

基于动态掩码率和知识蒸馏的图像篡改定位算法

黎思力¹, 廖桂樱¹, 谭舜泉², 黄继武², 李斌¹

¹ 深圳市媒体信息内容安全重点实验室 深圳 中国 518060

² 深圳北理莫斯科大学工程系, 智能感知与计算广东省重点实验室 深圳 中国 518116

摘要 尽管基于深度学习的图像篡改定位模型的性能不断提高, 但是网络大小和参数数量的增加影响了其推理速度和实际部署。知识蒸馏和模型量化是压缩模型的有效方法。其中, MGD(Mask Generation Distillation)是一种使用掩码特征的生成式特征蒸馏算法。然而, 在使用该算法对图像篡改定位经典模型 DenseFCN 进行量化时, 不同的掩码率会导致量化模型的精度差异较大。此外, 在蒸馏训练中, 维持固定的掩码率难以提升模型性能, 尤其是在训练后期。因此, 本文在 MGD 算法的基础上提出了一种使用动态掩码率的方法——DMGD(Dynamic MGD)。通过增加一个动态的模块, DMGD 算法将掩码率可训练化。此模块将学生特征和教师特征作为输入, 在蒸馏过程中根据不同阶段给出适当的逐实例和逐通道的掩码率。实验结果表明, 在 Defacto、PS-boundary 等数据集上, 相较于 MGD 和未使用蒸馏方式, DMGD 能够使 DenseFCN 低比特量化模型在 F1 等指标上达到最优。且在跨域数据集的实验结果也显示 DMGD 量化模型的 F1 等指标也更优秀。此外, 消融实验结果表明, DMGD 相较于 MGD 的时间代价增加也非常有限。

关键词 深度学习; 图像篡改定位; 模型量化; 知识蒸馏

中图分类号 TP391 DOI号 10.19363/J.cnki.cn10-1380/tn.2025.11.08

Image Tampering Localization Algorithm Based on Dynamic Masking Rate and Knowledge Distillation

LI Sili¹, LIAO Guiying¹, TAN Shunquan², HUANG Jiwu², LI Bin¹

¹ Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen Key Laboratory of Media Security, Shenzhen 518060, China

² Guangdong Laboratory of Machine Perception and Intelligent Computing, Faculty of Engineering, Shenzhen MSU-BIT University, Shenzhen 518116, China

Abstract In recent years, the demand for accurate and efficient image tampering detection systems has surged, researchers have intensified their efforts to enhance the performance of such systems while also addressing the challenges posed by their deployment in real-world scenarios. One of the primary concerns has been the computational overhead associated with deep learning-based models, particularly as they become increasingly complex to achieve higher levels of accuracy. This trade-off between model complexity and inference speed has prompted the exploration of model compression techniques, with knowledge distillation and model quantization emerging as promising solutions. The Mask Generation Distillation (MGD) algorithm, which harnesses the power of masked features to distill knowledge from a teacher model to a student model, has shown great potential in compressing image tampering detection models. However, the effectiveness of MGD can be hindered by the need to maintain a fixed mask rate throughout the distillation process. This rigidity limits the adaptability of the model and can lead to suboptimal performance, especially in scenarios where the optimal mask rate varies across different stages of training. To address this limitation, the proposed Dynamic MGD (DMGD) approach introduces a dynamic mechanism for adjusting the mask rate during the distillation process. By allowing the mask rate to be trainable and adaptive, DMGD enables the student model to learn from the teacher model more effectively, leading to improved performance in terms of accuracy and efficiency. The dynamic nature of DMGD not only enhances the model's ability to capture relevant features but also mitigates the risk of overfitting by adapting to the evolving characteristics of the training data. Experimental evaluations conducted on diverse datasets, including Defacto and PS-boundary, validate the efficacy of DMGD in improving the performance of image tampering detection models. Comparative analyses with traditional MGD-based approaches demonstrate the superiority of DMGD, particularly in scenarios where the optimal mask rate varies significantly. Furthermore, the negligible increase in time cost associated with DMGD highlights its practical viability and efficiency in real-world deployment scenarios.

Key words deep learning; image tampering localization; model quantization; knowledge distillation

通信作者: 黄继武, 教授, Email: jwhuang@smbu.edu.cn.

本课题得到国家自然科学基金项目(No. 62272314, No. U23B2022, No. U22B2047), 深圳市基础研究项目(No. JCYJ20250604181211016, No. SYSPG20241211174032004)资助。

收稿日期: 2023-12-12; 修改日期: 2024-06-11; 定稿日期: 2025-10-17

1 引言

近年来, 图像编辑算法和应用软件层出不穷, 导致不法分子能够轻松利用这些软件篡改内容。因此, 图像篡改检测和取证技术显得尤为重要。

大多数基于深度学习的图像篡改取证方法采用通用的端到端检测方法。许多篡改定位模型使用卷积神经网络(Convolutional Neural Networks, CNN)^[1]进行设计, 例如 RRUNet^[2]、Mantra-Net^[3]、SPAN^[4]、RGB-N^[5]、MVSSNet^[6]、PSCC-Net^[7]等。全卷积网络(Fully Convolutional Network, FCN)^[8]近年来在语义分割领域被广泛应用, 部分图像篡改定位研究也借鉴了 FCN 的设计思路。Salloum 等人^[9]参考并改进了 FCN, 将其应用于拼接图像篡改定位领域, 提出了双路多任务全卷积网络(MFCN)。其中一路用于预测拼接区域的内部, 另一路用于预测拼接区域的篡改轮廓。两路分支在训练过程中被联合优化。Zhuang 等人^[10]提出的 DenseFCN 是一种针对图像篡改定位非常有效的端到端网络。其网络结构整体借鉴了 FCN 的设计, 并引入了密连接块(Dense Block), 以捕捉细微的篡改特征。该网络在图像篡改定位领域具有较高的地位, 近年来被许多图像篡改定位算法模型借鉴并用于对比算法, 例如 SAT-IFL^[11]、ViT-VAE^[12]和 SNIS^[13]等。

基于卷积神经网络设计的模型变得越来越复杂, 以提高检测精度, 网络的深度和广度不断扩大。然而, 对于计算能力有限、显存容量较小的设备(如手机和嵌入式设备), 使用复杂网络模型部署后, 推理速度可能非常缓慢。同样, 图像篡改定位模型也面临这一问题。随着这类模型在工业界应用越来越广泛, 研究如何压缩图像篡改定位模型变得尤为重要。目前, 主流模型压缩技术主要包括模型量化和知识蒸馏。

模型量化是一种用于加速底层矩阵运算的方法, 其通过降低数据存储比特数和改变数据类型来实现。量化感知训练 (Quantization Aware Training, QAT) 是模型量化中常用的一种方式, 它通过使用伪量化算子模拟量化误差来重新量化参数^[14]。Courbariaux 等人^[15]提出了二值网络, 将模型权重数值量化为-1或 1, 从而显著减少权重占用空间。Esser 等人^[16]提出了学习步长量化(Learned Step Size Quantization, LSQ)方法, 将量化范围设定为可训练参数。Zhou 等人^[17]则提出了 DoReFa 量化方式, 提出梯度的低比特量化, 该方法可在可编程逻辑门阵列(Field Programmable Gate Array)芯片上实现低功耗的量化训练。

Xu 等人^[18]探讨了将元量化器集成到主网络计算图中的方法, 允许可微操作以适应不同的优化策略, 如 SGD 和 Adam。这种方法旨在改进量化梯度, 提升量化感知训练的效率和效果。

