计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(5):20−27 [doi: 10.15888/j.cnki.csa.009062]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于可分离卷积和注意力机制的晶圆缺陷检测①

付　强[1,2], 王红成[1]

[1](东莞理工学院 电子工程与智能化学院, 东莞 523808)
[2](东莞理工学院 计算机科学与技术学院, 东莞 523808)
通信作者: 王红成, E-mail: wanghc@dgut.edu.cn

摘　要: 为对半导体晶圆的表面缺陷进行快速检测, 提出一种基于深度可分离卷积和注意力机制的轻量级网络, 并在 WM-811K 数据集上进行了实验. 为解决该数据集中 9 种不同类别的缺陷比例相对不平衡问题, 采用了数据增强方法对较少数据的缺陷类别进行数据扩充. 本文模型中的深度可分离卷积可以降低模型的参数量, 提高模型的推理速度; 注意力机制可以使模型更加关注晶圆图像中有缺陷的区域, 使模型达到更好的分类效果. 实验表明, 所提方法在 WM-811K 数据集上的平均准确率高达 96.5%, 相对于 ANN、VGG16、MobileNetv2 等方法均有不同程度的提高, 并且参数量和运算量只是经典轻量级网络 MobileNetv2 的 73.5% 和 28.6%.

关键词: 深度可分离卷积; 缺陷检测; 注意力机制; 轻量级网络; 半导体晶圆; 深度学习; 残差网络

引用格式: 付强,王红成.基于可分离卷积和注意力机制的晶圆缺陷检测.计算机系统应用,2023,32(5):20–27. http://www.c-s-a.org.cn/1003-3254/9062.html

Wafer Defect Detection Based on Separable Convolution and Attention Mechanism

FU Qiang[1,2], WANG Hong-Cheng[1]

[1](School of Electrical Engineering and Intelligentization, Dongguan University of Technology, Dongguan 523808, China)
[2](School of Computer Science, Dongguan University of Technology, Dongguan 523808, China)

Abstract: A lightweight network based on depthwise separable convolution and the attention mechanism is proposed for fast detection of surface defects on semiconductor wafers, and experiments are conducted on the WM-811K dataset. As the proportions of defects of nine different categories in this dataset are imbalanced, a data enhancement method is used to expand the data for defect categories with few data. The depthwise separable convolution in this model can reduce the number of parameters and improve the inference speed of the model. The attention mechanism can make the model pay more attention to the defective regions in the wafer image so that the model can achieve better classification results. The experiments show that the average accuracy of the proposed method on the WM-811K dataset is as high as 96.5%, which is improved to varying degrees compared with that of ANN, VGG16, and MobileNetv2. In addition, the number of parameters and the amount of operation are only 73.5% and 28.6% of those of the classical lightweight network MobileNetv2, respectively.

Key words: depthwise separable convolution; defect detection; attention mechanism; lightweight networks; semiconductor wafers; deep learning; residual network

半导体晶圆生产需要经历拉单晶、切片、磨片、抛光、增层、光刻、掺杂、热处理、针测以及划片等生产工序, 不可避免在这些生产过程中出现表面缺陷. 因此, 半导体晶圆表面缺陷检测就变得非常重要. Wu

① 基金项目: 广东省普通高校重点科研平台和项目 (2020ZDZX3075); 东莞市科技特派员项目 (20201800500232)
收稿时间: 2022-10-01; 修改时间: 2022-11-04; 采用时间: 2022-11-16; csa 在线出版时间: 2023-02-10
CNKI 网络首发时间: 2023-02-13

