	91.2
DREML [106] (ResNet-18 + ensemble)	78.4	93.7	95.8	96.7
HTG [118] (Inception_v1 + att)	80.3	93.9	95.8	96.6
HTL [22] (BN-Inception)	80.9	94.3	95.8	97.2
A-Bier [58] (Inception_v1 + ensemble)	83.1	95.1	96.9	97.5
ABE [32] (Inception_v1 + ensemble)	87.3	96.7	97.9	98.2
Hard negative mining (Inception_v1)	<u>89.9</u>	<u>97.4</u>	<u>98.2</u>	<u>98.6</u>
Hard negative mining + Local	**91.9**	**98.0**	**98.7**	**99.0**

Table 3.14: Accuracy comparison on Stanford online products

Method	R@1	R@10	R@10^2	R@10^3
N-pair [72] (Inception_v1)	66.4	83.2	93.0	–
DAML [17] (Inception_v1)	68.4	83.5	92.3	–
HDC [109] (Inception_v1 + ensemble)	69.5	84.4	92.8	97.7
Margin [99] (ResNet-50)	72.7	86.2	93.8	98.0
Proxy NCA [56] (Inception_v1)	73.7	–	–	–
A-Bier [58] (Inception_v1 + ensemble)	74.2	86.9	94.0	97.8
HTL [22] (BN-Inception)	74.8	88.3	**94.8**	**98.4**
ABE [32] (Inception_v1 + ensemble)	76.3	88.4	**94.8**	98.2
Hard negative mining (Inception_v1)	73.4	86.3	93.1	97.0
Hard negative mining + Local	**76.6**	**88.7**	94.6	97.9

3.8 Summary

We proposed a scalable hard class mining method for triplet loss. The proposed method avoids heavy computational cost to mine hard samples by learning a set of class signatures and estimating the neighbor of the feature embedding based on them. In particular, we propose a stochastic batch construction framework which diversifies the examples seen during the training while keeping them difficult enough for efficient training. Experimental results show that our method consistently improves the baseline.

Chapter 4

Integrated System for Person Re-identification

4.1 Introduction

In the previous chapter, we focused on designing a general batch construction method based on the hard negative mining. Although it consistently improved the accuracy from the baseline in the image retrieval benchmarks, it showed relatively small accuracy gain for the person re-identification task (Section 3.7.4). In this chapter, we propose a batch construction method while focusing on improving the person re-identification performance. In particular, we propose a hard positive mining method that encourages the batch constructor to contain not only hard negatives but also the hard positives. Since the performance of person re-identification heavily relies on the ability to identifying the same person with large viewpoint or pose difference, hard positive mining is specifically effective on the person re-identification task.

Finally, based on the updated batch constructor, we propose an integrated person re-identification system by combining it with the part-aligned feature extractor. The experiments show that the integrated system improves the performance from each of them, achieving the state-of-the-art accuracy in the two popular person re-identification benchmarks: Market-1501 and DukeMTMC.

4.2 Hard Positive Mining

In Alg. 3, there is no constraint on the number of samples per each class within a minibatch. As a result, in an extreme case when every negative samples come from different classes, the positive pairs can be generated only from the anchor class. Since a triplet consists of one positive and negative with respect to an anchor, the number of positive pair is an upper-bound of the number of triplets, *i.e.* η^2. Compared to the class mining baseline which has $K(K-1)\eta^3$ triplets, it severely reduces the number of triplets composable from each minibatch and results in inefficient training.

To eliminate this potential drawback and stabilize the training, we propose a variant of Alg. 3. To make the number of hard positive pairs within each minibatch to be large enough, we explicitly enforce the number of samples per each class to η. In addition, to construct each minibatch with harder examples, we formulate the problem of hard positive sampling as a k-center problem [29], which selects k most diverse elements from the input set $\mathcal{P}$, *i.e.* $\mathcal{S} = g_{kcenter}(\mathcal{P}, k)$. It is formulated as the following optimization problem:

$$\begin{aligned} &\text{maximize} \quad \min_{\mathbf{p}\in\mathcal{P}} s(\mathbf{p}, \mathcal{S}) \\ &\text{s.t.} |S| = k \end{aligned} \tag{4.1}$$

For person re-identification, the major difficulty comes from the large intra class variation from viewpoint / pose change compared to the small inter class differences. Therefore, seeing an enough number of hard positive pairs is critical for training an distinctive feature extractor. Meanwhile, outliers are not likely to be frequently sampled by random sampling because their ratio is relatively small compared to the regular detections. With this formulation, outlier instances are sampled every time with a high probability for each class.

Algorithm 6 Greedy k-Center [29]

```
    **Inputs** $\mathcal{P}, k$
    **Outputs** $\mathcal{S}$
$\mathcal{S} \leftarrow$ Random sample from $\{1, \cdots, k\}$
**for** $\mathbf{p} \in \mathcal{P}$ **do**
    $s[\mathbf{p}] \leftarrow -1$
**end for**
$\mathcal{S} \leftarrow \phi$
**for** $i = 1 : k$ **do**
    $\mathbf{u} \leftarrow \arg\min_{\mathbf{p}} s[\mathbf{p}]$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathbf{u}$
    **for** $\mathbf{p} \in \mathcal{P}$ **do**
        $s[\mathbf{p}] \leftarrow \max(s[\mathbf{p}], S(\mathbf{p}, \mathbf{u}))$
    **end for**
**end for**
```

