# 基于多特征融合的小样本视频行为识别算法

蒲瞻星[1),2)] 葛永新[1)]

[1)](重庆大学大数据与软件学院 重庆 401331)

[2)](北京理工大学计算机学院 北京 100081)

**摘 要** 现有基于小样本学习的视频行为识别方法,在解决小样本学习中信息量稀缺问题时存在信息重复度高以及类间相似性大等不足,而且鲜有关注小样本学习中的域偏移与枢纽点问题,从而导致动作类表达能力弱和行为识别中错误分类的问题,此外,复杂的网络结构导致参数量与计算量成倍增加.针对以上问题,本文提出一种基于多特征融合的小样本视频行为识别算法,具体来说,该方法提出深度特征与流形特征的融合策略.首先,针对特征形式之一的流形特征,提出使用表征传播对流形结构进行平滑操作,更好地缓解了小样本学习中的域偏移与枢纽点问题.其次,通过同时使用对视频特征表达能力不同的深度特征与流形特征,获得更多的样本有效信息,进而缓解小样本学习中样本稀缺的问题.最后,为减小模型的参数量与计算量,选择基于2D方法构建模型.在HMDB51、UCF101以及Kinetics三个数据集上进行实验,结果表明,本文方法在"5-way 1-shot"任务下表现突出,识别率优于现有的小样本视频行为识别方法,在HMDB51上提高了8.5%,在UCF101上提高了9.5%,在Kinetics上提高了1.0%.

**关键词** 小样本学习;行为识别;视频分类;数据的流形分布;多特征融合

**中图法分类号** TP391 **DOI号** 10.11897/SP.J.1016.2023.00594

# Few-Shot Action Recognition in Video Based on Multi-Feature Fusion

PU Zhan-Xing[1),2)] GE Yong-Xin[1)]

[1)](*School of Big Data & Software Engineering, Chongqing University, Chongqing* 401331)

[2)](*School of Computer Science and Technology, Beijing Institute of Technology, Beijing* 100081)

**Abstract** Few-shot learning means knowing some unseen classes by learning a few samples of unseen classes with some prior knowledge of seen classes. In the field of action recognition in video, the difficulty of collecting and annotating a large amounts of video data has made many scholars more interested in few-shot learning. In recent years, studies of few-shot learning with images have been made great progress. But in the field of videos, it poses a great challenge to few-shot action recognition because videos are more complex than images and data information is relatively scare in few-shot learning. In this paper, we focus on the task of few-shot action recognition in video from the following aspects. First of all, discriminative information is not sufficient due to lacking of support set data in few-shot learning. Secondly, the classes of training set and test set are disjoint, so their data distribution may be very different. This problem is called domain shift, which usually leads to poor effect in generalization. In addition, the change of dimension in data features would lead to the phenomenon that "some unrelated points become the closest point of most points" when using deep learning in few-shot learning, which may cause hubness. Various existing models of few-shot action recognition in video have achieved high repeatability of information and increased similarity between different classes when solving the scarcity

收稿日期:2021-07-14;在线发布日期:2022-02-16.本课题得到国家自然科学基金(62176031,61772093)和中央高校基本科研业务费(2021CDJQY-018)资助.**蒲瞻星**,硕士研究生,主要研究领域为计算机视觉和机器学习.E-mail:20171745@cqu.edu.cn.**葛永新**(**通信作者**),博士,副教授,中国计算机学会(CCF)会员,主要研究领域为计算机视觉、机器学习和大数据智能.E-mail:yongxinge@cqu.edu.cn.

of information in few-shot learning, and they rarely pay attention to domain shift and hubness in few-shot learning, which leads to weak expression ability of action classes and false action recognition. At the same time, because of the complex network, the model has a great number of parameters and FLOPs(floating point operations). Considering the difficulties in few-shot learning and the deficiencies of previous methods, we propose a new method for few-shot action recognition in video by combining deep feature and manifold feature of data. Firstly, we aim at the manifold, which is one type of the chosen features, and can be utilized to effectively maintain the structure of data and retain more information of data. Moreover, we propose to use embedding propagation to smooth the manifold, so as to better alleviate domain shift and hubness in few-shot learning. Secondly, fusing the deep feature and manifold feature increases the effective information of samples, thereby alleviating the scarcity of information. The specific operation of multi-feature fusion is to perform a weighted sum for the label prediction scores corresponding to deep features and manifold feature, so as to obtain a more effective label prediction score. Finally, in order to reduce the number of parameters and FLOPs of the model, we build the model based on 2D method and strive to improve the performance of the model. Experiments on the HMDB51, UCF101 and Kinetics datasets show that the recognition accuracies of the proposed method are higher than those of the existing few-shot action recognition methods under the task of "5-way 1-shot", which are improved by 8.5% on HMDB51, 9.5% on UCF101 and 1.0% on Kinetics. The proposed model successfully applies the idea of few-shot learning to action recognition in video, so it can be quickly adapted to new dataset and achieve effective performance in action recognition depending on limited samples.

**Keywords**　few-shot learning; action recognition; video classification; manifold; multi-feature fusion

## 1　引　言

人工智能时代的来临,使得行为识别[1]作为智能家居[2]、智能监控[3]、人机交互[4]等领域的一项重要基础技术,得到了广泛应用.如今随着通讯技术的发展,网络传输速度越来越快,网络视频用户规模也在逐年增加,视频创作者越来越多,直播行业也愈发火热,行为识别技术将给这些领域带来便利,例如将行为识别技术应用到视频审核以及直播监管当中.此外,虚拟现实、自动驾驶等前沿技术也随着 5G 的应用得以进一步发展,这些技术的继续发展也少不了行为识别技术的支持.

传统的视频行为识别方法依赖大规模且强标记的视频数据集,但是这些视频数据集的获取需要花费大量的人力和物力,难度较大.在现实生活中,很多场景下的视频行为识别任务中带有标注的数据是明显不足的,例如安全领域.因此,如何降低视频行为识别对数据集的要求显得尤为重要,基于小样本学习[5]的视频行为识别研究也随之应运而生.人类具有很强的知识迁移能力,例如一个小朋友在了解体育项目时,他只需要浏览几个视频就能区分跳远与跳高两种动作,这就是小样本学习在现实生活中的直观体现.近几年,图像方面的小样本学习研究取得了不少进展,但在视频方面,由于视频比图片多了时序信息,处理起来更为复杂困难,这给小样本视频行为识别提出了不小的挑战.

目前小样本视频行为识别中主要存在以下难点:(1)信息量不足问题,指小样本学习中的模型迁移到新数据集时,可供学习的样本数量非常少,进而导致判别信息不足;(2)域偏移[6]问题,是指在更换数据集或数据模态后,造成模型的认知偏差;(3)枢纽点[7]问题,其描述是某个点会成为大多数点的最邻近点,即使该点与其近邻点无关.

针对小样本视频行为识别中的难点,研究人员从多个方面进行探究.以信息量不足问题为出发点,现有方法的策略是增加数据集样本数量和获取视频时序信息.(1)增加数据集样本数量:为了增加小样本学习中支持集样本的数量,Fu 等人[8]提出一种扩充数据集的办法.该方法从每个类中先选出部分视

频,并分割成一些小片段,然后将类中剩余的视频与这些片段进行匹配,匹配成功后替换掉原视频中等长的视频片段,从而生成新的增强视频,因为被替换的视频片段非常短,所以可以认为新视频的标签保持不变;(2)获取时序信息,首先是 3D 特征提取器[9-10]的出现,通过同时对多张视频帧图像进行卷积操作提取视频数据中的时序特征.之后为了更加有效地利用时序信息,提出了时序信息对齐[11]与时序信息增强[12]的方法.此外,还有方法致力于提升模型的效率,旨在构建更为有效的映射模型,如 CMN[13](Compound Memory Network,复合记忆网络),它是一种基于 2D 方法的模型,查询集样本将在记忆网络中搜索最邻近类别作为分类结果.

虽然以上方法在小样本视频行为识别上取得了一定的进展,但其方法仍存在不足之处.首先是增加数据集信息量的方法,尽管单个类别的信息总量得到了提升,但存在过多冗余信息,同时数据增强的效果也十分有限,真正得到增强的视频样本数量有限.其次,通过使用 3D 特征提取器获取时序信息,模型的网络复杂度大大提升,且要面临巨大的计算压力和资源消耗.CMN 使用的是 2D 特征提取器,极大减小了参数量与运算量,但样本数量少导致记忆网络中存储的类原型判别性较弱,从而影响分类精度.但值得注意的是 CMN 在没有使用增加数据集信息量策略的情况下,依然取得了不错的效果,间接地说明 2D 方法有不错的前景.此外,不难发现以往的方法缺少对域偏移[6]与枢纽点[7]问题的关注,而且有研究已经证明了通过对数据的流形结构进行平滑操作可以平滑决策边界和提升模型对噪声的鲁棒性,从而缓解域偏移问题,同时经过平滑操作后能更好地保持数据的流形结构,进而保持数据的判别性以缓解枢纽点问题.在文献[14]中证明了表征传播具有平滑流形结构的作用,表征传播的效果就是基于插值的思想将样本的特征表示为其在流形结构图中近邻点的加权和.

