文章编号：1005-5630(2022)04-0016-10 **DOI**: 10.3969/j.issn.1005-5630.2022.004.003

一种用于动作识别的双分支网络

秦晓飞[1]，蔡 锐[1]，陈 萌[2]，张文奇[2]，何常香[1]，张学典[1]
（1. 上海理工大学 光电信息与计算机工程学院，上海 200093；
2. 上海宇航系统工程研究所，上海 201109）

摘要：动作识别是计算机视觉领域的一项重要任务，主要有基于 RGB 视频和人体骨架两种数据模态的领域，主流方法分别是 3D 卷积神经网络和图卷积神经网络。针对视频和人体骨架两种数据模态的不同特点，设计了双分支网络分别对两种数据模态进行建模。对于人体骨架数据，基于自注意力机制设计了图卷积神经网络，该算法能在基于骨架的动作识别任务中达到先进的性能。对于视频数据，采用 3D 卷积网络进行特征提取。同时，利用深监督方法对两种数据模态的中间特征进行监督，提高两种数据特征的耦合度，进一步提高网络效率。这种算法的网络结构简单，在 NTU-RGBD60（CS）数据集上仅用 3.37×10^7 的参数量可达到 95.6% 的精度。

关键词：基于人体骨架的动作识别；图卷积神经网络；自注意力机制；3D 卷积神经网络
中图分类号：TP 391 **文献标志码**：A

A dual-branch network for action recognition

QIN Xiaofei[1]，CAI Rui[1]，CHEN Meng[2]，ZHANG Wenqi[2]，HE Changxiang[1]，ZHANG Xuedian[1]
（1. School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China;
2. Institute of Aerospace System Engineering Shanghai, Shanghai 201109, China）

Abstract: Action recognition has always been an important task in the field of computer vision. There are mainly two tasks based on RGB video and human skeleton. The mainstream methods are 3D convolutional neural network and graph convolutional neural network. For the data modality of human skeleton, this work designs a graph convolutional neural network based on the self-attention mechanism. The algorithm can achieve advanced performance on skeleton-based action recognition tasks. In addition, a method is proposed to use deep supervision methods to supervise the intermediate features of video and human skeleton, which improves the coupling of the two data features and further improves network efficiency. The network structure of this algorithm is simple, and only 3.37×10^7 parameters are used to achieve an accuracy of 95.6% on the NTU-RGBD60 (CS) dataset.

收稿日期：2021-12-21
基金项目：上海市人工智能专项（2019-RGZN-01077）
作者简介：秦晓飞 (1982—)，男，高级工程师，研究方向为人工智能算法。E-mail：xiaofei.qin@foxmail.com

Keywords: skeleton-based action recognition； graph convolutional neural network； self-attention mechanism；3D convolution neural network

引　言

人体动作识别是视频理解中的一项重要任务，可用于视频分析、人机交互等应用场景[1-3]，主要的数据模态包括视频、人体骨架序列等。视频是生活中最常见的数据形式，由连续图像排列而成，利用图像的连续变化和视觉暂留原理，达到平滑连续的视觉效果。人体骨架序列是一种结构良好的人体姿态数据，通过使用目标检测和人体姿态估计技术，从视频数据的每一帧图像中获取人体主要关键点的空间位置，根据人体结构自然地将这些关键点连接起来，构成一个图。每个关键点代表图的顶点，每条连接线代表图的边[4]，人体骨架数据可以看作对视频数据提炼后的结果，仅保留了每个时刻的人体姿态信息。与视频数据相比，人体骨架数据只关注人本身的动作，忽视了背景、视角和人物外观。

主流的人体动作识别算法大多基于视频数据，研究者们提出了各种有影响力的算法，主要包括基于 2D、3D 卷积神经网络的算法。直接采用 3D 卷积神经网络对视频进行处理，是目前结构最简单、应用最广、效果最好的算法，其中 slow-fast[5] 算法为这个领域的算法提供了一个高效且结构简单的基准网络结构。这类方法能提取视频中丰富的特征信息，但是同时也会带来噪声，在动作识别任务中，降低了对人这一主体的关注度，可以引入人体关键点检测技术来提升对人的关注度。

近些年来，随着人体姿态估计技术和图神经网络（GCN）技术的发展，人体骨架数据的提取变得简单。许多研究人员开始使用图卷积神经网络来解决基于骨架的动作识别问题。图神经网络旨在保留图的拓扑结构和节点信息的情况下，使用机器学习方法，学习每个相连节点之间的相关性，实现节点之间的信息融合。通过堆叠图神经网络层，充分融合每个顶点的信息，实现对图中每一个节点的特征提取。针对基于人体骨架的动作识别这一任务，对人体骨架数据进行空间维度上的建模时，也要对时间维度进行建模。首先，对于空间维度而言，有些研究人员通过手动设计的人体的拓扑结构，提升人体关节点之间的信息融合效率，例如 MS-G3D [6]。还有一些研究人员提出了自适应 GCN 层[7-9]，即基于人体的自然结构，自适应地动态产生图结构。总而言之，这些方法都在试图通过优化人体骨架图的空间拓扑结构，在图结构中设计出高效的连接边，实现更加轻量化的网络模型和更高的精度。对于时间维度而言，一些研究者在相邻帧之间添加时间上的连接边，对图进行拓展，或者把相邻帧的特征直接交换，实现时间上的信息融合[10-14]。还有一些研究者通过将循环神经网络（RNN）和长短期记忆网络（LSTM）中的卷积神经网络层（CNN）替换为 GCN 层[15-17]，来实现时间上的信息交换。但是这种对时间维度的建模方式并不高效，伴随大量的计算量和参数，本文考虑采用简单的卷积方式进行替代，结合自注意力机制，设计轻量级模块在时间和空间维度上同时进行建模。

