**基于深度回归网络在100FPS中学习跟踪**

David Held, Sebastian Thrun, Silvio Savarese

Department of Computer Science

Stanford University

{davheld,thrun,ssilvio}@cs.stanford.edu

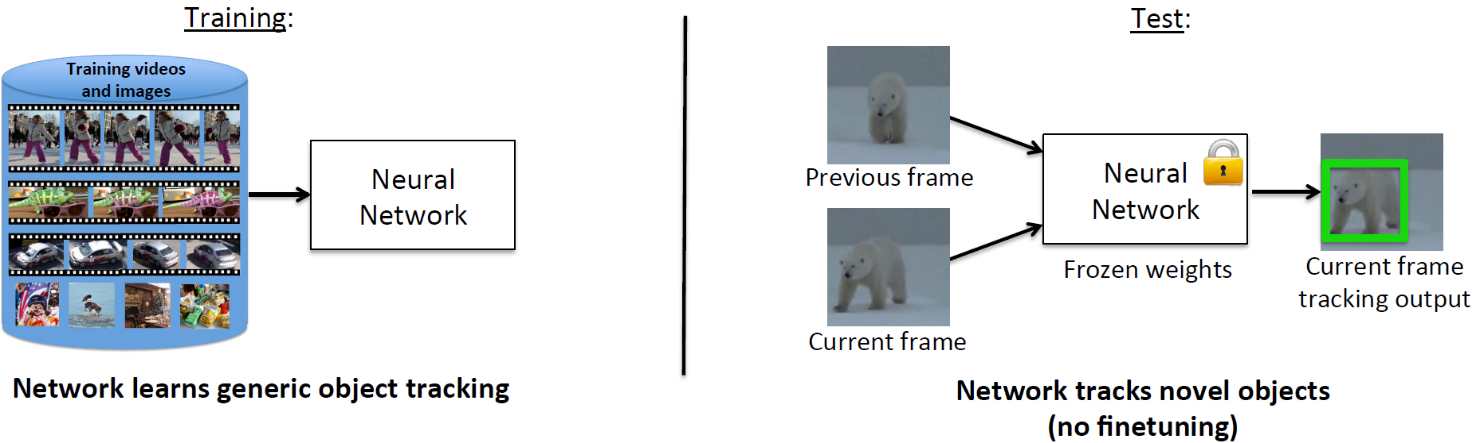
**摘 要**：机器学习技术经常被用在计算机上，因为他们有能力利用大量的培训数据提高性能。不幸的是，大多数通用的对象跟踪器仍然是在线开始训练，并没有从很多可以随时用于离线训练的视频中获益。我们建议一种神经网络离线训练方法，这种方法能在100fps、测试时间内跟踪新的对象。我们的跟踪器显着快于以前使用神经网络进行跟踪的方法，运行速度非常慢，对于实时应用来说不实用。我们的跟踪器使用一个简单的，不需要在线训练的前馈网络。跟踪器学习对象运动和表现之间的一般关系，并且可以用来追踪没有在训练集中出现的新对象。我们在一个标准的跟踪基准上测试我们的网络来演示我们的跟踪器的最先进的表现。进一步，随着我们为离线训练集添加更多的视频，跟踪器的表现也会提高。据我们所知，我们的跟踪器[[1]](#footnote-1)是第一个在100fps中学习跟踪通用对象的神经网络跟踪器。

**关键词**：跟踪；深度学习；神经网络；机器学习。

**1 介绍**

考虑到在视频的一帧中标记了一些感兴趣的对象，“单目标跟踪”是在随后的视频帧中将该对象定位，尽管是对象有运动，视点变化，照明变化或其他变化。单目标跟踪是许多系统的重要组成部分。 对于跟踪人的应用，机器人必须跟踪这个人在环境中的移动。对于自主驾驶，机器人必须跟踪动态障碍物，来预估他们估算哪里移动，并预测他们将来如何移动。

通用对象跟踪器（此跟踪器不专门针对特定类型的物体）完全从头开始在线训练（即在测试期间）[15,3,36,19]，而不进行离线训练。 这样的跟踪器性能不佳，因为他们无法利用大量的很容易获取的视频来改善他们的性能。离线训练视频能被用来教授跟踪器处理旋转，视点改变，亮度改变和其他复杂的挑战。



**图1 使用边界框标签的视频和图像的集合（但没有类信息），我们训练神经网络来跟踪通用对象。 在测试的时候，网络能够跟踪新的对象而不需要任何调整。 通过避免微调，我们的网络能够以100fps的速度跟踪**

在计算机视觉的许多其他领域，如图像分类，对象检测，分割或活动识别，机器学习允许视觉算法来从离线数据训练并了解世界[5,23,13,25,9,28]。在每种情况下，算法的性能随着训练图像集的迭代而提高。 这样的模型是从神经网络在大规模的数据中学习复杂功能的能力中获益。

在这项工作中，我们表明了学习通过观看世界上运动物体的离线视频来即时跟踪通用对象是可能的。 为了实现这个目标，我们介绍*GOTURN*，Generic Object Tracking Using Regression Networks使用回归网络跟踪通用对象。 我们训练神经网络以完全离线的方式进行跟踪。在测试的时候，当追踪新物体时，网络权重被冻结，不需要在线调整（如图1所示）。 通过培训过程中，跟踪器学习了以快速，稳健和准确的方式

2 Held, Thrun, Savarese

跟踪新的对象。

虽然使用神经网络跟踪的一些初步工作已经完成，但这些努力制造出了在实际使用中很慢的神经网络跟踪器。相比之下，以我们最好的结论来看，我们的跟踪器能够以100fps的速度跟踪物体，成为迄今为止最快的神经网络跟踪器。我们的实时速度由两个因素决定。首先，大多数以前的神经网络跟踪器是在线训练的[26,27,34,37,35,30,39,7,24];然而，训练神经网络是一个缓慢的过程，会导致跟踪缓慢。相比之下，我们的跟踪器是经过离线训练来学习外观和动作之间的一般关系，所以不需要在线训练。其次，大多数追踪器采取分类方法，分类许多图像补丁找到目标对象[26,27,37,30,39,24,33]。比较来说，我们的跟踪器使用基于回归的方法，只需要一个前馈通过网络直接回归到目标对象的位置。离线训练和一次回归的结合导致了一个比起之前方法显著的速度提升，并且允许我们以实时速度追踪目标。

GOTURN是第一个能以100 fps运行的通用对象神经网络跟踪器。 我们使用一个标准的跟踪基准来证明我们的跟踪器优于最先进的跟踪器。 我们的追踪器从一个标记过的训练视频和图像训练而来，但我们不需要任何级别的标签或关于正在跟踪的对象的类型的信息。 GOTURN成立一个新的跟踪框架，其中外观和动作的关系是以一种通用的离线方式学习的。我们的代码和其他经验可以在以下网址找到：

<http://davheld.github.io/GOTURN/GOTURN.html>

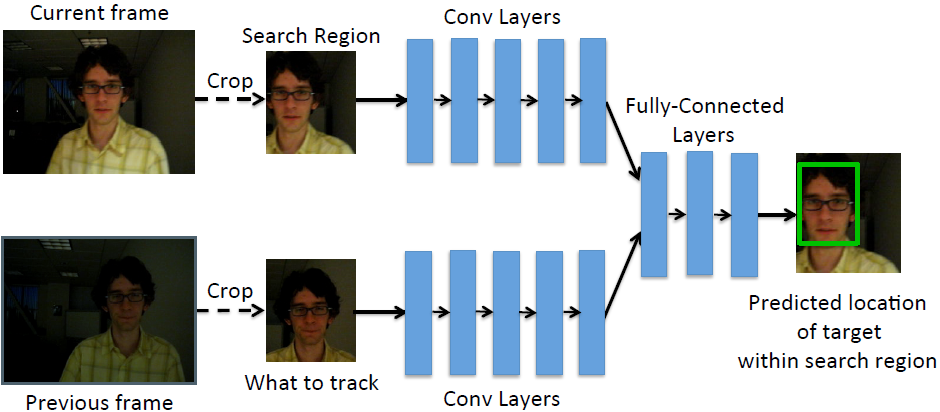
**2 相关的工作**

**在线跟踪培训。**用于通用对象跟踪的跟踪器通常是从视频的第一帧开始完全在线训练[15,3,36,19]。 一个典型的跟踪器将在目标对象的补丁附近采样，这被认为是“前景”[3]。还有一些距目标物体较远的区域也被采样，这些被认为是“背景”。然后这些补丁被用来训练一个前景背景分类器，这个分类器被用来给目标对象下一帧估测的新位置的补丁评分[36,19]。不幸的是，由于这些跟踪器完全在线培训，他们不能从很多可以随时用于离线训练的可能可以提升性能的视频中获益。

一些研究人员也试图使用神经网络以传统的在线训练框架进行跟踪[26,27,34,37,35,30,39,7,24,16]，显示了最先进的成果[30,7,21]。 不幸的是，神经网络的训练非常缓慢，并且如果需要在线训练，就会导致跟踪器在测试时也非常的慢。 这种从0.8 fps [26]到15 fps [37]的最高性能的神经网络跟踪器在GPU上只能以1 fps运行[30,7,21]。因此，这些跟踪器不适用于大多数实际应用。 因为我们跟踪器是以一种离线通用的方式进行训练的，不会有在线培训需求，使我们能够在100帧/秒跟踪。

