arXiv:2303.01589v1 [cs.CV] 2 Mar 2023

# AZTR：采用自动缩放和时序推理的航拍视频动作识别

Xijun Wang*[1], Ruiqi Xian*[2] , TianruiGuan[1],Celso M.de Melo[3],Stephen M.Nogar[3], Aniket Bera[4] 和 Dinesh Manocha[1]

**摘要—我们提出了一种新颖的航拍视频动作识别方法。该方法专为无人机拍摄的视频设计，可在边缘设备或移动设备上运行。我们提出了一种基于学习的方法，利用定制的自动缩放自动识别人体目标并调整至合适尺度。这使得提取关键特征更加容易，并降低了计算开销。我们还提出了一种高效的时序推理算法，能够以可控的计算成本捕获空间和时间域中的动作信息。我们已在配备高端图形处理器的桌面计算机上，以及用于机器人和无人机的 Robotics RB5 平台上实现并评估了该方法。在实践中，我们的方法在 RoCoG-v2 数据集上的 Top-1 准确率较最先进水平提升了 6.1–7.4%，在 UAV-Human 数据集上提升了 8.3–10.4%，在 Drone Action 数据集上提升了 3.2%。**

## I. 引言

使用无人机和 UAV（无人驾驶飞行器）拍摄人类的航拍视频，正受到广泛关注。这也带来了与空中数据中的检测、跟踪、识别、行人再检测和动作识别相关的挑战性问题 [1]。本文主要研究航拍视频识别问题，并提出可在低功耗或边缘硬件上良好运行的解决方案。

近年来，已有许多深度学习方法被提出用于视频动作识别。尽管这些方法在地面摄像机视频上取得了巨大成功，但将其直接应用于无人机摄像机拍摄的视频时，准确率却会大幅下降。这是由不同视角和摄像机特性造成的域偏移所致。一些挑战包括：

低分辨率： 由于摄像机高度较高，航拍数据中的目标人体演员显得明显更小。从空中拍摄时，背景覆盖范围更广，占据视频帧中的大部分像素，而对应于人体动作的像素只占很小一部分。

多尺度： 根据无人机的飞行高度，人体主体的大小和尺度可能会有显著差异。这种差异使得难以提取用于模型训练的特征，从而降低整体准确率。

*这些作者贡献相同[1]作者来自美国马里兰州 College Park 的马里兰大学计算机科学系。xijun@umd.edu[2]作者来自美国马里兰州 College Park 的马里兰大学电气与计算机工程系rxian @umd.edu[3]作者来自美国马里兰州 Adelphi 的 DEVCOM 美国陆军研究实验室计算与信息科学部[4]作者来自美国印第安纳州 West Lafayette 的普渡大学计算机科学系。ab@cs.purdue.edu

移动摄像机： 由于无人机移动，人体主体在视频中的位置可能会不断变化。与人体行为变化相比，无人机的运动会导致背景发生更多变化。这使得模型更多地根据背景而非人体主体进行推断，尤其是在高分辨率视频中。

此外，通用动作识别方法大多是为桌面计算机或云端图形处理器设计的，其内存或功耗要求较高。它们无法部署在内存有限且不支持复杂算术运算的移动或边缘设备、无人机平台上。当前面向移动平台的识别方法 [2][3][4]针对地面摄像机视频设计，无法在空中数据上取得出色表现。因此，我们需要更好的航拍视频动作识别技术。

### A. 主要贡献

我们提出了一种用于视频动作识别的新型深度学习方法，其中包括一种新的自动变焦算法，可高效识别目标，并将目标区域缩放到适合给定设备或处理器内存的大小。我们的自动变焦算法采用了一种适用于低分辨率、多个尺度和移动摄像机的高效技术。此外，从长程时空关系的角度出发，我们引入了一种时序推理算法来捕捉动作信息。我们工作的创新之处包括：

1) 我们为航拍视频动作识别模型设计了一种自动变焦算法。该算法能够高效地应用自动对焦、裁剪和缩放策略，从人体主体中提取关键动作信息。聚焦视图会根据人体主体的位置逐帧变化，并相应补偿无人机的运动，使人体主体始终位于视频中央。这能减少背景噪声，提取更多有用特征用于人体行为分析，从而增强模型的鲁棒性。

2）我们提出了一种时序推理算法，结合卷积与注意力机制，以实现更高的准确率。为了兼顾准确率与推理速度，我们在高端桌面 GPU 上进行三维卷积，在低功耗边缘设备或无人机平台上进行（2D+1）卷积。该注意力算法由交叉注意力和自注意力组成，可提供空间和时间表征，且计算复杂度为线性。总体而言，我们的方法能够获取长程

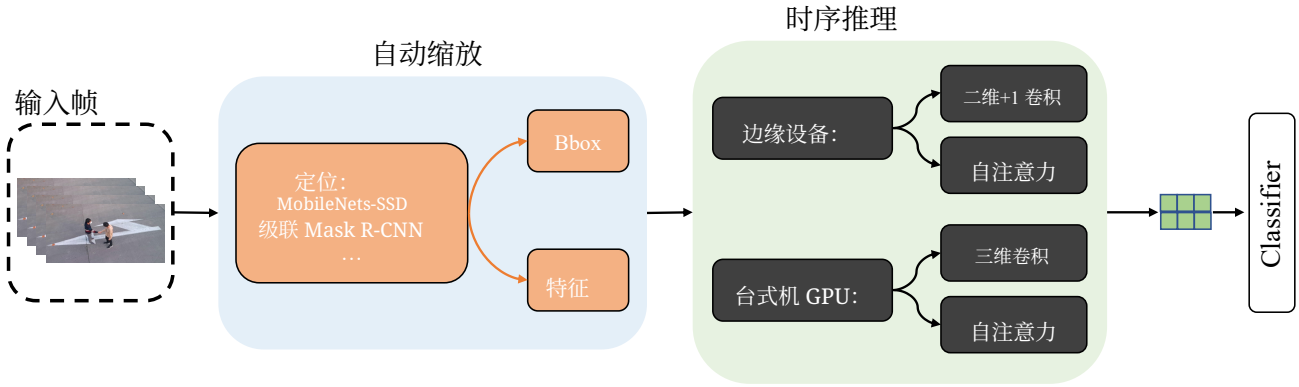



图1：我们的学习流程由自动缩放学习算法和时序推理算法组成。对于自动缩放学习，我们提供了不同的边界框（bbox）和特征操作。详情请参阅第III节。对于时序推理算法，我们在边缘设备上执行（2D+1）卷积，在台式机 GPU 上执行三维卷积，并在边缘设备和台式机 GPU 上都采用自注意力（Atten）机制。台式机 GPU 上的注意力层更深、更宽。

