中国科学: 信息科学 2024 年 第 54 卷 第 4 期: 893–910
SCIENTIA SINICA Informationis
论文

面向连续手语识别的自适应关键帧选择

闵越聪[1,2], 陈熙霖[1,2*]

1. 中国科学院计算技术研究所智能信息处理重点实验室, 北京 100190
2. 中国科学院大学计算机科学与技术学院, 北京 100049
* 通信作者. E-mail: xlchen@ict.ac.cn

收稿日期: 2022–12–28; 修回日期: 2023–05–24; 接受日期: 2023–09–16; 网络出版日期: 2024–04–11

新一代人工智能国家科技重大专项 (批准号: 2021ZD0111900) 资助项目

摘要 基于视觉的连续手语识别旨在从图像序列中识别出对应的手语词序列, 可以为手语使用者提供一种便利的辅助工具. 现有的连续手语识别方法大多需要从图像序列中, 逐帧提取视觉和时序特征, 而相邻帧中存在的相似视觉信息带来了大量的冗余计算. 本文通过分析帧率对连续手语识别算法的影响, 发现降低帧率可以显著地提升计算效率, 但也会带来一定的性能损失. 为了在降低帧率的同时保留更多手语关键信息, 本文提出了自适应动态池化层 (adaptive dynamic temporal pooling, ADTP), ADTP 基于序列特征的自相似性对序列进行动态下采样. 在此基础上, 本文进一步提出了一种两阶段的训练方式, 以更充分地利用原始帧率中的时空信息. 具体而言, 该训练方式在第一阶段只训练基于原始帧率的手语识别模型, 并以此模型为教师网络, 通过知识蒸馏的方式引导第二阶段含 ADTP 模块的模型训练. 实验结果表明, 本文所提的方法在损失少量性能的情况下, 可以大幅度减少识别所需的计算量. 此外, 本文所提出的 ADTP 也可用于手语视频结构分析, 生成简略直观的手语视频摘要.

关键词 连续手语识别, 时间序列分析, 视觉语言, 知识蒸馏, 计算效率

1 引言

根据世界卫生组织的统计数据 [1], 全世界大约有 5% 的人口 (约 4.3 亿人) 患有残疾性听力损失 (听力阈值大于 35 分贝) 并预测这个比例在 2050 年会增加到 10%. 作为一种被听力、语言功能障碍群体广泛使用的自然语言, 手语通过手势、肢体动作、面部表情等视觉元素传递语言信息. 然而, 并不存在一种全世界通用的手语, 同一国家的不同地区也可能使用不同的手语方言, 这为手语使用者的日常沟通交流带来了很大的困难. 手语通过肢体动作和面部表情等视觉因素传递信息, 而基于视觉的手语识别技术则可以通过非介入式的技术手段自动地从手语视频流中识别出所表达的手语词, 为手语使用者的交流提供一种便利的辅助工具.

引用格式: 闵越聪, 陈熙霖. 面向连续手语识别的自适应关键帧选择. 中国科学: 信息科学, 2024, 54: 893–910, doi: 10.1360/SSI-2022-0467
Min Y C, Chen X L. Adaptive keyframe selection for continuous sign language recognition (in Chinese). Sci Sin Inform, 2024, 54: 893–910, doi: 10.1360/SSI-2022-0467

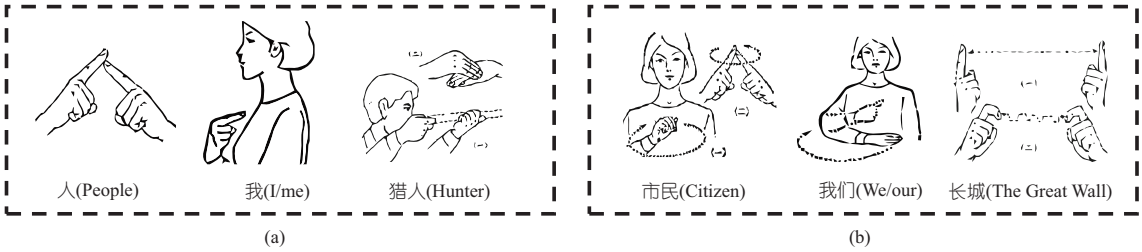

图 1　中国手语词示意图. (a) 姿态/手型主导的手语词; (b) 轨迹主导的手语词

Figure 1　Examples of Chinese sign language words. (a) Posture dominant signs; (b) trajectory dominant signs

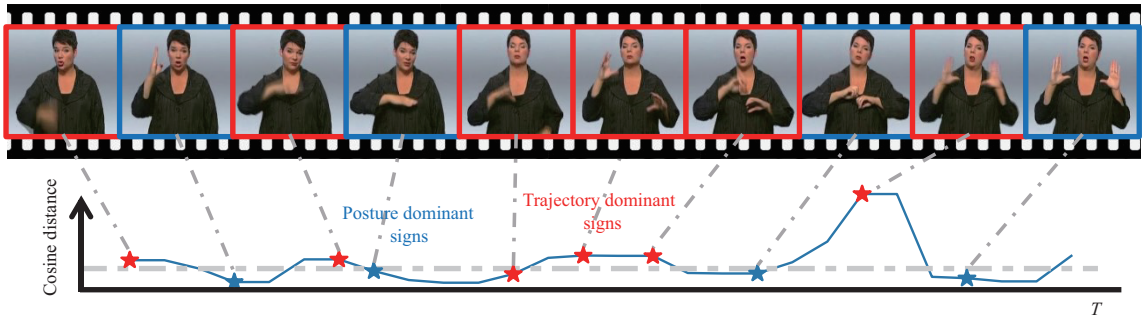

图 2　(网络版彩图) 手语视频相邻帧的余弦距离可视化. 余弦距离小的帧和手型更相关, 余弦距离大的帧和轨迹更相关

Figure 2　(Color online) Cosine distance visualization of adjacent frames in sign language videos. Frames with small cosine distances are more relevant to posture dominant signs and frames with large cosine distances are more relevant to trajectory dominant signs

随着深度学习的兴起, 数据驱动的连续手语识别方法 [2~5] 在主流数据集上达到了很好的性能. 受限于计算资源和数据的数量, 现有的连续手语识别方法, 往往采用二维卷积神经网络逐帧提取视觉特征, 再通过 CNN [6]/RNN [3, 5]/Transformer [7, 8] 等方法提取时序特征进行识别. 依次提取视觉特征和时序特征的方式可以减少计算资源的需求量, 并且降低了对数据集的过拟合程度, 实现更好的泛化性能. 但是这种特征提取方式仍有很多缺点: 由于相邻帧视觉内容相似, 视频中存在大量的冗余信息, 而逐帧进行视觉特征提取也随之带来了较多的冗余计算; 时序特征的提取依赖于高度抽象的视觉特征, 对细粒度的运动建模能力不足. 作为一种视觉语言, 手语包含很多的细粒度的手部动作和面部表情, 如何高效地建模手语中的动态信息是一个值得研究的问题.

人类的视觉系统每秒能够独立地处理 10 ~ 12 帧的图像, 更高频率的图像信息则以运动信息的形式进行处理 [9]. 而作为一种视觉语言, 手语的演示速度也需要符合视觉系统的处理速度. 一些早期的手语研究 [10~12] 发现 6 帧每秒 (frame per second, FPS) 是可接受的手语演示速度, 手势拼写等更高密度的信息传递则需要更高的帧率. 如图 1 所示, 手语通过静态 (如某一特定的姿态和手型) 和动态 (手部运动的方向和速度) 两种方式传递相关信息 [13], 降低采样的帧率虽然可以增加手语识别的效率, 同时也增加了丢失关键手型的风险, 并带来一定运动信息的损失. 此外, 类似于口语中的语音和语调, 基于语言学的研究工作 [14] 发现, 手语中也存在着韵律系统, 其韵律成分的层次结构在一定程度上对应于形态句法成分, 但目前鲜有工作结合韵律成分对手语进行分析.

为了分析手语中的层次结构信息, 进而实现更高效的手语识别算法, 本文基于视觉特征之间的相似关系定位手语关键帧. 之前的工作揭示了余弦距离可以反映视觉序列特征之间的相似关系 [15], 图 2

可视化了手语视频相邻帧之间的余弦距离, 可以看到相邻帧视觉特征的余弦距离在一定程度上可以反映手语的节奏信息, 且静态手型和动态轨迹之间的过渡往往具有较大的视觉差异. 基于此现象, 本文提出了一种自适应动态时序池化 (adaptive dynamic temporal pooling, ADTP) 方法. 该方法利用浅层的抽象视觉特征, 基于预先设定的阈值或比例对连续手语序列进行自适应分段, 进而实现关键帧的选择和分段时序信息的池化. 相较于常见的最大池化和平均池化, ADTP 可以自适应地对序列进行切分, 减少了因不合理切分所带来的信息损失. 视觉特征的提取是连续手语识别算法主要的计算瓶颈. 基于所提出的 ADTP 方法, 本文在特征提取的早期引入时空卷积模块以捕捉细粒度的动态信息, 并进一步设计了一种两阶段的训练方法: 在第一阶段, 不进行时序降采样, 并将训练好的模型作为教师网络引导第二阶段的训练. 而在第二阶段, 在浅层模块进行时序下采样并基于浅层特征的帧间相似度重建原始的深度特征序列, 以恢复原始视频序列中的节奏信息. 此外, 在第二阶段的训练过程中, 通过知识蒸馏的方式, 利用第一阶段训练好的模型引导模型捕捉更多的时序信息, 进而增强泛化性. 相较于常用的逐帧的视觉特征提取网络, 本文所提出的方法在减少冗余计算的同时, 保留了更多的视觉和节奏信息.