知识蒸馏是将复杂教师网络所学的特征表示传递给参数较少或参数空间较小的学生网络的过程。Hinton 等人^[19]最早提出了知识蒸馏, 这种方法将教师网络的预测输出转化为软标签, 并将其与学生网络的预测输出计算交叉熵作为目标损失函数的一部分, 以指导学生网络的学习。Zhang 等人^[20]和 Fukuda 等人^[21]借鉴集成学习的思想, 引入多个教师模型来指导学生网络。Romero 等人^[22]则进一步提出了特征蒸馏, 将中间层特征作为蒸馏损失函数的目标。而 Zhao 等人^[23]将目光放在输出层的蒸馏上, 将知识蒸馏损失拆分成了两部分的和, 分别是目标类的二分类蒸馏和非目标类的多分类蒸馏。Yang 等人^[24]提出掩码生成蒸馏(Mask Generation Distillation, MGD), 认为通过使用部分特征值和生成模块来恢复教师网络的全部特征, 学生网络无须直接模仿教师即可提高学生特征表现力。这是因为用于蒸馏的特征通常都是网络深层的高阶语义, 特征值通常已经在不同程度上包含了相邻特征值的信息。

将蒸馏应用于模型量化可以有效地提高量化模型的精度, 因为模型量化会导致数据截断, 从而降低模型的特征表示能力。使用蒸馏有助于恢复量化模型因数据截断而损失的特征。掩码生成蒸馏(MGD)通过一个与学生特征尺寸相同的 Mask 矩阵来随机掩码学生特征的特征值, 掩码率是调整 Mask 中零值数量的概率值。然而, 在将 MGD 应用于 DenseFCN 的量化感知训练时, 存在以下缺点: 首先, 不同的掩码率对量化训练的模型精度影响较大。根据第 4 节的实验设置, 表 1 展示了设定不同掩码率时, MGD 蒸馏

表 1 MGD 蒸馏使用不同掩码率对 DenseFCN 在 PS-boundary 数据集下进行 4 比特 LSQ 量化结果
Table 1 Results of performing 4-bit LSQ quantification on DenseFCN using varying mask rates for MGD distillation under PS-boundary dataset

τ	PS-boundary		
	F1	IOU	AUC
0.4	0.673	0.582	0.932
0.5	0.724	0.613	0.963
0.6	0.716	0.619	0.958
0.7	0.703	0.585	0.964
0.8	0.717	0.605	0.970

算法在 DenseFCN 进行 LSQ 4 比特量化感知训练的表现。根据文献[24]中的范围, 实验对掩码率以 0.1 的间隔设置了 5 组实验, 使用 PS-boundary 数据集。可以观察到, 在量化感知训练下, MGD 蒸馏算法表现并不稳定, 不同的掩码率 τ 取值对蒸馏得到的模型精度影响较大, F1 的最高值和最低值差距达到了 5.1%。其次, 固定掩码率下, MGD 蒸馏在训练后期存在瓶颈, 即训练已经适应了固定掩码率带来的提升。

本文针对 MGD 在量化训练中的缺陷, 提出动态掩码生成蒸馏(Dynamic MGD, DMGD)算法, 并应用于 DenseFCN 的量化。MGD 在设置掩码率时采用的是经验性的固定值, 而 DMGD 则构建了一个动态模块, 融合学生特征和教师特征的信息, 能够在训练的整个过程中生成动态掩码率, 并且逐通道逐实例地生成掩码特征。解决了 MGD 中掩码率调参困难的问题, 并且在蒸馏训练后期可以突破固定掩码率的瓶颈。主要的贡献包括。

(1) 针对 MGD 特征蒸馏算法的不足, 提出了基于动态掩码率的 DMGD 算法。实验结果表明, DMGD 算法的精度优于不同固定掩码率的 MGD 蒸馏算法, 同时减少了大量的调参时间并提高了精度。

(2) 提出对图像篡改定位经典模型 DenseFCN^[10]

进行量化的方法。我们使用 DMGD 蒸馏算法提高低比特量化模型的精度, 从而减少模型存储占用和运行显存占用。实验结果表明, 在多个数据集中, 与原量化模型相比, 使用 DMGD 蒸馏算法的 F1 分数均取得了较大幅度的提升。

2 相关工作

2.1 图像篡改定位模型 DenseFCN

本论文使用 Zhuang 等人^[10]提出的 DenseFCN 作为量化的基础网络。DenseFCN 是针对图像篡改定位非常有效的一个端到端的网络, 如图 1 所示, 该模型采用编码器-解码器结构, 整体结构借鉴了 FCN^[8], 并采用 DenseNet^[14]中的密集连接结构来设计了密集连接块(Dense Block)。密集连接块的作用是捕捉细微的篡改特征。在密集连接中, 通过较短的连接可以让各个层从损失函数中得到额外的监督效果, 为定位任务提供更强的监督支持。此外, 密集连接还能有效地重复利用特征, 因为密集连接块的后一层卷积能够接收前面不同层学习到的特征, 从而有效促进各层学习到篡改痕迹特征。我们认为 DenseFCN 是一个有效的网络架构, 适合用于解决图像篡改定位问题。

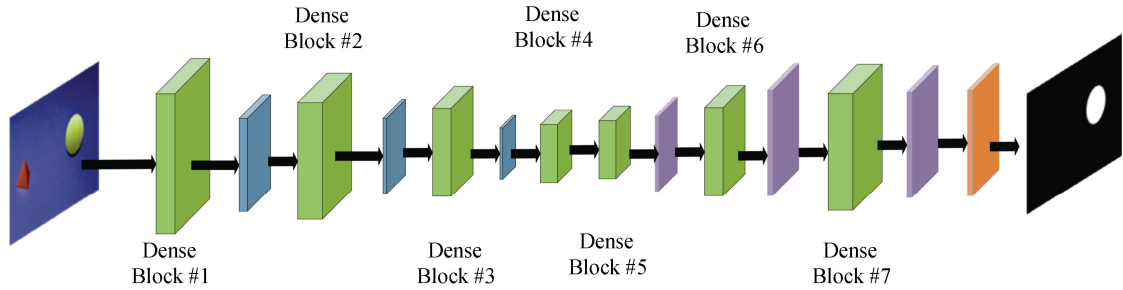


图 1 DenseFCN 网络结构

Figure 1 DenseFCN network structure

2.2 LSQ 量化感知训练

本文目的是将蒸馏算法应用于 DenseFCN 的量化训练, 量化算法采用了基于量化感知训练的 LSQ^[16]算法, 其与普通的量化感知训练一样, 需要用到尺度因子(scale)和零点值(zero_point)两个量化参数。通常, 这两个参数都是由待量化的数据如权重或者特征值进行数值统计后得到的。而 LSQ 算法则将这两个量化参数当作可训练参数来获得, 其核心思想就是将量化参数可训练化。在反向传播阶段, 尺度因子和零点值对量化训练比较敏感, 因此其学习率应比模型自身的权重更小, 否则会导致量化训练不收敛的情况。为解决这个问题, LSQ 引入了一个系数因子 g , 如下所示。

$$g = \frac{1}{n * q_{\max}} \quad (1)$$

其中, n 代表该量化层中元素的个数。 $q_{\max}$ 是量化范围的最大值。在更新尺度因子和零点值时, 使用 g 进行限制。LSQ 中, 零点值初始化通常为 0, 而尺度因子遵循如下公式:

$$s = \frac{2|\bar{v}|}{\sqrt{q_{\max}}} \quad (2)$$

其中, $|\bar{v}|$ 表示该量化层绝对值的平均值。

2.3 MGD 蒸馏算法

Yang 等人^[24]提出的 MGD 方法旨在利用学生特征部分特征值恢复至完整的特征图, 以获取更好的

蒸馏表现。其认为, 在模型中用于蒸馏的深层特征在不同程度上具有相邻特征值的信息。该方法首先对学生网络的特征执行随机掩码(Random Mask)操作, 通过一个掩码矩阵来实现, 该矩阵的尺寸与学生特征一致, 并使用 0-1 均匀分布初始, 然后进行以下公式正则化:

$$M_{i,j} = \begin{cases} 0, & \text{if } M_{i,j} < \tau \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

