**摘要**

空间冗余广泛存在于视觉识别任务中。也就是说，一张图像或者一个视频帧的明显（discriminative）特征通常只对应于一部分（a subset of）像素，而剩余的区域对眼前的（at hand）工作无关紧要。因此，使用同等算力的处理所有像素的静态模型导致了相当大的（considerable）冗余时空消耗。在这篇论文中，我们将图像识别问题视作（formulate）一个连续的（sequential）、由粗到细的特征学习过程，模仿人眼视觉系统。具体而言（Sepecially），提出的GFNet（Glance and Focus Network）首先对输入的图像在低卷积程度提取一个全局的代表（representation），然后策略性地（Strategically）处理（attend to）一系列小区域来学习更细的特征。连续的进程自然有利于在测试期间的（facilitates）适应化推理（adaptive inference），因为一旦模型足够确定其预测，它可以被终止，这样就能避免更多冗余计算。在我们的模型定位明显区域作为一个加强的学习任务是没有意义的，所以除了分类标签（ classification labels）以外，我们不需要额外的手工标注（manual annotations）。GFNet是通用而灵活的，因为它和现有的任何主干网络兼容，比如MobileNets, EfficientNets和TSM，这些网络都可以很方便地作为提取器（extractor）部署。我们的方法在很多有关图像分类和视频识别的实验上效率极高。例如，在不牺牲精确性的情况下，减少了MobileNet-V3在iPhone XS Max的平均延迟。代码和预训练模型在https://github.com/blackfeather-wang/GFNet-Pytorch。

**关键词Index Terms**--动态神经网络dynamic neural networks,高效率深度学习 efficient deep learning,图像识别 image recognition,增强学习 reinforcement learning,视频识别 video recognition

**1 INTRODUCTION**

高分辨率图像或者视频的普及使得深度学习算在很多有挑战性的视觉任务上达到了超越人类水平的性能。最近的工作表明了深度模型的精确性可以通过提高图像分辨率来增强，例如提升分辨率到480\*480或者甚至更高。尽管使用现代相机来获取高质量数据不难，但实际上在上面执行视觉识别（visual recognition）却很有挑战，这是因为深度神经网络引入的高计算开销和高内存足迹（memory footprint）。在现实世界的应用，例如在边缘节点的视觉识别，基于内容的图像搜索或者自动驾驶。计算转化为延迟（latecy）和电力消耗，这些应该出于安全和经济性来缩小。

本文旨在从空间冗余（spatial redundancy）来减少高分辨率视觉识别的计算开销。事实上，深度模型能够在很少的，像狗的头或者鸟的翅膀这样的分类鉴别碎片（class-discriminative patches）精确地进行对象识别。这些区域主要在图像的小面积区域，所以不需要太多的计算资源。因此，如果能够动态地定位每个图象的”分类鉴别“（class-discriminative）区域，只在这些小区域执行高效的推断，那么空间类型（spatial-wise）的计算冗余可以在不牺牲精确性的前提下显著减小。为了实现这个想法，我们需要解决两个挑战：1）如何高效地识别（identify）分类鉴别（class-discriminative）区域（在不需要借助于额外的人工注释情况下）2）在鉴别区域（discriminative regions）的数量/体积在不同的输入情况下各不相同的情况下，如何适应性地分配算力到每一个单独的图像。

为了解决前面提到的问题，我们提出了（present）一个两阶段的框架，名为瞥见和集中（glance and focus），这是受到由粗到细的人眼视觉系统识别过程启发而来的。具体来说，区域选择操作被定义为一个序列决策过程，在每一个步骤中我们的模型处理一个相对小的输入，从而产生一个做出带有置信度分数的分类预测，以及下一步的区域建议。每一步由于输入规模缩小，可以被高效地完成。例如，推断一个96\*96的图像算力消耗仅仅是224\*224的18%。整个序列进程（sequential process）从在一个下采样比例（ down-sampled scale）（例如96\*96）处理整个图像开始，作为初始步骤，我们称为一瞥步骤（glance step），在这个步骤中模型使用全局特征产生一个快速预测。在实践中，我们发现图像拥有明显特征的很大一部分已经能够在glance step被正确分类，这和在[17]中的观察一致。当glance step不能足够确认自己的预测时，它将输出一个最具鉴别力的区域的建议（图像中的相对位置），供后续步骤处理。因为建议区域常常是原始图像原分辨率的很小的一个区域，我们称这些接下来的步骤叫做聚焦阶段（focus stage）。这个阶段将逐步迭代定位和处理类别区分（class-discriminative）的图像区域，促进了以一种自适应方式提前终止。也就是说，在每一个图像输入，决策过程可以被有条件地（conditioned）动态中断。如图1，我们的方法分给不同的图像不均衡的算力，在整体效率上有所提升。我们称这个方法为Glance and Focus Network (GFNet)。

GFNet的一个明显特征就是它是一个通用框架，其中分类器（classifier）和区域建议网络（region proposal network）被视为两个独立模块，因此，任何现有的主干模型例如MobileNets [9], [13], [18], CondenseNets [12], [19], ShuffleNets [20], [21] and EfficientNets [7]都可以被作为我们的特性提取器（feature extractors）部署。这将我们的方法和其他的方法区分开来。另外，我们专注于在适应性推断环境（adaptive inference setting）提升计算效率，而现有的其他工作目的是在给定序列长度下提升精度。

除了高计算效率，GFNet还有其他方面的优势。例如内存消耗大大降低，并且它是独立于原始图像分辨率，只要我们固定聚焦区域的大小。另外，GFNet的计算开销可以在不需要额外训练的情况下调整（通过简单地调整终止条件（termination criterion））。这使得GFNet能够充分地灵活利用计算资源或者在最低能耗前提下达到目标性能，这是一个现实世界的需求，例如搜索引擎和移动应用。

本文实证验证了GFNet在图像分类（ImageNet [1]）和视频识别任务（Something-Something V1&V2 [22]和Jester [23]）中的效果，使用了各种骨干模型（例如MobileNet-V3 [18]、RegNet [24]、EfficientNet [7]和TSM [25]）。考虑了两种实际情况：一是有预算的批量分类设置 [26]，测试集在给定的计算预算下进行分类；二是任意时间预测设置 [26]，网络可以被强制在任何时间点输出预测。我们还在iPhone XS Max和NVIDIA 2080Ti GPU上测试了GFNet的实际速度。

实验结果表明，GFNet在理论和实证上都有效地提高了最先进网络的效率。例如，当MobileNets-V3和ResNets作为骨干网络时，GFNet在达到相同精度的情况下，相对于原始模型，可以使乘加操作减少1.4倍和3倍，而在iPhone XS Max上的实际加速比（通过平均延迟测量）分别为1.3倍和2.9倍。

