行为检测与识别方法研究

华俊豪†

2014 年 4 月 15 日

1 引言

春学期修了一门《图像与视频分析》课程，很随机地选了大作业，叫" 行为检测与识别"。看了一些论文，通过作者提供的代码大致了解了下这个领域的基本内容，然后复现并改善了一些算法。浙大分春夏秋冬四个学期，每个学期共八周课，而我这春季学期需要修五门课，加上自己的研究项目，总的来看，时间是相当紧凑的。虽然如此，在这个充满生机与朝气的四月天里，我依旧能感受到懒散的味道。

2 动作识别的研究现状

这部分留给我的另一位组员写了，这里就不贴了。

3 动作检测与识别实验

3.1 概要

行为检测与识别可通过多种方法，本文是基于时空兴趣点的。此处不对文献进行综述，仅讨论本文所涉及到的相关论文。Dollár[6] 通过提取时空特征点，进而形成动作的特征描述子，通过对特征描述子的学习与分类，对视频中行为或动作进行检测与识别。然而，该方法只能处理单动作的视频序列。Niebles [9] 提出一个无监督的视频动作学习方法，其同样采用上诉方法提取时空特征点，但并不建立单个动作的特征描述子，而是对兴趣点聚类，构成类似于文本分析中的词包模型 (bag of words)。每一个类对于一个词 (word), 每一个动作类型对应一个主题 (topic)，每一个视频序列就是一个文档 (document)。用图模型表示视频的生成模型，主要包括概率潜在语义分析 (pLSA) 和潜在狄利克雷分配 (LDA)。对 pLSA 或 LDA 概率推断，得到每个 word 在不同 document 里属于某个 topic 的概率。采用这种无监督学习算法能够对视频中的多个动作进行分类和定位。

本文的工作正在建立在上述 [6, 9] 两篇论文的基础上。首先复现了 Dollár 的工作；然后用自己写的 LSSVM[12] 与林智仁的 libsvm[4] 在不同的核下跑识别结果；最后，借鉴 [9] 的思想提出一种简化的多动作视频的检测识别算法。其基本思路是在提取时空特征点并后建立其特征向量后，对其聚类建立词库 (codebook), 再分配每个词属于某个动作类别的概率，确定测试样本中的兴趣点属于哪个词，由此确定所属类别。通过每一帧图像的兴趣点及其所属类别进行动作的分类与定位。

3.2 算法描述

对一堆输入的训练样本先检测并提取特征，再将特征转化为动作描述子，用分类器对动作描述子学习得到相应模型。对于测试数据，采用与训练样本相同的方法提取特征，包括相同的参数设置于基空间，得到动作描述子后，用分类器识别分类。如图 (1) 所示。

†Junhao Hua, Department of Information Science and Electronic Engineering, Zhejiang University, huajh7@gmail.com

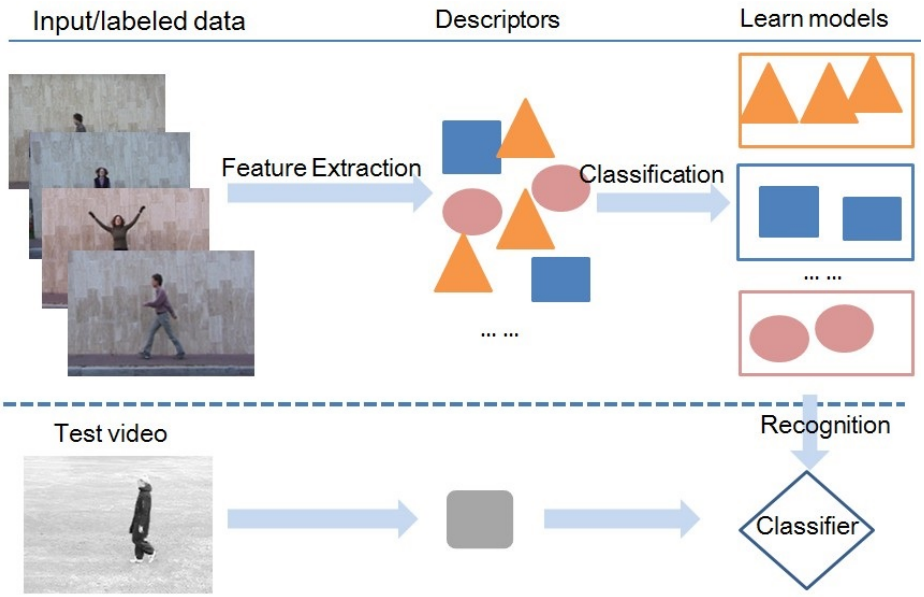

图 1: 动作检测与识别的基本框架

3.2.1 特征检测与提取

图像兴趣点检测最广为使用方法之一是角点检测，比如 Harris 角点，直观讲就是要求在水平和竖直方向上变化都比较大。通过对高斯平滑的图像 $L(x,y,\sigma) = I(x,y) * g(x,y,\sigma)$ 求一阶梯度，由协方差矩阵的特征值确定每个点的响应强度，从而提取出兴趣点。另一种常用方法是用 Laplacian of Gaussian (LoG) 构造响应函数，比如 Lowe [8] $D = L(x,y,;k\sigma) - L(x,y;\sigma)$. 然而，这些方法都是在空间维度上的 (因为处理对象是图像)，那么针对一个视频序列 $I(x,y,t)$，就需要将其扩展到时空维上。

