



基于近似存在性查询的高效图像异常检测方法

伍凌川¹, 史慧芳¹, 邱枫^{1,2}, 石义官^{1,2*}

(1. 中国兵器装备集团自动化研究所有限公司, 绵阳 621000; 2. 北京理工大学 机械与车辆学院, 北京 100081)

摘要 对于图像异常检测问题, 查询测试样本在正常样本集中的 K 近邻距离并估计其异常程度, 是一类准确率较高、对复杂分布的效果较稳定的方法。此类方法采用近似最近邻搜索 (Approximate Nearest Neighbour Search, ANNS) 索引进行 K 近邻搜索。但由于 ANNS 查询操作较高的计算开销和现实问题中庞大的数据量, 此类方法的计算效率难以应对低时延、高吞吐量的应用场景。该文基于局部敏感哈希和布隆过滤器, 提出了一种近似存在性查询 (Approximate Membership Query, AMQ) 方法, 用特征近似存在性预测异常样本。相比于 ANNS, AMQ 具有更低的计算复杂度且更适合单指令多数据并行, 可以有效解决基于特征库检索方法的计算性能瓶颈。在 MVTec-AD 数据集上的实验结果显示, 基于 AMQ 的方法的异常分割准确率仅比 ANNS 方法降低 1% 左右, 但推理时延、吞吐量、内存开销显著较优, 接近端到端深度学习异常检测模型的计算效率。

关键词 异常检测; 无监督学习; 近似存在性查询; 布隆过滤器; 局部敏感哈希

中图分类号 TP391.4

文献标志码 A

DOI 10.12178/1001-0548.2024032

An Efficient Image Anomaly Detection Approach Based on Approximate Membership Query

WU Lingchuan¹, SHI Huifang¹, QIU Feng^{1,2}, and SHI Yiguan^{1,2*}

(1. Automation Research Institute Co., Ltd. of China South Industries Group Corporation, MianYang 621000, China; 2. School of mechanical engineering, Beijing Institute of Technology, Beijing 100081, China)

Abstract An accurate and stable approach to image anomaly detection is to query the K-nearest neighbours of the image features from normal examples and estimate the anomaly score, relying on approximate nearest neighbour search (ANNS) indices. ANNS query operation has high computational cost on large datasets, unpractical for low-latency and high-throughput scenarios. Based on locality sensitive hashing and Bloom filters, an approximated membership query (AMQ) based approach is proposed to predict anomalies by approximate membership of features. AMQ can address the performance bottleneck of search-based methods, given its lower complexity and better compatibility with single-instruction multiple-data parallelism than ANNS. Experimental results on MVTec-AD show that the accuracy of AMQ-based method is just decreased about 1% in comparison with ANNS-based methods, while the inference latency, the throughput and the memory footprint are significantly improved, close to the efficiency of end-to-end deep learning anomaly detection models.

Key words Anomaly Detection; Unsupervised Learning; Approximate Membership Query; Bloom Filter; Locality Sensitive Hashing

无监督图像异常检测 (Unsupervised Image Anomaly Detection) 任务在仅有正常图像训练样本的条件下, 训练模型从可能含有异常区域的图像中识别异常区域位置 (通常以图像语义分割的形式输出)。无监督图像异常检测在工业缺陷检测、安防

监控、卫星遥感、军事侦察、医学图像分析等领域具有广泛的应用场景^[1]。特别是在工业制造领域的产品表面缺陷检测任务中, 产品缺陷类型种类繁多、形态尺度多变、标注成本高, 给有监督模型训练带来了困难。因此经常依赖于无监督图像异常检

收稿日期: 2024-02-12; 修回日期: 2024-03-20

基金项目: 国防基础科研 (JCKY2022209A002); 国家自然科学基金重点项目 (T2293771)

作者简介: 伍凌川 (1978—), 男, 正高级工程师, 主要从事弹药药装配工艺技术、智能制造技术研究。

*通信作者 E-mail: shiyg888@163.com

测技术, 仅使用无缺陷的样本进行训练, 识别任何未知的缺陷。

目前, 基于深度学习的无监督图像异常检测领域, 主要有基于重建 (Reconstruction) 的方法、基于概率密度估计 (Probability Density Estimation) 的方法和基于特征库检索 (Memory Bank Retrieval) 的方法等。