本文的一部分结果最初发表在会议论文中[28]。但是，这个版本在几个重要方面扩展了我们早期的工作：

•我们通过提出对比奖励函数（第4.1节）改进了强化学习算法，用于学习补丁选择策略。

•我们引入了多尺度GFNet（MS-GFNet），允许根据输入和计算预算动态调整补丁大小（第4.2节）。实证结果表明，MS-GFNet不仅提高了准确性，而且为在线调整提供了更灵活的计算成本范围。

•我们将GFNet扩展到高分辨率或非面向对象的图像识别和视频识别，并在广泛使用的基准测试中报告了令人鼓舞的结果（第5.3节，第5.4节）。

•我们进行了实验验证GFNet也可以有效地加速GPU设备上的大批量推理（表1），而原始版本仅考虑了在iPhone上的单张图像推理。

•我们还提供了其他分析结果，包括全面的消融研究和对训练超参数的讨论（第5.5.2节，第5.5.3节）。

**2 相关工作**

**2.1 计算效率高的网络**

现代深度学习模型，如卷积神经网络（CNN），通常需要大量的计算资源。因此，许多研究工作专注于降低网络的推理成本。一个有前途的方向是开发高效的网络架构，如MobileNets[9]、[13]、[18]，CondenseNets[12]、[19]，ShuffleNets[20]、[21]和EfficientNet[7]。由于深度网络通常存在大量冗余权重[29]，一些其他方法专注于剪枝[30]、[31]、[32]、[33]或权重量化[34]、[35]、[36]。另一种技术是知识蒸馏[37]，它训练一个小网络来复制大模型的预测。

**2.2 空间冗余**

最近的研究表明，在推理深度网络时存在相当大的空间冗余。已经提出了几种方法来减少空间维度中的冗余计算。OctConv（54）通过使用低频特征减少了空间分辨率。空间自适应计算时间(SACT)动态调整不同图像区域执行的层数。[55] 和 [50] 中提出的方法跳过了特征映射上一些不太重要的区域的计算。这些工作主要是通过修改卷积层来减少空间冗余，而我们提出的是以顺序的方式处理图像。我们的方法是通用的，因为它不需要改变网络架构。

**2.3 视觉注意模型**

我们的GFNet与视觉注意模型有关，这些模型与人类的感知相似，因为人类通常会集中注意力在环境的部分来执行识别。许多现有的作品将关注机制集成到图像处理系统中，特别是在语言相关的任务中。例如，在图像标题和视觉问答中，模型被训练去集中注意力在生成单词序列时相关区域上。对于图像识别，关注机制通常被利用来从一些任务相关区域中提取信息。

我们的GFNet与[14]中提出的循环视觉注意模型有一些相似之处。然而，我们的方法在两个重要方面与它不同:1)我们采用灵活通用的基于CNN的框架，可以兼容各种CNN来实现最先进的计算效率，而不是坚持纯RNN模型；2) 我们的网络专注于执行自适应推理以获得更高的效率，并且循环过程可以根据每个输入而被终止。 借助这些设计创新，所提出的GFNet已在ImageNet上实现了理论计算效率和实际推理速度方面的最新性能。此外，[64]也与我们在使用强化学习选择重要特征方面有类似的想法，但它不是基于CNN或图像数据的。

**2.4 自动缩放进入方法**

特别是，一些现有的方法提出自动识别和关注输入图像的某些信息性细节。作为代表性的例子，RA-CNN、TASN 和 NTS-Net 通过放大识别局部模式（如鸟类的后脑颜色）来区分具有微妙视觉差异的细粒度类别。S3N 和 MGE-CNN 利用类激活映射（CAMs）定位并突出视觉证据以用于细粒度图像识别。该想法也被应用到车辆重新识别、图像翻译和零点识别中。

在这方面，与GFNet类似的工作可能是RACNN。但是，从我们的动机和技术贡献来看，GFNet与RA-CNN有着根本性的不同。首先，GFNet的目标是提高深度网络的效率。我们提出的“Glance and Focus”设计使得GFNet能够以小的输入(例如，96x96下采样图像或局部补丁)有效地定位和利用具有类别区分性的图像区域，仅占原始224x224图像处理所需FLOPs的18%。相反，RA-CNN 15在计算上是密集的，因为它遵循了与我们不同的“全局到局部”原则，即所有样本都必须首先以高分辨率处理(例如，224x224)，而输入中最有信息的区域将被裁剪放大(例如，到224x224)，并递归地作为新的输入。此外，除了在效率方面的优势之外，我们还引入了一种新的自适应推理算法，进一步显著降低了推理的总体计算成本。

第二，GFNet有能力以任何形状或大小的灵活方式捕捉分类区别区域。相比之下，RA-CNN旨在从输入图像片段中递归裁剪信息子区域，逐步缩小注意窗口，并最终将注意力集中在原始图像的一个小区域上。这种机制是量身定制的用于细粒度视觉识别任务，但是在我们考虑的更一般和复杂的情况下（例如，在ImageNet和Something-Something V1 / V2等大规模综合图像/视频基准测试中）却是次优的。 在图2中提供了这个问题的说明。

第三，按照我们的表述，补丁提议网络的训练需要探索整个图像中所有潜在的与任务相关的区域，而这不能通过提出的训练技术来实现。相比之下，我们通过开发一种基于强化学习的新型训练算法来解决这个问题。正如我们通过广泛的实证结果所验证的那样，我们的算法产生了一种有效且灵活的补丁选择策略。

**3. GLANCE AND FOCUS NETWORKS (GFNET)**

在本节中，我们介绍我们方法的详细信息。为了便于理解，我们在这里考虑识别图像的任务。我们将在第5节中展示，对于处理视频来说，采用相同的技术是有效的。如上所述，像CNN这样的深度网络能够通过“类别判别”图像区域，例如狗的脸或鸟的翅膀，产生准确的图像分类结果。受到这一观察的启发，我们提出了一个GFNet框架，旨在通过在最小的图像区域上进行计算来提高模型的计算效率，从而获得可靠的预测结果。具体来说，GFNet根据它们对所需任务的贡献，逐渐地将计算分配到图像的不同区域，而在推理过程中，一旦网络足够自信，这个过程就会终止。

**3.1 概述**

在这个子部分中，我们首先概述所提出的GFNet（如图3所示）。其组成部分的详细信息将在第3.2节中介绍。

给定尺寸为H×W的图像x，我们的方法使用一系列H’×W’更小的输入来处理它，其中H’< H; W’< W。这些输入是从图像的特定位置裁剪的图像补丁（除了x1，稍后会进行描述）。每个补丁的具体位置是由网络动态确定的，利用所有先前输入的信息。

理想情况下，输入对分类的贡献应该按顺序递减，这样计算资源首先用于最有价值的区域。然而，对于任意给定的图像，当生成第一个补丁x1时，我们没有任何具体的先验知识来确定哪些区域更重要。因此，我们简单地将原始图像x调整大小为H’×W’，作为x1，这不仅避免了在随机定位初始区域时浪费计算资源在不重要的区域的风险，而且提供了有助于确定接下来的补丁位置的必要全局信息。