其中, τ 为掩码率, $M_{i,j}$ 为矩阵使用均匀分布初始化后的值, 公式(3)令每次迭代使用的掩码位置都是随机的, 但总体的掩码比例由掩码率决定。将掩码矩阵与学生特征相乘即是掩码操作。接着通过一个生成模块生成最终用于蒸馏的学生网络特征。

3 DMGD 特征蒸馏算法

3.1 DMGD 特征蒸馏算法

DMGD 对 MGD 算法中的掩码率生成方式进行

了改进。在 MGD 特征蒸馏中, 学生特征需要经过 Random Mask 矩阵运算随机遮掩部分特征值, 经过生成模块恢复之后与教师特征蒸馏。生成模块由两个 3×3 的卷积和 ReLU 激活组成。Mask 矩阵由 0 或 1 组成, 并且其位置是随机的, 掩码率控制 0 的数量。不同的掩码率会导致蒸馏结果不同, 但其值是一个经验值且不固定。

DMGD 在 MGD 的基础上增加了一个动态模块(见 3.2 节), 该模块能够自适应学习学生特征和教师特征之间的关系, 并给出当前训练情况的掩码率参考值。DMGD 的整体框架如图 2 所示, 应用于 DenseFCN 网络的 LSQ 量化感知训练。虽然学生和教师网络的结构相同, 但由于网络参数空间不同, 因此无须任何尺寸转换模块即可对齐学生特征和教师特征。学生和教师的特征通过动态模块生成相应的掩码率, 然后使用该掩码率产生的 Mask 多维矩阵与学生特征进行运算, 生成掩码特征并输入生成模块生成特征。最后, 使用生成特征进行蒸馏。

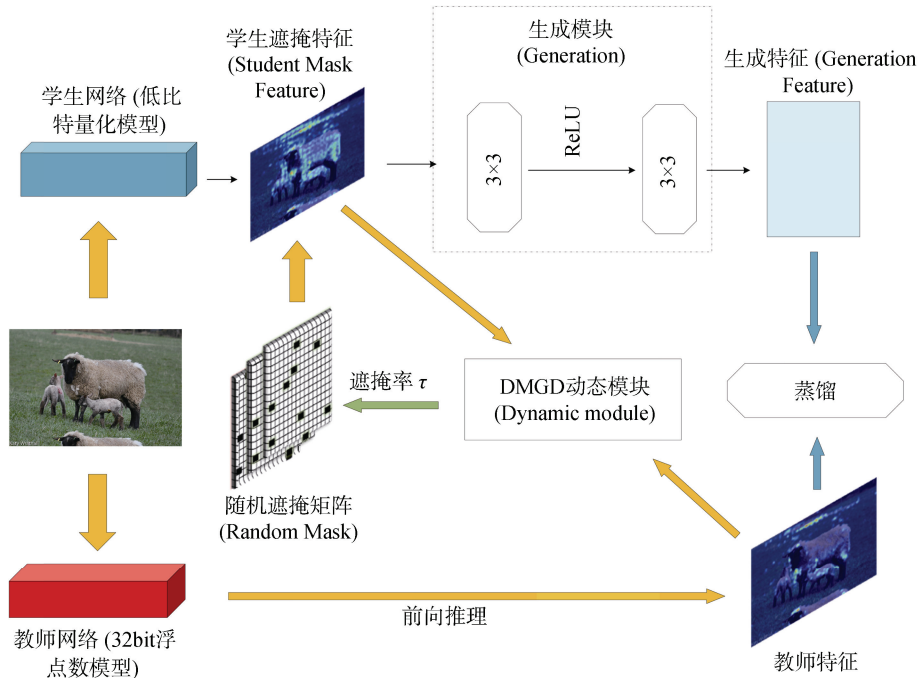


图 2 DMGD 蒸馏算法网络结构

Figure 2 Network structure of DMGD distillation algorithm

3.2 动态模块

DMGD 算法中的动态模块如图 3 所示, 该模块通过 Cat 拼接操作融合了学生特征和教师特征, 以综合两者的信息; 接着, 经过全局平均池化层(Global Average Pool, GAP), 即图 3 中的 GAP 模块, 将融合后特征的空间信息集中于通道上。最后, 通过一个 1×1 的卷积和 Sigmoid 激活函数将输出值限制在 0-1

范围内。图 3 中将特征进入该模块时的尺寸变化都标识出来, 该模块最终可以得到一个尺寸为 $N \times C \times 1 \times 1$ 的掩码率 τ 。改进的 DMGD 动态模块不仅可以获取动态掩码率, 还可以根据训练数据的每个实例和特征的每个通道生成不同的掩码率 τ , 即实现逐实例(Per-Instance)和逐通道(Per-Channel)的掩码率生成方式, 从而在每轮训练迭代时获得更好的训练

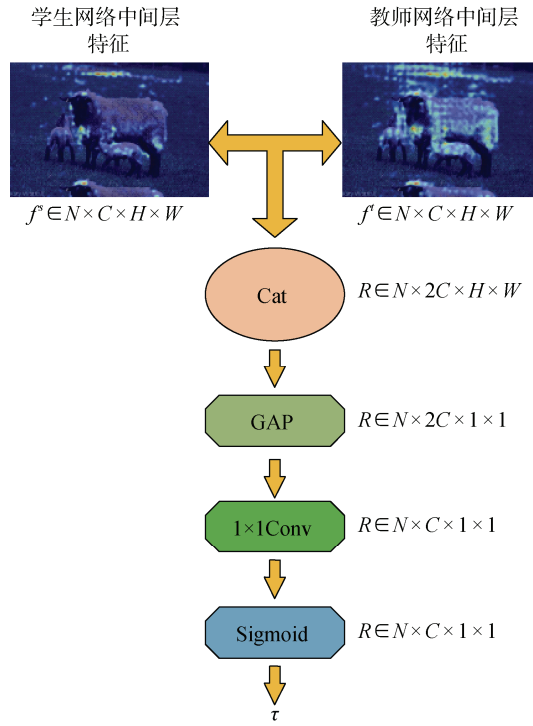


图3 DMGD 动态模块网络结构

效果。随着训练的进行, 动态模块可以捕捉学生特征和教师特征之间的关系, 并根据训练的情况调整生成的掩码率。简言之, 为了改进 MGD 特征蒸馏网络, 我们引入了一个动态模块, 以提供动态遮掩率。该模块融合了学生网络和教师网络中间层的特征, 并输出全局动态掩码率, 克服了 MGD 蒸馏中固定遮掩率的局限性。该模块与整个网络协同训练, 通过网络损失函数的反向传播更新其参数, 监督信号结合了网络中间层的特征与遮掩率, 对 MGD 蒸馏效果产生影响。相较于使用数据集标签作为监督信号相同的 MGD 蒸馏, DMGD 的训练过程更平滑, 更容易突破训练瓶颈。