为了说明方法的有效性, 本文在公开的德国手语数据集 PHOENIX14 [16], CSL-Daily [17] 和自采的中国手语数据集上进行了验证. 实验结果表明, 在视觉特征提取阶段使用所提出的 ADTP 可以在损失少量性能的情况下, 大幅度减少手语识别所需的计算量. 此外, 本文所提出的 ADTP 也可用于手语识别结构分析, 生成简略直观的手语视频摘要.

2 相关研究现状

连续手语识别通常被看作一种典型的弱监督序列识别任务: 由于难以确定手语词的边界, 现有的连续手语识别数据大多只有句子级别的标注, 而没有每个手语词在视频中的开始和结束时间的标注信息. 早期的连续手语识别方法 [18,19] 采用隐马尔可夫 (Markov) 模型, 使用多个隐状态表示一个手语词, 通过建模状态转移过程捕捉手工提取的视觉特征和句子标签之间的对应关系. 随着深度学习的发展, 一些工作尝试结合 CNN 的视觉特征提取能力和 HMM 的时序建模能力. Koller 等 [2] 提出了一种迭代式的混合模型, 利用 HMM 解码出的状态序列作为标签训练 CNN, 再利用训练好的 CNN 提取更具判别性的视觉特征. Camgöz 等 [20] 使用了不同的专家网络分别建模整体和手部序列特征并进行融合, 利用 CTC 实现了端到端的训练. 后续的一些工作发现 [3,4], 使用 CTC 端到端地训练连续手语识别模型并不能得到鲁棒的特征提取器, 迭代地训练特征提取器和整体网络可以达到更好的效果. Min 等 [5] 将该问题归因于时序模型的过拟合, 通过在视觉特征上引入了额外的视觉对齐约束, 实现了更高效的连续手语识别训练方法. 上述方法大多采用了 2D-CNN 逐帧提取视觉特征, 一些工作 [4,21~25] 尝试使用 3D-CNN 提取时序特征且取得了一定的性能提升, 但同时也需要更多的计算资源. 除此之外, 3D-CNN 具有更大的参数量和搜索空间, 现有的连续手语数据量不足以使之得到充分训练. 不同于上述工作, 本文在 2D-CNN 的浅层引入少量时空卷积, 并通过额外的辅助损失加以约束, 同时与时序下采样层相结合, 旨在减少计算量的同时保留更多的细粒度时空信息.

除了时空特征提取模块的选择, 帧率是影响手语识别效率的另一个要素, 但还未引起足够的重视. 高帧率带来更好的性能, 但需要更多的计算资源; 低帧率可以更快地获取识别结果, 但难以保障识别结果的准确性. 早期的手语研究工作 [10] 发现, 针对手语孤立词的识别, 帧率从 30 FPS 降到 10 FPS 不会对手语词的可理解性带来实质性的影响, 但是从 10 FPS 降到 5 FPS 会显著地影响手语的可理解性. 6 FPS 被一些工作 [10~12] 认为是保证手语可理解性的一个帧率下限. 现有的连续手语识别工

作 [3~5, 16] 使用数据集的原始帧率 (通常大于 25 FPS), 虽然达到很好的性能, 但存在很多冗余的计算. 为了减少冗余的视觉信息, 本文基于视觉信息对手语视频序列动态地下采样. 动态时间规整 (dynamic time warping, DTW) 是一种经典的度量两个不同长度序列相似度的方法, 通过对时间序列进行缩放, 实现两个序列之间最小代价的单调对齐. 近期的一些算法 (如 soft-DTW [26]) 也对其进行改进, 使得对齐过程可微并可用作神经网络的监督信号. 和本文工作比较相近的是动态时序池化 (dynamic temporal pooling, DTP) [27], DTP 利用分类器的权重实现序列的动态分段及池化, 以获得更准确的预测. 不同于 DTP, 本文所提出的 ADTP 利用浅层特征的自相似性进行模型加速, 可应用于模型的浅层, 无需借助分类器权重即可实现序列的动态分段.

动作识别是计算机视觉中与手语识别任务相似, 但更为迫切地需要模型加速以实现下游应用的研究方向. 基于深度学习的动作识别方法通常使用 2D [28, 29] 或 3D 卷积模型 [30~32] 从视频中提取时空特征以进行动作识别, 虽然在公开数据集上已经取得了较好的性能, 但也需要较大的计算资源. 一些工作 [33~36] 从分辨率和帧率等角度对时空特征的提取过程进行优化. SlowFast [34] 使用快慢通道分别提取不同帧率的特征并进行融合, 慢通道使用较低的固定帧率和常规的 3D 网络, 而快通道则使用较高的固定帧率和轻量化的 3D 网络, 利用帧间的互补信息在降低每帧平均计算量的同时提升识别精度. SCSampler [35] 则通过一个额外的评分模型, 基于视觉和音频信息选择显著的视频片段以进行动作识别, 可以在降低计算量的同时提升识别性能. AR-Net [37] 通过一个策略网络, 选择合适的分辨率和关键帧进行识别. 除了降低识别过程中的计算量, 自适应的关键帧选择算法还可以生成视频摘要 [38, 39], 快速定位关键事件的发生 [40, 41], 在视频相关领域中有着巨大的应用前景. 与上述动作识别方法不同, 本文提出的方法基于手语识别的特点, 利用浅层特征的自相似进行关键帧的选择, 无需额外的关键帧选择模型.

3 方法

3.1 问题定义及所用框架

连续手语识别旨在按顺序预测图像序列 $\boldsymbol{x}=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_T\}$ 中出现的手语词 $\boldsymbol{w}=\{w_1,\ldots,w_N\}$, 其中的每一个手语词来自于手语字典 $\mathbb{D}$. 图 3 展示了一个典型的连续手语识别框架, 该框架使用卷积神经网络 2D-CNN 逐帧提取视觉特征, 利用时序卷积 1D-CNN 和双向长短期记忆 BiLSTM 依次提取短时序 $\boldsymbol{v}=\{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_{T'}\}$ 和长时序特征 $\boldsymbol{z}=\{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_{T'}\}$, 最终将长时序特征送入主分类器 $\mathcal{F}_p$ 进行预测. 为了获得鲁棒的视觉特征, 一个有效的方式是在训练阶段将视觉特征 $\boldsymbol{v}$ 送入辅助分类器 $\mathcal{F}_a$ 并加以约束 [5].

连续手语识别通常被视为一种弱监督序列识别任务, 为了利用序列标签 $\boldsymbol{w}$ 所提供的监督信号, CTC 引入了一个特殊的背景类 b 来表示无含义的片段 (如手语中的过渡片段), 含背景类的手语字典记作 $\mathbb{D}^+=\mathbb{D}\cup\{b\}$. 借助于这个背景类, 可以构建一个多对一的映射 $\mathcal{B}:\mathbb{D}^{+T}\to\mathbb{D}^{\leqslant T}$ 以对齐含背景类的预测结果 (称之为路径) $\boldsymbol{\pi}\in\mathbb{D}^{+T}$ 和与之对应的序列标签 $\boldsymbol{w}\in\mathbb{D}^{\leqslant T}$. 该映射的实现通过合并相同相邻预测结果, 随后去除其中的背景类, 如 $\mathcal{B}(\text{-}aaa\text{--}aabbb\text{-})=\mathcal{B}(\text{-}a\text{-}ab\text{-})=aab$. 基于这个映射, 给定序列标签的条件概率可以通过计算所有可行路径概率之和得到:

$$p(\boldsymbol{w}|\boldsymbol{x})=\sum_{\boldsymbol{\pi}}p(\boldsymbol{\pi},\boldsymbol{w}|\boldsymbol{x})=\sum_{\boldsymbol{\pi}\in\mathcal{B}^{-1}(\boldsymbol{w})}p(\boldsymbol{\pi}|\boldsymbol{x}),\tag{1}$$

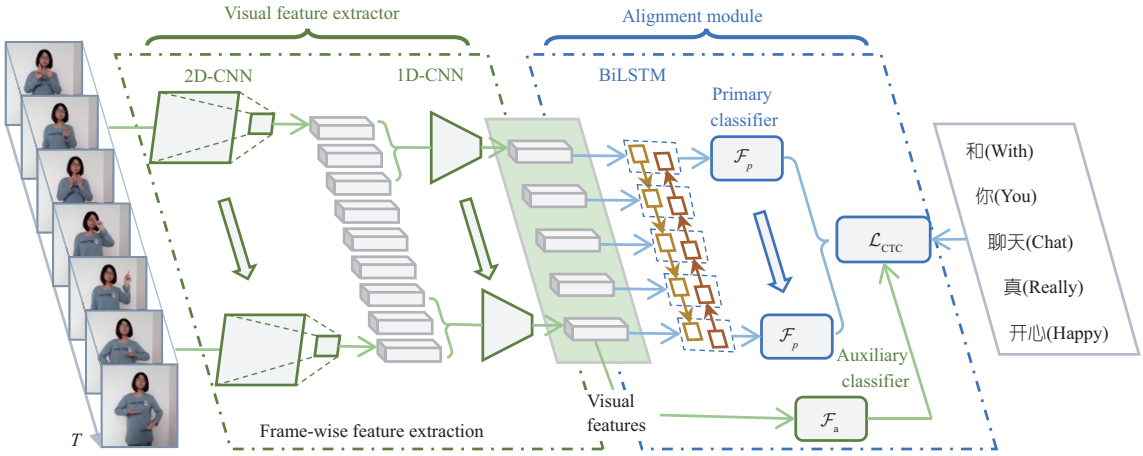

图 3 (网络版彩图) 连续手语识别框架结构图. 图像序列依次经过 2D-CNN, 1D-CNN 和 BiLSTM 模块提取视觉、短时序特征和长时序特征, 通过 CTC 实现预测结果和标签的对齐