按 [6]，响应函数定义为

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{1}$$

其中 $g(x,y;\sigma)$ 为作用在空间的二维高斯滤波核，h_{ev} 和 h_{od} 为作用在时间上的一维 Gabor 滤波，

$$h_{od}(t;\tau,\omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2}$$
$$h_{ev}(t;\tau,\omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2}$$

其中采用 $\omega = 4/\tau$, σ 和 τ 分别表示空间和时间上的尺度。

然后用非极大值抑制 (Non-max suppression) 方法搜索局部极大值，即判断该点是否为其时空窗 (大小为 $(x,y,t) = (2\lceil 3\sigma\rceil + 1, 2\lceil 3\sigma\rceil + 1, 2\lceil 3\tau\rceil + 1)$) 内的最大值且满足一定的阈值条件。按照这种方法可以检测到相当多的极值点，但太多特征点反而会使结果变坏，因此控制兴趣点个数是很有必要的，可以只取最大或最小程度最大的前几百个。如图 (2) 所示为每一帧中的可视化的兴趣点。

通过以上步骤得到兴趣点，将包含兴趣点的时空窗定义为 cuboids. 如图 (3) 所示为每一帧中的可视化的 cuboids。对于一个长方体（cuboids), 数据量是比较大的，且直接定义相似度函数来用于比较是不合适的。因此更进一步地，需要创建一个 cuboids 描述子。转换方法是计算 cuboids 中每个点的亮度梯度 (brightness gradient), 为了得到更丰富的特征，应先在 cuboid 上加以不同尺度的高斯滤波 (实验用了 2 不同尺度)。这里得到的依然是一个三维向量，要转化为一维特征向量，可直接将其拉直，或者采用局部直方图（local histogram)，这里采用前者。

得到的特征向量维度是相当高的 ($x \times y \times t \times 3 \times 2$)，采用 PCA 降维。显然，需要保持每个 cuboid 主成分的一致性，否则无法比较。因此 PCA 降维是这样做的：所有训练样本提取出 cuboids 后，随机在其中取一定数目的 cuboids 特征向量，然后 PCA 降维，取特征值最大的前 K 个主成分, 如图 (4)。实际上就是提取了 k 个基 basis)。然后将每个 cuboids 投影到这些基上，构成了 $K \times 1$ 的 cuboid 特征描述子。这种描述子可看成了 PCA-SIFT 描述子 [7] 的推广。

图 2: 一个 walk 视频的兴趣点可视化 ($\sigma = 1.5, \tau = 1.5$)

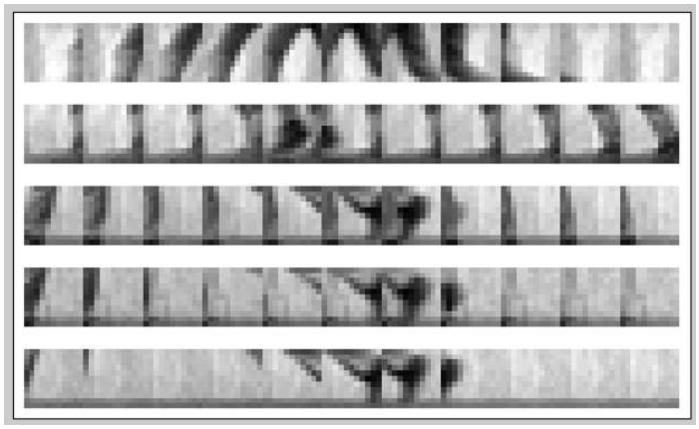

图 3: 一个 walk 视频中极值程度最强的 5 个 cuboids ($\sigma = 1.5, \tau = 1.5$)

3.2.2 特征描述

Cuboid Prototypes 虽然每个动作可能千差万别，但可发现许多兴趣点是具有相似性的。由此可见，虽然可能的 cuboid 有很多，但不同的 cuboid 类型数却不多。于是可通过对训练样本的 cuboid 描述子聚类得到 cuboid prototypes 字典库。这里聚类采用简单的 k-means 算法，在实验中聚类个数设为 50 个。

Behavior Descriptor 有了 Cuboid 类型库，再提取出动作描述子作为一个视频序列的特征向量。这个采用的方法是用 cuboid 类型的直方图表示。一个视频的兴趣点对于的 cuboids 属于某个 cuboid prototype 的个数构成的向量。图 (5) 举了 4 个例子。直方图的比较通常用卡方距离 (χ^2) 来表示相似度，定义为，

$$d_{\chi^2}(x, y) = \sum_i \frac{(x_i - y_i)^2}{2(x_i + y_i)} \tag{2}$$

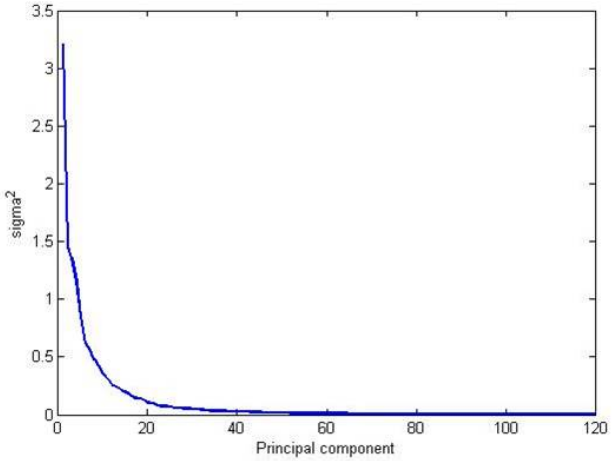

图 4: Weizmann 人类动作数据集 (见下文) 降维结果 ($\sigma = 1.2, \tau = 1.2$)