空间—时间关系，从而更好地理解视频中的动作。

我们提出的学习方法既可适配高端图形处理器，也可适配低功耗边缘设备。我们在边缘设备 Qualcomm Robotics RB5 平台（Kryo 585 CPU 和 Adreno 650 GPU）以及台式机 GPU（Nvidia RTX A5000 GPU）上实现并评估了该方法。评估结果显示，在 RoCoG-v2 数据集上，我们的 Top-1 准确率比最先进水平提升了 6.1–7.4%；在 UAV-Human 数据集上提升了 8.3–10.4%；在 Drone Action 数据集上提升了 3.2%。我们的方法在 RB5 CPU 上可达到 40.2% 的准确率，推理时间为 56.5 毫秒，在准确率和速度方面均优于 SOTAMoViNets。

## II. 相关工作

### A. 航拍视频识别的学习方法

深度学习方法的最新进展提升了地面摄像机视频数据集上的动作识别性能[5], [6], [7], [8], [9], [10], [11]。然而，这些方法在航拍视频上的准确率并不理想[1]。[12], [13], [14], [15] 使用 ResNet 和 InceptionNet 执行单帧分类，并融合所有输出进行识别。 [16], [17], [18] 利用双流卷积神经网络提取外观和运动属性。三维卷积神经网络也广泛应用于航拍动作识别。 [19], [4], [20], [15], [21]采用 I3D[22] 提取时空特征，以实现更准确的识别。 [23] 通过三维卷积改进 Inception-ResNet 模型，使其更适用于航拍视频处理。其他技术则采用基于 Transformer 的解决方案。 [24] 将傅里叶变换与注意力机制相结合，以实现更好的特征提取。我们的方法可以作为这些方法的补充，提高航拍动作识别的整体准确率。

### B. 无人机与无人机数据集

为了推动进一步研究，人们使用价格实惠的现成无人机采集了许多无人机数据集。

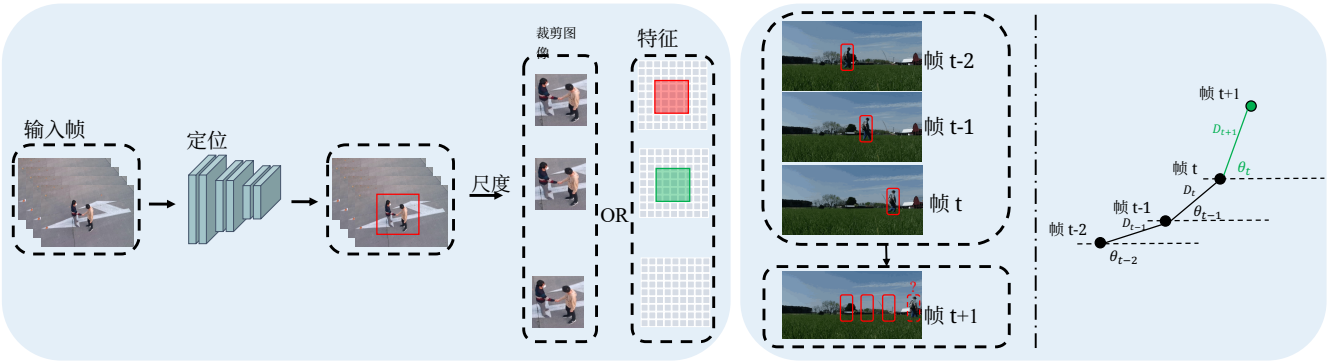
其中包括许多公开的无人机数据集，涵盖大多数基于无人机的任务，例如人体检测、目标跟踪、行人再识别和动作识别 [20], [25],[26], [27], [17], [19], [28]。Drone Action[17] 是一个户外无人机视频数据集，包含从低空、低速条件下拍摄的 240 个高清视频片段，涵盖 13 种动态人体动作。UAV-Human[20] 是一个用于无人机人体行为或动作理解的大型基准数据集，其中包含 67,248 个多模态视频序列和 119 名受试者，用于动作识别；22,476 帧用于姿态估计；41,290 帧和 1,144 个身份用于行人再识别；以及 22,263 帧用于属性识别。

### C. 边缘架构上的活动识别

大多数深度学习方法采用规模较大、层数较深的架构来获得更高的准确率，但这会显著增加计算成本。因此，这些方法通常在桌面计算机或云端 GPU 上运行，不适用于边缘设备或低功耗移动设备。采用轻量化方法提高视频模型的效率，正受到越来越多的关注 [29], [30], [31], [32], [33]。MobileNet[34]和 Yolo[35] 是广泛用于边缘设备视频建模的网络。 [36] 提出了一种在无人机上使用 MobileNet、结合焦点损失与自注意力机制的轻量级动作识别模型。 [3] 介绍了移动视频网络（MoViNets），这是一系列计算高效且内存占用低的视频网络，可处理流式视频以进行在线推理。 [37] 提出了 Tiny Video Networks，这是一种自动设计的高效视频架构。 [38]介绍了时序移位模块（TSM），该模块沿时间维度移动通道，支持离线和在线视频识别。我们的自动缩放算法可以与上述任意方法结合，在资源需求有限的情况下实现更高的准确率。

## III. 我们的方法：AZTR

我们提出了一种通用的航拍视频动作识别学习方法，可自动识别人体目标、适当调整其尺度，并通过




（a）基于裁剪图像或特征的自动缩放　　(b) 边界框偏移

图 2：我们针对高端台式机以及移动设备或边缘设备，分别设计了两种不同的自动缩放方法，分别使用裁剪图像或特征。(a) 对于使用裁剪图像的自动缩放，我们使用检测器获取目标边界框，并从原始帧中裁剪出目标，然后调整裁剪图像的大小。对于使用特征的自动缩放，我们利用特征生成边界框并进行分类。(b) 我们在关键帧上使用检测器生成边界框，以降低计算成本。我们预测下一关键帧中的边界框，并比较预测边界框与生成边界框的位置，以避免错误的检测结果。最后，我们通过线性插值生成关键帧之间的边界框。详情见第III节。