等人[1]用真实的晶圆图像构建了一个公开晶圆数据集WM-811K, 他们提出了基于拉冬变换特征和支持向量机的方法实现对晶圆的缺陷进行分类. 在 WM-811K 上测试集的准确率为 94.63%. Yu 等人[2]提出了一种结合局部和非局部线性判别分析的方法来进行晶圆图缺陷检测与识别. Piao 等人[3]提出了一种基于拉东变换特征和决策树的晶圆缺陷识别方法, 在 WM-811K 上的识别准确率为 78.43%. Nakazawa 等人[4]提出了一种使用卷积神经网络的方法进行晶圆缺陷的分类, 它在合成晶圆图像上的准确率为 98.2%, 但在对 191 个真实的晶圆图像测试过程时各类正确率在 66.7% 和 100% 之间. Kim 等人[5]提出了一种基于神经网络的 bin 着色方法, 称为 Bin2Vec, 这种方法可以对晶圆图像进行更加方便的识别和确认缺陷. Saqlain 等人[6]提出了一种具有多种机器学习方法的投票集成分类器, 包括逻辑回归、随机森林、梯度提升机和人工神经网络, 它在 WM-811K 真实数据集的准确率为 95.86%. Ishida 等人[7]提出了一种基于 VGG 的分类网络, 在所提出的方法中, 提出了一种基于降噪的数据增强技术, 该技术很好地改善了数据集标签不平衡的问题. Nakazawa 等人[8]使用了一个降低权重的全卷积网络, 该网络基于编码器解码器架构, 可以检测和分割晶圆图上的缺陷图案. Kyeong 等人[9]提出了一种卷积神经网络模型, 用于对晶圆图像的单一缺陷或混合缺陷进行分类. 他们还开发了多个模型, 每个模型都分类一个特定的缺陷类别, 这种方法将花费大量的运算时间. Cheon 等人[10]提出了一种卷积神经网络模型, 可以从真实晶圆图像中提取特征, 并将输入数据准确分类为 5 个不同的晶圆缺陷类别. 该模型在将卷积神经网络模型与 K 最近邻算法相结合后, 还可以对未知缺陷的类别进行分类. Yu 等人[11]提出了一种基于增量学习的晶圆表面缺陷检测方法并在实验过程中采用 ResNet 作为骨干网络. Gómez-Sirvent 等人[12]提出了一种基于特征向量的方法, 它首先采用计算机视觉的技术将缺陷从背景中分离出来, 然后通过缺陷的形状、大小、纹理等信息来创建一个用于描述缺陷的特征向量, 最后通过支持向量机分类器来对特征向量进行评估. Chauhan 等人[13]提出了一种基于卷积神经网络的分类器, 对晶圆图像中两种不同的缺陷进行分类, 使用较小的训练数据取得了较高的检测精度. Wei 等人[14]针对混合型晶圆缺陷识别提出了一种新的多尺度信息融合和 Transformer 相结合的模型, 通过多尺度信息融合网络来关注晶圆的详细特征, 通过 Transformer 网络中的多头注意力机制对晶圆图像的全局上下文特征进行编码, 从而对晶圆图像和缺陷类型之间的关系进行建模, 最终实现对混合型晶圆缺陷的识别. Wei 等人[15]提出了一种基于多面动态卷积的混合型缺陷检测模型, 主要是采用了一个一维的自编码器来处理输入的信息并将其扩展到三维, 再通过通道合并机制来合并跨通道信息. 为了更好地解决这一复杂的实际问题, 本文提出了一种基于深度可分离卷积和注意力机制的端到端的轻量级缺陷检测模型, 它以更小的参数量和运算量达到较高的识别准确度.

1 数据集

1.1 WM-811K 数据集介绍

WM-811K 数据集是一个半导体晶圆数据集, 来源于真实环境, 它是目前世界上最大的可公开访问的晶圆图像数据集, 包括 811 457 张真实世界的晶圆图像. 这些图像是从 46 293 个批次中收集的, 每个批次包括 25 幅晶圆图像, 因此总共应该为 1 157 325 幅晶圆图像. 但是由于传感器故障和一些未知的原因, 造成某些批次的晶圆图像空缺, 因此最终只获得了 811 457 张晶圆图像. 该数据集除了包括晶圆图像的信息和缺陷类别的信息外, 还包括批次名称、晶圆尺寸、训练标签、测试标签等额外信息. 该数据集有共有 632 种不同尺寸的晶圆图像, 范围从 (6×21) 到 (300×202). WM-811K 的数据集虽然有 811 457 张晶圆图像, 但是有标签的图像只有 172 950 张. 带有标签的晶圆图像类别共有 9 种, 具体分布如表 1 所示. Center 缺陷类别的晶圆图像有 4 294 张, 占有标签图像的 2.5%; Donut 缺陷类别的晶圆图像有 555 张, 占有标签图像的 0.3%; Edge-Loc 缺陷类别的晶圆图像有 5 189 张, 占有标签图像的 3.0%; Edge-Ring 缺陷类别的晶圆图像有 9 680 张, 占有标签图像的 5.6%; Loc 缺陷类别的晶圆图像有 3 593 张, 占有标签图像的 2.1%; Random 缺陷类别的晶圆图像有 866 张, 占有标签图像的 0.5%; Scrath 缺陷类别的晶圆图像有 1 193 张, 占有标签图像的 0.7%; Near-full 缺陷类别的晶圆图像有 149 张, 占有标签图像的 0.1%; None 缺陷类别的晶圆图像有 147 431 张, 占有标签图像的 85.2%. 晶圆图像具体的缺陷类型如图 1 所示.