3.2.3 动作分类算法

将视频数据转为相应的动作描述子后，便可以通过对训练样本的描述子得到分类器，然而用分类器对测试数据进行分类。可以采用最简单的分类器为 KNN(K-nearest neighbors)，也可以采用相对复杂一点的支持向量机 (SVM) 分类器。

KNN KNN 是一种无参分类器，对于一个测试样本，在训练样本的特征空间中搜索与之最近的 k 个样本点，这 k 个样本中属于某一个最多，那么就认为该测试数据属于哪一类。实验采用 1NN. 这里对距离的度量便采用的是卡方距离。

SVM 就两类分类问题而言，简单得讲，支持向量机就是用两个 $N-1$ 维的超平面分割 N 维的线性空间，使得在两个超平面两侧的半闭空间分别属于两类。SVM 优化求解就是使两个分隔面之间的距离最大。当然，引入松弛变量后，上一句就不那么严格了。采用 Lagrange 乘子法将其转化为求解它的对偶问题，该对偶问题是一个二次规划问题，直接求解比较复杂，可以采用 Chunking 简化 (Vanik,1982;Burges,1998 [3])，但更好更广为使用的 SMO 算法 (Sequential minimal optimization,Platt,1999 [10])。此外，还有一种 Least square SVM [12], 将目标函数中的松弛变量改成二次，使得最终的问题变成了一个线性方程求解，只需要几步矩阵运算。由于其为 2 范式, 最后的 Lagrange 乘子不再稀疏，使得所有的训练样本都为支持向量。当然, 作为改进的 Sparse LSSVM。此外，考虑到原特征空间非线性，使用 kernel 方法，将低维的非线性空间映射到高维的 (线性) 特征空间。这里对直方图的比较，[5] 提出采用所谓的卡方核 (chi-squared kernel) 相对于高斯 RBF 具有更好的效果。对于一个多分类器，可以采用多种策略，本实验用最基本最简单的 one-versus-the-rest 方法。关于 SVM 的详细介绍见 PRML 的第 6,7 章节 [1]。简单起见，实验中仅实现了最基本的 LSSVM 算法以及使用 SMO 优化的 libsvm[4] 作为分类器。

3.2.4 多动作识别框架:Voting

以上算法框架，将整一个视频序列序列做个分类对象，因此单视频中包含多个动作时，便无法处理。针对此类情况，我们根据 [9] 的词包模型，提出简化版的多动作识别框架。

训练样本依然用单动作的视频，视频的时空特征提取与以上完全一致，但不再通过聚类得到数量较少的 cuboid prototypes, 而是聚类得多数目较多的时空词 (spatial-temporal words)。其实除了聚类数量级不同外 (比如前者 50，后者 1000)，聚类方法一模一样, 得到的类中心点，称之为 CodeBook。训练样本的每一个 cuboid 属于某一个 word，而 cuboid 所在的视频动作的标签（即属于哪个动作）是已知的。这样，通过每个 word 中的 cuboid 标签比例，计算 word 属于某个 topic(动作) 的概率 $P(z_k|w_i)$, 取概率最大的 topic 为 word 的标签。该模型类似于主题模型，如 pLSA(如图 6) 和 LDA.

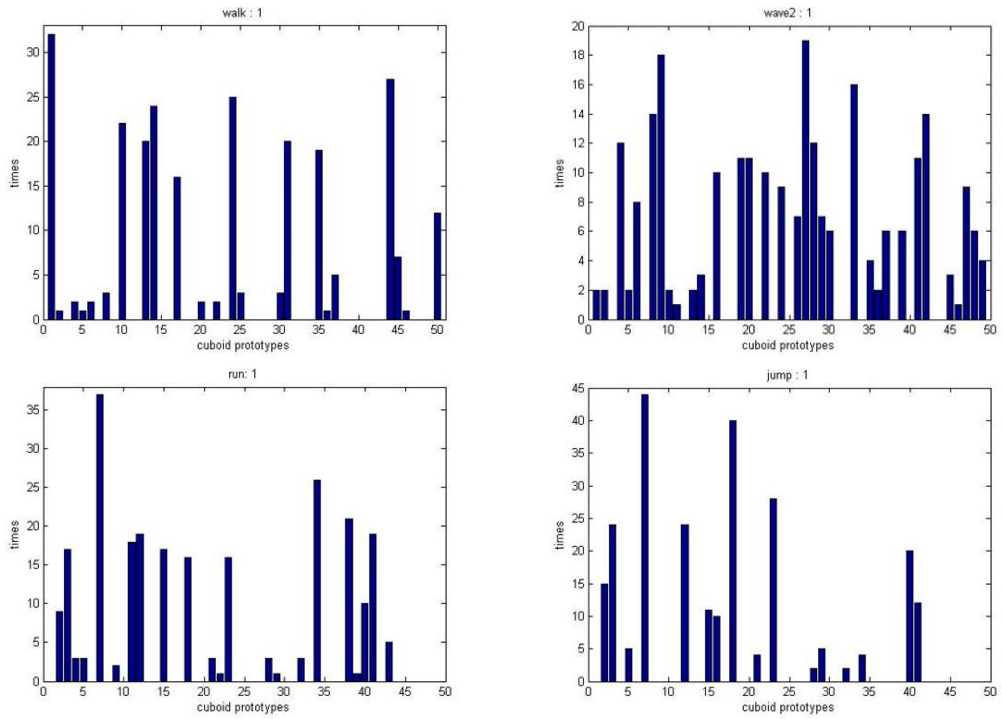

图 5: 左上：walk；左下：Run; 右上:wave2; 右下：jump.(Weizmann 人类动作数据集)

