使用轻量级卷积神经网络进行文件片段分类

Mustafa Ghaleb ，Kunwa r Saaim, Muhamad Felemban*IEEE* 会员，、Saleh Al-Saleh、Ahmad Al-Mulhem

arXiv:2305.00656v1 [cs.CR] 1 May 2023

摘要—**在数字取证中，文件片段分类是完成文件雕刻过程的重要步骤。目前存在多种无需依赖元数据即可识别文件片段类型的技术，例如利用文件头/文件尾和 N 元语法等特征来识别片段类型。近年来，卷积神经网络（CNN）模型已被用于构建分类模型，以完成这一任务。然而，随着层数增加，卷积神经网络中的参数数量往往呈指数级增长。这导致训练和推理时间大幅增加。本文提出了基于深度可分离卷积神经网络的轻量级文件片段分类模型。评估结果表明，与最先进的基于卷积神经网络的模型相比，我们提出的模型推理速度更快，准确率相当。具体而言，我们的模型在 FFT-75 数据集上达到了 79% 的准确率，参数量接近 100K，浮点运算次数为 164M；与文献中最先进的分类器相比，模型规模缩小至四分之一，速度提升 6 倍。**

索引词—**IEEE、IEEEtran、期刊、LaTeX、论文、模板。**

I. 引言

数字取证是一门通过收集数字证据来调查网络犯罪和网络攻击的科学。数字取证过程包括对受损或遭入侵设备中的文件和数据进行保全、识别和提取。有多种数字取证技术和工具可用于协助这一过程 [38, 31]。攻击者常试图清除所有可能指证自己的证据，例如格式化硬盘 [6]。在此类场景中，基于文件系统元数据的传统恢复方法被认为无效。

为克服这一挑战，数字取证调查人员使用文件雕刻，根据文件内容重建文件 [16, 28, 30]。文件雕刻过程无需使用文件系统中的任何元数据，即可从原始二进制数据块中恢复受损的数据文件。人们采用多种文件雕刻技术，利用文件头/文件尾匹配 [39], 、概率度量 [3], 和 n 元语法分析 [49]等方法，完整或部分地重建文件。

没有元数据时，识别雕刻文件的类型颇具挑战性。对碎片化文件进行文件雕刻时，这一问题更为棘手。硬盘上没有足够的连续空间来写入大型文件时，通常会发生文件碎片化。在碎片化文件雕刻中，可以识别出两项任务：选择一个候选序列，

M. Ghaleb、M. Felemban、S. Al-Saleh 和 A. Al-Mulehm 任职于法赫德国王石油与矿业大学计算机工程系及智能安全系统跨学科研究中心，该校位于沙特阿拉伯宰赫兰。K. Saaim 任职于阿尔伯塔大学计算机科学系

文件块并对片段的类型进行分类 [30]。后一项任务称为文件片段分类。

文献中提出了多种文件片段分类方法 [50, 33, 5, 49, 39, 9, 29, 13, 1, 4, 53,48, 18]。例如，*Sceadan* 工具根据文件内容确定基于批量数据的类型 [5]。近年来，深度学习已被用于识别文件片段的类型。例如，Gray-scale [9]和 FiFTy [33] 工具展示了如何高效地利用卷积神经网络进行文件片段分类。尽管卷积神经网络是分类中最常用的深度学习模型，我们认为，通过修改神经网络的架构，现有模型在性能和准确率方面仍有进一步提升的空间。我们尤其关注轻量级卷积神经网络，以便在资源受限的设备上执行文件片段分类。

卷积神经网络的一个主要缺点是，随着层数增加（即模型深度增加），参数数量会呈指数级增长。参数数量的增加会提高模型的复杂度，进而增加训练和推理时间。另一方面，深度卷积神经网络模型能够提供更高的准确率。因此，通过优化卷积神经网络的结构（即层数、卷积神经网络滤波器大小和通道数），在推理速度和准确率之间找到平衡至关重要。为克服这一挑战，研究人员提出了一种计算成本更低的模型，称为深度可分离卷积 [23, 45, 46],, 该模型大幅减少了参数数量。

在我们之前的论文中， [40],我们提出了一种基于卷积神经网络的轻量级文件片段分类方法，称为深度可分离卷积（DSC）。在本文中，我们进一步提出了两种模型：带有挤压与激励模块的深度可分离卷积（DSC-SE）和改进的深度可分离卷积（M-DSC）。我们表明，与 CNN 相比，所提出的模型减少了参数数量。因此，文件片段分类过程无需专用硬件（例如图形处理器和 TPU）即可执行。为了评估模型的性能，我们将所提出模型的结果与 FiFTy [33] 和基线循环神经网络（RNN） [21]进行了比较。我们使用 FFT-75 数据集 [32],对模型进行了训练，该数据集包含六种场景。场景 1 的结果表明，在 4096 字节片段上，我们的模型使用少 6.3 倍的浮点运算次数（FLOPs）即可达到与 FiFTy 相当的准确率；在 512 字节片段上，所需 FLOPs 少 87 倍。此外，我们的模型以约 100K 个参数和 164M FLOPs 达到了 79.27% 的准确率，而 FiFTy 使用超过 400K 个参数和 1 GFLOPs 达到了 77.04% 的准确率。最后，我们展示了

我们的模型在推理时间（每个块的毫秒数）方面优于 FiFTy：在图形处理器上的速度比 FiFTy 快 15 倍，比基线 RNN 快 105 倍。

本文其余部分的组织结构如下：第2节介绍本研究所需的背景知识。第3节介绍我们提出的模型。第4节讨论实验及评估结果。第5节讨论文件片段分类领域的最新技术。最后，我们在第6节总结研究发现，并提出未来研究的潜在方向。

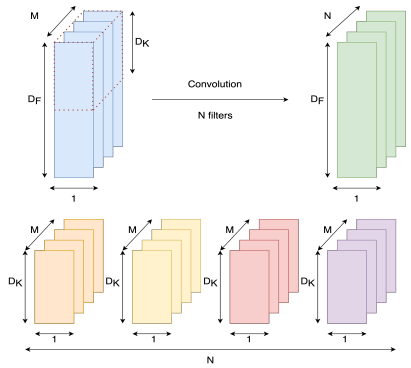
II. 预备知识

A. 文件雕刻

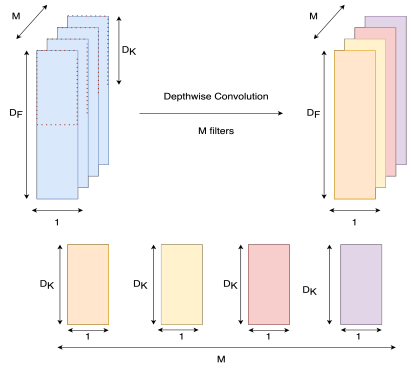
文件雕刻是根据文件内容，而不是使用文件系统的应用程序接口或元数据，从存储介质（例如硬盘）中提取文件的过程 [30]。当底层文件系统损坏或未知时，通常会使用这一过程。当文件被操作系统分割成片段时，文件雕刻可能会变得复杂。因此，文件雕刻过程包括雕刻文件片段、识别每个片段的类型、对片段排序以及组装文件。实际应用中使用了多种文件雕刻工具和技术，包括文件头/文件尾匹配和基于文件结构的雕刻[35, 36]。然而，仍有一些挑战需要解决：1）提高文件雕刻的性能，因为如今数字取证案件的平均大小已达到太字节级别；2）提高文件雕刻的整体准确率。

