第一人称动作识别技术报告——小数据集的动作识别模型探究

Bit-计算机视觉-2022.12

张荐科 1120200784　　袁奕晨 1120200776

Abstract

视频动作识别是视频理解的代表性任务之一，其中更具特点的第一人称动作识别在可穿戴设备上有着广泛的应用。随着深度学习的出现，视频动作识别得到了巨大进步。与此同时我们也遇到了新的挑战，包括如何建模时间信息，如何解决高计算成本和高数据依赖，以及由于近年更受重视的模型迁移能力。在本文中，我们对如何通过在相对较小的第一人称动作数据集（*9GB*）上使用轻量模型进行时空建模进行了探究。建立了几个小型的轻量化的模型，用于提取视频中的时间和空间特征，然后在此基础上对其进行分类任务的训练。在测试集中使用训练好的轻量模型进行预测，评估比较了每种模型的准确率，*F1-measure* 以及过拟合情况，基于此分析了不同模型的建模特点。最后我们对实验结果进行了分析比较，对小型视频动作数据集上的训练给出了一些经验性的启发建议。

1. 研究背景

视频动作识别一直是 CV 领域研究的重点。在深度学习广泛使用之前，表现最好的算法是 2013 年左右出现的 iDT(improved Dense Trajectories) [8]，之后的工作基本上都是在 iDT 方法上进行改进。iDT 算法基本思路是利用光流场来获得视频序列中的一些轨迹，再沿着轨迹提取 HOF，HOG 等 4 种特征，对特征进行编码后训练 SVM 分类器用于动作识别。深度学习方法出现后，有了 Two-Stream [5]、Conv3D 网络 [7]，以及后来的各种基于时序模型的方法（RNN、ViT）等。

对于第一人称动作识别，早先的非深度学习算法是 [4] 提出的 “multi-channel kernels” 用于提取第一人称视频的融合了时间和空间结构的特征，这是一种基于传统计算机视觉的实现方法。使用的数据集是 JPL First-Person Interaction，这是一个非常小的只有 64 个视频 9 类动作共 9GB 的小型数据集。在后续的采用深度学习的第一人称动作识别中，总体的基线架构与常规动作识别类似，包括 [6] 等等。但就最近两年来说，第一人称动作识别的研究并不多，更高的热度集中在通用视频的时空建模方面。

任务中我们希望通过深度学习模型对第一人称视频进行时空建模，但考虑到我们本地计算能力和时间的限制（期望能够在单张 2060 显卡训练几十个 epoch），我们采用了工作 [4] 中使用的 9GB 小型第一人称动作数据集。不过由于深度学习中很多模型都是高度依赖数据的，因此如何在小型数据上得到较好的训练效果依旧是十分具有挑战性的工作。

2. 视频理解模型

我们采用深度学习的方式对第一人称动作视频进行特征提取，然后输入到分类器中进行动作分类。为了更好的提取视频信息，模型中需要具备两个方面的能力：

- 空间特征提取（后简成 Spatial）：视频每一帧内部像素的空间信息，这一部分是图像理解中的核心内容，已经具有了相当成熟的模型算法（如卷积网络，ViT 等等）
- 时间信息提取（后简成 Temporal）：视频同一帧和其他帧之间的时序信息，这一部分偏向于 NLP 领域中的语义理解，但是由于图片的性质，其时序信息更加丰富，因此模型也会更加庞大复杂

另外，考虑到使用的数据集相对较小，计算资源较少，因此模型的计算量应该大小适中，同时设计合适的机制避免过拟合。在后续实验中我们发现在一些具有复杂机制的架构下小的 batch 等因素会导致模型不容易收敛，因此还需要再其中加入合适的促进收敛的模块。