基于重建的研究思路以生成对抗网络 (Generative Adversarial Network, GAN)、变分自编码 (Variational Auto-Encoder, VAE) 网络等图像生成模型或知识蒸馏架构的教师-学生模型为基础, 用正常图像数据训练模型进行图像压缩重建或表征学习训练。推理时异常的图像特征没有参与过训练、难以被模型准确地编码-解码, 因此重构误差较高的区域可被判定为异常。代表性的方法有文献 [2] 提出的基于 GAN 提出的重构异常检测模型和文献 [3] 提出的基于蒸馏的单分类方法等。文献 [4] 提出的 Efficient-AD 模型, 结合了蒸馏和生成模型两种思路并使用轻量级主干网络, 取得了毫秒级的时延。文献 [5] 提出了一种使用对比学习训练 GAN 模型的方法进行基于重建的异常检测, 获得了较高的准确率。

基于概率密度估计的研究思路源于近年来标准化流 (Normalizing Flow) 模型的快速发展。虽然异常检测问题很容易抽象为概率密度估计问题 (低概率密度样本可认为是异常), 但传统的参数概率模型、核密度估计等方法难以适应高维、复杂的图像特征分布。如文献 [6] 提出的基于区域的核密度估计方法主要适用于较小的数据集和较低的维度。标准化流模型则为图像特征提供了一种高效的概率建模方式。文献 [7] 提出的 CFlow 模型, 利用条件标准化流 (Conditional Normalizing Flow) 对不同场景下的概率密度建模并检测异常。文献 [8] 结合 U-Net 的模型结构特点和标准化流模型, 提出了 U-Flow 模型用多层次的检测结果增强鲁棒性。文献 [9] 在标准化流基础上改进了更高效的模型结构, 提升了推理速度。

基于特征库检索的方法沿袭了基于 K 近邻的机器学习异常检测思路, 用深度学习模型从图像中提取不同尺度的图像特征并加入近似最近邻搜索 (Approximate Nearest Neighbour Search, ANNS) 索引结构; 推理时, 提取测试图像的深度特征为查询向量搜索 K 近邻, 并使用 K 近邻的距离计算异

常得分, 距离越远、异常得分越高。文献 [10] 提出以马氏距离 (Mahalanobis Distance) 计算测试样本与正常样本的深度特征距离, 但其距离计算复杂度限制了可扩展性。文献 [11] 提出的 PatchCore 模型, 通过 Core Set 采样和 ANNS 索引结构大幅度减少了计算量, 在文献 [12] 等数据集上取得了领先的准确率。文献 [13] 对特征库采样与检索结构进行了改进, 在保持准确率的同时提高了检索效率。

文献 [14] 的实验结果显示, 相比于另两种研究思路, 基于特征库检索的方法在各类图像异常检测数据集上能更稳定地取得较高的准确率, 对复杂分布的适应性更强。然而, 特征库检索的计算量较大、内存开销较高。相比于端到端的生成模型和标准化流模型, 更难以部署到要求高吞吐量、低时延的应用场景。

基于特征库搜索的基本思路如图 1 所示。此类方法使用卷积神经网络从正常样本图像中提取各个区域的深度特征向量, 然后存储进层次导航小世界图 (Hierarchical Navigable Small World, HNSW)^[15]、乘积量化倒排索引 (Inverted File Product Quantization, IVFPQ)^[16] 等 ANNS 索引结构。在推理时, 使用 ANNS 索引搜索测试图像深度特征的 K 近邻, 并以 K 个最近邻的正常特征的距离计算测试集样本的异常分数。通常距离越远, 则异常分数越高。

该技术路径的主要计算性能瓶颈在于 K 近邻搜索。目前最高效的几类 ANNS 索引结构的搜索操作时间复杂度为对数级, 在正常样本训练集非常大时, 搜索操作时延较长。此外, 基于 K-D 树或近邻图的索引结构难以利用 GPU 的高并行度优势, 在高吞吐量的应用场景中存在性能劣势, 搜索速度远低于特征提取速度。

实际上, 推理流程中异常检测模型并不需要知道 K 近邻具体来自于哪些样本, 而只需要知道 K 近邻与查询向量的距离。换言之, 如果能低代价地估计查询向量在特征空间中近邻区域是否存在近似向量, 并以此替代 ANNS, 就能解决此类方法的性能瓶颈。因此, 本文所提出方法的核心思路是将计算量较大的深度特征近似 K 近邻搜索替换为近似存在性查询 (Approximate Membership Query, AMQ), 从而减少计算复杂度、提高模型推理速度。