在时间维度上对输入帧进行推理来分析动作。为实现这些功能，我们引入了两个组件：自动变焦算法和时序推理算法，可供任何视频模型使用，以高效提取空间和时间特征。

### *A.* 整体学习方法

如图 1 所示，我们的方法以采样视频帧作为输入。首先，这些帧通过自动变焦算法提取关键空间信息，然后通过时序推理算法获取动作时间信息。对于自动缩放组件，我们使用目标定位器获取稀疏边界框（bbox）或相应的特征。通过对稀疏边界框进行边界框/特征移位，我们可以降低检测频率，从而节省计算成本。此外，我们使用这些边界框在原始视频中对齐目标，或使用这些特征序列进行推理。对于时序推理模块，我们针对边缘设备（内存和功耗要求较低）和台式机 GPU 采用不同的方法。在边缘设备上，时序推理模块可以使用 2D+1卷积和注意力机制。在台式机 GPU 上，则使用注意力机制和 3D 卷积来获得高性能。

### *B.* 自动缩放

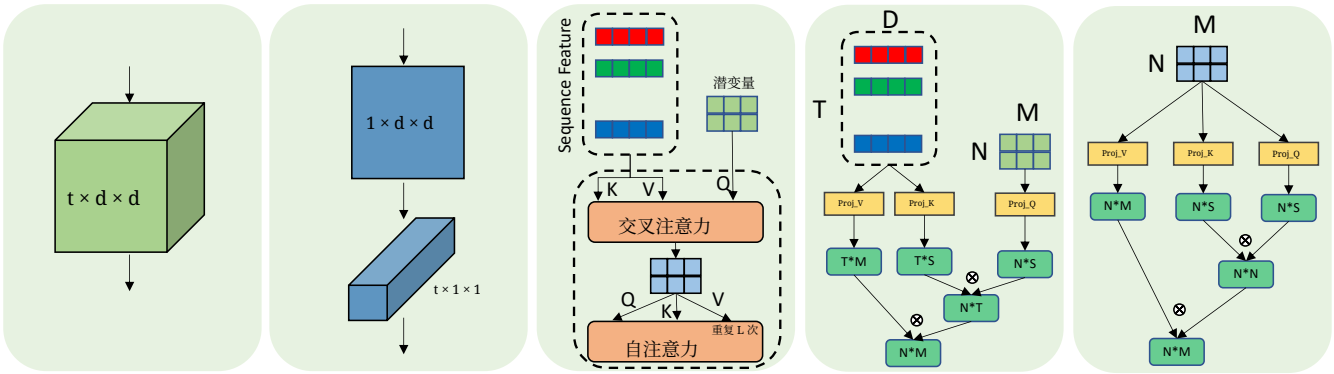
我们为航拍视频动作识别模型提出了一种自动变焦算法。如图 2 所示，该算法首先识别并定位目标，然后裁剪目标区域，并将其缩放至模型的输入尺寸，以便更轻松地提取关键特征并适配硬件资源。由于高空拍摄的空中数据中，目标物体显著更小，我们的自动变焦算法可以自动采用放大策略，从目标物体中获取更多动作信息。例如，在 UAV-Human 数据集中，人体主体仅占每帧像素的 2% 至 5%。采用我们的自动变焦算法后，人体主体所占像素约为每帧的 16% 至 22%。这能提供更多局部细节，使模型主要关注动作，而非背景。此外，我们的自动变焦算法还能减少无人机移动造成的噪声和异常值。放大视图会根据目标物体的位置逐帧变化，从而抵消无人机的运动，使目标物体始终出现在视频画面中央。这能提取更多有用特征，并增强模型的鲁棒性。

当前最先进水平的检测器采用宽而深的网络结构来获得高性能，但在推理时需要大量计算资源。为了在获得相同准确率的同时提高效率，我们的自动变焦算法分为以下三个步骤：

（1）裁剪区域的大小以动态方式变化。裁剪区域的高度和宽度会从 $[480, 640, 720, 960]$ 中动态选择。这不仅取决于原始视频分辨率，还取决于边界框的大小。我们首先计算人体主体的边界框包含多少像素，然后选择合适的尺寸，使人体主体占据该区域的 15% 至 20%。根据我们的实验，我们发现这一比例较为合适，因为它既能展示人体主体的更多细节，也能呈现周围环境的信息。这确保模型拥有足够的信息来推断人体主体与周围环境之间的关系。最后，我们将目标区域缩放至模型输入尺寸，用于训练。

（2）我们仅在关键帧上使用检测器进行推理。由于无人机数据集中的大多数视频都是以高帧率采集的，相邻帧中只有极少部分像素的值会发生变化。因此，没有必要为每一帧获取边界框，因为结果可能只会相差几个像素。为此，我们仅为 10% 或 20% 的帧生成边界框。这些帧就是关键帧，将作为第 3 步中边界框偏移的锚点。这样可以显著降低计算成本，并为识别留出更多资源。

(3) 初始化时，我们均匀分布关键帧，




(a) 3D (b) 2D + 1 (c) 注意力算法 (c.1) 交叉注意力 (c.2) 自注意力

图 3：我们在台式机 GPU 和边缘设备上，采用 2D+1 卷积、3D 卷积和高效 Transformer 的不同组合，以实现时序推理。基于高效 Transformer 的算法包含两个组件：交叉注意力用于将输入序列映射到具有特定大小的新序列，以满足计算成本要求；自注意力则是 Transformer 中的常规组件。