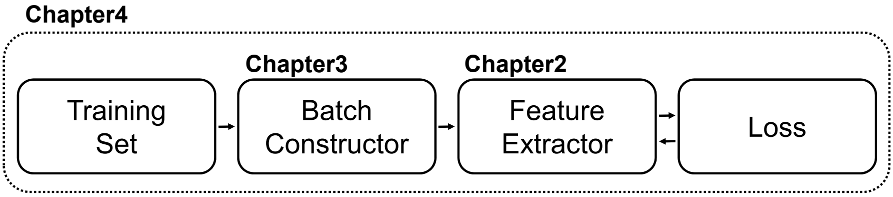


Figure 4.1: Overview of the training process of the integrated system

Algorithm 7 Stochastic hard sample mining

```
Parameters $K, \eta, \mathcal{A}, \beta$
for $t$ = 1 : $T$ do
    $\alpha \leftarrow$ Random sample from $\mathcal{A}$
    Random sample an anchor class $c_a$
    if rand $> 0.5$ then
        $\mathcal{B}_{c_a} \leftarrow$ Sample $\eta$ instances from $\{\mathbf{x} | y_{\mathbf{x}} = c_a\}$
    else
        $\mathcal{B}_{c_a} \leftarrow g_{kcenter}(\{\mathbf{x} | y_{\mathbf{x}} = c_a\}, \eta)$
    end if
    $\mathcal{B} \leftarrow \mathcal{B}_{c_a}$
    $\mathcal{P}_c \leftarrow g(\mathcal{B}_{c_a}, \mathcal{W} \backslash \{\mathbf{w}_{c_a}\}, \alpha(K-1))$
    $\mathcal{P}_s \leftarrow g(\mathcal{B}_{c_a}, \{\mathbf{x} | y_{\mathbf{x}} = c, c \in \mathcal{P}_c\}, \beta(K-1)\eta)$
    $\mathcal{C} \leftarrow$ Random sample $K - 1$ instances from $\mathcal{P}_s$
    for $\mathbf{c} \in \mathcal{C}$ do
        $\mathcal{B} \leftarrow \mathcal{B} \cup g_{kcenter}(\{\mathbf{x} | y_{\mathbf{x}} = y_{\mathbf{c}}\}, \eta)$ with intial point $\mathbf{c}$
    end for
    Perform one iteration of training to minimize the loss Eq.(3.20) using mini-batch $\mathcal{B}$
end for
```

Table 4.1: Recall@K (%) comparison with the baseline (Inception_v1)

	Market-1501				DukeMTMC			
K	1	5	10	20	1	5	10	20
Baseline	83.4	92.9	95.5	97.1	72.8	84.4	88.9	91.6
Hard negative mining (Alg. 3)	83.7	92.8	95.4	97.3	73.2	85.6	89.0	91.5
Hard positive mining	83.8	93.0	95.6	97.3	74.1	85.2	89.0	91.9
Hard positive mining + *stochastic* (Alg. 7)	83.7	92.8	95.2	97.3	74.6	85.2	89.5	92.0

4.3 Integrated System for Person Re-identification

Finally, we propose an integrated system for person re-identification by combining the part-aligned feature extractor (Chapter 2) and hard example mining technique (Chapter 3) as shown in Figure 4.1. Hard positive mining technique (Section 4.2) can be optionally added to the batch constructor to further enhance the performance of person re-identification.

4.4 Experiments

4.4.1 Comparison with the baselines

We evaluate the effect of each components, hard positive sample mining and stochastic mining. Table 4.1 shows the accuracy comparison of the proposed method. Here, the baseline is a random sampling method and the hard negative mining refer to the Alg. 3 proposed in Chapter 3. Compared to the case when using only hard negative mining, additional hard positive mining improves the accuracy in both datasets. Stochastically varying the strategy for anchor class and the instance selection (+ *stochastic*) further improves the accuracy.

Figure 4.2–4.5 compares the minibatches constructed using different batch construction methods on the Market-1501 dataset. For a given anchor and its sampled