**基于模型的跟踪器**。 一个单独的跟踪器类是基于模型的跟踪器，旨在跟踪特定类别的对象[12,1,11]。 比如，如果只对跟踪行人感兴趣，那么可以训练一个行人探测器。 在测试期间，这些检测可以使用时间信息连接在一起。 这些追踪器是离线训练的，但是它们是有限制的，因为他们只能跟踪特定类别的对象。 我们的跟踪器已经过通用方法离线训练，并且可用于在测试时间跟踪新类别的对象。

**其他神经网络跟踪框架。** 一个相关的研究领域是补丁匹配[14,38]，最近在[33]中用于追踪，以4 fps运行。 在这样的方法中，很多候选补丁都通过了网络，选择匹配得分最高的补丁作为跟踪输出。 相比之下，我们的网络只通过网络传递两个图像，网络直接回归到目标对象的边界框位置。得以避免给很多候选补丁评分的需求，我们能够以100fps的速度追踪。



**图2.我们的跟踪网络架构。 我们从当前帧的和前一帧的目标向网络输入一个搜索区域。 网络学习到比较这些裁剪以找到当前图像中的目标对象**

之前已经尝试使用神经网络在不同的其他方式跟踪，包括视觉注意力模型[4,29]。但是，这些方法在复杂的跟踪数据集的评估时不是其他最先进的跟踪器的对手。

**3 方法**

**3.1 方法概观**

在较高的层面上，我们将视频的帧提供给神经网络，网络能依次输出每帧中被跟踪物体的位置。 我们用视频序列和图像完全离线训练跟踪器。 通过我们的离线训练进程，我们的跟踪器学习外观和运动之间的一般关系，这可以在测试时用来跟踪新的对象而不需要在线训练。

**3.2输入/输出格式**

**跟踪什么。**如果视频中有多个对象，则为网络必须接收哪一个对象正在被跟踪的信息。为了达到这个目的，我们把目标对象的图像输入到网络中。 我们裁剪并缩放前一帧，以目标对象为中心，如图所示在图2中。这个输入允许我们的网络跟踪它没有见过的新物体; 网络将跟踪该作物中输入的任何对象。我们填充该作物以允许网络接收一些关于目标物体的周围环境的上下文信息。

更详细地说，假设在帧t-1，我们的跟踪器之前预测该目标位于以c =（,）为中心的宽度为w，高度为h的边界框中。 在t时刻，我们以（cx; cy）为中心的第t - 1帧作裁剪，其宽度和高度分别为k1 w和k1 h。这个裁剪告诉网络正在哪一个是正在被追踪的对象。 k1的值决定了网络接收多少前一帧的上下文信息。

**找哪里**。为了找到当前帧的目标对象，跟踪器应该知道这个对象之前是在什么位置。因为对象往往会很光滑的移动过空间，所以对象之前的位置会提供一个好的，关于网络应该在哪里找到对象。我们实现这个，是在基于对象前一帧的位置的当前帧选择一个搜索区域。我们用这个搜索区域来裁剪当前帧，并将这个裁剪输入我们的网络，正如图2所示。网络的目标是在搜索区域回归到目标对象的位置。

更详细地说，当前帧t的裁剪以c’ =（, ）为中心，其中c’是目标对象的预期平均位置。 我们设置c’ = c，相当于一个恒定位置的运动模型，虽然更复杂运动模型也可以使用。 当前帧的裁剪有一个宽度高度分别为k2 w和k2 h，其中w和h分别为前一帧的预测边界框中的宽度和高度，k2定义我们的搜索目标对象的半径。 在实践中，我们使用k1 = k2 = 2。

只要目标对象没有被遮挡，也不会移动太快，目标将定位于这个区域。 对于快速移动的对象可以增加搜索区域的大小，但代价是增加网络的复杂性。 另外，要处理长期的遮挡或大动作，我们的跟踪器可以结合另一种方法，比如一个在线训练过的物体探测器，如TLD框架[19]，或者一个视觉注意力模型[4,29,2]; 我们留下这个作为未来的工作。

**网络输出。** 网络相对于搜索区域输出对象在当前帧的坐标，。 网络的输出包括边界框的左上角和右下角的坐标。

**3.3 网络架构**

对于单目标跟踪，我们定义了一个新的图像比较跟踪架构，如图2所示（请注意，相关的“两帧”架构）也被用于其他任务[20,10]）。 在这个模型中，我们将目标对象以及搜索区域都输入进一系列卷积层。该这些卷积层的输出是捕捉了图像高层次表象的特征集。

这些卷积层的输出，被提供给一些数字完全连接的层。 完全连接层的作用是比较从目标对象到当前帧中的特征来目标对象已移动到了哪里。 在这些帧之间，对象可能经历过翻转，旋转，照明改变，遮挡或变形。该由完全连接层学习到的功能因此是一个复杂的特征对照功能，这是当输出被追踪对象相关运动时，通过许多例子学习来变得更强大以应对多变的因素的功能。

更详细地说，我们的模型中的卷积层是从CaffeNet结构的五卷积层的第一层开始[17,23]。 我们将这些卷积层的输出串联在一起（即池5的特征）输出到单个矢量中。这个向量被输入到3个完全连接的层，每层有4096个节点。 最后，我们将最后一个完全连接层连接到一个包含4个节点代表输出边界框的输出层。 我们把输出提高10倍，使用我们的验证集（与我们所有的超参数一样）来选择。 网络超参数取自CaffeNet的默认值，以及在每个完全连接的层之间，我们如CaffeNet 里一样非线性的使用压缩和的ReLU。我们的神经网络是使用Caffe实现的[17]。

**3.4 跟踪**

在测试期间，我们从第一帧开始用一个实际边界框来初始化跟踪器，就像单一目标跟踪的标准做法一样。 在随后的每一个帧t，我们将帧t-1和帧t的裁剪输入到网络中（如3.2节所述）来预测对象在帧t中的位置。 我们继续重新裁剪，并将帧对提供给我们的网络剩余的视频，我们的网络将贯穿整个视频序列跟踪目标对象的移动。

**4 训练**

我们用视频和静止图像的组合来训练我们的网络。 培训程序如下所述。 在这两种情况下，我们都用预测的边界框和实际边界框之间的L1损失来训练网络。

**4.1从视频和图像训练**

我们的训练集由一系列视频组成，其中包含一部分帧在每个视频都标有一些对象的位置。 对于训练集中的每对连续帧，我们按3.2节所述裁剪。在训练期间，我们将这一对帧送入网络并尝试来预测对象如何从第一帧移动到第二帧（如图3所示）。 我们也用我们的运动模型来增加这些训练的例子，如4.2节所述。

我们的训练程序也可以利用每个都标有一个对象的位置的一组静止图像。 这套训练图像教导我们的网络跟踪更多元化的对象并防止和我们的培训视频中的对象过度拟合。 从图像训练我们的跟踪器，我们根据我们的运动模型采取图像的随机裁剪（见第4.2节）。 在这两个裁剪之中，目标对象已经经历了明显的翻转和规模的变化，如图4所示。我们认为这些裁剪就好像他们被采取来自视频的不同帧。 虽然这些裁剪比我们其他训练视频中找到的运动的变化少，但是这些图像仍然对我们的网络追踪多变的不同对象有作用。

**图3.培训视频的例子。网络的目标是预测在第一行视频帧被转移到下一行之后，中心所示的目标对象的位置。真值框被标记为绿色。**

**图4.训练图像的例子。网络的目标是预测在第一行裁剪视频帧被移动到如下一行的样子之后的目标对象位置。 地面真值边框被标记为绿色**

**4.2 Learning Motion Smoothness**

现实世界中的物体倾向于在空间中平稳地移动。 给一个模糊不清的图像，目标对象的位置不确定，一个跟踪器应该预测目标对象的位置在它上次被观测到的位置的附近。 这在包含多个相似对象的视频中尤为重要，例如相同类型的多个水果。 因此我们希望教我们的网络，其他的都是平等的，比起大的动作更喜欢小的动作。

为了体现运动平滑的思想，我们模拟了当前帧中边界框的中心（;）相对于在前一帧中边界框的中心（; ）的关系为

(1)

(2)

其中w和h分别是前一帧边界框的宽度和高度。 术语和是根据边界框的尺寸来捕捉它的位置改变的随机变量。在我们的训练集，我们发现物体这样以改变位置都能用平均值为0的拉普拉斯分布建模（详见附录）。这种分布对较大运动来说，较小的运动的可能性更高。

相似的，我们模型尺寸由以下公式改变γ

(3)

(4)

其中和是当前边界框的宽和高， 和是前一个边界框的宽度和高度。 术语是捕捉边界框尺寸变化的随机变量。 我们在我们的训练集中发现，由平均数为1的拉普拉斯分布建模。这样的分布使边界框有更高概率保持与前一帧的尺寸大小相同。

为了教导我们的网络更喜欢小动作，我们用从上述拉普拉斯分布中提取的随机裁剪来加强训练集如上所述（见图3和4为例）。 因为这些培训是从拉普拉斯分布中采样，小的运动比起大的运动将被更多的采样，因此我们的网络将会学习更喜欢小动作，其他都是平等的。 我们将展示这拉普拉斯剪裁程序与用于分类任务的标准相比提高了我们跟踪器的性能[23]。

拉普拉斯分布的尺度参数通过 = 1/ 5（对于边界框中心的运动）和 = 1/15（用于边界框大小的改变）来选择。 我们限制随机裁剪这样它必须包含每个维度中至少一半的目标对象。 我们也限制了尺寸的变化：（0.6,1.4），以避免过度拉伸或者收缩边界框对网络学习造成困难。