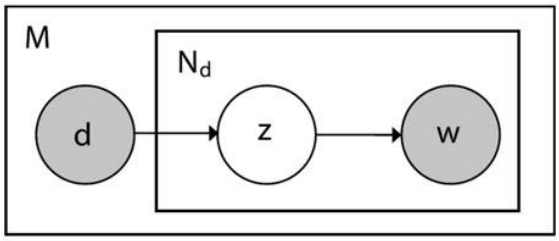

图 6: pLSA 图模型

对一个具有多动作的测试视频，将检测得到的兴趣点分配给某个 word，从而分配其相应的 topic. 然后对每一帧图像进一步动作分类和定位。首先确定该帧图像中有多少比较显著的动作的个数，判断方法是 topic 占用比例，以及对于的特征点个数，其值是需要微调的，一种可用的方案如图 (7)。然后在对兴趣点 k-means 聚类，得到各类所在的时空位置，而该类属于哪里 topic, 由盖类中的兴趣点投票确定。这种方法这本文中取名为：**Voting**. 多动作检测与视频框架如图 (8) 所示。

3.3 实验结果

3.3.1 实验一:Toy SVM 分类

第一个实验做了简单的 SVM 分类实验，测试了多类的 LSSVM 分类结果。如图 (9) 所示生成数据为各个圆环上的点，加圆圈的为测试数据. 使用径向基核函数，最终三类分类精度为 0.98, 四类分类精度为 0.94.

3.3.2 实验二：Weizmann human action dataset

第二个实验使用 Weizmann 人体动作数据集 [2]，总过包含 90 个低分辨率 (180×144,50fps) 视频，含有 10 种不同的动作（bend, jack, jump, pjump, run, side, skip, walk, wave1,wave2, 如图

```
% how many action categories ?
categ_idx = [];
k = 0;
for j=1:ntype
    if type_prob(j)>0.5 ||...
        (type_prob(j)>0.4 && type_cnts(j) > 5) ||...
        (type_prob(j)>0.3 && type_cnts(j) > 14) ||...
        (type_prob(j)>0.2 && type_cnts(j) > 13) ||...
        (type_prob(j)>0.1 && type_cnts(j) > 15)
        categ_idx = [categ_idx,j];
        k = k + 1;
    end
end
```

图 7: 部分代码截图：判断一帧图像有多个类

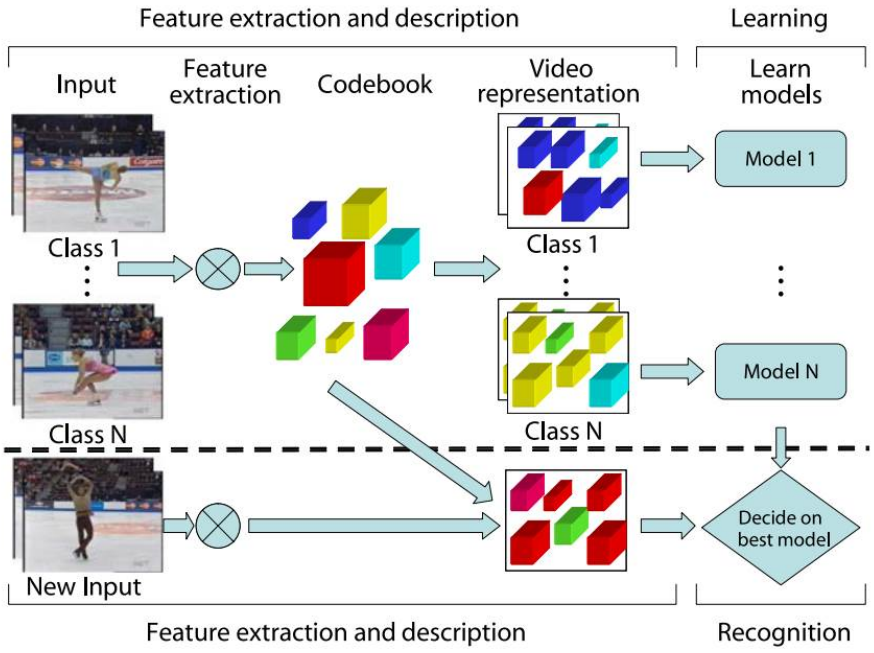

图 8: 多动作检测与识别的基本框架，图来自于 [9]

(10))，分别由 9 个人完成。

用以上所述的方法检测和提取时空特征点，其参数设定为 $\sigma = 1.2, \tau = 1.2$, 兴趣点个数最大值为 200,PCA 降到 100 维。由于数据样本数不多，采用 Leave-one-out 交叉验证，将 90 个数据份分成 9 组，每组 9 个数据，每次取 9 组训练另 1 组测试。重复计算 5 遍，取平均值。这个包含三种分类方法：