针对小样本视频行为识别中的难点以及现有方法的不足,本文提出融合多种形式的特征来增加信息量,同时使用表征传播构建平滑的流形结构来缓解域偏移与枢纽点问题,且基于 2D 方法构建模型.本文主要贡献包含以下三个方面:

(1)结合多特征融合的思想,提出了一种基于深度特征与流形特征的 2D 小样本视频行为识别模型.该模型通过使用多种形式的特征增加视频样本的信息量,其中,深度特征能反应视频数据的本质特征,流形特征能缓解域偏移与枢纽点问题,且这两种特征间冗余信息少,能避免引入信息重复度高和类间相似性大的问题,将它们融合后能实现优势互补,从而提升视频信息的判别性;

(2)针对选取的特征形式中的流形特征,使用表征传播完成对数据流形结构的平滑,以此构建更加稳定的流形结构,从而更好地缓解小样本学习中的域偏移与枢纽点问题,最终获得性能更出色的小样本视频行为识别模型;

(3)通过实验,探究深度特征与流形特征对应预测分数的最佳融合方案,寻求最佳的权重分配.对不同分类器的性能进行比较,以找出最适合本文方法的分类器,同时证明表征传播构建的平滑流形结构能提升模型性能.

## 2 相关工作

小样本视频行为识别任务中涉及两个关键任务,即小样本学习与视频分类,同时本文在多特征融合中引入了数据的流形分布,它们共同组成了本文行为识别方法的三大要素.

### 2.1 小样本学习

小样本学习凭借其在有限数据情况下能快速适应新数据并且进行有效识别的优势,近几年受到了越来越多的关注.基于度量的小样本学习是目前的主流方案,其主要方法包括匹配网络[15]、孪生网络[16]、原型网络[17]和关系网络[18].基于模型的方法希望在模型的结构设计上取得优势,提高模型的效率,其中 Santoro 等人[19]提出的网络借鉴了神经图灵机的思想,通过引入额外的记忆模块,使得模型对样本类进行记忆增强.基于优化的小样本学习[20]方法相对来说略少,它们的做法是通过调整模型的优化方案来完成小样本分类任务,因为该方法认为普通的梯度下降方法在小样本学习场景下难以拟合.此外,有方法的出发点是扩充数据集样本,如 SalNet[21],它将不同图像的前景与背景进行拼接,从而获取数量可观的合成图像.

### 2.2 视频分类

为了增加视频数据的信息量,Simonyan 等人[22]提出双流卷积网络.该网络结构分别将 RGB 图像与光流图像输入到两个卷积神经网络中,使得识别过程兼顾了视频的空间流与时间流.此后,TSN(Temporal Segment Network)[23]也在 RGB 图像与光流图像上使用了双流卷积神经网络,同时提出了

稀疏采样策略.CMN 引入了一种显著性嵌入算法,将视频数据编码为固定大小的矩阵表示,利用提出的复合记忆网络来存储类原型,通过匹配和排列对查询集样本进行分类.上面提到的方法仅利用了视频数据空间维度上的信息,而视频比图像多了时间维度,因此,C3D[9]提出了一个 3D 卷积核,其能提取时间与空间维度上的信息.之后 I3D[10]将双流卷积与 3D 卷积结合起来,也有学者基于 I3D 使用 LSTM[24].但值得注意的是,3D 方法的计算量是比较高的,尤其是使用光流图像后,计算量更为巨大,这会极大地增加模型训练所需的时间以及对设备性能的要求.

### 2.3　数据的流形分布

在使用简单的数据流形时,有的学者通过欧氏距离和余弦距离来构建多个类原型之间的结构图,该构建方法的解释性较强,能很直观地反应两个类别之间的相似性,具体表现就是研究者们构建了各式各样的正则化方法.Xu 等人[25]基于映射模型,通过在训练集和测试集数据之间共同构建近邻图,实现流形正则化约束.也有学者用多个流形正则化来进行多层约束,如在[26]中使用了两个流形正则化,分别用来维持数据映射过程中在视觉特征空间与属性空间中的数据结构.但简单的数据流形结构在局限性,它极有可能忽略了数据间存在的更为复杂的几何分布.

考虑到简单数据流形结构的不足,有的学者开始探究数据中存在的更为复杂的数据流形分布.在 Fu 等人[27]的工作中,相较于其之前工作[28]中构建的简单的单近邻图,他们在此基础上使用数据的特征表示构建起了超图,进而可以更加准确的描述存在多元关系的对象之间的联系.有的学者考虑的则是嵌入空间中存在更为丰富的流形结构[29],他们使用类标签图对嵌入空间中的流形结构进行建模,度量方法采用了吸收马尔可夫链过程(AMP,absorbing Markov chain process).使用复杂流形结构的效果可能更好,但过程复杂且可解释性下降.

## 3　模型结构

本文的模型整体框架如图 1 所示,该模型分为预训练阶段、微调阶段和测试阶段.在对查询集样本进行标签预测时,首先由特征提取器提取动作视频的特征,之后模型分别进行基于深度特征与流形特征的标签预测,进而得到两个标签预测分数,再将这两个分数进行加权求和,完成多特征融合操作,得到最终的标签预测分数.

### 3.1　问题设置

在进入正式介绍之前,需要了解问题相关的设定.本文提出的模型在预训练阶段按照传统的训练模式进行训练,但在微调阶段时,训练的模式有所变化,目的是让模型在小样本学习任务下拥有更好的性能表现.

小样本视频行为识别任务中的基本设置如下:首先将数据集的类别划分为三部分,分别是训练类 $c_{train}$ 、验证类 $c_{val}$ 和测试类 $c_{test}$ ,其对应有训练数据集 $D_{train}=\{(v_i,c_i),c_i\in c_{train}\}$ 、验证数据集 $D_{val}=\{(v_i,c_i),c_i\in c_{val}\}$ 以及测试数据集 $D_{test}=\{(v_i,c_i),c_i\in c_{test}\}$ .识别算法在 $D_{train}$ 上进行训练,在 $D_{val}$ 上进行验证,最后在 $D_{test}$ 上完成测试,其中验证集是为了知晓模型在训练过程中的性能,进而调整模型训练时的一些参数,而测试集则用来评估模型的最终性能.需要注意的是以上三个数据集两两之间交集均为空集.在小样本学习中,定义 episode 为训练单位,它包含支持集和查询集,而小样本指的就是支持集样本少.小样本学习中的“N-way K-shot”指的是支持集有 N 个类以及每个类有 K 个样本,在测试的时候,这 N 个类对于模型来说是全新的类,模型需要依靠支持集中的 $N\times K$ 个样本完成对查询集样本归属类别的预测.

### 3.2　方法介绍

#### 3.2.1　特征提取器

在特征提取器的选择上,模型使用了比较主流的残差神经网络[30](ResNet,Residual Neural Network),具体使用的是网络深度为 18 层的 ResNet18.在使用神经网络进行特征提取之前,由于视频数据中冗余信息较多且整个视频的数据量过大,因此需要对视频数据进行采样处理,本文采用的是和 TSN[23]中一致的稀疏时间采样策略.

特征提取器 ResNet18 是基于 2D 方法的,相比于一些 3D 方法,尽管它因为缺少对时序信息的获取而导致特征提取性能相比之下没有那么强,但是它对于算力的要求较低,模型的复杂度也更低.使用基于 Kinetics 数据集的预训练模型在 HMDB51 和 UCF101 数据集上进行训练,结果如表 1 所示,基于 2D 方法的特征提取器的计算量比 3D 方法少得多,而且准确率表现也普遍不错,更加凸显其“性价比”.在 UCF101 数据集上使用 2D 和 3D 的 ResNet 进行直接训练,其结果如表 2 所示,从结果来看,3D 残差

神经网络的表现更好，但是从表中的结果对比可以看出，3D 网络对性能的提升十分有限，同时网络深度的增加对于分类准确率的提升也十分有限，因为复杂的网络伴随更多的参数，训练的难度也就更大. 在权衡分类性能和资源消耗后，本文选择了极具“性价比”的 ResNet18-2D 卷积神经网络作为特征提取器，其在极小的资源消耗代价下，获得了不错的分类效果.

