Progressive Learning Algorithm for Efficient Person Re-Identification

Zhen Li¹,², Liang Niu³,⁴, Hanyang Shao¹, Nian Xue³,⁴

Shanghai Grandhonor Information Technology¹
Nanjing University of Aeronautics and Astronautics²
New York University³
New York University Abu Dhabi⁴

ITALY 10 - 15 January 2021
Motivations

- We develop a novel learning strategy to find efficient feature embeddings while maintaining the balance of accuracy and model complexity.
- Existing triplet loss methods select only the hard identity examples which may not be optimal without considering easy examples in the triplet anchor.

![Diagram of training mini-batches consisting of easy, medium-hard, and over-hard examples.](image)

Fig 1. Training mini-batches consisting of easy, medium-hard, and over-hard examples.
Fig. 2: Overview of the proposed ReID network architecture. “Anc.”, “Pos.” and “Neg.” represent anchor image, positive images that belong to the same identity and negative images that belong to different identities, respectively.
Problem Formulation

Main idea: Training the ReID model based on Progressive Learning Algorithm.

- **Input:** A fixed-size mini-batch consisting of $P = 16$ randomly selected identities and $K = 8$ randomly selected images per identity from the training set.
- **Output:** The optimal hyperparameter $w^*$ along with the well trained CNN.
- **Initialization:** Randomly initialize $N$ sets of hyper-parameters $\mathcal{W} = \{w_1, w_2, \cdots, w_N\}$, where $w_i = (\lambda_i, m_i, k_i, p_i)$, $\lambda_i \in [0, 2]$, $m_i \in [-0.1, 0.3]$, $k_i \in [1, 8]$, $p_i \in [1, 16]$ for $i = 1, \cdots, N$.
- **Loss function:**

\[
\mathcal{L}_{GBH}^{k,p}(\theta; X) = \sum_{l=1}^{P} \sum_{a,b} \sum_{\gamma_a = \gamma_b = l} \ln \left( 1 + e^{m(T_k,p(a,b,n))} \right) \tag{1}
\]

\[
\mathcal{L}^{k,p}(\theta; X) = \mathcal{L}_{softmax}(\theta; X) + \lambda \mathcal{L}_{GBH}^{k,p}(\theta; X) \tag{2}
\]
Repeat:
  for each hyperparameter $i = 1$ to $N$ do
    Exploration: Backpropagate CNN in 20 epochs and evaluate the loss $\mathcal{L}$ according to Eq. 1 and Eq. 2, and evaluate the Bayesian objective $f(w_i)$.
    Restoration: CNN weights are restored to that before 20 epochs of exploration.
  end for
  Exploitation: Based on $f(\mathbb{W})$, obtain a new improved candidate $w'$ and update Gaussian process according to Eq. 3 and Eq. 4, and add $w'$ to $\mathbb{W}$, and update $\hat{w}$ as well;
    Backpropagate to update CNN weights for 300 epochs based on the new hyperparameter $w$ and the feed-forward loss $\mathcal{L}$;
    Save the model with lowest loss $\mathcal{L}$ for the current hyperparameter $w$;
Until maximum epochs ($M = 3,000$) reached
Bayesian Algorithm

For $N$ sets of such parameters we denote $\mathbb{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N\}$, and the corresponding $f(\mathbb{W})$, the posterior belief of $f$ at a new candidate $\mathbf{w}$ is given by

\[
\tilde{f}(\mathbf{w}) \sim \mathcal{GP}(\mu(\hat{\mathbf{w}}) + \Delta \mu, \mathcal{K}(\mathbf{w}) - \Delta \mathcal{K})
\]
\[
\Delta \mu = \mathcal{K}(\mathbf{w}, \mathbb{W}) \mathcal{K}(\mathbb{W})^{-1} (f(\mathbb{W}) - \mu(\mathbb{W}))
\]
\[
\Delta \mathcal{K} = \mathcal{K}(\mathbf{w}, \mathbb{W}) \mathcal{K}(\mathbb{W})^{-1} \mathcal{K}(\mathbb{W}, \mathbf{w})
\]

The expected improvement of a candidate $\hat{\mathbf{w}}$ is defined as

\[
\mathcal{EI}(\mathbf{w}) = (\mathcal{K}(\mathbf{w}) - \Delta \mathcal{K})^{\frac{1}{2}} (Z\Phi(Z) + \phi(Z))
\]

Bayesian optimization minimizes the following objective function:

\[
f(\mathbf{w}) = \left| \frac{\mathcal{L}^{k,p}_t (\theta; \mathbf{w}; X) - \mathcal{L}^{k,p}_{t'} (\theta; \mathbf{w}; X)}{\mathcal{L}^{k,p}_t (\theta; \mathbf{w}; X)} - \mathcal{E}D \right|
\]
Datasets