Figure 3 (Color online) Details of the adopted CSLR framework. The image sequence is fed into the 2D-CNN, 1D-CNN, and BiLSTM modules to extract visual, short-term and long-term features, respectively. CTC loss provides supervision by aligning predictions and label sequences

表 1 连续手语识别框架中不同模块的参数量和计算量 (基于单个 200 帧的图像序列)

Table 1 Parameters and FLOPs of different modules in CSLR framework based on an image sequence with 200 frames

Module	2D-CNN (ResNet18)	1D-CNN	BiLSTM	Classifier	All
Params (M)	11.2	9.2	12.6	1.3	34.3
FLOPs (G)	363.8	1.1	0.6	0.1	365.5

可行路径条件概率 $p(\boldsymbol{\pi}|\boldsymbol{x})$ 的计算通过条件独立假设进行简化:

$$p(\boldsymbol{\pi}|\boldsymbol{x}) = \prod_{t=1}^{T'} y_{\pi_t}^t, \tag{2}$$

其中, $y_{\pi_t}^t = p(\pi_t|\boldsymbol{x})$ 是模型在 t 时刻预测类别为 π_t 的概率, 是 t 时刻的预测结果经过 softmax 函数计算得到的. 连续手语识别模型的优化即可通过最大化样本 $(\boldsymbol{x}, \boldsymbol{w})$ 的条件概率 $p(\boldsymbol{w}|\boldsymbol{x})$ 实现, 对应的损失函数为

$$L_{\mathrm{CTC}}(\boldsymbol{w}, \boldsymbol{x}) = -\ln\left(p(\boldsymbol{w}|\boldsymbol{x})\right). \tag{3}$$

相较于视觉特征提取模块 (2D-CNN), 时序模块所带来的计算量几乎可以忽略不计. 表 1 列出了基于 ResNet18 的连续手语识别网络中不同模块的参数量和计算量 (基于单个在 25 FPS 下长度为 200 帧的图像序列). 从表 1 中可以总结出手语识别模型训练和测试过程中主要的计算瓶颈在于视觉特征的提取, 占总计算量的 99.5% 以上. 减少视觉特征提取阶段所需的计算量, 是加速连续手语识别的一个重要途径, 也是本文所关注的重点.

3.2 自适应动态时序池化

受到计算资源和数据集规模的限制, 目前的连续手语识别算法往往逐帧提取视觉特征 (如图 3 所示), 再结合时序模型进行后续的时序特征提取. 但是作为一种视觉语言, 手语是通过静态和动态两种方式传递相关信息的: 静态手势的相邻帧存在较大的相关性, 逐帧提取特征的方法并未充分考虑这种

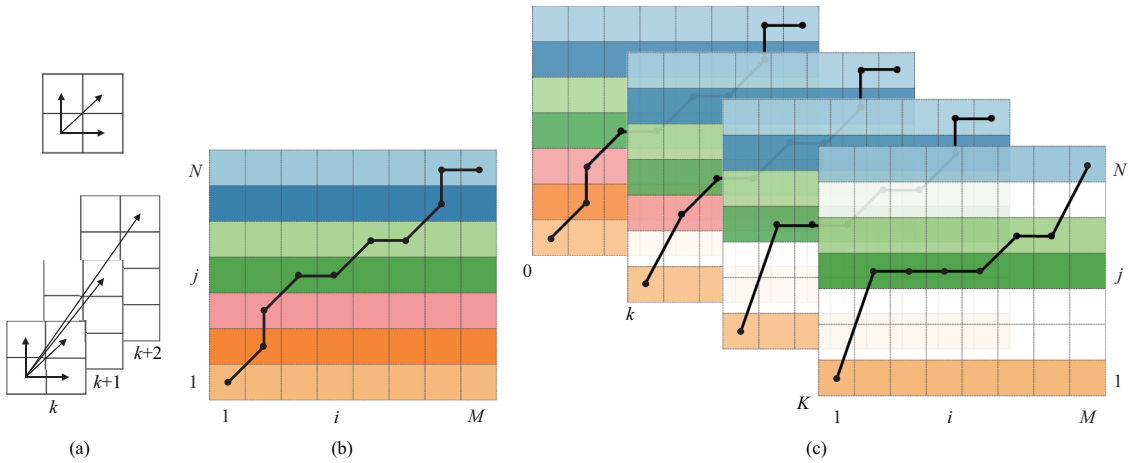

图 4 (网络版彩图) DTW 和 ADTW 对齐过程的对比示例. (a) 一步状态转移过程示意图, (上) DTW, (下) ADTW; (b) DTW 对齐结果; (c) 丢弃不同的帧数 (从 0 到 K), ADTW 的状态转移过程

Figure 4 (Color online) Comparison of warping results between DTW and ADTW. (a) Illustration of one-step transitions, DTW (top), ADTW (bottom); (b) the warping result of DTW; (c) the warping results of ADTW with different drop frames (from 0 to K)

冗余性, 也因此带来了额外的计算量; 动态手势的相邻帧差异较大, 虽然降低帧率可以减少运算量, 但也会带来一定的性能损失. 这是因为均匀的降采样会丢失动态信息, 进而影响最后的识别性能. 如何动态地根据手语内容选择合适的采样帧, 是一个值得探索的方向.

动态时间规整 [42] 是一种度量两个时间序列 $\boldsymbol{a}=\{\boldsymbol{a}_1,\dots,\boldsymbol{a}_M\}$ 和 $\boldsymbol{b}=\{\boldsymbol{b}_1,\dots,\boldsymbol{b}_N\}$ 之间相似度的算法. $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的长度可以不相等, 序列中每个事件的发生时间和持续时间也可能不同. DTW 通过对时间序列进行缩放, 实现两个序列之间最小代价的单调对齐, 进而计算两个序列的相似性. 整个对齐过程通过动态规划的方式计算得到:

$$\mathrm{DTW}[i,j]=\mathcal{D}[i,j]+\min(\mathrm{DTW}[i-1,j],\mathrm{DTW}[i,j-1],\mathrm{DTW}[i-1,j-1]), \tag{4}$$

其中, $\mathrm{DTW}[i,j]$ 表示将序列 $\boldsymbol{a}_{1:i}$ 和序列 $\boldsymbol{b}_{1:j}$ 对齐的最小累积代价, $\mathcal{D}[i,j]$ 表示 $\boldsymbol{a}_i$ 和 $\boldsymbol{b}_j$ 的对齐代价. 通过对动态规划的结果进行回溯, 可以进一步得到两个序列的对齐路径.

DTW 通过对序列的缩放实现不同序列的对齐, 进而计算两个序列的相似性. 受到 DTW 的启发, 本文提出了一种自适应动态时序规整 (adaptive dynamic time warping, ADTW) 算法, 给定两个时间序列 $\boldsymbol{a}=\{\boldsymbol{a}_1,\dots,\boldsymbol{a}_M\}$ 和 $\boldsymbol{b}=\{\boldsymbol{b}_1,\dots,\boldsymbol{b}_N\}$ 和它们的距离矩阵 $\mathcal{D}$, 通过动态规划的方式, 获得序列 $\boldsymbol{b}$ 在给定条件下与序列 $\boldsymbol{a}$ 具有最小代价的子序列 $\boldsymbol{b}'$, 进而对序列动态地下采样. 一种直观的约束条件就是设置下采样率, 而在此时, 状态转移的过程也会随之变化 (如图 4(a) 所示). 当约束时序下采样率为 β (即序列需要丢弃 $K=\beta N$ 帧, 子序列长度为 $N-K$) 时, 序列 $\boldsymbol{a}$ 和 $\boldsymbol{b}'$ 的相似度计算对应的动态规划转移方程为

$$\begin{aligned}\mathrm{ADTW}[i,j,k]=\mathcal{D}[i,j]+\min\Bigg(&\mathrm{ADTW}[i-1,j,k],\mathrm{ADTW}[i,j-1,k],\mathrm{ADTW}[i-1,j-1,k],\\&\min_{\Delta k\in[1,\eta]}\mathrm{ADTW}[i-1,j-1-\Delta k,k-\Delta k]\Bigg),\end{aligned} \tag{5}$$

其中, $\mathrm{ADTW}[i,j,k]$ 表示将序列 $\boldsymbol{a}_{1:i}$ 和 $\boldsymbol{b}'_{1:j-k}$ (采样自序列 $\boldsymbol{b}_{1:j}$) 对齐且 $\boldsymbol{b}'_{1:j-k}$ 以 $\boldsymbol{b}_j$ 结尾的最小累积

Algorithm 1 自适应动态时序规整

Input: 距离矩阵 $\mathcal{D} \in \mathbb{R}^{M\times N}$, 丢弃帧数 K 和最大连续丢帧间隔 η;
Output: 回溯对齐路径 $\boldsymbol{w}$;

```
dp ∈ ℝ^{(M+1)×(N+1)×(N−K+1)}, dp 矩阵初始化为浮点数最大值;
for k = 0, 1, ..., T − K do
    dp[k, 0, k] ⇐ 0;
end for
for k = 0, 1, ..., N − K do
    for i = 1, ..., M do
        for j = 1, ..., N do
            if i − 1 == k then
                dp[i, j, k] ⇐ D[i − 1, j − 1] + min(dp[i, j − 1, k], dp[i − 1, j − 1, k]);
            else
                v = min(dp[i − 1, j, k], dp[i, j − 1, k], dp[i − 1, j − 1, k]);
                for Δk = 1, ..., η do
                    v = min(v, dp[i − 1, j − 1 − Δk, k − Δk]);
                end for
                dp[i, j, k] ⇐ D[i − 1, j − 1] + v;
            end if
        end for
    end for
end for
```

Return: $\boldsymbol{w} = \text{backtrace}(\mathcal{D}, \text{dp}, K)$.

