

《车辆再识别》毕业论文节选

本人毕业论文做的是监控视频中的车辆再识别研究，后面工作中可能不会接触到计算机视觉方面的工作，因此，特意将毕业论文所做的工作整理一下，方便大家交流。

《车辆再识别》毕业论文节选 1—车辆再识别概述

视频监控是计算机视觉领域的一项研究热点，广泛应用于公共安全、智能交通等领域。车辆再识别是视频监控系统中的关键技术，可以实现对嫌疑车辆的跨相机追踪，对城市安全具有重大意义。车辆再识别是指对不同监控相机下的车辆图片进行匹配，在非重叠相机不同时间拍摄的视频中找到目标车辆的过程。为实现监控视频中的车辆再识别，本文主要研究了运动车辆检测和车辆特征提取，并形成了一套具有较好准确性的车辆再识别系统。车辆再识别示意图如图 1 所示。

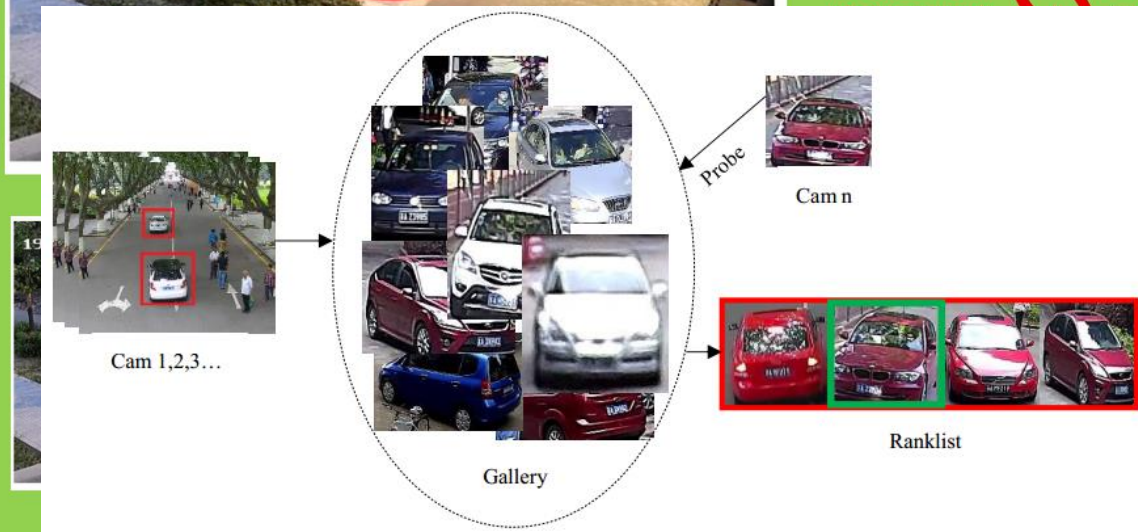


图 1 车辆再识别示意图

在实际情况中，由于光照、姿态、分辨率、相机差异等因素会导致相同车辆在不同监控相机下的差异较大，不同车辆间的差异较小。实际场景中的车辆如图 2 所示。



图 2 相同车辆样本对和不同车辆样本对的异同

监控视频中的车辆再识别主要有两个方面值得关注：一是运动车辆检测；二是特征提取，本文也将围绕着这两个方面展开研究工作，本文的车辆再识别系统如图 3 所示。

视频输入库

图像查询

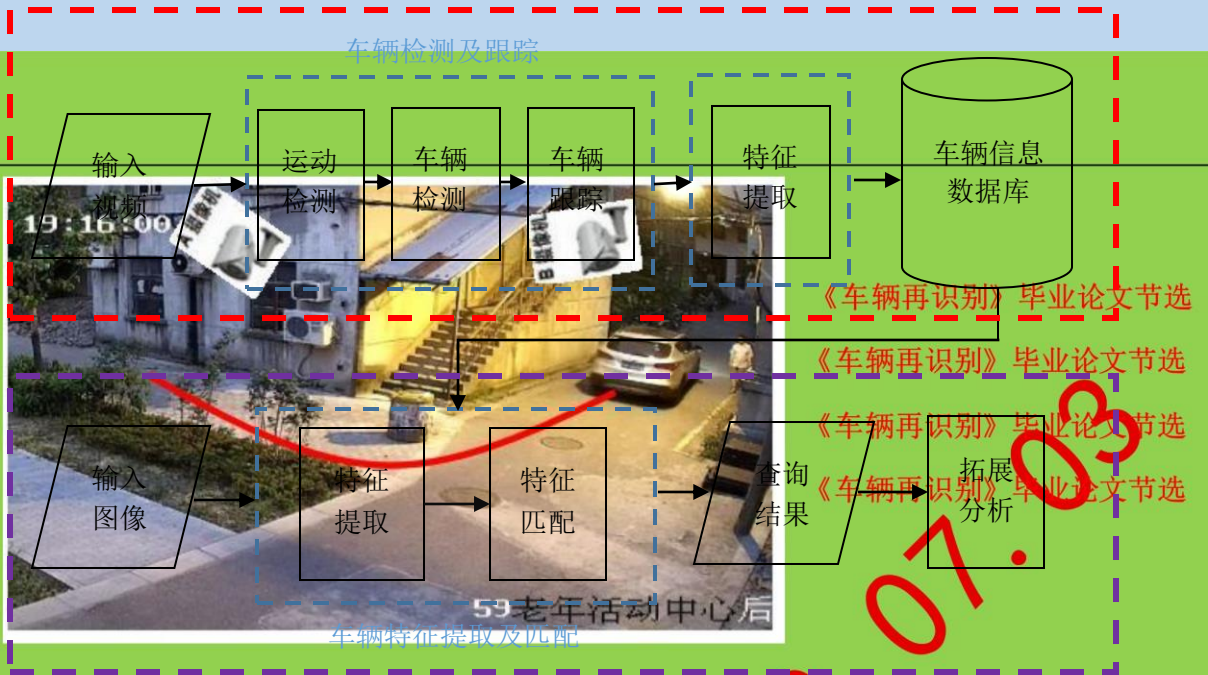


图3 车辆再识别系统



车辆再识别

《车辆再识别》毕业论文节选 2—运动检测

对于监控视频中的车辆再识别，首要的一步就是将视频中的车辆准确地定位出来。本节关注的是监控场景下的运动车辆检测问题，因此可以先进行运动检测得到车辆候选区域，然后再进行车辆精检，这样不仅可以加快检测速度，还能在一定程度上降低虚警。对于运动检测，本文采用 ViBe 算法。本节简要介绍一下 ViBe 算法原理，懂的小伙伴可以略过。

ViBe 背景建模主要包括三部分：背景模型的初始化、前景检测、背景模型的更新。

(1) 背景模型的初始化

ViBe 算法可以通过前几帧视频序列或者第一帧图像为每个点建立背景样本集，即初始化过程。初始化过程利用了空间上相近的点具有相似的像素值这一特性，每个点的背景样本集是随机的选择附近的点得到的。ViBe 背景样本集的建立只需要一帧图像就可以实现。这种方法的好处是计算量较小、初始化速度快，能够很快地用于运动检测。

(2) 前景检测

前景检测就是将下一帧中每个点的像素值和其对应的背景样本集进行距离计算从而确定其是否是前景点。如果该距离小于设定的阈值 R ，那么增加近似背景样本点数目，如果该数目不小于设定的阈值 T ，则该点为背景。背景模型样本集数目 N ，阈值 T 和 R 是前景检测过程中的三个重要参数，一般设置为 $N=20$ ， $T=2$ ， $R=20$ 。前景检测算法如表 1 所示。