1.2 WM-811K 数据集预处理

由于在训练模型时, 必须把输入图像的尺寸调整为相同的大小以供模型使用. 但是 WM-811K 数据集的图像尺寸大小是不相等的, 因此需要先将图像调整为 (128×128), 以供模型使用. 原始的晶圆图像是单通道的, 每一个像素值都是一个状态变量, 共有 0、1、2 这 3 种状态: 0 表示未知状态, 1 表示正常状态, 2 表示缺陷状态. 然后对每个像素值采用独热的方式进行编码, 将单通道的图像扩展为三通道的图像. 由于缺陷类别标签是不平衡的, 因此对于独热编码后的图像数据进行数据增强. 本文实验中, 对于数据增强后的图像每个标签各取 10 000 张图像, 用做本次的实验数据. 然后对每类标签数据各取 80% 用于训练集, 10% 用于验证集, 10% 用于测试集.

表 1 WM-811K 数据集晶圆图像不同缺陷类别的数量分布

缺陷类型	Center	Donut	Edge-Loc	Edge-Ring	Local	Random	Scratch	Near-full	None
数量	4294	555	5189	9680	3593	866	1193	149	147431

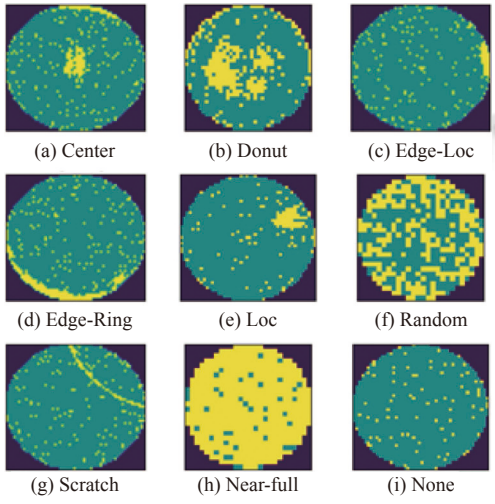


图 1 晶圆图像不同类型的缺陷

1.3 WM-811K 数据集数据增强

从表 1 可知, WM-811K 带有标签的图像的数量是非常不平衡的, 这将严重影响模型的训练结果, 使模型的准确率大幅度降低. 为了解决类别标签不平衡的问题, 我们对图像采用了随机旋转, 调整分辨率、对比度、饱和度, 增加随机噪声, 改变透明度等方法进行数据增强, 使数据各个类别的标签达到平衡, 提高模型分类的准确率.

2 模型介绍

当用神经网络解决图像分类任务时几乎都采用简单的卷积堆叠的方式进行模型设计, 这样的方式往往不能获得很高的准确率, 达不到实际生产和生活中的要求. 本文为了解决晶圆表面缺陷检测这一实际问题, 将深度可分离卷积[16] 和注意力机制[17] 相结合, 并在模型中运用了残差网络, 同时使用了 ReLU6 作为激活函数. 这些改进使模型在晶圆缺陷数据集上达到了很好的分类效果.

2.1 深度可分离卷积

深度可分离卷积是将普通的卷积操作分成了两个过程, 分别是逐通道卷积和逐点卷积.