**推断**。GFNet的推断过程被制定为基于输入序列的动态决策过程。在第t步，骨干特征提取器（fG或fL）接收输入xt，模型产生一个softmax预测pt。然后，将pt的最大值（即maxj ptj，根据先前的工作[17]，[26]将其视为置信度）与预定义的阈值进行比较。如果maxj ptj > ，则顺序过程停止，并且pt将输出为最终预测结果。否则，将决定下一张图像补丁xt+1的位置，并将xt+1从图像中裁剪作为第（t + 1）步的输入。请注意，预测pt和xt+1的位置是使用两个递归网络获得的，从而它们利用了所有先前输入的信息。通过设置来限制输入序列的最大长度为T，而其他置信度阈值根据给定计算预算的实际要求确定。有关获取阈值的详细信息，请参见第3.4节。

**训练**。在训练过程中，我们通过设置来禁用早期终止，并确保所有预测都具有高置信度。对于补丁定位，我们训练网络选择最大化在相邻两步之间 softmax 预测增量的补丁的能力。换句话说，我们寻找最具类别区分性的图像补丁，这些补丁尚未被网络看到。此过程利用策略梯度算法来解决不可微分性问题。

## 3.2GFNet架构

提出的GFNet架构由四个组件组成:一个全局编码器fG，一个局部编码器fL，一个分类器fC，图像块选择网络π。

**全局编码器fG和局部编码器fL**都是我们用来从输入中提取深层表示的主干网络，他们使用相同架构但是不同参数的网络，前者适用于调整过大小的原始图片，而后者应用于选定的图像补丁，我们使用两个网络而不是一个网络，因为我们发现低分辨率输入x1和高分辨率局部补丁的规模之间存在差异，这导致单个编码器的性能下降

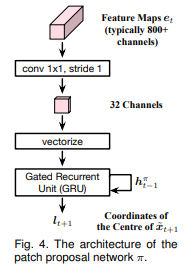
**分类器fC**是一个递归网络，它聚合来自所有先前输入的信息，并在每个步骤产生预测。我们假设第tth个输入xt被输入到编码器，获得相应的特征映射集。我们对et执行全局平均池化以获得特征向量et，并通过产生预测pt

其中hCt−1是fC的隐藏状态，在（t−1）th步骤中更新。注意，没有必要维护分类器的特征图，因为分类通常不依赖于它们所包含的空间信息。循环分类器fC和上述两个编码器fG、fL在具有以下分类损失的情况下被同时训练：

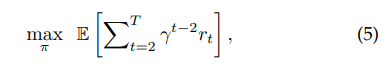
补丁建议网络π是另一个确定每个图像补丁位置的递归网络。假设π的输出用于不可微的裁剪操作，我们将π建模为代理，并使用策略梯度方法对其进行训练。具体而言，它在tth步接收xt的特征图et，并从由π参数化的分布中随机选择定位动作lt+1

：

其中lt+1∈[0，1]2被公式化为下一个补丁xt+1的中心归一化坐标。在这里，我们在训练期间使用高斯分布，其平均值由π输出，标准差预定义为超参数。在测试时，对于确定性推理过程，我们简单地采用平均值lt+1。我们用hπt−1表示π内保持的隐藏状态，它聚合了所有过去特征图{e1，…，et−1}的信息。请注意，我们不在et上执行任何池化，因为特征图中的空间信息对于定位判别区域至关重要。另一方面，我们通过使用1×1卷积来减少特征通道的数量，从而节省了计算成本。这样的设计放弃了对分类有价值但对定位没有必要的部分信息。π的体系结构如图4所示。

在训练期间，在获得lt+1之后，我们从原始图像x中裁剪以lt+1为中心的H0×W0区域作为下一个输入xt+1，并将其输入到网络中以产生预测pt+1。然后，补丁建议网络π接收到动作lt+1的奖励rt+1，其被定义为地面实况标签上的softmax预测概率的增量，即

其中y∈{1，…，C}是C类中x的标签。π的目标是最大化折扣奖励的总和：



其中γ∈（0，1）是预定义的贴现因子。直观地，通过等式（5），我们强制π选择补丁，使网络能够以尽可能少的补丁以高置信度产生正确的预测。本质上，我们训练π来预测每一步对图像分类最有利的区域的位置。注意，这个过程也考虑了之前的输入，因为我们计算预测概率的“增量”

图4 补丁建议网络π的体系结构

## 3.3训练策略

为了确保GFNet得到适当的训练，我们提出了一个三阶段的训练方案，其中前两个阶段是必不可少的，第三个阶段旨在进一步提高性能。

**第一阶段**：首先，我们没有将补丁建议网络π集成到GFNet中。相反，我们在每个步骤随机裁剪在整个输入图像上具有均匀分布的补丁，并训练fG、fL和fC以最小化分类损失Lcls（等式（2））。在这个阶段，网络被训练以适应任意的输入序列。

**第二阶段**：我们固定从第一阶段获得的两个编码器和分类器，并调用随机初始化的补丁建议网络π来决定图像补丁的位置。然后，我们使用策略梯度算法来训练π，以最大化总奖励（等式（5））。

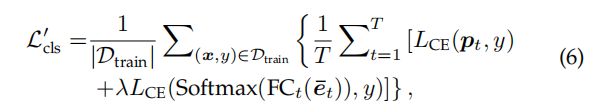
**第三阶段**：最后，我们用第二阶段的固定π对两个编码器和分类器进行微调，以通过学习的补丁选择策略提高GFNet的准确性。

**3.4 实施细节**

**fG和fL的初始化。**我们使用ImageNet预训练模型初始化本地编码器fL。由于全局编码器fG以较低的分辨率处理调整后的图像，我们首先对ImageNet的预训练模型进行微调，并将所有训练样本的大小调整到H’×W‘，然后使用微调参数初始化fG。

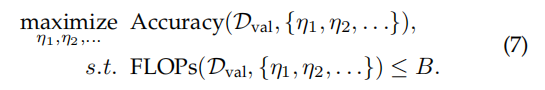
**递归网络。**我们在分类器fC中采用门控递归单元（GRU）[82]和补丁建议网络π。对于MobileNets-V3和效率网，我们使用一个全连接的分类器的级联来进行有效的推断。详情详见附录A.1。

**规范网络。**在我们的实现中，我们向等式中添加了一个正则化术语(2)，旨在保持两个编码器学习线性可分表示的能力，即



其中，λ > 0是一个预定义的系数。在此，我们为每一步定义了一个全连接层FCt（·），并利用FCt计算特征向量***e*¯***t*上的软最大交叉熵损失。请注意，当最小化等式(2)时，这两个编码器没有被直接监督，因为所有的梯度都通过分类器fC，而等式(6)明确地强制执行一个线性化的深度特征空间。