(3) 背景模型的更新策略

更新背景模型的主要目的就是让模型能够适应环境的变化，比如光照强度、视频中其他目标的出现与消失等情况。在保守的背景模型更新中，前景点与背景模型的更新无关，这种方法通常会导致死锁现象出现。例如在视频开始的时候，如果一个背景点被错误的判断成前景点，在这种更新策略下该点将会一直被视为前景点，即运动的点。因此，在背景模型的更新中，本文还结合了前景点计数方法，前景点计数就是统计视频帧中的每个点识别为前景的次数，假如视频帧中的一个点连续 M 次被判断成前景点，那么就需要对该点进行更新，将它作为背景点处理。

此外，在每一帧视频中，没有必要对每个点的背景样本集都进行更新。如果一个像素点被判断成背景，那么有 $1/\phi$ 的概率去更新它周围点的背景样本值，同时也有 $1/\phi$ 的概率去更新该点自己的背景样本值；并且假如一个点连续 M 次被判断成前景点，那么就需要将它作为背景点处理，并有 $1/\phi$ 的概率更新它自己的背景样本值，这也就是所谓的随机更新策略。本文的背景更新策略如表 2 所示。

表 1 前景检测算法

算法 1 ViBe 前景检测算法

输入：背景模型 $samples$ ，当前帧图像 $image$ ，背景模型样本集数目 N ，阈值 T ，判定半径 R

输出：前景掩膜图像 $mask$ ，前景点计数 $matchcount$

```
1: for i := 1 to height do
2:   for j := 1 to width do
3:     count=0
4:     for s := 1 to N do
5:       d=dist(image[i][j], samples[s][i][j]) //计算当前点和其对应的背景集的距离
6:       if d<R then count++
7:     end if
8:   end for
9:   if count>T then mask[i][j]=0
10:  else mask[i][j]=1, matchcount[i][j] += 1
11: end if
```




```
12: end for
13: end for
```

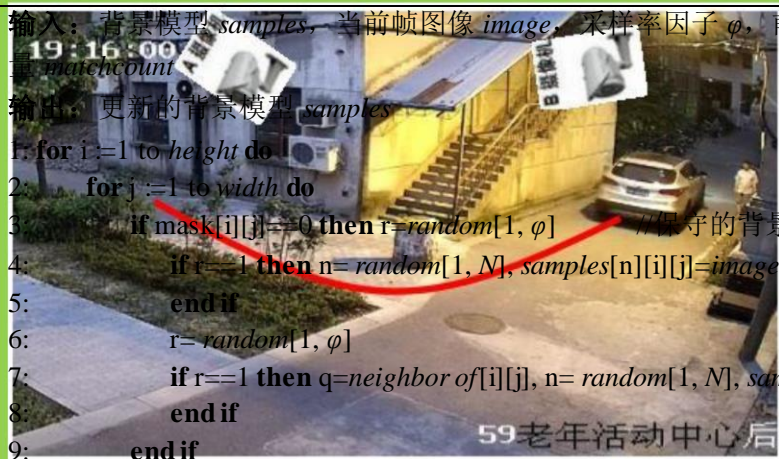
表 2 背景更新算法

算法 2 ViBe 背景更新算法

输入：背景模型 $samples$ ，当前帧图像 $image$ ，采样率因子 ϕ ，前景点计数阈值 M ，前景点计数变量 $matchcount$

输出：更新的背景模型 $samples$

```
1: for i = 1 to height do
2:   for j = 1 to width do
3:     if mask[i][j] == 0 then r = random[1,  $\phi$ ] //保守的背景模型更新
4:       if r == 1 then n = random[1, N], samples[n][i][j] = image[i][j]
5:     end if
6:     r = random[1,  $\phi$ ]
7:     if r == 1 then q = neighbor of [i][j], n = random[1, N], samples[n][q] = image[i][j]
8:   end if
9: end if
10: if matchcount[i][j] > M then r = random[1,  $\phi$ ] //前景点计数
11:   if r == 1 then n = random[1, N], samples[n][i][j] = image[i][j]
12: end if
13: mask[i][j] = 0, matchcount[i][j] = 0
14: end if
15: end for
16: end for
```



在得到前景区域后，通过滤波以及连通域标记就可以得到运动矩形框，即车辆候选区域。



《车辆再识别》毕业论文节选 3—车辆检测

运动检测可以获得车辆候选区域，但是并不能准确的得到车辆矩形框，因此需要车辆精检。实际场景中，车辆姿态多种多样，为了能将各种姿态的车辆都检测出来，本文研究了 DPM 模型。具体 DPM 模型算法原理可以参考论文，或者网上博客，因为这种模型太牛逼了，在此不再将罗列论文链接。本节再罗列一下 DPM 检测流程，具体细节参考原文。

DPM 模型的尺寸是固定的，但是图片中的车辆尺寸是变化的，为了能够检测到不同尺寸的车辆，需要进行多尺度分析，因此需要对图片进行缩放，建立图像金字塔，得到输入图像不同尺度下的 HOG 特征。下面分析单个组件模型在 HOG 特征图像金字塔其中一层的检过程。

3.1.1 Object Hypothesis

假设一个尺寸为 $w \times h$ 的滤波器用 F 表示，HOG 特征金字塔用 H 表示，金字塔第 l 层坐标为 (x, y) 的位置用 $p = (x, y, l)$ 表示，金字塔 H 中第 l 层左上角坐标为 (x, y) 尺寸为 $w \times h$ 的矩形区域上的 HOG 特征用 $\phi(H, p, w, h)$ 表示，那么 $F \cdot \phi(H, p, w, h)$ 即是位置 p 的响应。因为 F 已经指定了矩形区域的大小，所以响应的表达式可以简化成 $F \cdot \phi(H, p)$ 。

一个目标假设(Object Hypothesis)指定每个根模型和部件模型的位置 $z = p_0, p_1, \dots, p_n$ ，其中 $p_i = (x_i, y_i, l_i)$ 指定了第 i 个滤波器放置的位置， l_i 是 HOG 金字塔的层编号， (x_i, y_i) 是滤波器 F_i 左上角的位置。部件滤波器放置的层是根模型的两倍，对于任意的 $i > 0$ ，有 $l_i = l_0 + 1$ 。位置 z 的得分是以下几部分的和：第一，根滤波器和 HOG 的内积，第二，每个部件滤波器和 HOG 特征的内积，减去部件滤波器位置偏移造成的代价，第三部分是偏移量。虽然 DPM 模型中规定了部件模型相对于根模型的偏移量，但是并不规定部件模型在检测时一定要处于训练出来的位置，而是允许部件模型任意移动。部件模型位置偏移后，要结合形变系数对内积运算得到的结果进行惩罚。位置 z 的响应为：

$$score(p_0, p_1, p_2, \dots, p_n) \quad (0-1)$$

$$= F_0 \cdot \phi(H, p_0) + \sum_{i=1}^n [F_i \cdot \phi(H, p_i) - d_i \cdot \delta(p_0, p_i)] + b \quad (0-2)$$

$$= F_0 \cdot \phi(H, p_0) + \sum_{i=1}^n [F_i \cdot \phi(H, p_i) - d_i \cdot \phi_d(\Delta x_i, \Delta y_i)] + b \quad (0-3)$$

$$= F_0 \cdot \phi(H, p_0) + \sum_{i=1}^n [F_i \cdot \phi(H, p_i) - (d_{i,x} \cdot \Delta x_i + d_{i,y} \cdot \Delta y_i + d_{i,x^2} \cdot (\Delta x_i)^2 + d_{i,y^2} \cdot (\Delta y_i)^2)] + b \quad (0-4)$$

其中，

$$\Delta x_i = x_i - (2x_0 + v_{i,x}), \Delta y_i = y_i - (2y_0 + v_{i,y}) \quad (0-5)$$

$$\delta(p_0, p_i) = \phi_d(\Delta x_i, \Delta y_i) = (\Delta x_i, \Delta y_i, (\Delta x_i)^2, (\Delta y_i)^2) \quad (0-6)$$