B. 卷积神经网络

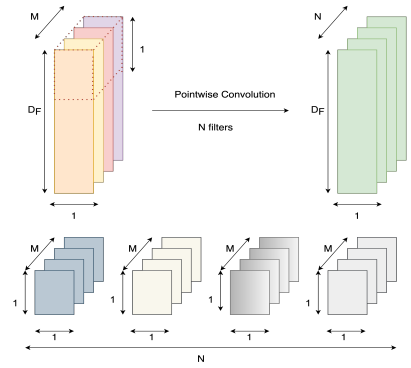
卷积神经网络（CNN）是一种人工神经网络，广泛用于目标检测和图像分类。CNN 包含一个或多个卷积层，可从输入中提取特征，并根据提取的特征预测输出 [14]。卷积滤波器也称为卷积核，是一种用于对图像执行卷积运算的二维矩阵。从数学角度来看，卷积运算是卷积核与层输入之间的点积。可以使用各种卷积核从输入数据点中提取不同的特征。通常，卷积核滤波器的尺寸小于输入，这样卷积核权重便可在输入的各个维度间共享。在文件片段分类中，CNN 的输入是 4096 维或 512 维向量，分别对应 4KB 或 512 字节的片段。随后，输入会传递到嵌入层，由其将输入向量转换为连续的 (4096,32) 或 (512,32) 维向量，具体取决于片段大小。嵌入层是预处理步骤，用于为后续 CNN 层准备输入向量。嵌入层的输出会馈入一个卷积层，该层的卷积核大小不限。可以堆叠多个卷积层 [33]。在构建深度神经网络（DNN）时，堆叠标准卷积层会导致模型包含大量参数。



(a) 标准卷积



(b) 深度卷积



(c) 逐点卷积

图 1：(a) 中的标准卷积分解为两层：(b) 中的深度卷积和 (c) 中的逐点卷积。

C. 深度可分离卷积

可分离卷积模型最初由 Sifre 和 Mallat 提出 [42]。此后，人们基于可分离卷积提出了若干模型，包括 Xception- net [10]，MobileNets [23],EfficientNet [44],和 ShuffleNet[52]。标准卷积的自动特征提取功能通过更小、更快的深度可分离卷积实现。深度可分离卷积是一种因式分解卷积，其中普通卷积核被拆分为两部分：一个深度卷积和一个1×1卷积，称为逐点卷积。将卷积分解为两层可大幅减少参数数量，从而缩短训练和推理所需的计算时间。深度卷积会对每个输入通道应用一个

滤波器。随后，逐点卷积应用 1×1 卷积，将深度卷积的输出合并起来 [23]。图 1a 展示了标准一维卷积运算：它以 $D_{\mathrm{F}} \times 1 \times M$特征图为输入，并生成 $D_{\mathrm{F}} \times 1 \times N$特征图，其中$D_F$ 是输入特征的长度，M 是输入通道数，N 是输出通道数。在卷积网络中，卷积核大小为K时，参数数量为 $D_{\mathrm{K}} \times 1 \times N \times M$。随后，总计算成本为

$$D_{\mathrm{F}} \cdot N \cdot M \cdot D_{\mathrm{K}} \tag{1}$$

与卷积神经网络不同，深度可分离卷积将卷积层分解为两个层：图 1b 所示的深度卷积和图 1c 所示的逐点卷积。深度卷积层为每个输入通道配置一个滤波器，其参数量为 $D_{\mathrm{K}} \times 1 \times M$，卷积核大小为 K，通道数为 M。因此，深度卷积的计算成本为：

$$D_{\mathrm{F}} \cdot M \cdot D_{\mathrm{K}} \tag{2}$$

另一方面，逐点卷积借助 1×1 卷积，对经过滤波的特征图进行线性组合。 1×1 卷积的成本与卷积核大小为 1 的标准卷积相同，即 $D_{\mathrm{F}} \times N \times M$。因此，深度可分离卷积的总计算成本为：

$$D_F \cdot M \cdot D_K + D_F \cdot N \cdot M \tag{3}$$

因此，使用深度可分离卷积时，计算成本的总降幅为：

$$\frac{D_F \cdot M \cdot D_K + D_F \cdot N \cdot M}{D_F \cdot N \cdot M \cdot D_K} \tag{4}$$

$$= \frac{1}{N} + \frac{1}{D_K} \tag{5}$$

N 的取值范围为 32 至 1024，一维卷积的 $\mathrm{D_K}$ 可以介于 9 和 27 之间。因此，计算成本可降低 85% 至 95%。

在这方面，已有多种采用深度可分离卷积进行图像分类的模型。这些模型准确率较高，同时还能实现快速推理。[23, 41, 10]。

III. 用于文件片段分类的轻量级卷积神经网络

本节介绍所提出的用于文件片段分类的轻量级卷积神经网络模型架构。以往的文件雕刻技术大多依赖统计特征提取 [26], 支持向量机 [5], 卷积神经网络 [33],循环神经网络[18], 和全连接神经网络（FCNN） [48]。循环神经网络的速度呈指数级变慢，而全连接神经网络和卷积神经网络需要较深的模型才能取得良好性能。我们的目标是在保持高准确率的同时，缩短文件片段分类的推理时间。为此，人们已针对图像分类任务开发了多种基于深度可分离卷积的模型。这些模型能够在保持高准确率的同时实现快速推理[23, 41, 10]。研究表明，与全连接神经网络、卷积神经网络和循环神经网络相比，深度可分离卷积通常所需的计算时间更短、参数更少 [23]。

图 2 展示了所提出模型的整体架构。所有模型均由一个嵌入层和一个标准 1D 卷积块组成。嵌入层将取值范围为 0 到 255 的文件片段字节转换为一个 32 维稠密连续向量。嵌入层用于将输入特征维度压缩到更小的空间。若不使用嵌入层，每个字节值都将由一个 256 维稀疏向量表示。通过多层标准卷积和深度可分离卷积，模型以非线性方式变换输入，从而自动提取特征。为了自动提取分类所需的关键特征，随后将稠密向量表示输入一个标准 1D 卷积块，以及多个深度可分离 Inception 块或挤压与激励模块。为获得用于分类的最终特征，沿空间维度对最后一个深度可分离卷积块的输出进行平均。在以下各节中，我们将进一步介绍每个模型的细节。

A. 深度可分离卷积（*DSC*）

在 DSC 中，一维卷积层的输出会传递给 Hardswish 激活函数，为模型引入非线性。Hardswish 函数 [22]如下所示。

$$Hardswish(x) = \begin{cases} 0 & if x \leq -3, \\ x & if x \geq +3, \\ \frac{x \cdot (x+3)}{6} & otherwise \end{cases} \tag{6}$$

随后，输出会经过三个 Inception 模块的处理。这些模块由多个并行的卷积层和池化层组成，旨在从输入中提取局部和全局信息。经过 Inception 模块后，输出会传递至平均池化层以缩减空间维度，随后经过最后一个一维卷积层，该层负责缩减特征图并生成最终预测。类别概率由特征向量通过1×1 卷积后，再经过 *softmax* 激活函数 [12]得出。