在训练的初始阶段, 由于 DMGD 的动态模块并不能快速捕捉学生特征和教师特征之间的关系, 因此需要限制动态模块产生的掩码率的范围, 以防止蒸馏算法在初始阶段被误导得到随机的结果。由于初始阶段模块的权重是随机初始化的, 并且还未经过足够的训练, 因此产生的掩码率缺乏足够的有效信息。为解决这一问题, 借助余弦退火^[25]的思想, 在不同的迭代轮次(epoch)中限制动态模块产生的掩码率。具体而言, 如公式(4)所示:

$$\tau = \tau_{\text{init}} + \lambda \text{range}(D(f_s, f_t), \min, \max) \quad (4)$$

$$\text{range}(x, \min, \max) = \begin{cases} \min, & x < \min \\ x, & \min \leq x \leq \max \\ \max, & x > \max \end{cases} \quad (5)$$

其中, $D(\cdot)$ 是动态模块, init 是初值, $\text{range}(\cdot)$ 是范围函数, 其定义如公式(5)所示, 其中参数 $\min$ 和 $\max$ 的作用是限制动态模块在训练初期由于信息不足而产生的遮掩率。公式(4)是动态遮掩率 τ 的整体定义, 其中不同训练阶段的掩码率 λ 受到正弦函数的调节, 具体表达式如公式(6)所示:

$$\lambda = \sin\left(\frac{n}{2 \cdot t} \pi\right) \quad (6)$$

其中, t 表示训练 epoch 的次数, 随着训练时 epoch 的增加, 动态模块产生的 τ 给模型训练的指导将会越来越重要, 而范围限制确保了整个模块中产生的 τ 不会有太大的偏差, 防止蒸馏算法在训练初始阶段被误导到无法收敛的情况。

3.3 掩码特征与生成特征

以下分析均不考虑批量(batch)维度。我们定义 $S^l \in R^{C \times H \times W}$ 和 $T^l \in R^{C \times H \times W}$ 分别是学生和教师网络并使用它们的第 1 层的特征进行蒸馏, M^l 是该学生特征相应的 Mask 矩阵, 其尺寸与学生特征一致, $R_{c,i,j}^l$ 为 Mask 矩阵的每个值, 其通过 0~1 均匀分布生成数据, 然后再根据公式(7)所定义的规则来进行正则化。

$$M_{c,i,j}^l = \begin{cases} 0, & \text{if } R_{c,i,j}^l < \tau_c \\ 1, & \text{otherwise} \end{cases} \quad (7)$$

如 3.2 节所述, 其中 c 是通道数, τ_c 为动态模块产生的逐通道的掩码率。经过正则化后的 Mask 矩阵与学生特征 S^l 相乘, 生成掩码特征, 如公式(8)所示:

$$G(S^l \cdot M^l) \rightarrow G^l \quad (8)$$

再经过生成模块生成最终蒸馏的特征, 如公式(9)所示:

$$G(F) = W_{l2}(\text{ReLU}(W_{l1}(F))) \quad (9)$$

生成模块由两个卷积 W_{l1} 、 W_{l2} 和 ReLU 激活函数组成。

3.4 损失函数

本文使用二分类交叉熵损失函数(BCELoss)和以及 Dice 损失(DiceLoss)^[26]组成的联合损失函数来训练蒸馏网络, 如公式(12)所示。训练教师网络和学生网络时均采用该损失。BCELoss 和 DiceLoss 如公式(10)和公式(11)所示。

$$L_{\text{bce}} = \frac{-1}{wh} \sum_{i=1}^{wh} y_i \log(x_i) + (1 - y_i) \log(1 - x_i) \quad (10)$$

$$L_{\text{dice}} = 1 - \frac{2 \cdot \sum_{i=1}^{w \cdot h} x_i y_i}{\sum_{i=1}^{w \cdot h} x_i^2 + \sum_{i=1}^{w \cdot h} y_i^2} \quad (11)$$

$$L_{\text{total}} = \lambda L_{\text{bce}} + (1 - \lambda) L_{\text{dice}} \quad (12)$$

上述公式中的 w 和 h 表示图像的宽高, L_{total} 表示总损失, 其中 λ 是调整 DiceLoss 比重的系数, 本实验设置中为 0.2。使用蒸馏算法时需要加入蒸馏损失, 如公式(13)所示。蒸馏损失使用 MSE 均方误差计算经过生成模块后的学生特征与教师特征的距离。加入蒸馏后的总损失如公式(14)所示:

$$L_{\text{kd}}(S, T) = \sum_{l=1}^L \sum_{k=1}^C \sum_{i=1}^H \sum_{j=1}^W (T_{k,i,j}^l - G(S_{k,i,j}^l \cdot M_{i,j}^l))^2 \quad (13)$$

$$L_{\text{total}} = \lambda L_{\text{bce}} + (1 - \lambda) L_{\text{dice}} + \partial L_{\text{kd}} \quad (14)$$

其中, ∂ 是控制蒸馏特征权重的系数, 实验中设置为 0.00002。实验使用浮点模型作为教师模型, 学生即为低比特量化模型。MGD 和 DMGD 均选择了 DenseFCN 中的 DenseBlcok5 产生的中间特征进行蒸馏, 因它是模型骨干网络的最后一个模块, 其输出包含丰富的特征信息。

4 实验结果与分析

为验证 DMGD 蒸馏算法在 DenseFCN 的 LSQ 量化感知训练中的有效性, 本文在不同数据集进行低比特量化实验。在对比实验中, 本文首先与未使用蒸馏方式以及采用 MGD 蒸馏方式的 DenseFCN 进行对比; 其次, 通过分析实验结果, 生成中间特征热图进行进一步比较。此外, 本文还测试了量化模型在跨数据集上的泛化性。

4.1 实验数据集

本文使用如下数据集进行实验。

Columbia^[27]: 该数据集总共有 363 张分辨率从 757×568 到 1152×768 不等的图片, 其中 183 张真实图片和 180 张拼接篡改图片。

NIST2016^[28]: 该数据集包含 560 原始图片和 564 张经过拼接、复制移动方式篡改的图像, 尺寸范围在 500×500 和 5616×3744 之间。

DEFACTO^[29]: 是在 MS COCO 数据集^[30]基础上构造的总数超过 22 万张图片的篡改图像数据集, 尺寸范围在 240×320 到 640×640 之间。实验使用的是原始和篡改各 98777 张的版本。

CASIA^[31]: 包括 CASIA v1 和 CASIA v2。v1 版本包含 800 张原始图片和 921 张篡改图片, 尺寸是 384×256; v2 版本包含 7200 张原始图片和 5123 张篡改图片, 尺寸范围在 240×160 到 900×600 之间。

PS-boundary^[10]和 **PS-arbitrary**^[10]: 通过 Photoshop 软件制作的私有数据集。两个数据集各包含 1000 张原图和 1000 张经过 PS 处理的篡改图片, 尺寸范围在

2340×4160 到 3744×4992 之间。

所有数据集都是使用训练集(Train):验证集(Validation):测试集(Test)=7:2:1 的比例进行划分。对于 CASIA 数据集, 使用 CASIA-v2 来训练, 使用 CASIA-v1 进行测试。在测试环节上, 实验选取训练中验证集的损失最低或者 F1 最高的那一轮作为最终测试的模型权重, 测试阶段使用整图测试, 因为某些数据集的图片尺寸过大, 实验环境的显存限制无法加载整图, 针对这种情况使用滑动窗口的形式测试然后拼接成完整的预测结果。