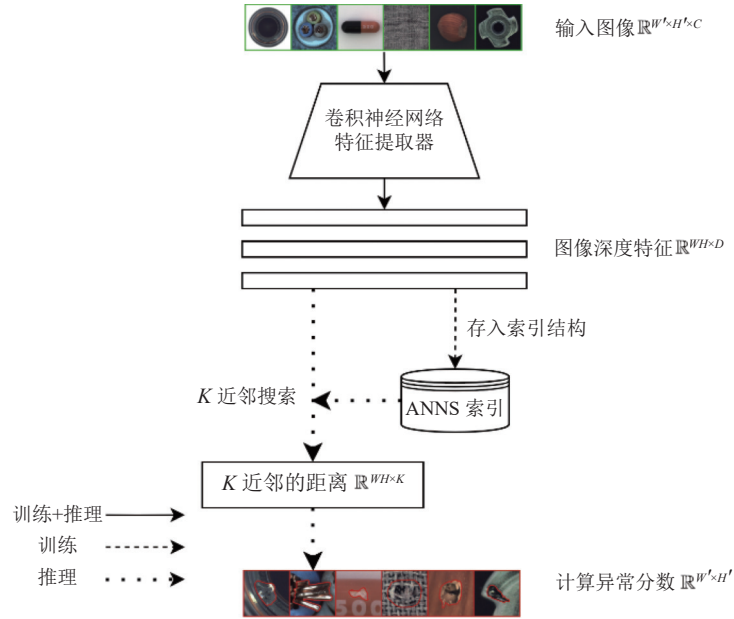


图1 基于特征库检索的图像异常检测流程图

1 方法

本文方法的主要流程如图2所示。在经由卷积神经网络提取深度特征后,使用多组局部敏感哈希(Locality Sensitive Hashing, LSH)把卷积神经网络提取的深度特征向量编码为多组整数,然后存入布隆过滤器(Bloom Filter, BF)。推理时,从测试图像中提取的深度特征进行局部敏感哈希编码后,

在布隆过滤器中查询,最后用存在性查询结果计算异常分数。该方法在推理时的主要计算性能优势在于,用常数时间复杂度的局部敏感哈希编码和布隆过滤器查询操作替代了复杂的ANNS搜索,并且此设计允许实现过程中充分利用GPU高并行度的优势,因此获得了更低的时延和更高的吞吐量。

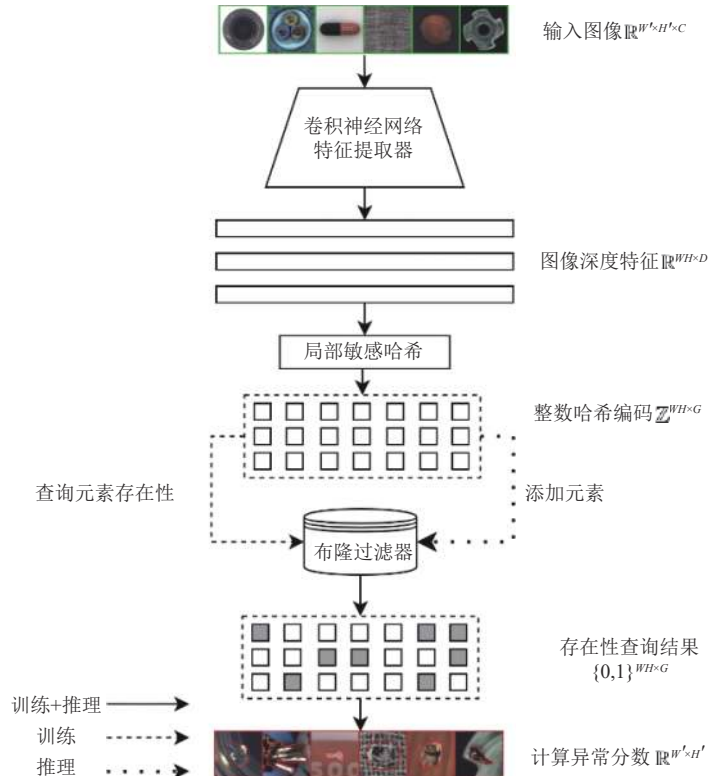


图2 基于特征近似存在性查询的图像异常检测流程图

1.1 图像深度特征提取

本文方法使用 Wide ResNet-50 作为特征提取网络，从图像中提取深度特征。Wide ResNet-50 作为一种具有代表性的卷积神经网络，比 ResNet-50 拥有更高的特征维度、推理计算量适中，在 PatchCore、CFlow、Efficient-AD 等相关工作中被广泛使用。与其他异常检测方法的实现方式相同，本文方法不对模型进行训练、而直接使用 ImageNet 数据集预训练的模型参数进行特征提取。