基于上述条件，可以将两类特征提取需求的实现划分为以下几种方案：

1. 直接使用连接的方式融合同一个视频的每一帧内部元素和帧之间的所有元素，这种方式在实现中就是通过 flatten 或者 cat 的方式对特征矩阵进行拼接。
2. 使用 2D 卷积网络提取图片内部的空间信息，然后将不同帧的空间信息拼接融合，再提取时间信息。这种方式直观上就是将多个图片特征进行按时间序列拼接（可以加入一些 positionEmbedding 用于描述时间序），然后将整体的融合特征作为分类器输入。
3. 使用 3D 卷积网络提取视频时空信息，3D 卷积将时间维度作为长、宽外的另一个尺度，使用高一维的卷积核可以描述图片局部范围内空间和时间的共同特征。这一方案与 VGG 等卷积网络的思路一致，即将卷积核作为特征分类依据，只不过其中不仅包含了空间信息，也包含了时间信息。
4. 将提取好的空间信息作为时序模型的输入，利用如 RNN/LSTM 等传统时序模型对提取好的空间特征进一步处理，挖掘其中的时间信息。最后将整体的特征信息作为分类依据。不过由于 RNN 等模型相对来说更难训练，并且计算成本很高，因此这一方案不太适合在小型数据集上进行训练。
5. 采用自注意力机制的时空序列模型。这一方案是由 Transformer 中自注意力模块衍生而来，ViT 模型 [3] 的出现提供了一种新的特征提取方案，尤其是对于具有位置关系的信息。Vivit [1] 是紧随其后的一个用于视频理解的 ViT 模型。这一方案在已有的研究中报告可以得到 71% 的 rank1 准确率。但是训练需要很大的数据集并且调参难度较高，大部分的使用都是基于迁移学习的。

下面就我们采用的三种方案进行详细阐述。（重要的代码实现详见 Sec. 3.2）

2.1. 简单多层感知机——MLP

对每个动作，将该视频中的若干帧图像提取后拼接压缩为 1 维向量，作为传统 MLP 的输入，本次实验中 MLP 共使用 3 层线性层。

2.2. 时空卷积网络——Conv3D

将各个帧的图像拼接后，利用 Conv3D 可以同时提取该数据的时空信息，该操作得益于 3 维卷积的计算过程，如图 Sec. 2.2所示。在图示的卷积过程中，可见该卷积核同时提取若干帧的相同区域图像特征，该行为关注了图像之间的时间关系。

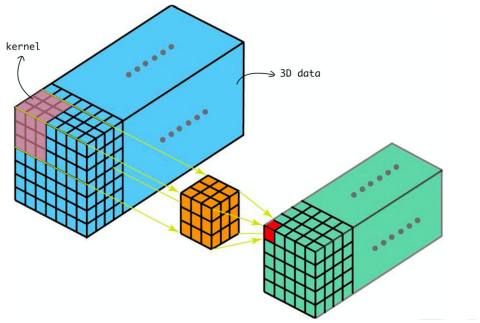


图 1. Conv3D 卷积计算过程

在模型架构方面，使用传统的 CNN 模型，对每个卷积层都采用类似 “卷积层-> 激活函数-> 池化层->BN 层” 的结构，最终模型由两个如此的卷积层与结果的线性分类层拼接而成。

2.3. 时空自注意力机制——ViT

采用自注意力机制的时空序列模型。这一方案基于 Vivit [1] 和 [2] 的工作。在我们的任务中，考虑到第一人称视角的动作环境中有很多信息是不重要的，更需要关注的是镜头中央的手部和人体的动作，因此可以减小输入网络的宽度（ViT 每一层的宽度）。另外，在原本 ViT 的空间自注意力模块的基础上加入时间自注意力模块，让每一帧分出的 16×16 大小的 patch 不仅仅与图片内部的其他 patch 进行注意力计算，还要让这个 patch 与其前后所有帧的对应位置进行计算。

具体来说，给出 f 个视频 $\mathcal{V} = \{v_1, v_2, \ldots, v_f\}$，第 i 个视频可以分为 $|v_i|$ 个有序的图像帧 $\{v_{i,1}, v_{i,2}, \ldots, v_{i,|v_i|}\}$。对视频 v_i，将第 j 帧 $\mathbf{v}_{i,j} \in \mathbb{R}^{H \times W \times C}$ 拆分为 $P \times P$ 大小的像素块，记为 $x_{i,j} \in \mathbb{R}^{N \times P^2 \times C}$，其中 $N = \frac{H}{P} \times \frac{W}{P} = h \times w$. 之后对这些块分别使用线性层进行嵌入，并加入体现空间特征的位置嵌入向量作为视频样本 i 的第 j 帧给 ViT 第一层的输入：

$$\mathbf{p}_{i,j,0} = Linear(x_{i,j}) + PosEmbedding$$

另外，使用一个可学习的 [CLS] 块加入的每个视频中（最终一个视频一共对应了 $f \times P \times P + 1$ 个 patch。这个块一般用于作为整个视频的最终特征，但是在实验中发现单独的一个 patch 大小的特征性能很差（数据像素太少有关），在输出时我们将这 $f \times P \times P + 1$ 个 patch 的输出特征拼接在一起输入到尾部的线性层进行分类。

