Learning Joint Gait Representation via Quintuplet Loss Minimization

Kaihao Zhang\(^1\) Wenhan Luo\(^2\) Lin Ma\(^2\) Wei Liu\(^2\) Hongdong Li\(^1\)

\(^1\)Australian National University \(^2\)Tencent AI Lab

\{kaihao.zhang, hongdong.li\}@anu.edu.au
\{whluo.china, forest.linma\}@gmail.com \(wl2223@columbia.edu\)

Abstract

Gait recognition is an important biometric technique relevant to video surveillance, where the task is to identify people at a distance by their walking patterns captured in the video. Most of the current approaches for gait recognition either use a pair of gait images to form a cross-gait representation or rely on a single gait image for unique-gait representation. These two types of representations empirically complement one another. In this paper, we propose a new Joint Unique-gait and Cross-gait Network (JUCNet) representation, to combine the advantages of both schemes, leading to significantly improved performance. A second contribution of this work is a tailored quintuplet loss function, which simultaneously boosts inter-class differences by pushing different subjects further apart and contracts intra-class variations by pulling same subjects closer. Extensive tests demonstrate that our method achieves the best performance tested on multiple standard benchmarks, compared with other state-of-the-art methods.

1. Introduction

Gait recognition is the task of identifying people at a distance using videos of their walking patterns [47]. This is an active research topic in the field of computer vision, due to its importance in real-world applications such as video surveillance, forensic identification, and evidence collection [6, 22]. As a behavioral biometric, gait exhibits unique advantages over other biometrics like fingerprint, iris and face [46], because gait based methods can identify subjects from low-resolution video sequences [33] without subject’s cooperation.

In real-world scenarios, variations such as clothing [35], walking speed [30, 43], carrying condition [42], and camera viewpoints [24] result in remarkable changes in gait appearance, which may further degrade the performance of gait recognition. Previous methods [18, 20, 48] have been proposed to alleviate these issues. Most of them focus on cross-gait representation, which is the concatenation of a pair of gait images and labeled to “Same” or “Different” like the input of Fig. 1. While being effective in capturing the relationship between a pair of gaits (gallery and probe [23]), these methods ignore the label (e.g., “X1”, “X2”, “Z1”, and “K2” in Fig. 1) of each single gait image. The potential of unique-gait/single-gait representation is ignored, which makes these methods confused in discriminating different subjects with similar clothing, illumination, and carrying conditions. For example, \(X_1\) and \(Y_1\) in Fig. 2 (a) may be predicted to be an identical subject as they are close in the feature space. Nowadays, some deep learning methods (e.g., [37]) tackle this problem based on unique-gait representation solely. They extract unique-gait features enclosed in a single image and then match them to predict the relationship. While these methods ignore the cross-gait representation.

To this end, we develop a deep network called JUCNet to jointly learn the unique-gait and cross-gait representations. Different from existing gait recognition methods, there are three output branches in our network, of which two branches learn unique-gait representation and one branch
Figure 2. A conventional network, our JUCNet without and with the quintuplet loss are shown in (a), (b), and (c), respectively. From (a) to (b), JUCNet additionally learns the identical unique-gait representation, which enlarges inter-class variations among subjects. From (b) to (c), not only the inter-class variations increases, but also the intra-class discrepancy is decreased, with the help of the proposed quintuplet loss. Red arch lines of each subject domain in (b) indicate the significant intra-class discrepancy, which is reduced as shown by the red circles in (c).

Our main contributions are as follows. 1) We develop a neural network called JUCNet, which jointly learns unique-gait representation and cross-gait representation. The two kinds of representations complement each other and boost the performance of gait recognition. 2) An effective loss function for gait recognition, termed as quintuplet loss, is proposed to guide an extension of JUCNet, named as Multi-Pair JUCNet, to extract powerful features with small intra-class variations and large inter-class differences. 3) Our proposed model outperforms the state-of-the-art models on public challenging gait datasets, showing its superiority.

2. Related Work

Model based methods. These methods aim to model the underlying structure of human body and extract motion features for recognition [2, 5, 18]. They have the advantage of recognizing gait under various situations like different clothing, carrying conditions, etc. It is difficult for these methods to model body structures from relatively low-resolution images, so they can merely work under uncontrolled conditions.