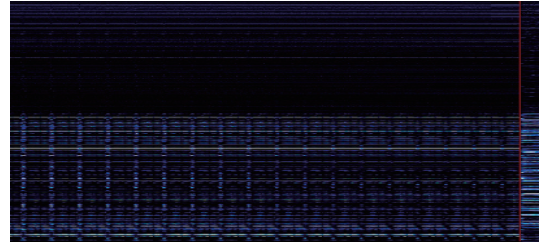
具体地，Wide ResNet-50 网络结构从宏观上可以分为 4 个阶段，每个阶段产生不同尺度的特征图 (Feature Map)。网络越深的阶段产生的特征图所包含的图像语义特征尺度越大、越复杂。Wide ResNet-50 特征提取网络中，第一个阶段的有效感受野非常小、比大多数细微的异常缺陷更小，无法提取出具有充足前后文信息的特征。考虑到像素级异常检测任务需要特征既精细 (分辨率高) 又语义丰富 (尺度大)，本文方法选择将后 3 个阶段的特征图融合。这三个特征图的特征维度分别为 512、1 024 和 2 048；当输入为宽 2 048、高 2 048 的图像时，特征图分辨率分 (高和宽仍相等) 别为 256、128 和 64 像素；感受野大小分别为 91、267 和 427 像素。具体方式为，使用双线性插值 (Bi-linear Interpolation) 方法将所有特征图插值为宽 256 像素、高 256 像素的尺寸，然后沿特征维度拼接到一起。最终，形成了单一的宽 256 (W)、高 256 (H)、维度 3 584 (D) 的深度特征图。

所得到的深度特征图中，包括异常区域特征和正常区域特征，图 3 所示是单个图像样本 (大小 256*256) 的缺陷区域及其特征分布。图 3c 中红线左侧每列是来自正常区域中一个像素的深度特征，红线右侧每列为来自异常区域中一个像素的深度特征 (为了直观，这里将缺陷部分的特征都集中在了最右边)。正常区域面积比例非常高，这里左侧只截取了部分正常区域的深度特征。从图 3c 可以直观地看到，异常区域的深度特征与正常区域存在差异。



(a) 原图

(b) 异常区域图



(c) 特征图

图 3 样本及其特征分布

1.2 局部敏感哈希

局部敏感哈希是一类将浮点数向量编码为整型或布尔型向量的哈希函数。本文使用随机投影局部敏感哈希 [17]，由式 (1) 所示局部敏感哈希函数 $h: \mathbb{R}^D \rightarrow \mathbb{Z}$ 对图像深度特征进行编码。

$$h(x) = \left\lfloor \frac{w^T x + b}{\varepsilon} \right\rfloor \quad (1)$$

式中， $w \in \mathbb{R}^D$ 是随机生成的归一化向量 (即 $\|w\| = 1$)、 $b \in \mathbb{R}$ 为随机采样的偏移量、 $\varepsilon \in \mathbb{R}^+$ 是用于控制编码区间宽度的超参。

本文选用的局部敏感哈希函数具有以下性质。对任意概率分布的两个浮点数向量之间的欧式距离越小，被编码为同一整型数的概率越高，即：

$$\forall x, y, z \in \mathbb{R}^D, d(x, y) \geq d(y, z) \Rightarrow p(h(x) = h(y)) \leq p(h(y) = h(z)) \quad (2)$$

式中， $d: \mathbb{R}^D \times \mathbb{R}^D \rightarrow \mathbb{R}^+$ 为实数向量欧式距离函数。而其逆否命题：

$$\forall x, y, z \in \mathbb{R}^D, p(h(x) \neq h(y)) < p(h(y) \neq h(z)) \Rightarrow d(x, y) < d(y, z) \quad (3)$$

是本文方法可以将较高时间复杂度的搜索问题转化为常数时间复杂度的存在性查询问题的关键。

虽然概率 $P(h(x) \neq h(y))$ 难以直接求解，但可以根据多组独立采样的局部敏感哈希函数对其进行估计。当使用一组独立随机生成的 G 个局部敏感哈希函数 $\{h_1, h_2, \dots, h_G\}$ 对向量进行哈希编码，两个向量被编码为不同哈希值的概率可以估计为：

$$\hat{p}(h(x) \neq h(y)) = \frac{1}{G} \sum_{i=1}^G 1(h_i(x) \neq h_i(y)) \quad (4)$$