在 ViT 每层内部，有一个前馈网络残差块（Feedforward）和两个多头注意力残差快，其中前馈网络就是由线性层，激活层和一个 LayerNorm 层（和一个 Dropout）组成的模块。网络中使用了两个多头注意力残差块，分别用于学习空间特征和时间特征。

- Spatial-Attn：每一帧内部的 $P \times P$ 个块进行自注意力计算（CLS 块单独计算）
- Temporal-Attn：不同帧之间图片的相同位置 f 个块进行自注意力计算（CLS 块单独计算）

对输入到第 l 层第 i 个视频全部 f 个帧的输入 $\mathbf{p}_{i,l}$，计算其该层的输出 $\mathbf{p}_{i,l+1}$：

$$\begin{aligned}
\widehat{\mathbf{p1}}_{i,l+1} &= \text{TemporalAttn}(\text{LN}(\hat{p}_{i,l})) + \hat{p}_{i,l} \\
\widehat{\mathbf{p2}}_{i,l+1} &= \text{SpatialAttn}(\text{LN}(\hat{p1}_{i,l}) + \hat{p1}_{i,l} \\
\mathbf{p}_{i,l+1} &= \text{Linear}(\text{GeLU}(\text{Linear}(\text{LN}(\hat{p2}_{i,l})))) + \hat{p2}_{i,l}
\end{aligned}$$

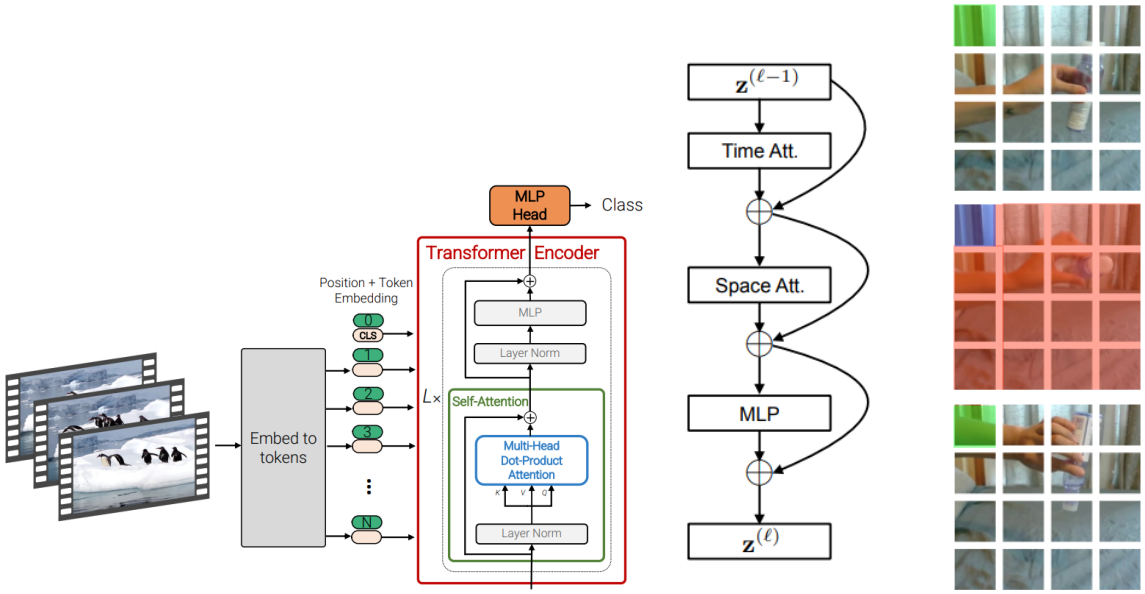
在最后一层，将一个视频的所有特征使用线性层投影到分类得分矩阵

$$\boldsymbol{Logits}_i = \text{Linear}(\text{LN}(\mathbf{p}_{i,L}))$$

3. 实验

3.1. 数据预处理

对一个动作所对应的视频来说，因为帧数过多，需要人为地截取若干帧作为训练的使用。具体地，使用等距截取法，在该视频中尽可能地均匀且相隔较远地截取若干帧，该帧数在处理之前给定，本次实验中设定为 4。



(a) 采用的 ViT 模型架构（图片取自 [1]）　(b) 单层计算 [2]　(c) 注意力 [2]

