Appendix of the Paper
(Unsupervised Cross-dataset Person Re-identification by Transfer Learning of Spatial-Temporal Patterns)

Jianming Lv\textsuperscript{1}, Weihang Chen\textsuperscript{1}, Qing Li\textsuperscript{2}, and Can Yang\textsuperscript{1}
\textsuperscript{1}South China University of Technology
\textsuperscript{2}City University of Hongkong
\{jmlv,cscyang\}@scut.edu.cn, csscut@mail.scut.edu.cn, qing.li@cityu.edu.hk

1. Architecture of the Visual Classifier $C$

(Extension of Section 4.2)

We select the recently proposed convolutional siamese network \cite{2} as $C$, which makes better use of the label information and has good performance in the large-scale datasets such as Market1501\cite{3}. As shown in Fig. 1, the network adopts a siamese scheme including two ImageNet pre-trained CNN modules, which share the same weight parameters and extract visual features from the input images $S_i$ and $S_j$. The CNN module is achieved from the ResNet-50 network \cite{1} by removing its final fully-connected (FC) layer. The outputs of the two CNN modules are flattened into two one-dimensional feature vectors $\vec{v}_i$ and $\vec{v}_j$, which act as the embedding visual feature vectors of the input images.

To measure the matching degree of the input images, their feature vectors $\vec{v}_i$ and $\vec{v}_j$ are fed into the following square layer to conduct subtracting and squaring element-wisely: $\vec{v}_s = (\vec{v}_i - \vec{v}_j)^2$. Finally, a convolutional layer is used to transform $\vec{v}_s$ into the similarity score as:

$$\hat{q} = \text{sigmoid}(\theta_s \circ \vec{v}_s)$$

(1)

Here $\theta_s$ denotes the parameters in the convolutional layer, $\circ$ denotes the convolutional operation, and $\text{sigmoid}$ indicates the sigmoid activation function. By comparing the predicted similarity score with the ground-truth matching result of $S_i$ and $S_j$, we can achieve the variation loss as a cross entropy form:

$$\text{LOSS}_v = -q \cdot \log(\hat{q}) - (1 - q) \cdot \log(1 - \hat{q})$$

(2)

Here $q = 1$ when $S_i$ and $S_j$ contain the same person, otherwise, $q = 0$.

Besides predicting the similarity score, the model also predicts the identity of each image in the following steps. Each visual feature vector ($\vec{v}_s(x = i,j)$) is fed into one convolutional layer to be mapped into an one-dimensional vector with the size $K$, where $K$ is equal to the total number of the pedestrians in the dataset. Then the following softmax unit is applied to normalize the output as follows:

$$\hat{P}(x) = \text{softmax}(\theta_x \circ \vec{v}_s)(x = i,j)$$

(3)

Here $\theta_x$ is the parameter in the convolutional layer and $\circ$ denotes the convolutional operation. The output $\hat{P}(x)$ is used to predict the identity of the person contained in the input image $S_x(x = i,j)$. By comparing $\hat{P}(x)$ with the ground-truth identify label, we can achieve the identification loss as the cross-entropy form:

$$\text{LOSS}_{id} = \sum_{k=1}^{K} (-\log \hat{P}_k \cdot P_k) + \sum_{k=1}^{K} (-\log \hat{P}_k \cdot P_k)$$

(4)

Here $P(x = i,j)$ is the identity vector of the input image $S_x$. $P_k = 0$ for all $k$ except $P_t(x) = 1$, where $t$ is ID of the person in the image $S_x$.

The final loss function of the model is defined as:

$$\text{LOSS}_{all} = \text{LOSS}_v + \text{LOSS}_{id}$$

(5)

According to \cite{2}, this kind of composite loss makes the classifier more efficient to extract the view invariant visual features for Re-ID than the single loss function.

While deploying this classifier to perform Re-ID, given two images $S_i$ and $S_j$ as input, the CNN modules extract their visual feature vectors $\vec{v}_i$ and $\vec{v}_j$ as shown in Fig. 1. The matching probability of $S_i$ and $S_j$ is measured as the cosine similarity of the two feature vectors:

$$\text{Pr}(S_i \mid C \, S_j | \vec{v}_i, \vec{v}_j) = \frac{\vec{v}_i \cdot \vec{v}_j}{\| \vec{v}_i \| \| \vec{v}_j \|}$$

(6)
2. Proof of Eq. (5)

\[
\begin{align*}
Pr(\Delta_{ij}, c_i, c_j | S_i \parallel S_j) &= Pr(\Delta_{ij}, c_i, c_j | \mathcal{Y}(S_i) = \mathcal{Y}(S_j)) \ast \\
&+ Pr(\mathcal{Y}(S_i) = \mathcal{Y}(S_j) | S_i \parallel S_j) \ast \\
&+ Pr(\Delta_{ij}, c_i, c_j | \mathcal{Y}(S_i) \neq \mathcal{Y}(S_j)) \ast \\
&+ Pr(\mathcal{Y}(S_i) \neq \mathcal{Y}(S_j) | S_i \parallel S_j)
\end{align*}
\]