从而得到固定的步长。然而，由于预训练检测器（例如 MobileNet、Cascade Mask RCNN）在初始关键帧上的表现可能不佳，因此可能会生成错误的边界框。为避免这种情况，我们只保留置信度高于 0.8 的边界框，以过滤掉可能性较低、其中不包含人体主体的边界框。然后，我们根据前 3 个关键帧生成的边界框，使用公式 (1) 预测关键帧 $t$ 中边界框的位置。

$$\begin{aligned}
&\begin{pmatrix} x_{t+1} \\ y_{t+1} \end{pmatrix} = \begin{pmatrix} x_t + (D_t + \delta_{D_t}) \cdot \cos(\theta_{t-1} + \delta_\theta) \\ y_t + (D_t + \delta_{D_t}) \cdot \sin(\theta_{t-1} + \delta_\theta) \end{pmatrix}, \\
&\delta_{D_t} = D_t - D_{t-1}, D_t = distance\left(\begin{pmatrix} x_{t-1} \\ y_{t-1} \end{pmatrix}, \begin{pmatrix} x_t \\ y_t \end{pmatrix}\right), \\
&\theta_{t-1} = \arctan(\frac{y_t - y_{t-1}}{x_t - x_{t-1}}), \delta_\theta = \theta_{t-1} - \theta_{t-2}.
\end{aligned} \tag{1}$$

$x_t$ 和 $y_t$ 是关键帧 $t$处边界框的坐标。$D_t$、$\theta_t$ 分别表示关键帧 $t-1$ 和 $t$之间的位移和角度。$\delta_{D_t}$ 和 $\delta_{theta}$ 表示帧 $t$处位移和角度的偏差。

接下来，我们计算预测边界框的位置与检测器生成的边界框位置之间的欧氏距离。如果距离在设定的阈值范围内，我们就认为这是正确的检测结果。如果距离超过阈值或没有检测结果，我们就将预测边界框指定为当前关键帧的边界框。对于关键帧之间的普通帧，我们采用线性插值来获得边界框。

### *C.* 时序推理

我们提出了时序推理算法，通过分析时间序列信息来推断动作。时序推理算法有三种不同的实现方式，分别采用 2D+1 卷积（conv）、三维卷积和注意力机制。2D+1卷积和注意力机制可用于在边缘设备上运行的高效模型。三维卷积和注意力机制可用于在性能出色的图形处理器上运行的高性能模型。对于基于 2D+1 卷积的时序推理算法，我们首先利用 2D 卷积根据每一帧的空间信息生成特征，然后执行 1D 卷积来融合所有空间特征，使模型能够从时序信息中学习。对于基于三维卷积的时序推理算法，则使用三维卷积同时处理空间和时序信息。接下来，我们将进一步介绍基于注意力机制的时序推理算法，其中包括交叉注意力和自注意力。

基于注意力机制的时序推理算法由 QKV（查询、键和值）注意力层组成。QKV 注意力机制将每个输入编码为所有其他输入的函数 [39], [40]。注意力函数的输出是各个值的加权和，其中分配给每个值的权重由查询与相应键之间的兼容性函数计算得到[39]。如图 3 所示，基于注意力机制的时序推理算法包含两个组件：交叉注意力和自注意力。交叉注意力用于根据计算成本要求，将输入序列映射为具有特定大小的新序列。自注意力是 Transformer 的标准组件[39]。交叉注意力和自注意力的区别在于，交叉注意力有两个输入 $X_Q \in \mathbb{R}^{N\times M}$ 和 $X_{KV} \in \mathbb{R}^{T\times D}$，而自注意力只有一个输入 $X_{QKV} \in \mathbb{R}^{N\times M}$。$T, D, N, M$ 分别表示输入视频帧、每个帧的嵌入维度、查询序列数量以及每个查询序列的维度。对于交叉注意力，$X_Q$ 是查询输入，$X_{KV}$ 是键和值输入。对于自注意力，查询、键和值共享同一个输入 $X_{QKV}$。输入将首先通过其投影函数 $p(\cdot)$，

$$F_Q, F_K, F_V = p(X_Q), p(X_{KV}), p(X_{KV}), \tag{2}$$

其中，$F_Q \in \mathbb{R}^{T\times M}, F_K \in \mathbb{R}^{T\times S}, F_V \in \mathbb{R}^{N\times S}$用于交叉注意力机制，$F_Q \in \mathbb{R}^{N\times M}, F_K \in \mathbb{R}^{N\times S}, F_V \in \mathbb{R}^{N\times S}$用于自注意力，其中 $F_Q$、$F_K$ 和 $F_V$ 分别表示查询、键和值的投影特征。投影函数

| 平台 | C3D | MP3D | AP3D | DC3D |
|---|---|---|---|---|
| TL（中央处理器）[41] | ✓ | × | × | × |
| TL +NNAPI（中央处理器/图形处理器/数字信号处理器）[41] | × | × | × | × |
| Tensorflow + MNN（中央处理器/图形处理器）[41][42] | ✓ | ✓ | ✓ | × |
| TensorFlow Lite +MNN [41][42] | × | × | × | × |

表 I：如图所示，大多数边缘设备或处理器对三维算子的支持并不完善。因此，我们在 RB5 平台上使用二维+1 卷积和高效的注意力机制。TL：TensorFlow Lite，C3D：三维卷积，MP3D：三维最大池化，AP3D：三维平均池化，DC3D：深度宽度三维卷积

会将键和值的维度映射到 $S$。然后，我们可以根据信息需求和计算限制调整 $S$ 。接下来，我们计算注意力机制，

$$\mathrm{X_{QK}} = \mathrm{Attention}(Q, K) = \mathrm{softmax}\left(\frac{F_Q {F_K}^T}{\sqrt{S}}\right), \quad (3)$$

然后，我们将其应用于值，

$$\begin{aligned}\mathrm{Attention}\left(X_Q, X_{KV}\right) &= \mathrm{Attention}(Q, K, V) \\ &= X_{QK} V.\end{aligned} \quad (4)$$

时序推理算法的输入来自自动缩放，每个序列代表一帧的关键信息。与 CNN 推理方法相比，基于注意力的方法可以在模型的第一层获取整个目标的信息，从而在整个过程中提供更多信息。至于计算复杂度，主要计算量在于矩阵乘法。对于交叉注意力，我们对形状为 $T \times M$, $N \times T$ 和 $T \times S$， $N \times S$ 的矩阵进行乘法运算，因此复杂度可以表示为 $\mathcal{O}(TMNS)$。对于自注意力，复杂度为 $\mathcal{O}(MNS)$。如果有 $L$ 个自注意力层，总复杂度为 $\mathcal{O}((T+L)MNS)$。因此，时序推理算法的复杂度相对于输入时间维度 $T$ 和模型深度 $L$ 呈线性。此外，我们还可以通过控制键和值的投影维度 $S$ 来控制计算成本。