Appearance based methods. Appearance based methods [11, 19, 20, 27, 29, 34, 44, 45] directly extract gait features from videos without modeling the underlying structure of human body. Therefore these methods can work in low-resolution conditions. They usually consist of three steps: 1) obtaining human silhouettes, 2) computing silhouette based representations such as Gait Energy Images (GEIs) [29], chrono-gait images [17], and gait flow [21], and 3) evaluating similarities between gaits.

Deep neural network based methods. Deep learning methods have achieved a great success in the field of computer vision [13, 38, 40, 41, 51, 25, 10]. Recent methods for gait recognition have also adopted CNNs [1, 7, 48, 49].
These methods learn features from pair GEIs in low-level [48, 49], middle-level, or high-level layers [1, 7, 9, 36, 48] and then forward features to a binary classifier for prediction. Wu et al. [48] conducted comprehensive experiments to evaluate these models. However, in these methods, models are trained by merely learning the cross-gait representation, ignoring the identical uniqueness. On the other hand, representative works like [7] train models based on unique-gait representations, without considering useful cross-gait representations. On the contrary, the proposed JUCNet learns both unique-gait and cross-gait representations. Meanwhile, we design a quintuplet loss to guide the model to extract features with smaller intra-class variations and larger inter-class differences.

3. Joint Learning with a Quintuplet Loss

Our method jointly learns unique-gait and cross-gait representations based on a proposed quintuplet loss. Before introducing JUCNet, we represent the method of joint learning and the quintuplet loss in the following.

3.1. Joint Learning

**Cross-gait Learning.** Methods based on cross-gait representation concatenate probe and gallery gait features and input them to a binary classifier to obtain the correct order via their ranking scores. In this work, we denote an instance of pair gaits as \( \{(x_p, x_g), \theta_{pg}\} \), where \( x_p \) is the \( p \)-th probe, \( x_g \) is the \( g \)-th gallery gait, and \( \theta_{pg} \) is the relationship between them. \( \theta_{pg} = 0 \) means that \( x_p \) and \( x_g \) come from an identical subject, and \( \theta_{pg} = 1 \) indicates that they are from different subjects. The cross-gait representation should satisfy the following conditions,

\[
\begin{align*}
    d(x_p, x_g) &\leq b_c - 1 + \delta_{pg}, \quad \text{if } \theta_{pg} = 0, \\
    d(x_p, x_g) &\geq b_c - 1 + \delta_{pg}, \quad \text{if } \theta_{pg} = 1,
\end{align*}
\]

where \( \delta_{pg} \) is a nonnegative slack variable, \( b_c \) is a distance threshold, and \( d(\cdot, \cdot) \) is a predefined or learned metric measuring discrepancy between a pair of gaits. We minimize the cross-entropy loss which is formulated as,

\[
\mathcal{L}_c(x_p, x_g) = -\sum_{p, g} P(x_{pg}) \log Q(x_{pg}),
\]

where \( x_{pg} \) is the cross-gait feature vector, \( P(x_{pg}) \) is the true distribution, and \( Q(x_{pg}) \) is the predicted distribution.

**Unique-gait Learning.** Similar to the learning of cross-gait representation, the unique-gait representation should satisfy the constraints as,

\[
\begin{align*}
    |U(x_p) - U(x_g)|^2_2 &\leq b_u - 1 + \delta_{pg}, \quad \text{if } \theta_{pg} = 0, \\
    |U(x_p) - U(x_g)|^2_2 &\geq b_u - 1 + \delta_{pg}, \quad \text{if } \theta_{pg} = 1,
\end{align*}
\]

where \( U(x_p) \) and \( U(x_g) \) are unique-gait representations and \( b_u \) is a distance threshold between them. In this formulation, the discrepancy between unique-gait representations from identical subjects in terms of Euclidean distance is expected to be smaller than \( b_u \), while that of unique-gait representations from different subjects is expected to be greater than \( b_u \).