其中 $1: \{True, False\} \rightarrow \{1, 0\}$ 为指示函数，命题为真时函数值为 1，为假时函数值为 0。

由式 (3) 可知，式 (4) 获得的估计值越大，向量间距离的估计值越大。而组数 G 越大，估计越准确。本文方法的特点之一是将连续的实数向量距

离度量问题近似地转化为离散的整数哈希值对比问题。本小节的理论依据支撑了该思路的有效性。

因为需要整数域的哈希值, 本文方法未使用如 Bit Sampling、Spherical LSH 等其他局部敏感哈希方法。这些方法虽然能通过哈希值的二进制拼接获得整数哈希值, 但相对计算开销更大、不适用于要求高计算效率的场景。

1.3 布隆过滤器与异常分数计算

对每一个图像样本的深度特征图, 采用随机投影局部敏感哈希进行映射, 将图中每一个像素的特征 (本文为 3 584 维) 映射到一个高维的哈希编码 (本文为 2 048), 从而得到一个高维的哈希映射图 (本文为 $256 \times 256 \times 2\,048$ 维)。实际进行异常检测时, 将测试样本的哈希编码与整个数据集的深度特征的哈希编码进行比较, 从而得到测试样本的异常分数。

多组独立的局部敏感哈希函数解决了由连续的实数向量转化为离散的哈希值的问题, 但没有完全解决搜索问题。直接对排序后的离散值集合进行搜索, 其最优的时间复杂度依然是对数级。实际上, 异常检测任务不需要搜索到哈希值在存储结构中的位置, 只需要确定它是否存在即可。因此, 本文方法使用布隆过滤器将问题转化为常数时间复杂度的存在性查询问题。

布隆过滤器^[18]是一种经典的概率数据结构, 用于在常数时间复杂度内查询一个元素 (哈希值) 是否存在于集合内。由于元素的哈希编码可能会冲突, 布隆过滤器查询会有一些的误报率 (False Positive Rate), 但可以通过调整哈希函数和内存空间消耗来控制。本文使用 Murmurhash3 哈希函数^[19]对局部敏感哈希的结果进行编码。哈希函数数量 B 由式 (5)^[18] 计算得到一个理论最优值:

$$B = \left\lceil \frac{M}{NWHG} \ln(2) \right\rceil \quad (5)$$

式中, N 为训练集图像数量, M 为布隆过滤器可以使用的内存空间比特数。

布隆过滤器获得查询结果 $E \in \{0, 1\}^{W \times H \times G}$, 其中 0 代表不存在、1 代表存在。然后, 通过公式 (6) 计算各个像素的异常分数 $S \in \mathbb{R}^{W \times H}$:

$$s_{ij} = 1 - \frac{1}{G} \sum_{k=1}^G E_{ijk}, 1 \leq i \leq W, 1 \leq j \leq H \quad (6)$$

最后, 使用最近邻插值 (Nearest Neighbour Interpolation) 将 S 插值为原图大小 $S' \in \mathbb{R}^{W' \times H'}$, 作

为图像的像素级异常分数输出。同时, 取 S' 的平均值作为图像级异常分数输出。

由于图像深度特征的概率分布维度高、分布模式非常复杂, 本文方法难以对局部敏感哈希和布隆过滤器的误差给出理论边界。但理论推导显示, 适当提高局部敏感哈希组数 G 、提高布隆过滤器分配内存大小 M 可以降低误差率。

2 实验分析

2.1 实验设置

本文使用 MVTEC-AD 数据集^[12]对本文方法与对比方法进行测试。MVTEC-AD 数据集包括 15 个不同种类产品及其表面缺陷的子集, 包括正常产品的训练集和可能有缺陷的测试集。本文实验将 15 个子集合并为统一的训练集和测试集, 以接近实际应用场景中样本量大、产品与缺陷复杂的情况。合并后数据集包括了 3 629 张正常训练图像和 1 722 张可能含异常缺陷的测试图像。

实验选取近年来各研究思路下具有代表性的方法进行对比, 包括基于重构的 Efficient-AD^[4]、特征库检索的 PatchCore^[11]和 CFA^[13]、概率密度估计的 CFlow^[7]等。其中, PatchCore 方法不限制其使用的 ANNS 索引结构 (论文作者也实验了多种 ANNS 索引结构), 因此本文实验了两种主流的 ANNS 索引结构, HNSW 和 IVFPQ。实验全部使用论文作者提供的源代码进行训练和测试。其中, 基于特征库检索的 PatchCore 和 CFA 都使用 Wide ResNet-50 作为主干网络。