4.2 评价指标

本文选择 F1、IOU 和 AUC 三个指标对不同算法下 DenseFCN 篡改定位的性能进行衡量。

F1 由查准率(Precision)和查全率(Recall)的调和平均给出, 相应定义如下:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (15)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (16)$$

$$F1 = 2 \cdot \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (17)$$

IoU 在篡改定位中被定义为算法预测的篡改区域和实际的篡改区域的交集与并集面积之比, 公式如下:

$$\text{IoU} = \frac{TP}{TP + FP + FN} \quad (18)$$

AUC 在篡改定位中, 是以像素预测结果为单位。公式如下:

$$\text{AUC} = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

F1 指标综合考虑了查全率和查准率, 与 IoU 和 AUC 相比更能反映篡改定位模型的性能。

4.3 对比实验与分析

(1) 在常规数据集上的精度对比

表 2 和表 3 列出了本文提出的动态掩码率蒸馏 DMGD 在公开的篡改定位数据集 CASIA、Columbia、NIST2016 和 DEFACTO 以及私人的 PS 书本封面篡改数据集 PS-arbitrary 和 PS-boundary 上与其他算法的对比结果。在表中, student 表示未使用蒸馏方式的 LSQ 量化方法, Student+MGD 表示使用了 MGD 蒸馏的 LSQ 量化方法, Student+DMGD 表示使用了 DMGD 蒸馏的 LSQ 量化方法。如表 1 所示, PS-boundary 数据集进行了 4 比特的量化实验。结果表明, 在固定遮掩率情况下, 不同的遮掩率会对训练后的量化模型精度产生不同的影响。然而, 随着固定遮掩率的降低, 模型的训练逐渐变得不稳定。在极

端情况下, 如固定遮掩率为 0 时, 模型将无法收敛。因此, 我们选择了 0.5 作为 DMGD 训练的动态遮掩率初始值。这可以确保模型正常收敛, 并且为动态遮

掩率参与训练提供足够的变化空间。同时, 为了控制变量, MGD 的遮掩率也统一选择为 0.5。实验中训练的 epoch 均设置为 100。

表 2 在常规数据集上的实验结果(一), DMGD 与其他算法的对比结果, 数据中-表示未收敛

Table 2 Results of experiments conducted on conventional datasets (1), comparing DMGD with other algorithms. Additionally, - indicates the instances where the data did not converge

方法	数据集	DEFACTO			PS-boundary			PS-arbitrary		
		F1	IOU	AUC	F1	IOU	AUC	F1	IOU	AUC
Float32	-	0.69	0.596	0.987	0.875	0.802	0.99	0.788	0.695	0.985
Int8	Student	0.719	0.626	0.981	0.879	0.807	0.99	0.818	0.738	0.991
	Student	0.548	0.462	0.942	0.553	0.473	0.937	0.294	0.201	0.925
Int4	Student+MGD	0.567	0.473	0.961	0.724	0.613	0.963	0.508	0.397	0.953
	Student+DMGD	0.580	0.481	0.977	0.758	0.650	0.975	0.517	0.414	0.967
Int3	Student	0.468	0.379	0.924	0.2291	0.151	0.829	0.148	0.09	0.867
	Student+MGD	0.467	0.380	0.939	0.242	0.158	0.812	0.308	0.214	0.902
Int2	Student+DMGD	0.4721	0.389	0.946	0.249	0.162	0.834	0.325	0.229	0.912
	Student	0.257	0.202	0.901	0.140	0.083	0.557	-	-	-
Int1	Student+MGD	0.2609	0.213	0.916	0.144	0.084	0.714	0.092	0.056	0.751
	Student+DMGD	0.269	0.215	0.922	0.151	0.089	0.746	0.091	0.053	0.762

表 3 在常规数据集上的实验结果(二), DMGD 与其他算法的对比结果

Table 3 Experimental results on conventional datasets (2), comparison results of DMGD with other algorithms

方法	数据集	Columbia			CASIA-V1			NIST2016		
		F1	IOU	AUC	F1	IOU	AUC	F1	IOU	AUC
Float32	-	0.736	0.682	0.949	0.390	0.333	0.715	0.397	0.319	0.803
Int8	Student	0.753	0.695	0.935	0.392	0.335	0.741	0.447	0.355	0.844
	Student	0.654	0.586	0.920	0.360	0.294	0.736	0.393	0.304	0.828
Int4	Student+MGD	0.671	0.598	0.916	0.367	0.313	0.747	0.385	0.293	0.808
	Student+DMGD	0.681	0.624	0.927	0.373	0.314	0.737	0.391	0.301	0.819
Int3	Student	0.547	0.445	0.825	0.314	0.247	0.745	0.356	0.267	0.797
	Student+MGD	0.562	0.494	0.841	0.336	0.271	0.729	0.355	0.274	0.784
Int2	Student+DMGD	0.632	0.544	0.874	0.341	0.264	0.743	0.358	0.276	0.768
	Student	0.464	0.353	0.766	0.099	0.063	0.549	0.184	0.129	0.648
Int1	Student+MGD	0.457	0.356	0.760	0.121	0.075	0.562	0.230	0.161	0.697
	Student+DMGD	0.456	0.340	0.77	0.152	0.096	0.596	0.235	0.169	0.703

表中加粗字体表示最高值。实验结果显示, 使用蒸馏的方式能有效提升 DenseFCN 量化训练的精度, 尤其对于 PS-arbitrary 和 PS-boundary 数据集效果显著。例如, 对于 PS-boundary 数据集, 采用蒸馏方法训练的 4 比特的 LSQ 量化模型的 F1 分别是为 0.724 和 0.758, 分别比未使用蒸馏方式的结果提高了 17.1%和 20.5%。在 PS-arbitrary 数据集中, MGD 和 DMGD 相对于未使用蒸馏的结果分别提高了 21.4%和 22.3%。

本文提出的 DMGD 算法与 MGD 算法进行了对比, 结果显示在多个数据集上, DMGD 实现了低比特

量化的 F1、IoU 和 AUC 的最优值。例如, 在 Defacto 数据集中, 4 比特量化的 DMGD 相比 MGD 的 F1、IOU 和 AUC 分别提高了 1.25%、0.8%和 1.6%。在 PS-boundary 数据集中, 4 比特量化的 DMGD 相比 MGD 提高了 F1、IOU 和 AUC 分别达到 2.79%、2.7%和 1.1%。特别地, 4 比特量化下, 在 NIST2016 数据集上, DMGD 与仅使用 Student 方法相比, 在 F1 和 IoU 指标上仅差 0.002 和 0.003。而 MGD 与 Student 方法相比, 在 F1 和 IoU 指标上差距更大, 分别为 0.008 和 0.011。这表明 DMGD 与 Student 方法的性能相近, 且在多次实验的平均结果中, 差异可视为

误差。相反, MGD 的性能差异较大, 这部分原因是原论文中 MGD 的蒸馏未涉及量化实验, 因此 MGD 在特定量化场景下的表现不佳是合理的。此外, 在 3 比特 NIST2016 数据集的场景下, MGD 的 F1 指标也略低于 Student 方法, 进一步证实了 MGD 在某些量化场景下的劣势。同时注意到, 在 2 比特 Columbia 数据集场景下, 也存在 Student 表现更佳的情况, 但是 Columbia 数据是可溯源的最早期的图像篡改检测数据集, 具有较小的数据量和单一的篡改方式, 因此在极端比特量化下模型的参数空间较小, 性能波动是合理的。对于 Columbia 数据集, 在 3 比特的量化蒸馏实验中, DMGD 的精度提升幅度较大, 相比 MGD 算法, F1、IOU 和 AUC 分别提高了 7%、5% 和 3.3%。在 2 比特 Ps-arbitrary 数据集的场景下, Student+MGD 的效果是略优于 Student+DMGD 的, 但差距不大。在低比特场景下, 由于模型的参数表达范围显著缩小, 如 2 比特时每个参数只有 4 种可能值, DMGD 和 MGD 之间的性能差距缩小, 加之波动因素, 因此出现了这一结果。