代价, η 表示采样的最大连续丢帧间隔, $\mathcal{D}[i,j]$ 表示 $\boldsymbol{a}_i$ 和 $\boldsymbol{b}_j$ 的对齐代价. 同样地, 通过对动态规划的结果进行回溯, 可以得到两个序列的对齐路径 $\boldsymbol{w} = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T\}$, 其中第 t 个元素定义为 $\boldsymbol{w}_t = (i,j)$, 表示 a_i 与 b_j 相对应.

图 4(a) 对比了 DTW 和 ADTW 的状态转移方式, 不同于原始 DTW 算法在相邻帧间进行状态转移, ADTW 通过记录丢弃帧数的状态实现了跨帧的状态转移 (如 $\text{ADTW}[i,j,k] = \mathcal{D}[i,j] + \text{ADTW}[i-1, j-2, k-1]$ 跳过了 b_{j-1}). 图 4(b) 和 (c) 分别展示了 DTW 和 ADTW 的对齐结果, 可以看出, ADTW 通过跨帧的状态转移实现了序列的下采样对齐. 算法 1 展示了基于保留帧数量更新状态的详细计算过程, 相较于原始的 DTW, ADTW 需要额外记录丢弃或保留的帧数以及进行跨帧状态转移, 也因此带来了一定的计算开销, 时间复杂度从 $O(MN)$ 增加到 $O\left(MN\min(N-K, K)\eta\right)$, 当进行两个序列的直接对齐 (即 $K=0$) 时, ADTW 会退化为 DTW 算法.

ADTW 提供了一种对序列动态采样的工具, 当用于计算序列及其子序列的最大相似度时, 所得到的对齐路径可用于序列特征的动态下采样, 在降低帧率的同时保留更多的关键信息. 本文进一步提出了自适应动态时序池化 (ADTP) 操作: 将待池化的视觉特征记作 $\boldsymbol{v} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_T\}$ 并给定池化后的关键帧序列长度 $T-K$, ADTP 根据由 ADTW 计算得到的对齐路径 $\boldsymbol{w}$ 使用最大池化操作聚合相似相邻帧特征:

$$\boldsymbol{v}'_{t,i} = \max_{(t,\tau)\in\boldsymbol{w}} \boldsymbol{v}_{\tau,i}. \tag{6}$$

相较于最大池化和平均池化的固定步长, 所提出的 ADTP 可以基于序列内容, 动态地调整池化步长, 在降低帧率的同时减少性能的损失.

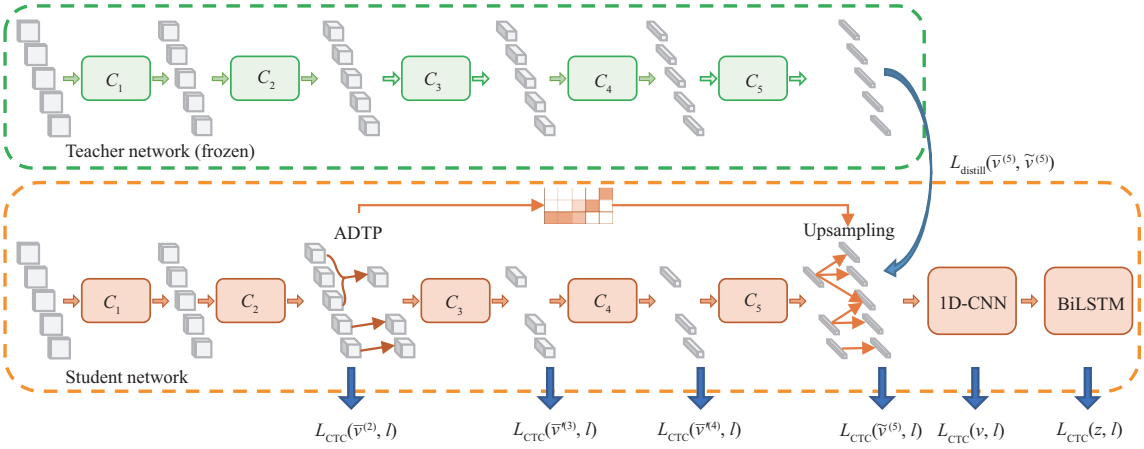

图 5 (网络版彩图) 第二阶段训练示意图. 在网络的浅层 (C_2) 基于帧间特征相似度进行下采样, 并在深层基于浅层的相似度恢复原始长度的特征序列. 第一阶段训练得到的模型作为教师网络引导第二阶段模型的训练

Figure 5 (Color online) Illustration of the second stage of the training process. The ADTP is performed based on the feature similarity among frames in the shallow layer (C_2), and the feature sequence of the original length is recovered in the deep layer based on the similarity of the shallow layer. The model trained in the first stage is used as the teacher to guide the training in the second stage

3.3 网络结构及损失函数设计

第 3.2 小节所提出的 ADTP 适用于任意模型, 本节中结合 ResNet [43] 讨论如何利用 ADTP 降低计算量, 将 ResNet 模型的第 1 个卷积层和第 1 ~ 4 个残差块记作 $C_1 \sim C_5$. 模型的计算量大多来自于卷积神经网络深层的特征提取, 如在 ResNet101 中, C_4 占总计算量的比例约为 66.3%. 在网络浅层进行时序下采样, 可以有效地减少特征提取所需的计算量.

具体来说, 给定输入图像序列 $\boldsymbol{x} \in \mathbb{R}^{T\times D\times H\times W}$ (T, D, H 和 W 分别对应于序列长度, 特征维数, 图像高度和图像宽度), 将经过第 k 个模块 C_k 的输出记作 $\boldsymbol{v}^{(k)} \in \mathbb{R}^{T\times D^{(k)}\times H^{(k)}\times W^{(k)}}$, 通过空间全局池化操作得到序列特征 $\bar{\boldsymbol{v}}^{(k)} \in \mathbb{R}^{T\times D^{(k)}}$. 式 (5) 中第 i 和 j 帧的对齐代价 $\mathcal{D}[i,j]$ 可以通过累加第 k 层相邻帧的余弦距离 ($d(x,y) = 1-\cos(x,y)$) 得到:

$$\mathcal{D}[i,j] = \begin{cases} 0, & \text{if } i=j, \\ \sum_{t=\min(i,j)}^{\max(i,j)-1} \left(1-\cos(\bar{\boldsymbol{v}}_t^{(k)}, \bar{\boldsymbol{v}}_{t+1}^{(k)})\right), & \text{else.} \end{cases} \tag{7}$$

为了减少时序降采样所带来的不可逆的信息损失, 进一步地提升时序下采样所依据视觉特征的连续性, 本文使用时空卷积操作保留更多的细粒度时空信息. 为了更好地平衡效率与性能, 本文仅将采样前模块 C_i 的第一个卷积层替换为 (2+1)D 卷积 [44]. 此外, 为了引导网络能够捕捉到泛化性更强的时序特征, 本文采用两阶段的方式进行模型的训练. 如图 5 所示, 在第一阶段, 不进行时序下采样, 并将训练好的模型作为教师网络引导第二阶段的训练. 而在第二阶段, 对 C_i 模块的输出使用 ADTP 进行时序下采样, 并基于浅层特征的帧间相似度重建原始的深度特征序列. 此外, 通过知识蒸馏 [45] 的方式, 利用第一阶段训练好的模型引导网络捕捉更多的时序信息, 进而增强模型的泛化性.

具体来说, 引入的时序卷积增加了模型的复杂性, 也增加了模型过拟合的风险. 受到之前工作 [5] 的启发, 本文在视觉特征提取的浅层网络 ($C_2 \sim C_5$) 中也引入了辅助分类器, 通过加权的 CTC 损失

为每一个样本 $(\boldsymbol{x}, \boldsymbol{w})$ 提供中继监督信号 L_{aux}. 第一阶段总的监督信号为

$$\begin{aligned} L_{\text{stage}_1} &= L_{\text{CTC}}(\boldsymbol{v}, \boldsymbol{l}) + L_{\text{CTC}}(\boldsymbol{z}, \boldsymbol{l}) + L_{\text{aux}}(\bar{\boldsymbol{v}}, \boldsymbol{l}), \\ L_{\text{aux}}(\bar{\boldsymbol{v}}, \boldsymbol{l}) &= \sum_{k=2}^{5} \alpha_k L_{\text{CTC}}(\bar{\boldsymbol{v}}^{(k)}, \boldsymbol{l}), \end{aligned} \tag{8}$$

其中, 前两项分别为辅助监督信号和主监督信号, 中继监督信号 $L_{\text{aux}}(\bar{\boldsymbol{v}}, \boldsymbol{l})$ 中的 α_k 是对第 k 个模块 C_k 的监督信号的权重.