实验运行硬件条件包括 Nvidia RTX4090 GPU、32 核 2 GHz Intel Xeon Gold 6330 CPU、512 GB 内存。深度学习模型推理统一使用 Python 3.10 环境下 PyTorch2.1 与 CUDA 11.8。部分方法涉及 ANNS 索引构建与查询, 统一使用基于 MKL 线性代数库的 Faiss 索引库^[20]。本文方法的局部敏感哈希和布隆过滤器 (包括其中的 Murmurhash3) 均使用 Cupy 13.0 实现, 在 GPU 上完成全部运算。

实验对比指标包括准确率指标和计算性能指标两部分。本文方法和对比方法都会输出图像级和像素级的异常分数, 取值区间为 $[0, 1]$ 。图 4 所示为部分样本及相应的异常分数, 图 4 中, 如果选择阈值=5.0%, 那么图 4c、图 4d、图 4e 和图 4g 属于异常样本, 其他属于正常样本。在区间内均匀设置 100 个阈值后, 计算图像级和像素级异常分类的受试者工作特性曲线下面积 (Image/Pixel Area Under

Receiver Operating Characteristics, Image/Pixel AUROC) 为准确率指标, 如公式 (7) 所示。

$$AUROC = \sum_{i=2}^{100} \frac{(TPR_{i+1} + TPR_i) \cdot (FPR_i - FPR_{i-1})}{2} \quad (7)$$

其中, TPR_i 为使用第 i 个阈值时异常样本的分类真阳率 (True Positive Rate), FPR_i 为使用第 i 个阈值时异常样本的分类假阳率 (False Positive Rate)。此计算方式同时应用于图像级与像素级的异常分类结果。

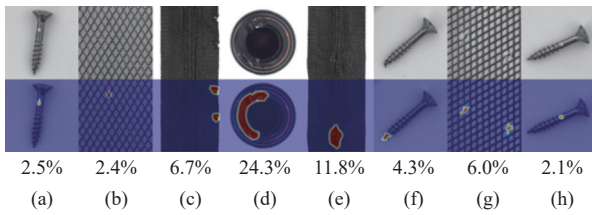


图 4 部分样本及其异常分数

表 1 本文方法与其他对比方法在 MVTEC-AD 数据集的准确率与性能

方法	Image AUROC	Pixel AUROC	时延/ms	吞吐量/FPS	显存峰值/GB	内存峰值/GB
CFA[13]	66.57%	89.83%	14.9	215	2.204	4.747
CFlow[7]	83.25%	93.22%	112.9	44	2.814	4.586
Efficient-AD[4]	54.68%	82.59%	8.4	325	2.146	4.616
PatchCore+HNSW[11]	89.82%	96.76%	176.9	12	18.606	20.135
PatchCore+IVFPQ[11]	91.19%	96.68%	108.3	11	21.188	21.304
本文方法	90.50%	95.29%	22.6	190	6.942	5.668

然而, 本文方法外的基于特征库检索的方法的计算速度有显著劣势, 时延均在 100 毫秒以上、吞吐量均在 50 以下。基于较复杂索引结构 HNSW 或 IVFPQ 的 PatchCore 方法, 内存和显存占用显著地高于端到端深度学习模型。而本文方法的时延和吞吐量较之则有显著提升, 总体水平与基于重建和基于概率密度模型的方法相近, 仅有显存占用相对较高。

2.3 效率分析

图 5 展示了使用 HNSW 的 PatchCore 方法与本文方法在推理阶段各个步骤的时延占比。其中, 特征提取步骤是指从 Wide ResNet-50 模型提取出深度特征, 异常估计步骤是指从 ANNS 或 AMQ 查询结果计算异常分数 (主要的时间消耗在插值和内存同步)。考虑到 GPU 计算的异步特性, 每个操作均在完全同步计算结果之后再计为结束。分析发现, 在 PatchCore 方法中, ANNS 搜索步骤消耗了 90% 以上的推理时间, 是明显的性能瓶颈。而本文

本文实验分别对模型推理时延、吞吐量进行测量, 测量时都只使用一张 GPU。其中模型推理时延为单张测试图像从加载图像至内存到计算出异常分数至内存的时间; 吞吐量为计算资源允许 (一般来说资源瓶颈为 GPU 显存) 下最大并发量除以这一并发量下的时延, 单位为帧每秒 (Frame Per Second, FPS)。另外, 本文实验也监控进程在单张图像推理时的内存和显存占用的峰值。

2.2 准确率与效率对比实验

表 1 展示了本文方法与其他对比方法在 MVTEC-AD 数据集 (合并子集) 的准确率与计算效率指标。实验发现, 包括 CFlow、PatchCore 和本文方法在内的基于特征库检索方法的准确率显著优于基于重建和基于概率密度模型的方法, 体现出基于特征库检索方法在更复杂的异常检测任务中更具有优势。本文方法的准确率指标相比 PatchCore 有约 1% 的损失, 但显著优于 CFlow 方法。