视频数据则包含了更丰富的环境信息，人体骨架数据利于对人体动作进行精准建模，将两者结合起来进行动作识别，能够有效提升动作识别的精度。一些研究者将人体骨架数据还原为热度图，与视频数据融合，增强了人体动作特征的表达，在动作识别这一任务中达到了空前的精度。本文受这类算法的启发，设计双分支网络分别对视频数据和人体骨架数据进行特征提取并融合，使用深监督[18] 的方法对网络进行训练。

对于视频数据分支，本文采用 slow-fast 网络。slow-fast 是一种影响广泛的视频理解算法，采用两个网络分支分别对高帧率和低帧率的视频进行处理，分别捕获运动特征和图像特征，在每个卷积模块的结尾都将高帧率分支的特征图下采样，融合进入低帧率的特征图。对于人体骨架分支，本文采用图卷积神经网络对人体骨架数据进行特征提取，该网络的设计主要受非局部神经网络[19] 的启发，能够自适应地生成邻接矩阵。非局部神经网络是一种自注意力模块，在传统视觉任务中，它将每个像素点的特征看作所有像素点特征的加权之和。这种算法有助于高效获取长距

离特征的相关性。

深监督[18]的方法在人体姿态估计和目标检测等众多领域中得到了广泛的应用，近年来在动作识别领域也引起了研究者的关注。深监督是在训练过程中，在网络的中间层加入损失函数，提升网络中间层的特征表达能力，而在推理过程中，不考虑这些中间层的输出。在本文设计的网络中，深监督有助于提高两个分支特征的耦合性，以达到更好的训练效果。

本文的贡献主要体现在两个方面：首先是设计了一种基于人体骨架的动作识别方法，包括提升骨架数据特征显著性的方法和用于该任务的自注意力图卷积模块；除此之外，本文设计了一种将骨架数据和视频数据相融合的方法。

1　网络结构和原理

网络的整体结构如图 1 所示，分为两个分支，分别处理人体骨架数据和视频数据。训练的时候，每个卷积模块的输出特征，都将被全局平均池化，与对应的特征拼接起来送入全连接层，输出结果作为分类结果并计算损失函数。在测试的时候，仅将最后一层的输出作为结果。下面阐述网络细节。

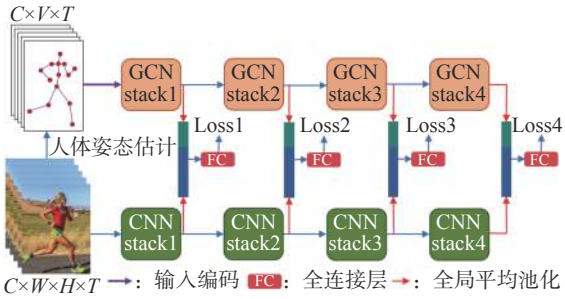

图 1　网络结构

Fig. 1　Model network architecture

1.1　人体骨架分支

人体骨架数据的获取方式是多样的，有的在数据集中给出，有的需要利用人体姿态估计算法得出，这部分内容将在实验部分详述。原始的人体关键点数据有三个维度 $\boldsymbol{P}\in \mathrm{R}^{3\times V\times T}$，其中 3 表示通道数；$V$ 表示人体关键点数量，由数据集决定；T 表示时间序列的长度，由数据的抽帧方式决定。