Inception 模块。图 3 展示了典型 Inception 模块的架构。前一层的输出会传入三个并行的深度可分离卷积以及一个1×1 卷积。在我们的模型中，深度可分离卷积层的卷积核大小分别为 11、19 和 27，步长分别为 1、1、1 和 4。卷积核大小是根据先前最先进的网络 [33]通过经验选定的。每个卷积层后都接一个批量归一化层。所有深度可分离卷积层的输出相加，然后通过大小为 4 的最大池化层进行下采样。 1×1 卷积分支的输出直接与最大池化层的输出相加。如果输入和输出通道数相同，则以直接将最大池化层的输出相加来替代 1×1 卷积层。

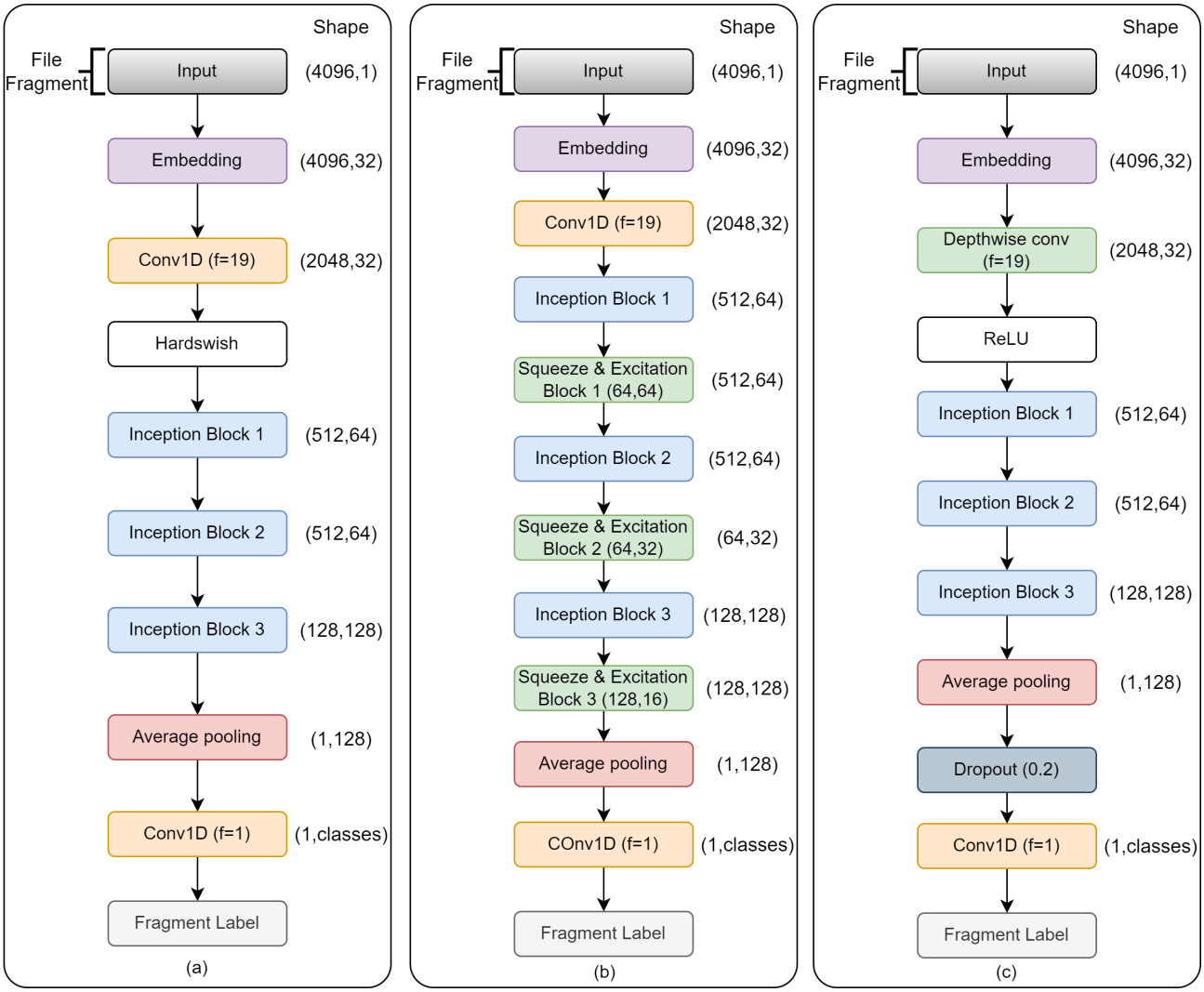


图 2：网络架构。a) DSC，b) DSC-SE，c) M-DSC

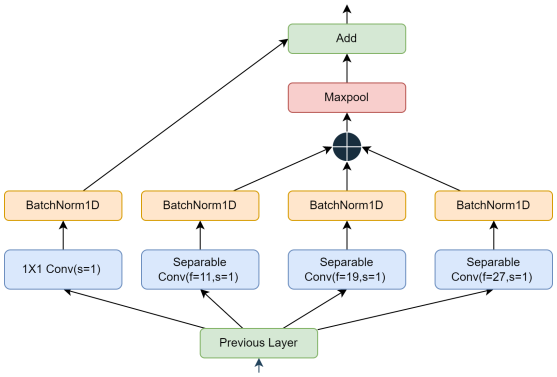


图 3：Inception 模块

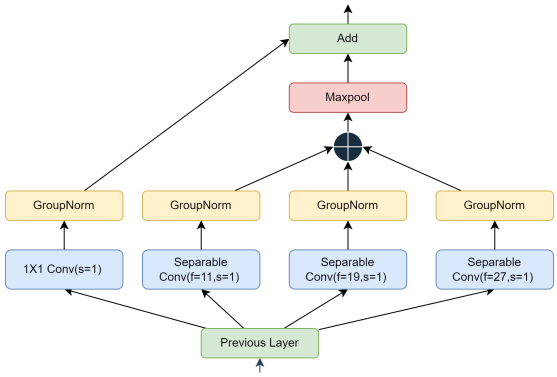


图 4：改进的 Inception 模块

B. 带挤压与激励的深度可分离卷积（*DSC-SE*）

DSC-SE 基于 DSC，并在每个 Inception 模块后添加 Squeeze-and-Excitation (SE) [24] 模块。SE 模块是一种用于提升卷积神经网络性能的注意力机制

性能。挤压与激励模块使网络能够聚焦于特征图中每个通道最重要的特征，这对图像分类、分割和目标检测等任务很有帮助。挤压与激励模块首先将特征图中的空间信息压缩为一个

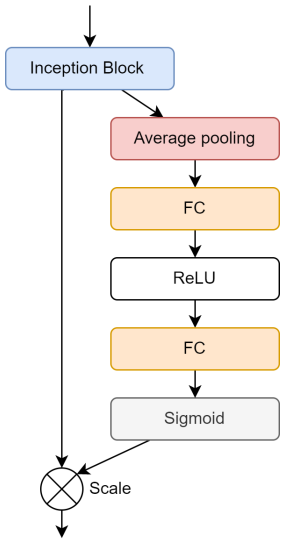


图 5：挤压与激励模块

通道级全局描述符，然后使用全连接层计算一组通道级权重。这些权重随后与特征图逐元素相乘。这样一来，包含任务最重要信息的通道便会获得更高的权重。近期，挤压与激励模块已被集成到 Inception 架构中，从而提升了其在各种任务中的性能 [24]。挤压与激励模块的核心原理是，更加重视那些对最终预测贡献更大的特征。此外，挤压与激励模块计算效率高且易于实现。图 5 详细展示了挤压与激励模块的结构。