(2) 消融分析

本小节为了评估余弦退火算法和 DMGD 算法的有效性, 我们进行了消融实验。表 2 和表 3 展示了 DMGD 结合余弦退火算法的结果。我们通过实验发现, 未使用余弦退火抑制动态遮掩率的模型未能收敛, 导致无法得出后续的结果。这是因为在训练初始阶段, 模型权重是随机生成的, 未经过监督训练, 缺乏有效的信息。因此, 如公式(4)所示, 我们引入了余弦退火的思想。在训练初始的 epoch, 我们通过抑制动态模块提供的动态遮掩率, 使大部分权重参考初始值。随着 epoch 的增加, 动态模块的权重逐渐增加。

其次, 加入为了评估加入蒸馏模块后的 DMGD 方法训练成本是否增加, 我们任选一个数据集实验, 这里是在 Columbia 数据集上以 2 比特、3 比特和 4 比特量化级别训练了 MGD 和 DMGD 方法。所有其他变量保持不变。如表 4 所示, 我们记录了训练稳定

后一轮的时间消耗, s 代表秒数。结果表明, 尽管 DMGD 方法在各个量化级别上的时间成本略高于 MGD 方法, 但增幅不超过 10%。考虑到 MGD 的参数量为 52778444, 而 DMGD 的为 5296372, 我们得出结论: 引入蒸馏模块后, DMGD 方法的参数略有增加, 但其时间效率几乎未受影响。这表明 DMGD 方法在增强性能的同时, 仍保持了较高的计算效率。

(3) 训练曲线分析

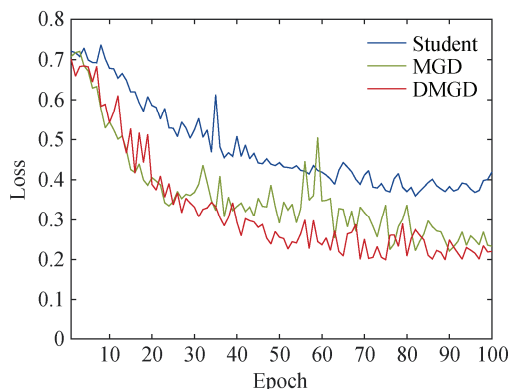
图 4 展示了在 PS-boundary、PS-arbitrary 和 Columbia 数据集在无蒸馏方式、MGD 蒸馏量化和 DMGD 蒸馏量化训练中每轮的验证集 loss 和 F1 的值绘制而成的折线图, 其中 Student 表示不使用任何蒸馏方式的量化感知训练曲线。

折线图显示, 在使用蒸馏方式的量化感知训练中, Loss 下降速度快, 瓶颈期较短。此外, MGD 和 DMGD 在训练初期(前 30~40 个 Epoch)时, Loss 下降走势相似。但是, 随着训练的进行, MGD 的 Loss 下降速率趋于平缓, 原因是 MGD 的训练方式令生成模块在后期已经适应了特定掩码率; 相反, DMGD 通过动态调整掩码率, 突破了固定掩码率的训练后期瓶颈, 这一特点在 Loss 变化图中有明显体现, 如在 PS-boundary 数据集第 30 个 Epoch 左右, MGD 和 DMGD 的 Loss 下降开始出现差距。这些特点在 F1 折线图也有相应的效果。总体来看, DMGD 的验证集效果显著优于 MGD。

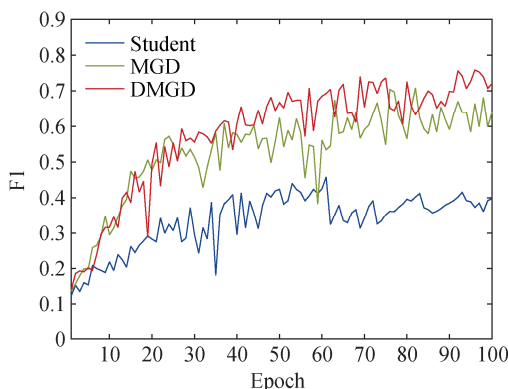
表 4 DMGD 和 MGD 消融实验

Table 4 Ablation Experiments of DMGD and MGD

比特	方法	时间代价
Int2	Student+MGD	131.07s
	Student+DMGD	131.88s
Int3	Student+MGD	74.24s
	Student+DMGD	74.69s
Int 4	Student+MGD	74.63s
	Student+DMGD	74.85s



(a) PS-boundary, int4, Loss



(b) PS-boundary, int4, F1

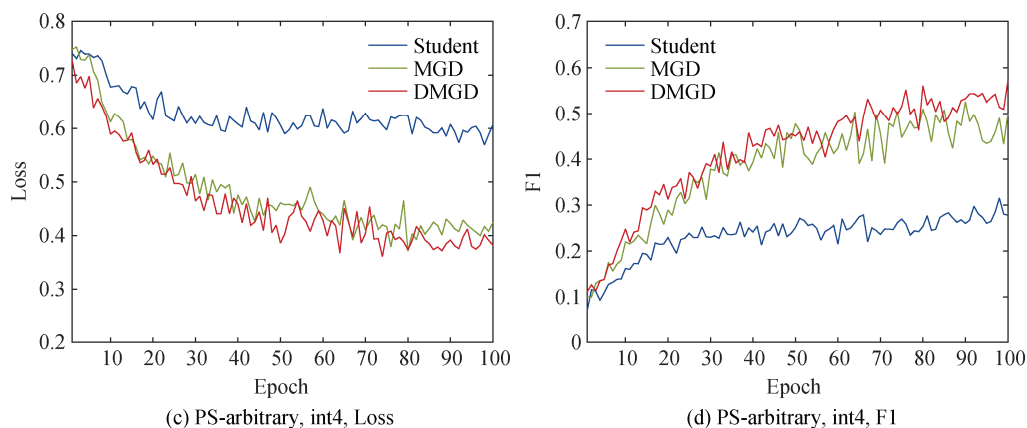


图 4 无蒸馏(Student)和使用 MGD、DMGD 在 PS-boundary 和 PS-arbitrary 数据集上分别使用 4、4 和 3 比特量化训练时每轮 epoch 验证集的 loss 和 F1 值的折线图

Figure 4 Line chart of the loss and F1 values of each epoch validation set when quantized training using 4, 4 and 3 bits on PS-boundary and PS-arbitrary datasets using 4, 4 and 3 bits using MGD and DMGD without distillation (Student) and ps-arbitrary, respectively

(4) 中间特征可视化

在实验中, 我们使用了 DenseFCNDenseBlock5 生成的特征进行特征蒸馏, 并将该模块产生的特征

以可视化热图的形式呈现, 如图 5 所示。图中使用了 Defacto 数据集中 copy-move 篡改类型的篡改图片为例, 最左和最右的图片是原始图片和真实标签, 其