在训练的第二阶段, 基于第一阶段的参数对网络进行微调并实现序列特征的下采样. 如图 5 所示, 在网络浅层 (以 C_2 层为例) 基于帧间特征相似度根据式 (5) ∼ (7) 进行动态时序池化, 得到降采样的浅层特征序列 $\boldsymbol{v}_t'^{(2)}$, 再利用逐帧的特征提取器得到降采样的深层特征序列 $\bar{\boldsymbol{v}}_t'^{(5)}$. 但是动态时序池化也破坏了特征本身的节奏信息 (如每个关键帧的持续时间长短, 会比较容易地被时序模型捕获). 为了还原序列的节奏信息, 基于浅层特征序列 $\bar{\boldsymbol{v}}'^{(2)}$ 和 $\bar{\boldsymbol{v}}^{(2)}$ 之间的相似度利用 $\bar{\boldsymbol{v}}'^{(5)}$ 线性插值恢复原始长度的特征序列 $\tilde{\boldsymbol{v}}^{(5)}$:

$$\tilde{\boldsymbol{v}}_t^{(5)} = \sum_{\tau} \frac{\mathrm{e}^{\gamma \cos(\theta(\bar{\boldsymbol{v}}_t^{(2)}), \phi(\bar{\boldsymbol{v}}_\tau'^{(2)}))}}{\sum_{t'} \mathrm{e}^{\gamma \cos(\theta(\bar{\boldsymbol{v}}_t^{(2)}), \phi(\bar{\boldsymbol{v}}_{t'}'^{(2)}))}} \bar{\boldsymbol{v}}_\tau'^{(5)}, \tag{9}$$

其中, $\theta(x) = W_\theta x$ 和 $\phi(x) = W_\phi x$ 是两个线性映射层, 用于增强模型的表达能力, γ 是控制加权系数平缓程度的超参数. 为了更好地捕捉到原始序列中的时序信息, 利用第一阶段模型完整的特征序列 $\bar{\boldsymbol{v}}^{(5)}$, 通过蒸馏的方式, 为线性插值过程提供约束信息:

$$L_{\text{distill}}(\bar{\boldsymbol{v}}^{(5)}, \tilde{\boldsymbol{v}}^{(5)}) = \text{CE}\left(f(\bar{\boldsymbol{v}}^{(5)}), g(\tilde{\boldsymbol{v}}^{(5)})\right), \tag{10}$$

其中, $\text{CE}(p, q) = -E_p[\log q]$ 计算概率分布 p 和 q 的交叉熵, 而 f 和 g 则是由线性映射层和 softmax 函数构成的, 将特征映射到类别的概率分布. 第二阶段总的监督信号为

$$\begin{aligned} L_{\text{stage}_2} &= L_{\text{CTC}}(\boldsymbol{v}, \boldsymbol{l}) + L_{\text{CTC}}(\boldsymbol{z}, \boldsymbol{l}) + L_{\text{aux}}(\bar{\boldsymbol{v}}, \boldsymbol{l}) + L_{\text{distill}}(\bar{\boldsymbol{v}}^{(5)}, \tilde{\boldsymbol{v}}^{(5)}), \\ L_{\text{aux}}(\bar{\boldsymbol{v}}, \boldsymbol{l}) &= \alpha_2 L_{\text{CTC}}(\bar{\boldsymbol{v}}^{(2)}, \boldsymbol{l}) + \sum_{k=3}^{4} \alpha_k L_{\text{CTC}}(\bar{\boldsymbol{v}}'^{(k)}, \boldsymbol{l}) + \alpha_5 L_{\text{CTC}}(\tilde{\boldsymbol{v}}^{(5)}, \boldsymbol{l}). \end{aligned} \tag{11}$$

需要注意的是, C_3 和 C_4 的辅助损失为下采样之后的序列提供监督信号, 而 C_5 的辅助损失则为重建之后的序列提供监督信号. 通过两阶段的训练方式, 可以在降低冗余帧引入的计算代价的同时, 保留更多的视觉和节奏信息.

4 实验结果

4.1 实验设置

数据集. 本文所提出的方法在主流的德国连续手语识别数据集 PHOENIX14 [16] 和近期的大规模中国手语识别和翻译数据集 CSL-Daily [17] 上进行验证. PHOENIX14 数据集从天气预报节目采集了长达 12.5 小时的手语数据, 采样帧率为 25 FPS, 所采图像的分辨率为 210×260. 整个数据集包含了由 9 位手语翻译者演示的 6841 个手语句子, 涉及 1295 个手语词, 平均每个词实例持续 14.8 帧. 数据集提供了官方的划分标准: 5672 个句子用于训练 (Train), 540 个句子用于验证 (Dev), 629 个句子用于测试 (Test). CSL-Daily 数据集则是由专业手语团队从中国手语教学书和中文百科全书中摘选约 2 万个

手语句子, 并由 10 位手语使用者录制而成, 总时长约 23.3 小时. 数据集提供了官方的划分标准: 18401 个句子用于训练, 1077 个句子用于验证, 1176 个句子用于测试.

除了上述公开数据集, 本文也在自采中国手语数据集上进行验证. 数据集在实验室场景下录制, 由 5 位手语者演示 209 个手语句子. 其中 4 位 (S1 ∼ S4) 每个句子演示 5 遍, 另一位 (S5) 每个句子演示两遍. 训练集由 S1 ∼ S4 演示的 3762 个句子 ($209 \times 3 \times 1 + 209 \times 5 \times 3$) 组成, 验证集和测试集各包含由训练集出现的演示者 (S1) 和未出现的演示者 (S5) 演示的 418 个句子 ($209 + 209$). 整个数据集共涉及 286 个手语词, 平均每个手语词实例持续 42.2 帧.

评测指标. 字错误率 (word error rate, WER) 是连续手语识别常用的评测指标, 通过动态规划将识别的单词序列和参考单词序列对齐, 统计对齐序列中替换 (sub)、删除 (del)、插入 (ins) 三类错误词数量占参考序列长度的比例, 反映识别准确性:

$$\mathrm{WER} = \frac{\#\mathrm{sub} + \#\mathrm{del} + \#\mathrm{ins}}{\#\mathrm{reference}}. \tag{12}$$

此外, 本文采用参数量 (Params) 和浮点运算数 (floating point operations, FLOPs) 衡量模型的计算效率.

实现细节. 本文采用 VAC [5] 作为基准模型 (baseline), 因为视觉特征序列和时序特征序列长度可能不一致, 为了简化训练过程, 只采用视觉强化约束而不使用视觉对齐约束. 此外, 本文对分类器权重和特征权重进行了归一化 [46] 以缓解类别不均衡的问题. 单阶段模型训练在单张 GeForce RTX 3090 上进行 40 轮, 在第 20 和 35 轮进行学习率衰减. 采用的数据增广与文献 [5] 一致. 为了减少视觉信息的丢失, ADTP 状态转移公式 (5) 中的最大连续丢帧间隔 η 设置为 10. 式 (9) 中的平滑超参数 γ 默认为 64. 式 (8) 和 (11) 中辅助监督信号的权重 $\alpha_2 \sim \alpha_5$ 分别为 0.2, 0.3, 0.3 和 0.5.

4.2 消融实验

本节通过设计消融实验, 对比了在不同位置引入时空卷积的效果, 以验证时序信息对视觉特征提取的影响. 通过定量分析帧率和时序下采样位置对识别性能等消融实验验证 ADTP 的有效性. 此外, 还进行了推断效率的对比, 与现有方法的性能对比以及在自采中国手语数据集上的实验.

时空卷积的位置选择. 使用 1D-CNN 提取时序信息虽然取得了较好的性能, 但需要完整地提取每帧图像的视觉特征. 如果在视觉特征提取的早期进行时序特征融合并降低采样率, 会带来计算效率的提升. 为了更高效地衡量时序降采样带来的影响, 考虑在 12 FPS 的输入帧率下, 在视觉特征提取的不同层引入一定的时序卷积操作, 通过比较时序卷积的位置对识别性能的影响, 为后续降采样的位置选择提供参考.

识别结果如表 2 所示, 将 ResNet18 不同残差块 ($C_1 \sim C_5$) 中涉及空间降采样的卷积层替换成 $(2+1)D$ 形式的时空卷积, 有助于结合时序信息保留更多重要的空间信息. 相较于基准模型, 在网络的不同层进行替换, 在增加少量计算代价的同时, 识别性能均有一定幅度的提升. 此外可以看到, 在特征提取网络的深层引入时空卷积, 会带来更多的性能提升 (在 C_1 处使用时空卷积, 在验证集和测试集分别减少了 0.5% 的字错误率, 在 C_4 处则分别减少了 1.6% 的字错误率), 这说明在进行一定程度抽象的视觉特征上进行时序融合, 会带来更好的效果.