C. 改进的深度可分离卷积（*M-DSC*）

M-DSC 架构同样基于 DSC，但进行了若干修改。M-DSC 不使用第一个一维卷积层，而是采用逐通道卷积层，分别对输入的每个通道执行卷积。Hardswish 激活函数被修正线性单元（ReLU）取代。我们使用组归一化代替批量归一化，一次对多个通道的激活进行归一化 [25, 51]。我们提出的模型中采用的改进 Inception 模块如图 4 所示。这减少了批量归一化的内存和计算开销。最后，在平均池化之后、最终一维卷积层之前，添加一个随机失活层，以便在训练过程中随机将部分激活值设为零，从而防止过拟合。

总之，这三种架构在激活函数、归一化和特征重校准的处理方式上有所不同，旨在提升模型的性能和效率。DSC-SE 在 Inception 块中加入挤压与激励模块，以改善特征重校准；M-DSC 则用逐通道卷积层替换一维卷积层，用 ReLU 替换 Hardswish，并引入组归一化和 dropout，以提升性能和效率。

表 I：不同文件类型的分组

分组	文件
归档文件	apk、jar、msi、dmg、7z、bz2、deb、gz、pkg、rar、rpm、xz、zip
音频	aiff、flac、m4a、mp3、ogg、wav、wma
位图	JPEG、TIFF、HEIC、BMP、GIF、PNG
可执行文件	exe, Mach-O , elf, dll
人类可读	Markdown、RTF、TXT、TeX、JSON、HTML、XML、日志文件、csv
办公	DOC、DOCX、KEY、PPT、PPTX、XLS、XLSX
已发布	DJVU、EPUB、MOBI、PDF
Raw	ARW、CR2、DNG、GPR、NEF、NRW、ORF、PEF、RAF、rw2、3fr
向量	AI、EPS、PSD
视频	MOV、MP4、3GP、AVI、MKV、OGV、WebM
杂项	pcap、ttf、dwg、SQLite

IV. 性能评估

本节将介绍我们提出的模型与基于循环神经网络和 FiFTy 的基线模型进行性能评估的结果 [33]。

A. 数据集

为了评估模型的性能，我们使用了 Mittal 等人在 [32]，中提供的数据集，其中每个类别包含数量均衡的文件。其他数据集（例如 [15],）则高度不平衡：20 种文件类型占数据集的 99.3%，其余 0.7% 属于 43 种文件类型。所用数据集由 75 种文件类型组成，并归入 6 个不同的场景。 [33]对这些场景的描述如下：

1)（全部；75 个类别）：所有文件类型均作为独立类别；这是最通用的情形，可进一步归并为更专门的使用场景。2)（用途特定；11 个类别）：根据用途将文件类型归为 11 个类别；这些信息可用于复杂的层次分类，也可用于确定未知设备的主要用途。3)（媒体雕刻器 - 照片和视频；25 个类别）：每种标记为位图（6 种）、RAW 照片（11 种）或视频（7 种）的文件类型均视为独立类别；其余所有类型归为一个其他类别。4)（粗粒度照片雕刻器；5 个类别）：为不同照片类型设置独立类别：JPEG、11 种 RAW 图像、7 种视频，以及 5 种归为同一类别的其余位图；其余所有类型归为一个其他类别。5)（专用 JPEG 雕刻器；2 个类别）：JPEG 为独立类别，其余 74 种文件类型归为一个其他类别；此场景旨在分析来自通用设备的磁盘映像。6)（相机专用 JPEG 雕刻器；2 个类别）：JPEG 为独立类别，其余照片/视频类型（11 种 RAW 图像、3GP、MOV、MKV、TIFF 和 HEIC）归为一个其他类别；此场景旨在分析来自数码相机 SD 卡的内容。

B. 基线模型

我们实现了一种循环神经网络（RNN）的变体，即基于长短期记忆网络（LSTM）的模型，作为基线模型

并根据 [18] 中的工作进行性能比较。结果发现，可学习参数数量较少，限制了模型对 75 种文件进行分类的效果。模型规格如下：将文件片段的字节序列输入一个 32 维嵌入层，随后依次经过双向和单向 LSTM 层。对于 512 字节片段，LSTM 层的两个层各有 128 个神经元；对于 4 KB 片段，其双向层和单向层分别有 128 和 256 个神经元。模型使用 Softmax全连接层生成输出标签。

C. 实验设置

我们的所有实验均在搭载双路 Intel Xeon CPU E5-2620 @ 2.40 GHz（12 个物理核心、24 个逻辑核心）、192 GB 内存和一块 Nvidia Titan X 图形处理器的系统上进行，操作系统为 Ubuntu 20.04。神经网络设计使用了 PyTorch 1.5.0。我们使用 TPE [7] （通过 Optuna [2] 实现）对学习率、优化器选择和激活函数进行自动超参数调优。我们发现，Adam优化器 [27]和 Hardswish [22] 分别是最佳优化器和激活函数。我们没有对卷积核大小进行调优，而是采用了先前研究 [33] 中的卷积核大小，这些尺寸已被证明效果良好。Inception 模块不同分支的卷积核大小分别设为 11、19 和 27。

不同网络的准确率按如下方式计算：

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

其中，TP 为真正例，TN 为真负例，FP 为假正例，FN 为假负例。

为了从更小的模型中获得更高的准确率，我们在相应片段大小的数据集上对网络进行了预训练。我们尝试利用迁移学习在提高准确率方面的性能 [37]。具体而言，为了开发 4096 字节的片段，模型在 512 字节片段数据集上进行了预训练，反之亦然。我们发现，在使用 512 字节片段数据预训练以开发 4096 字节片段模型时，准确率提高了 6-8%。然而，在使用 4096 字节片段数据预训练 512 字节片段模型时，并未获得类似的性能。

D. 结果

总体而言，我们的模型在准确率没有显著损失的情况下，推理时间优于 FiFTy。结果汇总于表 II 中。具体而言，在场景1（75 种文件类型）下，就准确率和推理时间而言，我们的模型均优于 FiFTy 和循环神经网络（基线）。特别是，在使用 4096 字节片段运行 1 GB 数据时，FiFTy 和循环神经网络模型的速度分别比我们提出的模型慢 25 倍和 660 倍。使用 512 字节片段时，FiFTy 慢 8 倍，而循环神经网络慢 87 倍。研究发现，由于循环神经网络能够处理较短序列，循环神经网络模型在使用 512 字节片段时表现良好。相比之下，当片段大小增加到 4096 字节时，由于循环神经网络的梯度消失问题，其性能会下降[20]。虽然循环神经网络模型在 512 字节片段上的准确率更高，但循环神经网络并不实用，因为其推理时间相对较长。

参数数量和推理时间。 表 III 总结了与 FiFTy 相比，所提出模型中的神经网络参数数量。在全部六种场景和两种片段大小下，我们的模型参数都远少于 FiFTy。此外，表 III 比较了六种场景分别在图形处理器和中央处理器上的推理时间。对于全部六种场景，就推理时间而言，我们的模型在图形处理器和中央处理器上的性能都大幅优于 FiFTy。与 FiFTy 在中央处理器上的表现相比，我们观察到，4096 字节文件片段的推理时间减少了 5 倍，512 字节文件片段的推理时间减少了 9 倍。在图形处理器上，两种片段类型的推理时间均减少了 9 倍以上。