人体骨架分支主要包括输入编码和 GCN 模块，二者在算法中都起到了重要的作用。

在输入编码部分，网络需要先对人体骨架序列进行编码，提升人体骨架数据的表征能力，同时融入骨架数据的关键点和时间序列信息。

骨架序列的表示方式有两种，分别为关键点和骨骼表示法，如图 2 所示。关键点表示法就是用每个关键点的空间位置坐标作为节点特征。而在骨骼表示法中，首先将索引为 0 的关键点作为根节点，每个节点都被转换为一个表示骨骼连接方式的向量，该向量从上一个关键点指向当前关键点，根节点的骨骼表示指定为零向量。关键点表示法强调了人体关键点的空间位置这一特征，而骨骼表示法强调了这些特征的相关性，也就是人体骨架图的基本拓扑结构。为了便于计算，首先给定一个单位矩阵 $\boldsymbol{W}\in \mathbf{R}^{V\times V}$，然后将索引与有向连接节点相同的元素设置为−1。例如，在任意一帧中，对于 3D 的人体骨架数据，$\boldsymbol{p}_2$，$\boldsymbol{p}_1$ 两个关键点相连。有向边 $\boldsymbol{e}_2$ 的计算公式为 $\boldsymbol{e}_2=\boldsymbol{p}_2-\boldsymbol{p}_1=(x_2-x_1, y_2-y_1, z_2-z_1)^{\mathrm{T}}$，并且 $\boldsymbol{W}$ 中的元素 (2,1) 被设置为 −1。骨骼的表示是 $\boldsymbol{E}=\boldsymbol{P}\cdot\boldsymbol{W}$，$\boldsymbol{P}$ 和 $\boldsymbol{E}$ 具有相同的维度。然后将 $\boldsymbol{E}$ 和 $\boldsymbol{P}$ 在通道维度上拼接起来，作为网络输入：

$$\boldsymbol{I}=\mathrm{cat}(\boldsymbol{P}, \boldsymbol{P}\cdot\boldsymbol{W}) \tag{1}$$

式中 cat 表示在通道上进行的拼接操作，$\boldsymbol{I}\in \mathbf{R}^{6\times V\times T}$。

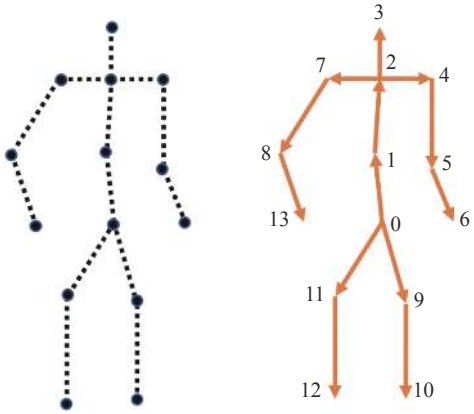

图 2　关键点表示方法（左）和骨骼表示方法（右）

Fig. 2　Representations of joints (left) and bones (right)

对于时间维度 T 上的每一个特征，计算它与前一个时刻的差值，作为基本的运动信息：

$$\boldsymbol{V}=\boldsymbol{I}(1:T)-\boldsymbol{I}(0:T-1) \tag{2}$$

此时 $\boldsymbol{I}\in\mathbf{R}^{6\times V\times(T-1)}$。然后用两个 1×1 卷积层，将通道数都扩张为 64，同时增强 $\boldsymbol{V}$ 和 $\boldsymbol{I}$ 的特征表达：

$$\tilde{\boldsymbol{I}}=\mathrm{ReLu}\left(\boldsymbol{W}_2\left(\mathrm{ReLu}\left(\boldsymbol{W}_1\boldsymbol{I}\right)\right)\right) \tag{3}$$

$$\tilde{\boldsymbol{V}}=\mathrm{ReLu}\left(\boldsymbol{W}_4\left(\mathrm{ReLu}\left(\boldsymbol{W}_3\boldsymbol{V}\right)\right)\right) \tag{4}$$

最后将两个部分的特征加以融合，得到强化的特征：

$$\boldsymbol{Z}=\tilde{\boldsymbol{V}}+\tilde{\boldsymbol{I}} \tag{5}$$

此时，$\boldsymbol{Z}\in\mathbf{R}^{64\times V\times T}$。增强了输入信号的表征能力之后，需要将关键点和时间序列信息融合到这部分信息中。这里采用 one-hot 编码方式分别对时间和空间索引的语义信息进行编码得到 $\boldsymbol{J}$ 和 $\boldsymbol{T}$，再采用与式(4)和式(5)相同的方式，即两个 1×1 卷积层对特征进行增强，得到 $\tilde{\boldsymbol{J}}\in\mathbf{R}^{64\times V\times T}$ 和 $\tilde{\boldsymbol{T}}\in\mathbf{R}^{128\times V\times T}$。最终融合到输入中：

$$\boldsymbol{Z}'=\mathrm{cat}\left(\boldsymbol{Z},\tilde{\boldsymbol{J}}\right)+\tilde{\boldsymbol{T}} \tag{6}$$

至此完成对输入的编码，通道数调整为 128。将编码后的输出传入自注意力图卷积模块，实现对骨架数据的特征提取。

对于图卷积模块，本文方法采用的是一种自适应图卷积网络。因为图的拓扑结构和边的权重值是自适应产生的，同时也可以看作一种自注意力机制。为了便于理解，先从自注意力机制进行描述。

注意力机制自从被提出就被广泛应用于各种应用场景，近年来也引起了广泛的关注。注意力机制的原理是将每一个特征看作所有特征的加权之和[20]，用公式可以简单表述为

$$\mathrm{Attention}(\mathrm{Query},\mathrm{Source})=\sum_{i=1}^{L_x}\mathrm{Similarity}(\mathrm{Query},\mathrm{Key}_i)\cdot\mathrm{Value}_i \tag{7}$$