消融实验. 本文提出了 ADTP 用于降低视觉特征提取过程中的计算量, 为了引导网络在降低计算量的同时减少性能的损失, 第 3.3 小节提出了多种网络结构和损失函数的设计. 为了更好地说明每个元件的有效性, 本小节对其进行消融实验并将结果展现在表 3 中. 可以看出, 上采样带来了一定的性能提升 (在验证集/测试集上字错误率减少了 1.2%/1.0%), 本文推测这是因为上采样恢复了一定的节

表 2 PHOENIX14 数据集上 (2 + 1)D 卷积的位置对识别性能 (字错误率, %) 的影响

Table 2 Evaluation results (WER, %) on PHOENIX14 with different positions of the (2 + 1)D convolutional module[a)]

Method	(2 + 1)D module position				FLOPs (G)	Params (M)	Dev	Test
	C_1	C_2	C_3	C_4				
Baseline (12 FPS)	–	–	–	–	**182.7**	**17.3**	21.2	21.6
Baseline + (2 + 1)D	✓	–	–	–	198.5	17.3	20.7	21.1
	–	✓	–	–	186.7	17.3	20.5	20.7
	–	–	✓	–	186.6	17.4	20.0	20.7
	–	–	–	✓	186.7	17.5	**19.6**	**20.0**

a) The best results are highlighted in bold.

表 3 PHOENIX14 数据集上 ADTP 的消融实验 (字错误率, %)

Table 3 Ablation results (WER, %) of ADTP on PHOENIX14[a)]

Upsample	Two-stage training	L_{aux}	L_{distill}	Dev		Test	
				del/ins	WER	del/ins	WER
–	–	–	–	11.4/3.3	25.7	9.9/3.1	25.4
✓	–	–	–	9.9/2.7	24.5	9.0/2.8	24.4
✓	✓	–	–	7.8/3.4	23.1	7.2/3.2	22.8
✓	✓	✓	–	8.1/3.1	22.7	7.6/2.7	22.4
✓	✓	✓	✓	7.3/3.3	**21.9**	7.1/3.0	**22.1**

a) The best results are highlighted in bold.

奏信息, 有利于固定感受野的时序卷积更准确地捕捉时序信息. 两阶段训练进一步带来明显的性能提升 (字错误率减少了 1.4%/1.6%), 这是因为 ADTP 的选帧过程不可微, 依赖于特征之间的相似度, 在第二阶段再进行时序降采样有助于提高训练的稳定性, 减少单阶段训练初期不稳定的特征所带来的负面影响. 此外, 在第二阶段引入的辅助损失和蒸馏损失, 也有助于进一步减少字错误率 (字错误率分别减少了 0.4%/0.4% 和 0.8%/0.3%). 这是因为这些中继监督信号有助于提升特征的判别性, 并对上采样重建得到的特征序列进行了约束, 保留了更多原始的序列信息.

推断效率的对比. 上述实验说明了所提出的 ADTP 的有效性, 为了进一步说明方法对推理速度的提升, 表 4 对比了在 PHOENIX14 数据集上不同帧率和时序下采样率组合的推断效率和性能, FLOPs 是基于单个在 25 FPS 下长度为 200 帧的图像序列计算得到的. 可以观察到, 降低帧率可以极大地减少计算量, 但性能也会有相对应的下降. 基准模型的帧率从 25 FPS 降低至 12 FPS 仅带来少量的相对性能损失 ($< 5\%$), 但是节省了大约 50% 的计算量. 从 25 FPS 到 6 FPS 可以节省 75% 的计算量, 但会带来明显的相对性能损失 ($> 15\%$). 由此可见, 降低帧率是一个降低计算量的有效方法, 且可以与其他基于图像或视频的加速方法兼容.

为了公平地对比在不同帧率下进一步进行时序下采样的性能, 本文根据帧率选择对应的下采样率, 使得时序下采样之后的特征序列平均帧率保持一致. 从实验结果可以看出, 相较于基准模型 (12 FPS, 182.7 GFLOPs), 所提出的方法可以在减少计算量的同时有效地减少性能的损失. 当中间层特征时序下采样到 3 FPS 时, 依然可以保持接近 24% 的字错误率 (而从输入层面采用 3 FPS, 仅达到约 33% 的字错误率), 相较于在不同位置进行最大池化下采样均有 4.2% 以上的性能提升, 这是因为 ADTP 可以根据视频内容动态地选择关键帧, 保留识别所需的关键信息. 当采用 8 FPS 的输入帧率时, 在 C_3 处使用

表 4　PHOENIX14 数据集上帧率和时序下采样的位置对识别性能 (字错误率, %) 的影响

Table 4　Evaluation results (WER, %) on PHOENIX14 with different framerates and temporal downsampling (TDS) positions[a)]

Framerate (FPS)	TDS ratio	TDS position			TDS method	FLOPs (G)	Params (M)	Dev	Test
		C_2	C_3	C_4					
25	–	–	–	–	–	365.3	17.9	**20.0**	**21.1**
12	–	–	–	–	–	182.7	17.3	21.2	21.6
8	–	–	–	–	–	120.7	16.8	20.7	22.1
6	–	–	–	–	–	91.5	16.5	23.1	23.7
3	–	–	–	–	–	**45.7**	**16.0**	32.9	33.2
12	0.25	✓	–	–	Max Pooling	**93.7**	**17.0**	29.3	28.7
		–	✓	–	Max Pooling	124.6	16.1	26.8	26.9
		–	–	✓	Max Pooling	155.5	16.2	**24.7**	**25.3**
12	0.25	✓	–	–	ADTP	**93.5**	**17.3**	24.3	24.4
		–	✓	–	ADTP	124.4	17.4	21.9	22.1
		–	–	✓	ADTP	155.2	17.6	**20.5**	**21.3**
8	0.375	✓	–	–	ADTP	**72.3**	**17.3**	23.8	24.0
		–	✓	–	ADTP	89.2	17.4	22.0	21.8
		–	–	✓	ADTP	106.1	17.6	**20.9**	**21.7**

a) The best results are highlighted in bold.

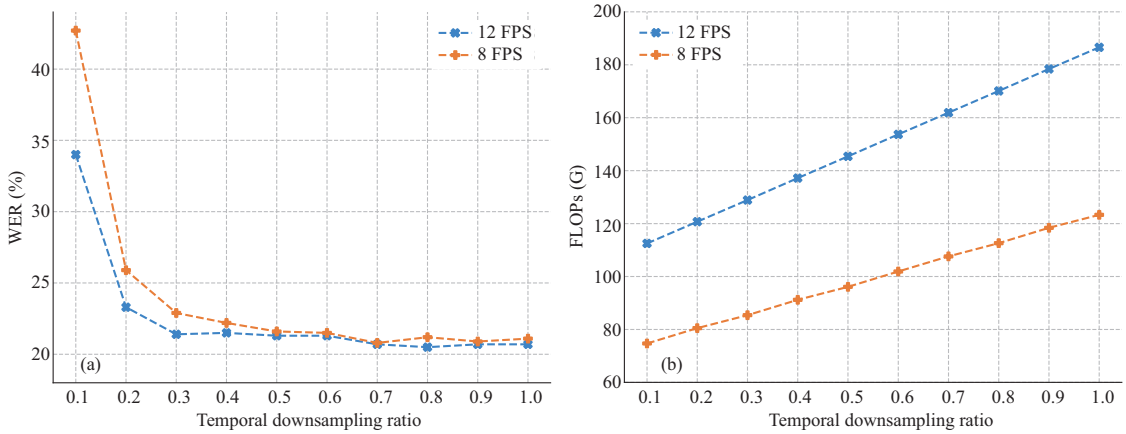

图 6　(网络版彩图) PHOENIX14 数据集上不同的帧率和时序下采样率对识别性能和计算量的影响

Figure 6　(Color online) (a) Performance and (b) computational cost of different framerates and temporal downsampling ratios on PHOENIX14

ADTP 可以达到比基准模型 (6 FPS) 更低的字错误率 (22.0%/21.8% vs. 23.1%/22.1%) 且更少的运算量 (89.2 GLOPs vs. 91.5 GLOPs), 进一步地说明了方法的有效性.