**4.3训练过程**

为了训练我们的网络，每个训练示例都是从视频或图像中交替采集的从图像。 当我们使用视频训练的例子时，我们随机选择一个视频，我们在这个视频中随机选择一对连续的帧。 然后我们根据第3.2节中描述的过程裁剪视频。另外如第4.2节所述，将当前帧的k3个随机裁剪结合起来增强数据集。 接下来，我们随机抽样一个图像，并且我们重复上述过程，其中随机裁剪创建人造“动作”（参见4.1和4.2节）。每次视频或图像被采样，新的随机裁剪即时被生产出来，这在我们的训练过程中创造多样化。 在我们的实验中，我们使用k3 = 10，我们使用50的批量大小。

我们网络中的卷积层是在ImageNet上预先训练好的[31,8]。由于我们有限的训练集大小，我们不会将这些图层调整以防止发生调整过度。 我们以学习率为1e-5来训练这个网络，并且还有其他超参数取自CaffeNet [17]的默认值。

**5 实验步骤**

**5.1 训练集**

如第4节所述，我们使用视频和静止图像的组合来训练我们的网络。我们的培训视频来自ALOV300 ++ [32]，一个314个视频序列的合集。我们删除了7个与我们的测试重叠的视频（详见附录），给我们留下了307视频用于培训。在这个数据集中，大约每个视频的第5帧都被标记了跟踪的某个对象的位置。这些视频一般很短，从几秒到几分钟不等。我们将这些视频分成251用于训练，56用于验证/超参数调整。培训一共包含了251,012个不同物体的13,082张图像，或者平均值为每个物体52帧。验证集由56个不同对象的2,795个不同的图像组成。选择我们的超参数后，我们使用我们的整个训练集模型重新训练模型（训练+验证）。删除7个重叠后视频中，训练和测试集中的视频之间没有重叠。

我们的培训程序还利用了一套用于培训的静止图像，如4.1节所述。 这些图像来自训练一套ImageNet Detection Challenge [31]，其中478,807个对象是用边界框标记过的。我们在训练时间内随机剪裁这些图像，如第4.2节所述，在两种随机裁剪之间创建明显的翻转或比例变化。 随机裁剪过程仅在以下情况下有用：被标记的对象不会覆盖整个图像; 因此，我们删除那些边界框在任一维度上至少占图像大小的66％的图片（使用我们的验证集选择）。 这给我们带来了134,821张图片中的239,283个注释。 这些图像有助于防止当教学我们的网络跟踪未出现在培训视频中的对象的时候出现的调整过度。

**5.2 测试集**

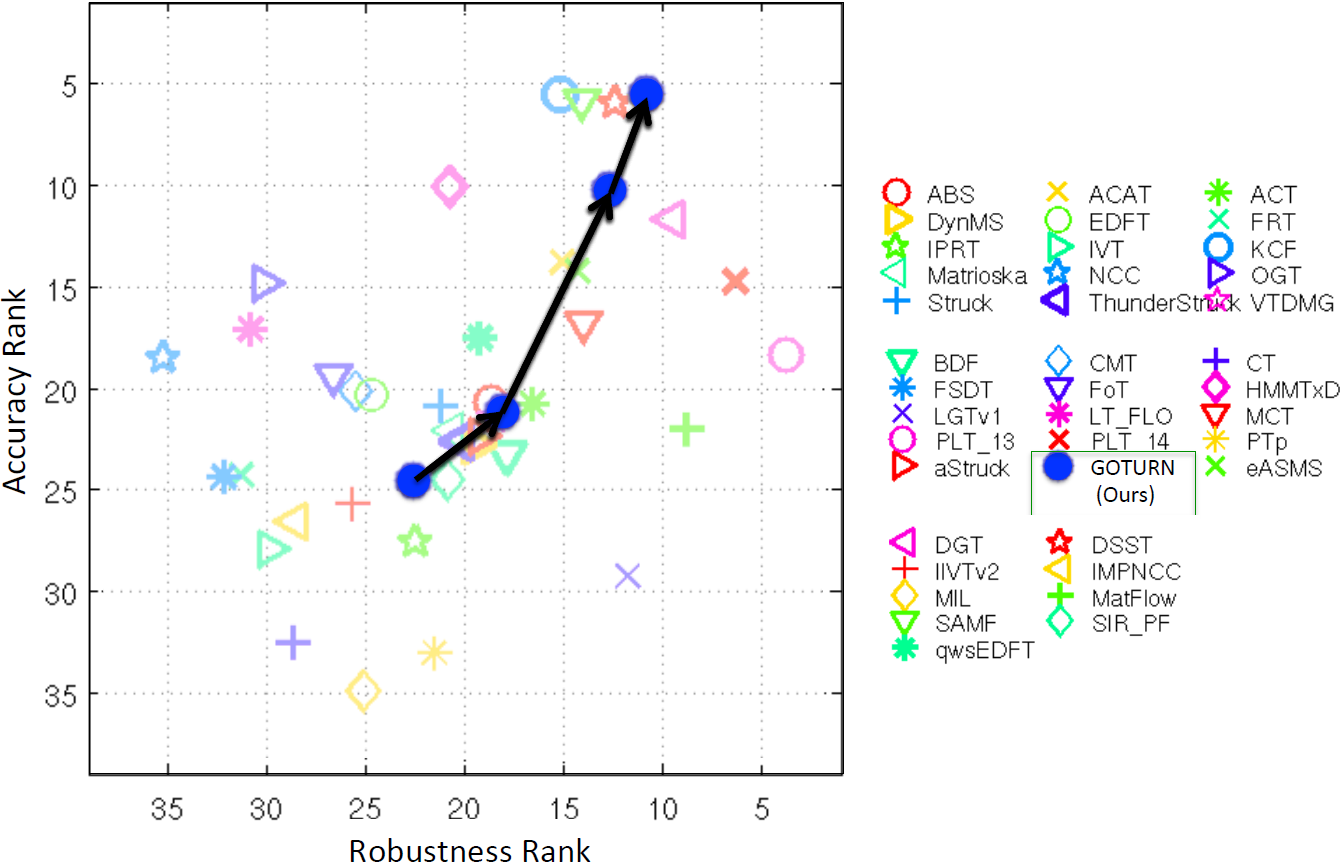
我们的测试集包含2014年VOT 2014追踪挑战赛的25个视频[22]。我们无法在VOT 2015挑战赛上测试我们的方法[21]，因为在那里在测试集和我们的训练集之间会有太多重叠。 然而，我们预计我们的方法的总体趋势仍然存在。

2014年VOT追踪挑战赛[22]是一个标准的追踪基准，这使我们能够将我们的跟踪器与各种先进的跟踪器进行比较。 跟踪器使用两个标准跟踪指标进行评估：准确性（A）和健壮性（R）[22,6]，范围从0到1.我们也计算精度误差（1-A），健壮性误差（1-R），以及总体误差1-（A + R）/ 2。

视频的每一帧都用许多属性进行注释：遮挡，照明变化，运动变化，尺寸变化和相机运动。 跟踪器的每个属性的准确性和健壮性也分别排名，并且排名然后跨属性平均得到最终的平均准确性和每个跟踪器的健壮性排名。 准确性和健壮性排名进行平均得到整体平均排名。

**6 结果**

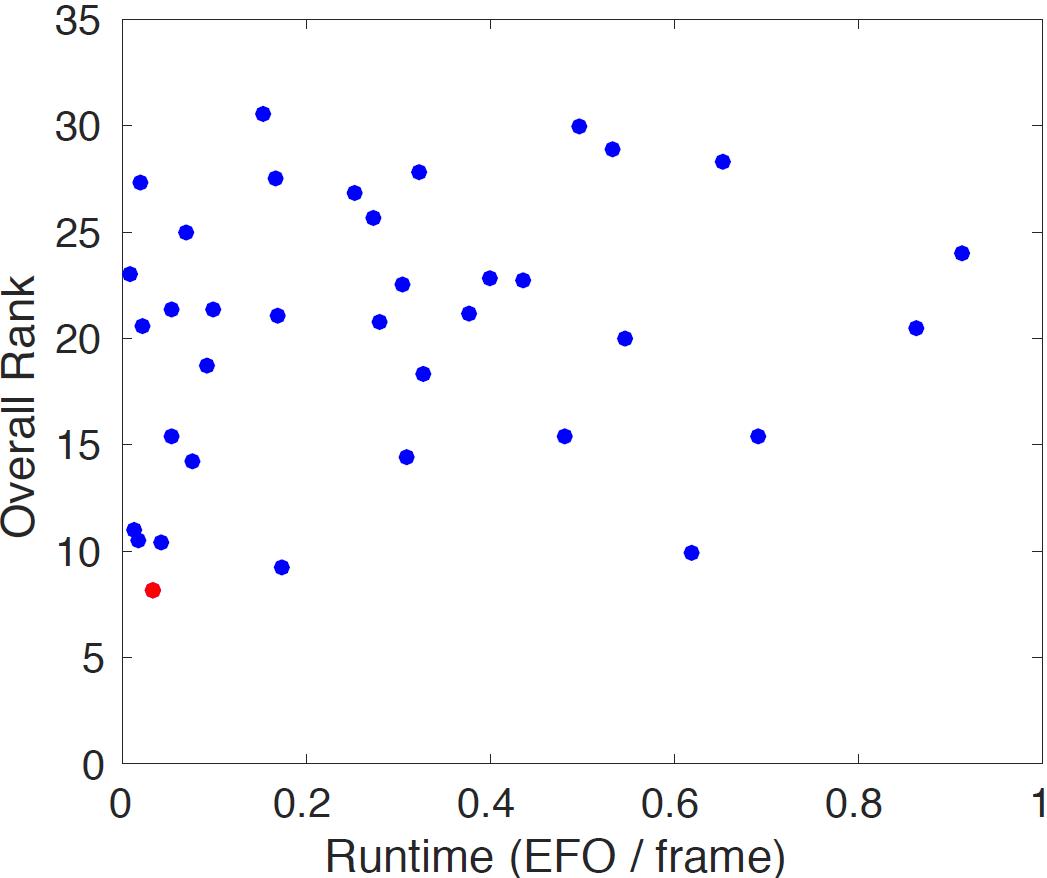
**6.1 整体表现**

我们的跟踪器的性能如图5所示，这表明我们的跟踪器具有很好的健壮性，在精度上接近于顶点。进一步，我们的整体排名（计算精度和健壮性的平均值）胜过以前在这个基准上所有的跟踪器。我们已经证明了离线训练对于提高跟踪性能的价值。此外，这些培训结果仅在307个短片中获得。图5以及附录中的分析表明，如果训练集通过标记更多视频来增大尺寸能取得更大的收获。定性结果，以及失败的案例，可以在项目页面：<http://davheld.github.io/>上找到；目前，跟踪器会因为阻塞或在训练集中过度条件对象而失败。