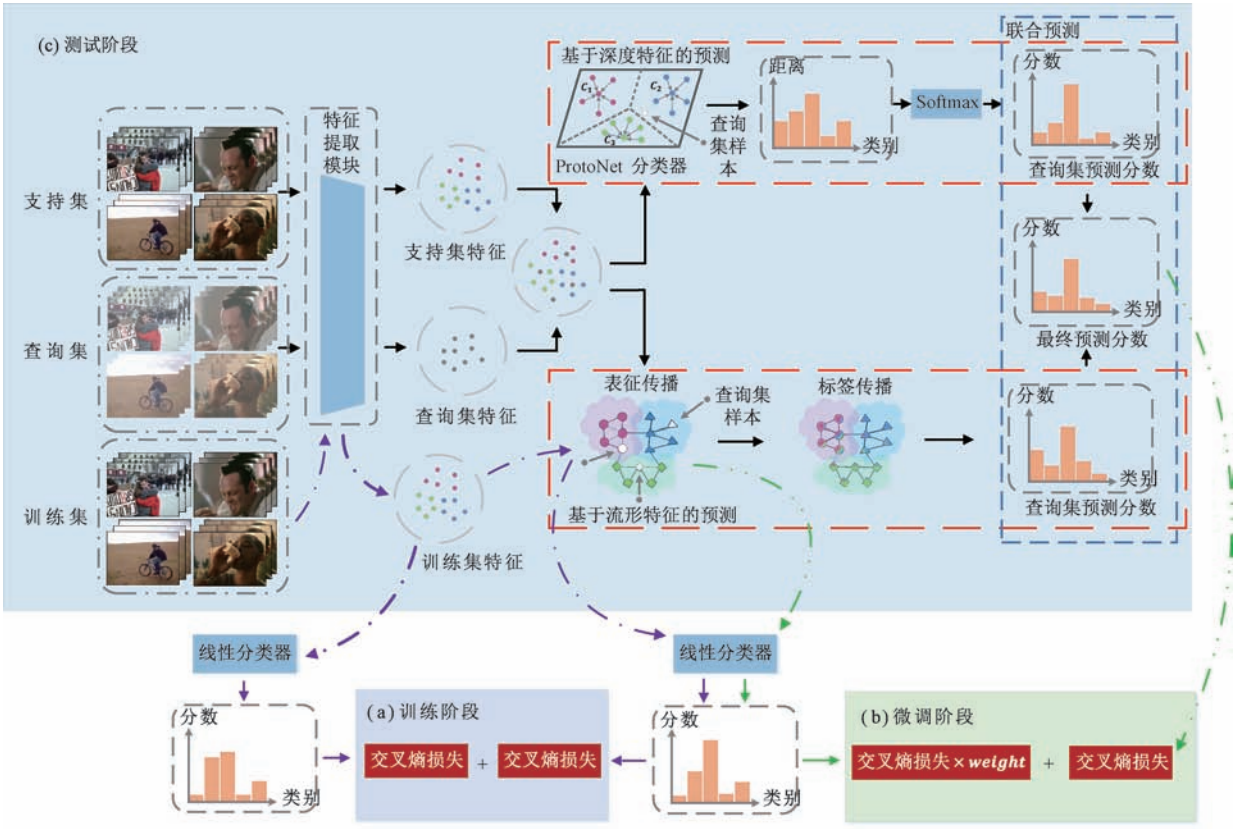


图 1 基于深度特征与流形特征的小样本视频行为识别模型

**表 1 不同特征提取方法的计算量及其预训练模型(基于 Kinetics 训练)在 HMDB51 和 UCF101 上微调后的分类准确率**

| 方法 | 特征维度 | Backbone | FLOPs | HMDB51 | UCF101 |
|---|---|---|---|---|---|
| ResNet18 | 2D | ResNet18 | 1.8G | 56.4% | 84.4% |
| Two-Stream CNN[22] | 2D | ConvNets | 3.3G | 59.4% | 88.0% |
| ResNet50 | 2D | ResNet50 | 4.1G | 61.0% | 89.3% |
| ResNet101 | 2D | ResNet101 | 7.6G | 61.7% | 88.9% |
| C3D[9] | 3D | ResNet18 | 38.5G | 62.1% | 89.8% |
| I3D-RGB[10] | 3D | Inception v1 | 107.9G | 74.8% | 95.6% |
| R(2+1)D-RGB[31] | 3D | ResNet50 | 152.4G | 74.5% | 96.8% |

**表 2 不同残差网络在 UCF101 数据集上的比较**

| 方法 | 特征维度 | 参数量 | FLOPs | 准确率 |
|---|---|---|---|---|
| ResNet18 | 2D | $11.2\times10^6$ | 1.8G | 42.2% |
| ResNet34 | 2D | $21.5\times10^6$ | 3.5G | 42.2% |
| ResNet18 | 3D | $33.2\times10^6$ | 19.3G | 45.6% |
| ResNet34 | 3D | $63.5\times10^6$ | 35.7G | 45.9% |

### 3.2.2 基于深度特征的预测

对特征提取器获取的深度视觉特征直接使用分类器获得标签预测分数，使用的分类器是普通的多分类线性分类器和 ProtoNet[17] 分类器. 其中 ProtoNet 分类器是基于欧氏距离的一种度量学习方法，其思想是将一个类用一个“代表”表示，称之为类原型，类原型为该类所有样本特征的均值. 假设类别为 $c$ 的 $N$ 个视频提取到的视频特征为 $\boldsymbol{f}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\ldots,\boldsymbol{f}_{N-1},\boldsymbol{f}_N]\in\mathbb{R}^{N\times M}$，那么其对应类原型应该表示为

$$\boldsymbol{Proto}_c=\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{f}_n \tag{1}$$

给定一个查询样本 $q$，它被预测为类别 $c$ 的概率的计算公式为

$$p(class=c \mid q)=\frac{\exp(-\| \boldsymbol{f}_q-\boldsymbol{Proto}_{\_c} \|_2)}{\sum_{i=1}^{C}\exp(-\| \boldsymbol{f}_q-\boldsymbol{Proto}_{\_i} \|_2)} \tag{2}$$

通过与所有类原型之间计算概率值,最后取最大概率值对应的类为样本 $q$ 的预测类.

3.2.3　基于流形特征的预测

基于流形特征的标签预测依赖表征传播和标签传播两个传播过程.表征传播作用是平滑视频特征间的流形结构同时生成表征传播特征,而标签传播则是根据表征传播特征完成标签预测.二者的使用可以缓解小样本学习中的域偏移与枢纽点问题.

表征传播的输入是特征提取器所获得的特征 $\boldsymbol{f}_i \in \mathbb{R}^M$.表征传播也用到了基于度量的方法,在传播的第一步就是计算所有的特征对之间的欧式距离,将一个特征对用 $(i,j)$ 表示,两者之间的欧式距离为 $d_{ij}=\| \boldsymbol{f}_i-\boldsymbol{f}_j \|_2$,然后计算邻接矩阵 $\boldsymbol{A}_{ij}=\exp(-{d_{ij}}^2/\sigma)$,其中 $\sigma$ 是一个缩放因子,定义为 $\sigma=\sqrt{Var({d_{ij}}^2)}$,$\sigma$ 就是 ${d_{ij}}^2$ 的标准差.邻接矩阵在这里可以看作是流形结构中各个点之间的关系矩阵,矩阵中的值越小,表示对应的两个点之间的相似度越高.接下来需要计算的是拉普拉斯矩阵:

$$\boldsymbol{L}=\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}},\boldsymbol{D}_{ii}=\sum\nolimits_j \boldsymbol{A}_{ij} \tag{3}$$

拉普拉斯矩阵中的第 $i$ 行实际反应了其它节点变化时,第 $i$ 个节点获得的累加收益.最后使用"经典的标签传播矩阵[32]",传播矩阵可以表示为

$$\boldsymbol{P}=(\boldsymbol{I}-\alpha\boldsymbol{L})^{-1} \tag{4}$$

其中 $\alpha$ 是缩放因子,控制着传播过程中当前节点的初始信息量以及其他节点的信息量占比,$\boldsymbol{I}$ 是身份矩阵,身份矩阵就是对角线的值为 1 的矩阵,对应的就是节点其对自身的影响力.所以,最后模型中的表征传播过程可以用以下公式表示:

$$\tilde{\boldsymbol{f}}_i=\sum\nolimits_j \boldsymbol{P}_{ij}\boldsymbol{f}_j \tag{5}$$

其中,$\tilde{\boldsymbol{f}}_i$ 表示节点 $i$ 的所有"邻居(包括当前节点自身)"的加权和,经过表征传播后,所有的节点的特征就都实现为其"邻居"的特征加权表示,联系"紧密"的邻居节点的权重就高一些,关系不密切的邻居权重就低.以上操作在小样本学习中实现后的计算复杂度不算高,图 2 中展示了表征传播的传播过程.

不同于表征传播,标签传播的对象由特征变为样本标签,而且标签传播是在表征传播的基础上的再传播.在标签传播进行之前,依旧先由特征提取器

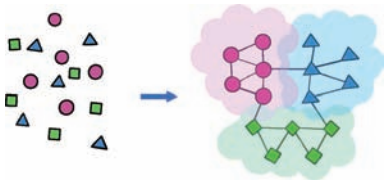

图 2　表征传播

获得特征,再将这些视频特征进行表征传播,然后才是标签传播.具体来说,标签传播的操作如下:将支持集和查询集视频特征经过表征传播后获得的特征表示为 $\tilde{\boldsymbol{F}} \in \mathbb{R}^{(k+q)\times m}$,然后计算 $\tilde{\boldsymbol{F}}$ 对应的邻接矩阵,进而计算拉普拉斯矩阵,再使用"经典的标签传播矩阵"计算得到传播矩阵 $\tilde{\boldsymbol{P}}$,对支持集中的样本标签进行独热编码(One-Hot Encoding),编码结果为 $\boldsymbol{Y}_s \in \mathbb{R}^{k\times n}$,同时对查询集建立对应的零矩阵 $\boldsymbol{Y}_q \in \mathbb{R}^{q\times n}$,然后将二者拼接得到 $\boldsymbol{Y} \in \mathbb{R}^{(k+q)\times n}$,最后对 $\boldsymbol{Y}$ 进行标签传播:

$$\tilde{\boldsymbol{Y}}_i=\sum\nolimits_j \tilde{\boldsymbol{P}}_{ij}\boldsymbol{Y}_j \tag{6}$$

经过以上传播过程可以得到查询集标签的传播结果 $\tilde{\boldsymbol{Y}}_Q$,标签传播的示意图如图 3 所示,图中非白色的几何图形即代表有标签的支持集数据样本,而白色的几何图形则表示未带标签的查询集样本.经过标签传播,从示意图不难看出所有标签的颜色变为了混色,意思是它们的标签被表示为其他样本标签的加权和,其中不同的颜色对应不同的类标签,随后将查询集样本标签被预测为占比最大的颜色对应的类标签,也就是各个类的累加预测分数中的最高值对应的类标签.