In our model, we consider multiple pairs of gaits as input, so the above constraints should be modified as follows,

\[
\begin{align*}
    |U(x_p) - U(x_g')|^2_2 &\leq b_u - 1 - \delta_{pg}, \\
    |U(x_p) - U(x_g)|^2_2 &\geq b_u - 1 + \delta_{pg},
\end{align*}
\]

where \( \{x_p, x_g\} \) come from an identical subject, while \( \{x_p, x_g'\} \) are from different subjects. Our aim is to make the distance between \( x_p \) and \( x_g' \) greater than the distance between \( x_p \) and \( x_g \). The above constraint should be satisfied no matter \( x_p \) and \( x_g \) are identical or not. Thus, the loss function of learning unique-gait representation is composed of two terms,

\[
\mathcal{L}_u(x_p, x_g, x_g', x_g') = \sum_{p, g, g', g''} \{1 + |U(x_p) - U(x_g)|^2 - |U(x_p) - U(x_g')|^2 + |U(x_p) - U(x_g')|^2 + \}
\]

\[
\nu (x_g')^2 + \nu \cdot [1 + |U(x_p) - U(x_g)|^2 - |U(x_p) - U(x_g')|^2]|^2 + \nu |U(x_p) - U(x_g')|^2,
\]
where \( [z]_+ = \max(z, 0) \). The first term corresponds to the case that \( x_p \) and \( x_g \) are identical (\( \hat{p} = p \)), the second term corresponds to the case that \( x_p \) and \( x_g \) are different (\( \hat{p} \neq p \) thus we employ \( p' \) for clarity). We note that, in both cases, \( \{x_p, x_{g'}\} \) and \( \{x_{p'}, x_g\} \) are from different subjects, individually.

**Joint Learning Function.** Finally, JUCNet is updated based on both unique-gait and cross-gait representations, so the overall loss function is the combination of \( \mathcal{L}_c \) and \( \mathcal{L}_u \),

\[
\mathcal{L}_o = \mathcal{L}_c + \eta_u \cdot \mathcal{L}_u, \tag{6}
\]

where \( \eta_u \) is a hyperparameter to balance cross-gait and unique-gait.

### 3.2. Quintuplet Loss

The popular methods for learning the cross-gait representation summarized in Wu et al. [48] are based on recognition signals in Eq. (2), which aims to classify concatenated cross-gait representation. Namely, one class is “identical subject”, and the other class is “different subjects”. In order to obtain more powerful cross-gait representation, we adopt both recognition and verification signals as our supervision and propose a quintuplet loss, targeting at simultaneously enlarging the inter-class differences and reducing the intra-class variations. Different from the traditional recognition-verification loss [32, 39, 8], we define a novel quintuplet loss associated with quintuplet gaits. This loss function considers not only discriminating gait instances, but also differentiating gait pairs.

The Euclidean distance can be employed to measure the similarity between two gaits in the quintuplet loss. While in this work, we replace the Euclidean distance with a learned metric \( C(\cdot, \cdot) \), which represents the distance between two gaits. Specially, the concatenated cross-gait features are forwarded to a fully-connected layer with two neurons. The output value of one neuron is set to be the metric. Considering multiple pairs, constraints in Eq. (1) are reformulated as,

\[
C(x_p, x_{g'}) - C(x_p, x_g) \geq 1 - \delta_{pp'g'}, \tag{7}
\]

where \( \{x_p, x_g\} \) are from an identical subject, while \( \{x_p, x_{g'}\} \) are from different subjects. \( \delta_{pp'g'} \) is a non-negative slack variable. Different from the loss function Eq. (2) utilized in [48], the loss with the learned metric \( C(\cdot, \cdot) \) can be denoted as

\[
\mathcal{L}_c(x_p, x_{g'}, x_p, x_{g'}, x_{p'}) = \sum_{p,g,p',g'} |C(x_p, x_g) - C(x_p, x_{g'}) + \delta_1|_+, \tag{8}
\]

where \( \delta_1 \) is the value of margin. The last fully-connected layer is followed by a softmax layer, which normalizes the learned metric into the range of \([0, 1]\).