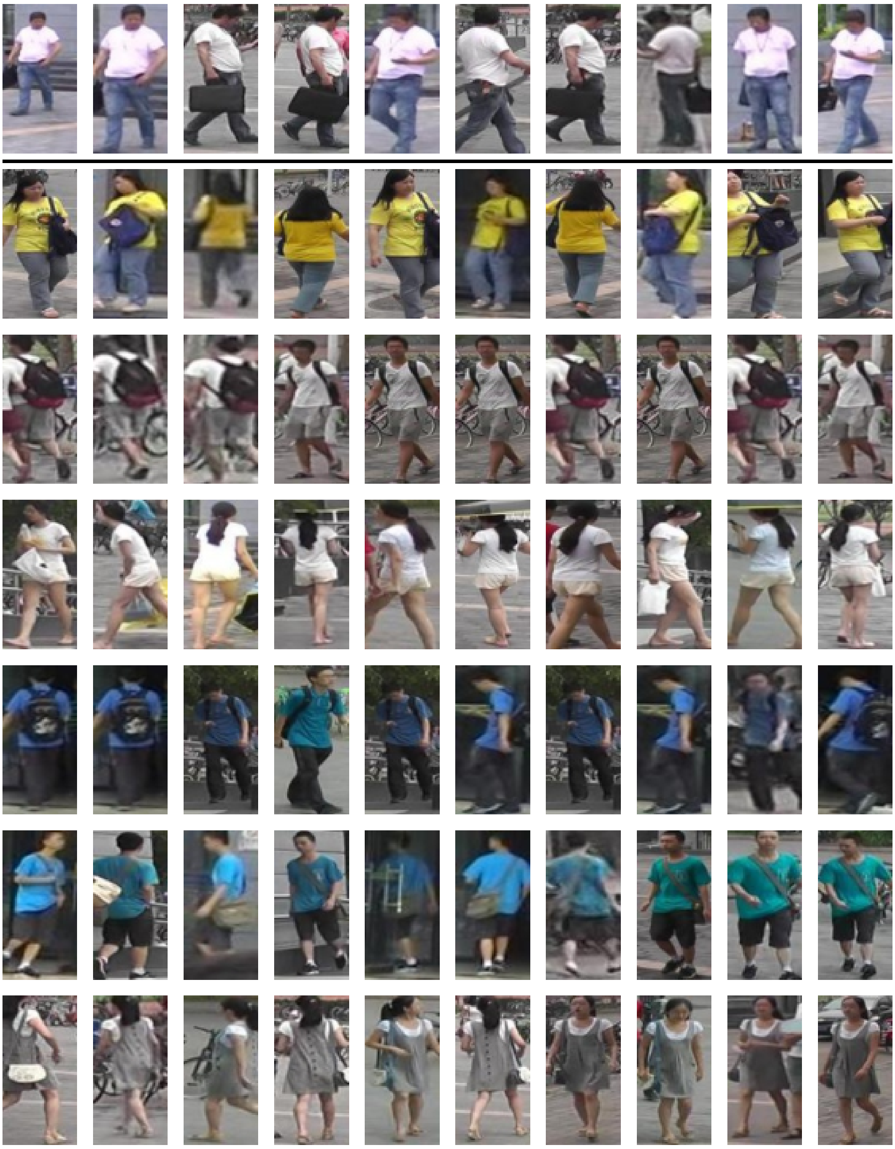


Figure 4.2: A minibatch example from the Market-1501 dataset constructed using the random sampling. For a given anchor class and its instances in the first row, selected negative instances are shown in the remaining rows.

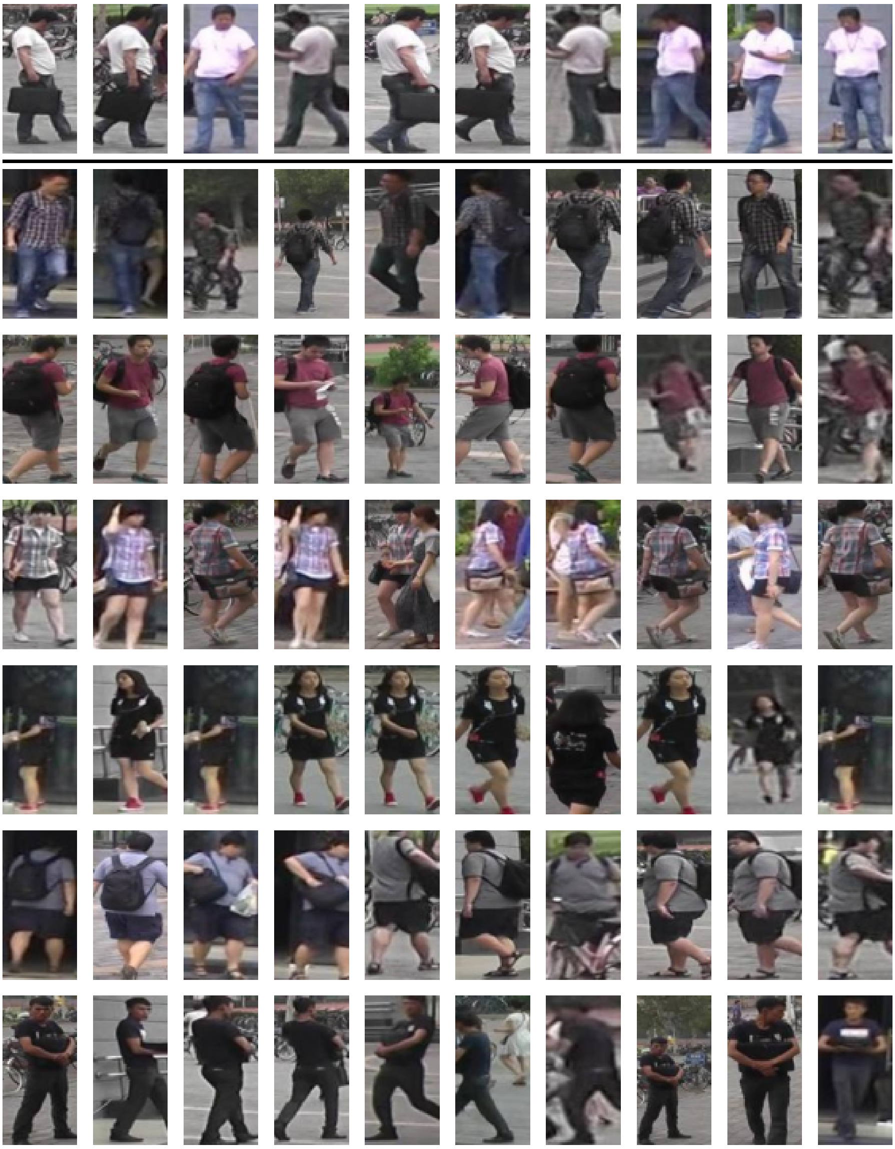


Figure 4.3: A minibatch example from the Market-1501 dataset constructed using the hard class mining (Alg. 1). For a given anchor class and its instances in the first row, selected negative instances are shown in the remaining rows.