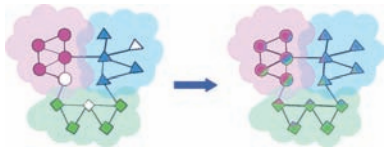

图 3　标签传播

3.2.4　多特征融合

针对以往方法在增加数据集信息量方面的不足,本文提出多特征融合的思想,即获取视频数据多种形式的特征,增加判别信息的同时避免了信息重复度高和类间相似性大问题的发生.

特征包含于知识,在行为识别中,知识可以划分为初级知识、抽象知识和外部知识[33],不同形式的知识优缺点各不相同.在知识的选择上,摒弃表示能力弱的初级知识和解释性弱且存在多义性的外部知识,选用表达能力强且广泛应用的抽象知识,具体为

深度视觉特征和流形特征.深度视觉特征能很好地反应数据的本质特征,但会损失部分的分布信息,因此同时选用了数据的流形特征,其中流形正则化项能有效保持数据的结构,保留更多的数据信息,从而缓解域偏移与枢纽点问题.以上两种特征之间的信息重复度较低,通过对二者进行融合,支持集样本中单个类别的信息总量得到有效的提升,较好地缓解了小样本学习中判别信息不足的问题.深度视觉特征与流形特征的优势各不相同,二者融合后能达到优势互补、相辅相成的效果,其中流形特征对与缓解小样本学习中的域偏移与枢纽点问题尤为有效.

多特征融合的具体操作是将深度特征与流形特征分别对应的标签预测分数进行加权求和,从而得到最终的标签预测分数.假设流形特征对应的预测分数为 $\boldsymbol{P}_{cls\_lab}$,深度特征对应的预测分数为 $\boldsymbol{P}_{cls\_proto}$,分配给它们的权重分别为 $w_1$ 与 $w_2$,且权重之和为 1,那么多特征融合可以表示为

$$\begin{cases} w_1 \times \boldsymbol{P}_{cls\_lab} + w_2 \times \boldsymbol{P}_{cls\_proto} = \boldsymbol{P}_{total} \\ w_1 + w_2 = 1 \end{cases} \tag{7}$$

最终获得多特征融合后的预测结果 $\boldsymbol{P}_{total}$,它兼顾了深度特征与流形特征各自的优势,取得了更好的标签预测效果.

### 3.3 模型流程

上面介绍的具体方法将在模型的流程中得到应用.模型流程分为预训练阶段、微调阶段和测试阶段,这三个阶段的网络模型细节略有差异,本节将分别介绍三个阶段各自的网络模型.

#### 3.3.1 预训练阶段

模型的训练集中在该阶段,该阶段基于训练集 $D_{train}$ 对模型进行训练,其网络如图 4 所示.在该阶段,将训练集的视频数据作为输入,通过特征提取器获取视频特征,然后此视频特征将分别用作基于深度特征的分类和基于流形数据分布的分类任务.在基于深度特征的分类对应的分支上,使用了最基本的线性分类器对训练样本进行分类,得到视频样本的预测值,然后通过交叉熵损失函数得到分类损失 $L_D$.在基于流形数据分布的分支上,将视频特征使用表征传播矩阵从而得到表征传播特征,再由此特征进行线性分类,随后也是用交叉熵损失函数得到损失 $L_E$.最后将深度特征损失和数据流形分布损失相加,并以此总损失进行反向传播.除了训练之外,需要在模型的验证流程中对训练的结果进行评估,同时根据此评估结果对模型训练中使用的参数进行调整.测试流程用来评估经过每轮训练之后的模型在测试集上的最终性能.模型中各个阶段的验证流程与测试流程的操作均一致,如图 5 中的实线部分所示,二者的区别是测试流程中输入数据变为 $D_{test}$,同时将融合后的查询集标签预测值 $\boldsymbol{P}_{total}$ 作为输出即可,无需计算分类损失.算法 1 和算法 2 中归纳了该阶段训练流程与验证流程的详细步骤.

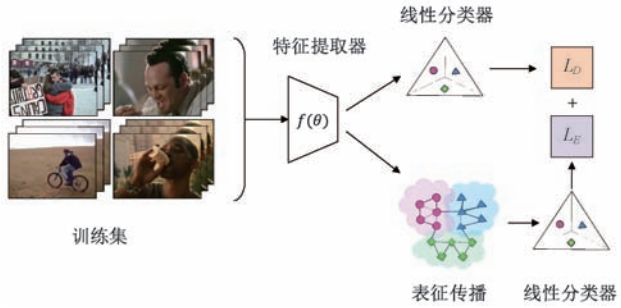


图 4 模型预训练阶段的训练流程

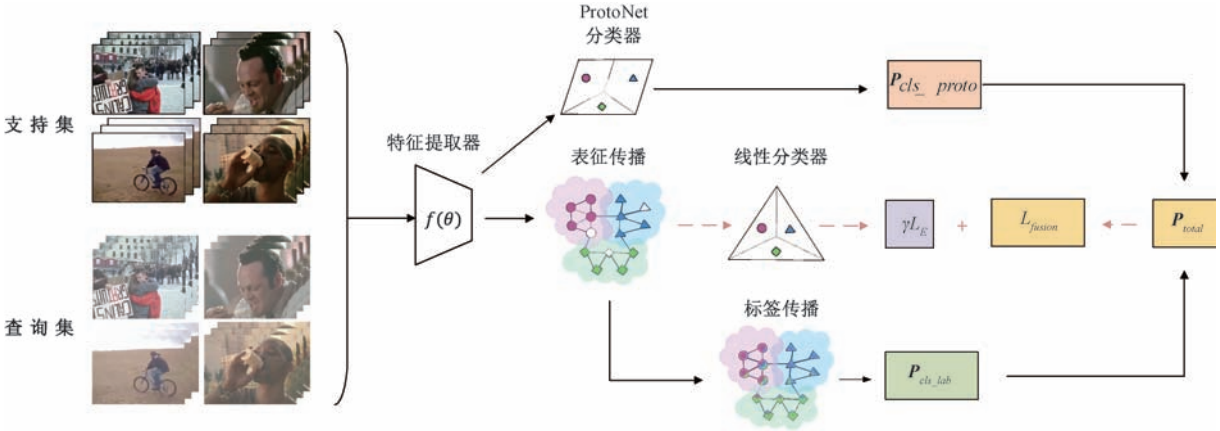


图 5 模型的微调与测试流程

**算法 1.** 预训练阶段的训练流程

训练流程

输入:训练集 $D_{train}$

输出:分类损失值

1. FOR i = 1, …, iters DO
2. get feature from $D_{train}$ as $\boldsymbol{X} \in \mathbb{R}^{(B\times T)\times m}$
3. get average value of frames from $\boldsymbol{X}$ as $\boldsymbol{F} \in \mathbb{R}^{B\times m}$
4. $L_D \leftarrow CrossEntropy(Linear(\boldsymbol{F}), \boldsymbol{Label})$
5. $\tilde{\boldsymbol{F}} \in \mathbb{R}^{B\times m} \leftarrow EmbeddingPropagation(\boldsymbol{F})$
6. $L_E \leftarrow CorssEntropy(Linear(\tilde{\boldsymbol{F}}), \boldsymbol{Label})$
7. $L_{train} = L_D + L_E$
8. back-propagation

**算法 2.** 预训练阶段的验证流程

输入:验证集 $D_{val}$,当前总训练轮数 $epoch\_num$,当前总损失 $totalLoss$,耐心值 $patience$

输出:分类损失值

1. set parameter: $Loss = 0$
2. FOR i = 1, …, iters DO
3. get feature from $D_{val}$ as $\boldsymbol{X} \in \mathbb{R}^{n\times(k+q)\times T\times m}$
4. get average value of frames from $\boldsymbol{X}$ as $\boldsymbol{F} \in \mathbb{R}^{n\times(k+q)\times m}$
5. $\boldsymbol{P}_{cls\_proto} \leftarrow ProtoNet(\boldsymbol{F})$
6. $\tilde{\boldsymbol{F}} \in \mathbb{R}^{n\times(k+q)\times m} \leftarrow EmbeddingPropagation(\boldsymbol{F})$
7. $\boldsymbol{P}_{cls\_lab} \leftarrow LabelPropagation(\tilde{\boldsymbol{F}})$
8. $\boldsymbol{P}_{total} \leftarrow w_1 \times \boldsymbol{P}_{cls\_lab} + w_2 \times \boldsymbol{P}_{cls\_proto}$
9. $L_{val} \leftarrow CrossEntropy(\boldsymbol{P}_{toral}, \boldsymbol{Label})$
10. $Loss = Loss + L_{val}$
11. IF $totalLoss/epoch\_num < Loss/iters$
12. $patience \leftarrow patience + 1$
13. IF $patience > patienceMax$
14. update learning rate
15. $totalLoss \leftarrow totalLoss + Loss/iters$

### 3.3.2　微调阶段

经过预训练阶段得到的模型还不够完善,因此还设计了一个针对小样本学习任务“N-way, K-shot”的微调阶段,目的是让模型适应小样本学习的模式,以获得更好的泛化性能.此阶段以 episode 为单位进行训练,当中包含的查询集样本标签是未知的,为了更好的进行标签预测,将预训练阶段使用的基于深度视觉特征的线性分类器替换成 ProtoNet 分类器.在流形的数据分布分支上也加入了标签传播,这些都是为了更好对小样本学习中的查询集样本进行预测,所以说此阶段是针对小样本学习任务进行的适应性调整.微调阶段训练流程对应的网络如图 5 所示,首先使用特征提取器获得深度特征,然后对特征直接使用 ProtoNet 分类器得到查询集的预测分数 $\boldsymbol{P}_{cls\_proto}$,另一分支上对特征使用表征传播和标签传播得到预测分数 $\boldsymbol{P}_{cls\_lab}$,两者加权相加得到最终预测分数 $\boldsymbol{P}_{total}$,进而得到分类损失 $L_{fusion}$,微调阶段的总损失 $L_{finetune}$ 还需要加上对表征传播特征使用线性分类器获得的交叉熵分类损失 $L_E$,并对 $L_E$ 赋予权重 $\gamma$.该阶段的验证和测试流程与预训练阶段相同,不同的地方在于微调阶段在训练时增加了一个损失,即对表征传播特征使用线性分类器得到的分类损失.微调阶段的模型训练流程在算法 3 中进行了归纳.