式(7)中 Similarity 是计算相似度权重的函数，可以通过学习得到。当 Query 、 Key 和 Value 相同时，这个公式所代表的就是自注意力机制。本文所用到的注意力网络基于非局部网络[19]，并针对当前数据特点进行了改进，首先将非局部网络迁移到本任务中可以得到图 3 所示的实现方法。

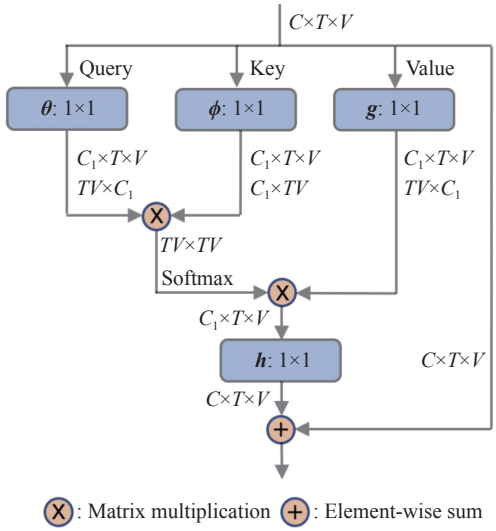

图 3　自注意力模块

Fig. 3　Self-attention block

在图 3 的描述中，输入是维度为 $C\times T\times V$ 的特征，同时被用作 Query 、 Key 和 Value 。

首先利用 $\boldsymbol{f}(x)$ 产生相似度矩阵的函数，对应式(7)中的 Similarity 函数，对应图 3 可以表示为

$$\boldsymbol{f}(x)=\mathrm{softmax}\left(\boldsymbol{\theta}(x)^{\mathrm{T}}\boldsymbol{\phi}(x)\right) \tag{8}$$

式中：$\boldsymbol{\theta}(x)$ 和 $\boldsymbol{\phi}(x)$ 对应于 Query 和 Key 分支上所作的 1×1 卷积操作和变形操作；上标 T 表示矩阵的转置操作。图 3 中的 C 要根据具体输入的特征通道数来决定。时空自注意力模块的具体计算方式可描述为

$$y(x)=\mathrm{ReLu}\left(\boldsymbol{h}\left(\boldsymbol{f}(x)\cdot\boldsymbol{g}(x)\right)+x\right) \tag{9}$$

式中 x 和 y 分别表示输入信号和输出。$\boldsymbol{g}(x)$ 采用 1×1 卷积通过增加通道数增强了 Value 分支的表征能力。$\boldsymbol{h}(x)$ 采用 1×1 卷积对输出的通道进行了调整，$+x$ 表示残差连接。

上述自注意力模块也可以理解为一种动态图卷积神经网络。在图模型中，边的连接方式用只含有 0 和 1 作为元素的邻接矩阵进行表示。对每条边加上连接的权重系数之后，邻接矩阵的形式就可以看作上述相似度矩阵，也就是 $\boldsymbol{f}(x)$ 产生了带有权重的邻接矩阵。本文产生的邻接矩阵和权重是通过自注意力网络动态生成的，故可以看作动态图卷积神经网络[9]。图也分为有向图和无向图，根据邻接矩阵是否对称可以判别。显然，上述注意力机制生成的邻接矩阵是非对称的。若

将 $\boldsymbol{\theta}(x)$ 和 $\boldsymbol{\phi}(x)$ 设置为相同的函数，矩阵乘以自身的转置将会得出对称矩阵。本文通过实验证明，这种方法减少了一个卷积计算层的计算量和参数量，但是对输出精度无影响。此时的相似度矩阵的计算方式变换为

$$\boldsymbol{f}(x)=\mathrm{softmax}\left(\boldsymbol{\theta}(x)^{\mathrm{T}}\boldsymbol{\theta}(x)\right) \tag{10}$$

同时，图 3 所示的自注意力模块的计算量过大，主要在于矩阵相乘，将矩阵的维度降低可以有效减少计算量。尝试将矩阵点乘的维度降低，减少计算量，故选用了空间自注意力模块，将对时间的建模部分添加到残差连接的分支上，用简单的 3×1 卷积加以实现，3 对应的是时间维度上的卷积。设计出图 4 所示的模型结构。通过实验对比可知，图 4 所示的自注意力模块在减少计算量的同时保证了更高的精度。

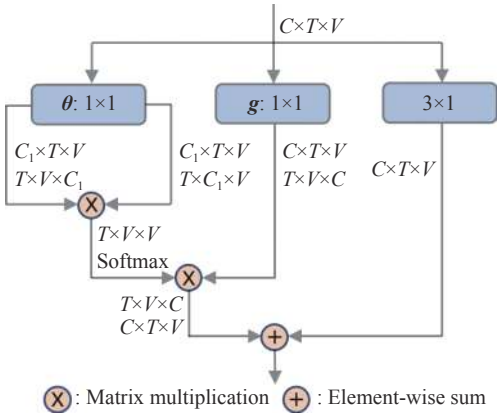

图 4　一种自注意力模块的变体

Fig. 4　A variant of spatial self-attention block

本文采用图 4 所示的网络结构，C_1 设置为模块特征的通道数的一半，以减少计算量，而模块的输出通道数的变化靠 $\boldsymbol{g}(x)$ 和残差连接中的 3×1 卷积来实现。对应于图 1 中的 GCN 模块的输出通道数分别设置为 128，256，256，512。