从而，位置 z 的 $score$ 可以表示成：

$$score(p_0, p_1, p_2, \dots, p_n) = \beta \cdot \phi(H, z) \quad (0-7)$$

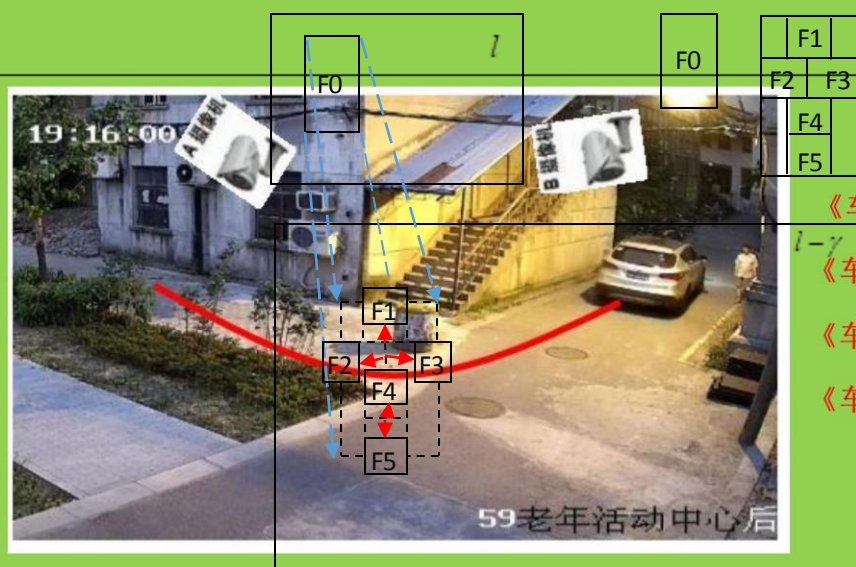
其中，

$$\beta = (F_0, F_1, \dots, F_n, d_1, \dots, d_n, b) \quad (0-8)$$

$$\phi(H, z) = (\phi(H, p_0), \phi(H, p_1), \dots, \phi(H, p_n), -\phi_d(\Delta x_1, \Delta y_1), \dots, -\phi_d(\Delta x_n, \Delta y_n), 1) \quad (0-9)$$

β 就是模型的滤波器、形变系数和偏置量的组合， $\phi(H, z)$ 是根据 z 从 HOG 金字塔中选定出合适的 HOG 特征以及距离偏移惩罚量的组合。图 4 是 Object Hypothesis 示意图，其中 F_0 表示根滤波器，

F_1, F_2, \dots, F_5 表示部件滤波器，部件滤波器放在两倍于根滤波器所在层上，并且允许部件滤波器在一定范围内移动，最后的响应就是此位置的得分。



《车辆再识别》毕业论文节选
《车辆再识别》毕业论文节选
《车辆再识别》毕业论文节选
《车辆再识别》毕业论文节选

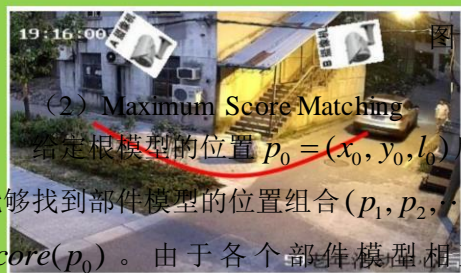


图4 Object Hypothesis 示意

(2) Maximum Score-Matching

给定根模型的位置 $p_0 = (x_0, y_0, l_0)$ 后，通过在 HOG 特征金字塔 $l_0 - \gamma$ 层调节部件模型的位置，总能够找到部件模型的位置组合 (p_1, p_2, \dots, p_n) ，使得 $score(p_0, p_1, \dots, p_n)$ 达到最大，记这个最大值为 $score(p_0)$ 。由于各个部件模型相互独立，各个部件模型的运算结果都达到最大时， $score(p_0, p_1, \dots, p_n)$ 达到最大。

令 $R_{i,l}(x, y) = F_i \cdot \phi(H, (x, y, l))$ 表示第 i 个部件滤波器和第 l 层 HOG 特征金字塔内积运算结果。假设根模型的位置是 $p_0 = (x_0, y_0, l_0)$ ，则第 i 个部件模型的理想位置是 $p_i = (2x_0 + v_{i,x}, 2y_0 + v_{i,y}, l_0 - \gamma)$ ；如果第 i 个部件模型在第 $l_0 - \gamma$ 层中相对理想位置发生偏移 $(\Delta x_i, \Delta y_i)$ ，达到 $(x_i + \Delta x_i, y_i + \Delta y_i, l_0 - \gamma)$ ，那么该部件对最终得分贡献是：

$$T_{i,l_0-\gamma}(x_i + \Delta x_i, y_i + \Delta y_i) = R_{i,l_0-\gamma}(x_i + \Delta x_i, y_i + \Delta y_i) - d_i \cdot \phi(\Delta x_i, \Delta y_i) \quad (0-10)$$

$$D_{i,l_0-\gamma}(x_i, y_i) = \max_{\Delta x_i, \Delta y_i} T_{i,l_0-\gamma}(x_i + \Delta x_i, y_i + \Delta y_i) \quad (0-11)$$

这个值就是给定了根模型的位置 $p_0 = (x_0, y_0, l_0)$ 后，第 i 个部件模型能够得到的最大值，求这个最大值可以采用距离变换实现。 $score(p_0)$ 是由 R_{0,l_0} 和 n 个部件模型 $D_{i,l_0-\gamma}(x_i, y_i)$ 与 b 相加得到的。

$$score(p_0) \quad (0-12)$$

$$= \max_{p_1, \dots, p_n} score(p_0, p_1, p_2, \dots, p_n) \quad (0-13)$$

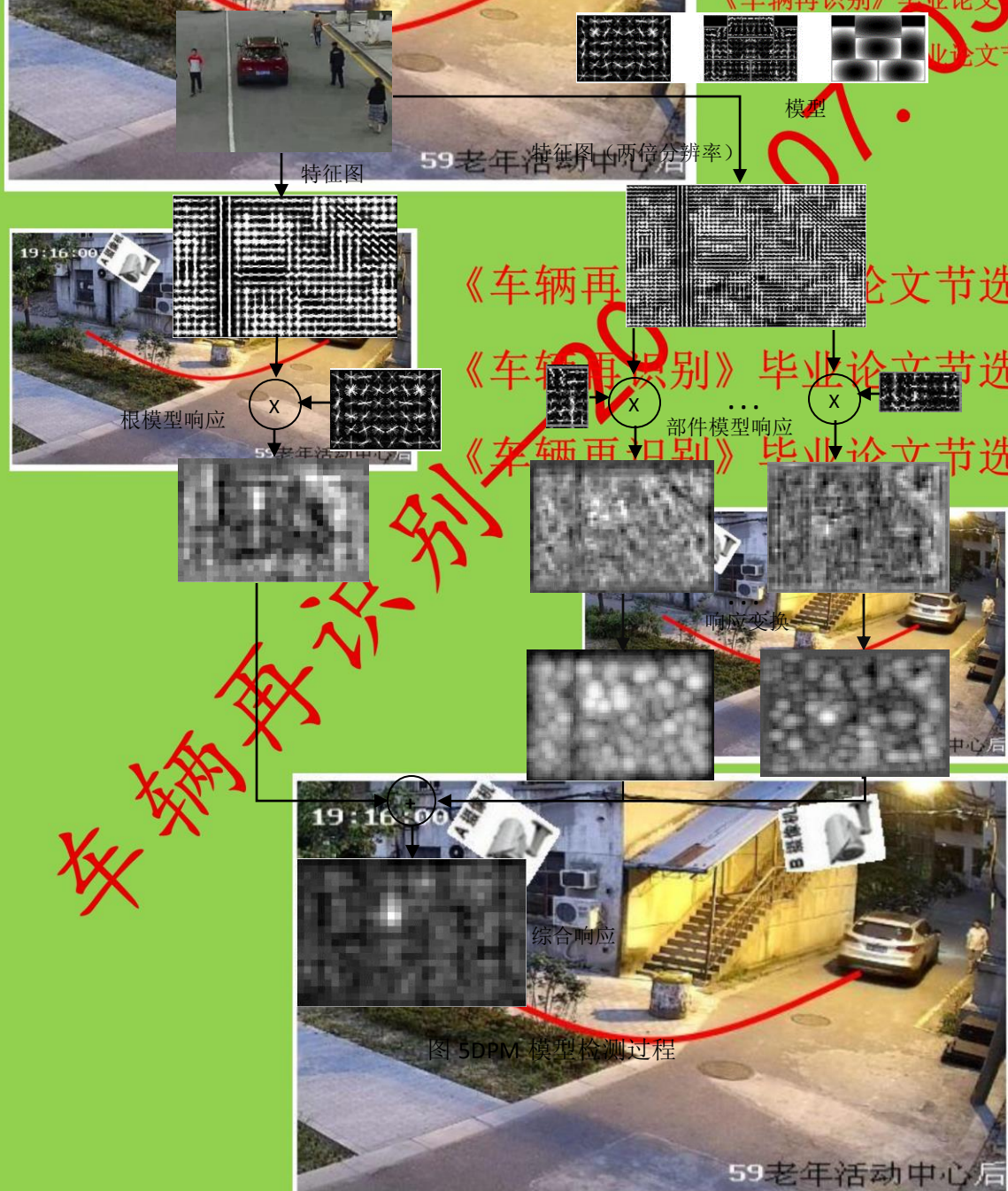
$$= F_0 \cdot \phi(H, p_0) + \max_{p_1, \dots, p_n} \sum_{i=1}^n [F_i \cdot \phi(H, p_i) - d_i \cdot \phi(p_0, v_i, p_i)] + b \quad (0-14)$$