**算法 3.** 微调阶段的训练流程

输入:训练集 $D_{train}$

输出:分类损失值

1. FOR i = 1, …, iters DO
2. get feature from $D_{train}$ as $\boldsymbol{X} \in \mathbb{R}^{n\times(k+q)\times T\times m}$
3. get average value of frames from $\boldsymbol{X}$ as $\boldsymbol{F} \in \mathbb{R}^{n\times(k+q)\times m}$
4. $\boldsymbol{P}_{cls\_proto} \leftarrow ProtoNet(\boldsymbol{F})$
5. $\tilde{\boldsymbol{F}} \in \mathbb{R}^{n\times(k+q)\times m} \leftarrow EmbeddingPropagation(\boldsymbol{F})$
6. $L_E \leftarrow CrossEntropy(Linear(\tilde{\boldsymbol{F}}), \boldsymbol{Label})$
7. $\boldsymbol{P}_{cls\_lab} \leftarrow LabelPropagation(\tilde{\boldsymbol{F}})$
8. $\boldsymbol{P}_{total} \leftarrow w_1 \times \boldsymbol{P}_{cls\_lab} + w_2 \times \boldsymbol{P}_{cls\_proto}$
9. $L_{fusion} \leftarrow CrossEntropy(\boldsymbol{P}_{toral}, \boldsymbol{Label})$
10. $L_{finetune} \leftarrow L_{fusion} + \gamma \times L_E$
11. back-propagation

### 3.3.3　测试阶段

此阶段用于检验模型的最终性能,在数据集 $D_{test}$ 进行,其网络如图 5 中的实线部分所示.注意,此阶段的流程不仅仅存在于测试阶段对模型的性能进行评估,其也贯彻于预训练阶段和微调阶段.前两个阶段都包含训练、验证和测试三个子流程,其中测试流程是方便了解当前模型的实际性能,并不参与前两个阶段的训练流程.此阶段的流程与微调阶段的训练流程颇为相似,但本阶段并没有对表征传播特征使用分类器进行分类,而仅仅通过对深度特征使用 ProtoNet 分类器和对表征传播特征使用标签传播得到各自的预测分数,然后融合得到总的预测分数,从而实现查询集标签的预测.

## 3.4　损失函数

模型使用了深度特征以及数据的流形分布两种特征,同时模型拥有多个阶段,不同阶段的损失函数也有所区别,但涉及到的任务都是分类任务,因此损

失均为分类损失，具体使用的分类损失函数是交叉熵损失函数.

3.4.1 预训练损失函数

该损失函数对应模型预训练阶段的训练流程，涉及到两个任务，分别是根据深度特征进行线性分类和根据表征传播特征进行线性分类，两者都属于分类损失，使用的都是交叉熵损失函数，假设在使用线性分类器后得到的预测分数为 $\boldsymbol{p} \in \mathbb{R}^C$，单个交叉熵损失函数表示为：

$$L_{cls} = -\sum_{c=1}^{C} \boldsymbol{y}_c \log \boldsymbol{p}_c \tag{8}$$

结合深度特征以及流形的数据分布两者的损失，以构成预训练阶段的总损失函数，如下所示：

$$L_{train} = L_D + L_E \tag{9}$$

其中 $L_D$ 和 $L_E$ 分别对应根据深度视觉特征和数据的流形分布特征得到预测分数进而计算出的交叉熵分类损失.

3.4.2 验证损失函数

模型的验证步骤也是根据深度特征和数据流形分布计算分类损失. 首先是根据深度特征进行分类，这里使用的不是线性分类器，而是 ProtoNet 分类器，使用该分类器得到的预测分数为 $\boldsymbol{P}_{cls\_proto}$，再者是在流形数据分布上根据标签传播得到的预测分数 $\boldsymbol{P}_{cls\_lab}$，之后将上面提到的两个预测分数按照多特征融合中的加权求和策略得到全新的预测分数 $\boldsymbol{P}_{total}$，最终的损失由该预测分数使用交叉熵损失函数得到. 因此，最终验证损失函数表示为：

$$L_{val} = -\sum_{c=1}^{C} \boldsymbol{y}_c \log \boldsymbol{P}_{total\_c} \tag{10}$$

3.4.3 微调损失函数

在模型的微调阶段中，基于验证损失函数的基础添加了表征传播特征对应的线性分类损失 $L_E$，同时其带有权重 $\gamma$，所以微调损失函数的表现形式如下：

$$L_{finetune} = -\sum_{c=1}^{C} \boldsymbol{y}_c \log \boldsymbol{P}_{total\_c} + \gamma \times L_E \tag{11}$$

# 4 实验与评估

## 4.1 数据集

实验中使用到的数据集包括 HMDB51、UCF101 与 Kinetics. 其中 HMDB51 数据集由 51 个动作类的 6849 个视频组成，每个动作类至少包含 101 个视频，视频主要来源是电影作品. 该数据集中数据的变化主要体现在目标外观和人体姿态方面，其包含的动作主要分为五类：一般面部动作、有操作对象的面部动作、一般的身体动作、人与人交互的动作和人与对象交互的动作. 该数据集划分为训练集、验证集和测试集三部分，这三部分各自包含 31、10 和 10 个动作类.

UCF101 是从 YouTube 收集的包含 101 个动作类的视频数据集. 该数据集追求视频动作的真实性，每个动作类中的视频都由 25 组人完成，每组人完成 4 至 7 个视频，同时每组人完成的视频内容在相机运动、目标姿势与变化、物体大小、视角、背景纯净度、光照等方面存在很大的差异，数据集共 13320 个视频，分辨率均为 320 × 240. 该数据集同样被划分为训练集、验证集和测试集三部分，每部分含有 70、10 以及 21 个动作类.

Kinetics 是从 YouTube 收集的一个动作视频数据集，视频长度 10s 左右，包含 400 个类的 306245 个视频，涵盖范围广泛的各类运动与事件. 具体使用的其实是 Kinetics100，即从 Kinetics400 中选取 100 个类，每个类取 100 个视频. 将 Kinetics100 划分为 64、12 和 24 个动作类的三部分，分别对应训练集、验证集和测试集.

以上所有数据集均依据 ARN[12] 中给出的划分标准划分为训练集、验证集和测试集.

## 4.2 实现细节

实验中，使用基于 ImageNet 预训练的 ResNet18 卷积神经网络作为特征提取器. 进行特征提取的时候，输入的一个视频经采样后由 16 帧的 RGB 图像代替，然后将每一帧图片都重新裁剪为 224×224 大小，并且对视频帧图片采取了 50% 概率的水平方向的随机翻转策略.

在模型预训练阶段的训练流程中，批处理大小被设置为 32(Kinetics 数据集下为 128)，训练、验证流程都迭代 200 次. 在微调阶段，针对 1-shot 和 5-shot 两种情形对模型进行微调，此时模型是基于 episode 进行训练的，batch-size 为 1，训练和验证流程的迭代次数为 100 和 200. 实验中融合深度特征与数据的流形分布特征进行预测的时候，流形分布特征对应的预测分数占比为 $w_1 = 0.8$，深度特征对应预测分数占比为 $w_2 = 0.2$. 模型进行微调时，微调损失函数中表征传播特征对应的分类损失 $L_E$ 的权重 $\gamma = 0.1$. 在学习率的设置上，预训练阶段和微调阶段的初始学习率都是 0.01，然后设置一个参数 patience，大小为 6，当模型在验证集上计算得到的损失值连续 patience 次小于之前所有验证损失的平均值时，学习率衰减一次，同时也设置了学习率的最

小值,当学习率衰减了 4 次之后就结束训练流程.实验中使用的是梯度下降算法是随机梯度下降.

实验中的运行环境为 CPU Intel Xeon E5-2620,其主频为 2.1 GHz×7 cores,缓存大小为 188G,采用 8 张 NVIDIA GTX 1080Ti 显卡,显存大小共 96GB,操作系统为 Ubuntu 16.04.7 LTS,深度学习框架是 PyTorch.