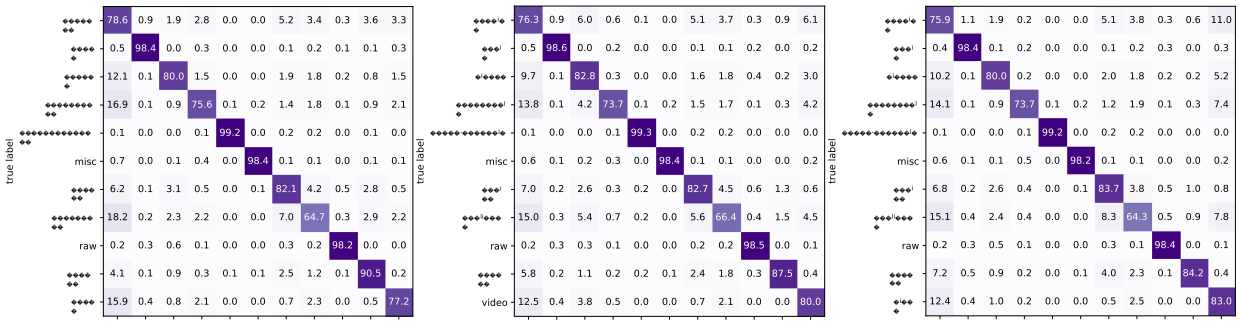
浮点运算次数。为提供硬件无关指标，我们计算了各模型和 FiFTy 的浮点运算次数。表 IV 展示了比较结果。与 FiFTy 相比，对于 4096 字节片段，我们的模型所需浮点运算次数减少了 6.3 倍；对于 512 字节片段，则减少了 87 倍。例如，在场景 1 中，对于 4096 字节的片段，DSC、DSC-SE 和 M-DSC 分别为 164.88、164.96 和 89.07 百万次浮点运算，而 FiFTy 为 1047.59 百万次浮点运算。这些结果表明，我们提出的模型能够有效减少预测所需的计算量，同时仍保持较高的准确率。因此，在计算资源有限的实际应用中，我们的模型更具适用性。

准确率。 场景1中按使用场景分组的文件类型混淆矩阵见图 6（分组信息列于表 I）。由于归档文件中嵌入了其他文件类型，大多数误分类都发生在归档文件组中。所提出模型和 FiFTy 的准确率列于表 V。总体而言，我们的模型准确率与 FiFTy 相当。不过，我们的模型推理时间更短。

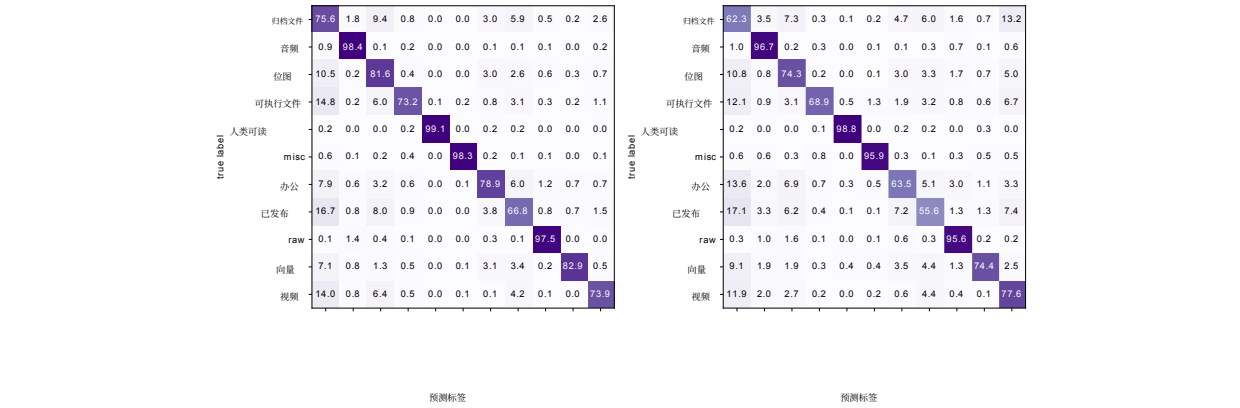
E. 讨论

总之，结果表明，这三种模型的推理时间相近，且不同运行之间的差异极小。这与 FiFTy [33],中报告的结果形成对比：FiFTy 的推理时间可能显著更长。我们的模型采用前沿技术进行了优化，以实现高效推理、最大限度地降低计算开销并减少延迟。因此，我们的推理时间能够与最先进模型相当，同时仍保持较高的准确率。这证明了所提出的模型在真实应用场景中的有效性；在这些场景中，快速、高效的推理至关重要。

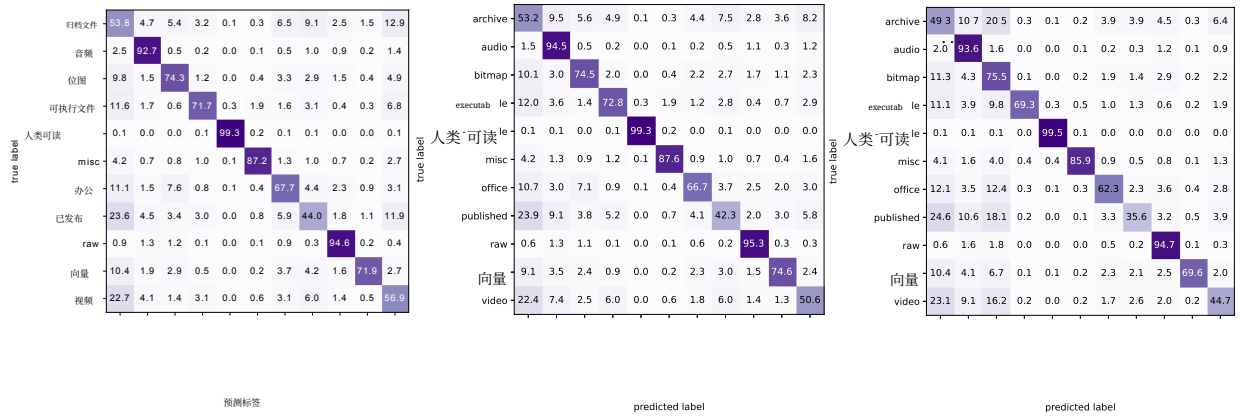
与先前的工作 [33, 9], 类似，我们观察到，高熵文件难以分类，因为卷积核无法从中提取任何统计痕迹。此外，许多文件属于容器类型，其中包含作为嵌入对象的其他文件，例如，*PDF* 文件可以包含嵌入的*JPEG* 图像。因此，分类器的表现会变得不稳定。



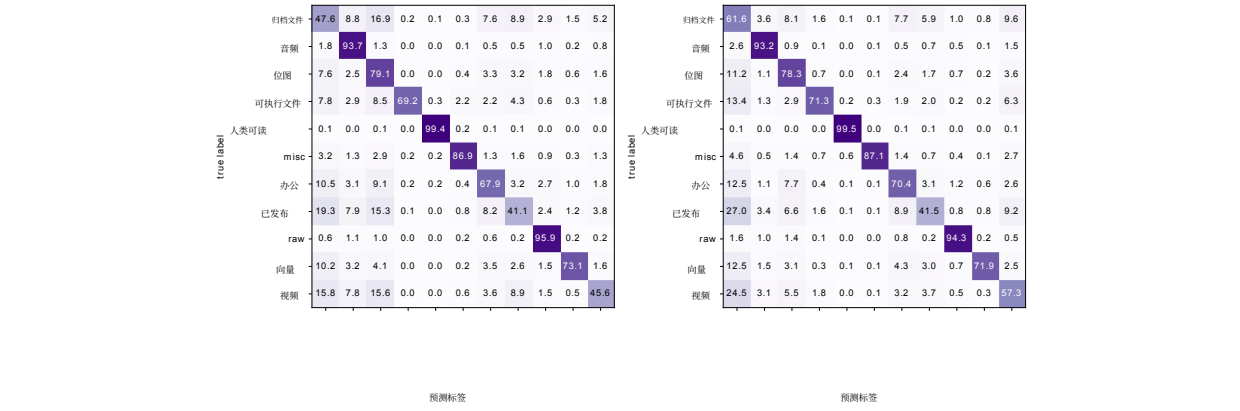
(a) DSC (78.45%) (b) DSC-SE (79.27%) (c) M-DSC（78.68%）