方法中, AMQ 查询时延占比则显著更低。各步骤中, Murmurhash3 哈希算法时延较长, 源于此哈希算法需要多次迭代进行位运算和整数运算。

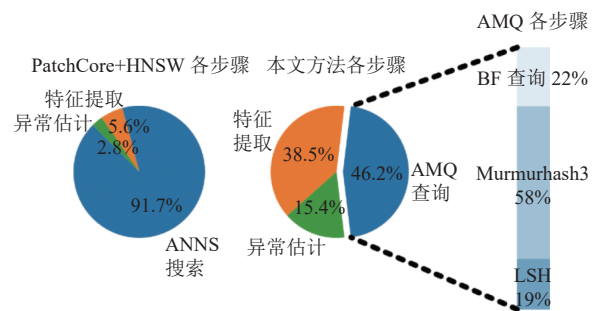


图 5 PatchCore 与本文方法各步骤计算时延占比对比

图 6 展示了使用 HNSW 的 PatchCore 方法与本文方法在推理阶段各个步骤的显存消耗占比。其中, HNSW 索引部分主要包含了 HNSW 构造的近邻图, HNSW 查询部分主要是指搜索时动态分配的显存。模型参数和模型推理分别指 Wide ResNet-

50 的模型参数与前馈运算时中间变量所占显存。分析发现, PatchCore 方法中主要的显存消耗来源于 HNSW, 特别是 HNSW 查询这一动态内存消耗会随着并发量提高而线性增加, 是阻碍吞吐量提升的瓶颈。而本文方法主要的显存占用来源于布隆过滤器 (BF), 但此部分显存占用是静态的、不会随着并发量提高而提升。LSH 和 MMH3 等动态的显存消耗相对占比较小。

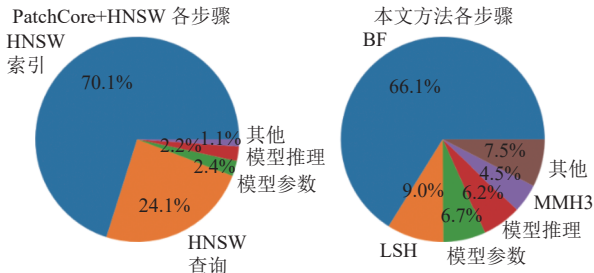


图6 PatchCore与本文方法各步骤显存消耗占比对比

2.4 超参分析

本文方法的可调节参数包括局部敏感哈希组数 G 和布隆过滤器空间比特数 M 。理论上, LSH 组数 G 越高, 估计越准确, 但计算量和对应部分内存/显存消耗会线性增大。布隆过滤器空间比特数 M 越高, 哈希冲突概率越低, 但计算量和内存/显存消耗会随之提升 (计算量提升是因为公式 (5) 所得 B 会提升)。

图7展示了将 LSH 组数 G 分别设置为 512、1 024、2 048、4 096 和布隆过滤器空间分别为 1、2、4、8、16 GB 时的准确率 (Image/Pixel AUROC) 与时延 (ms) 变化。实验发现, 随着 G 与 M 增大, 来自于 LSH 和 BF 的误差率降低, 准确率指标有相应提升。由于 GPU 高并行的特点, 时延的提升并非线性, 但时延结果基本符合理论预期趋势。

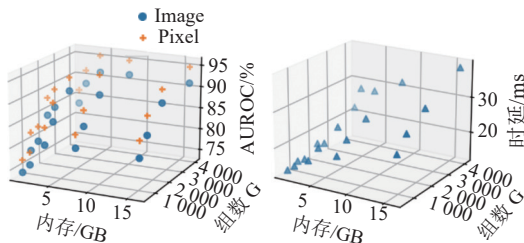


图7 不同局部敏感哈希组数 G 与布隆过滤器空间 M 设置对准确率 (Image/Pixel AUROC) 与时延的影响

3 结束语

本文提出了一种基于近似存在性查询的图像异

常检测无监督方法, 用局部敏感哈希与布隆过滤器对近邻样本存在性进行估计并计算异常分数, 替代现有的基于特征库检索的方法中计算量较大的近似最近邻搜索。实验结果显示, 本文方法能在基本保持准确率的同时, 显著提升算法运行速度与吞吐量, 适宜于要求低时延、高并发的无监督图像异常检测应用场景。