图 2. 时空 ViT 模型可视化

3.2. 模型细节实现

3.2.1 MLP

在本次实验的 MLP 模型中，该模型由 3 层线性层组成，输出通道数分别为：120, 84, 9。其中 9 为数据集提供的动作种类即分类器的输出维度。

优化器使用 AdamW 优化器，具体参数设置如 Tab. 1：

参数	数值
epoch	50
lr	10^{-3}
weight_decay	10^{-5}

表 1. MLP 参数设置

3.2.2 Conv3D

在本次实验的 CNN 模型中，该模型由两层相似的卷积结构拼接而成，每个卷积结构均由 Conv3d, ReLU, MaxPool3d, BatchNorm3d, Dropout 组成，其中卷积核大小均设为 3，$p_{dropout} = 0.3$。最后使用一个两层的 MLP 进行分类，第一层输出通道为 256，第二层则为种类数 9。

优化器使用 AdamW 优化器，具体参数设置如 Tab. 2：

参数	数值
epoch	50
lr	10^{-3}
weight_decay	10^{-4}

表 2. CNN 参数设置

3.2.3 ST-ViT

根据 Fig. 2的模块设计，其他模块比较简单，因此这一部分中主要介绍两个 attention 层的实现方式。

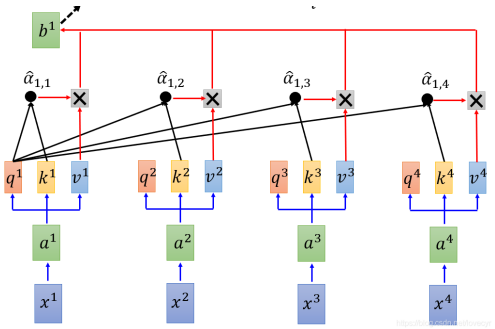


图 3. 注意力头的计算

输入到 attention 层的是整个视频的全部 patch 的特征，即 $x \in \mathbb{R}^{B\times N\times D}$，其中 B 为 batch-size，N 是一个视频的所有 patch 数量（$N = F \times \frac{HW}{P^2}$），$D$ 是 ViT 的宽度。Attn 层中首先利用线性层切分生成 Q、K、V，然后分别根据 attn 的类型移动维度，在最后两维上做自注意力机制中的查询加权操作。具体来说，首先通过线性层映射并切分得到 q、k、v：

```
q,k,v = self.linear_qkv(x).chunk(3, dim=-1)
```

随后将要做注意力计算的维度挪到最后两维中：

- Spatial-Attn：矩阵形状变换

  $(B, F \times \frac{HW}{P^2}, D) \to (B \times F, \frac{HW}{P^2}, D)$

- Temporal-Attn：矩阵形状变换

  $(B, F \times \frac{HW}{P^2}, D) \to (B \times \frac{HW}{P^2}, F, D)$

最后按照正常的自注意力机制模块进行计算：

$$\alpha_{i,j} = \frac{q_i k_j^T}{\sqrt{d}}$$

$$y_i = \sum_j softmax(\frac{q_i k_j^T}{\sqrt{d}}) v_j$$

值得注意的是，实际代码中输入的 $x \in \mathbb{R}^{B\times F\times(\frac{HW}{P^2}+1)\times D}$，其中多了一个 [CLS] 特征，因此在上述对 QKV 的处理中还需要将 [CLS] 取出单独与其他所有不包含 [CLS] 的图像特征进行自注意力计算。

3.3. 训练与评价

这一部分中我们将 Sec. 2实现的三种模型进行训练，具体流程细节如下：

1. 查看是否由训练好的模型参数，有则加载继续训练
2. 将模型加载的合适的 device（CPU 或者 CUDA）上，并设置为 model.train()
3. 按照 epoch 和加载器为模型加载多轮数据 batch，每次使用 model 前向传播得到 logits 分类得分。将得分与正确类型标签比较，采用 CrossEntrophy 交叉熵作为损失函数
4. 采用 Adam 作为优化器，优化器内梯度零，loss.backward 反向传播，梯度更新
5. 打印相关准确率信息，根据结果保存模型参数（包含 model 参数和 optimizer 梯度参数），用于迁移预测或下次继续训练

训练完成后使用验证集进行模型预测：

1. 加载训练好的模型参数
2. 将模型加载的合适的 device（CPU 或者 CUDA）上，模型中可能含有 dropout 和 batchnorm 层，注意设置 model.eval()，
3. 按照加载器为模型批量加载验证集数据，每次使用 model 前向传播得到 logits 分类得分。
4. 取出得分中最高的索引作为分类类别，与正确分类进行比较，计算准确率，召回率等评价系数
5. 打印相关准确率信息