**置信阈值。**GFNet的一个显著优点是，通过改变置信度阈值{η1，η2，……}，其计算成本和推理延迟都可以根据实际需求进行在线调整。为了解决它们的值，我们考虑了一个预算批分类[26]场景，其中模型需要在给定的计算预算B > 0内对一组样本Dtest进行分类。设精度（Dval，{η1，η2，…}）和FLOPs（Dval，{η1，η2，…}）表示GFNet在具有{η1，η2，…}的验证集Dval上的精度和计算成本。然后，通过求解以下优化问题，可以得到阈值：

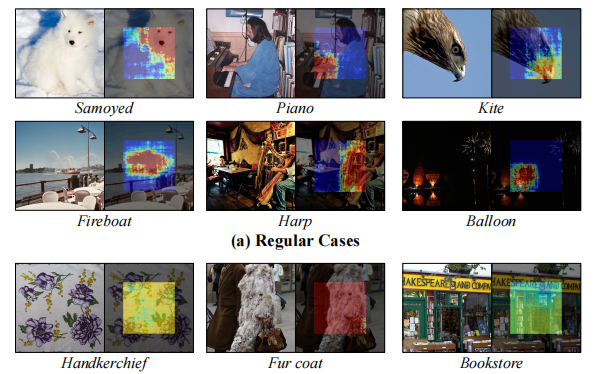


除非另有规定，问题(7)将通过以下程序来解决。我们假设获得最终预测的概率在第t步为qt，相应的计算成本或延迟为Ct。然后每个样本的平均成本可以计算为Σ*t qtCt*，导致约束|Dval|Σ*t qtCt*≤b。我们可以为一个适当的qt解决这个约束并且确定验证集上的阈值ηt。在我们的实现中，在[26]之后，我们让*qt* =*z*(1 *− q*) *t−*1 *q*，其中z是一个归一化常数，以确保Σ *t qt* = 1, 和0*<q <*1是要求解的变量。

**策略梯度算法。**我们实现了[83]提出的非近端策略优化（PPO）算法来训练补丁建议网络π。详情详见附录A.2。

**4 针对GFNET的改进技术**

在本节中，我们提出了一个对比奖励函数和一个可调整大小的补丁机制，以进一步提高GFNet的计算效率。前者提高了学习到的补丁选择策略的性能。后者使单个GFNet在其计算成本在线调整时实现较高的精度。第5.2.4节和第5.2.5节验证了这两种改进技术的有效性。



(b) 即使是随机策略也能持续获得较大回报的情况

图5 当第2个补丁的中心位于GF-ResNet-50（T =5，H‘ =W‘ = 96）中的每个位置时，奖励的价值用热图表示。原始图像显示在左边。我们假设(b)中显示的样本可能会混淆使用等式中提出的奖励的补丁选择策略的学习(4)，在那里所有的行动都将被鼓励。

**4.1对比奖励**

与在等式中提出的奖励在一起(4)，我们理想地希望补丁建议网络π能够找到最大化置信度增加的补丁。然而，根据经验观察，在某些情况下，即使随机选择补丁也会带来巨大的奖励。示例如图5 (b).所示热图中的颜色表示当第二个斑块（˜***x***2）的中心位于每个相应位置时的置信值。

我们假设这些实例可能会混淆π的训练，因为每一个行动都会因为其巨大的奖励而受到鼓励。为了缓解这一问题，我们提出了一个对比奖励函数，即



其中，E˜***x****t*+1*~*RandomCrop(***x***) [*p*(*t*+1)*y* ]是当第（t+1）个输入~***x****t*+1是随机从图像中裁剪时的地面真实标签上的期望软最大概率。使用等式(8)，将把π采取的行动与随机政策的平均效果进行比较，以确定是否应该鼓励它。因此，只有在战略性地选择补丁会导致网络预测发生显著变化的情况下，才会产生不同的回报。如果所有可能的行动都同样重要，并且因此学习π的意义不足，那么奖励的价值将在零左右，以避免混淆训练过程。

**4.2多尺度GFNet**

如前所述，普通的GFNet将补丁的大小（即H‘×W‘）固定为一个预定义的超参数。然而，这个约束限制了单个GFNet在其计算成本在大范围内在线调整时实现高计算效率的灵活性。在给定计算预算的前提下，采用较小的补丁的大小通常比使用更大的补丁具有更高的精度。然而，一旦可用的计算资源相对足够，较大的补丁通常会超过较小的补丁。关于这一现象的实证结果见表3。

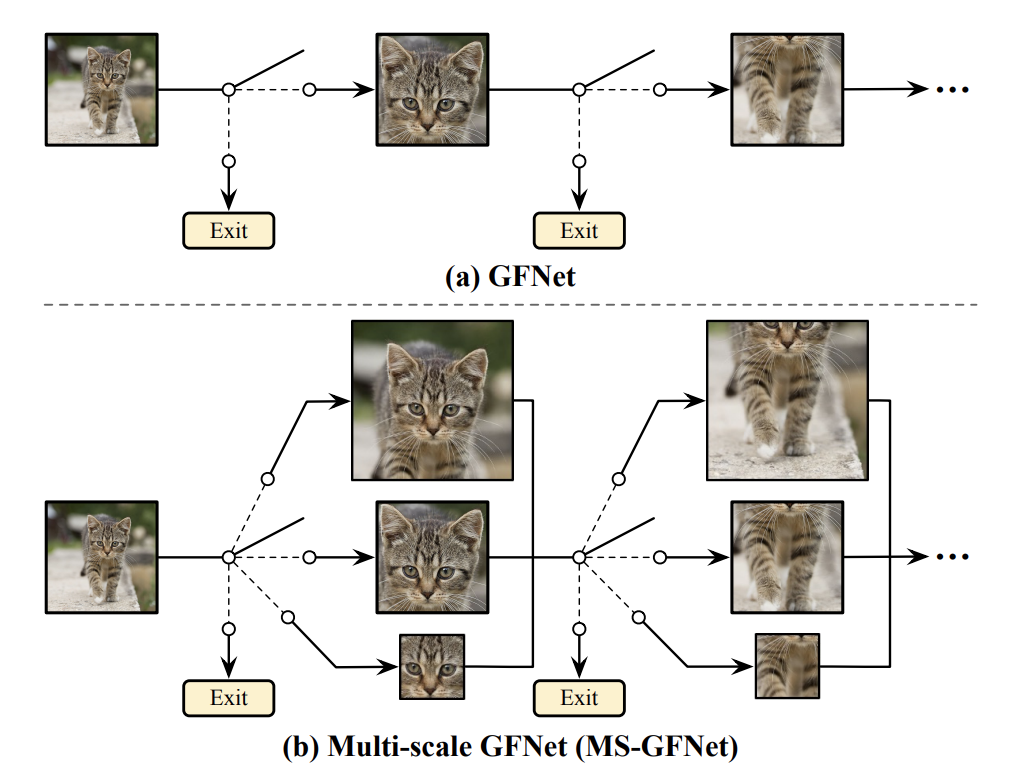


图6. 纯 GFNet 和多尺度 GFNet (MS-GFNet) 的比较。后者可以使模型动态地调整补丁的大小，从而可以在小的计算预算和相对充足的计算预算下都实现高准确度。请注意，GFNet 和 MS-GFNet 的计算成本可以在线调整，无需额外的训练。