Due to the normalization operation, the parameter \( \delta_1 \) is set to 1 in our model. The purpose of the above loss can be concluded as two aspects: 1) Gaits from the same subject \( \{x_p, x_{g'}\} \) are predicted to the class with label 0 and gaits from different subjects \( \{x_p, x_{g''}\} \) are predicted to the other class (label = 1). 2) The distance between \( C(x_p, x_g) \) and \( C(x_p, x_{g''}) \) is enlarged as far as possible. The first aspect can be regarded as a binary classification problem, which is to classify the concatenated cross-gait representation with recognition signals. The second aspect can be treated as a verification problem, which aims to make a distinction between the cross-gait representation from an identical subject and the cross-gait representation from different subjects.

To employ both recognition and verification signals for more powerful cross-gait features with smaller intra-class variations and larger inter-class differences, the loss function of cross-gait is reformulated as

\[
\begin{align*}
\mathcal{L}_c \left( x_p, x_{g'}, x_p, x_{g'}, x_{p'} \right) = & -\sum_{p,g,p',g'} \left[ P(x_{p|p'}) \log Q(x_{p|g'}) + P(x_{p'|g'}) \log Q(x_{p|g'}) \right] \\
& + \eta_c \cdot \sum_{p,g,p',g'} \left[ \delta_2 - D(C(x_p, x_g), C(x_{p'}, x_{g'})) + D(C(x_p, x_{g'}), C(x_{p'}, x_{g'})) \right], \tag{9}
\end{align*}
\]

where \( D(x, y) = \|x - y\|_2^2 \). The pair gaits \( \{x_p, x_{g'}\} \) come from an identical subject, while \( \{x_p, x_{g'}\} \) and \( \{x_{p'}, x_{g'}\} \) are from different subjects. The first term in the right hand is based on the recognition signal, which denotes the classification of gait-cross representation. The second term is based on the verification signal, denoting whether two pairs of gait-cross representations are of the same pair-wise class label (both pairs from identical subjects or both pairs from different subjects, which is the case in Fig. 4) or not (one pair from an identical subject and the other pair from different subjects).

Similar to the extension from Eq. (4) to Eq. (5), the constraint in Eq. (7) should be satisfied no matter \( p = \hat{p} \) and \( p \neq \hat{p} \) as

\[
\begin{align*}
\mathcal{L}_c \left( x_p, x_{g'}, x_p, x_{g'}, x_{p'} \right) = & -\sum_{p,g,p',g'} \left[ \left( P(x_{p|p'}) \log Q(x_{p|g'}) + P(x_{p'|g'}) \log Q(x_{p|g'}) \right) \\
& + \eta_c \cdot \sum_{p,g,p',g'} \left[ \delta_2 - D(C(x_p, x_g), C(x_{p'}, x_{g'})) + D(C(x_p, x_{g'}), C(x_{p'}, x_{g'})) \right], \tag{9}
\end{align*}
\]

where \( \{x_p, x_g\} \) are from an identical subject, while \( \{x_p, x_{g'}\}, \{x_{p'}, x_{g'}\}, \) and \( \{x_{p''}, x_{g'}\} \) come from different subjects, respectively. The hyperparameters \( \eta_c \) and \( \eta_i \) are used to balance different terms. We replace \( \mathcal{L}_c \) in Eq. (6) with the above formulation in the training stage. As it may be noticed, there are quintuplet gait instances
Figure 4. The Multi-Pair JUCNet structure based on the Quintuplet loss in the training stage. The input to our network is several pairs of gaits. Features are extracted from each pair individually, and are processed based on the quintuplet loss. Here, the quintuplet associated with our quintuplet loss can be regarded as X1, X2, Y2, Z1, and K1.

$$(x_p, x_g, x_p', x_g', x_{p''})$$ in Eq. (10), which are the proposed quintuplet loss named after.

4. JUCNet

In this section, we introduce the architecture of the JUCNet, then present a Multi-Pair JUCNet model and the training procedure based on the quintuplet loss.

4.1. Basic JUCNet

As shown in Fig. 3, given a pair of gray-scale gait images, our JUCNet model jointly learns the unique-gait and cross-gait representations, in both low-middle and middle-high levels. The components for learning unique-gait and cross-gait representations are presented in the purple and blue parts, respectively.