$$= R_{0,l_0}(x_0, y_0) + \sum_{i=1}^n \max_{\Delta x_i, \Delta y_i} T_{i,l_0-\gamma}(x_i + \Delta x_i, y_i + \Delta y_i) + b \quad (0-15)$$

$$= R_{0,l_0}(x_0, y_0) + \sum_{i=1}^n D_{i,l_0-\gamma}(x_i, y_i) + b \quad (0-16)$$

$$= R_{0,l_0}(x_0, y_0) + \sum_{i=1}^n D_{i,l_0-\gamma}(2(x_0, y_0) + v_i) + b \quad (0-17)$$

DPM 模型采用了窗口滑动的检测策略，也就是通过构建不同尺度的特征金字塔，在多个尺度下计算滤波器与特征图的响应。图 5 为某一尺度下车辆的检测过程，即滤波器与特征图的匹配过程，模型在 (x, y) 处的响应得分是该模型与以该坐标为锚点的子窗口区域内的特征的内积，此处的模型即是一个滤波算子。图左侧为根模型与特征图的计算过程，图右侧是各部件模型与特征图的计算过程，通过模型和特征图的内积可以得到响应图，然后进行响应变换，得到该尺度下根模型和部件模型的最佳位置。



《车辆再识别》毕业论文节选 4—颜色特征提取

从本节开始，将着手研究提取车辆特征的方法。在监控视频下，车牌信息已经不能再利用，因此需要寻找其他手段，纹理特征在这种条件下不见得有多好。虽然在论文《Person Re-identification by Local Maximal Occurrence Representation and Metric Learning》里面，作者提出 LOMO (HSV+SILTP) 特征结合 XQDA 度量方法，并且取得了很好的结果；但是后来我单独验证了一下 SILTP，其实效果一般，也可能是我姿势不对。我觉得更有效的低级特征可能也就是颜色特征吧，当然了形状啥的也能用。本文只研究了颜色特征。

目前对颜色研究比较好的两篇论文《Learning Color Names for Real-World Applications》和《Parametric fuzzy sets for automatic color naming》，这两篇论文是通过建模得到一个像素点到 11 种颜色的映射，我比较菜，只能人工建立映射，哎。后面有时间的话，我会专门写一篇比较这三种颜色命名方法的博文。

本文将 HSV 空间划分为黑、灰、白、红、黄、蓝、绿、青、紫等 9 种基本颜色。为了更好地从 HSV 空间划分出上述 9 种颜色各自所属的颜色区域，采用产生式颜色表构建方法，颜色表的构建策略具体如下：

- (1) 采用亮色分离策略，首先按照 V 分量的值将 HSV 颜色空间分成若干类色盘 $V_i (i=1, 2, \dots, m)$ ；
- (2) 对每个 V_i 色盘，按照 S 分量的值将其分成若干类色环 $VS_j (j=1, 2, \dots, n)$ ；
- (3) 在 VS_j 空间中按照 H 分量的值依次生成一系列颜色块，通过人眼观察，并进行投票表决出该颜色块属于每种颜色的概率，选择概率最高且大于一定阈值的类别为该块的颜色标签。

通过上述 3 个步骤即可确定出每种颜色各自的 H、S、V 三个分量的取值范围。经过上述步骤，可以确定当 $V \leq V_{low}$ 时，由于亮度过低，则将该区域视为黑色；当 $V_{low} < V \leq V_{high}$ 且饱和度 $S < S_{th1}$ 时，则将该区域视为灰色；当 $V_{high} < V$ 且饱和度 $S < S_{th2}$ 时，则将该区域视为白色。经过大量的人工标记，生成如表所示的颜色对照表。

表 3 颜色对照表

V=255-225 颜色对照表							
S \ H	Red	Yellow	Green	Cyan	Blue	Purple	Red
255-225	0-20	24-36	37-87	88-92	94-130	134-152	154-179
224-175	0-20	24-36	37-87	88-92	94-128	132-152	154-179
174-125	0-20	24-36	37-87	88-92	94-124	128-152	156-179
75-45	0-20	26-32	38-87	88-92	96-115	130-150	156-179
44-25	0-20	26-32	38-87	88-92	96-115	130-150	158-179
74-50	0-10	30-30	40-88	90-90	96-124	136-148	162-179
V=44-40 颜色对照表							
S \ H	Red	Yellow	Green	Cyan	Blue	Purple	Red
255-225	0-16	28-30	38-86	90-90	94-128	134-150	156-179
224-195	0-16	30-30	40-86	90-90	93-128	136-150	156-179
194-190	0-10	30-30	40-86	90-90	93-128	136-148	160-179

根据实际需要,把 HSV 空间划分成红、黄、绿、青、蓝、紫、黑、灰、白 9 种颜色,并用不同的数值将其标记,结果如表所示。

表 4 颜色标记

颜色	黑	白	红	黄	绿	青	蓝	紫	灰
	0	1	2	3	4	5	6	7	8

因此,通过颜色对照表可以得到 HSV 空间的一个点到 9 种索引颜色的映射,即将一个点映射成为一个 9 维向量,利用此映射可以对图像进行编码。

从表可知,每个色环的 H 值并不连续,也即每两种颜色之间存在过渡区域,因为该区域凭借人眼难以判定属于哪种颜色。此时采用模糊量化策略,将其与相邻的颜色类关联,属于各类颜色的隶属度,进而估计属于各颜色类的概率。

通过查表 3,可以将一个像素点 n 同 C 个颜色类别进行关联: $u(n) = \{u_1(n), u_2(n), \dots, u_C(n)\}$, $u(n)$ 是像素点 n 属于各个颜色类别的隶属度向量, C 是颜色类数。图是采用 9 维颜色概率直方图对车辆进行描述,可以看出该向量可以较好地车辆进行描述,同时可以作为一种语义特征进行索引。