较小的补丁尺寸通常比使用较大的补丁尺寸具有更高的准确性。然而，一旦可用的计算资源相对充足，较大的补丁通常表现更好。这种现象的经验结果可以在表3中找到。

受此问题的启发，我们提出了一个多尺度GFNet (MS-GFNet)，如图6所示。在测试时，处理每个图像补丁后，MS-GFNet确定是否像纯GFNet一样输出预测。一旦推理开始，下一个补丁的尺寸将从几个候选项中选择。在这里，我们将这两个决策步骤整合到一个基于置信度的标准中，我们在经验上发现这是有效的。形式化地说，在第t步，我们假设第k个补丁尺寸的候选项对应于阈值ηt k。给定预测pt，具有maxj ptj > maxk ηt k的样本的推理过程将终止，而具有较小的最大预测值的样本将使用更小的补丁进行处理，以提高准确性。

在（t+1）步骤中，将使用第k个补丁大小，并使用{ηt1+1，ηt2+1，...}重复此过程。所有阈值的值都可以视为超参数，并根据问题（7）在验证集上采用遗传算法[84]确定。因此，可以在计算预算的条件下动态调整输入序列长度和补丁大小。MS-GFNet的设计可以在线切换使用更小或更大的补丁（无需额外的训练），以在有限或足够的计算资源下实现相对更好的性能。在训练期间，我们只是随机选择所有三个阶段的补丁大小。

在MS-GFNet中，两个编码器（fG和fL）和分类器（fC）的架构保持不变。对于前者，大多数深度网络可以自然地处理不同大小的输入，而对于后者，由于全局池化，输入特征向量（e¯t）的维数不会发生变化。值得注意的是，补丁提议网络π接收多个尺寸的特征图（et），这在原始网络中是不可行的。为此，在π中的1x1卷积后，我们将所有特征图上采样到最大可能的大小。此外，为了提供有关上一个和下一个补丁的大小的信息，我们将它们编码为one-hot，并将它们与输入GRU之前的特征串联在一起。

5 实验 本节中，我们在基于图像和基于视频的识别任务上对所提出的GFNet进行了经验评估。代码和预训练模型可在 https://github.com/blackfeather-wang/GFNet-Pytorch 获取。

**5.1 实验**

**5.1.1 数据集**

**图像分类**。 (1) ImageNet是一个来自ILSVRC2012 [1]的1,000类数据集，包含120万张用于训练的图像和5万张用于验证的图像。我们采用与[3]、[6]、[85]相同的数据增强和预处理配置。除非另有说明，图像的分辨率设置为224x224。 (2) 瑞典交通标志数据集[86]，[87]由747张训练图像和684张测试图像组成，分辨率为960x1280。这些样本根据其包含的速度限制标志或无限速标志进行注释。采用与[87]相同的数据增强技术。对于两个数据集，我们在训练集上估计GFNet的置信阈值，因为我们发现它在几乎相同的性能上与交叉验证相比。

**视频识别**。 (1) Something-Something V1&V2 [22]是两个大规模视频识别数据集，分别包括98k和194k个视频。它们都被注释为174个人类动作类别（例如，假装拿起某物）。 (2) Jester [23]是一个包含148,092个视频的大规模动作识别数据集，分为27个类别。我们对所有三个数据集使用官方的训练验证划分，并从每个视频中均匀采样8/12/16帧。我们使用与[25]、[51]相同的数据增强和预处理配置。在推理期间，我们将所有帧调整为256x256大小，并将其裁剪为224x224大小。在训练集上估计置信阈值。

**5.1.2 两种推理设置**

我们考虑两种设置来评估我们的方法： (1) 有预算的批次分类[26]，其中网络需要在给定的计算预算内对一组测试样本进行分类； (2) 任何时候预测[26]，[27]，其中网络可以在任何时间强制输出预测。如[26]所讨论的，这两种设置在许多实际应用中是普遍存在的。对于(1)，我们估计置信阈值以执行介绍中的自适应推理，而对于(2)，我们假设所有测试样本的输入序列长度相同。

**5.1.3主干网络**

对于图像分类任务，GFNet利用MobileNet-V3、RegNet-Y、EfficientNet、ResNet和DenseNet这几个基于CNN的SOTA水平的分类网络进行了实验，这些网络在我们的GFNet中作为两个深层编码器来使用。在有算力资源预算限制的这种实验条件设置下的图像批分类任务中，对于每种分类网络，我们固定输入序列的最大长度T，并改变模型大小(宽度、深度或两者同时)或patch大小(H'， W')，以获得覆盖不同计算预算范围的网络。请注意，我们总是让H'=W'。我们还将GFNet与一些性能不错的基线模型进行了比较，即MnasNets、ShuffleNets-V2、MobileNets-V2、CondenseNets、FBNets、ProxylessNAS、SkipNet、SACT、GoogLeNet和MSDNet

在视频识别方面，我们在ResNet-50上使用时态移位模块TSM实现了GFNet。以用最少的修改为目标，GFNet中的缩放和裁剪操作仅仅在空间维度上执行和图像一样的操作。按照文献[25]的方式，整个视频的所有帧都会入模。我们固定T，并改变H'， W'(H'= W')和从视频中抽帧的数目（8帧12帧或16帧）。几种高效的视频识别模型，如：TRN、ECO和 AdaFuse的性能被视为基线。关于网络配置和训练超参数的更多细节在附录A.3和A.4中给出。

**5.2ImageNet中的图像分类**

**5.2.1理论计算效率**

有算力资源限制的**图像批分类**的结果见图7(a-e)。我们首先将每个GFNet的性能绘制为灰色曲线，然后把每种算力预算条件下在验证集中的最佳准确率绘制为黑色曲线。可以看到，即使是SOTA水平的模型，在相同的计算量下，GFNet也能显著提升它们的效果。例如：在7千万次乘加运算的算力预算限制下，基于MobileNet-V3实现的GFNet在验证集上实现了最高的准确率，约等于71%，比普通的MobileNet-V3提高了大约2个点。基于EfficientNets实现的GFNet相比于普通的EfficientNets，达到相同性能的情况下，GFNet的计算量减少了1.4倍。基于ResNets和DenseNets实现的GFNet达到指定的精度要求所需的乘加次数减少了大约2-3倍。此外，GFNet的计算成本可以精确地调整，从而在给定的预算下实现尽可能好的性能。