**Middle-level features.** The component for capturing middle-level features is shown as the yellow part in Fig. 3, consisting of six convolutional layers. The numbers of kernels in each convolutional layer are sequentially 16, 64, 128, 256, and 256, respectively. The activation function of convolutional layers is Rectified Linear Unit (ReLU). The size of all filters in this stage is $3 \times 3$ with stride 1. Each of the convolutional layers is followed by a max-pooling layer of size $2 \times 2$ and stride 2.

**High-level features.** The part learning high-level features is composed of three branches, of which two learn unique-gait representation and one learns cross-gait representation. Each branch of learning unique-gait representation includes two convolutional layers and two fully-connected layers. Middle-level feature maps with 256 channels are forwarded to the first convolutional layer with 256 kernels of size $3 \times 3$ and stride 1. The second convolutional layer also contains 256 kernels of size $3 \times 3$ and stride 1. Both of them are followed by a max-pooling layer with pooling size $2 \times 2$ and stride 2. After the convolutional layers, two fully-connected layers project feature maps extracted from previous layers into a subspace by 2048 and 512 neurons, respectively.

The component for learning cross-gait representation is also comprised of two convolutional layers and two fully-connected layers. Middle-level features are concatenated as cross-gait feature vectors, which are input into a convolutional layer with 256 kernels of size $3 \times 3$ and stride 1. The difference from the first layer of learning the unique-gait representation is that the number of kernels is doubled due to concatenation. The second convolutional layer and the first fully-connected layer are the same as those learning unique-gait representation. The second fully-connected layer contains 2048 neurons.

4.2. Multi-Pair JUCNet

As described above, JUCNet learns both unique-gait and cross-gait representations. The proposed quintuplet loss can enlarge inter-class differences and reduce intra-class variations simultaneously. To this end, we extend the basic JUCNet as a Multi-Pair JUCNet, which serves as the final framework during training, and train it with the quintuplet loss.

Fig. 4 shows the overview of the Multi-Pair JUCNet. A pair of gaits can be combined as a whole, with the label of Same-ID or Different-ID. The basic JUCNet model extracts both unique-gait and cross-gait representations. For Multi-Pair JUCNet, three pairs of gaits are input to extract features. Two pairs of gaits are from different subjects, while one pair of gaits is from an identical subject. Our model learns unique-gait representation based on the loss in Eq. (5), and learns cross-gait representation based on the
quintuplet loss in Eq. (10).

4.3. Training

We choose the popular GEIs [29] as the input of Multi-Pair JUCNet because of its robustness to noise and its simplicity for computation [16]. GEIs images are resized to the size of $256 \times 372 \times 1$. In order to augment training samples, we crop a set of $224 \times 326 \times 1$ patches from GEIs images and flip them horizontally at random. It is worth noting that a pair of GEIs are flipped at the same time to ensure the same walking direction. It is trained based on stochastic gradient descent. Weights are initialized as a Gaussian distribution with mean 0 and standard deviation 0.01. The momentum is set as 0.9. The model is updated every time after learning one mini-batch of size 32.

5. Experiments

To verify our model, we test it on three public datasets, which are at first introduced in this section. These datasets cover challenges like clothing variation, cross view, etc. in the task of the gait recognition. Based on the datasets, we then investigate effectiveness of JUCNet and the quintuplet loss. Meanwhile, comparison with the state-of-the-art methods is also reported. Finally, we study the performance of our method with the protocol of cross view.

5.1. Datasets

The OUTD-B dataset. The OU-ISIR Gait Database, Treadmill Dataset B (OUTD-B) [26], is challenging due to its considerable clothing diversities, such as wearing hat, regular pants, and half shirt. It is composed of 68 subjects with up to 32 clothing conditions. There are three subsets in this dataset, a training set, a gallery set, and a probe set. The training set includes 20 subjects with 446 sequences. The gallery set and probe set are employed in the testing stage. There are 48 subjects with standard clothing types in the gallery set. The probe set contains 856 sequences of subjects with other clothing types. Note that subjects in the gallery set and probe set are disjoint from those in the training set.