逐通道卷积, 就是先对每个通道进行单独的卷积, 相当于在一个二维平面上进行卷积, 每个卷积核的通道数为 1, 此时卷积核的数量与上一层的通道数相同. 由于逐通道卷积的特性, 当卷积结束以后得到的输出数据与输入数据只是在宽高维度上发生了, 通道数仍然相等. 因为逐通道卷积只是在宽高的维度上面进行卷积, 每个通道都是独立运算的, 并没有利用通道间的信息, 因此需要逐点卷积对通道间的空间信息进行处理.

逐点卷积, 就是运用 1×1×C 的卷积核对所有通道间的信息进行整合处理, C 为上一层输出的通道数, 因此这里就是将逐通道得到的输出在通道的维度上面进行加权求和.

当进行完逐点卷积和逐通道卷积后, 输入数据的通道位置信息和空间位置信息都得到了有效的利用, 且模型实现了普通卷积核的功能. 但深度可分离卷积与普通卷积相比, 大大降低了参数量和运算量, 因此使用深度可分离卷积可以极大地减少模型运行的内存占用量, 并提高神经网络模型的推理速度.

2.2 通道注意力机制

通道注意力机制是一种采取权值的方式使其对特征层的通道信息更加关注的网络结构. 该模块的每一个通道都会在训练后得到一个对应的权值. 具体结构如图 2 所示.

通道注意力机制首先对输入的单个特征层进行全局平均池化, 得到 1×1×C1 的通道信息, C1 为特征层的

通道数. 然后对得到的通道信息做一个神经元较少的全连接, 再做一次和输入特征层通道数相等的全连接, 从而获得每个通道对应的权值. 通过对通道注意力机制的训练, 产生的权值就代表模型对不同通道的关注程度. 然后把训练得到的权值与输入特征层按照通道对应一一相乘, 从而获得更好的对于图像分类的有用信息, 从而提高模型的分类性能.

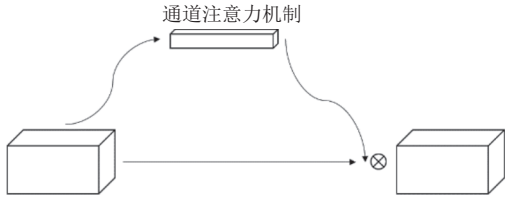


图 2 通道注意力机制示意图

2.3 ReLU6 激活函数

经过 ReLU 激活函数后的数值范围是 0 到正无穷, 这样就有可能造成权值之间相差过大, 从而降低模型的精度. ReLU6 激活函数是在 ReLU 函数的基础上进行了改进, 把经过激活函数的数值固定到 0 到 6 这个范围中. 该操作可以缓解 ReLU 造成权值范围相差过大的现象, 有效地防止了数值爆炸. 其表达式可表示为:

$$f(x)=\begin{cases}0, & x\leqslant 0\\ x, & 0<x<6\\ 6, & x\geqslant 6\end{cases} \tag{1}$$

2.4 模型结构

本文的数据集是一个具有 9 种缺陷类别的标记数据集, 每种类别根据缺陷的不同特征进行区分. 为了解决这个缺陷分类问题, 本文提出了一种端到端的半导体晶圆缺陷检测模型. 该模型主要运用了深度可分离卷积和通道注意力机制, 并配合残差网络的思想, 对部分网络层进行残差连接, 模型结构如图 3 所示. 首先, 模型对输入的图像进行特征提取, 然后把提取到的特征输入到通道注意力机制模块, 使模型更加关注图像中有缺陷的区域, 再把从通道注意力机制输出的信息输入特征整合模块和全连接层, 最后输入图像的缺陷类别. 特征提取和特征整合模块如图 4(a) 和图 4(b) 所示.