将实验条件设置为“在任何时间都可能需要进行预测推理”时：我们将基于DenseNet-121实现的GFNet和另一个自适应推理网络MSDNet进行了比较。为了进行公平的比较，我们按照文献[26]的方式，提供了50000张训练图像。我们的实验结果中还包括了不同深度的densenet的结果。由图可见，当算力预算在5亿次到22亿次乘加时，GFNet比MSDNet的准确率高出了4%到10%。

5.2.2 实际推理速度

**这是在iPhone上面的的实验**。一个合适的平台GFNet的实施可能是移动应用，其中每个图像的平均推理延迟和耗电量的平均推理延迟和功耗与计算量呈近似线性关系(Multiply-Adds)这样就能减少计算开销，来提升用户体验，减少耗电。最后，我们调查了我们的方法在iPhone XS Max上的实际推理速度（with Apple A12 Bionic）使用TFLite。这个批处理大小（batch size 1）为1的单线程模式是用来追随[9]，[13]，[18]。我们首先测量了使用每个输入序列的可能长度测量预测的时间开销，然后使用每一个长度，根据验证样本（validation samples）的数量（number）来获取（take）数加权平均值（weighted average）。结果在图8。可以看到GFNet显著加速了MobileNets-V3和ResNets的推理速度。例如，我们的方法减少了22%必要的延迟，达到75.4%的测试准确度（MobileNets-V3-Large），也就是从16.3ms降到了12.7ms。对于ResNet，要达到相同的性能作为基准，GFNet总体降低了2-3倍（2-3×times）的延迟。

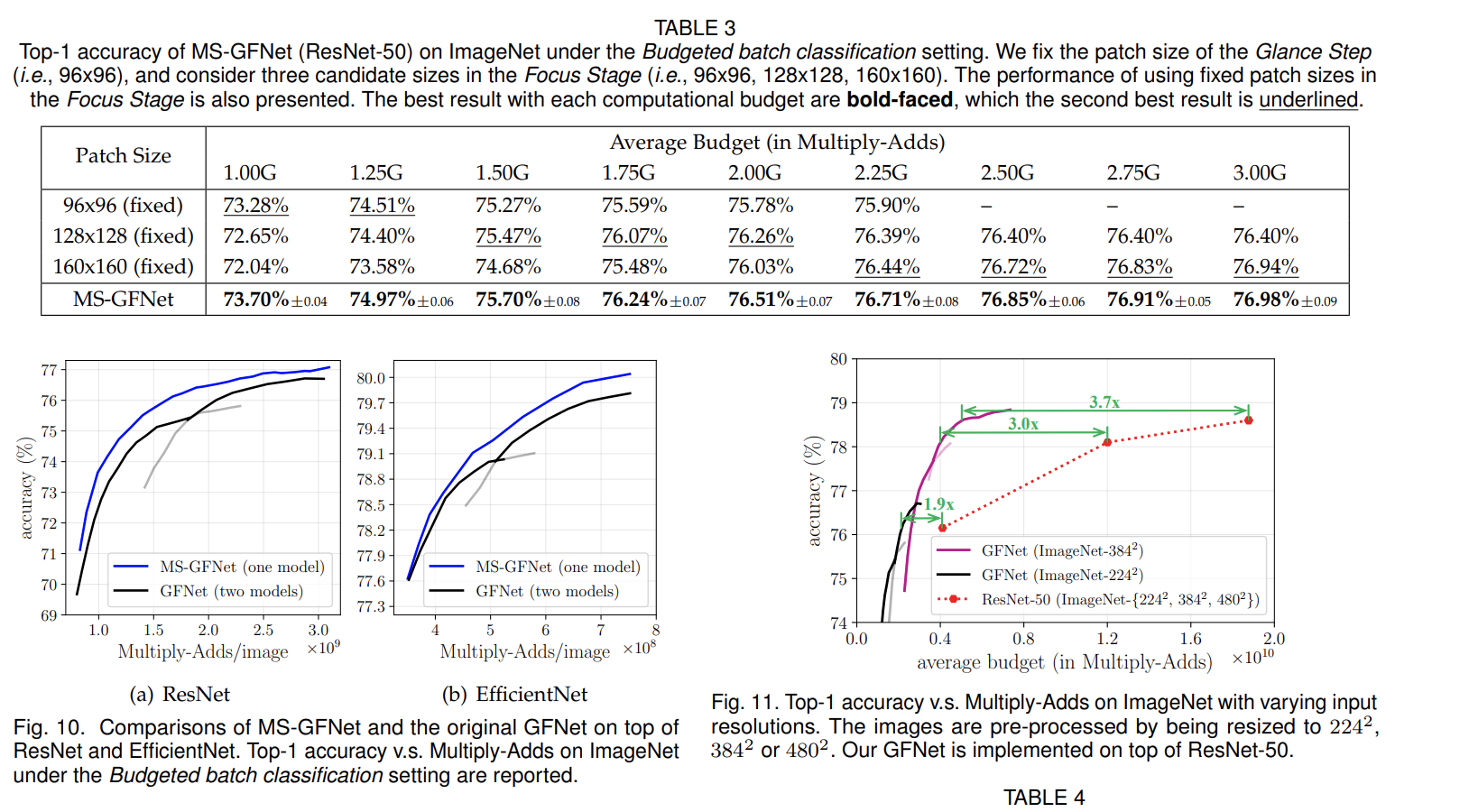
**GPU上的实验。**GFNet也可用于（be leveraged to）在GPU设备上加速深网推理。考虑批量推理环境（settings），这里有大量的测试样本需要被GPU处理，更具体一点，ImageNet验证集里的50,000张图片将被送入（fed into）GFNet，每一批的尺寸是128每张NIVIDIA 2080Ti GPU。每一个小批次（mini-batch），如果推理流程到了出口（exit）（也就是当生成了p1, p2, ...）达到了早结束标准的样本将成为输出，剩下的图像继续被处理（遵循Focus Stage过程），吞吐量记为50,000/T，T是总共处理所有小批次的时间。结果在表1。相同的算力资源下，GFNet提升了1.5-2.1倍的ResNet和DenseNet的吞吐量，同时没有牺牲准确度。

**5.2.3 可视化**

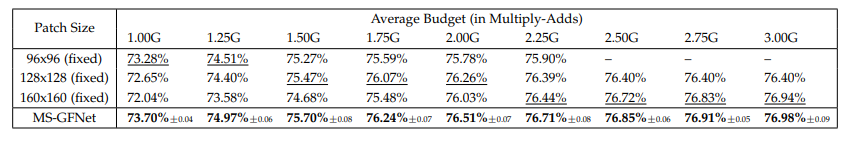
我们在图9展示了被基于GFNet的ResNet-50在一些测试样本上识别出的图块（image patches）。根据需要输入的数量，样本被分割为不同的列来获取正确的分类结果。可以看到GFNet高度自信地在扫视阶段（Glance step）使用典型特征（prototypical features）正确地分类了包含大目标的“简单”图像。对于相对“困难”的，没有明显特征的复杂图像，我们的网络能够聚焦于一些有明显分类特征（class-discriminative）的区域来逐步地（progressively）提升确信度。