The OU-LP-Bag $\beta$ dataset. The OU-LP-Bag $\beta$ database [28] is built to alleviate the problem of too small variations in existing datasets. There are one training set, one gallery set, and one probe set in this dataset. The training set includes 1,034 subjects. For each subject, there are two sequences, one carrying objects while the other one not. The gallery and probe sets contain 1,036 subjects which are disjoint from the subjects in the training set. Subjects in the gallery set carry objects while subjects in the probe set carry nothing. This dataset provides GEIs of all sequences, so we directly use these GEIs to carry out our experiments.

The CASIA-B gait dataset. The CASIA-B gait database [50] is composed of 124 subjects, with 110 sequences per subject. It contains eleven views and there are ten sequences per view. Among the ten sequences, six are taken under normal walking conditions (NM), two are taken when subjects are with coats (CL), and two are taken when subjects are with bags (BG).

5.2. Effectiveness of JUCNet and Quintuplet Loss

To demonstrate the effectiveness of the JUCNet and quintuplet loss, we develop three third-party baseline networks, $MT$, Deeper $MT$, and $CNet$. MT and Deeper MT are representative methods from [48] for learning sole cross-gait representation to predict the relationship between a pair of gaits. CNet is a simplified version of JUCNet without the component of learning unique-gait representation. We also conduct ablation analysis by comparing our full method JUCNet (Metric & Quintuplet) with two versions of self baseline networks, $JUCNet$ and $JUCNet$ (Metric). All these networks are illustrated in the following.

- MT is a CNN consisting of two convolutional layers, two pooling layers, and one fully-connected layer. The input of this model is a pair of GEIs. The MT extracts features by the convolutional layers and concatenates features as the cross-gait representation by the fully-connected layer. Finally, the cross-gait representation will be input to a binary classifier to predict their relationship.

- Deeper MT is a deeper version of MT. It contains two additional fully-connected layers. Two convolutional layers and two fully-connected layers are utilized to learn two feature sets from the input GEIs. Then they will be concatenated as a whole to learn cross-gait representation by the third fully-connected layer. MT and Deeper MT have achieved state-of-the-art performance on some datasets [48].

- CNet is a network which excludes the unique-gait part from our JUCNet. It contains eight convolutional layers and two fully-connected layers. As shown in the yellow and blue parts of Fig. 3, CNet shares a similar structure with both MT and Deeper MT. The major difference from them is that when the feature maps are concatenated as a whole, more layers are built in order to learn powerful cross-gait representation.

- JUCNet is our proposed network that jointly learns unique-gait and cross-gait representation. This JUCNet model is updated based on the loss functions in Eq. (2), (4), and (6).

- JUCNet (Metric) is our JUCNet model plus metric learning. It learns the metric $C(\cdot, \cdot)$ to represent the distance between a pair of gaits. The loss functions employed to train this model are Eqs. (4), (6), and (8).
JUCNet (M & Quintuplet) is our JUCNet model plus both metric learning and our proposed quintuplet loss. Fig. 4 and the section of Quintuplet Loss present the details of this model and the quintuplet loss. The model is trained based on the loss functions in Eqs. (4), (6) and (10).

Parameter analysis. Different loss terms are weighted by hyperparameters in our loss functions. In order to set them appropriately, we utilize a part of the training set as a validation set and investigate the effect of hyperparameters by varying the weighting parameters \( \eta_i, \eta_u, \) and \( \eta_c \) in Eqs. (5), (6), and (10) from 0 to 1. When hyperparameters are equal to 0, only the first term in the above equations works. With the increase of hyperparameters, the binding term plays a more and more important role in our model. When varying one hyperparameter, the other two hyperparameters are set to be fixed. According to the results shown in Fig. 5, in general the accuracies become higher with the increase of hyperparameters until becoming lower with increased values. The best performance is achieved when \( \eta_i = 0.6, \eta_u = 0.4, \) and \( \eta_c = 0.6 \), which are set in our following experiments.