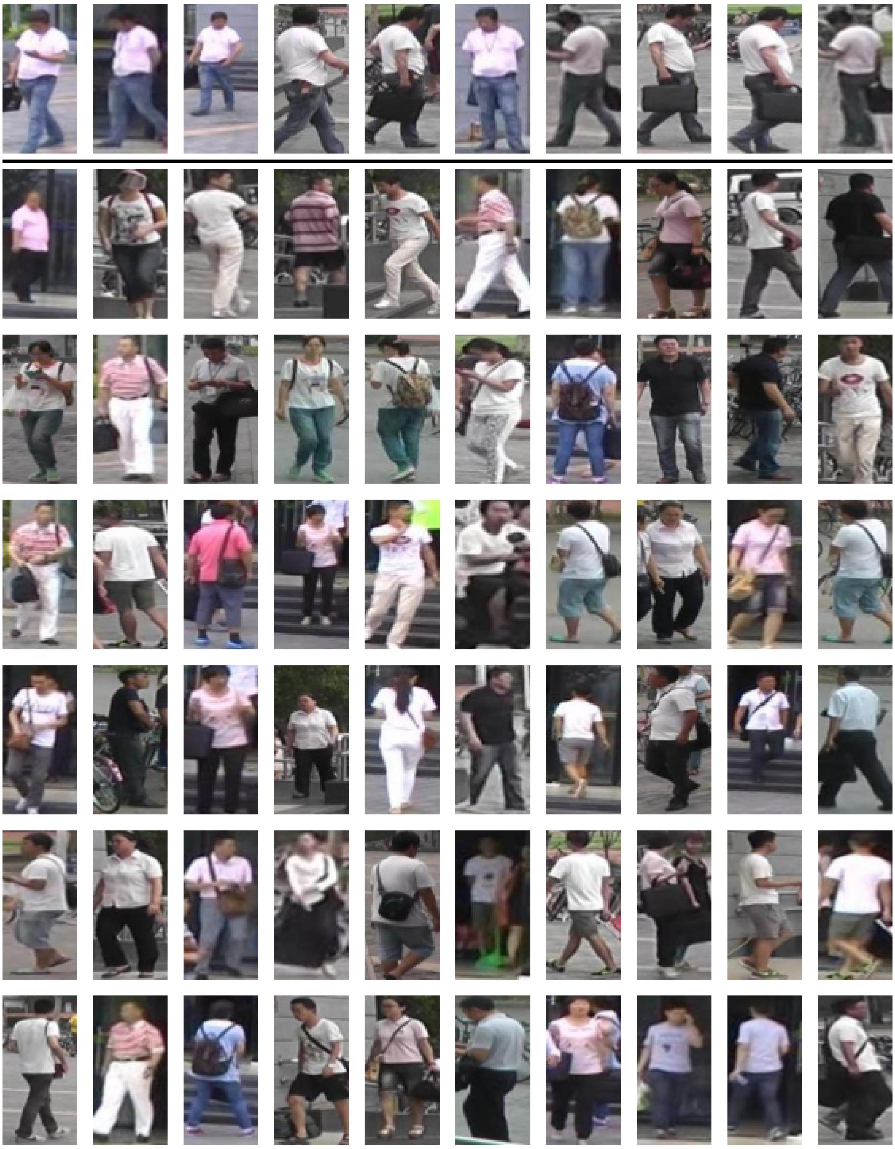


Figure 4.4: A minibatch example from the Market-1501 dataset constructed using the hard negative mining (Alg. 3). For a given anchor class and its instances in the first row, selected negative instances are shown in the remaining rows.