1.2　双分支的融合

视频分支采用的是 slow-fast 网络。该网络结构简单，如表 1 所示，包含两个分支，每个分支都包含四个 3D 残差网络模块。在每个残差网络模块结束的时候，采用大小为 12×5^2、步幅为 8×1^2 的 3D 卷积核将 fast 分支的时间维度降采样为原来的 1/8，通道数扩张 8 倍，这样既能保证特征数不变，同时达到和 slow 分支一样的特征维度。最后和 slow 分支的特征相加，实现将 fast 分支的特征向 slow 分支的融合。模块的输出如果进行下采样，则是在该模块的第一个卷积块中，将 1×3^2 的步幅设置为 1×2^2。

表 1　slow-fast 网络结构

Tab. 1　Model architecture of slow-fast network

Stage	Slow pathway	Fast pathway	Output sizes $T\times S^2$
Input data	—	—	64×224^2
Data layer	Stride 16, 12	Stride 2, 12	Slow : 4×224^2 Fast: 32×224^2
Conv1	1×7^2, 64 Stride 1, 2^2	5×7^2, 8 Stride 1, 2^2	Slow: 4×112^2 Fast: 32×112^2
Pool1	1×32 max Stride 1, 2^2	1×3^2 max Stride 1, 2^2	Slow: 4×56^2 Fast: 32×56^2
Res1 x 3	1×1^2, 64 1×3^2, 64 1×1^2, 256	3×1^2, 8 1×3^2, 8 1×1^2, 32	Slow :4×56^2 Fast :32×56^2
Res2 x 4	1×1^2, 128 1×3^2, 128 1×1^2, 512	3×1^2, 16 1×3^2, 16 1×1^2, 64	Slow: 4×28^2 Fast: 32×28^2
Res3 x 6	3×1^2, 256 1×3^2, 256 1×1^2, 1 024	3×1^2, 32 1×3^2, 32 1×1^2, 128	Slow: 4×14^2 Fast: 32×14^2
Res4 x 3	3×1^2, 512 1×3^2, 512 1×1^2, 2048	3×1^2, 64 1×3^2, 64 1×1^2, 256	Slow: 4×7^2 Fast: 32×7^2
	GAP	GAP	1×2 304

视频分支和人体骨架分支都具有 4 个大的卷积模块，在每个模块结束的时候，都进行 1 次监督，如图 1 所示。以两个分支的第 1 个模块举例，GCN 模块的输出进行全局平均池化之后输出 128 个特征，CNN 模块的两个分支在池化之后分别得到 256 和 32 个特征，全部堆叠起来有 416 个特征，最后经过全连接层后输出分类并计算损失。这里采用标签平滑损失函数，本质是在交叉熵损失函数的标签中加入噪声，防止因过度收敛而导致训练效果变差。

此时一共得出 4 个损失函数 Loss，通过系数 a（$a<1$）来控制 4 个损失函数的权重。

$$\mathrm{Loss}=\sum_{i=1}^{N}a^{N-i}\mathrm{Loss}_i \tag{11}$$

式中：N 为模块的数量，取 4；a 的具体取值将通过实验进行论证。训练结束后，只将最后一层的输出作为最后的推理结果，所以在训练的时候

这一层的权重值最高。对于中间监督层的损失，越是低级的特征，其权重值越低，按 a 的指数倍递减。

2　实验

2.1　数据集

NTU-RGBD60[4]：这是一个大规模动作识别数据集，包含 60 个动作类的 56 880 个数据，由 40 个不同的人执行这些动作，并由三个处于相同高度但不同水平角度−45°、0°、45°的 Kinect 相机捕获。数据集提供深度信息、3D 骨骼信息、RGB 帧以及红外序列。对于 Cross-View(CV) 设置，来自两个摄像头的数据用于训练，而其他摄像头的数据用于测试。对于 Cross-Subject(CS) 设置，选用来自 20 个人的 40 320 个人体关键点序列用于训练，其余用于测试。在实验过程中，本文随机从 CS 和 CV 两种不同设置的训练集里面，选择 10% 用作验证集。

NTU-RGBD120[21]：它是 NTU-RGBD60 数据集的扩展。它包含 120 个动作类的 114 480 个数据，由 106 个不同的人执行这些动作，每一个数据设置都包含一个 ID 序号。对于 Cross-Subject 设置，一半执行者的数据用于训练，而其他人的数据用于测试。在 Cross-Setup 设置中，根据 ID 序号对数据进行划分，分别用于训练和测试，同样，本文从训练集中选取 10% 作为验证集。