## IV. 实验

在本节中，我们在 3 个无人机数据集上评估我们的算法，并将其性能与其他最先进的视频动作识别方法进行比较。我们还评估了这些方法在边缘设备和台式机 GPU 上的性能。

**边缘设备：**我们使用搭载 Qualcomm Kryo 585 CPU 和 Qualcomm Adreno 650 的机器人平台（ Qualcomm Robotics RB5）。高效模型使用 TensorFlow 训练，并通过机器人操作系统 2（ROS2）Galactic 部署。

**台式机 GPU：** 我们使用配备 Intel Xeon W-2288 CPU 和 8 块 Nvidia RTX A5000 GPU 的高端台式计算机。我们使用 PyTorch 训练和测试高性能模型。

### *A.* 数据集

**RoCoG-v2 [43], [28]:** 我们使用了 99 段长篇原始视频，涵盖 17 个动作类别。我们将原始视频剪辑成 5,828 段时长为 1–5 秒的视频片段，每段片段对应一个动作类别。我们还获得了 70,124 帧带有边界框的图像，

| 方法 | 输入尺寸 | 每帧推理时间 |
|---|---|---|
| MoViNet A0 [3] | $172 \times 172$ | 33.2 毫秒 |
| MoViNet A2 [3] | $224 \times 224$ | 106.4 毫秒 |
| MoViNet A3 [3] | $256 \times 256$ | 124.0 毫秒 |
| **AZTR（我们的方法）** | $172 \times 172$ | **56.5 毫秒** |

表 II：RB5 CPU 上的推理时间。我们的方法平均每帧推理耗时 56.5 毫秒，比 RB5 上的 MoViNet A3 快 2 倍，同时 Top-1 准确率也有所提升，详见表 III。

| 方法 | 帧数 | 输入尺寸 | 初始化 | Top-1准确率 |
|---|---|---|---|---|
| MoViNet A0 [3] | 8 | $172 \times 172$ | None | 17.8 |
| MoViNet A0 [3] | 8 | $172 \times 172$ | Kinectics | 23.4 |
| MoViNet A2 [3] | 8 | $224 \times 224$ | Kinectics | 28.1 |
| MoViNet A3 [3] | 8 | $256 \times 256$ | Kinectics | 29.0 |
| **AZTR（我们的方法）** | 8 | $172 \times 172$ | Kinectics | **29.5** |
| MoViNet A0 [3] | 20 | $172 \times 172$ | None | 27.5 |
| MoViNet A0 [3] | 20 | $172 \times 172$ | Kinectics | 32.8 |
| MoViNet A2 [3] | 20 | $224 \times 224$ | Kinectics | 34.1 |
| **AZTR（我们的方法）** | 20 | $172 \times 172$ | Kinectics | **40.2** |

表 III：RoCoG-v2 上的结果。我们展示了我们的方法可以将 Top-1 准确率提高 6.1%-7.4%，并且优于所有可部署在 RB5 平台上的最先进方法。

采用半自动方式。我们首先使用人体检测器（级联 Mask R-CNN）计算粗略的边界框，然后由人工进行检查。我们还在 RoCoG-v2 上微调检测模型，从而缩小检测领域的差距。总体而言，在该数据集上使用我们的方法可将准确率提高 6.1%-7.4%。

**UAV Human [20]:** 这是规模最大的人类行为理解无人机数据集之一。该数据集包含 15172 个训练视频和 5556 个测试视频。所有视频均采集自多个城市和乡村地区，涵盖白天和夜间场景，且具有丰富的多样性（涉及受试者、背景、光照等）。与最先进水平相比，我们的方法在 UAV-Human 上将准确率提高了 8.3-10.4%。

**Drone Action [17]:** 这是一个室外无人机视频数据集，使用可自由飞行的无人机在低空低速条件下采集。该数据集包含涵盖 13 个人体动作的 240 个高清 RGB 视频。我们的方法超越最先进水平 3.2%，在 Drone Action 上达到 95.9% 的 Top-1准确率。

### *B.* 实现细节与训练

**骨干网络架构：**我们使用 MoViNets[3] 作为轻量级模型的骨干网络，并使用 X3D-M [44]作为台式机 GPU 上模型的骨干网络，对我们的模型进行基准测试。

**定位：** 我们将 MobileNet v2[45]用作轻量级模型的定位器，并将级联 Mask R-CNN[46]用作台式机 GPU 上模型的定位器。

**训练详情：**针对边缘设备或无人机平台上的模型，我们将 0.00005 设为初始学习率。针对高端桌面GPU上的模型，我们采用与 [24] 相同的初始化方式。从头开始训练时，我们将初始学习率设为 0.1；使用 Kinetics 预训练权重进行初始化时，则设为 0.05。我们使用随机梯度下降（SGD）优化器，权重衰减系数为 0.0005，动量为 0.9，并使用余弦/多项式退火进行学习率衰减。所有

