基本任务：

框架图

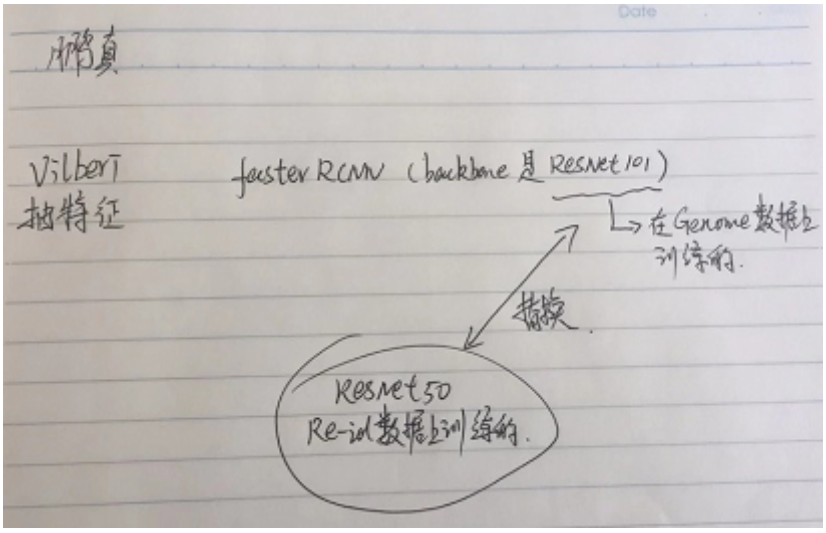
ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks

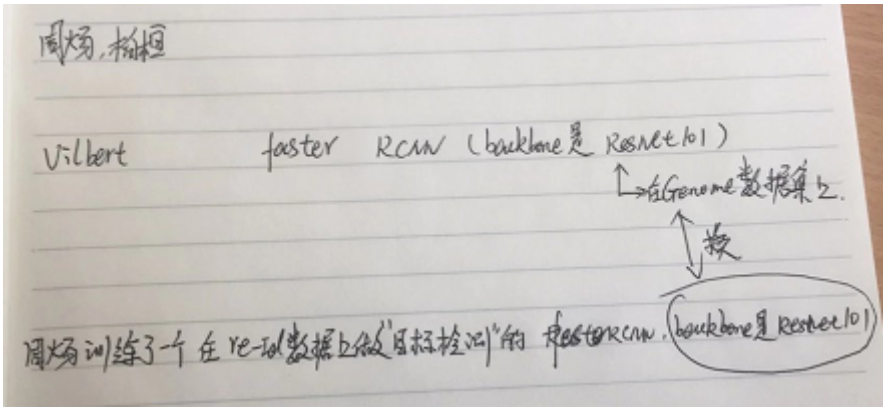
Dual-Path Convolutional Image-Text Embeddings with Instance Loss