- 1NN 用卡方距离度量特征向量的相似度，最终分类准确率为:**0.7667** 如图 (11). 抽取了各个动作的正确分类结果，见图 (14).
- LSSVM 用 LSSVM 且分别采用线性核 (linear)，多项式核 (Polynomial)，径向基 (RBF) 和卡方核 (Chi-squared) 分类. 结果表明，采用径向基核（高斯或 χ^2) 的方法并没有得到理想的效果, 见图 (12)。在实验过程中，由于首先考虑的是 χ^2 核，且一直达不到好的效果，本以为是由于数据量不够的原因。因此又做了具有更大数据量的数据集实验，即实验三。但后面会发现，并非是由于数据量的原因。
- Voting 采用投票的方法，得到的总精度为 **0.7333**, 主要的错误是将 wave1 识别为 wave2, 这是因为两个具有局部一致性。这也暴露了该方法的缺点，没有利用时空特征的相对位置信息。

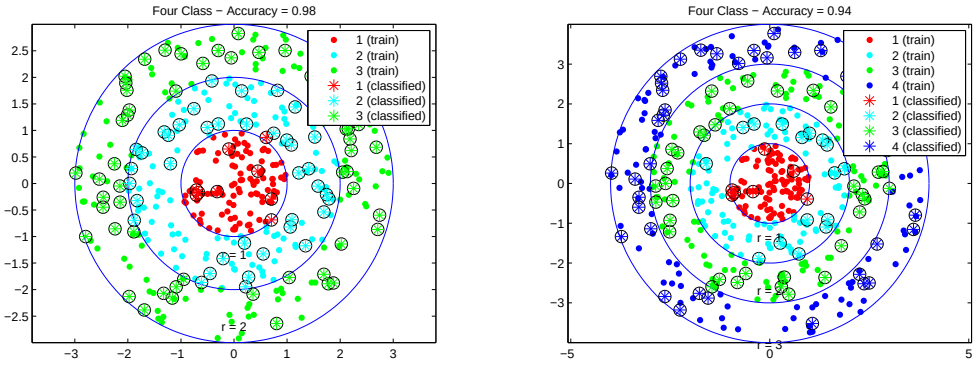

图 9: 左：3-class; 右:4-class

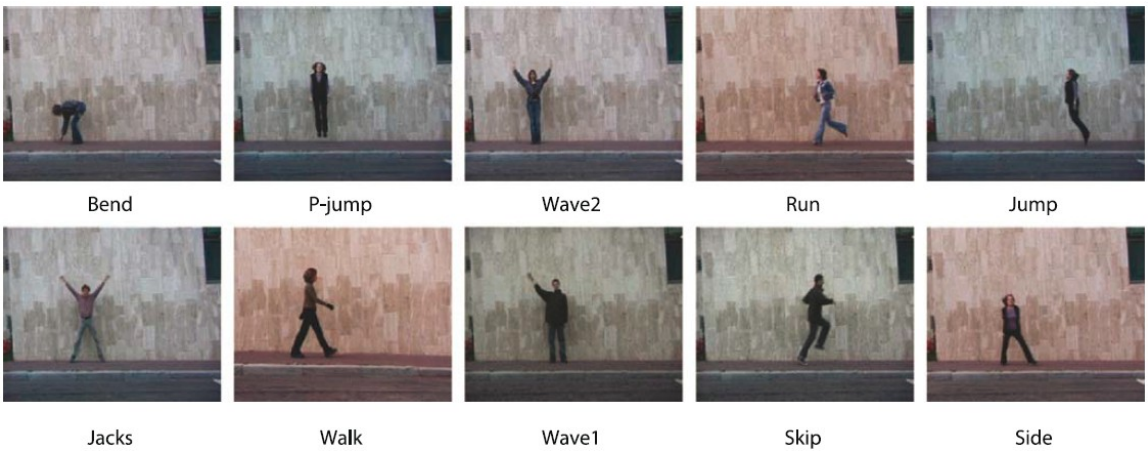

图 10: weizmann 视频数据例子

3.3.3 实验三：KTH 数据集

上文已经提到 weizmann 数据量比较小，可能导致 SVM 训练不充分。这里使用了 KTH 数据集 [11], 包含 6 类人体动作 (walking, jogging, running, boxing, handwaving 和 handclapping), 由 25 个不同的人在 4 个不同场景下 (室内, 室外, 尺度变化)，总共 598 个 160×140 的视频序列（两个视频丢失），如图 (15)。

用以上所述的方法检测和提取时空特征点，其参数设定为 $\sigma = 1.5, \tau = 1.5$, 兴趣点个数最大值为 300,PCA 降到 200 维。由于数据样本数不多，采用 Leave-one-out 交叉验证，将 598 个数据份分成 25 组，每组 24 个数据，每次取 24 组训练另 1 组测试。重复计算 5 遍，取平均值, 用 1NN,LSSVM 和 Voting 共 4 种方法得到分类结果见图 (16)(17)(18)。由于数据集的数据量更多，且环境比较简单，得到的分类精度都有所提高，并且在 SVM 的使用中，基于 RBF 的高斯核与卡方核都没有得到有的效果，实际上，我们用开源的 libsvm 同时训练一遍结果与用我们的 LSSVM 一样，因此可排除代码错误，至于为什么没有好的结果，目前还找到不好的解释。另外，采用 Voting 方法，由于各个动作之间没有了局部一致性，因此达到最佳分类精度:**0.9047**。