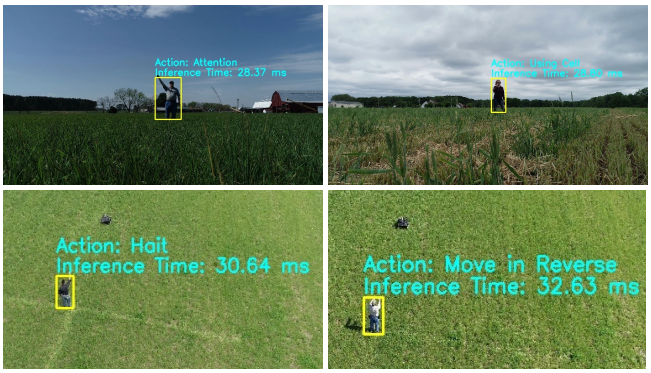



图 4：RoCoG-v2 航拍视频上的动作识别。更多详情请参见视频。

我们的模型均使用 NVIDIA GeForce 2080 Ti GPU 和 NVIDIA RTX A5000 GPU 进行训练。

### *C. RoCoG-v2* 上的结果

我们在预装 Ubuntu Linux 18.04 的环境中，使用机器人操作系统 2（ROS2）Galactics，在 RoCoG-v2 数据集上实现了我们的方法。我们使用从原始视频数据中提取并标注的 70,124 帧，对 MobileNet v2[45]进行微调。微调后的 MobileNet v2[45] 用作自动变焦算法的核心检测器。我们使用移动视频网络（MoViNets）[3]评估我们的方法。具体而言，我们在 RB5 上使用 MoViNet A0、A2 和 A3 测试我们的方法。MoViNet A0 是 MoViNets 中最小的模型，输入尺寸为 172×172。A2 采用更深的网络设计，输入尺寸为 224×224。A3 是规模更大的网络，输入尺寸为 256×256。此外，由于边缘设备不支持三维卷积，我们选择了 MoViNets 的流式版本，如表 I 所示。我们使用二维 + 1D 卷积和流缓冲区融合前序帧的识别结果，使其能够进行在线推理。我们在 Qualcomm RB5 平台上部署了 MoViNet A0、A2、A3 和我们的方法，并收集了 RB5 CPU 上的设备端推理时间。结果如表 II 所示。我们的方法在移动设备上实现了更优的速度—准确率权衡。与 MoViNet A3 相比，我们的方法在 RB5 CPU 上的平均推理时间为 56.5 ms，速度快 2 倍，且准确率略有提升。我们还通过采用不同的帧采样和模型初始化方式进行了更多实验，以评估算法性能（见表 III）。在相同配置下，我们的方法在 RoCoG-v2 数据集上的性能提升了 6.1%–7.4%。

### *D. UAV Human* 上的结果

为说明我们所提出方法的有效性，我们在最大的人体动作识别无人机数据集 UAV-Human [20] 上评估了我们的算法。如表 IV 所示，我们在 UAV-Human 上针对不同的骨干网络架构、帧率、输入尺寸和权重初始化方式开展了实验。在实验中，我们使用 X3D-M 作为骨干网络，并采用两种不同的初始化设置：一种是从头开始训练，另一种是使用 Kinetics 预训练权重初始化权重。根据实验结果，我们观察到

| 方法 | 主干网络 | 帧数 | 输入尺寸 | 初始化 | Top-1准确率 |
|---|---|---|---|---|---|
| X3D-S [44] | - | 16 | 224 × 224 | None | 21.5 |
| X3D-M [44] | - | 16 | 224 × 224 | None | 27.0 |
| X3D-L [44] | - | 16 | 224 × 224 | None | 27.6 |
| FAR [24] | X3D-M | 16 | 224 × 224 | None | 27.6 |
| FAR [24] | X3D-M | 8 | 540 × 540 | None | 28.8 |
| **AZTR（我们的方法）** | X3D-M | 16 | 224 × 224 | None | **39.2** |
| I3D [22] | ResNet-101 | 8 | 540 × 960 | Kinetics | 21.06 |
| X3D-M [44] | - | 16 | 224 × 224 | Kinetics | 30.6 |
| FNet [47] | I3D | 8 | 540 × 960 | Kinetics | 24.39 |
| FAR [24] | I3D | 8 | 540 × 960 | Kinetics | 29.21 |
| FAR [24] | X3D-M | 16 | 224 × 224 | Kinetics | 31.9 |
| FAR [24] | X3D-M | 8 | 620 × 620 | Kinetics | 39.1 |
| **AZTR（我们的方法）** | X3D-M | 16 | 224 × 224 | Kinetics | **47.4** |

表 IV：UAV Human 基准数据集及与先前方法的比较。我们与最先进的方法进行了比较，结果表明性能提升了 8.3% – 10.4，优于最先进的方法。训练使用高端桌面GPU。

| 方法 | 帧数 | 输入尺寸 | 初始化 | Top-1准确率 |
|---|---|---|---|---|
| HLPF [48] | All | 1920 × 1080 | None | 64.36 |
| PCNN [49] | - | 1920 × 1080 | None | 75.92 |
| X3D-M [44] | 16 | 224 × 224 | Kinetics | 83.4 |
| FAR [24] | 16 | 224 × 224 | Kinetics | 92.7 |
| **AZTR（我们的方法）** | 16 | 224 × 224 | Kinetics | **95.9** |

表 V：数据集：Drone Action 的结果。我们证明，AZTR 将最先进水平的准确率提升了 3.2%，在 Drone Action 上达到 95.9%。训练使用高端桌面 GPU。

使用 Kinetics 预训练权重进行初始化的效果优于从头开始训练。此外，与最先进的方法所采用的大输入尺寸相比，即使使用更小的输入尺寸（ 224 × 224），我们仍取得了显著的改进。对于从头开始训练的情况，我们的 AZTR 相比基线性能提升了 11.6%，相比最先进水平提升了 10.4%。对于使用 Kinetics 预训练权重初始化的情况，AZTR 相比基线性能提升了 16.8%，相比最先进水平提升了 8.3%。

### *E. Drone Action* 上的结果

我们还在 Drone Action [17]上评估了我们的方法。如表 V 所示，我们证明 AZTR 在 Drone Action 上以 95.9% 的 Top-1 准确率超越了最先进水平 3.2%，这体现了我们提出的 AZTR 的稳定性能和说服力。

## V. 结论、局限性与未来展望

我们提出了一种新颖的视频动作识别方法。该方法专为无人机使用的边缘设备（例如 Qualcomm Robotics RB5）和台式机 GPU（Nvidia RTX A5000 GPU）设计。我们提出了一种自动变焦算法，可自动识别目标并调整其尺度。此外，我们还引入了一种时序推理算法，以捕捉动作信息。我们观察到，该方法在不同数据集上的准确率均有所提升。我们的方法也存在一些局限性。首先，整体性能取决于定位方法。其次，我们假设输入视频中只有一名预设动作的人类演员进行表演。我们还希望开发能够在包含多名人体主体的视频中执行动作识别，并能应对光照和天气条件变化的方法。