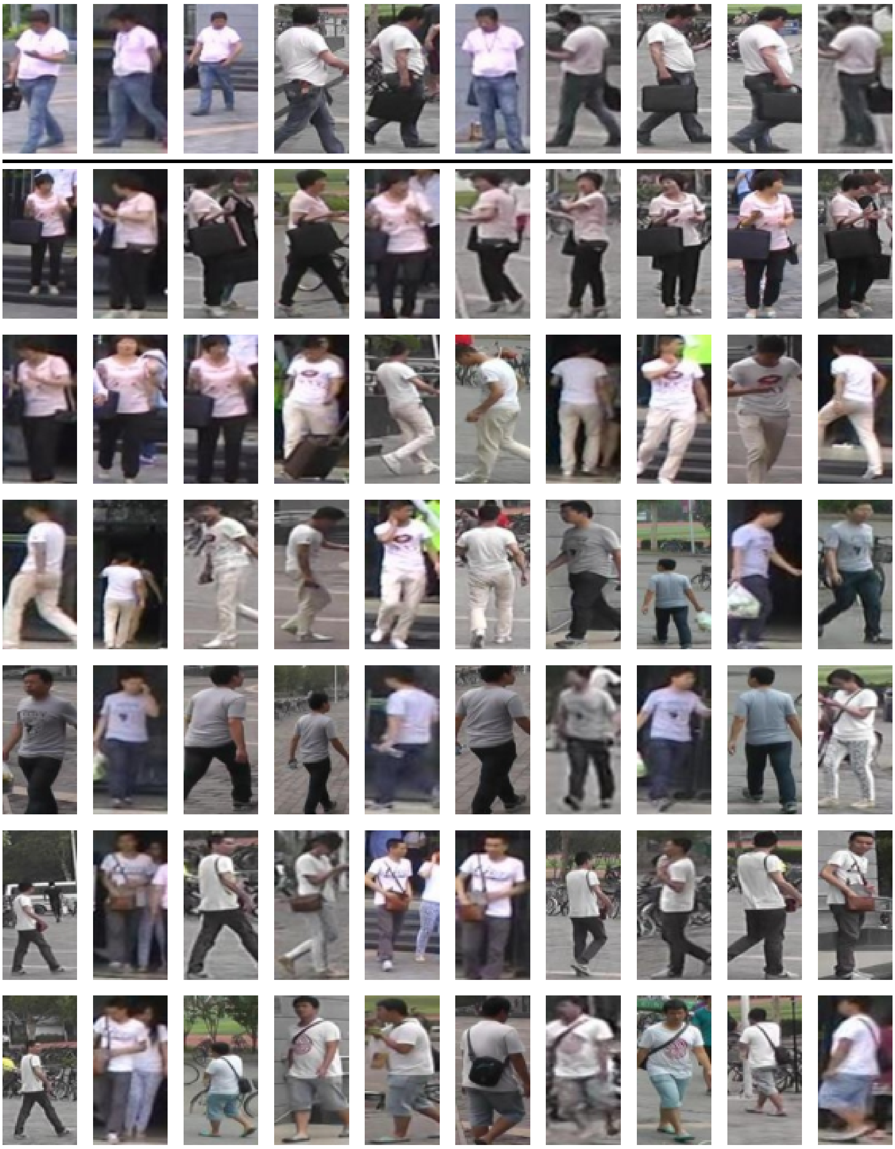


Figure 4.5: A minibatch example from the Market-1501 dataset constructed using the hard positive mining (Alg. 5). For a given anchor class and its instances in the first row, selected negative instances are shown in the remaining rows.

instances in the first row, the remaining images show the selected negative examples. Compared to the random sampling baseline (Figure 4.2), which selects diverse and easily distinguishable negative examples, the hard class mining (Figure 4.3) samples classes that are more likely to be confused with the anchor class. For example, as the person in the anchor class is wearing pink shirts, people wearing shirts in similar colors (pink, red, and gray) are selected as the negative classes in Figure 4.3. Figure 4.4 shows that the proposed hard negative mining chooses more similar images in an instance level. However, it shows that the negative examples are from too diverse classes and there are only one or two samples for each negative person. When the number of positive pairs composable from a minibatch is small, the training becomes inefficient because the number of triplets used for training becomes correspondingly small. Figure 4.5 shows that the proposed hard positive mining chooses more similar images in an instance level while keeping the number of samples per each person to be larger than a certain threshold. It keeps the number of composable triplets within each minibatch large enough, making the training efficient.

4.4.2 Comparison with the existing works

Table 4.2 and 4.3 show the accuracy compared with the existing works on the Market-1501 and MARS dataset, respectively. Our integrated method further improves the result from the part-aligned model and achieves the state-of-the-art accuracy.

4.5 Summary

In this chapter, we proposed a hard positive mining method. By combining two independent approaches proposed in Chapter 2 and Chapter 3 with the new hard positive mining method, we propose an integrated person re-identification. Experimental results show that the components are orthogonal to each other, and the integrated system improves the accuracy from each of them.

Table 4.2: Accuracy comparison on Market-1501

	Sinlge Query				
Rank	1	5	10	20	mAP
Varior et al. 2016 [87]	61.6	-	-	-	35.3
Zhong et al. 2017 [128]	77.1	-	-	-	63.6
Zhao et al. 2017 [115]	80.9	91.7	94.7	96.6	63.4
Sun et al. 2017 [81]	82.3	92.3	95.2	-	62.1
Geng et al. 2016 [23]	83.7	-	-	-	65.5
Lin et al. 2017 [45]	84.3	93.2	95.2	97.0	64.7
Bai et al. 2017 [3]	82.2	-	-	-	68.8
Chen et al. 2017 [12]	72.3	88.2	91.9	95.0	-
Hermans et al. 2017 [27]	84.9	94.2	-	-	69.1
Zhang et al. 2017 [113]	87.7	-	-	-	68.8
Zhong et al. 2017 [129]	87.1	-	-	-	71.3
Chen et al. 2017 [11] (MobileNet)	90.0	-	-	-	70.6
Chen et al. 2017 [11] (Inception-V3)	88.6	-	-	-	72.6
Ustinova et al. 2017 [85] (Bilinear)	66.4	85.0	90.2	-	41.2
Zheng et al. 2017 [120] (Pose)	79.3	90.8	94.4	96.5	56.0
Zhao et al. 2017 [114] (Pose)	76.9	91.5	94.6	96.7	-
Su et al. 2017 [75] (Pose)	84.1	92.7	94.9	96.8	65.4
Part-aligned (Inception-V1, OpenPose)	90.2	96.1	97.4	98.4	76.0
+ dilation	91.7	**96.9**	**98.1**	**98.9**	**79.6**
+ dilation + hard sample mining	**92.6**	**96.7**	**98.0**	**98.6**	**79.4**

Table 4.3: Accuracy comparison on MARS

Rank	1	5	10	20	mAP
Xu et al. [104] (Video)	44	70	74	81	-
McLaughlin et al. [55] (Video)	45	65	71	78	27.9
Zheng et al. [119] (Video)	68.3	82.6	-	89.4	49.3
Liu et al. [47] (Video)	68.3	81.4	-	90.6	52.9
Zhou et al. [130]	70.6	90.0	-	97.6	50.7
Li et al. [36]	71.8	86.6	-	93.1	56.1
Liu et al. [50]	73.7	84.9	-	91.6	51.7
Hermans et al. [27]	79.8	91.4	-	-	67.7
Part-aligned (Inception V1, OpenPose)	83.0	92.8	95	96.8	72.2
+ dilation	83.1	94.2	95.8	–	74.8
+ dilation + hard sample mining	**83.8**	**94.4**	**96.2**	–	**74.9**

Chapter 5

Conclusion

5.1 Contributions

In this thesis, we proposed a person re-identification system that improves the performance of the baseline in two aspects: 1) A better image representation model using human poses and 2) an effective training strategy using hard sample mining.