3.3.4 实验四：多动作数据测试

最后，我们测试了多动作视频的检测与识别，由于没有合适的数据集提供测试，我们自己拍了几段视频，如图 (19). 用 weizmann 训练的识别结果部分截图见图 (21)，用 KTH 训练的识别结果部分截图见图 (22)。采用不同的训练集需要采用其相对于的参数配置，比如如果用 weizmann 数据集则在提取测试数据的兴趣点时，设置 $\sigma = 1.2, \tau = 1.2, kpca = 100$ 而用 KTH 数据集才设置

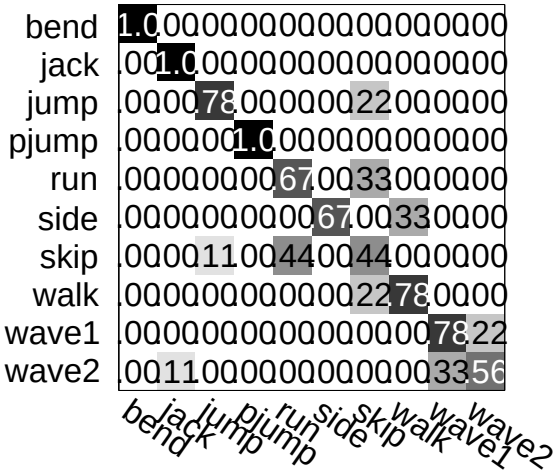

图 11: weizmann dataset by 1NN, 总分割精度为 0.7667

$\sigma = 1.5, \tau = 1.5, kpca = 200$。否则其 cuboid 描述子无法对齐。也正是因为这个原因，使得同一个视频数据提取出的兴趣点有差异，显然由于拍摄的原因，具有稍大的尺度 (σ, τ) 更好的提取视频中原地站立且离摄像机较远的人的特征点。由于视频中人物的动作是根据 weizmann 数据集定义的动作制作的，因此在用 KTH 训练分类的结果中的部分分类错误也在情理之中。更多信息详见输出视频。

4 总结

所有的测试输出结果实际为一段视频，文档不可能全部展现，详见相关文件夹的输出视频。

作为非计算机视觉为研究对象的研究生，通过对" 动作检测与识别 "这个专题内容的学习, 研究与程序实现，不仅很好的窥测到该领域的惊鸿掠影，而且在程序实现的级别上对图像与视频处理中的各个细节有了一个基本认识和基本技能，而不是停留在对一个方程或一个算法流程的宏观理解上。实验代码将会放在我的 github 主页 (**https://github.com/huajh**) 上，如有疑问或发现一些错误，可电子邮箱联系我。

References

[1] Christopher M Bishop et al. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[2] Moshe Blank, Lena Gorelick, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1395–1402. IEEE, 2005.

[3] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[4] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[5] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *Neural Networks, IEEE Transactions on*, 10(5):1055–1064, 1999.

[6] Piotr Dollár, Vincent Rabaud, Garrison Cottrell, and Serge Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72. IEEE, 2005.

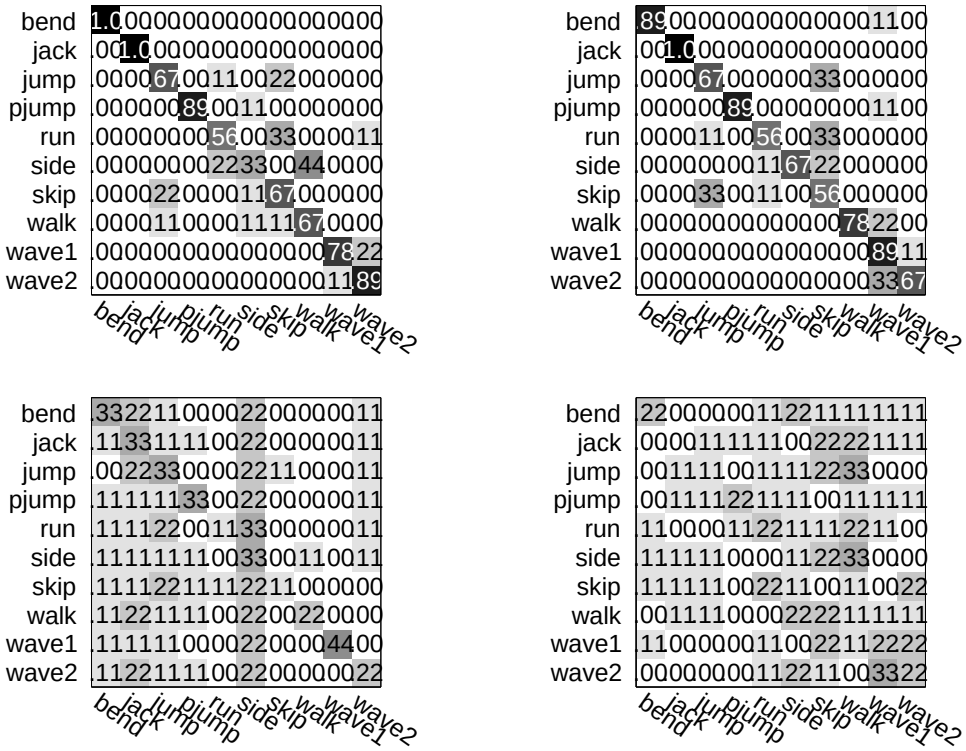

图 12: weizmann dataset bySVM. 左上：linear, 分割精度 0.7444；左下：rbf,0.2778; 右上:Polynomial,0.7556; 右下：Chi-squared,0.1444

[7] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506. IEEE, 2004.