Improving Text-based Person Search by Spatial Matching and Adaptive Threshold

Person Search with Natural Language Description

Re-ranking Person Re-identification with k-reciprocal Encoding





layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

1. 基本任务：

- person RE-ID: 使用图片搜图片
- person search：使用文本搜图片

2. 框架图

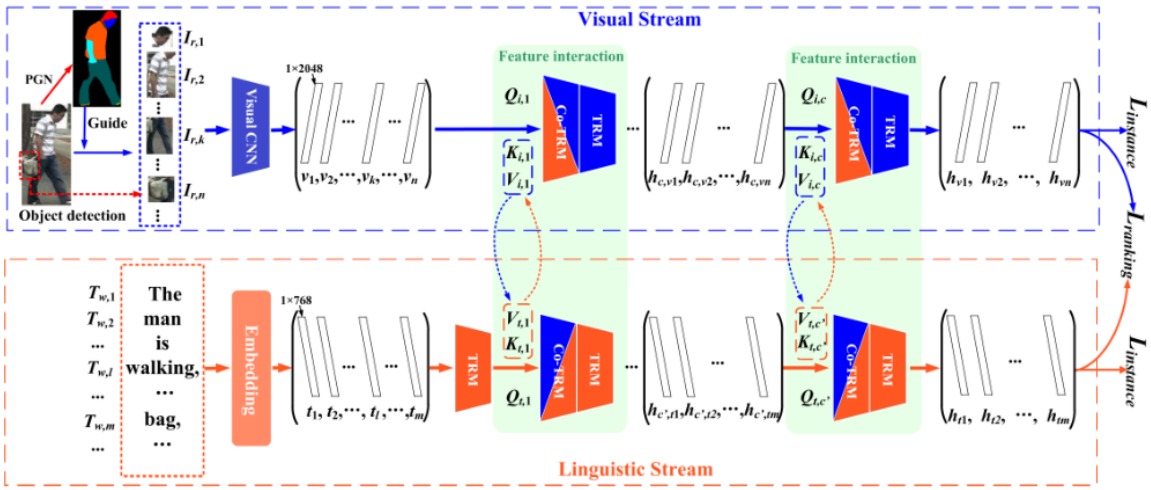


Figure 2. Our text-based person search model consists of two streams for visual and textual processing with triple loss, and the interactions between two modalities are achieved through co-attentional transformer layers.

3. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks

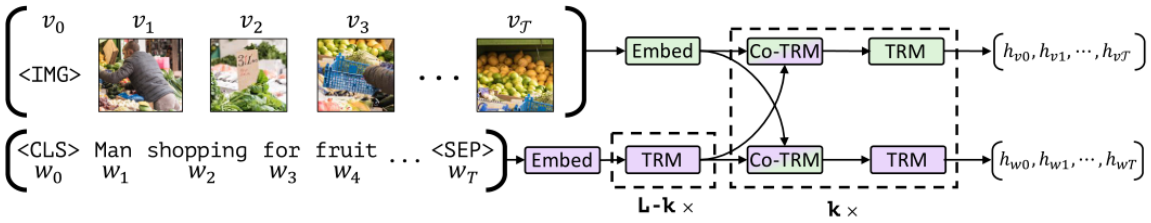


Figure 1: Our ViLBERT model consists of two parallel streams for visual (green) and linguistic (purple) processing that interact through novel co-attentional transformer layers. This structure allows for variable depths for each modality and enables sparse interaction through co-attention. Dashed boxes with multiplier subscripts denote repeated blocks of layers.

- 数据集： Visual Genome dataset
- 任务：文本搜索图片
- 视觉特征的提取：
  - We use Faster R-CNN [31] (with ResNet-101 [11] backbone) pretrained on the Visual Genome dataset [16] (see [30] for details) to extract region features. We select regions where class detection probability exceeds a confidence threshold and keep between 10 to 36 high-scoring boxes. For each selected region i, v_i is defined as the mean-pooled convolutional feature from that region. Transformer and co-attentional transformer blocks in the visual stream have hidden state size of 1024 and 8 attention heads.
  - # 3.1. [30]Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering
    - 视觉特征提取的方法：

      To pretrain the bottom-up attention model, we first initialize Faster R-CNN with ResNet-101 pretrained for classification on ImageNet [35]. We then train on Visual Genome [21] data. To aid the learning of good feature representations, we add an additional training output for predicting attribute classes (in addition to object classes). To predict attributes for region i, we concatenate the mean pooled convolutional feature $\boldsymbol{v}_i$ with a learned embedding of the ground-truth object class, and feed this into an additional output layer defining a softmax distribution over each attribute class plus a 'no attributes' class.