Kinetics Skeleton 400[13]：这是一个包含 240 000 个训练和 20 000 个测试的大规模动作识别数据集，类别数达到 400。原本的 Kinetics 400 数据集仅包含视频数据集[22],采用 OpenPose 姿态估计算法[23] 工具包对视频数据中的每一帧提取人体骨架数据。每个骨架图包含 18 个主要关节，每个关节用 (X, Y, C) 表示，其中 (X, Y) 是像素坐标系中的二维坐标，C 是 OpenPose 算法给出的置信度分数。对于视频中含有多人的情况，在每个视频中选择关节点平均置信度得分最高的两个人作为被选取对象进行动作识别。

2.2　实验细节

数据处理。首先，将每个视频数据按时间维度平均分割成 64 个片段，从每个片段中随机选取一帧，得到 64 帧，视频分辨率降采样为 224×224 作为视频分支的输入，并对每一帧提取骨架数据作为总的骨架分支的输入。基于第一帧的人体关键点坐标位置，保持初始位置不变，每一帧的关键点都转化为该位置的相对坐标。如果一帧包含两个人，则通过使每一帧包含一个关键点序列将这一帧分成两帧。对于数据增强方式，视频和关键点数据均被随机按照某个角度进行旋转。随机在 $[-17°, 17°]$ 之间生成两个角度，分别作为 X、Y、Z 轴的旋转角度。特别的是，在 NTU-RGBD60 的 Cross-View(CV) 设置中，角度选择在 $[-30°, 30°]$ 区间，因为该数据集的视角变化很大。对于 Kinetics 400 数据集，未使用增强功能。

训练细节。所有的工作都是在 4 张 A30 显卡上实现。实验过程中，视频分支采用 Kinetics 400 数据集上的预训练权重，骨架数据分支采用随机初始化参数。本文采用 Adam 优化器并将初始学习率设置为 0.001。网络训练 120 个 epoch，并分别在 60、90 和 110 个 epoch 时将学习率衰减 10 倍 epoch。权重衰减设置为 0.000 1。每个数据集的 batch 设置为 64。使用标签平滑损失函数，平滑因子设为 0.1。

2.3　消融分析

本文提出的方法包含两个分支，这两个分支分别单独作用时都能达到很好的效果。本部分实验的数据均为 NTU-RGBD60 数据集上的结果，这主要考虑到该数据集的人体骨架数据的可信度。Kinetics 数据集的骨架数据是采用 OpenPose 算法得出，人体姿态估计算法对实验结果影响大。NTU-RGBD60 数据集给出的人体骨架数据精度高，使得这部分研究更加合理。

基于人体骨架的动作识别近些年来发展迅猛，本文所设计的骨架分支单独工作时，也具有一定的先进性。先对这一分支的效果进行单独分析。

在对人体骨架分支的输入进行编码的过程中，本文采用两种不同的表示法对输入进行编码，表 2 所示为不同的输入对结果的影响。J 表示仅人体关键点作为输入；B 表示仅用骨骼向量作为输入；2-stream 表示目前比较流行的双流输入法[8]，由两个完全相同的并行分支构成网络；

两个分支的输入分别是 J+B，表示本文采用的两种输入拼接后作为输入的方法，也是本文设计的方法。

表 2　不同数据表示方法参数量和精度对比
Tab. 2　Comparison of parameters and accuracies between different representations

Input	Parameter amount/10^6	Accuracy/%	
		CS	CV
J	0.89	89.1	94.9
B	0.89	87.4	94.9
2-stream	1.78	90.5	96.0
J + B (proposed)	0.89	90.6	96.0

实际上，本文提出的方法相比于单个关键点或者骨骼向量作为输入，只增加了大约 7000 个参数，这是完全可以忽略不计的参数量。相比于只用某一个流作为输入的方法，2-stream 方法有效提升了精度，却也带来了两倍的参数和计算量。就此而言，本文所提出的方法中，两种表示方法的低级特征进行融合，既没有带来更多的参数量，同时达到了更高的精度，优于现有的方法。

GCN 模块堆叠数量的影响如表 3 所示。当块堆叠得越深时，性能越好。但是当 N 大于 4 并开始过度拟合训练集时，在测试集上的性能似乎没有再提升，甚至略微下降。这可能是模型变得过于复杂，导致了过拟合问题。

由于视频分支采用的是 slow-fast 算法，该方法的研究者给出了详细的报告，故本文不对这部分作过多分析，而将重点放在两分支融合的情况。各阶段损失函数的权重值 a 对精度的影响如表 4 所示，在 a 取值为 1/3 时达到最高精度。

表 3　GCN 模块堆叠数量变化的精度对比
Tab. 3　Accuracy comparison between different number of stacked GCN blocks

a	Accuracy/%	
	CS	CV
2	73.0	80.0
3	86.5	90.3
4	90.6	96.0
5	90.6	95.9

表 4　a 的取值导致的精度变化
Tab. 4　Accuracy change caused by the value of a

a	Accuracy/%	
	CS	CV
1	94.9	98.1
1/2	95.4	98.6
1/3	**95.6**	**99.0**
1/4	95.3	99.0
1/5	95.3	98.7