[8] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[9] Juan Carlos Niebles, Hongcheng Wang, and Li Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International journal of computer vision*, 79(3):299–318, 2008.

[10] John Platt et al. Sequential minimal optimization: A fast algorithm for training support vector machines. 1998.

[11] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.

[12] Johan AK Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, Joos Vandewalle, JAK Suykens, and T Van Gestel. *Least squares support vector machines*, volume 4. World Scientific, 2002.

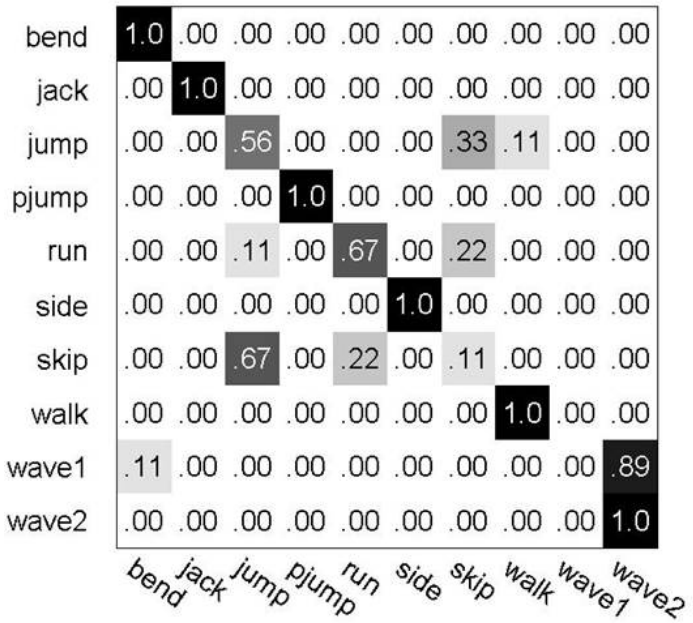