    - 模型框架：

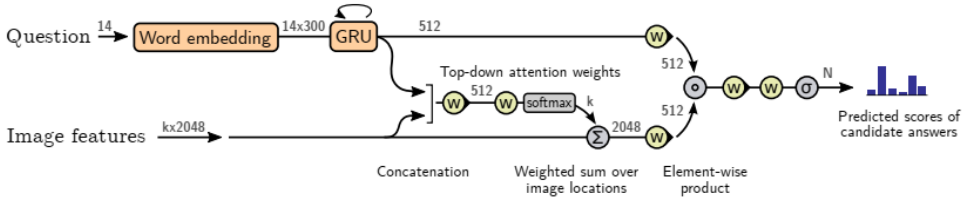


Figure 4. Overview of the proposed VQA model. A deep neural network implements a joint embedding of the question and image features $\{\boldsymbol{v}_1, ..., \boldsymbol{v}_k\}$. These features can be defined as the spatial output of a CNN, or following our approach, generated using bottom-up attention. Output is generated by a multi-label classifier operating over a fixed set of candidate answers. Gray numbers indicate the dimensions of the vector representations between layers. Yellow elements use learned parameters.

4. Dual-Path Convolutional Image-Text Embeddings with Instance Loss

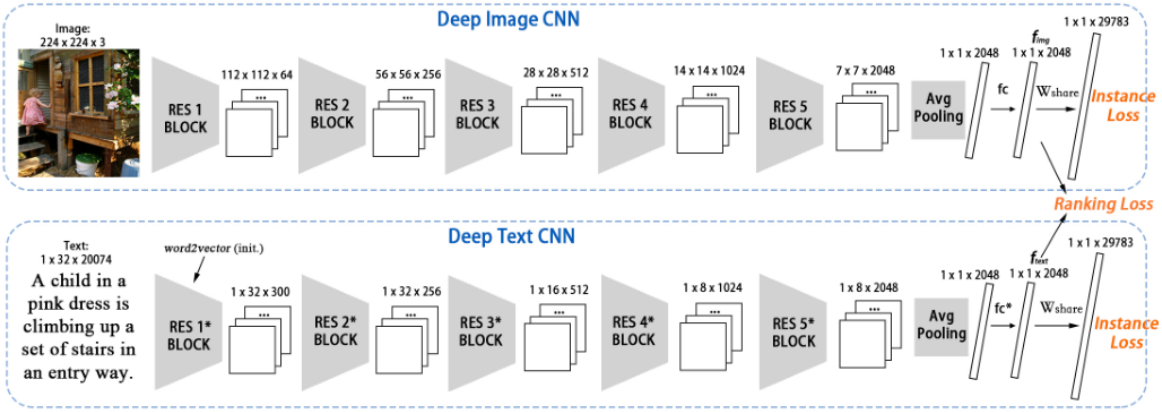


Fig. 2: We learn the image and text representations by two convolutional neural networks, *i.e.*, deep image CNN (top) and deep text CNN (bottom). The deep image CNN is a ResNet-50 model [28] pre-trained on ImageNet. The deep text CNN is similar to the image CNN but with different basic blocks (see Fig. 3). After the average pooling, we add one fully connected layer (input dim: $2,048$, output dim: $2,048$), one batchnorm layer, relu and one fully connected layer (input dim: $2,048$, output dim: $2,048$) in both image CNN and text CNN (We denote as fc and fc* in the figure, and the weights are not shared). Then we add a shared-weight W_{share} classification layer (input dim: $2,048$, output dim: $29,783$). The objectives are the ranking loss and the proposed instance loss. On Flickr30k, for example, the model needs to classify 29,783 classes using instance loss.

- 视觉特征的提取：
  - 之前的任务：

    structure. In the field of image-text matching, most recent methods directly use fixed CNN features [8]–[10], [16]–[22] as input which are extracted from the models pre-trained on ImageNet. While it is efficient to fix the CNN features and learn a visual-textual common space, it may lose the fine-grained differenc Administrator e images. This motivates us to fine-tune the image CNN branch in the image-text matching to provide for more discriminative embedding learning.