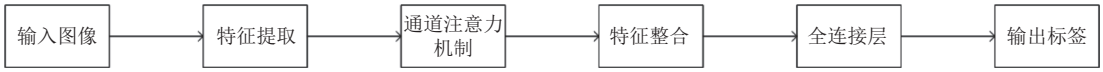
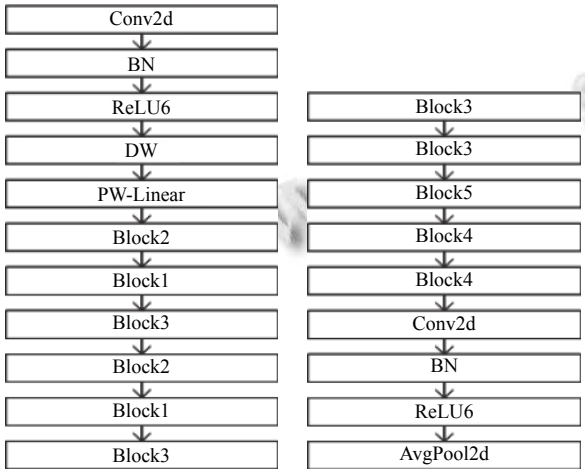


图 3 模型整体结构图



(a) 特征提取模块 (b) 特征整合模块

图 4 模型核心模块结构图

图 4 中各主要模块如表 2 所示. Block1 和 Block2 都采用双分支结构, 一个分支进行深度可分离卷积, 另外一个分支进行普通的卷积, 然后对两个分支的卷积结果进行特征融合. 这里需要注意的是, 为了保证两个分支卷积后的输出特征能够进行特征融合, 两个分支都必须要进行相同的下采样. Block3 借鉴了 ResNet 的残差结构以加快模型的优化速度, 其中跳跃连接有效防止模型增加深度带来梯度消失问题的发生. Block4 和 Block5 只采用了深度可分离卷积并进行了不同程度的下采样, 使用这种结构可以大大降低模型的参数量, 加快模型的训练速度和推理速度.

模型整体结构和参数如表 3 所示. 首先对输入图像进行卷积, 增加数据的通道数, 并进行下采样. 然后通过深度可分离卷积的 Block1、Block2 和 Block3 分别对得到的数据进行特征提取, 使用通道注意力机制获取对模型训练更有利的特征. 接下来使用 Block3、Block4、Block5 对数据进行处理, 最后使用全局平均池化和全连接层输出各个类别的预测分数. 模型全部采用了 ReLU6 激活函数.

表 2 模型的主要模块

模块	参数
Block1	Conv1×1 (stride=1), BN, ReLU6, Conv3×3 (stride=1), BN, ReLU6, Conv1×1 (stride=1), BN Conv1×1 (stride=1), BN, ReLU6
Block2	Conv1×1 (stride=2), BN, ReLU6, Conv3×3 (stride=2), BN, ReLU6, Conv1×1 (stride=2), BN Conv1×1 (stride=2), BN, ReLU6
Block3	Conv1×1 (stride=1), BN, ReLU6 —
Block4	Conv1×1 (stride=1), BN, ReLU6, Conv3×3 (stride=1), BN, ReLU6, Conv1×1 (stride=1), BN
Block5	Conv1×1 (stride=2), BN, ReLU6, Conv3×3 (stride=2), BN, ReLU6, Conv1×1 (stride=2), BN

表 3 提议的深度模型参数

Layer	Input size	Input channel	Filter size	Filter num
Conv2d	128×128	3	3×3×3	32
BN	64×64	32	—	—
ReLU6	64×64	32	—	—
DW	64×64	32	3×3×32	32
PW-Linear	64×64	32	1×1×32	16
Block2	64×64	16	—	—
Block1	32×32	24	—	—
Block3	32×32	24	—	—
Block2	32×32	24	—	—
Block1	16×16	32	—	—
Block3	16×16	32	—	—
SE-Attention	16×16	32	—	—
Block3	16×16	32	—	—
Block3	16×16	32	—	—
Block5	16×16	32	—	—
Block4	8×8	64	—	—
Block4	8×8	128	—	—
Conv2d	8×8	256	1×1×256	512
BN	8×8	512	—	—
ReLU6	8×8	512	—	—
AvgPool2d	8×8	512	8×8×512	1
FC	1×1	512	1×1×512	9

3 实验及结果

3.1 实验过程

本次实验所用的数据集共有 9 个类别, 每个类别 10 000 张晶圆图像, 共计 90 000 张. 然后把数据集按照 8:1:1 的比例划分成训练集、验证集和测试集, 分别在 VGG16[18]、MobileNetv2[19]、ANN、SVM[20] 和本文提出的模型上面进行训练、验证和测试. 在模型训练阶段, 图像的输入大小为 3×128×128, 学习率为 1E−4, 批量大小为 32.