## 参考文献


[1] K. Nguyen, C. Fookes, S. Sridharan, Y. Tian, X. Liu, F. Liu, and A. Ross, “航空监控的现状：综述”，arXiv预印本 arXiv:2201.03080，2022年。[2] J. Pan, A. Bulat, F. Tan, X. Zhu, L. Dudziak, H. Li, G. Tzimiropoulos, and B. Martinez, “Edgevits：在移动设备上以视觉变换器与轻量级卷积神经网络一较高下”，arXiv预印本 arXiv:2205 03436，2022年。[3] D. Kondratyuk, L. Yuan, Y. Li, L. Zhang, M. Tan, M. Brown, and B. Gong, “MoViNets：面向高效视频识别的移动视频网络”，载于 IEEE/CVF计算机视觉与模式识别会议（CVPR），2021年，第16 020–16 030页。[4] U. Demir, Y. S. Rawat, and M. Shah, “Tinyvirat：低分辨率视频动作识别”，载于2020年第25届国际模式识别会议（ICPR）。IEEE，2021年，第7387–7394页。[5] F. Rezazadegan, S. Shirazi, B. Upcrofit, and M. Milford, “动作识别：从静态数据集到移动机器人”，载于2017年IEEE国际机器人与自动化会议（ICRA）。IEEE，2017年，第3185–3191页。[6] N. Soans, E. Asali, Y. Hong, and P. Doshi, “Sa-net：面向从观察中学习的稳健状态动作识别”，载于2020年IEEE国际机器人与自动化会议（ICRA）。IEEE，2020年，第2153–2159页。[7] B. van Amsterdam, M. J. Clarkson, and D. Stoyanov, “用于手术手势识别与进度预测的多任务循环神经网络”，载于2020年IEEE国际机器人与自动化会议（ICRA）。IEEE，2020年，第1380–1386页。[8] J. Massardi, M. Gravel, and ´E. Beaudry, “Parc：面向辅助机器人的计划与活动识别组件”，载于2020年IEEE国际机器人与自动化会议（ICRA）。IEEE，2020年，第3025–3031页。[9] L. Yao, S. Liu, C. Li, S. Zou, S. Chen, and D. Guan, “Pa-awcnn：用于 RGB-D 动作识别的双流并行注意力自适应权重网络”，载于2022年国际机器人与自动化会议（ICRA）。IEEE，2022年，第8741–8747页。[10] Z. Shao, Y. Li, Y. Guo, J. Yang, and Z. Wang, “一种基于身体部位的动作识别分层模型”，载于2018年IEEE国际机器人与自动化会议（ICRA）。IEEE，2018年，第1978–1985页。[11] C. Lea, R. Vidal, and G. D. Hager, “学习卷积动作基元以实现细粒度动作识别”，载于2016年IEEE国际机器人与自动化会议（ICRA）。IEEE，2016年，第1642–1649页。[12] R. Geraldes, A. Goncalves, T. Lai, M. Villerabel, W. Deng, A. Salta, K. Nakayama, Y. Matsuo, and H. Prendinger, “基于深度学习的无人机态势感知系统”，IEEE Access，第7卷，第122 583–122 594页，2019年。[13] H. Mliki, F. Bouhlel, and M. Hammami, “基于无人机拍摄的视频序列的人体活动识别”，模式识别（PR），第100卷，文章编号107140，2020年。[14] B. Mishra, D. Garg, P. Narang, and V. Mishra, “自然灾害中的无人机监控与搜救”，Computer Communications，第156卷，第1–10页，2020年。[15] L. Mou, Y. Hua, P. Jin, and X. X. Zhu, “利用深度神经网络和新数据集识别航拍视频中的事件与活动”，载于 IGARSS 2020—2020年IEEE国际地球科学与遥感研讨会（IGARSS）。IEEE，2020年，第952–955页。[16] M. Barekatain, M. Mart´ı, H.-F. Shih, S. Murray, K. Nakayama, Y. Matsuo, and H. Prendinger, “Okutama-action：用于并发人体动作检测的航拍视角视频数据集”，载于 IEEE/CVF计算机视觉与模式识别会议研讨会（CVPRW），2017年，第28–35页。[17] A. G. Perera, Y. W. Law, and J. Chahl, “Drone-action：用于动作识别的户外录制无人机视频数据集”，Drones，第3卷，第4期，文章编号82，2019年。[18] A. G. Perera, Y. W. Law, T. T. Ogunwa, and J. Chahl, “用于人体动作识别的户外多视角数据集”，IEEE人机系统汇刊，第50卷，第5期，第405–413页，2020年。[19] J. Choi, G. Sharma, M. Chandraker, and J.-B. Huang, “面向无人机动作识别的无监督与半监督领域自适应”，载于 IEEE/CVF冬季计算机视觉应用会议（WACV），2020年，第1717–1726页。