### 4.3　评估标准

对小样本学习结果进行评估时,使用了"N-way K-shot"这一重要概念,具体来说,使用的是目前主流的两种评估方案,即"5-way 1-shot"和"5-way 5-shot".以上提到的两种方案都是从测试集数据中取 5 个类的数据作为支持集,然后 1-shot 指每个类取 1 个样本,5-shot 为每个类取 5 个样本.除了支持集,还需要选取查询集,其类别应与支持集一致,且二者包含的样本无交集,本文在每个类中选取 15 个查询样本.最后由支持集和查询集组成一个 episode 进行模型性能评估.

在最后计算查询集预测结果的时候,使用准确率这一判定标准,进行 200 次的重复实验取平均值以及 95%置信度区间.

### 4.4　对比方法定量分析

为了验证本文方法的有效性,表 3 和表 4 展示了现有的小样本视频行为识别方法以及本文方法在 HMDB51、UCF101 和 Kinetics 数据集上针对小样本学习中的"5-way 1-shot"和"5-way 5-shot"任务进行测试的结果,并给出了不同方法的 Backbone 对应的特征维度、参数量与计算量(FLOPs).从表中的统计信息不难发现,以往的模型大多用到了 3D 维度的特征提取器,这能让特征提取器获取更多时间维度上的信息,而基于 2D 方法的特征提取器对时间维度的信息没有那么敏感,本文方法选择基于 2D 特征提取器构建小样本视频行为识别模型,并通过多特征融合、平滑流形结构的操作来发掘 2D 方法的潜力.模型使用 ResNet18 作为 Backbone,它的 FLOPs 仅 1.8G,参数量仅 11.7M,而 C3D 与 I3D 等 3D 特征提取器的这两项数值是它的数倍甚至数十倍,最终本文模型基于该 2D 方法以极小的计算量与参数量代价获得了先进的性能.在小样本学习的 1-shot 情况下,本文方法对分类准确率的提升尤为明显,与现有方法相比,在 HMDB51 数据集上提升了 8.5%,在 UCF101 数据集上提升了 9.5%,在 Kinetics 数据集上提升了 1.0%,而在 5-shot 情况下,本文方法与以往 3D 方法相比也毫不逊色,在 HMDB51 和 UCF101 数据集上性能有明显提升,在 Kinetics 数据集上也与 3D 方法基本持平.其中值得注意的是,"Video Disentangling Attentive Relation Network"(VDARN)[34] 方法也使用了多种形式的特征,通过 ST-GCN[35] 网络提取时空骨架信息,同时使用 GoogLeNet[36] 和语义嵌入提取目标信息,该方法使用的两个 Backbone 的维度分别是 3D 与 2D,其总的 FLOPs 与 ResNet18 相当,但本文仅使用 ResNet18 作为 Backbone,模型的网络结构更为简洁,参数量也更低.与 VDARN 方法相比,本文方

**表 3　本文方法与现有方法在 HMDB51 和 UCF101 上测试结果对比**

| 方法 | Backbone | | | | HMDB51 上准确率/% | | UCF101 上准确率/% | |
|---|---|---|---|---|---|---|---|---|
| | 名称 | 特征维度 | 参数量 | FLOPs | 5way-1shot | 5way-5shot | 5way-1shot | 5way-5shot |
| I3D[10] | I3D | 3D | 107.9M | 12.1G | 13.7 | — | 23.4 | — |
| I3D+TSF[37] | I3D | 3D | 107.9M | 12.1G | 27.0 | — | 48.4 | — |
| GenApp[38] | C3D | 3D | 34.8M | 38.5G | — | 52.5±3.1 | — | 78.6±2.1 |
| ProtoGAN[39] | C3D | 3D | 34.8M | 38.5G | 34.7±9.2 | 54.0±3.9 | 57.8±3.0 | 80.3±1.3 |
| I3D+SLDG[40] | I3D | 3D | 107.9M | 12.1G | 36.0 | — | 68.3 | — |
| VDARN[34] | ST-GCN + GoogLeNet | 3D+2D | 3.0M+7.0M | 8.1G+1.6G | 39.7±3.6 | 56.3±2.9 | 59.9±2.9 | 81.6±3.2 |
| ARN[12] | C3D | 3D | 34.8M | 38.5G | 44.6±0.9 | 59.1±0.8 | 62.1±1.0 | 84.8±0.8 |
| Ours | ResNet18 | 2D | 11.7M | 1.8G | **53.1±0.8** | **68.4±1.3** | **77.8±2.0** | **91.9±1.1** |

**表 4　本文方法与现有方法在 Kinetics 上测试结果对比**

| 方法 | Backbone | | | | Kinetics 上准确率/% | |
|---|---|---|---|---|---|---|
| | 名称 | 特征维度 | 参数量 | FLOPs | 5way-1shot | 5way-5shot |
| CMN[13] | ResNet50 | 2D | 25.5M | 3.8G | 60.5 | 78.9 |
| ARN[12] | C3D | 3D | 34.8M | 38.5G | 63.7 | **82.4** |
| TRAN[41] | C3D | 3D | 34.8M | 38.5G | 66.6 | 80.7 |
| Ours | ResNet18 | 2D | 11.7M | 1.8G | **67.6±1.9** | 82.2±1.3 |

法在 HMDB51 和 UCF101 数据集均有更好的分类精度，可见本文方法在特征类型的选择上和多特征使用策略的构建上更合理. 此外，也有少数基于 2D 特征提取器的方法，如表 5 中的“Compound Memory Networks”(CMN)[13]，从表 5 中可以看出本文的 2D 方法拥有更低的资源消耗，同时拥有更先进的性能，可见本文方法更好地挖掘了 2D 方法的潜力.

### 4.5 消融实验

为了更加深入地研究基于深度特征与流形特征的 2D 小样本视频行为识别模型，本节将介绍一系列消融实验的结果. 以预训练阶段结束后得到的模型性能作为评估标准，因为微调阶段是为了让模型更适应小样本学习，提高模型的稳定性，且通过实验发现，其对于模型在测试集上的最终性能的提升比较有限. 为了验证评估标准的合理性，本文在 HMDB51、UCF101 及 Kinetics 数据集上进行了实验，如表 5 所示，分别对预训练和微调阶段(针对 5-shot 情形微调)结束后得到的模型进行测试，可以发现模型在验证集和测试集上分类精度的提升都比较小，说明通过了解预训练阶段得到模型的性能，基本可以预见模型微调后的性能与最终性能. 因此，在消融实验中只比较模型完成预训练阶段之后的性能即可，且未特别注明时，本文进行的消融实验基于 HMDB51 数据集.

**表 5 预训练阶段与微调阶段结束后模型分类准确率**

| 数据集 | 验证与测试 | 预训练阶段 | 微调阶段 |
|---|---|---|---|
| HMDB51 | 验证结果 | 72.81% | 74.98% |
| | 测试结果 | 67.48% | 67.88% |
| UCF101 | 验证结果 | 90.90% | 92.22% |
| | 测试结果 | 90.52% | 91.92% |
| Kinetics | 验证结果 | 83.88% | 83.65% |
| | 测试结果 | 81.49% | 81.95% |

#### 4.5.1 多特征融合策略分析

本文方法同时使用了深度视觉特征与数据的流形分布特征，通过对二者对应的标签预测分数进行加权求和得到了全新的预测分数，在使用合理的权重分配方案后，该预测分数具有更强的判别性. 为了证明模型中选取的权重分配方案是最合适的，在消融实验中进行了各种不同权重组合的实验，同时保证两者的权重值之和始终为 1，然后从 0 至 1 之间选取流形分布特征的对应权重进行实验，相反基于深度视觉特征对应的预测分数的权重取值则是从 1 到 0. 实验结果如图 6 所示，从图中可以看出，本文提出的模型在流形分布特征权重 $w_1=0.8$ 和深度特征权重 $w_2=0.2$ 时的性能最优，而且当 $w_1=1.0$ 的时候，表示只考虑了数据的流形分布特征，其表现不如分配了一定权重给深度视觉特征对应的结果，在 $w_1=0$ 的时候，单独使用深度视觉特征的结果也是不如两者融合后的分类结果. 可见，这两种形式的特征经过合理的权重分配进行融合后可以达到相辅相成的效果，形成优势互补，两者在 $w_1=0.8$、$w_2=0.2$ 处达到一种平衡状态，以此情况为分界线，流形分布和深度视觉特征的权重不管是增加还是减少，模型的性能都呈下降趋势. 此外，还测试了 $w_1$ 分别为 0、0.8 以及 1 时的模型在测试集中 10 个动作类的具体分类性能表现，结果如图 7 所示，从图中可以看出，当 $w_1$ 为 0.8 的时候，模型在 6 个动作类上取得了最佳分类效果，在动作类 kick 上的表现与 $w_1$ 为 0 时基本持平，在剩余的 3 个动作类上虽然效果不是最佳，但均在朝最佳项结果靠拢，较最差项结果有明显提升. 通过以上实验，可以发现将深度视觉特征与数据的流形分布特征进行合理的融合后，在判别信息的总量得到增加的同时，两种形式的特征也能互相影响，相互促进，进而实现优于单独使用任何一种特征的小样本视频行为分类性能. 此外，多特征融合方案中样本特征信息的冗余度低，在解决信息量稀缺问题时避免了引入信息重复度高和类间相似性大的新问题.