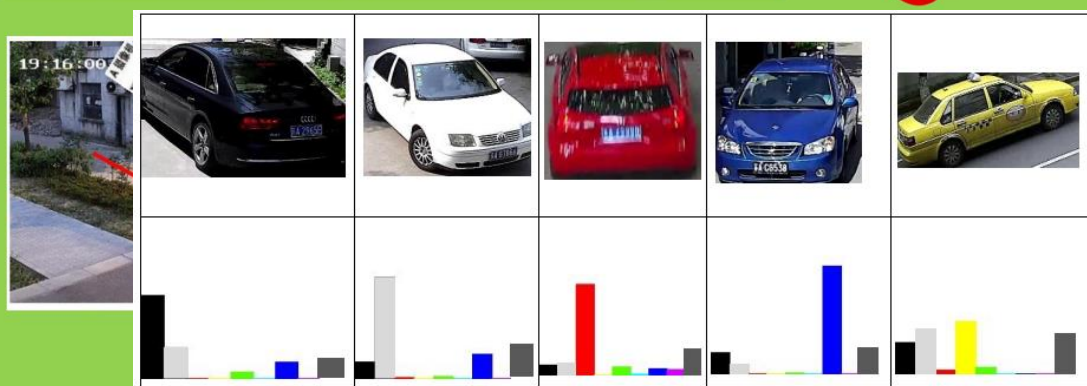


图 6 颜色概率直方图

为了能够利用局部位置的颜色信息,本文在提取车辆的颜色特征描述符时,采用滑窗的方式描述车辆的细节,该方法主要借鉴局部最大出现频次特征(Local Maximal Occurrence, LOMO)的思想。详细的车辆颜色特征描述符的提取参考图。

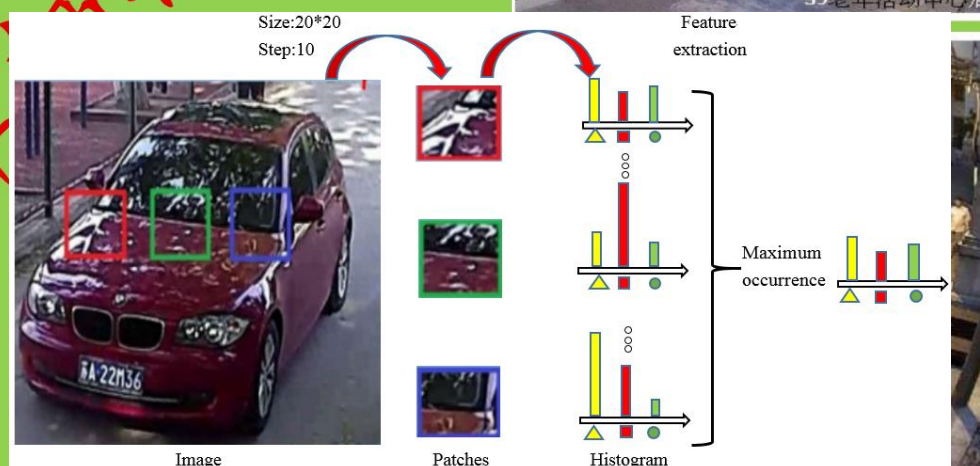


图 7 提取颜色特征描述符

《车辆再识别》毕业论文节选 5—深度特征(AlexNet 网络)提取之一

对于深度特征,本文首先对 AlexNet 网络进行了微调,将再识别看作分类为体对待,从而提取特征。有不少学者采用分类网络进行再识别研究,虽然也有一定的效果,但是目标再识别并不是一个分类问题,区别如表 5 所示。

表 5 车辆再识别与图像分类和图像检索的异同			
任务	类别	训练数据类别	测试数据类别
图像分类	有限	每类都有	全部可见
图像检索	无限	无	不可见
车辆再识别	无限	部分类别有	全部类别有

AlexNet 网络的具体细节我就不再介绍,对于如何微调 AlexNet 我也不再介绍,后面有时间会写博客对这些细节进行介绍。是采用微调后的 AlexNet 网络训练得到的第一个卷积层(conv1 $96 \times 3 \times 11 \times 11$)权重可视化,该层包含 96 个卷积核,每个卷积核的大小为 11×11 。一个卷积核可以对图像提取一种特征,多个卷积核表示可以对图像提取多种特征。可以看出,本文的 AlexNet 网络学习到的权重较为平滑。



图 8 卷积层 conv1 权重可视化
图是输入到该网络的车辆图片($23 \times 227 \times 227$), 图是通过第一个卷积层(conv1 $96 \times 3 \times 11 \times 11$)作用后得到的 96 个特征图, 第一个卷积层中的卷积核移动步长为 4, 从而得到的每个特征图大小为 55×55 。



图 9 输入图片

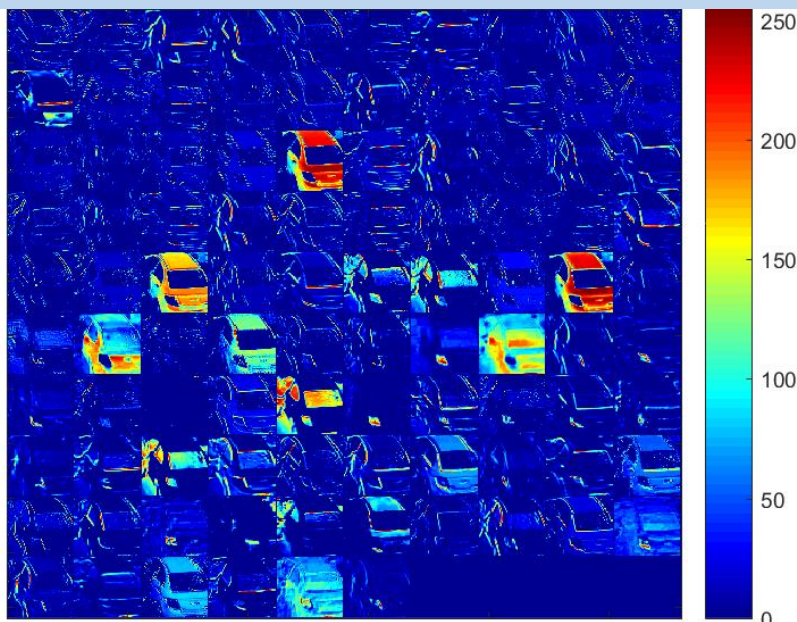


图 10 卷积层 conv1 学习的特征



《车辆再识别》毕业论文节选
《车辆再识别》毕业论文节选
《车辆再识别》毕业论文节选

车辆再识别

《车辆再识别》毕业论文节选 5—深度特征(Siamese 网络)提取之二

车辆再识别简单来说就是计算两张图片的相似度。前面将车辆再识别看作一个分类问题，虽然 AlexNet 网络可以提取出有效的特征用于车辆检索，但是实际上车辆再识别并不是一个分类问题。而 Siamese 网络是一种用于图片验证的网络，该网络将特征提取和度量学习融合在一起训练，可以度量两张图片的相似程度，因此该网络比较适用于本文的再识别问题。

图 11 是经典的 Siamese 网络结构图，图中包含两个并行、权重共享的网络。其中 x_1 和 x_2 为输入的两张车辆图片， $f(x_1)$ 和 $f(x_2)$ 分别是经过网络后得到的特征，然后构造两个向量的欧式距离作为两幅输入图片的相似度。