**图5。VOT 2014跟踪挑战的跟踪结果。我们跟踪器的性能用蓝色圆圈表示，在整体排名上比以前所有的方法都要好（平均精度和健壮性排名）。沿着黑线显示的点代表从14, 37, 157，和307个视频中训练，与每种情况下使用的训练图像数量是同样的。**

在一个cuDNN加速的NVIDIA GeForce GTX Titan X GPU上，我们的跟踪器以每帧6.05ms的速度运行（不包括在opencv加载每张图像用的1毫秒），或165fps。在一个GTX 680的GPU上，我们的跟踪器以每帧9.98ms的平均速度运行，或者说100fps。如果只有CPU是可用的，跟踪器运行为2.7fps。因为我们的跟踪器是能够执行所有的离线训练，在测试期间，在跟踪时只需要有一个单前馈通过网络，因此，跟踪器能以实时速度运行。

我们在图6用我们的跟踪器的速度和等级与其他38个提交到VOT 2014 Tracking Challenge的跟踪器进行比较 [ 22 ]，使用第5.2节所述的总体排名分。我们展示了在EFO单元的跟踪器的运行时间（等效过滤操作），标准化了跟踪器在[ 22 ]上面测试的硬件类型。图6演示了与其他38个基线相比，我们的追踪速度最快，而且在整体排名上超越的所有其他方法（以精度和健壮性排名平均数计算）。注意一些其他的跟踪器，如ThunderStruck [ 22 ]，还使用了GPU。为了更详细的分析速度作为准确性和健壮性的函数，见附录。



**图6。我们的跟踪器（红）与VOT 2014 Tracking Challenge的38种基线方法（蓝）相比的整体排名和运行时间图像。每个蓝色圆点代表的性能（最好看颜色）。精度和健壮性指标如附录所示**

由于我们的两个方面，我们的跟踪器能够实时跟踪对象模型：首先，我们学习通用的跟踪模型，所以没有在线培训需要。 神经网络的在线培训往往非常缓慢，阻碍了实时性能。 在线训练的神经网络跟踪器范围从0.8fps [26]至15 fps [37]，最高性能追踪器以1 fps运行在GPU上 [30,7,21]。 其次，大多数追踪器评估的样本数量很少并选择得分最高的作为跟踪输出[26,27,37,30,39,24,33]。采样方法的准确度受样品数量的限制，但是增加样本数量也增加了计算复杂度。另一方面，我们的跟踪器直接回归到输出边界框，所以GOTURN可以实现准确的跟踪，无需额外的计算成本，实现它以100 fps的速度跟踪物体。

**6.2 它怎样工作？**

我们的神经网络跟踪器如何工作？ 有两个假设，一个可能会是：

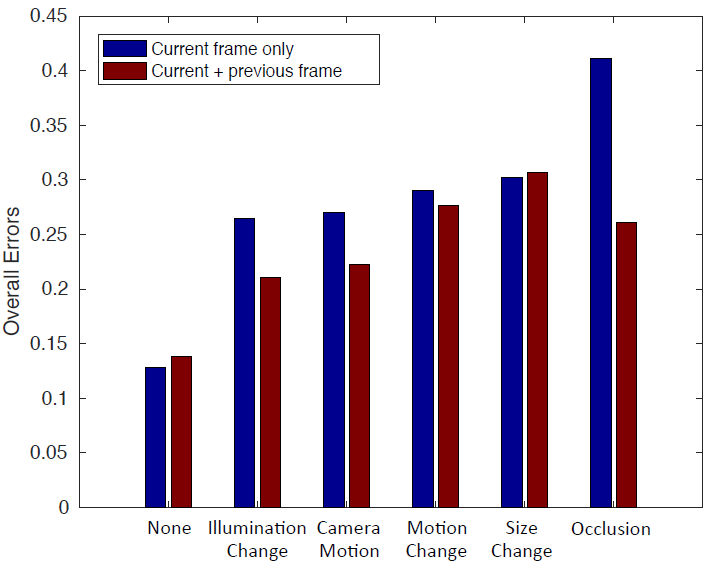
1.网络将前一帧与当前帧进行比较以找出当前帧中的目标对象。

2.网络充当本地通用“对象检测器”，只是简单地定位了最近的“对象”。

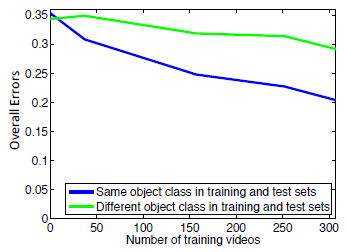
我们通过比较我们的网络性能（如图2所示）和另一个不接受前一帧输出的网络性能来区分这些假设网络（即，网络仅接收当前帧作为输入）。 对于这个实验，我们分别训练每个网络。如果网络没有收到前一帧作为输入，那么跟踪器可以仅作为本地通用对象检测器（假设2）。

图7显示了每个假设对于其的适用程度不同的跟踪条件。 例如，当存在遮挡或大的相机的运动，跟踪器从使用前一帧得到很大的好处，使追踪器能够“记住”正在追踪哪个对象，如图7所示在前一帧不包括在内的情况下，跟踪器的表现更差。 在这种情况下，假设1起着重要的作用，即追踪者是将前一帧与当前帧进行比较以找出目标对象。

另一方面，当尺寸改变或没有变化时，跟踪器在使用前一帧时的性能会稍稍差一点（或者近似于相同）。 在大尺寸变化下，相应的外观变化也是如此对于我们的网络要在当前帧和前一帧之间作出精确的比较来说太剧烈了。 因此，在这种情况下跟踪器表现的就像本地通用对象检测器一样，并且假设2占主导地位。 每个假设在不同跟踪条件的不同程度下都是成立的，如图7所示。

**图7.我们网络的整体跟踪误差，它接收当前帧和前一帧作为输入，和只接收当前帧作为输入帧的网络相比（越低越好）。 这个比较使我们能够消除两个可以解释我们的神经网络跟踪器是如何工作的假设（见第6.2节）。准确性和健壮性度量标准显示在附录中**

**6.3一般性与特异性**

我们的跟踪器能够将我们的训练中未发现的新物体归纳的有多好？ 对于这种分析，我们将测试集分为至少25个出现在训练集中相同类型的视频的对象和少于25个出现在训练集里同类型视频的对象。 图8显示，即使在训练集中没有（或很少）相似的对象，我们的跟踪器依旧表现良好。即使在视频不相关的对象被加入到训练集中，跟踪器的性能在仍在继续改善，因为我们的追踪器能够学习对象的外观变化和它的运动的关系，这种关系可以推广到新的物体。

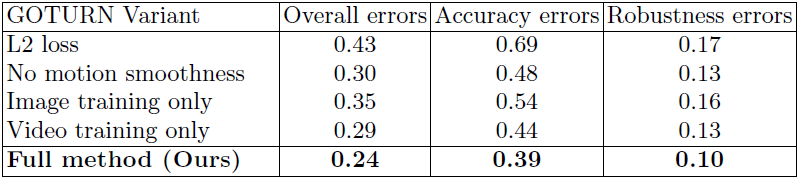
**图8.测试集中不同类型物体的整体跟踪错误作为我们训练集中的视频数目的函数（越低越好）。 类标签不被使用我们的跟踪器; 这些标签仅用于此分析的目的。 准确性和健壮性度量标准在附录中显示**

另外，我们的追踪器也可以专门追踪某些对象，效果特别好。 图8显示了对于至少25个同级别视频的测试对象出现在训练集中，在更多的训练视频中的这类对象被添加，我们获得了很大的提高。 这允许用户对特定应用选择专门针对的跟踪器。 例如，如果跟踪器正被用于自动驾驶，那么用户可以添加更多的人对象，自行车和汽车对象加入训练集，跟踪器将很好的学习跟踪这些对象。 与此同时，图8也证明了这一点我们的跟踪器可以跟踪未出现在我们训练集中的新物体，这在追踪非受控环境中的物体时非常重要。

**6.4夺格分析**

在表1中，我们展示了我们的系统中哪些组件对我们的跟踪器的表现其贡献最大。 我们用拉普拉斯分布的随机裁剪来训练我们的网络以教导我们的跟踪器偏好于较小的运动（例如，运动平滑度），如第4.2节所述。 表1显示了这一点与均匀抽样随机裁剪的基线的方法进行比较的好处（“没有运动平滑度”），正如已经完全做好了的分类[23]。正如展示的，我们通过从拉普拉斯分布中提取随机裁剪而减少了20%的错误。