**5.2.4 对比奖励的效率**

在表2，我们展示了当4.1中提出的对比性奖励函数应用时，GFNet在ResNet-50，MobileNet-V3-Large和EfficientNet-B2上的性能。



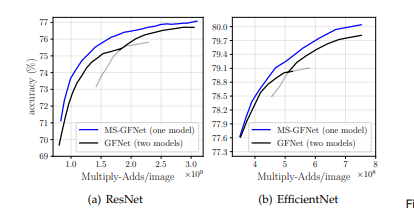
在预算批分类设置下，MS-GFNet (ResNet-50)在ImageNet上的准确率为Top-1。我们固定了Glance Step的补丁大小(即96x96)，并在Focus Stage中 考虑三个候选尺寸(即96x96, 128x128, 160x160)。并给出了在聚焦阶段中使用固定补丁大小的性能。每个计算预算的最佳结果用粗体表示，次优结果用下划线表示。



平均预算（以倍数为单位）

补丁大小

MS-GF网络



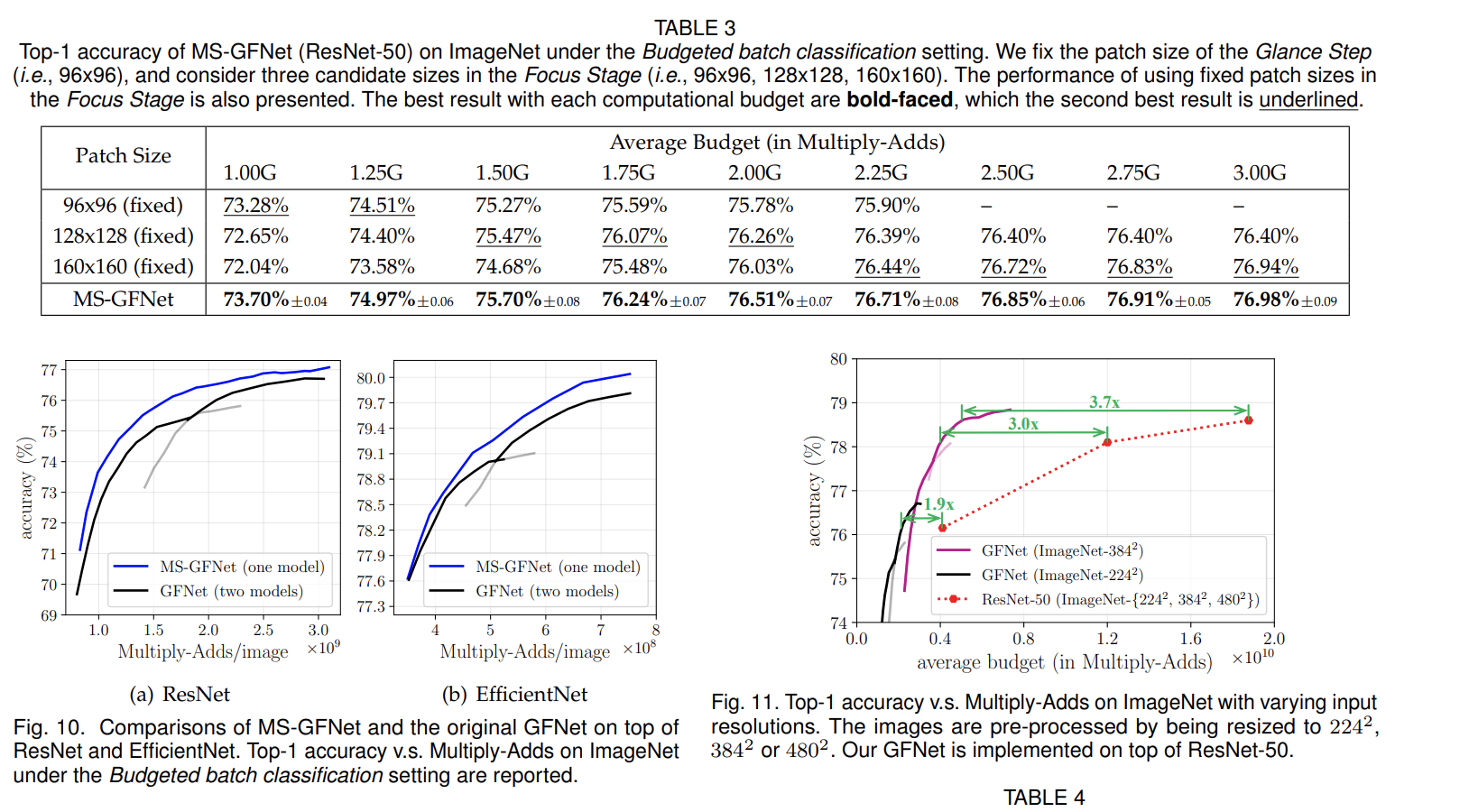
MS-GFNet (一个模型)

GFNet (两个模型)

Fig.10. MS-GFNet与基于ResNet和EfficientNet的原始GFNet的比较。报告了在预算批分类设置下，ImageNet上的Top-1精度与乘法-加法的比较。

**5.2.4 对比奖励的有效性**

在表2中，我们展示了使用4.1节中提出的对比奖励函数 时，在ResNet-50、MobileNet-V3-Large和EfficientNet-B2之 上的GFNets的性能，其中我们用单次蒙特卡罗抽样估计 了Eq. (8)中的期望。原始奖励的结果被称为“增量”。训练 阶段II(强化学习)在同一阶段I检查点的基础上应用不同的奖 励。为了明确对比，这里我们不进行训练阶段III。可以观察 到，提出的对比奖励持续地提高了所有三个模型的准确性， 特别是在焦点阶段的第一步(即t = 2)。



**5.2.5 MS-GFNet**

MS-GFNet的性能见表3。将所提出的模型与在Glance步骤中采用相同补丁大小(即96x96)， 但在Focus阶段使用固定补丁大小(即96x96, 128x128, 160x160)的vanilla GFNets进行比较。表3中的所有模型都使用上述的对比奖励进行训练。可以观察到，当计算预算较小时，较小的补丁优于较大的补丁，而较大的补丁则获得更高的精度。相比之下，MS-GFNet在所有计算预算下都始终优于固定补丁大小基线的最佳结果，这证明了所提出的多尺度补丁机制的有效性。请注意，MS-GFNet与基线具有相同的网络架构，只是π略有变化，并且参数的数量大致相同。

在图10中，我们比较了vanilla GFNet和具有对比奖励的MS-GFNet。后者不仅在相同的计算预算下始终优于前者，而且能够在更大范围的乘法-加法中实现更高的效率。这使得MS-GFNet能够更灵活地调整其计算成本，而无需在现实场景中进行额外的训练。