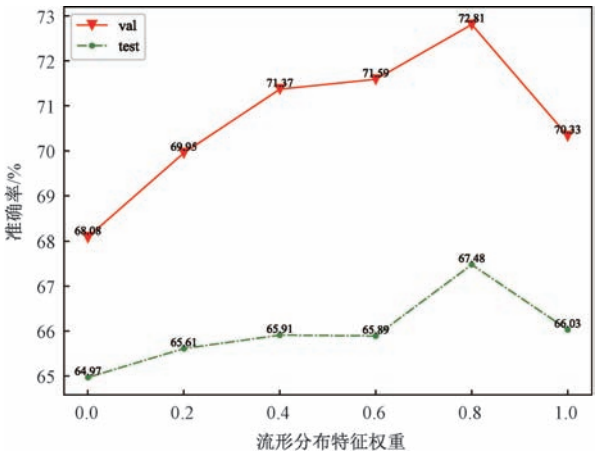


图 6 使用不同权重分配融合后对应的预测结果

#### 4.5.2 不同分类器的对比实验

在模型各个阶段的验证与预测流程当中，需要对深度视觉特征与流形特征使用分类器获得标签预测值，此时就涉及到了分类器的选择，本文选择使用 Logistic(逻辑回归，Logistic Regression)、KNN(K-Nearest Neighbor)、SVM(Support Vector Machine)、LP(标签传播，Label Propagation)以及 Pro-

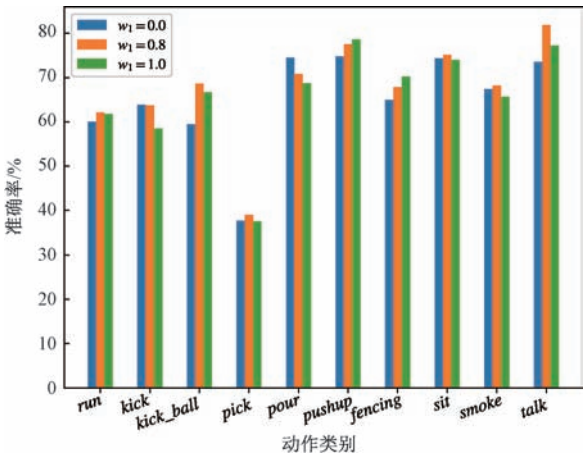


图 7　不同权重分配下测试集中所有类的分类准确率

toNet 这五种分类器进行对比实验,评估标准是模型在预训练阶段结束后在验证集和测试集上的性能表现,此时深度特征与流形特征分支对应的预测权重分别是 0.2 和 0.8,实验结果如表 6 所示.分别对基于视觉特征和流形特征的预测分支使用不同的分类器进行对比实验,对视觉特征分支的分类器进行选择时,流形特征分支的分类器默认是 LP,当选择流形特征分支的分类器时,视觉特征分支则默认使用 ProtoNet 分类器.从表 6 的实验结果可以看出,不管先确定视觉特征分支使用 ProtoNet 分类器还是先确定流形特征分支使用 LP,最后结果均一致,即视觉特征分支使用 ProtoNet 分类器和流形特征分支使用 LP 进行标签预测时,能达到最好的分类效果.

**表 6　两个预测分支使用不同分类器的结果**

| 分类器 | 视觉特征预测分支 | | 流形特征预测分支 | |
|---|---|---|---|---|
| | 验证阶段 | 测试阶段 | 验证阶段 | 测试阶段 |
| Logistic | 64.89% | 59.68% | 71.65% | 65.73% |
| SVM | 66.84% | 61.53% | 69.73% | 64.64% |
| KNN | 70.27% | 65.07% | 65.16% | 61.37% |
| LP | — | — | **72.81%** | **67.48%** |
| ProtoNet | **72.81%** | **67.48%** | 70.12% | 65.37% |

### 4.5.3　缩放因子的对比实验

为了了解公式 4 中缩放因子 $\alpha$ 的取值对模型性能的影响,以及寻找合适的缩放因子取值,本文对表征传播与标签传播分别使用不同的缩放因子,尝试多种组合形式,通过大量实验进行对比,最终预训练模型在验证集上的分类准确率统计结果如表 7 所示.从表 7 中不难看出表征传播和标签传播中缩放因子都取 0.3 能达到最佳效果,虽然缩放因子取值较小时模型的性能没有显著差异,但以 $\alpha_1=0.3$ 和 $\alpha_2=0.3$ 为中心,缩放因子增加与减小时分类准确率都呈下降趋势.值得注意的是,表征传播中缩放因子取值过大会使模型性能显著下降.

**表 7　缩放因子取不同值时分类准确率对比**　(单位:%)

| 标签传播中 $\alpha_2$ 的取值 | 表征传播中 $\alpha_1$ 的取值 | | | | |
|---|---|---|---|---|---|
| | 0.0 | 0.1 | 0.3 | 0.5 | 0.7 |
| 0.1 | 70.27 | 70.84 | 70.44 | 65.19 | 45.23 |
| 0.2 | 70.63 | 71.09 | 71.57 | 69.16 | 58.05 |
| 0.3 | 69.63 | 71.45 | **72.81** | 69.47 | 57.29 |
| 0.5 | 68.91 | 69.83 | 70.97 | 67.43 | 55.28 |

以上关于缩放因子的实验是基于 HMDB51 数据集进行的,得出最适合该数据集的 $\alpha_1=0.3$ 和 $\alpha_2=0.3$ 两个缩放因子.在 UCF101 数据集上,最佳缩放因子大小为 $\alpha_1=0.1$ 和 $\alpha_2=0.2$,设置得更小是因为该数据集视频来源的特殊性,UCF101 数据集中每个动作类中的视频都由 25 组人完成,每组人完成 4 至 7 个视频,这就造成不少视频间的相似度是相对较大的,此时缩放因子设置过大的话,虽然平滑程度更高,同类的联系更加紧密,但单个视频特征中也将引入更多“噪声”,其自身的特征会被稀释,对于小样本学习这种模式来说,此时让支持集视频保留更多自身的特性是更合适的,因为查询集视频极有可能与支持集视频为同一组人拍摄,所以说在 UCF101 数据集下设置较小的缩放因子能让这些视频更加精准的匹配.Kinetics 数据集较为庞大且视频来源复杂,缩放因子取值偏小时对模型的性能无明显影响,本文在该数据集上最终设置为 $\alpha_1=0.5$ 和 $\alpha_2=0.2$.

### 4.5.4　平滑操作的定量分析

为了验证表征传播对数据的流形结构进行平滑操作后能提升模型的性能,本文在 HMDB51、UCF101 以及 Kinetics 数据集上进行对比实验,并控制唯一变量为是否使用表征传播.此对比实验中选择多特征融合效率最高的模型进行实验,即深度特征对应预测值权重为 0.2,流形特征对应预测值权重为 0.8,此时取消使用表征传播,在基于流形特征的预测分支上直接对特征提取器获取的特征使用标签传播获取标签预测值.是否使用表征传播的小样本视频行为识别结果如表 8 所示,从表中验证流程与测试流程的分类准确率结果对比可以看出使用表征传播后模型的性能得到了提升.此外,表 7 中 $\alpha_1=0$ 这一列为表征传播中缩放因子等于 0 对应的结果,即未进行表征传播操作,与表征传播缩放因子 $\alpha_1=0.1$ 和 $\alpha_1=0.3$ 这两列的结果进行比较,能明显发现模型性能均有提高,同时不难发现当 $\alpha_1$ 取值

偏大时,性能相比没有经过平滑操作的模型反而下降,说明应当控制好流形结构的平滑程度,合理的进行表征传播.以上实验均说明本文从缓解域偏移与枢纽点问题出发的角度引入表征传播进行适度的平滑操作是正确的.

**表 8 是否使用表征传播分类准确率对比**

| 数据集 | 验证结果 是否使用表征传播 | | 测试结果 是否使用表征传播 | |
|---|---|---|---|---|
| | 否 | 是 | 否 | 是 |
| HMDB51 | 69.63% | 72.81% | 63.07% | 67.48% |
| UCF101 | 89.12% | 90.90% | 88.22% | 90.52% |
| Kinetics | 82.63% | 83.88% | 80.40% | 81.49% |

## 5 总 结

针对现有小样本视频行为识别方法在增加视频信息量时存在的信息重复度高、类间相似性大、计算量大等问题,同时鲜有关注域偏移与枢纽点问题,提出了一种有效的多特征融合方法.该方法基于深度特征与数据的流形分布特征两种特征进行标签预测,可以在增加判别信息的同时缓解冗余信息的引入,从而缓解小样本学习中信息量不足的问题,同时使用流形分布还能缓解域偏移与枢纽点问题.此外,基于 2D 方法构建模型极大地减少了参数量与计算量.在 HMDB51、UCF101 和 Kinetics 数据集上的大量实验,均验证了本文方法的有效性.