图 6 具体展示了不同的时序下采样率对识别性能和计算量的影响. 可以注意到, 虽然特征序列平均帧率一致, 采用 12 FPS 的视频序列在不同时序下采样率下都取得了更好的效果, 这是因为时空卷积可以保留更多的时空信息且为 ADTP 提供了更详细的时空信息. 此外还可以注意到, 输入帧率对计算量有较大影响, 这是因为 ADTP 需要具有一定判别性的特征, 所以在特征提取的中间层进行, 无法

表 5 在公开数据集上和主流方法的性能对比 (字错误率, %)

Table 5 Performance comparison (WER, %) on PHOENIX14 dataset[a)]

Method	Backbone	Frame-wise FLOPs (G)	PHOENIX14		CSL-Daily	
			Dev	Test	Dev	Test
Re-Sign [47]	GoogLeNet	1.5	27.1	26.8	–	–
STMC [48]	VGG11	7.5	25.0	–	–	–
SFL [7]	ResNet18	1.8	24.9	25.3	–	–
DNF [3]	GoogLeNet	1.5	23.8	24.4	32.8	32.4
FCN [6]	Custom	**1.4**	23.7	23.9	33.2	33.5
CMA [49]	GoogLeNet	1.5	21.3	21.9	–	–
VAC [5]†	ResNet18	1.8	19.9	20.9	33.3	32.6
SMKD [46]†	ResNet18	1.8	**19.8**	**20.5**	28.4	27.5
TwoStream-SLR-V [25]	S3D	1.1	21.1	22.4	28.9	28.5
CNN+LSTM+HMM [50]*(M)	GoogLeNet	**1.5**	26.0	26.0	–	–
DNF [3]*(O)	GoogLeNet	**1.5**	23.1	22.9	–	–
STMC [48]*(H+F+P)	VGG11	7.5	**21.1**	**20.7**	–	–
TwoStream-SLR [25]*(P)	S3D	2.2	**18.4**	**18.8**	25.4	24.3
Baseline	ResNet18	0.9	**21.2**	**21.6**	29.6	29.5
Baseline + ADTP (S1)	ResNet18	0.6	21.9	22.1	32.8	32.0
Baseline + ADTP (S2)	ResNet18	**0.4**	22.0	21.7	33.5	32.6

a) The entries denoted by "*" used extra clues (M for mouth, O for optic flow, H for hand, F for face, and P for pose). The entries denoted by "†" are our reimplementation using released codes, which achieve better performance than the original papers because of the use of syncBN and normalization. On the PHOENIX14 dataset, the framerate for baseline is 12 FPS, and settings S1 and S2 correspond to (12 FPS, 0.25) and (8 FPS, 0.375), respectively. On the CSL-Daily dataset, the framerate for baseline is 15 FPS, and settings S1 and S2 correspond to (15 FPS, 0.25) and (8 FPS, 0.375), respectively. The best results are highlighted in bold.

影响之前的特征提取过程. 即使只保留 10% 的帧, 采用 12/8 FPS 的输入也依然有 112.5/74.7 GLOPs 的计算量. 如需进一步的时序层面的加速, 可与更高效的浅层视觉特征提取方法结合.

与现有方法的性能对比. 表 5 [3,5~7,25,46~50] 展示了本文所提出的 ADTP 模块和其他方法在主流数据集上的性能对比. 近期的连续手语识别方法可大致分为两类: 一类是基于迭代式优化的训练策略 [50,51], 结合其他的视觉线索 (如嘴型、姿态) 等, 提取更强的视觉特征; 另一类是基于端到端的训练策略 [5,6], 通过中继监督信号的设计更有效地引导视觉和时序特征的提取. 本文沿用了后一类的思路, 基于 VAC [5] 的识别框架, 在浅层网络引入时序信息并进行时序下采样. 从表 5 中可以看到, 所提出的方法 (在 C_3 层使用 ADTP 进行时序下采样) 虽然为了提升计算效率损失少量性能, 但依然和其他方法是可比的, 甚至好于部分使用额外标注数据的方法 [3,50]. 因为鲜有方法汇报模型所需计算量, 且本文所提方法主要针对视觉特征提取阶段, 所以采用等效主干 FLOPs (平均每帧特征提取所需的计算量) 比较各个方法的计算效率. 可以看到, 相较于其他方法而言, 降低帧率并使用 ADTP 可以在损失少量性能的情况下, 节省 70% 以上的计算量, 这也说明了手语识别中存在的冗余计算. 随着连续手语识别算法精度的提升, 如何设计一个更高效的手语识别模型, 是一个值得进一步探索的方向.

帧率对连续中国手语识别的影响. 世界上并没有一种通用的手语, 为探究帧率对连续中国手语识别的影响并进一步验证方法的有效性, 表 6 展示了所提出的 ADTP (对 C_2 层的输出使用 ADTP 进行

表 6　在自采中国手语数据集上帧率在不同方法下对识别性能 (字错误率, %) 的影响

Table 6 Evaluation results (WER, %) on the collected Chinese sign language dataset with different framerates and settings[a)]

Framerate (FPS)	Baseline			Baseline + ADTP			
	E-mode	T-mode	FLOPs (G)	TDS ratio	E-mode	T-mode	FLOPs (G)
30	5.7/5.8	4.6/4.2	364.4	0.125	3.2/2.8	4.0/5.3	228.1
15	**3.0**/**4.1**	3.2/**3.2**	182.2	0.25	**1.8**/**1.5**	3.6/3.0	124.3
10	4.4/4.0	3.5/3.8	120.4	0.375	2.8/3.7	3.9/4.1	89.2
8	4.5/3.9	**3.1**/3.7	**91.2**	0.5	2.1/1.9	**2.5**/**2.4**	**72.5**

a) E/T-mode represents using statistics of the training or the current sequence for evaluation, respectively. The best results are highlighted in bold.

时空降采样) 在自采中国手语数据集不同帧率下的性能情况. 因为测试集中存在训练集中未出现的手语演示者, 训练集中的统计量并不一定适用于未见过的手语演示者, 所以表 6 中也展示了使用训练模式 (BN 基于单个视频序列的统计量) 的测试结果. 可以注意到, 所提方法在显著降低计算量的情况下, 在不同帧率下都达到了和基准模型可比的性能, 验证了方法在不同手语语种上的泛化性. 由于中国手语数据集是在实验室场景下采集得到的, 速度慢于正常交流的手语速度, 采用较低的帧率 (如 15 FPS) 反而可以取得更好的性能. 这是因为较多冗余帧信息会带来较大的插入错误, 也说明了帧率在手语识别算法实际应用过程中的重要性. 此外, 可以注意到, 统计量在连续手语识别中发挥了重要的作用, 受限于有限的手语演示者, 基准模型在推断阶段使用当前样本统计量 (T-mode) 可以达到更好的识别性能, 而基准模型 + ADTP 则相反. 本文推测这是因为 ADTP 对于不同手语演示者采样得到的关键帧子集具有一定的一致性, 有助于提高连续手语识别模型的泛化性.

4.3　定性试验

除了降低视觉特征提取过程中的计算量, 本文所提出的自适应动态时间规整还可用于分析手语视频的结构信息, 生成简略直观的视频摘要. 自动视频摘要生成 [52] 旨在利用计算机视觉的方法生成可以概括视频主要内容的图像帧序列或视频片段. 不同于普通视频, 手语视频天然地存在结构信息, 即其中的手语词和手语基元 (类似于语音中的音素 [53]), 自动手语视频摘要生成有助于手语使用者快速地定位并获取到所需信息. 图 7 对比了基于 ADTW 的采样结果和均匀采样结果 (8 倍下采样率). 相较于均匀采样, 基于 ADTW 的采样结果减少了对运动速度较慢片段的采样帧数 (如开始和结束过程中的静态帧), 同时更准确地捕捉到了手语演示过程中的关键帧 (如 “已经” 和 “顾客” 中的动态帧), 这也解释了为什么 ADTP 可以在减少计算量的同时达到具有竞争力的性能. 此外还可以观察到, 基于姿态/手型 (如 “满”) 和基于轨迹的手语词 (如 “顾客”), 捕捉得到的关键帧数量不同. 本文提出的上采样方法, 有助于恢复原始序列中的节奏信息 (关键词的持续时间), 进而减少对手语词的漏检和误检.

5　总结与展望

连续手语识别需要捕捉图像序列中的细粒度视觉和运动信息, 为了更高效地利用手语中的时空信息, 本文在视觉特征提取的浅层引入时序卷积融合时空信息, 并提出了自适应动态时序池化层, 基于视觉特征的相似性自适应地选择手语视频序列中的关键帧并对视频进行下采样, 此外提出了一种两阶段的训练方式以更好地捕捉时空信息. 在公开数据集上的实验结果验证了所提方法的有效性, 即在损失少量性能的情况下节省大量的计算成本, 可以更好地平衡连续手语识别的准确率和效率. 除此之外,

图 7 (网络版彩图) 时序下采样结果可视化. (a) ADTW 采样结果; (b) 均匀采样结果

Figure 7 (Color online) Visualization of downsampled frames with ADTW (a) and uniform sampling (b)

所提出的 ADTP 也可用于手语识别结构分析, 生成简略直观的手语视频摘要.

随着连续手语识别与翻译技术的发展, 推动手语技术的进一步落地是未来的一个重要方向. 在实际应用场景中, 需要基于视频传输速度和算力的实际情况, 选择合适的模型进行识别. 本文所提出的方法还提供了一个探索性思路, 即在端侧运行浅层模型, 使用 ADTP 对时序特征压缩并进行传输; 云侧设备接收到数据后, 使用深层模型进行后续识别操作. 同时如图 7 所示, 下采样的结果也可以为模型的识别结果提供一定的可解释性. 此外, 模型的压缩与加速是任何深度学习任务落地时需要面临的共性问题, 而手语识别模型在压缩与加速过程中的特性问题还未被充分挖掘, 本文尝试从帧率的角度进行探索, 并得到一些有趣的结论, 如在 PHOENIX14 数据集上, 将基准模型的输入帧率降低到 6 FPS 时, 字错误率仅增加至 23.1%; 当使用 ADTP 方法将中间特征层降至 3 FPS 时, 字错误率仅增加至 24.3%. 这些结论说明了手语视频在时序上的冗余性, 如何更好地利用这种冗余性设计更稳定的识别模型和更高效的推断模型, 是一个值得进一步探索的方向.