Similarly, we have:

\[
\begin{align*}
Pr(\Delta_{ij}, c_i, c_j | S_i \nparallel S_j) &= Pr(\Delta_{ij}, c_i, c_j | \mathcal{Y}(S_i) = \mathcal{Y}(S_j)) \ast \\
&+ Pr(\mathcal{Y}(S_i) = \mathcal{Y}(S_j) | S_i \nparallel S_j) \ast \\
&+ Pr(\Delta_{ij}, c_i, c_j | \mathcal{Y}(S_i) \neq \mathcal{Y}(S_j)) \ast \\
&+ Pr(\mathcal{Y}(S_i) \neq \mathcal{Y}(S_j) | S_i \nparallel S_j)
\end{align*}
\]

From (7) and (8), we have:

\[
\begin{align*}
Pr(\Delta_{ij}, c_i, c_j | \mathcal{Y}(S_i) = \mathcal{Y}(S_j)) &= (1 - E_n - E_p)^{-1}((1 - E_n) \ast Pr(\Delta_{ij}, c_i, c_j | S_i \parallel c S_j) \\
&- E_p \ast Pr(\Delta_{ij}, c_i, c_j | S_i \nparallel c S_j))
\end{align*}
\]

3. Proof of Theorem 1

Proof of Theorem 1: By analyzing the relationship between \( Pr(\mathcal{Y}(S_i) = \mathcal{Y}(S_j) | v_i, v_j, \Delta_{ij}, c_i, c_j) \) and \( Pr(S_i \parallel S_j | v_i, v_j, \Delta_{ij}, c_i, c_j) \), we have:

\[
Pr(\mathcal{Y}(S_i) = \mathcal{Y}(S_j) | v_i, v_j, \Delta_{ij}, c_i, c_j) = Pr(\mathcal{Y}(S_i) = \mathcal{Y}(S_j) | S_i \parallel S_j) \ast Pr(S_i \parallel S_j | v_i, v_j, \Delta_{ij}, c_i, c_j) + \\
Pr(\mathcal{Y}(S_i) = \mathcal{Y}(S_j) | S_i \nparallel S_j) \ast Pr(S_i \nparallel S_j | v_i, v_j, \Delta_{ij}, c_i, c_j)
\]

\[
=(1 - E'_n) \ast Pr(S_i \parallel S_j | v_i, v_j, \Delta_{ij}, c_i, c_j) + \\
E'_n \ast (1 - Pr(S_i \parallel S_j | v_i, v_j, \Delta_{ij}, c_i, c_j))
\]

According to the Eq.(11) of the original paper, we have:

\[
Pr(S_i \parallel S_j | v_i, v_j, \Delta_{ij}, c_i, c_j) = \frac{(M_1 + \alpha(1 - \alpha - \beta)^{-1})}{Pr(\Delta_{ij}, c_i, c_j)} (0 \leq \alpha, \beta \leq 1)
\]

By substituting Eq.(11) into Eq.(10), we have:

\[
Pr(\mathcal{Y}(S_i) = \mathcal{Y}(S_j) | v_i, v_j, \Delta_{ij}, c_i, c_j) = (1 - E'_n) \ast \frac{(M_1 + \alpha(1 - \alpha - \beta)^{-1})}{Pr(\Delta_{ij}, c_i, c_j)} (0 \leq \alpha, \beta \leq 1)
\]

From (13) and (12) we have:

\[
(1 - E'_n) \ast ((1 - \alpha) M_2 - \beta M_3) \ast E'_n
\]

Thus, we have:

\[
\sum_{\Delta_{ij}, c_i, c_j} [(M_1 + E_n(1 - E_p - E_n)^{-1}) \ast ((1 - E_n) \ast M_2 - E_p \ast M_3)]
\]

\[
= \sum_{\Delta_{ij}, c_i, c_j} (((1 - E_p - E_n) \ast ((1 - \alpha) M_2 - \beta M_3) + E'_n \ast Pr(\Delta_{ij}, c_i, c_j))
\]

From (15) have:

\[
(M_1 + E_n(1 - E_p - E_n)^{-1})(1 - E_n - E_p) = (1 - E'_n) \ast ((1 - \alpha) M_2 + \alpha p) + E'_n
\]

After taking the derivative with respect to \( M_1 \) in the both sides of Eq. (16), we can get:
\[ 1 - E_p - E_n = (1 - \alpha - \beta)(1 - E'_p - E'_n) \]  

(17)

Thus, when \( E_p + E_n < 1 \) and \( \alpha + \beta < 1 \), we can infer from Eq.(17) that:

\[ E'_p + E'_n < E_p + E_n. \]  

(18)

4. Learned Spatio-temporal Patterns  
(Extension of Fig.4)

Fig. 2 shows the spatio-temporal distribution \( \Pr(\Delta_{ij}, c_i, c_j | Y(S_i) \supseteq C \searrow Y(S_j)) \) learned (a) in the ‘GRID’ dataset, and (b) in the ‘Market1501’ dataset.

References