(d) FiFTy（77.04%） (e) 基线（70.51%）



(f) DSC（65.89%） (g) DSC-SE（66.33%） (h) M-DSC（64.04%）



(i) FiFTy（65.66%） (j) 基线（67.5%）

图6：我们的模型、FiFTy[33] 和基线循环神经网络模型在（a）–（e）的4 KB片段以及（f）–（j）的512字节片段上的混淆矩阵。由于类别数量较多（75个），属于同一超类的类别被聚类为一类。

表 II：五种模型在全部 75 种文件类型上的结果比较

模型	神经网络	块大小	参数数量	准确率	速度[毫秒/块]†	速度 [分钟/GB]†
DSC	深度可分离	4096	103,083	78.45	**2.567**	**0.055**
	CNN	512	103,083	65.89	3.531	0.382
DSC-SE	深度可分离	4096	105,515	**79.27**	3.908	0.064
	带 SE 的卷积神经网络	512	105,515	66.33	3.473	0.471
M-DSC	改进的深度可分离	4096	**85,291**	78.68	2.606	0.057
	卷积神经网络	512	**85,291**	64.03	**2.776**	**0.378**
FiFTy	一维卷积神	4096	449,867	77.04	38.189	1.366
	经网络	512	289,995	65.66	38.67	3.052
基线 RNN	LSTM	4096	717,643	70.51	268.58	36.375
		512	379,851	**67.5**	126.54	33.431

† 在 Nvidia Titan X 上计算

表 III：C　从推理时间和模型参数方面比较 FiFTy 与我们的模型

场景	片段	#param				推理时间（中央处理器）[毫秒/块]				推理时间（GPU）[毫秒/块]			
	Size	DSC	DSC-SE	M-DSC	FiFTy	DSC	DSC-SE	M-DSC	FiFTy	DSC	DSC-SE	M-DSC	FiFTy
1	4096	103,083	105,515	**85,291**	449,867	18.595	23.291	**17.338**	121.476	**2.567**	3.908	2.606	38.189
	512	103,083	105,515	**85,291**	289,995	6.958	8.216	**6.279**	75.551	3.531	3.473	**2.776**	38.673
2	4096	94,827	97,259	**77,035**	597,259	22.587	18.023	**17.214**	92.324	2.763	3.300	**2.610**	29.826
	512	94,827	97,259	**77,035**	269,323	7.827	6.657	**6.346**	89.344	3.239	3.504	**2.910**	29.809
3	4096	96,633	99,065	**78,841**	453,529	**17.068**	18.860	17.361	102.808	3.284	3.182	**2.632**	37.286
	512	96,633	99,065	**78,841**	690,073	**6.066**	6.613	6.300	99.710	3.078	3.499	**2.708**	35.361
4	4096	94,053	96,485	**76,261**	684,485	**17.149**	20.477	17.537	100.117	**2.524**	3.197	2.608	40.176
	512	94,053	96,485	**76,261**	474,885	**6.029**	6.533	6.285	79.346	**2.620**	3.258	2.677	35.965
5	4096	93,666	96,098	**75,847**	138,386	22.427	**16.362**	17.474	99.855	**2.524**	3.362	2.769	39.262
	512	93,666	96,098	**75,847**	336,770	**5.967**	6.327	6.275	79.831	**2.581**	3.259	2.873	42.413
6	4096	93,666	96,098	**75,847**	666,242	18.418	23.000	**17.279**	98.536	**2.530**	3.389	2.641	34.931
	512	93,666	96,098	**75,847**	242,114	**6.133**	8.373	6.398	89.104	**2.612**	3.274	2.702	38.982

表 IV：FiFTy 与我们的模型在浮点运算次数（百万浮点运算）方面的比较

场景	片段	浮点运算次数			
	size	DSC	DSC-SE	M-DSC	FiFTy
1	4096	164.88	164.96	**89.07**	1047.59
	512	20.63	20.64	**11.15**	1801.71
2	4096	164.86	164.95	**89.05**	1327.90
	512	20.61	20.63	**11.13**	918.06
3	4096	164.86	164.95	**89.05**	647.78
	512	20.61	20.63	**11.14**	3579.57
4	4096	164.86	164.95	**89.05**	2378.51
	512	20.61	20.63	**11.13**	1576.71
5	4096	164.86	164.95	**89.05**	488.37
	512	20.61	20.63	**11.13**	2330.48
6	4096	164.86	164.95	**89.05**	1126.00
	512	20.61	20.63	**11.13**	611.30

表 V：FiFTy 与我们的模型在准确率方面的比较

场景	#of 文件	片段 Size	准确率 DSC	DSC-SE	M-DSC	FiFTy
1	75	4096	78.45	**79.27**	78.68	77.04
		512	65.89	**66.33**	64.04	65.66
2	11	4096	85.7	87.10	85.33	**89.91**
		512	75.84	74.99	72.18	**78.97**
3	25	4096	93.06	93.32	91.87	**94.64**
		512	80.79	80.79	78.53	**87.97**
4	5	4096	94.17	**94.61**	92.66	94.03
		512	87.14	87.32	83.41	**90.30**
5	2	4096	99.28	**99.37**	99.24	99.12
		512	98.94	98.96	98.94	**99.07**
6	2	4096	99.59	**99.69**	99.62	99.59
		512	98.76	98.65	98.87	**99.23**

最后，格式不同但文件类型相似的文件，例如*PPT*、*PPTX*和*KEY*，会被相互误分类。然而，*DOC*和*DOCX* 不受此影响，因为*DOCX* 使用*XML* ，而*DOC* 以二进制格式存储。

V. 相关工作

目前已有多种用于识别和分类文件片段的技术。这些技术涵盖了从利用魔数和文件头的基础方法[11] 到使用机器学习和深度学习的更先进方法 [33, 5]。一般而言，文件雕刻方法大致可分为三类：1）统计方法，2）机器学习方法，以及 3）深度学习方法。**统计方法**。Karresand 等人提出了*Oscar* ，用于确定二进制数据片段可能的文件类型[26]。*Oscar* 根据不同文件类型中大小为 4096 字节的片段的字节频率分布（BFD），创建向量模型。计算待分类片段的均值和标准差与各质心之间的向量距离。如果距离低于阈值，该片段就会被分类为建模文件。*Sceadan*是一款知名的开源文件雕刻工具，它使用多种统计特征，包括一元组、二元组、熵、平均字节值和最长连续段等 [5]。该工具根据这些统计特征生成十个独立的输入向量，并将其划分为四组：一元组、二元组、除 n 元语法以外的所有全局特征，以及一组全局特征子集。这些特征组作为分类模型的输入。Ahmed 等人 [1] 提出了一种根据文件内容识别文件类型的方法。为缩短识别所需的计算时间，该方法采用了两种技术。首先，选取基于出现频率的部分特征