表1还显示了与L2损失相比使用L1损失的好处。使用L1损失显着地将整体跟踪误差从0.43降低至0.24。 因为二级惩罚是相对接近0时，网络没有充分惩罚接近但不正确的产出，并且网络通常会输出一个稍大或稍小的边界框。当应用于一系列帧时，边界框会没有限制的增大或缩小，直到预测的边界框只是一个点或是整个图像。 相比之下，L1损失惩罚更厉害的结果只有稍微不正确，这使边界框的尺寸更接近正确大小，并防止边界框不受限制地收缩或增长。

**表1.将我们的完整GOTURN跟踪方法与各种修改版本进行比较来分析系统的不同组件的影响**

我们使用图像和视频的组合来训练我们的跟踪器。 表格1表明，考虑到图像和视频之间的选择，仅对视频进行培训给我们的跟踪器性能带来了更大的改进。 与此同时，对视频和图像进行培训可为我们提供最佳性能跟踪器。 用少量带标签的视频进行培训已经教会我们的跟踪器对于背景运动，平面外旋转，变形，照明更改和轻微遮挡不变。 从大量标签图像进行培训已经教会了我们的网络如何跟踪各种不同类型的对象。通过对视频和图像进行培训，我们的跟踪器学习跟踪各种视频和图像在不同条件下的对象类型，实现最高性能。

**7 总结**

我们已经证明我们可以训练一个通用对象跟踪器，它通过观看更多培训视频提高其性能。 测试期间时间，我们纯粹以前馈的方式运行网络，而不需要在线调整，使跟踪器以100 fps运行。 我们的跟踪器离线学习一个物体的外观与其运动之间的一般关系，让我们的网络以实时速度跟踪新物体。

**致谢。**我们感谢Toyota grant 1186781-31-UDARO and ONR grant 1165419-10-TDAUZ的支持。

**参考**

1. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: Computer Vision and Pattern Recognition, 2008. CVPR2008. IEEE Conference on. pp. 1{8. IEEE (2008)

2. Ba, J., Mnih, V., Kavukcuoglu, K.: Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755 (2014)3. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009.IEEE Conference on. pp. 983{990. IEEE (2009)

4. Bazzani, L., Larochelle, H., Murino, V., Ting, J.a., Freitas, N.D.: Learning attentional policies for tracking and recognition in video with deep networks. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11).pp. 937{944 (2011)

5. Bo, L., Ren, X., Fox, D.: Multipath sparse coding using hierarchical matchingpursuit. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. pp. 660{667. IEEE (2013)

6. Cehovin, L., Kristan, M., Leonardis, A.: Is my new tracker really better than yours?In: Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on.pp. 540{547. IEEE (2014)

7. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M.: Learning spatially regularized correlation \_lters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 4310{4318 (2015)

8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale

hierarchical image database. In: Computer Vision and Pattern Recognition, 2009.

CVPR 2009. IEEE Conference on. pp. 248-255. IEEE (2009)

9. Donahue, J., Hendricks, L.A., Guadarrama, S., Rohrbach, M., Venugopalan, S.,

Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual

recognition and description. arXiv preprint arXiv:1411.4389 (2014)

10. Dosovitskiy, A., Fischery, P., Ilg, E., Hazirbas, C., Golkov, V., van der Smagt, P.,Cremers, D., Brox, T., et al.: Flownet: Learning optical ow with convolutionalnetworks. In: 2015 IEEE International Conference on Computer Vision (ICCV).pp. 2758-2766. IEEE (2015)

11. Fan, J., Xu, W., Wu, Y., Gong, Y.: Human tracking using convolutional neuralnetworks. Neural Networks, IEEE Transactions on 21(10), 1610-1623 (2010)

12. Geiger, A.: Probabilistic Models for 3D Urban Scene Understanding from MovablePlatforms. Ph.D. thesis, KIT (2013)

13. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Computer Vision and PatternRecognition (CVPR), 2014 IEEE Conference on. pp. 580-587. IEEE (2014)

14. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: Unifying feature and metric learning for patch-based matching. In: Proceedings of the IEEEConference on Computer Vision and Pattern Recognition. pp. 3279-3286 (2015)

15. Hare, S., Sa\_ari, A., Torr, P.H.: Struck: Structured output tracking with kernels.In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 263-270.IEEE (2011)

16. Hong, S., You, T., Kwak, S., Han, B.: Online tracking by learning discriminativesaliency map with convolutional neural network. In: Proceedings of the 32nd International Conference on Machine Learning, 2015, Lille, France, 6-11 July 2015(2015)

17. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Ca\_e: Convolutional architecture for fast feature embedding.arXiv preprint arXiv:1408.5093 (2014)

18. Jin, J., Dundar, A., Bates, J., Farabet, C., Culurciello, E.: Tracking with deepneural networks. In: Information Sciences and Systems (CISS), 2013 47th AnnualConference on. pp. 1-5. IEEE (2013)

19. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(7), 1409-1422 (2012)

20. Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., Fei-Fei, L.: Large-scale video classi\_cation with convolutional neural networks. In: Proceedings ofthe IEEE conference on Computer Vision and Pattern Recognition. pp. 1725-1732(2014)

21. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Cehovin, L., Fernandez, G., Vo-jir, T., Hager, G., Nebehay, G., Pugfelder, R.: The visual object tracking vot2015challenge results. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops. pp. 1-23 (2015)

22. Kristan, M., Pugfelder, R., Leonardis, A., Matas, J., \_Cehovin, L., Nebehay, G.,Voj\_\_\_r, T., Fernandez, G., Luke\_zi\_c, A., Dimitriev, A., et al.: The visual object tracking vot2014 challenge results. In: Computer Vision-ECCV 2014 Workshops. pp.191{217. Springer (2014)

23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classi\_cation with deep convolutional neural networks. In: Advances in neural information processing systems.pp. 1097-1105 (2012)

24. Kuen, J., Lim, K.M., Lee, C.P.: Self-taught learning of a deep invariant representation for visual tracking via temporal slowness principle. Pattern Recognition48(10), 2964-2982 (2015)

25. Levi, G., Hassner, T.: Age and gender classi\_cation using convolutional neuralnetworks. Computer Vision and Pattern Recognition Workshop (CVPRW), 2015IEEE Conference on (2015)

26. Li, H., Li, Y., Porikli, F.: Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking. In: Proceedings of theBritish Machine Vision Conference. BMVA Press (2014)

27. Li, H., Li, Y., Porikli, F.: Deeptrack: Learning discriminative feature representations online for robust visual tracking. arXiv preprint arXiv:1503.00072 (2015)

28. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semanticsegmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3431-3440 (2015)

29. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In:Advances in Neural Information Processing Systems. pp. 2204-2212 (2014)

30. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. arXiv preprint arXiv:1510.07945 (2015)

31. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z.,Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision pp. 1-42 (2014)

32. Smeulders, A.W., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah,M.: Visual tracking: an experimental survey. Pattern Analysis and Machine Intelligence, IEEE Transactions on 36(7), 1442-1468 (2014)

33. Tao, R., Gavves, E., Smeulders, A.W.M.: Siamese instance search for tracking. In:Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(2016)

34. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 3119-3127 (2015)

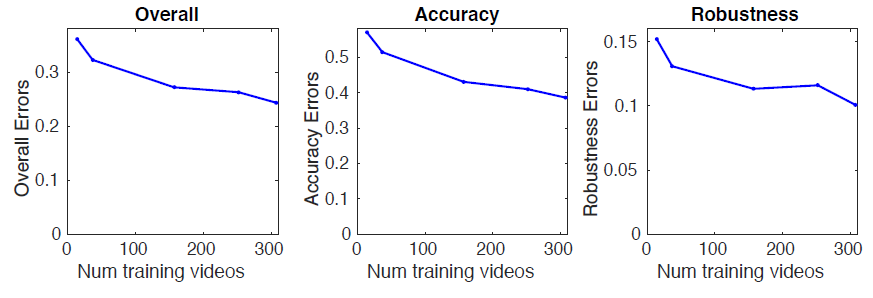
35. Wang, N., Li, S., Gupta, A., Yeung, D.Y.: Transferring rich feature hierarchies for robust visual tracking. arXiv preprint arXiv:1501.04587 (2015)

36. Wang, N., Shi, J., Yeung, D.Y., Jia, J.: Understanding and diagnosing visual tracking systems. arXiv preprint arXiv:1504.06055 (2015)

37. Wang, N., Yeung, D.Y.: Learning a deep compact image representation for visual tracking. In: Advances in neural information processing systems. pp. 809-817(2013)

38. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4353-4361 (2015)

39. Zhang, K., Liu, Q., Wu, Y., Yang, M.H.: Robust visual tracking via convolutional networks. arXiv preprint arXiv:1501.04505 (2015)