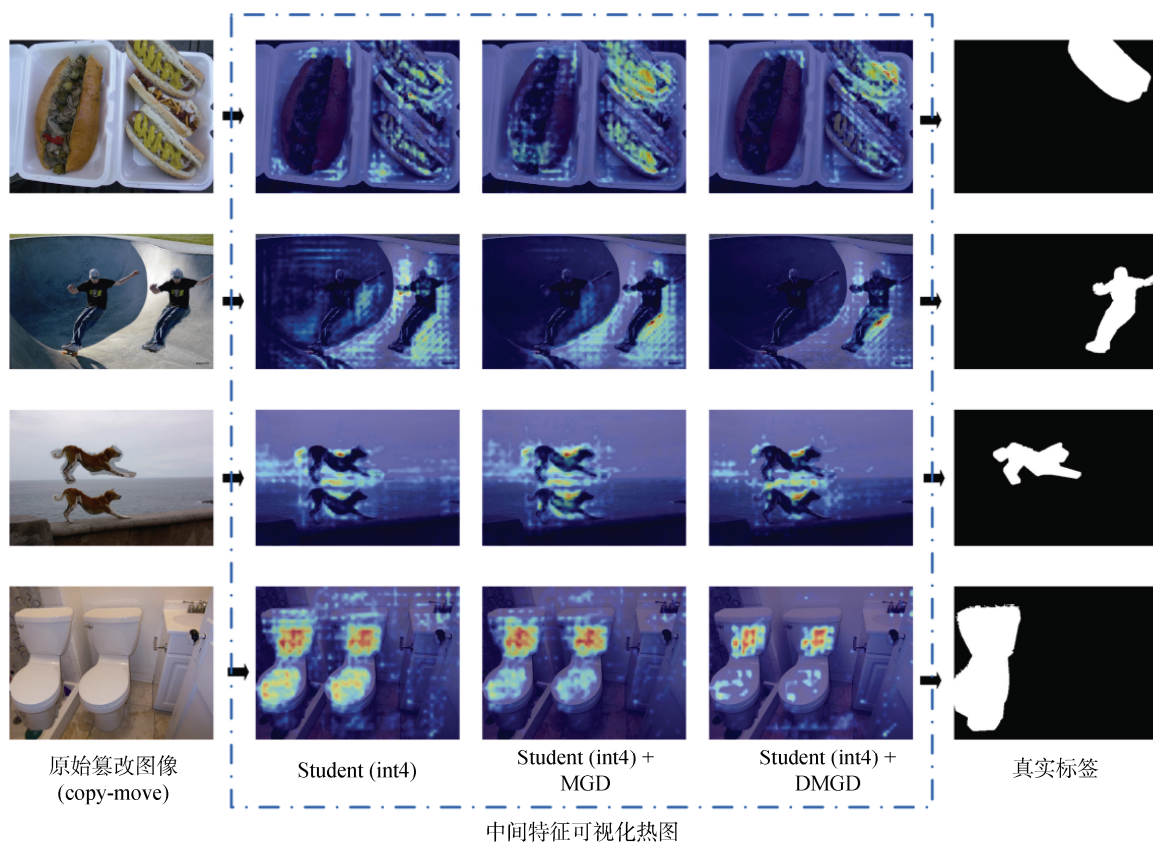


图 5 分别使用无蒸馏(Student)、MGD 蒸馏 DMGD 蒸馏在 Defacto 数据集上进行 4 比特量化感知训练下的 DenseFCN 的 Denseblock5 模块产生的特征可视化热图

Figure 5 Heat map of feature visualization generated by DenseFCN's Denseblock5 module under 4-bit quantitative perception training on Defacto dataset using studentless (Student) and MGD distillation DMGD distillation, respectively

中间三个图分别是 4 比特量化下的中间特征热图叠加到原图的可视化合成图, Student 是未使用任何蒸馏算法的情况。

结果表明, 虽然无蒸馏方式产生的特征注意力范围覆盖了真实的篡改区域, 但同时也产生了大量的噪声, 这是影响精度的主要因素。相比之下, 另外两种蒸馏方式 MGD 和 DMGD 产生的噪声明显要少很多。对比 MGD 和 DMGD 产生的特征图, DMGD 的注意力更加精准, 同时产生的噪声也更少。此外, DMGD 产生的特征显示其注意力的分布较为密集且集中, 这有利于模型精度的提升。

4.4 跨数据集性能测试实验

本实验对 DMGD 蒸馏在跨数据集的性能进行了测试, 实验使用了 4、3 和 2 低比特量化, 分别对比无蒸馏的 LSQ 量化和使用 MGD 和 DMGD 的 LSQ 量化, 另外还加入了 8 比特和满精度的浮点数模型进行测试。实验使用 CASIA-v2 作为训练, 然后分别

使用 Columbia、Nist2016、Defacto 和 CASIA-v1 作为测试集, CASIA-v2 的训练参数跟 4.1 节描述的实验设置保持一致。

实验结果如表 5 所示, 结果表明, 使用 CASIA-v2 训练后的模型, 在其他数据集上的精度都会有一定幅度的下降, 这是因为跨域数据集涵盖一些未知的篡改特征。除此之外, 使用了蒸馏的量化训练普遍比未使用蒸馏的效果占优, 说明蒸馏对篡改定位模型量化训练的提升较为明显。DMGD 蒸馏方式在多个数据集上的不同比特数中, F1 的精度都表现优秀, 如 Columbia 数据集 4 比特量化方式的 F1、IOU 和 AUC 比未使用蒸馏和 MGD 蒸馏分别提高了 12.5%、9.1%、4.7%和 2.5%、1.8%、0.2%; 在 NIST2016 数据集的 2 比特量化中, 提升的幅度也较大, 分别为 4.6%、2.8%、7.4%和 3.8%、2.2%、2.5%。综上所述, 本文提出的 DMGD 算法在 CASIA-v2 数据集上训练后, 进行跨域数据集测试时具有更好的性能。

表 5 分别使用无蒸馏 LSQ(Student)、MGD 和 DMGD 蒸馏进行 4、3 和 2 低比特量化, 在 CASIA-v2 数据集训练, 在 Columbia、NIST2016、Defacto 和 CASIA-v1 数据集上的跨域测试表现

方法		Student			Student+MGD			Student+DMGD		
模型类型	数据集	F1	IOU	AUC	F1	IOU	AUC	F1	IOU	AUC
Float	Columbia	0.394	0.254	0.503	-	-	-	-	-	-
	Nist2016	0.242	0.181	0.686	-	-	-	-	-	-
	Defacto	0.156	0.108	0.781	-	-	-	-	-	-
	CASIA-v1	0.39	0.333	0.715	-	-	-	-	-	-
Int8	Columbia	0.475	0.327	0.648	-	-	-	-	-	-
	NIST2016	0.233	0.162	0.692	-	-	-	-	-	-
	Defacto	0.175	0.121	0.806	-	-	-	-	-	-
	CASIA-v1	0.392	0.335	0.741	-	-	-	-	-	-
Int4	Columbia	0.408	0.268	0.538	0.360	0.228	0.557	0.411	0.275	0.606
	NIST2016	0.103	0.062	0.608	0.094	0.061	0.579	0.088	0.053	0.576
	Defacto	0.111	0.071	0.734	0.115	0.074	0.723	0.117	0.075	0.730
	CASIA-v1	0.360	0.294	0.736	0.367	0.313	0.747	0.373	0.314	0.737
Int3	Columbia	0.266	0.161	0.542	0.366	0.234	0.587	0.391	0.252	0.589
	NIST2016	0.084	0.048	0.566	0.080	0.044	0.575	0.107	0.061	0.589
	Defacto	0.130	0.085	0.084	0.130	0.086	0.074	0.131	0.089	0.854
	CASIA-v1	0.314	0.247	0.745	0.336	0.271	0.729	0.341	0.264	0.743
Int2	Columbia	0.084	0.046	0.395	0.123	0.068	0.430	0.157	0.089	0.439
	NIST2016	0.038	0.020	0.446	0.046	0.026	0.495	0.084	0.048	0.520
	Defacto	0.045	0.026	0.559	0.042	0.024	0.575	0.048	0.028	0.577
	CASIA-v1	0.099	0.063	0.549	0.121	0.075	0.562	0.152	0.096	0.596