[20] T. Liu, J. Lian, W. Zhang, 等, “Uav-human：基于无人机进行人类行为理解的大型基准数据集”，载于 IEEE/CVF计算机视觉与模式识别会议（CVPR），2021年，第16 266–16 275 页。[21] W. Sultani 和 M. Shah，“使用少量空中训练样本识别无人机视频中的人类动作”，《计算机视觉与图像理解》，第 206 卷，页码 103186，2021年。[22] J. Carreira 和 A. Zisserman，“动作识别何去何从？一种新模型与 Kinetics 数据集”，载于 IEEE/CVF 计算机视觉与模式识别会议（CVPR），2017年，第 6299–6308 页。[23] H. Peng 和 A. Razi，“利用空中图像实现全自主无人机动作识别系统”，载于国际视觉计算研讨会（ISVC）。Springer，2020年，第 276–290 页。[24] D. Kothandaraman、T. Guan、X. Wang、S. Hu、M. Lin 和 D. Manocha，“FAR：傅里叶空中视频识别”，arXiv 预印本 arXiv:2203.10694，2022年。[25] D. Du、Y. Qi、H. Yu、Y. Yang、K. Duan、G. Li、W. Zhang、Q. Huang 和 Q. Tian，“无人机基准数据集：目标检测与跟踪”，载于欧洲计算机视觉会议（ECCV），2018年，第 370–386 页。[26] P. Zhu、L. Wen、D. Du、X. Bian、H. Fan、Q. Hu 和 H. Ling，“检测与跟踪遇上无人机挑战赛”，arXiv 预印本 arXiv:2001.06303，2020年。[27] A. G. Perera、Y. Wei Law 和 J. Chahl，“Uav-gesture：用于无人机控制和手势识别的数据集”，载于欧洲计算机视觉会议研讨会（ECCVW），2018年，第 0–0 页。[28] C. M. de Melo、B. Rothrock、P. Gurram、O. Ulutan 和 B. Manjunath，“在使用合成数据的人机团队中进行基于视觉的手势识别”，载于2020年 IEEE/RSJ 智能机器人与系统国际会议（IROS）。IEEE，第 10 278–10 284 页。[29] Q. Fan、C.-F. R. Chen、H. Kuehne、M. Pistoia 和 D. Cox，“少即是多：通过大小网络和逐通道时间聚合学习高效的视频表示”，《神经信息处理系统进展》（NeurIPS），第 32 卷，2019年。[30] S. Bhardwaj、M. Srinivasan 和 M. M. Khapra，“使用更少帧的高效视频分类”，载于 IEEE/CVF计算机视觉与模式识别会议（CVPR），2019年，第 354–363 页。[31] C.-F. Chen、Q. Fan、N. Mallinar、T. Sercu 和 R. Feris，“大小网络：用于视觉与语音识别的高效多尺度特征表示”，arXiv 预印本 arXiv:1807.03848，2018年。[32] X. Li、Y. Wang、Z. Zhou 和 Y. Qiao，“SmallBigNet：融合核心视图与上下文视图进行视频分类”，载于 IEEE/CVF计算机视觉与模式识别会议（CVPR），2020年，第 1092–1101 页。[33] D. Tran、H. Wang、L. Torresani 和 M. Feiszli，“使用通道分离卷积网络进行视频分类”，载于 IEEE/CVF国际计算机视觉会议（ICCV），2019年，第 5552–5561 页。[34]A. G. Howard、M. Zhu、B. Chen、D. Kalenichenko、W. Wang、T. Weyand、M. Andreetto 和 H. Adam，“MobileNets：面向移动视觉应用的高效卷积神经网络”，arXiv 预印本 arXiv:1704.04861，2017年。[35] J. Redmon、S. Divvala、R. Girshick 和 A. Farhadi，“You Only Look Once：统一的实时目标检测”，载于 IEEE/CVF计算机视觉与模式识别会议（CVPR），2016年，第 779–788 页。[36] M. Ding、N. Li、Z. Song、R. Zhang、X. Zhang 和 H. Zhou，“一种用于无人机视频的轻量级动作识别方法”，载于2020年 IEEE 第三届电子与通信工程国际会议（ICECE）。IEEE，2020年，第 181–185 页。[37] A. Piergiovanni、A. Angelova 和 M. S. Ryoo，“Tiny Video Networks”，《Applied AI Letters》，第 3 卷，第 1 期，页码 e38，2022年。[38] J. Lin、C. Gan 和 S. Han，“TSM：用于高效视频理解的时序移位模块”，载于 IEEE/CVF国际计算机视觉会议（ICCV），2019年，第 7083–7093 页。[39] A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、Ł. Kaiser 和 I. Polosukhin，“注意力机制就是一切”，《神经信息处理系统进展》（NeurIPS），第 30 卷，2017年。[40] A. Jaegle、F. Gimeno、A. Brock、O. Vinyals、A. Zisserman 和 J. Carreira，“Perceiver：通过迭代注意力实现通用感知”，载于国际机器学习会议（ICML）。PMLR，2021年，第 4651–4664 页。

[41] M. Abadi、A. Agarwal、P. Barham、E. Brevdo、Z. Chen、C. Citro、G. S. Corrado、A. Davis、J. Dean、M. Devin、S. Ghemawat、I. Goodfellow、A. Harp、G. Irving、M. Isard、Y. Jia、R. Jozefowicz、L. Kaiser、M. Kudlur、J. Levenberg、D. Man´e、R. Monga、S. Moore、D. Murray、C. Olah、M. Schuster、J. Shlens、B. Steiner、I. Sutskever、K. Talwar、P. Tucker、V. Vanhoucke、V. Vasudevan、F. Vi´egas、O. Vinyals、P. Warden、M. Wattenberg、M. Wicke、Y. Yu 和 X. Zheng，“TensorFlow：异构系统上的大规模机器学习”，2015年，软件可从 tensorflow.org 获取。[在线]。网址：https://www.tensorflow.org/ [42] C. Lv、C. Niu、R. Gu、X. Jiang、Z. Wang、B. Liu、Z. Wu、Q. Yao、C. Huang、P. Huang、T. Huang、H. Shu、J. Song、B. Zou、P. Lan、G. Xu、F. Wu、S. Tang、F. Wu 和 G. Chen，“Walle：面向设备—云协同机器学习的端到端通用大规模生产系统”，载于第*16*届 *USENIX*操作系统设计与实现研讨会（*OSDI 22*）。加利福尼亚州卡尔斯巴德：USENIX协会，2022年7月，第 249–265 页。 [在线]。网址：https://www.usenix.org/conference/osdi22/presentation/lv [43] A. V. Reddy、K. Shah、W. Paul、R. Mocharla、J. Hoffman、K. D. Katyal、v Dinesh、C. M. de Melo 和 R. Chellappa，“RoCoG-v2”，评审中，*https://github.com/reddyav1/RoCoG-v2*，2022年。[44] C. Feichtenhofer，“X3d：扩展架构以实现高效视频识别”，载于*IEEE/CVF*计算机视觉与模式识别会议（*CVPR*），2020年，第 203–213 页。[45] M. Sandler、A. Howard、M. Zhu、A. Zhmoginov 和 L.-C. Chen，“Mobilenetv2：倒置残差与线性瓶颈”，载于*IEEE/CVF*计算机视觉与模式识别会议（*CVPR*），2018年，第 4510–4520 页。[46] I. Hasan、S. Liao、J. Li、S. U. Akram 和 L. Shao，“通用行人检测：显而易见却被忽视的问题”，载于*IEEE/CVF*计算机视觉与模式识别会议（*CVPR*），2021年，第 11 328–11 337 页。[47] J. Lee-Thorp、J. Ainslie、I. Eckstein 和 S. Ontanon，“FNet：使用傅里叶变换混合词元”，*arXiv* 预印本 *arXiv:2105 03824*，2021年。。[48] H. Jhuang、J. Gall、S. Zuffi、C. Schmid 和 M. J. Black，“迈向动作识别理解”，载于*IEEE/CVF*国际计算机视觉会议（*ICCV*），2013年，第 3192–3199 页。[49] G. Ch´eron、I. Laptev 和 C. Schmid，“P-cnn：用于动作识别的基于姿态的 CNN 特征”，载于*IEEE/CVF*国际计算机视觉会议（*ICCV*），2015年，第 3218–3226 页。