本文对每个模型训练 20 个 epoch, 前 10 个 epoch 使用 Adam 梯度优化器, 后 10 个 epoch 的训练使用 SGD 梯度优化器. 这是由于 Adam 优化器的特性可以使模型快速收敛, SGD 优化器可以对模型的参数进行微调, 提高模型的准确率. 如表 4 所示, 在实验中证明, 先使用 Adam 梯度优化器, 再使用 SGD 梯度优化器这种优化方案是正确的, 它既可以节省模型的训练时间, 也可以提高模型的准确率. 在使用 Adam 优化器进行训练时, 每一个 epoch 结束后得到的模型都使用验证集进行验证, 选择在验证集上准确率最高的模型进行后 SGD 优化器的训练. 同样在使用 SGD 优化器的训练中, 每一个 epoch 结束后也使用验证集验证. 最终选取在验证集上准确率最高的模型作为训练的结果. 在得到最优模型后, 使用测试集对模型的性能进行测试, 以获得准确率、精准率、召回率和 $F1$ 分数等图像分类的指标, 以分析模型的性能.

表 4 使用不同梯度优化器在验证集上精度的变化 (%)

模型	*Max Accuracy* (Adam)	*Max Accuracy* (SGD)
SVM	—	—
ANN	86.1	88.8
VGG16	91.4	92.8
MobileNetv2	94.6	95.8
Ours	**95.7**	**96.5**

本文采用交叉熵损失作为损失函数, 其表达式为:

$$loss_{\text{cross}} = \frac{-1}{C \times N} \sum_{i}^{C} \sum_{j}^{N} (y_{i,j} \times \log(p_{i,j})) \tag{2}$$

其中, $y_{i,j}$是该图像类别的真实标签, $p_{i,j}$是该图像的预测标签, N是单次迭代的样本数量, C是类别的数量, 当预测值和真实值之间的差异越大时, 交叉熵损失值也就越大.

实验使用的深度学习框架为 PyTorch 1.11, 解释器为 Python 3.9. 该实验是在个人笔记本电脑上进行的, 操作系统为 Windows 10, CPU 为 Intel(R) Core(TM) i5-7300HQ, 内存 8 GB, GPU 为 NVIDIA GTX1050Ti, 显存为 4 GB.

3.2 模型评价指标

在图像分类任务中, 常见的评价指标为准确率 (*Accuracy*)、精确率 (*Precision*)、召回率 (*Recall*) 和

*F*1 分数 (*F*1-*score*), 其表达式分别为:

$$Accuracy = \frac{TN+TP}{FP+TN+TP+FN} \tag{3}$$

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

$$F1\text{-}score = \frac{2\times Precision\times Recall}{Precision+Recall} \tag{6}$$

其中, TP是预测为正样本实际为正样本的个数; FP是预测为正样本实际为负样本的个数; TN是预测为负样本实际为负样本的个数; FN是预测为负样本实际为正样本的个数. 显然, $F1$分数为准确率和召回率的一种加权平均, 其值越大意味着模型越好.

3.3 实验结果分析

3.3.1 对比实验

将本文方法与 VGG16、MobileNetv2、SVM、ANN 等方法进行对比, 发现本文的方法在参数量、运行速度、准确率上都具有一定的优势. 从表 5 中可以看出, 除了 SVM 方法不能量化参数外, 在其余的 4 个模型中, 本文方法的参数量最低的, 但是却获得了最高的准确率, 与经典的轻量级网络 MobileNetv2 相比, 参数量只是它的 73.5%, 但是准确率却比它高了 0.7%. 在参数量方面, 本文方法的参数量也是最少的, 它只是 VGG16 计算量的 0.48%, 是 ANN 计算量的 0.62%, 是轻量级网络 MobileNetv2 计算量的 28.6%. 本文的方法在计算量和参数量都处于最少的情况下, 却表现出最高的准确率, 这意味着本文的方法在同等的硬件条件下更有优势.