参考文献

1 World Health Organization. Deafness and hearing loss. 2023. [2023-09-19]. https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

2 Koller O, Zargaran O, Ney H, et al. Deep sign: hybrid CNN-HMM for continuous sign language recognition. In: Proceedings of the British Machine Vision Conference, 2016

3 Cui R, Liu H, Zhang C. A deep neural framework for continuous sign language recognition by iterative training. IEEE Trans Multimedia, 2019, 21: 1880–1891

4 Pu J, Zhou W, Li H. Iterative alignment network for continuous sign language recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 4165–4174
5 Min Y, Hao A, Chai X, et al. Visual alignment constraint for continuous sign language recognition. In: Proceedings of the IEEE International Conference on Computer Vision, 2021. 11542–11551
6 Cheng K L, Yang Z, Chen Q, et al. Fully convolutional networks for continuous sign language recognition. In: Proceedings of the European Conference on Computer Vision, 2020. 697–714
7 Niu Z, Mak B. Stochastic fine-grained labeling of multi-state sign glosses for continuous sign language recognition. In: Proceedings of the European Conference on Computer Vision, 2020. 172–186
8 Li D, Xu C, Yu X, et al. TSPNet: hierarchical feature learning via temporal semantic pyramid for sign language translation. In: Proceedings of Advances in Neural Information Processing Systems, 2020. 12034–12045
9 Read P, Meyer M P. Restoration of Motion Picture Film. Oxford: Butterworth-Heinemann, 2000
10 Sperling G, Landy M, Cohen Y, et al. Intelligible encoding of ASL image sequences at extremely low information rates. In: Human and Machine Vision II. Amsterdam: Elsevier, 1986. 256–312
11 Cherniavsky N, Cavender A C, Ladner R E, et al. Variable frame rate for low power mobile sign language communication. In: Proceedings of the International ACM SIGACCESS Conference on Computers and Accessibility, 2007. 163–170
12 Foulds R A. Biomechanical and perceptual constraints on the bandwidth requirements of sign language. IEEE Trans Neural Syst Rehabil Eng, 2004, 12: 65–72
13 Ong S C W, Ranganath S. Automatic sign language analysis: a survey and the future beyond lexical meaning. IEEE Trans Pattern Anal Machine Intell, 2005, 27: 873–891
14 Sandler W, Lillo-Martin D. Sign Language and Linguistic Universals. Cambridge: Cambridge University Press, 2006
15 Min Y, Jiao P, Li Y, et al. Deep radial embedding for visual sequence learning. In: Proceedings of the European Conference on Computer Vision, 2022. 240–256
16 Koller O, Forster J, Ney H. Continuous sign language recognition: towards large vocabulary statistical recognition systems handling multiple signers. Comput Vision Image Understanding, 2015, 141: 108–125
17 Zhou H, Zhou W, Qi W, et al. Improving sign language translation with monolingual data by sign back-translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021. 1316–1325
18 Gao W, Fang G, Zhao D, et al. A Chinese sign language recognition system based on SOFM/SRN/HMM. Pattern Recognition, 2004, 37: 2389–2402
19 Farhadi A, Forsyth D. Aligning ASL for statistical translation using a discriminative word model. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006. 1471–1476
20 Camgöz N C, Hadfield S, Koller O, et al. SubUNets: end-to-end hand shape and continuous sign language recognition. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 3075–3084
21 Zhou H, Zhou W, Li H. Dynamic pseudo label decoding for continuous sign language recognition. In: Proceedings of the IEEE International Conference on Multimedia and Expo, 2019. 1282–1287
22 Li D, Rodriguez C, Yu X, et al. Word-level deep sign language recognition from video: a new large-scale dataset and methods comparison. In: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2020. 1459–1469
23 Zhou Z, Lui K S, Tam V W, et al. Applying (3 + 2 + 1)D residual neural network with frame selection for Hong Kong sign language recognition. In: Proceedings of the International Conference on Pattern Recognition, 2021. 4296–4302
24 Zhou Z, Tam V W L, Lam E Y. SignBERT: a BERT-based deep learning framework for continuous sign language recognition. IEEE Access, 2021, 9: 161669–161682
25 Chen Y, Zuo R, Wei F, et al. Two-stream network for sign language recognition and translation. In: Proceedings of Advances in Neural Information Processing Systems, 2022
26 Cuturi M, Blondel M. Soft-DTW: a differentiable loss function for time-series. In: Proceedings of the International Conference on Machine Learning, 2017. 894–903
27 Lee D, Lee S, Yu H. Learnable dynamic temporal pooling for time series classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 8288–8296
28 Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. In: Proceedings of Advances in Neural Information Processing Systems, 2014

29 Wang L, Xiong Y, Wang Z, et al. Temporal segment networks: towards good practices for deep action recognition. In: Proceedings of the European Conference on Computer Vision, 2016. 20–36

30 Tran D, Wang H, Torresani L, et al. A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 6450–6459

31 Wang X, Girshick R, Gupta A, et al. Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 7794–7803

32 Feichtenhofer C. X3D: expanding architectures for efficient video recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 203–213

33 Wu Z, Xiong C, Jiang Y G, et al. LiteEval: a coarse-to-fine framework for resource efficient video recognition. In: Proceedings of Advances in Neural Information Processing Systems, 2019

34 Feichtenhofer C, Fan H, Malik J, et al. SlowFast networks for video recognition. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 6202–6211

35 Korbar B, Tran D, Torresani L. SCSampler: sampling salient clips from video for efficient action recognition. In: Proceedings of the IEEE International Conference on Computer Vision, 2019. 6232–6242

36 Gao R, Oh T H, Grauman K, et al. Listen to look: action recognition by previewing audio. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. 10457–10467

37 Meng Y, Lin C C, Panda R, et al. AR-Net: adaptive frame resolution for efficient action recognition. In: Proceedings of the European Conference on Computer Vision, 2020. 86–104

38 Gong B, Chao W L, Grauman K, et al. Diverse sequential subset selection for supervised video summarization. In: Proceedings of Advances in Neural Information Processing Systems, 2014

39 Zhang K, Chao W L, Sha F, et al. Summary transfer: exemplar-based subset selection for video summarization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 1059–1067

40 Yeung S, Russakovsky O, Mori G, et al. End-to-end learning of action detection from frame glimpses in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 2678–2687

41 Lei J, Li L, Zhou L, et al. Less is more: ClipBERT for video-and-language learning via sparse sampling. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021. 7331–7341

42 Sakoe H, Chiba S. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech Signal Process, 1978, 26: 43–49

43 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778

44 Qiu Z, Yao T, Mei T. Learning spatio-temporal representation with pseudo-3D residual networks. In: Proceedings of the IEEE International Conference on Computer Vision, 2017. 5533–5541

45 Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. In: Proceedings of Neural Information Processing Systems Deep Learning and Representation Learning Workshop, 2014

46 Hao A, Min Y, Chen X. Self-mutual distillation learning for continuous sign language recognition. In: Proceedings of the IEEE International Conference on Computer Vision, 2021. 11303–11312

47 Koller O, Zargaran S, Ney H. Re-sign: re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 4297–4305

48 Zhou H, Zhou W, Zhou Y, et al. Spatial-temporal multi-CUE network for continuous sign language recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 13009–13016

49 Pu J, Zhou W, Hu H, et al. Boosting continuous sign language recognition via cross modality augmentation. In: Proceedings of the ACM International Conference on Multimedia, 2020. 1497–1505

50 Koller O, Camgöz N C, Ney H, et al. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. IEEE Trans Pattern Anal Mach Intell, 2020, 42: 2306–2320

51 Cui R, Liu H, Zhang C. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 7361–7369

52 Zhang K, Chao W L, Sha F, et al. Video summarization with long short-term memory. In: Proceedings of the European Conference on Computer Vision, 2016. 766–782

53 Wang H, Chai X, Chen X. Sparse observation (SO) alignment for sign language recognition. Neurocomputing, 2016, 175: 674–685

Adaptive keyframe selection for continuous sign language recognition

Yuecong MIN[1,2] & Xilin CHEN[1,2*]

1. *Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190*, China*;
2. *School of Computing Science and Technology, University of Chinese Academy of Sciences, Beijing* 100049*, China*
* Corresponding author. E-mail: xlchen@ict.ac.cn

Abstract Vision-based continuous sign language recognition (CSLR), which aims to recognize unsegmented signs from image sequences, provides a convenient communication tool for sign language users. Recent CSLR approaches often extract visual and contextual features frame by frame from image sequences, leading to redundant computations due to the presence of similar visual information in adjacent frames. This paper analyzes the impact of framerate on continuous sign language recognition algorithms and finds that reducing the framerate significantly improves computational efficiency but may also result in performance degradation. To preserve more key sign language information while reducing computational cost, this paper proposes an adaptive dynamic temporal pooling (ADTP) layer that dynamically downsamples sequences based on their self-similarity in sequence features. Furthermore, a two-stage training scheme is introduced to better utilize the spatiotemporal information in original sequences. Specifically, in the first stage, the CSLR model is trained based on original sequences, and in the second stage, the model with the ADTP module is trained with knowledge distillation guided by the teacher network from the first stage. Experimental results demonstrate that the proposed method significantly reduces the computational requirements for recognition while only sacrificing a small amount of performance. Additionally, the proposed ADTP can also be applied to sign language video structure analysis, generating concise and intuitive summaries of sign language videos.

Keywords continuous sign language recognition, time series analysis, visual languages, knowledge distillation, computational efficiency