First, we proposed a novel way of using the human pose to solve person re-identification problem. In particular, we represented parts using part maps, differently from the previously work which used box-based representations [75, 120, 75, 5, 114], and use bilinear pooling to obtain a part-aligned representation. The part maps are learned to minimize the re-identification loss with the guidance of the pre-trained pose estimation model. The learnt part map representation provides a fine-grained/robust differentiation of the body part depending on their usefulness for re-identification and effectively handles the body part misalignment problem. Second, we proposed a scalable hard class mining method for triplet loss. The proposed method avoids heavy computational cost to mine hard samples by learning a set of class signatures and estimating the neighbor of the feature embedding based on them. In particular, we propose a stochastic batch construction framework which diversifies the examples seen during the training while keeping them difficult enough for efficient training. Finally, the combined system shows the promising performance on the popular benchmarks of person re-identification.

5.2 Future Works

First, extension of the proposed part-aligned representation to generic object is an interesting direction to explore. To replace the strong supervision on human poses, which is not available in general, one may need to exploit the prior knowledge such as consistent spatial part arrangements. Second, we believe that our insight on the desired property of the batch constructor for deep metric learning shows promising future research directions. It can be extended to improve various metric learning losses such as contrastive loss and structured losses, beyond the triplet loss. Also, since current class-level approximation becomes less accurate when there are only a small number of examples, one possible future research is to develop a fast and adaptive approximate of pairwise feature distances without using class information.

Bibliography

[1] K. Adamczewski, Y. Suh, and K. M. Lee. Discrete tabu search for graph matching. In *ICCV*, 2015.

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

[3] S. Bai, X. Bai, and Q. Tian. Scalable person re-identification on supervised smoothed manifold. *arXiv:1703.08359*, 2017.

[4] S. Bak, E. Corvée, F. Brémond, and M. Thonnat. Person re-identification using spatial covariance regions of human body parts. In *AVSS*, 2010.

[5] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

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

[7] D. Chen, Z. Yuan, G. Hua, N. Zheng, and J. Wang. Similarity learning on an explicit polynomial kernel feature map for person re-identification. In *CVPR*, 2015.

[8] S. Chen, C. Gong, J. Yang, X. Li, Y. Wei, and J. Li. Adversarial metric learning. In *IJCAI*, 2018.

[9] S.-Z. Chen, C.-C. Guo, and J.-H. Lai. Deep ranking for person re-identification via joint representation learning. *IEEE TIP*, 25(5):2353–2367, 2016.

[10] W. Chen, X. Chen, J. Zhang, and K. Huang. Beyond triplet loss: a deep quadruplet network for person re-identification. In *CVPR*, 2017.

[11] Y. Chen, X. Zhu, and S. Gong. Person re-identification by deep learning multi-scale representations. In *CVPR Workshop*, 2017.

[12] Y.-C. Chen, X. Zhu, W.-S. Zheng, and J.-H. Lai. Person re-identification by camera correlation aware feature augmentation. *IEEE TPAMI*, 2017.

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

[14] D. S. Cheng and M. Cristani. Person re-identification by articulated appearance matching. In *Person Re-Identification*. Springer, 2014.

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

[16] Y. Cui, F. Zhou, Y. Lin, and S. Belongie. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *CVPR*, 2016.

[17] Y. Duan, W. Zheng, X. Lin, J. Lu, and J. Zhou. Deep adversarial metric learning. In *CVPR*, 2018.

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

[19] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. In *CVPR*, 2016.

[20] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell. Compact bilinear pooling. In *CVPR*, 2016.

[21] J. Garcia, N. Martinel, C. Micheloni, and A. Gardel. Person re-identification ranking optimisation by discriminant context information analysis. In *ICCV*, 2015.

[22] W. Ge, W. Huang, D. Dong, and M. R. Scott. Deep metric learning with hierarchical triplet loss. In *ECCV*, 2018.

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

[24] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

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

[26] B. Harwood, V. K. B. G, G. Carneiro, I. Reid, and T. Drummond. Smart mining for deep metric learning. In *ICCV*, 2017.

[27] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv:1703.07737*, 2017.

[28] A. Hermans, L. Beyer, and B. Leibe. In defense of the triplet loss for person re-identification. *arXiv:1703.07737*, 2017.

[29] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.

[30] X.-Y. Jing, X. Zhu, F. Wu, X. You, Q. Liu, D. Yue, R. Hu, and B. Xu. Super-resolution person re-identification with semi-coupled low-rank discriminant dictionary learning. In *CVPR*, 2015.

[31] J.-H. Kim, K.-W. On, J. Kim, J.-W. Ha, and B.-T. Zhang. Hadamard product for low-rank bilinear pooling. In *ICLR*, 2017.

[32] W. Kim, B. Goyal, K. Chawla, J. Lee, and K. Kwon. Attention-based ensemble for deep metric learning. In *ECCV*, 2018.

[33] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Person re-identification by unsupervised ℓ_1 graph learning. In *ECCV*, 2016.

[34] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, 2013.

[35] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. Deepfashion: powering robust clothes recognition and retrieval with rich annotations. In *ICCV Workshops*, 2013.

[36] D. Li, X. Chen, Z. Zhang, and K. Huang. Learning deep context-aware features over body and latent parts for person re-identification. In *CVPR*, 2017.

[37] W. Li, R. Zhao, and X. Wang. Human reidentification with transferred metric learning. In *ACCV*, 2012.

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

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

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

[41] S. Liao and S. Z. Li. Efficient psd constrained asymmetric metric learning for person re-identification. In *ICCV*, 2015.

[42] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[43] T.-Y. Lin, A. RoyChowdhury, and S. Maji. Bilinear cnn models for fine-grained visual recognition. In *ICCV*, 2015.