表 5 各个模型参数量、计算量、准确率对比

模型	Total params	FLOPs (M)	*Accuracy* (%)
SVM	—	—	70.4
ANN	102 762 752	102.78	88.7
VGG16	5 130 727 424	134.30	92.8
MobileNetv2	102 187 264	2.24	95.8
Ours	**75 068 032**	**0.64**	**96.5**

表 6 展示了所有模型的性能对比, 结果表明, 本文的方法除了测试集的正确率比 VGG16 和 MobileNetv2 低 0.1% 外, 在验证集、测试集、查准率、查全率、*F*1 分数这几个性能上均获得了最高值, 分别为 96.5%, 96.5%, 96.4%, 96.5% 和 96.5%. 对于实际应用来说, 模型在测试集上的正确率才是最具有价值的性能指标, 因为测试集是模型训练时没有用到的数据, 与模型的训练过程完全无关. 在测试集这一性能中, 仍然是本文提出的方法达到了更高的准确率. 虽然 MobileNetv2 在验证集上的正确率也高达 95.8%, 只比本文提出的方法低了 0.7%, 但是本文方法的参数量和计算量更低, 所以综合来看本文提出的方法更具有优势.

表 6 不同模型的准确率、查准率、查全率、*F*1 分数对比 (%)

模型	*Training Accuracy*	*Validation Accuracy*	*Testing Accuracy*	*Precision*	*Recall*	*F1-score*
VGG16	**99.9**	92.8	92.7	92.7	92.7	92.6
MobileNetv2	**99.9**	95.8	95.8	95.8	95.8	95.8
SVM	71.5	70.4	70.4	72.8	70.4	67.6
ANN	93.9	88.8	88.7	88.7	88.7	88.7
Ours	99.8	**96.5**	**96.5**	**96.4**	**96.5**	**96.5**

表 7 展示了不同模型对晶圆缺陷分类的精度对比, 从表 7 中可以清晰看出, 本文的模型在对于 Center、Donut、Random、Scratch 和 Near-full 类进行分类时, 准确率分别为 97.9%, 100.0%, 96.7%, 100.0%, 99.1% 和 96.7%, 都高达 96.0% 以上. 虽然本文的方法对 Edge-Loc、Local、None 类别的正确率只有 91.5%, 92.6% 和 94.0%, 但却是这几类模型中对该类缺陷类别分类正确率最高的. 但是对于 Edge-Ring 类别缺陷来说, VGG16 的分辨率却获得了最高的正确率, 高达 97.9%, 说明对于某一类缺陷来说, 某一模型可能会有更好的适应性. 从表 7 中还可以看出 VGG16 对于 Local 类别缺陷的正确分辨率只有 74.9%, 与 MobileNetv2 和本文的模型相比分别低了 14.8% 和 17.7%. SVM 分类器在 Edge-Loc、Scratch、Near-full、None 类别上的正确分辨率只有 54.3%、64.7% 和 45.1%. 在 ANN 模型中, Edge-Loc、Local、None 的正确分辨率也不是很高, 分别只有 77.2%、66.4% 和 79.2%. 本文的模型与 MobileNetv2 相比, 本文的方法虽然只有 Edge-Loc 和 Near-full 这两类的准确率高于 MobileNetv2 模型 1.0% 和 2.1%, 但是仍然可以说明本文的方法具有比 MobileNetv2 模型更高的准确率.