5 结束语

针对篡改取证模型规模日益庞大、参数量膨胀等因素导致部署后推理速度慢的问题, 本文使用改进的知识蒸馏方式对图像篡改定位模型 DenseFCN 进行量化感知训练, 旨在提高模型低比特量化后的精度。针对 MGD 蒸馏算法在 DenseFCN 量化中存在的缺陷, 即固定掩码率会导致 MGD 蒸馏遇到瓶颈, 本文提出了构建动态模块, 根据不同的学生特征和教师特征产生动态掩码率, 形成 DMGD(Dynamic MGD)蒸馏的方法。改进后的算法在多个公开和私人图像篡改数据集上均取得了最优的精度, 并对跨域数据具有良好的泛化性。在后续的研究工作中, 作者将致力于进一步优化现有方法, 并将深入探讨量化蒸馏与篡改特征相结合的方式。例如, 在特征蒸馏过程中融入篡改注意力机制, 以提高对篡改特征的提取、模仿以及学习的效率。

参考文献

- [1] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks[J]. *Communications of the ACM*, 2017, 60(6): 84-90.
- [2] Bi X L, Wei Y, Xiao B, et al. RRU-Net: The Ringed Residual U-Net for Image Splicing Forgery Detection[C]. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020: 30-39.
- [3] Wu Y, AbdAlmageed W, Natarajan P. ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries with Anomalous Features[C]. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020: 9535-9544.
- [4] Hu X F, Zhang Z H, Jiang Z Y, et al. SPAN: Spatial Pyramid Attention Network for Image Manipulation Localization[C]. *Computer Vision – ECCV 2020*, 2020: 312-328.
- [5] Zhou P, Han X T, Morariu V I, et al. Learning Rich Features for Image Manipulation Detection[C]. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018: 1053-1061.
- [6] Chen X R, Dong C B, Ji J Q, et al. Image Manipulation Detection by Multi-View Multi-Scale Supervision[C]. *2021 IEEE/CVF International Conference on Computer Vision*, 2022: 14165-14173.
- [7] Liu X H, Liu Y J, Chen J, et al. PSCC-Net: Progressive Spatio-Channel Correlation Network for Image Manipulation Detection and Localization[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2022, 32(11): 7505-7517.
- [8] Shelhamer E, Long J, Darrell T. Fully Convolutional Networks for Semantic Segmentation[C]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016: 640-651.
- [9] Salloum R, Ren Y Z, Jay Kuo C C. Image Splicing Localization Using a Multi-Task Fully Convolutional Network (MFCN)[J]. *Journal of Visual Communication and Image Representation*, 2018, 51: 201-209.
- [10] Zhuang P Y, Li H D, Tan S Q, et al. Image Tampering Localization Using a Dense Fully Convolutional Network[J]. *IEEE Transactions on Information Forensics and Security*, 2021, 16: 2986-2999.
- [11] Wu H W, Zhou J T, Tian J Y, et al. Robust Image Forgery Detection over Online Social Network Shared Images[C]. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022: 13430-13439.
- [12] Chen T, Li B, Zeng J H. Learning Traces by Yourself: Blind Image Forgery Localization via Anomaly Detection with ViT-VAE[J]. *IEEE Signal Processing Letters*, 2023, 30: 150-154.
- [13] Chen J X, Liao X, Wang W, et al. SNIS: A Signal Noise Separation-Based Network for Post-Processed Image Forgery Detection[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023, 33(2): 935-951.
- [14] Huang G, Liu Z, Van Der Maaten L, et al. Densely Connected Convolutional Networks[C]. *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017: 2261-2269.
- [15] Courbariaux M, Hubara I, Soudry D, et al. Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1[EB/OL]. 2016: arXiv: 1602.02830. <https://arxiv.org/abs/1602.02830>.
- [16] Esser S K, McKinstry J L, Bablani D, et al. Learned Step Size Quantization[EB/OL]. 2019: arXiv: 1902.08153. <https://arxiv.org/abs/1902.08153>.
- [17] Zhou S C, Wu Y X, Ni Z K, et al. DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients[EB/OL]. 2016: arXiv: 1606.06160. <https://arxiv.org/abs/1606.06160>.
- [18] Xu K X, Lee A H X, Zhao Z Y, et al. MetaGrad: Adaptive Gradient Quantization with Hypernetworks[EB/OL]. 2023: arXiv: 2303.02347. <https://arxiv.org/abs/2303.02347>.
- [19] Hinton G, Vinyals O, Dean J. Distilling the Knowledge in a Neural Network[EB/OL]. 2015: arXiv: 1503.02531. <https://arxiv.org/abs/1503.02531>.
- [20] Zhang Y, Xiang T, Hospedales T M, et al. Deep Mutual Learning[C]. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018: 4320-4328.
- [21] Fukuda T, Suzuki M, Kurata G, et al. Efficient Knowledge Distillation from an Ensemble of Teachers[C]. *Interspeech 2017*, 2017: 3697-3701.
- [22] Romero A, Ballas N, Kahou S E, et al. FitNets: Hints for Thin Deep Nets[EB/OL]. 2014: arXiv: 1412.6550. <https://arxiv.org/abs/1412.6550>.
- [23] Zhao B R, Cui Q, Song R J, et al. Decoupled Knowledge Distillation[C]. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022: 11943-11952.
- [24] Yang Z D, Li Z, Shao M Q, et al. Masked Generative Distillation[C]. *Computer Vision – ECCV 2022*, 2022: 53-69.
- [25] Loshchilov I, Hutter F. SGDR: Stochastic Gradient Descent with Warm Restarts[EB/OL]. 2016: arXiv: 1608.03983. <https://arxiv.org/abs/1608.03983>.
- [26] Milletari F, Navab N, Ahmadi S A. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation[C]. *2016 Fourth International Conference on 3D Vision*, 2016:

- 565-571.
- [27] Hsu Y F, Chang S F. Detecting Image Splicing Using Geometry Invariants and Camera Characteristics Consistency[C]. *2006 IEEE International Conference on Multimedia and Expo*, 2006: 549-552.
- [28] Guan H Y, Kozak M, Robertson E, et al. MFC Datasets: Large-Scale Benchmark Datasets for Media Forensic Challenge Evaluation[C]. *2019 IEEE Winter Applications of Computer Vision Workshops*, 2019: 63-72.
- [29] Mahfoudi G, Tajini B, Retraint F, et al. DEFACTO: Image and Face Manipulation Dataset[C]. *2019 27th European Signal Processing Conference*, 2019: 1-5.
- [30] Lin T Y, Maire M, Belongie S, et al. Microsoft COCO: Common Objects in Context[C]. *Computer Vision – ECCV 2014*, 2014: 740-755.
- [31] Dong J, Wang W, Tan T N. CASIA Image Tampering Detection Evaluation Database[C]. *2013 IEEE China Summit and International Conference on Signal and Information Processing*, 2013: 422-426.



黎思力 于 2023 年在深圳大学计算机技术专业获得硕士学位。研究领域为图像篡改取证、模型压缩。Email: 912824958@qq.com



廖桂樱 于 2024 年在深圳大学人工智能专业获得硕士学位。研究领域为图像篡改取证、模型压缩。Email: hug_lgy@163.com



谭舜泉 现任深圳北理莫斯科大学教授, CCF 会员。研究领域为多媒体取证、隐写及隐写分析、深度学习等。Email: tansq@smbu.edu.cn



黄继武 现任深圳北理莫斯科大学教授, CCF 会员。研究领域为多媒体取证、信息隐藏、多媒体信号处理、模式识别等。Email: jwhuang@smbu.edu.cn



李斌 现任深圳大学电子与信息学院教授, CCF 会员。研究领域为多媒体信息安全、智能处理等。Email: libin@szu.edu.cn