[44] X. Lin, Y. Duan, Q. Dong, J. Lu, and J. Zhou. Deep variational metric learning. In *ECCV*, 2018.

[45] Y. Lin, L. Zheng, Z. Zheng, Y. Wu, and Y. Yang. Improving person re-identification by attribute and identity learning. *arXiv:1703.07220*, 2017.

[46] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *PAMI*, 5(33):978–994, 2011.

[47] H. Liu, Z. Jie, K. Jayashree, M. Qi, J. Jiang, and S. Yan. Video-based person re-identification with accumulative motion context. *arXiv:1701.00193*, 2017.

[48] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphereface: Deep hypersphere embedding for face recognition. *arXiv:1704.08063*, 2017.

[49] X. Liu, H. Zhao, M. Tian, L. Sheng, J. Shao, S. Yi, J. Yan, and X. Wang. Hydraplus-net: Attentive deep features for pedestrian analysis. *arXiv:1709.09930*, 2017.

[50] Y. Liu, J. Yan, and W. Ouyang. Quality aware network for set to set recognition. In *CVPR*, 2017.

[51] B. Ma, Y. Su, and F. Jurie. Local descriptors encoded by fisher vectors for person re-identification. In *ECCV*, 2012.

[52] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 2008.

[53] N. Martinel, A. Das, C. Micheloni, and A. K. Roy-Chowdhury. Temporal model adaptation for person re-identification. In *ECCV*, 2016.

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

[55] N. McLaughlin, J. Martinez del Rincon, and P. Miller. Recurrent convolutional network for video-based person re-identification. In *CVPR*, 2016.

[56] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh. No fuss distance metric learning using proxies. In *ICCV*, 2017.

[57] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[58] M. Opitz, G. Waltner, H. Possegger, and H. Bischof. Bier - boosting independent embeddings robustly. In *ICCV*, 2017.

[59] S. Paisitkriangkrai, C. Shen, and A. van den Hengel. Learning to rank in person re-identification with metric ensembles. In *CVPR*, 2015.

[60] P. Peng, T. Xiang, Y. Wang, M. Pontil, S. Gong, T. Huang, and Y. Tian. Unsupervised cross-dataset transfer learning for person re-identification. In *CVPR*, 2016.

[61] N. Pham and R. Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *SIGKDD*, 2013.

[62] N. Pham and R. Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *SIGKDD*, 2013.

[63] O. Rippel, M. Paluri, P. Dollar, and L. Bourdev. Metric learning with adaptive density discrimination. *ICLR*, 2016.

[64] 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 on Benchmarking Multi-Target Tracking*, 2016.

[65] E. Ristani and C. Tomasi. Features for multi-target multi-camera tracking and re-identification. In *CVPR*, 2018.

[66] 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*, 2015.

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

[68] A. Schumann and R. Stiefelhagen. Person re-identification by deep learning attribute-complementary information. In *CVPR workshops*, 2017.

[69] Y. Shen, W. Lin, J. Yan, M. Xu, J. Wu, and J. Wang. Person re-identification with correspondence structure learning. In *ICCV*, 2015.

[70] Z. Shi, T. M. Hospedales, and T. Xiang. Transferring a semantic representation for person re-identification and search. In *CVPR*, 2015.

[71] E. Smirnov, A. Melnikov, S. Novoselov, E. Luckyanets, and G. Lavrentyeva. Doppelganger mining for face representation learning. In *ICCVW*, 2017.

[72] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*, 2016.

[73] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy. Deep metric learning via facility location. *CVPR*, 2017.

[74] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.

[75] C. Su, J. Li, S. Zhang, J. Xing, W. Gao, and Q. Tian. Pose-driven deep convolutional model for person re-identification. In *ICCV*, 2017.

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

[77] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian. Deep attributes driven multi-camera person re-identification. In *ECCV*, 2016.

[78] Y. Suh, K. Adamczewski, and K. M. Lee. Subgraph matching using compactness prior for robust feature correspondence. In *CVPR*, 2015.

[79] Y. Suh, M. Cho, and K. M. Lee. Graph matching vis sequential monte carlo. In *ECCV*, 2012.

[80] Y. Suh, J. Wang, S. Tang, T. Mei, and K. M. Lee. Part-aligned bilinear representations for person re-identification. In *ECCV*, 2018.

[81] Y. Sun, L. Zheng, W. Deng, and S. Wang. Svdnet for pedestrian retrieval. In *ICCV*, 2017.

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

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

[84] S. Tang, M. Andriluka, B. Andres, and B. Schiele. Multi people tracking with lifted multicut and person re-identification. In *CVPR*, 2017.

[85] E. Ustinova, Y. Ganin, and V. Lempitsky. Multiregion bilinear convolutional neural networks for person re-identification. In *AVSS*, 2017.

[86] E. Ustinova and V. Lempitsky. Learning deep embeddings with histogram loss. In *NIPS*, 2016.

[87] R. R. Varior, M. Haloi, and G. Wang. Gated siamese convolutional neural network architecture for human re-identification. In *ECCV*, pages 791–808, 2016.

[88] 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.

[89] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[90] C. Wang, X. Zhang, and X. Lan. How to train triplet networks with 100k identities? *arXiv:1709.02940*, 2017.

[91] F. Wang, X. Xiang, J. Cheng, and A. L. Yuille. Normface: l_2 hypersphere embedding for face verification. *arXiv:1704.06369*, 2017.

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

[93] H. Wang, S. Gong, X. Zhu, and T. Xiang. Human-in-the-loop person re-identification. In *ECCV*, 2016.

[94] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014.