We perform all the experiments on three commonly used benchmarks: Market-1501, DukeMTMC-ReID (briefed as DukeMTMC), and CUHK03(D) & CUHK03(L) datasets. These ReID datasets are summarized in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Market1501</th>
<th>DukeMTMC</th>
<th>CUHK03(D/L)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identities</td>
<td>1,501</td>
<td>1,812</td>
<td>1,360</td>
</tr>
<tr>
<td>Bboxes</td>
<td>32,668</td>
<td>36,411</td>
<td>13,164</td>
</tr>
<tr>
<td>Camera</td>
<td>6</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>Train images</td>
<td>12,936</td>
<td>16,522</td>
<td>7,365/7,368</td>
</tr>
<tr>
<td>Train ids</td>
<td>751</td>
<td>702</td>
<td>767</td>
</tr>
<tr>
<td>Query images</td>
<td>3,368</td>
<td>2,228</td>
<td>1,400</td>
</tr>
<tr>
<td>Query ids</td>
<td>750</td>
<td>702</td>
<td>700</td>
</tr>
<tr>
<td>Gallery images</td>
<td>19,732</td>
<td>17,661</td>
<td>5,332</td>
</tr>
</tbody>
</table>

Table 1: ReID Benchmark datasets used in our experiments.
<table>
<thead>
<tr>
<th>Category</th>
<th>Methods</th>
<th>Market1501(SQ)</th>
<th>Market1501(MQ)</th>
<th>CUHK03(D)</th>
<th>CUHK03(L)</th>
<th>DukeMTMC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP</td>
<td>Rank-1</td>
<td>mAP</td>
<td>Rank-1</td>
<td>mAP</td>
<td>Rank-1</td>
</tr>
<tr>
<td>part</td>
<td>HA-CNN[7]</td>
<td>75.7</td>
<td>91.2</td>
<td>82.8</td>
<td>93.8</td>
<td>38.6</td>
</tr>
<tr>
<td></td>
<td>Deep-Person[36]</td>
<td>79.6</td>
<td>92.3</td>
<td>85.1</td>
<td>94.5</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>PCB[1]</td>
<td>77.4</td>
<td>92.3</td>
<td>-</td>
<td>-</td>
<td>54.2</td>
</tr>
<tr>
<td></td>
<td>PCB+RPP[1]</td>
<td>81.6</td>
<td>93.8</td>
<td>-</td>
<td>-</td>
<td>57.5</td>
</tr>
<tr>
<td></td>
<td>Aligned-ReID[35]</td>
<td>82.3</td>
<td>92.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>MGN[2]</td>
<td>86.9</td>
<td>95.7</td>
<td>90.7</td>
<td>96.9</td>
<td>66.0</td>
</tr>
<tr>
<td>global</td>
<td>SVDNet[5]</td>
<td>62.1</td>
<td>82.3</td>
<td>-</td>
<td>-</td>
<td>37.2</td>
</tr>
<tr>
<td></td>
<td>TriNet[6]</td>
<td>69.1</td>
<td>84.9</td>
<td>76.4</td>
<td>90.5</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>GP-reid[24]</td>
<td>81.2</td>
<td>92.2</td>
<td>82.8</td>
<td>93.8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DaRe[9]</td>
<td>74.2</td>
<td>88.5</td>
<td>-</td>
<td>-</td>
<td>58.1</td>
</tr>
<tr>
<td></td>
<td>PLA</td>
<td>83.6</td>
<td>93.7</td>
<td>88.4</td>
<td>95.2</td>
<td>63.2</td>
</tr>
<tr>
<td>RK</td>
<td>Trinet [6]</td>
<td>81.1</td>
<td>86.7</td>
<td>87.2</td>
<td>91.8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>DaRe [9]</td>
<td>85.9</td>
<td>90.8</td>
<td>-</td>
<td>-</td>
<td>71.2</td>
</tr>
<tr>
<td></td>
<td>MGN [2]</td>
<td>94.2</td>
<td>96.6</td>
<td>95.9</td>
<td>97.1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>PLA</td>
<td>89.4</td>
<td>94.7</td>
<td>92.9</td>
<td>95.7</td>
<td>77.2</td>
</tr>
</tbody>
</table>

Table 2: Comparing PLA with different global models on all datasets. “RK” stands for reranking.
Computing and Memory Cost

Fig. 3 a) Accuracy vs. computation cost (number of Mul-Add); b) Accuracy vs. inference memory (MB). Accuracy is reported as the average mAP on all datasets.
Conclusions

- A novel learning approach is proposed to find efficient feature embeddings while maintaining the balance of accuracy and model complexity.
- A novel method is developed to explore the hard examples and build a discriminant feature embedding yet compact enough for large-scale applications.
- A novel Bayesian approach is employed to progressively learn the triplet loss from simple to hard samples.
- The developed reid system is efficient in both computation and memory, rendering it a commercial system.
References I


References II