  - 这篇文章中，先在ImageNet上进行预训练，之后再进行微调。

    A. Deep Image CNN

    We use ResNet-50 [28] pre-trained on ImageNet [26] as a basic model (the final 1000-classification layer is removed) before conducting fine-tuning for visual feature learning. Given an input image of size 224×224, a forward pass of the network produces a $2,048$-dimension feature vector. Followed

  - 调整策略：

Stage I: In this stage, we fix the pre-trained weights in the image CNN and use the proposed instance loss to tune the remaining part. The main reason is that most weights of the text CNN are learned from scratch. If we train the image and text CNNs simultaneously, the text CNN may compromise the pre-trained image CNN. We only use the proposed instance loss in this stage ($\lambda_1 = 0, \lambda_2 = 1, \lambda_3 = 1$). It can provide a good initialization for the ranking loss. We note that even after Stage I, our network can achieve competitive results compared to previous works using off-the-shelf CNNs.

Stage II: After Stage I converges, we start Stage II for end-to-end fine-tuning of the *entire network*. Note that the weights of the image CNN are also fine-tuned. In this stage, we combine the instance loss with the ranking loss ($\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 1$), so that both classification and ranking errors are considered. In Section VI-D, we study the mechanism of the two losses. It can be observed that in Stage II, instance loss and ranking loss are complementary, thus further improving the retrieval result. Instance loss still regularizes the model and provides more attentions to discriminate the images and sentences. After Stage II (end-to-end fine-tuning), another round of performance improvement can be observed, and we achieve even more competitive performance.

- 这个貌似任务是挺匹配的，但是代码是matlab的

5. Improving Text-based Person Search by Spatial Matching and Adaptive Threshold

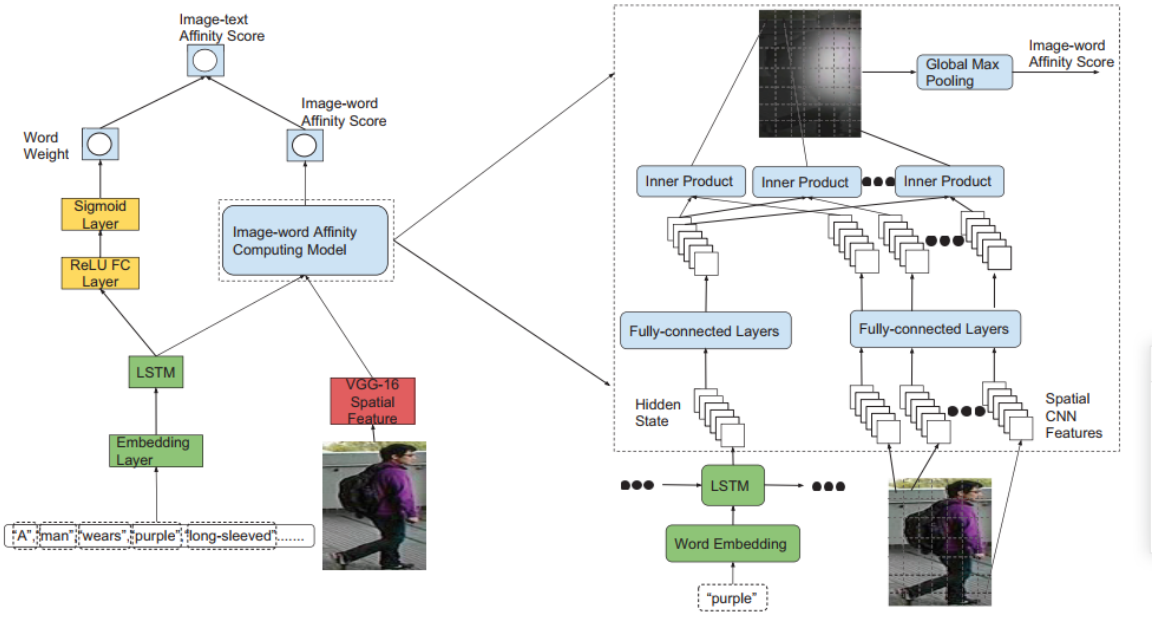


Figure 2. The structure of the proposed patch-word matching framework. It contains an image encoder (red), a text encoder (green), a word attention sub-network (orange) and a computing part to predict the image-text affinity score.