两个分支的实际结果如表 5 所示，这里 a 取值为 1/3。对于表中 Skeleton 和 RGB 数据的两行，本文先对整个网络进行训练，然后将两个分支拆开，分别对两个分支的最后一个全连接层进行训练。两个分支完全各自训练的方式与本文方法相比，精度都有所下降。这是因为在训练过程中，两个分支的中间特征都互相耦合以达到高的精度，也只有在两个分支合并起来工作时，才能达到更高的精度。

表 5　双分支网络的精度对比
Tab. 5　Accuracy comparison of dual branch network

Branch	Accuracy/%	
	CS	CV
Skeleton	90.2	94.7
RGB	82.3	90.1
Skeleton+RGB	95.6	99.0

2.4　与其他方法的对比

这部分工作主要关注两方面，分别是人体骨架单分支和双分支协作的精度。最终模型与许多最先进的动作识别方法进行了比较,主要是基于人体骨架和人体骨架&视频数据。在表 6 中，将精度和参数量与许多有影响力的方法进行了比较。有些论文中直接给出了一些数据，但是没有提供计算量的数据。本文结果是使用 ptfloaps 库和源代码计算获得的。表 6、表 7 和表 8 分别列出了本文的方法与其他方法在 NTU-RGBD60、NTU-

表 6　NTU-RGBD60 数据集精度对比
Tab. 6　Comparison of the accuracy and parameters on NTU-RGBD60 dataset

Algorithm	Parameter amount/10^6	Accuracy/%	
		CS	CV
AS-GCN[12]	7.40	86.8	94.2
GR-GCN[11]	—	84.8	92.4
2s-AGCN[8]	6.92	88.5	95.1
AGC-LSTM[14]	22.81	89.2	95.0
2s-SDGCN[18]	—	89.6	95.7
SGN[15]	0.69	89.0	94.5
DGNN[24]	8.16	89.9	96.1
Shift-GCN(2s)[10]	1.48	89.7	96.0
Shift-GCN(4s)[10]	2.94	90.7	96.5
MS-G3D(Joint)[6]	3.20	89.4	95.0
MS-G3D(2s)[6]	6.40	91.5	96.2
MST-GCN[1]	2.03	91.7	95.7
Hierarchical Action	46.8	95.3	98.3
Skeleton	2.85	90.6	96.0
Skeleton + RGB	33.7	**95.6**	**99.0**

表 7　NTU-RGBD120 数据集精度对比
Tab. 7　Comparison of the accuracy on NTU-RGBD120 dataset

Algorithm	Accuracy/%	
	c-sub	c-set
ST-LSTM[25]	55.7	57.9
GCA-LSTM[26]	58.3	59.2
Pose Evolution Map[27]	64.6	66.9
2s-AGCN[8]	82.5	84.9
Shift-GCN[10]	85.9	87.6
MS-G3D[6]	86.9	88.4
SGN[15]	79.2	81.5
MST-GCN[1]	88.5	87.8
Hierarchical Action	93.7	94.5
Skeleton	84.5	85.6
Skeleton + RGB	94.7	95.2

表 8　Kinetics 400 骨架数据集精度对比
Tab. 8　Comparison of the accuracy on Kinetics 400 skeleton dataset

Algorithm	Accuracy/%	
	Top 1	Top 5
ST-GCN[13]	30.7	52.8
AS-GCN[12]	34.8	56.5
2s-AGCN[8]	36.1	58.7
DGNN[25]	36.9	59.6
MS-AAGCN[7]	37.8	61.0
MS-G3D[6]	38.0	60.9
Slow-fast[5]	75.6	92.1
Skeleton	37.6	60.1
Skeleton+RGB	78.1	93.3

RGBD120 和 Kinetics 400 数据集上的精度对比。本文算法以较少的参数和较低的计算成本达到了具有竞争力的精度。

3　结　论

在动作识别这一领域，主要有基于视频和基于人体骨架两种任务，以往的大多数方法是将两者作为两种不同的领域进行研究。将视频数据和人体骨架数据作为两种不同的数据模态，以合理的方式进行融合，可以达到很好的效果。同时，针对基于人体骨架的动作识别这一任务，输入数据的编码方式和图结构设计起着决定性的作用。根据骨架数据的不同形式，以及人体骨架图的特点，加入自注意力机制对网络结构进行设计，可以有效提升网络的效率。

相对于视频数据，图模型的实例更加广泛，如蛋白质结构等，本文所设计的图卷积神经网络在这些领域的应用可以开展进一步研究。同时，本网络设计的方法是一种将图神经网络和卷积神经网络相结合的方法，在多模态数据融合方面有一定启发性意义。

参考文献：

[1] CHEN Z, LI S, YANG B, et al. Multi-scale spatial tem-

poral graph convolutional network for skeleton-based action recognition[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(2): 1113 – 1122.