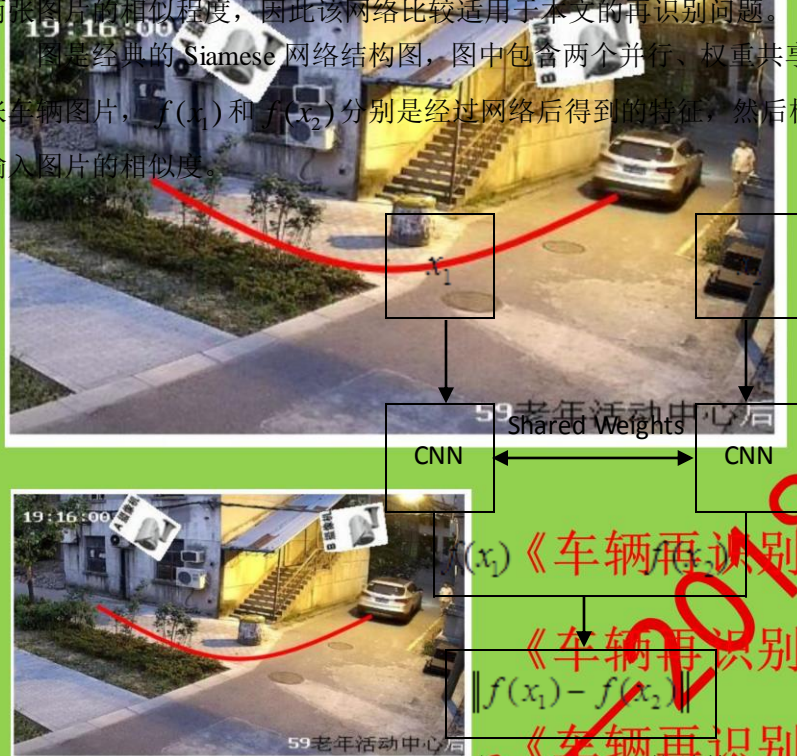


图 11 Siamese 网络结构

该网络的主要思想就是将输入映射到一个目标空间，该空间采用欧氏距离等度量函数进行特征相似度计算。在训练阶段，使得同一类别样本对的距离尽可能小，不同类别样本对的距离尽可能大。

本文参考了 AlexNet 网络的层参数设置并作适当调整，设计了一个用于车辆图片验证的 Siamese 网络结构。其中一些层参数的设置来源于 AlexNet 网络，并做了适当改进，因为 AlexNet 是一种很成功的网络，利用 AlexNet 的一些层参数设置比从头设计效果要好。

对于如何把 AlexNet 网络改成 Siamese 网络结构，并且如何生成自己的 Siamese 网络的数据格式，后面有时间的话，写一些博客进行介绍。



《车辆再识别》毕业论文节选 5—深度特征(改进的 Siamese 网络)提取之三

本文对传统的 Siamese 网络进行了改进，在损失函数中除了对比损失，还结合了 Softmax 损失，实验结果表明比只利用单一损失效果要好。

下面对改进的 Siamese 网络结构的训练过程进行描述。该网络有两个损失函数，一个是 Softmax 损失，一个是对比损失。Softmax 层输出一个 n 维的向量表示输入的的车辆图片属于每一类的概率，采用的交叉熵损失，其表达如下：

$$L_{ident}(f, t, \theta_{id}) = -\sum_{i=1}^n p_i \log \hat{p}_i = -\log \hat{p}_t \quad (0-18)$$

设 x 表示输入的的车辆图片， θ_{cn} 表示卷积网络的参数，则 $f(x, \theta_{cn})$ 是通过 FC6 层提取的向量。 n 表示车辆类别总数， t 表示车辆类别标签， θ_{id} 表示 FC7 层的参数， p_i 是车辆类别的概率分布，有 $p_i = 0 (i \neq t)$ ， \hat{p}_i 表示预测的概率分布。

对比损失公式如下：

$$L_{verif}(f_i, f_j, y_{i,j}) = \begin{cases} \frac{1}{2} \|f_i - f_j\|_2^2, & y_{i,j} = 1 \\ \frac{1}{2} \max(0, m - \|f_i - f_j\|_2)^2, & y_{i,j} = 0 \end{cases} \quad (0-19)$$

设 x 表示输入的的车辆图片， θ_{cn} 表示卷积网络的参数，则 $f(x, \theta_{cn})$ 是通过 FC6 层提取的向量。 $y_{i,j} = 1$ 表示 f_i 和 f_j 来自于相同类别的车辆图片，此时最小化两个向量的欧氏距离， $y_{i,j} = 0$ 表示 f_i 和 f_j 来自于不同类别的车辆图片，意味着要使这两个向量的距离大于间隔 m ，实验中设置 $m = 1$ 。

改进后的 Siamese 网络结构的损失函数为两个损失的加权和：

$$L = L_{ident}(f, t, \theta_{id}) + \lambda \cdot L_{verif}(f_i, f_j, y_{i,j}) \quad (0-20)$$

本文采用梯度下降法优化 θ_{id}, θ_{cn} ，算法如表 6 所示

图所示是输入到该网络的车辆图片(3*227*227)，图所示为通过第一个卷积层(conv1:96*3*11*11)作用后得到的 96 个特征图。



图 12 输入图片

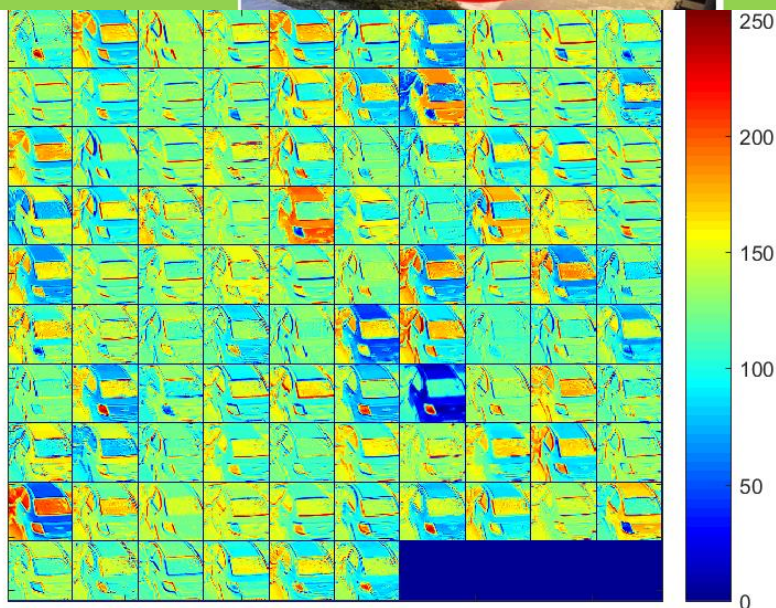


图 13 卷积层 conv1 学习的特征

表 6 参数求解

算法 3 梯度下降法求解参数 θ_{id} 和 θ_{cn}

输入：训练样本集 $\mathcal{X} = \{(x, l)\}$ ，初始化参数 θ_{id} 和 θ_{cn} ，超参数 λ ，初始学习率 $\eta(t), t \leftarrow 0$

输出：参数 θ_{id} 和 θ_{cn}

1 while 损失值不收敛 do

2 $t \leftarrow t + 1$, 从 \mathcal{X} 中取出两个样本 $\{(x_i, l_i)\}$ 和 $\{(x_j, l_j)\}$

3 $f_i = \text{Conv}(x_i, \theta_{cn}), f_j = \text{Conv}(x_j, \theta_{cn})$

4 $\nabla \theta_{id} = \frac{\partial L_{ident}(f_i, l_i, \theta_{id})}{\partial \theta_{id}} + \frac{\partial L_{ident}(f_j, l_j, \theta_{id})}{\partial \theta_{id}}$

5 $\nabla f_i = \frac{\partial L_{ident}(f_i, l_i, \theta_{id})}{\partial f_i} + \lambda \frac{\partial L_{verif}(f_i, f_j, y_{i,j})}{\partial f_i}$

6 $\nabla f_j = \frac{\partial L_{ident}(f_j, l_j, \theta_{id})}{\partial f_j} + \lambda \frac{\partial L_{verif}(f_i, f_j, y_{i,j})}{\partial f_j}$