- 视觉特征的提取：
  - Vgg16model：

of the text. Same as [6], the image encoder is a VGG-16 model, which is pre-trained on person re-identification dataset. However, for image I, instead of extracting its global feature from the last fully-connected layer of the image encoder, we extract the feature of I from the last pooling layer, which is a $7 \times 7 \times 512$ tensor. In other words, the

6. Person Search with Natural Language Description

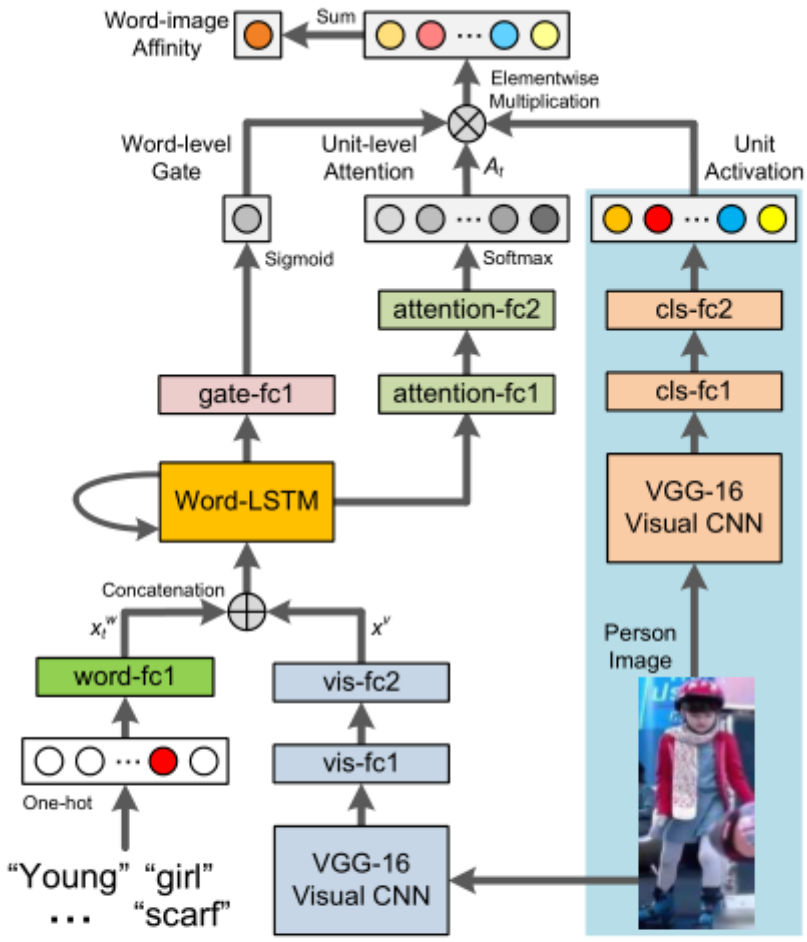


Figure 5. The network structure of the proposed GNA-RNN. It consists of a visual sub-network (right blue branch) and a language sub-network (left branch). The visual sub-network generates a series of visual units, each of which encodes if certain appearance patterns exist in the person image. Given each input word, The language sub-network outputs world-level gates and unit-level attentions for weighting visual units.