4. 结果及分析

将上述全部过程实现后，我们尝试调试超参数达到更好的效果，具体结果如表 Tab. 3 可以看

模型	训练集准确率	验证集准确率
MLP	0.766	0.304
Conv3D	1.000	0.429
ST-ViT	0.487	0.392

表 3. Results

到，训练效果最好的是 Conv3D 模型，而 MLP 模型和 Conv3D 模型的验证集准确率远低于训练集，说明训练过程中存在着严重的过拟合现象。相比之下，ST-ViT 模型则差距不是很大。这充分体现出了 ViT 学习时空信息能力强于前两者，它所具有的信息迁移能力远高于传统卷积网络。这一结果实际上也是必然的，因为卷积网络本质上是带有对于图片信息局部空间的归纳偏置的（即人们已经在模型设计中蕴含了对于图片信息的特征），而大型的 ST-ViT 模型则不具有这种归纳偏置。后者的短板一方面在于很难使训练收敛，（在实验中尝试了很多超参数才得到一个相对好的结果），另一方面在于对小数据集的训练能力很弱。但是在更广泛的大型模型上，我们认为 ST-ViT 拥有者更强大的时空信息提取能力，这是只能局部提取信息的 Conv3D 所不具备的优势。

5. 总结

在本任务中我们针对第一人称动作识别任务进行了模型实现。完成了比较高准确率的预测。同时针对三个不同类型架构的模型在视频动作识别的表现上进行了比较分析。一方面 Conv3D 的卷积网络在结果上有着更强的优势，这得益于其得天独厚的卷积归纳偏置的优势，另一方面作为时序模型新星的 transformer 在视觉领域的实现 ViT 在改造后具有了很强的空间时间信息提取能力，具有着很大潜力。

另外，值得注意的是我们使用的是相对传统大型视频识别模型的小数据集进行训练，因此不论是训练策略合适模型参数架构设计都进行了不少调整。MLP 和 Conv3D 能够快速将小数据集进行过拟合，而相对更大的 ST-ViT 则十分难以训练。他所包含的大量隐藏层单元很容易在高层的反向传播中梯度消失或爆炸。不过在更大的数据集上进行训练 ViT 很有可能达到或者超过 Conv3D 模型的效果，而且由于其更全局的信息提取能力，它能够更轻易的迁移到其他下游任务中，这一点也是卷积网络所不具备的优势。

最后，我们认为可以在小型数据集中采用 Conv 提取空间信息，ViT 融合空间和提取时间信息的方式。这样既能够利用 Conv 的空间提取优势，同时也能够让模型顾及更广泛的区域并构建更容易迁移的小模型。

参考文献

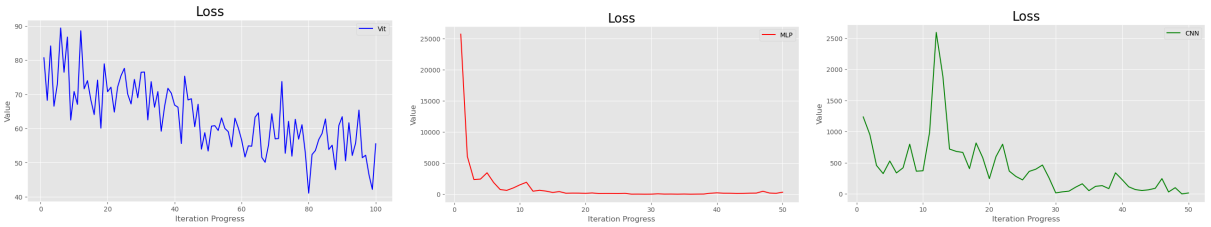
[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6836–6846, 2021. 2, 3, 4

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 3, 4

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[4] Michael S Ryoo and Larry Matthies. First-person activity recognition: What are they doing to me? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2730–2737, 2013. 1

[5] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recogni-



(a) ViT 的损失变化　(b) MLP 的损失变化　(c) Conv3D 的损失变化

图 4. 三个模型的 loss 变化趋势

tion in videos. *Advances in neural information processing systems*, 27, 2014. 1

[6] Suriya Singh, Chetan Arora, and CV Jawahar. First person action recognition using deep learned descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2620–2628, 2016. 1

[7] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 1

[8] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 1