。其次，对大小为 100 字节的文件块进行采样。研究发现，该分类器在低熵文件片段（例如纯文本文件、未压缩图像等）上表现良好，但在高熵文件片段（例如压缩文件、二进制可执行文件等）上效果不佳。

机器学习方法在 [29], Li 等人提出了一种支持向量机（SVM）模型，该模型使用基于数据字节直方图的特征向量进行训练。SVM 用于二分类，其中一个类别固定为 JPEG，另一个类别则有所不同，包括动态链接库、可执行文件、PDF 和 MP3。类似地，Fitzgerald 等人 [13]提出了一种用于片段分类的 SVM 模型，将文件片段视为字节集合，并用特征向量表示，该向量包含一元组和二元组计数，以及熵等统计度量。Zheng 等人[53]提出了一种基于 SVM 的方法，使用 BFD 或直方图和熵作为特征向量。Amirani 等人 [4]提出了一种将主成分分析（PCA）与多层感知机（MLP）相结合的层次方法，用于特征提取。提取出的特征被输入分类器。PCA 从原始特征的 BFD 中选取 N_1 个特征。这 N_1 个特征用于训练一个自联想神经网络，该网络提取 N_2 个特征（$N_2<N_1$）。Bhatt 等人 [8]提出了一种用于文件片段分类的层次分类方法，采用 SVM 作为基础分类器。研究发现，该研究所用的特征不足以处理复杂文件类型，因为这些特征无法描述文件片段与其所属父类之间的成员关系。作者建议细化复杂文件的层次分支，并设计能够更好地捕捉文件片段与其所属父文件类型之间成员关系的特征。Haque 和 Tozal [17] 提出了 Byte2Vec，这是一种新颖的特征生成模型，将 word2vec 概念扩展到将文件片段映射为稠密向量表示。Byte2Vec 使用 Skip-gram 模型生成向量表示，并基于这些表示训练 k 近邻（kNN）分类器，以识别片段类型。Byte2Vec 语料库模型适用于各种块大小和文件类型。作者使用公开可用的数据集，将 Byte2Vec 与 kNN 算法（Byte2Vec+kNN）结合，用于特征提取和分类。

深度学习方法Wang 等人 [50] 提出了一种具有多个卷积核大小的单层卷积。原始字节值被转换为对应的二进制表示，并输入嵌入层，将二进制值转换为连续的稠密向量。作者研究了 GovDocs1 数据集中的 20 种文件类型。Chen 等人 [9]采用了不同的方法，将 4096 字节的文件片段转换为 64x63 灰度图像。他们的直觉是，不同文件中的数据片段会具有不同的纹理特征，并体现在灰度图像中。灰度图像被输入类似 VGG [43] 的深度卷积神经网络，其中包含许多卷积层和最大池化层，随后进行稠密分类。Hiester [18] 比较了前馈、循环和卷积神经网络，输入片段采用二进制表示（片段比特）。研究的文件类型包括 JPEG、GIF、XML 和 CSV。对于这些容易区分的文件类型，只有循环网络能够给出令人满意的结果。他们强调，无损表示有助于实现高准确率，但这种二进制表示会增加输入的维度。

Mittal 等人 [33] 提出了*FiFTy*，这是一种用于文件片段类型识别的开源卷积神经网络。他们还开发了一个开源数据集，据称这是目前用于文件片段分类的最大开源数据集。他们开发了一种紧凑型神经网络，采用可训练嵌入空间和卷积神经网络。与基于神经网络的分类器[33, 50, 9]相比，我们的模型在更多文件类型上取得了更高的准确率。我们仅将模型与深度学习方法进行比较，因为这些方法可以利用图形处理器加速。与基于循环神经网络的分类器 [18] 相比，我们的模型速度快 660 倍。基于机器学习的分类器 [5], 需要 9 分钟/GB，而我们的模型仅需 0.05 分钟/GB。

VI. 结论

本文提出了基于深度可分离卷积神经网络的轻量级文件片段分类模型。我们的目标是在不牺牲准确率的前提下，开发推理时间优于当前最佳模型的方案。具体而言，我们设计的分类模型参数量约为 10 万。我们使用涵盖不同场景的 FFT-75 数据集对模型进行了评估。评估结果表明，在所有场景下，所提出的模型均比当前最先进的文件片段分类模型运行更快，同时准确率没有显著下降。对于误分类率较高的类别，还可以通过多种方式进一步提高分类准确率。在不受硬件限制的情况下，神经架构搜索 [54, 34]是为特定数据设计架构的最佳方法。此外，对于将 JPEG 文件与其他文件类型分类等简单场景，可以通过网络内蒸馏 [19, 47]移除神经网络中的冗余连接。

VII. 致谢

作者谨向法赫德国王石油与矿业大学智能安全系统跨学科研究中心致谢。

参考文献

[1]Ahmed, I., Lhee, K.S., Shin, H.J., Hong, M.P.：快速的基于内容的文件类型识别。《数字取证进展 VII》，国际信息处理联合会信息与通信技术进展，第 65–75 页（2011）。
https://doi.org/10.1007978-3-642-24212-0 5[2] Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.：Optuna：新一代超参数优化框架。载于：第 25 届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集（2019）[3] Alghafli, K., Yeun, C.Y., Damiani, E.：测量恢复的视频片段之间相邻概率的技术：vidcarve 方法。《IEEE 可持续计算汇刊》（2019）