[95] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin. Deep metric learning with angular loss. In *ICCV*, 2017.

[96] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[97] C. Weinrich and M. V. H.-M. Gross. Appearance-based 3d upper-body pose estimation and person re-identification on mobile robots. In *ICSMC*. IEEE, 2013.

[98] Y. Wen, K. Zhang, Z. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.

[99] C.-Y. Wu, R. Manmatha, A. J. Smola, and P. Krahenbuhl. Sampling matters in deep embedding learning. In *ICCV*, 2017.

[100] L. Wu, C. Shen, and A. van den Hengel. Personnet: Person re-identification with deep convolutional neural networks. *arXiv:1601.07255*, 2016.

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

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

[103] T. Xiao, S. Li, B. Wang, L. Lin, and X. Wang. Joint detection and identification feature learning for person search. In *CVPR*, 2017.

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

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

[106] H. Xuan, R. Souvenir, and R. Pless. Deep randomized ensembles for metric learning. *arXiv preprint arXiv:1808.04469*, 2018.

[107] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Deep metric learning for person re-identification. In *ICLR*, 2014.

[108] R. Yu, Z. Dou, S. Bai, Z. Zhang, Y. Xu, and X. Bai. Hard-aware point-to-set deep metric for person re-identification. In *ECCV*, 2018.

[109] Y. Yuan, K. Yang, and C. Zhang. Hard-aware deeply cascaded embedding. In *ICCV*, 2017.

[110] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *CVPR*, 2016.

[111] Y. Zhang, B. Li, H. Lu, A. Irie, and X. Ruan. Sample-specific svm learning for person re-identification. In *CVPR*, 2016.

[112] Y. Zhang, X. Li, L. Zhao, and Z. Zhang. Semantics-aware deep correspondence structure learning for robust person re-identification. In *IJCAI*, pages 3545–3551, 2016.

[113] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu. Dual mutual learning. *arXiv:1706.00384*, 2017.

[114] H. Zhao, M. Tian, S. Sun, J. Shao, J. Yan, S. Yi, X. Wang, and X. Tang. Spindle net: Person re-identification with human body region guided feature decomposition and fusion. In *CVPR*, 2017.

[115] L. Zhao, X. Li, Y. Zhuang, and J. Wang. Deeply-learned part-aligned representations for person re-identification. In *ICCV*, 2017.

[116] L. Zhao, X. Li, Y. Zhuang, and J. Wang. Deeply-learned part-aligned representations for person re-identification. In *ICCV*, 2017.

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

[118] Y. Zhao, Z. Jin, G. jun Qi, H. Lu, and X. sheng Hua. A principled approach to hard triplet generation via adversarial nets. In *ECCV*, 2018.

[119] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian. MARS: A video benchmark for large-scale person re-identification. In *ECCV*, 2016.

[120] L. Zheng, Y. Huang, H. Lu, and Y. Yang. Pose invariant embedding for deep person re-identification. *arXiv:1701.07732*, 2017.

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

[122] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian. Query-adaptive late fusion for image search and person re-identification. In *CVPR*, 2015.

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

[124] W.-S. Zheng, X. Li, T. Xiang, S. Liao, J. Lai, and S. Gong. Partial person re-identification. In *ICCV*, 2015.

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

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

[127] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. *ICCV*, 2017.

[128] Z. Zhong, L. Zheng, D. Cao, and S. Li. Re-ranking person re-identification with k-reciprocal encoding. *CVPR*, 2017.

[129] Z. Zhong, L. Zheng, G. Kang, L. Shaozi, and Y. Yi. Random erasing data augmentation. *arXiv:1708.04896*, 2017.

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

초 록

동일인 판별문제는 다른 카메라로 촬영된 각각의 영상에 찍힌 두 사람이 같은 사람인지 여부를 판단하는 문제이다. 이는 감시카메라와 보안에 관련된 다양한 응용분야에서 중요한 도구로 활용되기 때문에 최근까지 많은 연구가 이루어지고 있다. 그러나 같은 사람이더라도 시간, 장소, 촬영 각도, 조명 상태가 다른 환경에서 찍히면 영상마다 보이는 모습이 달라지므로 판별을 자동화하기 어렵다는 문제가 있다.

본 논문에서는 주로 감시카메라 영상에 대해서, 각 영상에서 자동으로 사람을 검출한 후에 검출한 결과들이 서로 같은 사람인지 여부를 판단하는 문제를 풀고자 한다. 이를 위해 1) 어떤 모델이 영상을 잘 표현할것인지 2) 주어진 모델을 어떻게 잘 학습시킬수 있을지 두 가지 질문에 대해서 연구한다. 먼저 벡터 공간 상에서의 거리가 이미지 상에서 대응되는 파트들 사이의 생김새 차이의 합과 같아지도록 하는 매핑 함수를 설계함으로써 검출된 사람들 사이에 신체 부분별로 생김새를 비교를 통해 효과적인 판별을 가능하게 하는 모델을 제안한다. 두번째로 학습 과정에서 클래스 정보를 활용해서 적은 계산량으로 어려운 예시를 많이 보도록 함으로써 효과적으로 함수의 파라미터를 학습하는 방법을 제안한다. 최종적으로는 두 요소를 결합해서 새로운 동일인 판별 시스템을 제안하고자 한다. 본 논문에서는 실험결과를 통해 제안하는 방법이 다양한 환경에서 강인하고 효과적으로 동작함을 증명하였고 보다 일반적인 환경으로의 확장 가능성도 확인 할 수 있을 것이다.

주요어: 동일인 판별, 영상 검색, 영상 임배딩, 거리 학습

학번: 2014-30305