图 13: weizmann dataset by Voting, 总分割精度为 0.7333

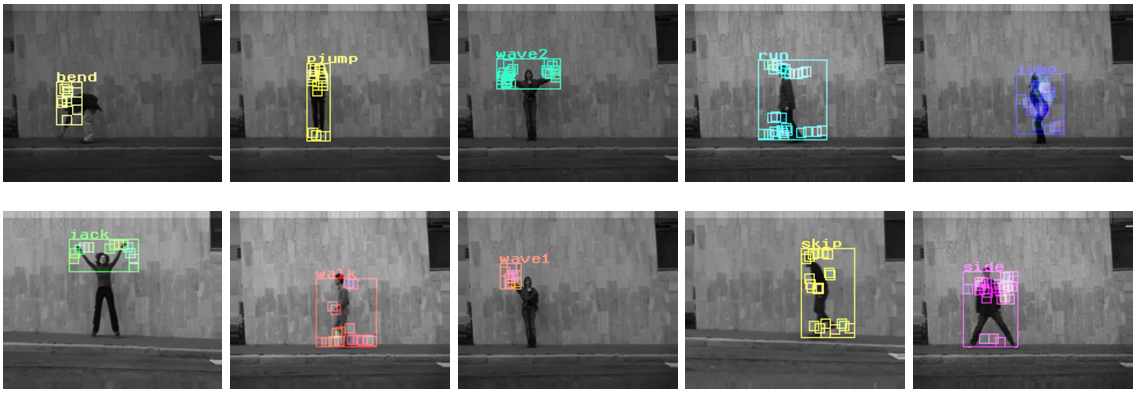

图 14: 分类结果截图 by 1NN

图 15: KTH 人体动作数据集

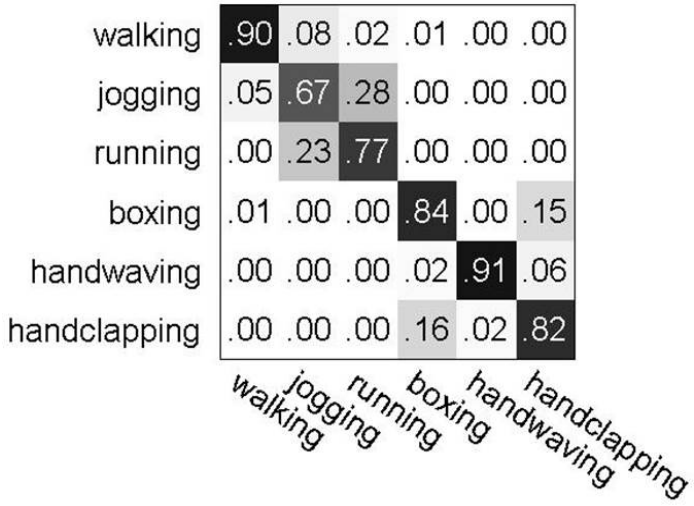

图 16: KTH dataset by 1NN, 总分割精度为 0.8179

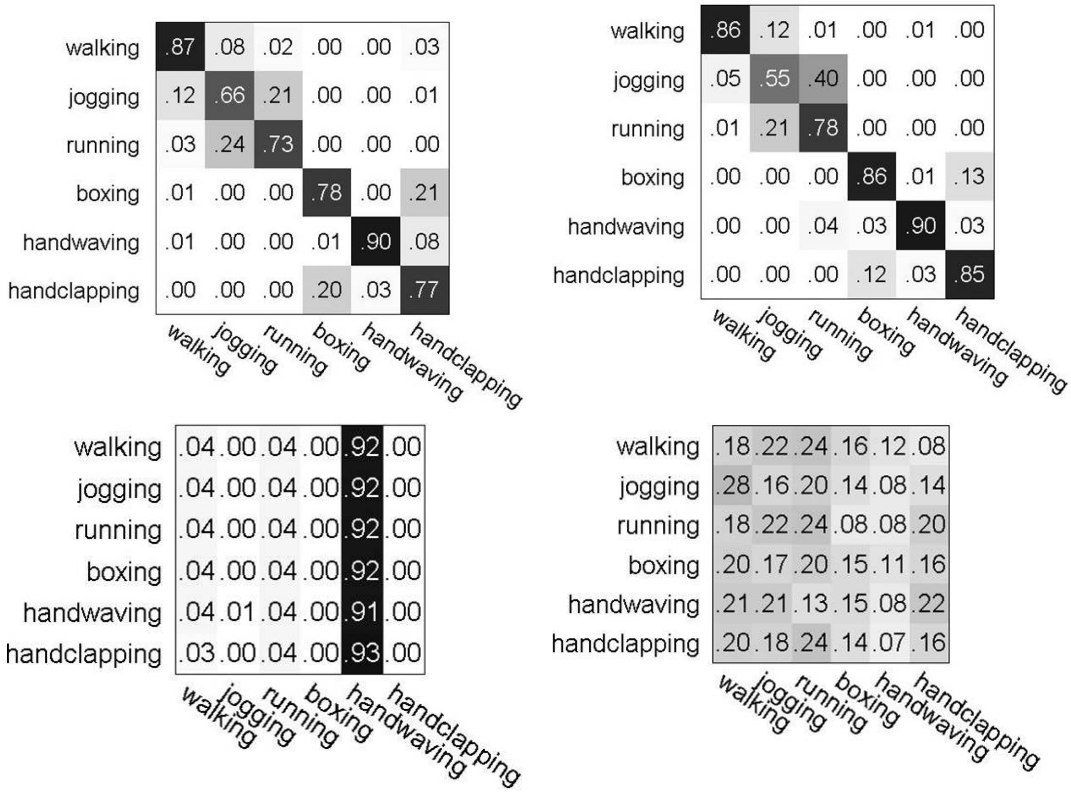

图 17: KTH dataset by LSSVM. 左上：linear, 分割精度 0.7844；左下：rbf,0.1656; 右上:Polynomial,0.7996; 右下：Chi-squared,0.1624

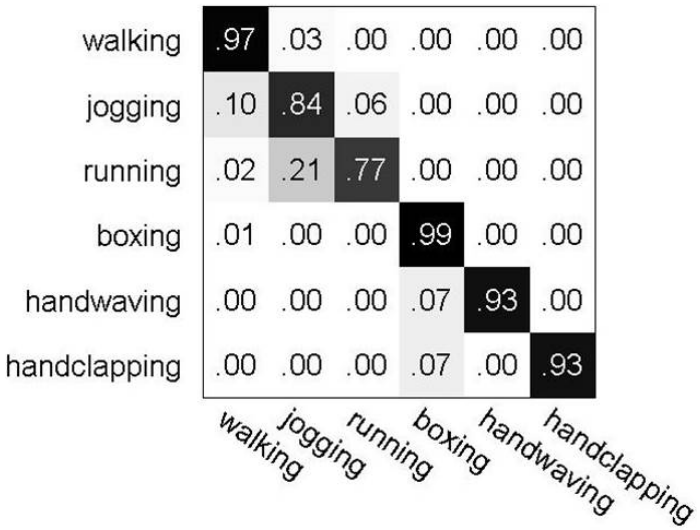

图 18: KTH dataset by Voting, 总分割精度为 0.9047

图 19: 自制的多动作视频截图

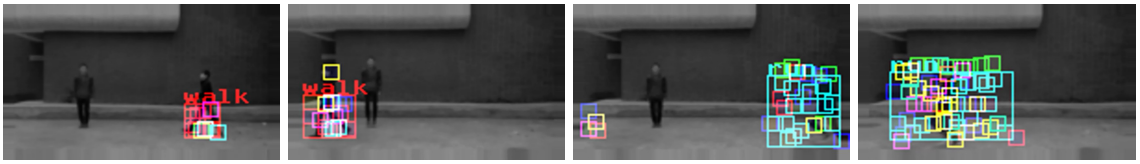

图 21: 一个视频在 weizmann 数据集上测试结果截图

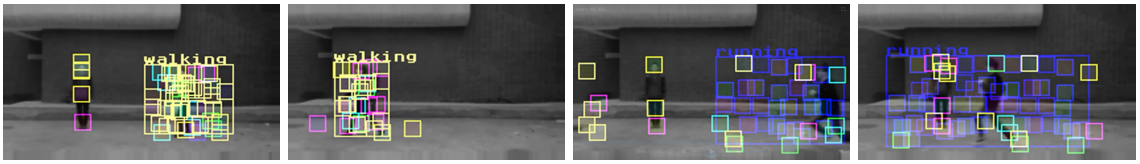

图 22: 一个视频在 KTH 数据集上测试结果截图