## 参 考 文 献


[1] Zhang H B, Zhang Y X, Zhong B, et al. A comprehensive survey of vision-based human action recognition methods. Sensors, 2019, 19(5): 1005

[2] Liu Yong, Xie Ruo-Ying, Feng Yang, et al. Survey on resident's daily activity recognition in smart homes. Computer Engineering and Aplications, 2021, 57(4): 35-42. (in Chinese)
(刘勇, 谢若莹, 丰阳等. 智能家居中的居民日常行为识别综述. 计算机工程与应用. 2021, 57(4): 35-42)

[3] Wu Yun-Peng, Zhao Chen-Yang, Shi Zeng-Lin, et al. A flow density based algorithm for detecting coherent motion with multiple interaction. Chinese Journal of Computers, 2017, 40(11): 2519-2532 (in Chinese)
(吴云鹏, 赵晨阳, 时增林等, 基于流密度的多重交互集体行为识别算法. 计算机学报, 2017, 40(11): 2519-2532)

[4] Ding Chong-Yang, Liu Kai, Li Guang, et al. Spatio-temporal weighted posture motion features for human skeleton action recognition research. Chinese Journal of Computers, 2020, 43(1): 29-40 (in Chinese)
(丁重阳, 刘凯, 李光等, 基于时空权重姿态运动特征的人体骨架行为识别研究. 计算机学报, 2020, 43(1): 29-40)

[5] Wang Y, Yao Q, Kwok J T, et al. Generalizing from a few examples: A survey on few-shot learning. ACM Computing Surveys (CSUR), 2020, 53(3): 1-34

[6] Fu Y, Hospedales T M, Xiang T, et al. Transductive multi-view zero-shot learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(11): 2332-2345

[7] Dinu G, Lazaridou A, Baroni M. Improving zero-shot learning by mitigating the hubness problem. arXiv preprint arXiv: 1412.6568, 2014

[8] Fu Y, Wang C, Fu Y, et al. Embodied one-shot video recognition: Learning from actions of a virtual embodied agent//Proceedings of the 27th ACM International Conference on Multimedia. Nice, France, 2019: 411-419

[9] Ji S, Xu W, Yang M, et al. 3D convolutional neural networks for human action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 35(1): 221-231

[10] Carreira J, Zisserman A. Quo vadis, action recognition? a new model and the kinetics dataset//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA, 2017: 6299-6308

[11] Cao K, Ji J, Cao Z, et al. Few-shot video classification via temporal alignment//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, USA, 2020: 10618-10627

[12] Zhang H, Zhang L, Qi X, et al. Few-shot action recognition with permutation-invariant attention//Proceedings of the European Conference on Computer Vision (ECCV). Glasgow, UK, 2020: 525-542

[13] Zhu L, Yang Y. Compound memory networks for few-shot video classification//Proceedings of the European Conference on Computer Vision (ECCV). Munich, Germany, 2018: 751-766

[14] Rodríguez P, Laradji I, Drouin A, et al. Embedding propagation: Smoother manifold for few-shot classification//European Conference on Computer Vision. Glasgow, UK, 2020: 121-138

[15] Vinyals O, Blundell C, Lillicrap T, et al. Matching networks for one shot learning. Advances in Neural Information Processing Systems, 2016, 29: 3630-3638

[16] Koch G, Zemel R, Salakhutdinov R. Siamese neural networks for one-shot image recognition//ICML Deep Learning Workshop. Lille, France, 2015, 1-22

[17] Snell J, Swersky K, Zemel R S. Prototypical networks for few-shot learning. arXiv preprint arXiv:1703.05175, 2017

[18] Sung F, Yang Y, Zhang L, et al. Learning to compare: Relation network for few-shot learning//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA, 2018: 1199-1208

[19] Santoro A, Bartunov S, Botvinick M, et al. Meta-learning

with memory-augmented neural networks//International Conference on Machine Learning. New York, USA, 2016: 1842-1850

[20] Ravi S, Larochelle H. Optimization as a model for few-shot learning//International Conference on Learning Representations. Toulon, France, 2017,1-11

[21] Zhang H, Zhang J, Koniusz P. Few-shot learning via saliency-guided hallucination of samples//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, USA, 2019: 2770-2779

[22] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. arXiv preprint arXiv: 1406.2199, 2014

[23] Wang L, Xiong Y, Wang Z, et al. Temporal segment networks: Towards good practices for deep action recognition// European Conference on Computer Vision. Amsterdam, Netherlands, 2016: 20-36

[24] Xie Zhao, Zhou Yi, Wu Ke-Wei, et al. Activity Recognition Based On Sptial-Temporal Attention LSTM. Chinese Journal of Computers, 2021, 44(2): 261-274 (in Chinese)
(谢昭, 周义, 吴克伟等, 基于时空关注度 LSTM 的行为识别. 计算机学报, 2021, 44(2): 261-274)

[25] Xu X, Hospedales T, Gong S. Transductive zero-shot action recognition by word-vector embedding. International Journal of Computer Vision, 2017, 123(3): 309-333

[26] Xu X, Shen F, Yang Y, et al. Matrix tri-factorization with manifold regularizations for zero-shot learning//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA, 2017: 3798-3807

[27] Fu Y, Hospedales T M, Xiang T, et al. Transductive multi-view zero-shot learning. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(11): 2332-2345

[28] Fu Y, Hospedales T M, Xiang T, et al. Transductive multi-view embedding for zero-shot recognition and annotation// European Conference on Computer Vision. Zurich, Switzerland, 2014: 584-599

[29] Fu Z, Xiang T, Kodirov E, et al. Zero-shot learning on semantic class prototype graph. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 40(8): 2009-2022

[30] He K, Zhang X, Ren S, et al. Identity mappings in deep residual networks//European Conference on Computer Vision. Amsterdam, Netherlands, 2016: 630-645

[31] Tran D, Wang H, Torresani L, et al. A closer look at spatiotemporal convolutions for action recognition//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA, 2018: 6450-6459

[32] Zhou D, Bousquet O, Lal T N, et al. Learning with local and global consistency//Advances in Neural Information Processing Systems. Vancouver, Canada, 2004: 321-328

[33] Feng Yao-Gong, Yu Jian, Sang Ji-Tao, et al. Survey on knowledge-based zero-shot visual recognition. Journal of Software, 2021,32(2): 370-405 (in Chinese)
(冯耀功, 于剑, 桑基韬等. 基于知识的零样本视觉识别综述. 软件学报, 2021, 32(2): 370-405)

[34] Su Yong, Xing Meng, An Simin, Peng Weilong, Feng Zhiyong. VDARN: Video disentangling attentive relation network for few-shot and zero-shot action recognition. Ad Hoc Networks, 2021, 113: 102380

[35] Yan S, Xiong Y, Lin D. Spatial temporal graph convolutional networks for skeleton-based action recognition//Thirty-Second AAAI Conference on Artificial Intelligence. New Orleans, USA, 2018: 7444-7452

[36] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Boston, USA, 2015: 1-9

[37] Piergiovanni A J, Ryoo M S. Learning latent super-events to detect multiple activities in videos//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA, 2018: 5304-5313

[38] Mishra A, Verma V K, Reddy M S K, et al. A generative approach to zero-shot and few-shot action recognition//2018 IEEE Winter Conference on Applications of Computer Vision (WACV). Nevada, USA, 2018: 372-380

[39] Kumar Dwivedi S, Gupta V, Mitra R, et al. Protogan: Towards few shot learning for action recognition//Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. Seoul, Korea, 2019: 1308-1316

[40] Bo Y, Lu Y, He W. Few-shot learning of video action recognition only based on video contents//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. Colorado, USA, 2020: 595-604

[41] Bishay M, Zoumpourlis G, Patras I. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. arXiv preprint arXiv:1907.09021, 2019

**PU Zhan-Xing**, M. S. candidate. His research interests include computer vision and machine learning.

**GE Yong-Xin**, Ph. D., associate professor. His research interests include computer vision, machine learning and intelligence of big data.

**Background**

This paper focuses on few-shot action recognition, which belongs to action recognition in computer vision. Traditional action recognition in video often requires large amounts of labeled data. When a pre-trained model needs to be adapted to recognize an unseen class, it necessary to manually collect hundreds of video samples for knowledge transfer, but such a procedure is time-consuming and labor intensive. Moreover, the complexity and cost of labeling videos are much higher compared to labeling images. There is a growing interest in making models effectively adapt themselves to recognize novel classes with only a few examples from support set. This is known as few-shot learning.

Considering the support set is very limited in few-shot learning, so more discriminative information is particularly important. In order to alleviate the lack of sample's information, lots of methods have been proposed to expand the dataset. As a result, more information will get from one sample, but repeatability of information and similarity between different classes will be higher. There are some methods proposed to get temporal information from video. First is the emergence of 3D feature extractors, and the scholars propose to align and enhance the temporal information. But the complex network and huge amount of computation consume too many resources. Moreover, some scholars use traditional 2D feature extractor in few-shot action recognition in video, and have achieved good results, which indicates that it is worthwhile to further tap the potential of 2D methods.

It is not difficult to find that previous methods seldom pay attention to domain shift and hubness in few-shot learning. Some researchers have proved that a smooth manifold could alleviate these two problems by smoothing decision boundary, improving robustness of model to noise and maintaining the manifold structure.

In this paper, we propose a new few-shot action recognition model, which is based on deep feature and manifold feature. With the use of two types of features to increase discriminative information, our method can alleviate the problem of insufficient sample's information in few-shot learning. And the operation of smoothing manifold structure could better alleviate domain shift and hubness. In addition, the model uses 2D convolutional neural network as feature extractor with lower source consumption. Finally, the proposed model achieves a good performance, especially in the task of one-shot action recognition in video.

This work was supported in part by the National Natural Science Foundation of China(62176031,61772093), and the Fundamental Research Funds for the Central Universities under Grant No. 2021CDJQY-018.