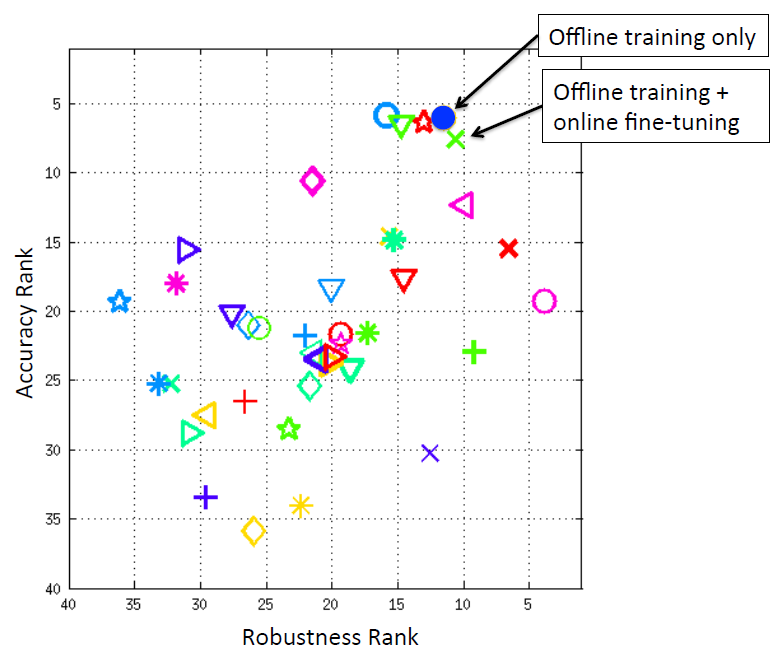
**A 离线训练**

**图9.跟踪性能作为训练视频数量的函数（越低越好）。 这一分析表明，通过标注更多的训练视频可以获得巨大的收益。**

我们的追踪器可以提高性能，因为它通过更多的离线数据训练。 通过观察更多的视频，GOTURN了解对象的外观随着他们的移动而变化。我们进一步分析了培训数据量对我们的跟踪器的性能的影响，如图9所示。我们看到那跟踪错误随着我们增加培训视频的数量而急剧下降。 我们正文中第6.1节所示的最先进的结果是只需307秒的视频训练之后获得的，时间从几秒到几分钟不等，平均每个视频有52个注释。 图9表明如果通过标记更多视频增加训练集的大小，则可以获得巨大收益。

**B 在线训练**

先前用于跟踪通用对象的神经网络跟踪器已经过训练在线[26,27,34,37,35,30,39,7,24,16]。 不幸的是，这种跟踪器训练非常慢，从0.8 fps [26]到15 fps [37]的范围，最高性能的神经网络跟踪器以1 fps运行[30,7,21]。 我们的跟踪器使用通用的方式进行离线训练，所以我们的跟踪器不需要在线培训。 结果，我们的跟踪器能够以100 fps的速度跟踪新物体。

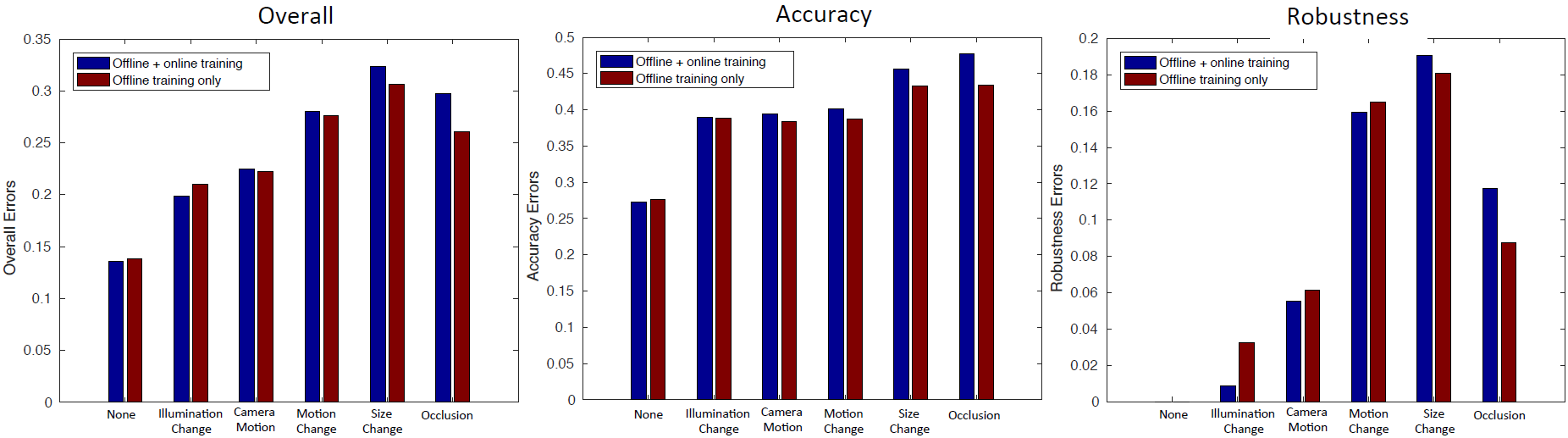
在图10和图11中，我们探讨了在线培训的好处。 我们使用跨界验证选择在线学习率为1e-9。 图10显示在线培训不会显着提高我们的离线培训程序。 正如可以预料的那样，在线培训使健壮性略有增加; 然而，这是以精确的代价，因为在线训练倾向于超过视频的前几个帧，并且不容易推广到新的变形或视点变化。 更详细的分析如图11所示。

**图10. 2014年VOT追踪挑战的追踪结果。 我们的跟踪器的性能用一个蓝色圆圈表示，在总体排名胜过以前的所有方法的（准确性和健壮性排名的平均值）。 我们的跟踪器版本在线训练以绿色X显示。两种版本的实现大致相同表现，证明我们的训练程序已经教会了网络如何跟踪各种物体。**

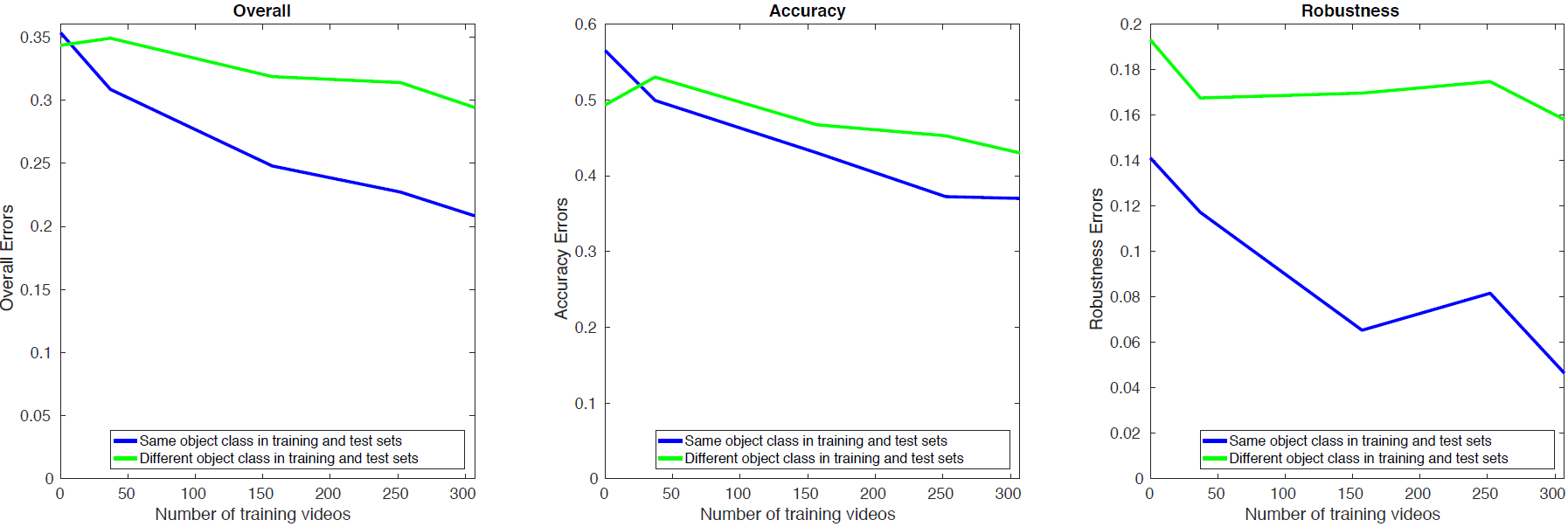
我们的培训过程中看到了许多变形，视点变化和其他变化的训练视频，因此我们的追踪器已经学会以通用的方式处理这种变化，并推广到新的对象。 虽然可能有其他方法来结合在线和培训，但我们的网络已经从其培训中学习了通用目标跟踪程序，并在没有任何在线训练的情况下实现最先进的跟踪性能。

**C 一般性与特异性**

在正文中，我们分析了跟踪器的一般性。 我们证明了我们的跟踪器可以给训练集里找不到的新对象归纳归类。 在同时，用户可以训练我们的跟踪器，通过提供更多的该类对象的训练示例来特别好地跟踪特定类别的对象。 这个如果跟踪器旨在用于特定的应用程序，那么它对某些类别的对象更普遍。



**图11.有和没有在线训练的跟踪器比较（越低越好）。两个版本的性能大致相同，说明我们的产品培训程序已经教会了网络如何跟踪各种对象。 在线培训可能会导致视频的前几帧出现问题，导致更多的错误。**

我们在图12中显示了这个实验的更详细的结果。分别分析准确性和健壮性，我们观察到一个有趣的模式。 由于培训视频的数量增加，对于出现在我们的训练集中的类型中没有出现在我们的训练集中的对象来说准确性误差同样降低。 另一方面，比起没有出现在训练集中的对象类型，健壮性错误的降低很多对于出现在我们的训练集中的对象类型更重要。