7. Re-ranking Person Re-identification with k-reciprocal Encoding

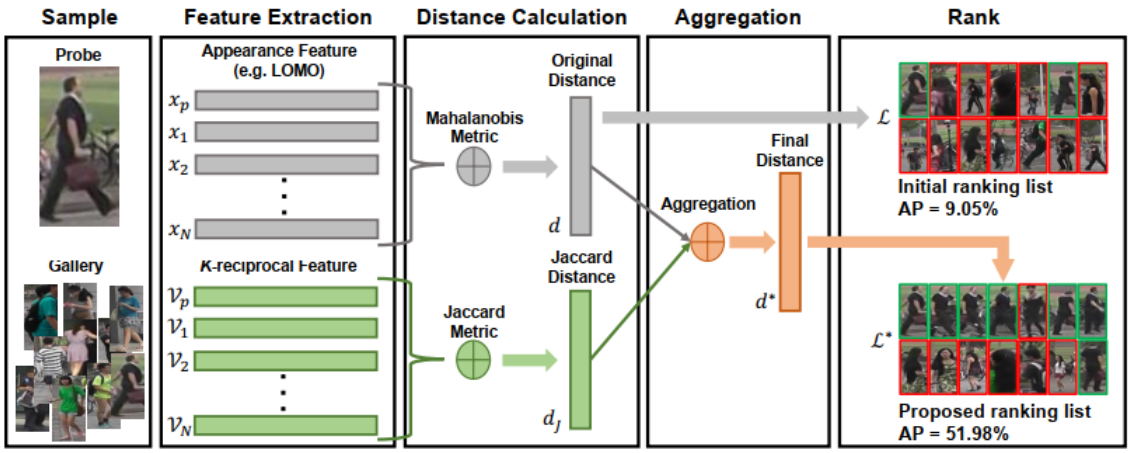


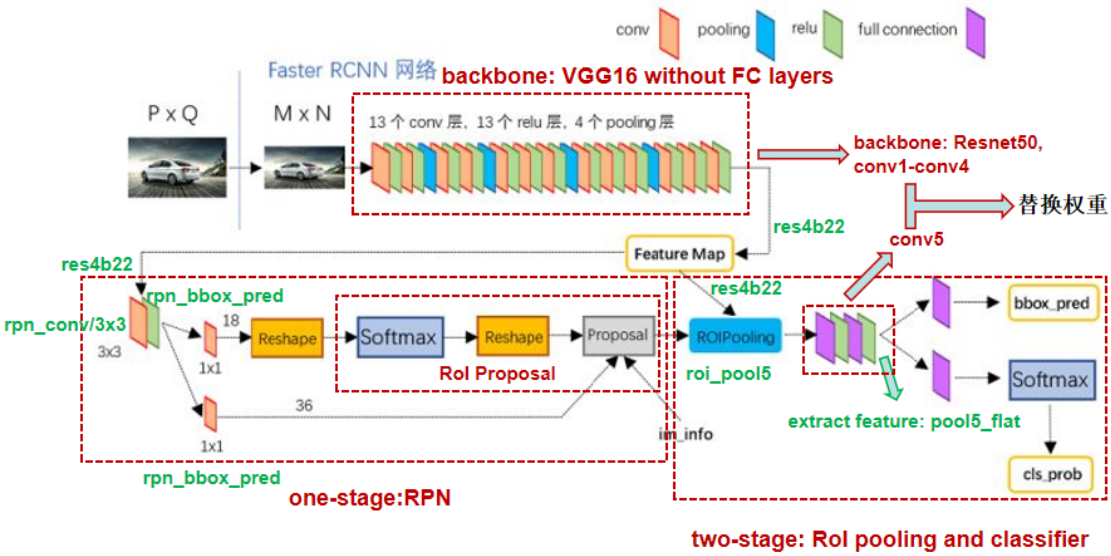
Figure 2. Proposed re-ranking framework for person re-identification. Given a probe p and a gallery, the appearance feature and k-reciprocal feature are extracted for each person. Then the original distance d and Jaccard distance d_J are calculated for each pair of the probe person and gallery person. The final distance d^* is computed as the combination of d and d_J, which is used to obtain the proposed ranking list.