参考文献

- [1] 吕承侃, 沈飞, 张正涛, 等. 图像异常检测研究现状综述[J]. 自动化学报, 2022, 48(6): 1402-1428.
LV C K, SHEN F, ZHANG Z T, et al. Review of image anomaly detection[J]. Acta Automatica Sinica, 2022, 48(6): 1402-1428.
- [2] AKCAY S, ATAPOUR-ABARGHOUEI A, BRECKON T P. GANomaly: semi-supervised anomaly detection via adversarial training[C]//JAWAHAR C, LI H, MORI G, et al. Asian Conference on Computer Vision. Cham: Springer, 2019: 622-637.
- [3] DENG H Q, LI X Y. Anomaly detection via reverse distillation from one-class embedding[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2022: 9727-9736.
- [4] BATZNER K, HECKLER L, KÖNIG R. EfficientAD: Accurate visual anomaly detection at millisecond-level latencies[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. New York: IEEE, 2024: 127-137.
- [5] 张玥, 陈锡伟, 陈梦丹, 等. 基于对比学习生成对抗网络的无监督工业品表面异常检测[J]. 电子测量与仪器学报, 2023, 37(10): 193-201.
ZHANG Y, CHEN X W, CHEN M D, et al. Unsupervised surface anomaly detection of industrial products based on contrastive learning generative adversarial network[J]. Journal of Electronic Measurement and Instrumentation, 2023, 37(10): 193-201.
- [6] ADEY P, HAMILTON O, BORDEWICH M, et al. Region based anomaly detection with real-time training and analysis[C]//Proceedings of the 18th IEEE International Conference on Machine Learning and Applications. New York: IEEE, 2019: 495-499.
- [7] GUDOVSKIY D, ISHIZAKA S, KOZUKA K. CFLOW-AD: Real-time unsupervised anomaly detection with localization via conditional normalizing flows[C]//Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. New York: IEEE, 2022: 1819-1828.
- [8] TAILANIAN M, PARDO Á, MUSÉ P. U-flow: A U-shaped normalizing flow for anomaly detection with unsupervised threshold[EB/OL]. [2024-01-08]. <http://arxiv.org/abs/2211.12353>.
- [9] 张兰尧, 陈晓玲, 张达敏, 等. ValidFlow: 基于标准化流的无监督图像缺陷检测[J]. 数据采集与处理, 2023, 38(6): 1445-1457.

- ZHANG L Y, CHEN X L, ZHANG D M, et al. ValidFlow: Unsupervised image defect detection based on normalizing flows[J]. *Journal of Data Acquisition and Processing*, 2023, 38(6): 1445-1457.
- [10] DEFARD T, SETKOV A, LOESCH A, et al. PaDiM: A patch distribution modeling framework for anomaly detection and localization[C]//International Conference on Pattern Recognition. Cham: Springer, 2021: 475-489.
- [11] ROTH K, PEMULA L, ZEPEDA J, et al. Towards total recall in industrial anomaly detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2022: 14298-14308.
- [12] BERGMANN P, BATZNER K, FAUSER M, et al. The MVTEC anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection[J]. *International Journal of Computer Vision*, 2021, 129(4): 1038-1059.
- [13] LEE S, LEE S, SONG B C. CFA: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization[J]. *IEEE Access*, 2022, 10: 78446-78454.
- [14] CUI Y J, LIU Z X, LIAN S G. A survey on unsupervised anomaly detection algorithms for industrial images[J]. *IEEE Access*, 2023, 11: 55297-55315.
- [15] MALKOV Y A, YASHUNIN D A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, 42(4): 824-836.
- [16] BARANCHUK D, BABENKO A, MALKOV Y. Revisiting the inverted indices for billion-scale approximate nearest neighbors[C]//European Conference on Computer Vision. Cham: Springer, 2018: 209-224.
- [17] CHARIKAR M S. Similarity estimation techniques from rounding algorithms[C]//Proceedings of the thirty-fourth annual ACM symposium on Theory of computing. New York: ACM, 2002: 380-388.
- [18] BLOOM B H. Space/time trade-offs in hash coding with allowable errors[J]. *Communications of the ACM*, 1970, 13(7): 422-426.
- [19] DAHLGAARD S, KNUDSEN M B T, THORUP M. Practical hash functions for similarity estimation and dimensionality reduction[C]//Proceedings of the Proceedings of the 31st International Conference on Neural Information Processing Systems. New York: ACM, 2017: 6618-6628.
- [20] JOHNSON J, DOUZE M, JÉGOU H. Billion-scale similarity search with GPUs[J]. *IEEE Transactions on Big Data*, 2021, 7(3): 535-547.

编辑 张莉