**图12.我们测试集中不同类型对象的整体跟踪误差作为我们的训练集中的视频数量的函数（越低越好）。 我们的跟踪器不使用类标签; 这些标签仅用于此分析的目的。**

因此，我们的跟踪器能够学习可以准确地跟踪的物体的通用属性，即准确地用边界框表示物体的边界。 另一方面，跟踪器的归纳能力健壮性更有限; 跟踪器很难跟踪复杂情景下未知物体的运动。 这个分析指向未来的工作，通过标记更多视频或学习用未标记的视频训练来提高健壮性。

**D 速度分析**

在正文中，我们将跟踪器的速度显示为整体的函数排名（按精度和稳健性排名的平均值计算）并显示我们拥有最低的整体排名，同时也是最快的追踪者之一。在图13中，我们展示了更详细的结果，展示了我们的跟踪器的速度作为精度等级和健壮性等级的函数。 我们的追踪器有第二高的准确性等级，最高健壮性等级之一和最高总体等级排名，同时运行在100 fps。 以前的神经网络跟踪器的速度从0.8 fps [26]到15 fps [37]，具有最佳性能的神经网络跟踪器在GPU上仅以1 fps 运行[30,7,21]，因为神经网络的在线训练是很慢的。 因此，通过执行我们所有的离线训练，我们能够使我们的神经网络跟踪器实时运行。

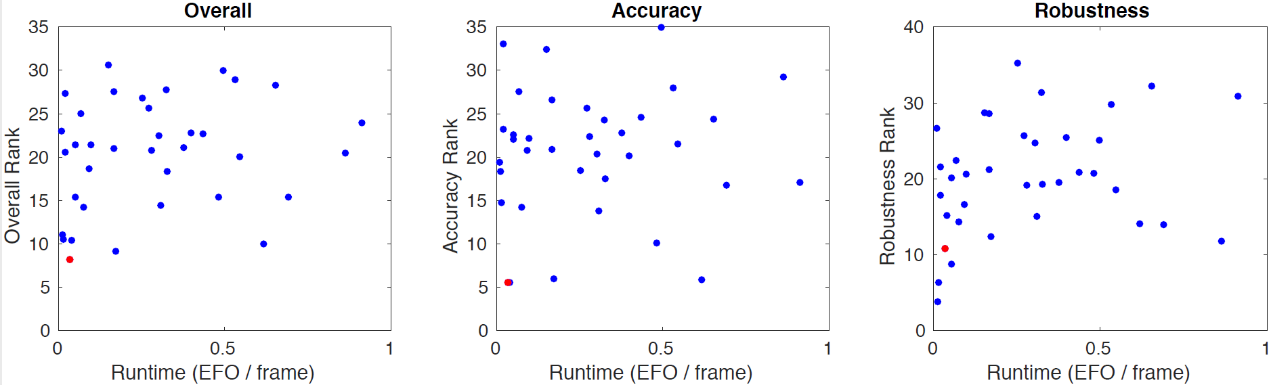


图13.我们的跟踪器的等级vs运行时间（红色）与2014年VOT Tracking Challenge（蓝色）的38个基准方法相比较。 每个蓝点表示一个单独的基准方法的性能（最好用彩色观看）。

**E 它如何工作？**

在正文中，我们探讨了我们的跟踪器如何作为两者的组合来工作假设：

1. 网络将前一帧与当前帧进行比较以找出在当前帧中的目标对象。

2. 网络充当本地通用\对象检测器“，只是简单地定位了最近的“对象”。

我们通过比较我们的网络和不接受前一帧作为输出的网络的性能表现来区分这些假设。 在图14中，我们展示了这个实验的更多细节还有准确性和健壮性排名。 有关更详细的解释结果见正文的第6.2节。

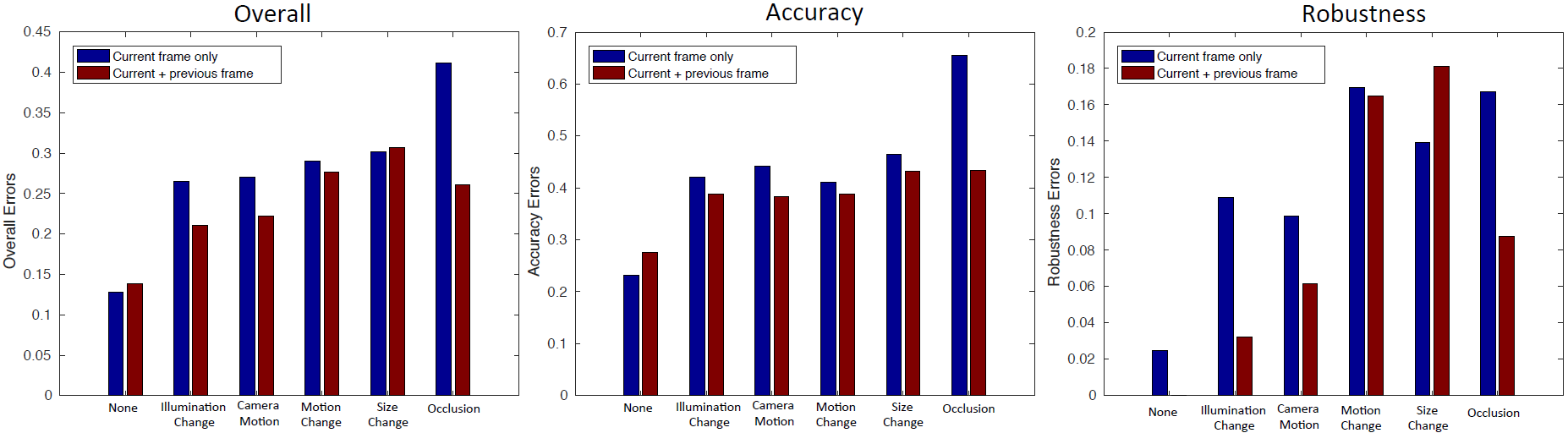


图14.我们网络的跟踪错误，它接收当前帧和前一帧作为输入，与仅接收当前帧作为输入的网络进行比较（越低越好）。 这种比较使我们能够在两个假设之间消除歧义，这两个假设可以解释我们的神经网络跟踪器是如何工作的（参见第6.2节正文）。

**F 运动平滑度分布**

在正文的第4.2节中，我们描述了我们如何使用随机裁剪来实现隐式地将小运动比大运动更可能的想法编码。为了确定使用哪种分布来编码这个想法，我们分析了在训练集中发现的物体运动的分布。 这种运动分布可以在图15中看到。从这个图中可以看出，每一个分布都可以用拉普拉斯分布建模。 因此，我们的随机裁剪程序使用拉普拉斯分布。 请注意，训练集是仅用于确定分布的形状（即拉普拉斯）; 我们用我们的验证集以确定分布的比例参数。

更详细地说，假设帧t-1中的边界框由下式给出（，，，）其中cx和cy是边界中心的坐标框和w和h是相应的宽度和高度。 然后是边界框在时间t可以看作是从分布中得出的：

(5)

(6)

(7)

(8)

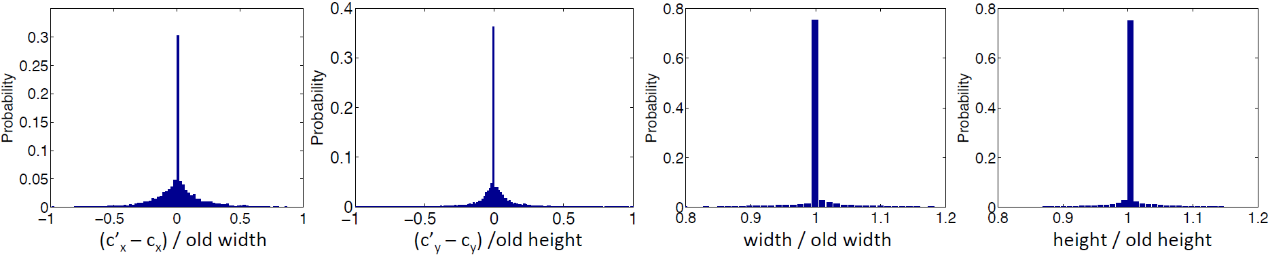


图15.我们的训练集中两个连续帧中边界框大小和位置变化的统计。

随机变量，，，其中（）在时间t处使用上述相同的表示参数化的边界框。 对于随机变量，我们可以将这些表达式重写为

(9)

(10)

(11)

(12)

这些随机变量在训练集上的经验分布为如图15所示。

**G 层数**

在图16中，我们探讨了在神经网络的顶层上改变完全连接数量对跟踪性能的影响。 这些完全连接层 在对每个图片进行初始卷积之后应用。 这表明使用3个完全连接的层执行比使用2层或4层更好。 但是，性能与此类似2,3或4个完全连接的层，表明即使3个完全连接层次最佳，跟踪器的性能对这个参数不是特别敏感。