Results in terms of rank-\( n \) accuracy. We report the results of rank-1, rank-3, rank-5, and rank-10 accuracies of the aforementioned six models on both OU-LP-Bag \( \beta \) and OUTD-B, shown in Table 1 and Table 2, respectively.

In both tables, we observe that: 1) JUCNet achieves higher accuracies than MT, Deeper MT and CNet. This verifies the effectiveness of the proposed JUCNet by jointly learning unique-gait and cross-gait representations. As shown in Fig. 3, the unique-gait representation and cross-gait representation complement each other to update the shared-weight layers, leading to more powerful high-level features. 2) The improvement from JUCNet to JUCNet (Metric) reveals the advantage of metric \( C(\cdot, \cdot) \), which learns to measure discrepancy between gaits driven by data automatically, in contrast to pre-defined metric like the Euclidean distance. 3) The JUCNet (Metric & Quintuplet) achieves better performance than both JUCNet and JUCNet (Metric), which additionally suggests the effectiveness of our proposed quintuplet loss. 4) The improvement from other models to JUCNet (Metric & Quintuplet) in terms of rank-1 accuracy is more evident than that in terms of rank-3, rank-5, and rank-10 accuracies. We that suspect the following reason justifies. Given a probe gait, other models may determine more than one gallery gait as from an identical subject, because they are trained with only classification loss. To the contrast, the quintuplet loss guides our model to not only obtain correct classification results, but also learn more powerful features ensuring enlarged inter-class differences and decreased intra-class variations, leading to correct ranking orders.

<table>
<thead>
<tr>
<th>Models</th>
<th>rank-1</th>
<th>rank-3</th>
<th>rank-5</th>
<th>rank-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT [48]</td>
<td>59.9</td>
<td>75.2</td>
<td>80.1</td>
<td>86.8</td>
</tr>
<tr>
<td>Deeper MT [48]</td>
<td>68.1</td>
<td>81.8</td>
<td>86.0</td>
<td>90.8</td>
</tr>
<tr>
<td>CNet</td>
<td>71.0</td>
<td>86.9</td>
<td>91.5</td>
<td>95.2</td>
</tr>
<tr>
<td>JUCNet</td>
<td>74.3</td>
<td>87.4</td>
<td>90.8</td>
<td>95.3</td>
</tr>
<tr>
<td>JUCNet (Metric)</td>
<td>74.8</td>
<td>88.9</td>
<td>92.3</td>
<td>95.6</td>
</tr>
<tr>
<td>JUCNet (M &amp; Quintuplet)</td>
<td><strong>78.2</strong></td>
<td><strong>89.6</strong></td>
<td><strong>92.8</strong></td>
<td><strong>95.8</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Models</th>
<th>rank-1</th>
<th>rank-3</th>
<th>rank-5</th>
<th>rank-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT [48]</td>
<td>70.7</td>
<td>87.7</td>
<td>91.9</td>
<td>97.9</td>
</tr>
<tr>
<td>Deeper MT [48]</td>
<td>72.4</td>
<td>90.3</td>
<td>95.8</td>
<td>98.4</td>
</tr>
<tr>
<td>CNet</td>
<td>71.1</td>
<td>88.2</td>
<td>94.3</td>
<td>97.9</td>
</tr>
<tr>
<td>JUCNet</td>
<td>73.2</td>
<td>88.9</td>
<td>94.2</td>
<td>98.0</td>
</tr>
<tr>
<td>JUCNet (Metric)</td>
<td>73.8</td>
<td>88.4</td>
<td>93.9</td>
<td>97.9</td>
</tr>
<tr>
<td>JUCNet (M &amp; Quintuplet)</td>
<td><strong>76.4</strong></td>
<td><strong>91.4</strong></td>
<td><strong>95.2</strong></td>
<td><strong>98.7</strong></td>
</tr>
</tbody>
</table>

Table 1. The rank-1, rank-3, rank-5, and rank-10 accuracies [%] of different models on the OU-LP-Bag \( \beta \) dataset. The best results are shown in bold, which also applies to the following tables.