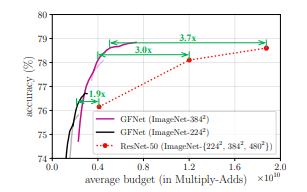
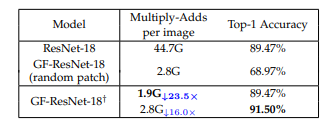


Fig. 11. Top-1精度vs在不同输入分辨率的ImageNet上的乘法添加。图像经过预处理，大小调整为224²,384²或480²。我们的GFNet是在ResNet-50之上实现的。

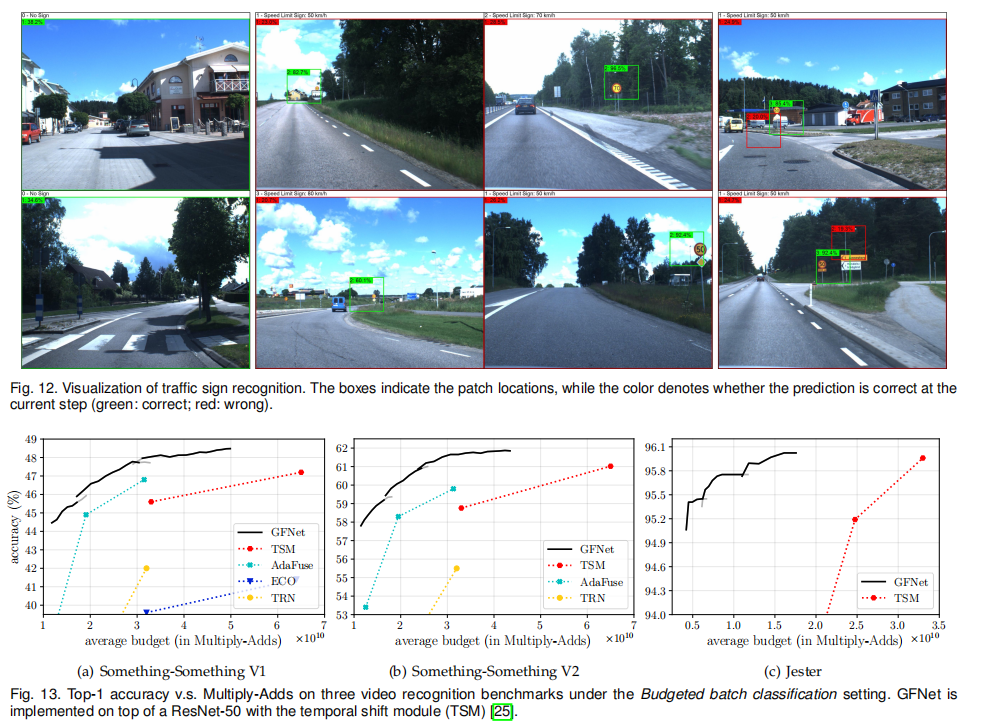
TABLE 4

瑞典交通标志数据集上的交通标志识别结果，该数据集由在真实移动车辆上收集的960x1280张道路场景图像组成。GFNet的补丁大小设置 为H′ =W′ = 192，而“random patch”表示在我们的方法中随机裁剪补丁。最好的结果是黑体字。blue编号基于骨干网(即ResNet-18)。GFNet的计算成本可以在线调整。



**5.3 高分辨率图像识别**

**使用更高的分辨率在ImageNet上的结果**。如前所述，利用高分辨率输入显著提高了现代深度网络的准确性。然而，大图像通常会产生很高的计算成本，它相对于图像的高度(或宽度)呈二次增长，有时甚至更快。我们的方法的主要优点之一是，GFNet可以有效地处理高分辨率 的输入，同时保留其精度收益。为了证明这一点，我们对ImageNet数据集进行预处理，以获得不同分辨率的图像(即224², 384²和480²)，并将GFNet与这些场景中的基线进行比较，如图11所示。对于2242和3842的原始图像大小，GFNet的patch大小设置为96x96/128x128和160x160/192x192。图11的结果表明，随着原始图像分辨率的增加，我们的方法优于骨干网的幅度越来越大。例如，在相同的224²/384²图像上，GF-ResNet-50在不牺牲精度的情况下，将ResNet-50的计算成本降低了1.9×/3.0×。另一个有趣的现象是，在利用相同的输入时，GFNet获得了比骨干更高的最佳准确性。我们初步将其归因于动态计算范式，该范式自适应地将更多的计算分配到任务相关区域，并可能学习更多的判别表示。



另一个有趣的现象是，当改变相同的输入时，GFNet获得比骨干网络获得更高的最佳精度。我们暂时将其归功于动态计算的范式，它自适应地分配更多的计算到与任务相关的区域，并可能学习更多的有区别的表示。

**交通标志识别。**一个标准的假设是，ImageNet数据集要主要包含已经被人类摄影师集中到相关对象为中心的图像。然而，我们的GFNet并不依赖于这一假设，并且适用于更一般的情况，例如，测试图像可以在野外采集，而没有指定的预处理。作为一个典型的例子，我们在表4中展示了瑞典交通标志数据集的结果。该数据集由在真实移动车辆上收集的960x1,280辆道路场景图像组成，其任务是识别限速标志的存在性和类型。请注意，感兴趣的对象通常都很小，分布多样，有时也不清楚（示例见图12）。从表4中可以看出，GFNet显著提高了主干网络的计算效率，即在保持精度的情况下，降低了23.5×的计算成本。图12中给出了几个具有代表性的可视化示例。我们发现GFNet能够在扫视阶段有效地识别交通标志的存在，并可以在关注阶段进一步关注包含标志的局部区域以识别其具体内容。此外，一旦GFNet未能定位感兴趣的交通标志（例如，由于其他分散注意力的标志），它可能会简单地通过额外的关注步骤来纠正这一点。

**5.4视频识别的主要结果**

**主要结果**。在三个具有代表性的基于视频的基准测试上的预算批分类结果如图13所示，其中可以获得与图像识别相似的观察结果。GFNet可以显著提高TSM的计算效率。例如，在V2的一些事情上，GFNet将平均每个视频的预算从6.5×1010减少到2.5×1010的乘法添加。此外，GFNet的性能远远优于最先进的高效视频识别框架AdaFuse[94]（∼1.5−5%）。然而，值得注意的是，GFNet实际上是与AdaFuse相正交的，因为后者主要专注于改进网络体系结构。此外，在视频识别中，我们的方法的计算成本也可以不需要额外的训练进行在线调整。

**可视化**。GFNet在视频基准测试上的可视化结果如图14所示。每个视频都有4个均匀采样的帧，注释的含义如图9所示。可以观察到GFNet能够在瞥步识别一些语义更简单的动作，例如“删除某物”，主要包括识别某个物体的消失。相比之下，对于更困难的样本，我们的模型可以自适应地定位和利用一些与任务相关的视频补丁来理解更复杂的行为（例如，假装做某事）或建模对象之间的关系（例如，两个对象的共现和“相邻”关系）。