**H 数据增强**

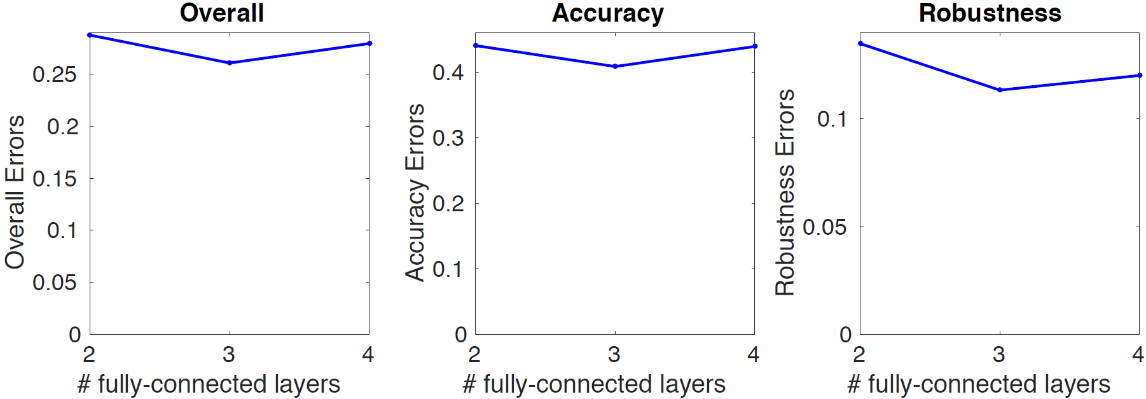
在图17中，我们探讨了改变为每批训练集而创建的增强图像数量的影响。 请注意，新的增强图像是现场为每一批创建的。 但是，改变增强图片的数量改变了由真实图像组成的每一批所占的百分比。 我们的批量大小是50，所以我们可以在每批增强图像中从0到49改变（为至少1个真实的空间留出空间）。

图16.跟踪性能作为神经网络的完全连接图层数量的函数（越低越好）。

如图17所示，当每一批用49个增强图像后可以获得最佳性能，即仅使用1个真实图像，其余部分被增强。 但是，性能都很相似，因为所有增强图像的值都大于20。在我们的情况下（批量大小为50），这表明只要批次中至少有40％的图像是增强的，性能就会相似。 增强图像显示与真实图像相同的例子，但目标对象有翻转或具有不同的规模时不相似。 增强图像因此教导网络怎样根据翻转和规模变化划定边界框。

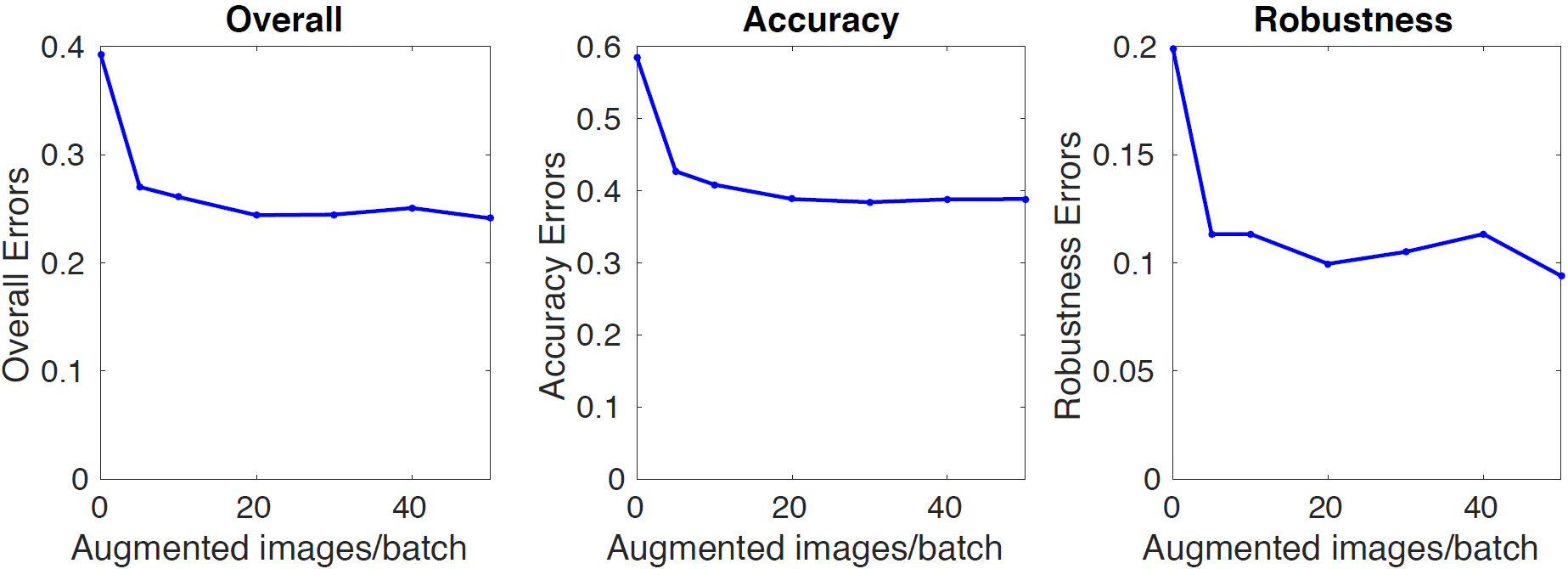


图17.跟踪性能作为每批次中的增强图像数量的函数（越低越好）。 请注意，新的增强图像是在现场创建的。

**I 训练集**

我们的训练集取自ALOV300 ++ [32]。 为了确保有与我们的测试集没有重叠，我们从我们的训练集中移除了7个视频。 这些视频是：

- 01-Light video00016

- 01-Light video00022

- 01-Light video00023

- 02-SurfaceCover video00012

- 03-Specularity video00003

- 03-Specularity video00012

- 10-LowContrast video00013

删除这7个重叠视频后，在训练集和测试集视频之间没有重叠。

**J 详细结果**

我们的方法的详细结果与其他38种提交给VOT 2014 Tracking Challenge [22]的方法相比较的结果如表2所示。VOT 2014 Tracking Challenge由两种类型的实验组成。 在里面在第一个实验中，跟踪器是用一个精确的地面实况边界初始化的（“精确”）。在第二个实验中，跟踪器用a有噪声的边界框初始化，这是稍微移动目标对象（“有噪声”）。对于有噪声的初始化实验，使用相同的15个噪声初始化每个跟踪器，所显示的结果是跨越这些初始化跟踪器性能的平均值。 这个实验让我们可以确定每个跟踪器初始化错误的鲁棒性。 这个嘈杂的初始化程序模仿了噪音自动初始化过程或嘈杂人类初始化。

跟踪器使用两个标准跟踪指标进行评估：准确性和健壮性[22,6]。 视频的每一帧都用一个数字标注属性：遮挡，照度变化，运动变化，尺寸变化和相机运动。 跟踪器分别在精度和鲁棒性方面为每个属性进行排名，然后跨属性平均得到每个跟踪器的最终准确性和健壮性等级。 准确性和对稳健性排名进行平均得到总体排名，如表2所示。

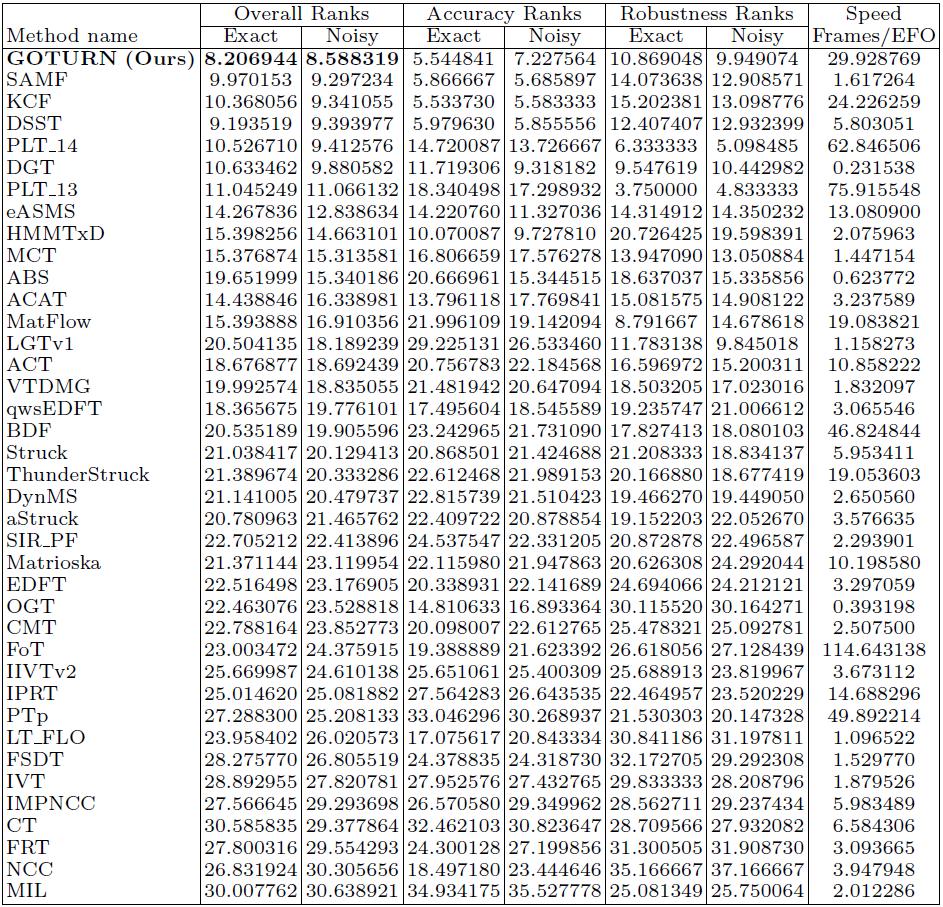


表2. VOT 2014 tracking challenge,的全部结果，我们的方法（GOTURN）提交给竞赛的其他38种方法相比较。 我们初始化了跟踪器有两种不同的方式：精确的地面真实边界框（“精确”）和一个有噪声的边界框（“有噪声”）。

1. Our tracker is available at http://davheld.github.io/GOTURN/GOTURN.html [↑](#footnote-ref-1)