7 $\nabla \theta_{cn} = \frac{\partial f_i}{\partial \theta_{cn}} \nabla f_i + \frac{\partial f_j}{\partial \theta_{cn}} \nabla f_j$

8 更新 $\theta_{id} = \theta_{id} - \eta(t) \nabla \theta_{id}, \theta_{cn} = \theta_{cn} - \eta(t) \nabla \theta_{cn}$

9 end while

对于如何在 Siamese 网络中加入 Softmax 损失，并且如何生成需要的数据的数据格式（此部分需要改动 caffe 源码），后面有时间的话写博客进行介绍。

《车辆再识别》毕业论文节选

《车辆再识别》毕业论文节选

《车辆再识别》毕业论文节选

《车辆再识别》毕业论文节选

《车辆再识别》毕业论文节选

《车辆再识别》毕业论文节选

《车辆再识别》毕业论文节选



《车辆再识别》毕业论文节选 6—实验结果

1 前期准备工作

为方便对车辆数据进行管理以及车辆再识别系统的搭建,本文采用 xml 格式对每一个车辆序列进行存储,一个车辆序列的xml层次示意图如图 14 所示。每个车辆序列的描述文件包含对象ID(objectID)、累计出现的帧数(nFrames)、以及车辆的各关键帧(frameData)的帧号(frameindex)、位置(rect)、特征(feature)以及截图(thumbnail)等。每段序列的车辆截图文件的路径存储在对象描述文件的 thumbnail 结点的内容中,特征文件和截图文件一一对应,特征文件的路径存储在 feature 节点的内容中,特征文件内容采用 JSON 格式存储。

通过对车辆进行检测及跟踪,可以得到一个车辆目标在一个监控相机下的视频序列。在进行匹配时,只要匹配到该序列中的任何一张图片,通过序列的存储信息即可索引出该车辆在这个监控相机下出现的起始、终止时间以及其他关键帧信息。

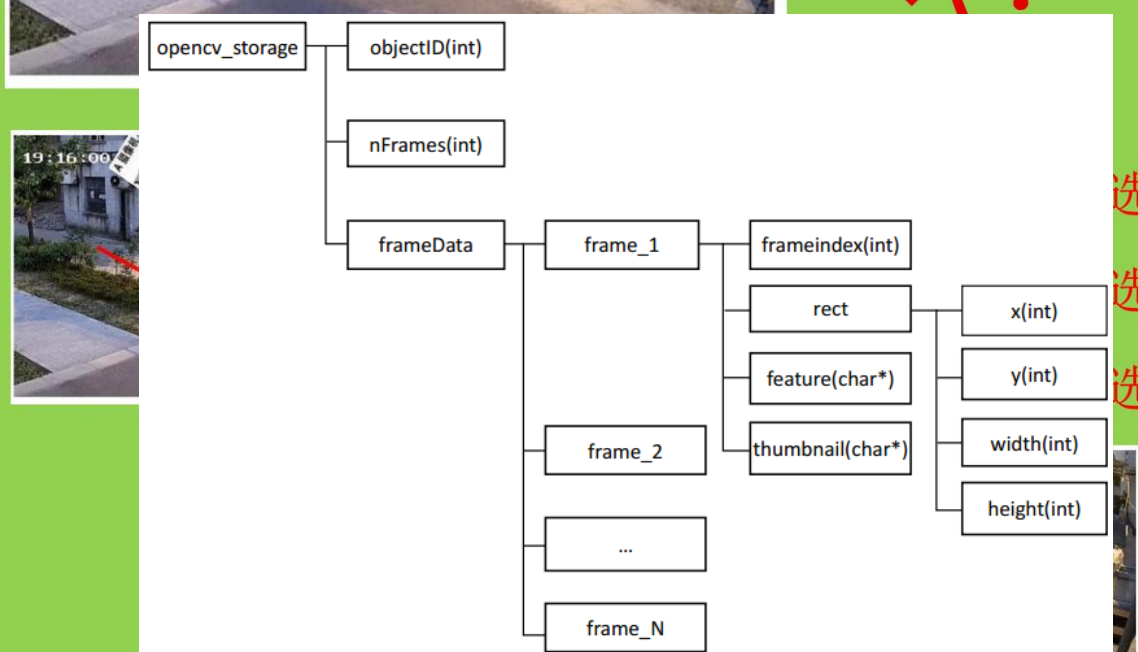


图 14 xml 文件的层次示意图

本文用于车辆再识别实验的数据集如表所示。本文共标记了校园监控视频中 176 个车辆目标,共计 391 段视频序列,10559 张车辆图片,每辆车至少在两个监控相机下出现,最多在四个监控相机下出现。

表 7 训练集和测试集

	图片数量	身份数
训练集	6570	93
测试集	3989	83

目前大多数目标再识别(如行人)都是对一个场景下的若干数据进行训练,然后将训练好的模型对相同场景下的其他数据进行测试。例如对 C1、C2 两个相机下的数据,现阶段主要的做法是取其中的一半用于训练,另一半用于测试,这种做法的识别效果很好,但是该场景下训练的模型可能对于新的

场景数据(如 C3、C4 相机)就不适用,也就是说已有的模型迁移性可能较差。

鉴于此,本文采用 A 场景的车辆数据进行训练,采用 B 场景的车辆数据进行测试。具体而言,本文采用相机编号为 41、81、82 三个相机下的数据进行训练,相机编号为 42、94、61、66 四个相机下的数据进行测试,用于训练和测试的数据集属于不同的场景,它们之间没有重叠,这种做法更适用于实际情况,也更考验算法的性能。本文用于测试的有 83 个车辆目标,186 段视频序列,我们对测试集中的 3989 张车辆图片全部进行了查询,即将每一张图片均作为查询图片,而与该查询图片不在一个序列中的其他序列作为候选集。因此,对于每张查询图片(Probe),都会有 185 个候选视频序列(Gallery)。

2 相似度计算方法

目标再识别可以分为 single-shot 和 multi-shot 方法。基于 single-shot 的车辆再识别是指每辆车在每个监控相机中只有一张图片,查询时只能利用单张车辆图片信息,而基于 multi-shot 的车辆再识别是指每辆车在每个监控相机中有多张图片或者序列,查询时可以利用多张图片的特征或者序列信息。

在本文中,候选集中存在的是车辆跟踪序列,当输入一张待查询车辆图片时,需要匹配到相关的序列。在度量一张车辆图片和车辆跟踪序列的距离时,将图片和一个跟踪序列里面所有图片都进行相似度计算,并返回最大相似度作为该图片和序列的计算结果。因为在实际场景中,只要匹配到跟踪序列中的任意一张图片通过存储文件即可找到目标车辆,因此该计算模式是合理的,也更适用于实际情况。

查询图片和候选集中的一个车辆序列相似度计算公式如下:

$$\text{sim}(q, \text{Seq}) = \max_{i=1}^N \{d(q, \text{Seq}^i)\} \quad (2.21)$$

其中, q 表示查询车辆图片的特征, Seq 表示跟踪序列, N 表示该车辆序列的关键帧数, Seq^i 表示该序列的第 i 帧图片的特征, $d(q, \text{Seq}^i)$ 表示相似度度量函数。最后对结果 $\text{sim}(q, \text{Seq})$ 按照相似度进行排序,返回查询车辆序列。

图形象地展示了本文中车辆图片(Probe)和候选序列(Gallery)的计算过程。



图 15 图片和序列的相似度计算过程

进一步地,对于实际监控视频下的车辆再识别,当输入一张车辆图片进行查询时,由于之前对车

辆进行了跟踪，所以除了只利用输入的单张车辆图片的特征外，还可以利用该车辆图片所在的跟踪序列信息，即查询时利用整个序列特征。因为对于任何一张车辆图片，其所属的跟踪序列信息都是可见的。本文中的两个序列之间的相似度计算如下：

$$sim(Seq_1, Seq_2) = \frac{\sum_{i=1}^{N_1} \max_{j=1}^{N_2} \{d(Seq_1^i, Seq_2^j)\}}{N_1} \quad (0-22)$$