[2] DU Y, WANG W, WANG L. Hierarchical recurrent neural network for skeleton based action recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston: IEEE, 2015: 1110-1118.

[3] POPPE R. A survey on vision-based human action recognition[J]. Image and Vision Computing, 2010, 28(6): 976 – 990.

[4] SHAHROUDY A, LIU J, NG T T, et al. NTU RGB+D: A large scale dataset for 3D human activity analysis[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, 2016: 1010 − 1019.

[5] FEICHTENHOFER C, FAN H Q, MALIK J, et al. Slowfast networks for video recognition[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019: 6201 − 6210.

[6] LIU Z, ZHANG H W, CHEN Z H, et al. Disentangling and unifying graph convolutions for skeleton-based action recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA, USA: IEEE, 2020: 140 − 149.

[7] SHI L, ZHANG Y F, CHENG J, et al. Skeleton-based action recognition with multi-stream adaptive graph convolutional networks[J]. IEEE Transactions on Image Processing, 2020, 29: 9532 – 9545.

[8] SHI L, ZHANG Y F, CHENG J, et al. Two-stream adaptive graph convolutional networks for skeleton-based action recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA: IEEE, 2019: 12028 − 12037.

[9] YE F F, PU S L, ZHONG Q Y, et al. Dynamic GCN: context-enriched topology learning for skeleton-based action recognition[C]//Proceedings of the 28th ACM International Conference on Multimedia. New York, NY, USA: Association for Computing Machinery, 2020: 55 − 63.

[10] CHENG K, ZHANG Y F, HE X Y, et al. Skeleton-based action recognition with shift graph convolutional network[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA, USA: IEEE, 2020: 180 − 189.

[11] GAO X, HU W, TANG J X, et al. Optimized skeleton-based action recognition via sparsified graph regression[C]//Proceedings of the 27th ACM International Conference on Multimedia. New York, NY, USA: Association for Computing Machinery, 2019: 601 − 610.

[12] LI M S, CHEN S H, CHEN X, et al. Actional-structural graph convolutional networks for skeleton-based action recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA: IEEE, 2019: 3590 − 3598.

[13] YAN S J, XIONG Y J, LIN D H. Spatial temporal graph convolutional networks for skeleton-based action recognition[C]//The Thirty-Second AAAI Conference on Artificial Intelligence. New Orleans: AAAI, 2018.

[14] SI C Y, CHEN W T, WANG W, et al. An attention enhanced graph convolutional LSTM network for skeleton-based action recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA: IEEE, 2019: 1227 − 1236.

[15] ZHANG P F, LAN C L, ZENG W J, et al. Semantics-guided neural networks for efficient skeleton-based human action recognition[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA, USA: IEEE, 2020: 1109 − 1118.

[16] ZHAO R, WANG K, SU H, et al. Bayesian graph convolution LSTM for skeleton based action recognition[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. Seoul: IEEE, 2019: 6881 − 6891.

[17] WU C, WU X J, KITTLER J. Spatial residual layer and dense connection block enhanced spatial temporal graph convolutional network for skeleton-based action recognition[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. Seoul: IEEE, 2019: 1740 − 1748.

[18] LEE C Y, XIE S, GALLAGHER P, et al. Deeply-supervised nets[C]//Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics. Lille, France: PMLR, 2015: 562 − 570.

[19] WANG X L, GIRSHICK R, GUPTA A, et al. Non-local neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA: IEEE, 2018: 7794 − 7803.

[20] VASWANI A, SHAZEER N, PARMAR N, et al. At-

tention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY, USA: Curran Associates Inc., 2017: 6000 − 6010.

[21] LIU J, SHAHROUDY A, PEREZ M, et al. NTU RGB + D 120: a large-scale benchmark for 3D human activity understanding[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(10): 2684 − 2701.

[22] KAY W, CARREIRA J, SIMONYAN K, et al. The kinetics human action video dataset[J]. arXiv: 1705.06950, 2017.

[23] CAO Z, SIMON T, WEI S E, et al. Realtime multi-person 2D pose estimation using part affinity fields[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017: 1302 − 1310.

[24] SHI L, ZHANG Y F, CHENG J, et al. Skeleton-based action recognition with directed graph neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, CA, USA: IEEE, 2019: 7904 − 7913.

[25] LIU J, SHAHROUDY A, XU D, et al. Spatio-temporal LSTM with trust gates for 3D Human action recognition[C]//14th European Conference on Computer Vision. Amsterdam, The Netherlands: Springer, 2016: 816 − 833.

[26] LIU J, WANG G, HU P, et al. Global context-aware attention LSTM networks for 3D action recognition[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, HI, USA: IEEE, 2017: 3671 − 3680.

[27] LIU M Y, YUAN J S. Recognizing human actions as the evolution of pose estimation maps[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, UT, USA: IEEE, 2018: 1159 − 1168.

（编辑：张　磊）