- 视觉特征提取：

  Feature representations The Local Maximal Occurrence (LOMO) features are used to represent the person appearance [23]. The LOMO extractor generate a 26,960-dimensional feature for each image. This feature is robust to view changes and illumination variations by concatenating the maximal pattern of joint HSV histogram and SILTP descriptor, and is also discriminative, by capturing local region characteristics of a person. In addition, the ID-discriminative Embedding (IDE) feature proposed in [52] is used. The IDE extractor is effectively trained on classification model including CaffeNet [18] and ResNet-50 [13]. It generates a 1,024-dim (or 2,048-dim) vector for each image, which is effective in large-scale re-ID datasets. For the convenience of description, we abbreviate the IDE trained on CaffeNet and ResNet-50 to IDE (C) and IDE (R) respectively. We use these two methods as the baseline of our re-id framework.

  - 目前是使用了这里的权重效果均为0.

8. Faster RCNN代码分析



- backbone：resnet
  - 101：

```
layer {
    bottom: "res4b21"
    bottom: "res4b22_branch2c"
    top: "res4b22"
    name: "res4b22"
    type: "Eltwise"
}

layer {
    bottom: "res4b22"
    top: "res4b22"
    name: "res4b22_relu"
    type: "ReLU"
}
```

    最后一层的输出为`res4b22_relu`

- RPN
  - 开始部分层：

```
layer {
  name: "rpn_conv/3x3"
  type: "Convolution"
  bottom: "res4b22"
  top: "rpn/output"
  param { lr_mult: 1.0 }
  param { lr_mult: 2.0 }
  convolution_param {
    num_output: 512
```

```
    kernel_size: 3 pad: 1 stride: 1
    weight_filler { type: "gaussian" std: 0.01 }
    bias_filler { type: "constant" value: 0 }
  }
}
```

输入：res4b22

- 最后一部分：

```
layer {
  name: "rpn_bbox_pred"
  type: "Convolution"
  bottom: "rpn/output"
  top: "rpn_bbox_pred"
  param { lr_mult: 1.0 }
  param { lr_mult: 2.0 }
  convolution_param {
    num_output: 48   # 4 * 12(anchors)
    kernel_size: 1 pad: 0 stride: 1
    weight_filler { type: "gaussian" std: 0.01 }
    bias_filler { type: "constant" value: 0 }
  }
}
layer {
   bottom: "rpn_cls_score"
   top: "rpn_cls_score_reshape"
   name: "rpn_cls_score_reshape"
   type: "Reshape"
   reshape_param { shape { dim: 0 dim: 2 dim: -1 dim: 0 } }
}
```

输出：`rpn_cls_score_reshape`，`rpn_bbox_pred`

- ROI
  - 开始部分层：

```
layer {
  name: "rpn_cls_prob"
  type: "Softmax"
  bottom: "rpn_cls_score_reshape"
  top: "rpn_cls_prob"
}
```

输出：rpn_cls_prob

- 最后一部分层：

```
layer {
  name: 'proposal'
  type: 'Python'
  bottom: 'rpn_cls_prob_reshape'
  bottom: 'rpn_bbox_pred'
  bottom: 'im_info'
  top: 'rois'
  python_param {
    module: 'rpn.proposal_layer'
    layer: 'ProposalLayer'
    param_str: "'feat_stride': 16 \n'scales':
!!python/tuple [4, 8, 16, 32]"
  }
}
```

输出：`rois`

- RCNN
  - 开始部分层

```
layer {
  name: "roi_pool5"
  type: "ROIPooling"
  bottom: "res4b22"
  bottom: "rois"
  top: "roipool5"
  roi_pooling_param {
    pooled_w: 14
    pooled_h: 14
    spatial_scale: 0.0625 # 1/16
  }
}
```

输入：`res4b22`，`rois`