其中 Seq_1 为查询车辆图片所在的跟踪序列， Seq_2 为待匹配序列， N_1 是序列 Seq_1 的帧数， N_2 是序列 Seq_2 的帧数， Seq_1^i 是序列 Seq_1 第 i 帧车辆图片的特征， Seq_2^j 是待匹配序列 Seq_2 第 j 帧车辆图片的特征， $d(Seq_1^i, Seq_2^j)$ 表示相似度度量函数。

3 实验结果

本文将查询时只利用了单张输入车辆图片的特征记为“图片&序列”模式，将利用了输入车辆图片所在的序列特征记为“序列&序列”模式。

本实验用于测试的有 83 个车辆目标，185 段跟踪序列，本文对测试集中的 3989 张车辆图片全部进行了查询，即将每一张图片均作为查询图片，而与该查询图片不在一个序列中的其他序列作为候选集。因此，对于每张查询图片(Probe)，候选集(Gallery)中都有 185 个序列。

本书采用颜色特征描述符以及 AlexNet 特征进行车辆再识别实验。

- (1) 颜色特征
- (2) AlexNet
- (3) AlexNet 微调
- (4) Siamese

(5) 改进的 Siamese

本文将车辆匹配时只利用了单张输入车辆图片的特征记为“图片&序列”模式，将利用了输入车辆图片所在的序列特征记为“序列&序列”模式。下面是本文的车辆再识别实验结果。

(1) 图片&序列模式

“图片&序列”模式是指查询时只利用了输入车辆图片的特征。图是采用欧式距离作为相似度计算方法得到的车辆再识别 CMC 曲线，图是采用余弦距离作为相似度计算方法得到的车辆再识别 CMC 曲线。表展示了不同特征得到的 mAP、HIT@1、HIT@5 和 HIT@10 结果。

表 8 不同方法的对比结果

方法	HIT@1	HIT@5	HIT@10	mAP
颜色特征+欧式	4.76%	18.63%	31.74%	11.39%
颜色特征+余弦	4.76%	18.63%	31.74%	11.39%
AlexNet+欧式	5.44%	12.56%	21.96%	10.64%
AlexNet+余弦	5.72%	13.26%	23.06%	10.90%
AlexNet 微调+欧式	7.72%	31.69%	47.98%	18.10%
AlexNet 微调+余弦	9.25%	32.21%	49.19%	19.37%
Siamese+欧式	16.37%	43.95%	63.22%	25.79%
Siamese+余弦	12.33%	42.29%	60.07%	22.98%
改进的 Siamese+欧式	28.15%	59.71%	74.83%	38.72%
改进的 Siamese+余弦	28.78%	60.32%	74.66%	39.39%

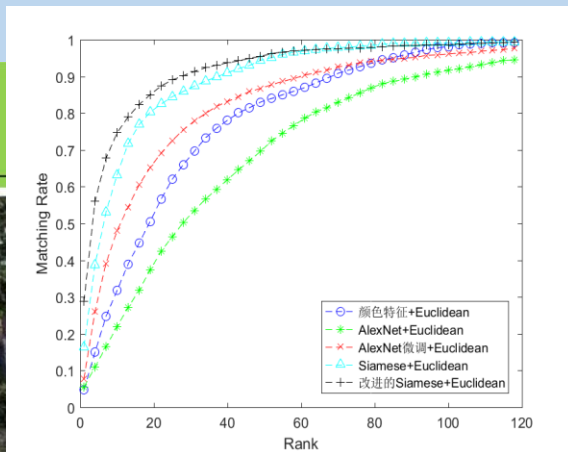


图 16 车辆再识别 CMC 曲线(欧式距离)

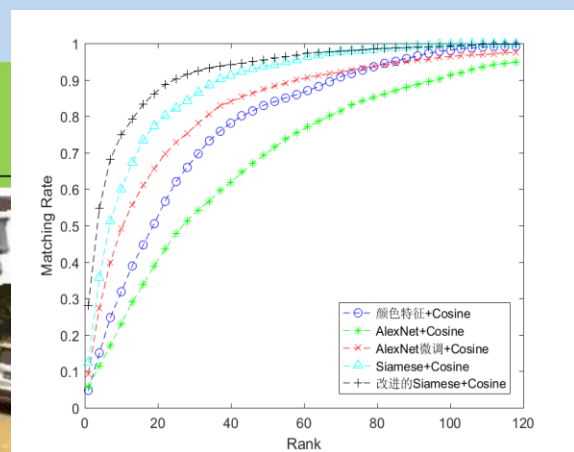


图 17 车辆再识别 CMC 曲线(余弦距离)

(2) 序列&序列模式

“序列&序列”模式是指查询时利用了输入车辆图片所在的序列特征，即采用多张车辆图片进行查询。在下面的实验中，采用改进的 Siamese 网络结构提取特征，并用余弦距离作为两个特征向量相似性的度量方法。车辆再识别结果如图所示，各项指标 mAP、HIT@1、HIT@5 和 HIT@10 如 9 所示。

表 9 图片&序列模式和序列&序列模式的对比结果

方法	HIT@1	HIT@5	HIT@10	mAP
图片&序列	28.78%	60.32%	74.66%	39.39%
序列&序列	30.13%	67.02%	81.38%	41.67%

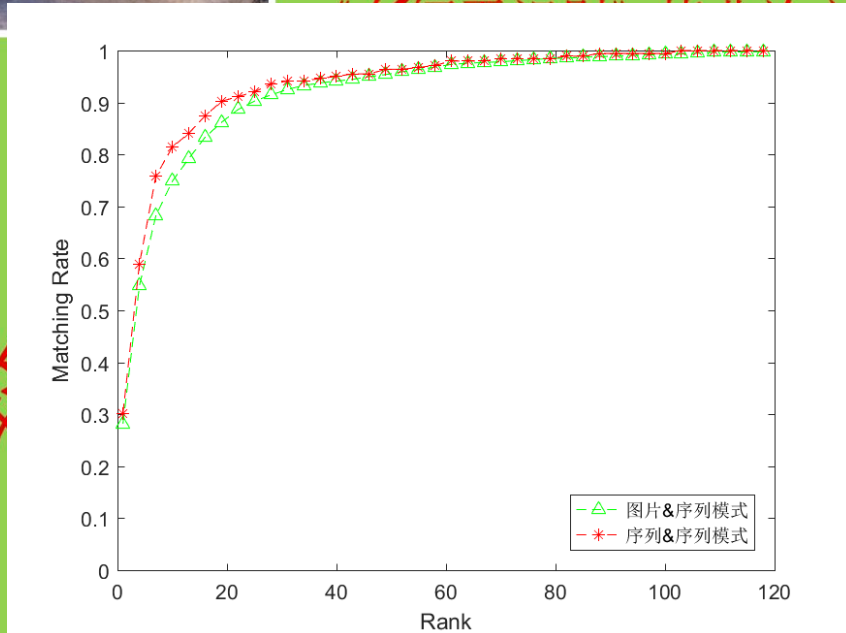


图 18 图片&序列模式和序列&序列模式的车辆再识别 CMC 曲线

如图是采用改进的 Siamese 网络结构用于车辆查询的一些结果示例。蓝色矩形框表示查询车辆，返回列表中正确的匹配结果由绿色矩形框标出，错误的匹配结果由红色矩形框注。Query 表示“图片&序列”模式，即只利用查询图片的特征，Multi.Queries 表示“序列&序列”模式，即利用了查询图片所在跟踪序列的特征信息，使用多张车辆图片进行查询。从图中可见，在查询时，利用同一辆车的多