[4] Amirani, M.C., Toorani, M., Mihandoost, S.：基于特征的文件片段类型识别。《Security and Communication Networks》**6**，115–128（2013）[5] Beebe, N.L., Maddox, L.A., Liu, L., Sun, M.：Sceadan：使用串联的 N 元语法向量改进文件和数据类型分类。《IEEE Transactions on Information Forensics and Security》**8**（9），1519–1530（2013）[6] Bennett, D.：刑事调查中从移动设备获取信息所面临的计算机取证调查员挑战。《Information Security Journal: A Global Perspective》**21**（3），159–168（2012）[7] Bergstra, J.S., Bardenet, R., Bengio, Y., K´egl, B.：超参数优化算法。载于：Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q.（编）《神经网络信息处理系统进展 24》，第 2546–2554 页。Curran Associates, Inc.（2011）[8]Bhatt, M., Mishra, A., Kabir, M.W.U., Blake-Gatto, S., Rajendra, R., Hoque, M.T., Ahmed, I.：基于层级的文件片段分类。《Machine Learning and Knowledge Extraction》**2**（3），216–232（2020）[9] Chen, Q., Hui, L.C., Liu, D., Zhang, E., Liao, Q., Jiang, Z.L., Fang, J., Yiu, S., Xi, G., Li, R.e.a.：数字取证中使用灰度图像转换和深度学习进行文件片段分类。2018 IEEE Security and Privacy Workshops（SPW）（2018）。https://doi.org/10.1109/spw.2018.00029[10]Chollet, F.：Xception：采用深度可分离卷积的深度学习。第 1800–1807 页（07 2017）。https://doi.org/10.1109/CVPR.2017.195[11] Darwin, I.F.：Libmagic（2008），ftp://ftp.astron.com/pub/file[12] Dunne, R.A., Campbell, N.A.：关于 softmax 激活函数与交叉熵惩罚函数的配对，以及 softmax 激活函数的推导。载于：第八届澳大利亚神经网络会议论文集，墨尔本。第 181 卷，第 185 页。Citeseer（1997）[13]Fitzgerald, S., Mathews, G., Morris, C., Zhulyn, O.：使用 NLP 技术进行文件片段分类。《数字调查》**9**　（2012）。https://doi.org/10.1016/j.diin.2012.05.008[14] Fukushima, K.：一种不受位置偏移影响的模式识别机制自组织神经网络模型。《Biol. Cybern.》**36**，193–202（1980）[15]Garfinkel, S., Farrell, P., Roussev, V., Dinolt, G.：通过标准化取证语料库将科学方法引入数字取证。《数字调查》**6**　（2009）。https://doi.org/10.1016/j.diin.2009.06.016[16] Garfinkel, S.L.：使用快速对象验证雕刻连续和碎片化文件。《数字调查》**4**，2–12（2007）[17] Haque, M.E., Tozal, M.E.：用于文件片段分类的字节嵌入。《FutureGeneration Computer Systems》**127**，448–461（2022）[18]Hiester, L.：使用无损表示的神经网络进行文件片段分类（2018）[19]Hinton, G., Vinyals, O., Dean, J.：在神经网络中提炼知识。arXiv 预印本 arXiv:1503.02531 (2015)[20]Hochreiter, S.：学习循环神经网络时的梯度消失问题及其解决方案。《国际不确定性、模糊性与知识系统》**6**(02), 107–116 (1998)[21] Hochreiter, S., Schmidhuber, J.：长短期记忆。《神经计算》**9**(8), 1735–1780 (1997)[22] Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., 等：MobileNetV3 搜索。载于：IEEE国际计算机视觉会议论文集，第1314–1324页 (2019)[23] Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.：MobileNets：面向移动视觉应用的高效卷积神经网络 (2017年4月)[24] Hu, J., Shen, L., Sun, G.：挤压与激励网络。载于：IEEE计算机视觉与模式识别会议论文集，第7132–7141页 (2018)[25] Ioffe, S., Szegedy, C.：批量归一化：通过减少内部协变量偏移加速深度网络训练。载于：国际机器学习会议论文集，第448–456页。pmlr (2015)[26] Karresand, M., Shahmehri, N.：Oscar——磁盘簇和 RAM 页中的二进制数据文件类型识别。载于：SEC (2006)[27] Kingma, D.P., Ba, J.：Adam：一种随机优化方法。arXiv 预印本 arXiv:1412.6980 (2014)[28] Lei, Z.：未分配空间的取证分析。博士论文，UOIT (2011)[29] Li, Q., Ong, A., Suganthan, P., Thing, V.：一种用于高熵数据片段分类的新型支持向量机方法 (2010年1月)[30] Lin, X.：文件雕刻。载于：《计算机取证导论》，第211–233页。施普林格 (2018)[31]Marziale, L., Richard III, G.G., Roussev, V.：大规模线程化：利用图形处理器提升数字取证工具的性能。《数字调查》**4**, 73–81 (2007)[32]Memon, G.M.P.K.N.：文件片段类型 (FFT)——75 数据集 (2019)。https://doi.org/10.21227/kfxw-8084, http: // dx.doi.org/10.21227/kfxw-8084[33] Mittal, G., Korus, P., Memon, N.：FiFTy：使用卷积神经网络进行大规模文件片段类型识别。《IEEE信息取证与安全汇刊》**16**, 28–41 (2020)[34] Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.：通过参数共享实现高效的神经架构搜索。arXiv 预印本 arXiv:1802.03268 (2018)[35] Poisel, R., Tjoa, S.：文件雕刻综合文献综述。载于：2013年国际可用性、可靠性与安全会议论文集，第475–484页。IEEE (2013)[36] Povar, D., Bhadran, V.：取证数据雕刻。载于：国际数字取证与网络犯罪会议论文集，第137–148页。施普林格 (2010)[37] Pratt, L.Y.：神经网络之间基于可分辨性的迁移——

神经网络。载于：神经信息处理系统进展，第204–211页 (1993)[38] Rafique, M., Khan, M.：探索静态与实时数字取证：方法、实践与工具。《国际科学与工程研究杂志》**4**(10), 1048–1056 (2013)[39] Richard III, G.G., Roussev, V.：Scalpel：一种精简、高性能的文件雕刻器。载于：DFRWS。Citeseer (2005)[40] Saaim, K.M., Felemban, M., Alsaleh, S., Almulhem, A.：使用深度可分离卷积进行轻量级文件片段分类。载于：信息通信技术系统安全与隐私保护：第37届国际信息处理联合会 TC 11 SEC 2022 会议，哥本哈根，丹麦，2022年6月13–15日，论文集，第196–211页。施普林格 (2022)[41] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.：MobileNetV2：倒置残差与线性瓶颈。载于：IEEE计算机视觉与模式识别会议论文集 (CVPR) (2018年6月)[42] SIfre, L., Mallat, S.：用于纹理分类的刚性运动散射 (2014年3月)[43]Simonyan, K., Zisserman, A.：用于大规模图像识别的超深卷积网络。CoRR**abs/1409.1556** (2014), http://arxiv.org/abs/1409.1556[44] Sze, V., Chen, Y.H., Yang, T.J., Emer, J.S.：深度神经网络的高效处理：教程与综述。《IEEE论文集》**105**(12), 2295–2329 (2017)[45]Szegedy, C., Liu, W., Jia, Y., Ser manet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.：通过卷积更深入。载于：计算机视觉与模式识别 (CVPR) (2015), http://arxiv.org/ abs/1409.4842[46] Tan, M., Le, Q.V.：EfficientNet：重新思考卷积神经网络的模型缩放。arXiv 预印本 arXiv:1905.11946 (2019)[47] Tian, Y., Krishnan, D., Isola, P.：对比表示蒸馏。arXiv 预印本 arXiv:1910.10699 (2019)[48] Vulinovi´c, K., Ivkovi´c, L., Petrovi´c, J., Skra˘ci´c, K., Pale, P.：用于文件片段分类的神经网络。载于：2019年第42届信息与通信技术、电子与微电子国际大会 (MIPRO) 论文集，第1194–1198页。IEEE (2019)[49] Wang, F., Quach, T.T., Wheeler, J., Aimone, J.B., James, C.D.：用于文件片段分类的 n 元语法特征提取与训练的稀疏编码。《IEEE信息取证与安全汇刊》**13**(10), 2553–2562 (2018)[50] Wang, Y., Su, Z., Song, D.：使用卷积神经网络进行文件片段类型识别。2018年国际机器学习技术会议论文集——ICMLT 18 (2018)。https://doi.org/10.1145/3231884.3231889[51] Wu, Y., He, K.：组归一化。载于：欧洲计算机视觉会议 (ECCV) 论文集，第3–19页 (2018)[52] Zhang, X., Zhou, X., Lin, M., Sun, J.：ShuffleNet：一种面向移动设备的极高效卷积神经网络。载于：IEEE计算机视觉会议论文集

计算机视觉与模式识别。第6848–6856页（2018）[53]Zheng, N., Wang, J., Wu, T., Xu, M.: 一种取决于数据类型的片段分类方法。2015IEEE计算机与信息技术国际会议；普适计算与通信；可信、自主与安全计算；普适智能与计算，第1948–1953页（2015）[54] Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: 学习用于可扩展图像识别的可迁移架构。载于：IEEE计算机视觉与模式识别会议论文集。第8697–8710页（2018）