5.3. Comparison with State-of-the-art Methods

We have verified that the proposed JUCNet with the quintuplet loss outperforms the conventional CNN mod-
els which are solely based on cross-gait representation. In this section, we compare our method with other state-of-the-art methods, including part-based FDF [14], part-based Entropy of the Discrete Fourier Transform (EnDFT) [35], GEnI [3], Masked-GEI [4], Gabor GEI [42] and spatial metric learning methods using GEI like ranking SVM [31], and a Joint Intensity and Spatial Metric Learning method (JISML) [28].

The results in Table 3 show that JUCNet plus metric learning outperforms the previous best method on both OU-LP-Bag β and OUTD-B databases, which reveals its effectiveness. JUCNet based on metric learning and quintuplet loss achieves better performance than JUCNet with metric, justifying the advantage of our proposed quintuplet loss again. The JISML method introduces joint learning of intensity and spatial metric in order to mitigate the large intra-class differences and leverage the subtle inter-class differences, while in our method the quintuplet loss accomplishes this task. A method proposed by Guan et al. [12] achieves better rank-1 accuracy on the OUTD-B dataset than ours. While their results are achieved under a different training/testing protocol. Meanwhile, their method requires a regular within-class matrix for the gallery set, so it cannot be applied on datasets including only a single probe and a single gallery per subject like the OU-LP-Bag β dataset.

In addition, Table 3 reveals that the improvement over existing methods achieved by our method on the OU-LP-Bag β dataset is greater than that on the OUTD-B dataset with regard to the rank-1 accuracy. This is because that there are more subjects (1,034) in the training set of the OU-LP-Bag β dataset, while there are only 20 subjects in the training set of OUTD-B. Larger scale of the training set benefits our model in gaining greater learning capacity. On the other hand, though there are more samples in the OU-LP-Bag β dataset, the final results regarding the rank-1 accuracy on both datasets are at the same level. We believe that, it is more difficult for models to recognize the correct subjects from the OU-LP-Bag β dataset than the OUTD-B dataset because the OUTD-B dataset includes only 48 subjects in the testing set, while there are 1,036 subjects in the testing set of the OU-LP-Bag β dataset.

It may be observed that the results of OU-LP-Bag β and OUTD-B in Table 3 are better than those in Tables 1 and 2. As mentioned above, we utilize a part of the training set in these two datasets as a validation set to tune weight parameters. Thus results in Tables 1 and 2 are reported by models trained without the validation set. While comparing with other methods in Table 3, we put the validation set back to the training set to re-train the model for a fair comparison, because the validation set belongs to the training set in other methods.

5.4. Cross-view Study

The issue of cross view is crucial for gait recognition, so we evaluate our method under the condition of cross view on the BG subset of the CASIA-B gait dataset. We evaluate our method on the more challenging BG set (the accuracy is between 86.0% and 93.3%), rather than the NM set (the accuracy is between 97.0% and 99.5%). As shown in Table 4, subjects in the probe and gallery sets are of different views. The comparison with Wu et al. [48] and Hu et al. [15] indicates that our method achieves satisfactory performance under the cross-view protocol.

6. Conclusion

We have proposed a JUCNet model to jointly learn unique-gait and cross-gait representations for gait recognition. The two kinds of representations complement each other to boost the performance of gait recognition. Moreover, a quintuplet loss for gait recognition was proposed to increase the inter-class differences by pushing the cross-gait representation learned from different classes apart and reduce the intra-class variations by pulling the representations learned from an identical class together. The experimental results on public datasets suggest that the JUCNet model outperforms existing CNN models based on sole cross-gait representation, demonstrating the effectiveness of the JUCNet model. JUCNet with the quintuplet loss further improves the performance, validating its superiority over the state-of-the-art methods.
References


[37] Kohei Shiraga, Yasushi Makihara, Daigo Muramatsu, Tomio Echigo, and Yasushi Yagi. Geinet: View-invariant gait recog-


[40] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. In CVPR, 2017. 2


[48] Zifeng Wu, Yongzhen Huang, Liang Wang, Xiaogang Wang, and Tieniu Tan. A comprehensive study on cross-view gait based human identification with deep cnns. TPAMI, 2017. 1, 2, 3, 4, 6, 7, 8