为了能够更直观地看到本文的方法在测试集上各个类别的分类情况, 本文绘制了混淆矩阵, 如表 8 所示. 其中, 横轴代表预测标签, 纵轴代表真实标签. 每一行

表示某类真实标签的晶圆图像被正确预测的类别数量和错误预测的类别数量, 以及被错误预测成某类类别的数量. 每一列表示哪些类别被预测成为该类缺陷类别, 哪些被错误的预测, 以及错误预测的数量. 并且从主对角线可以直观地看出每一个类被成功预测的数量. 从混淆矩阵中可以看出, 模型对 Donut 和 Random 类的正确分辨数量达到了 100.0%, 对 Scratch 类的正确分辨数量达到 99.1%, 都具有较高的正确率, 说明模型对这 3 类有很强的分辨能力. 但是在 Edge-Loc 类上的正确的分辨数量只有 91.5%, Local 和 None 类的正确分辨数量分别也只有 92.6% 和 94.0%. 在 Edge-Loc 类中, 被错误分类成 Local 和 None 类的数量竟然高达 2.3% 和 2.6%. 由于缺陷类别形状的原因, 给模型对这些类别的分辨造成了一定的困难, 又有一些晶圆图像上面有多种缺陷, 增加了模型识别的难度, 将一些类别与其他类别进行分类, 从而造成分类错误. 在 Center、Edge-Ring 和 Near-full 的正确分辨数量分别为 97.9%, 96.7% 和 96.7%.

表 7 不同模型在各个缺陷类别上的正确率 (%)

模型	Center	Donut	Edge-Loc	Edge-Ring	Local	Random	Scratch	Near-full	None
VGG16	96.0	99.2	85.6	**97.9**	74.9	**100.0**	95.3	93.7	92.0
MobileNetv2	**97.9**	**100.0**	90.5	97.0	89.7	**100.0**	**99.2**	94.6	**94.0**
SVM	92.7	95.2	54.3	97.1	11.3	**100.0**	64.7	73.0	45.1
ANN	93.7	99.9	77.2	96.7	66.4	**100.0**	98.6	86.8	79.2
Ours	**97.9**	**100.0**	**91.5**	96.7	**92.6**	**100.0**	99.1	**96.7**	**94.0**

表 8 本文模型的混淆矩阵 (%)

Predicted label	Center	Donut	Edge-Loc	Edge-Ring	Local	Random	Scratch	Near-full	None
Center	97.9	0.2	0.0	0.0	1.2	0.0	0.1	0.1	0.5
Donut	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Edge-Loc	0.9	0.1	91.5	1.5	2.3	0.1	0.2	0.8	2.6
Edge-Ring	0.0	0.0	2.9	96.7	0.0	0.0	0.0	0.0	0.4
Local	1.5	0.9	2.4	0.0	92.6	0.0	0.0	1.4	1.2
Random	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
Scratch	0.2	0.1	0.3	0.0	0.2	0.1	99.1	0.0	0.0
Near-full	0.0	0.0	1.3	0.1	1.6	0.0	0.0	96.7	0.3
None	0.9	0.0	1.6	0.0	1.4	0.0	0.2	1.9	94.0

3.3.2 消融实验

在表 9 中, no_se_model 表示在本文的方法上去掉注意力机制的模型, no_dpw_model 表示在本文的方法上不使用深度可分离卷积的模型. 从表 9 中可以看出, 当不使用深度可分离卷积时, 虽然也能达到和使用深度可分离卷积的模型几乎相等的准确率, 但是参数量确实使用深度可分离卷积的 12.3 倍, 运算量是深度可分离卷积的 14.9 倍. 虽然使用注意力机制使模型的参数量增加了 2.1%, 但是在运算量等同的条件下, 模型的准确率却提高了 2.3%. 在实际生产过程中, 这种提升是非常有必要的.

表 9 是否使用深度可分离卷积和注意力机制对比

模型	Total params	FLOPs (M)	*Accuracy* (%)
no_se_model	**73 495 168**	**0.64**	94.2
no_dpw_model	925 954 560	9.56	96.3
Ours	75 068 032	**0.64**	**96.5**

3.3.3 模型的收敛性分析

图 5 给出了所提模型在训练集和验证集上的准确率变化曲线. 可以看出, 模型在训练集和验证集上的准确率都呈现先上升后平稳的状态. 在训练集中, 模型的准确率在第 14 个 epoch 中达到最大值 99.8%. 在验证集中, 模型的准确率在第 16 个 epoch 中达到最大值 96.5%. 因此可以看出, 模型具有较强的收敛性, 可以通过较少的 epoch 迅速达到较高的准确率, 并保持相对稳定.

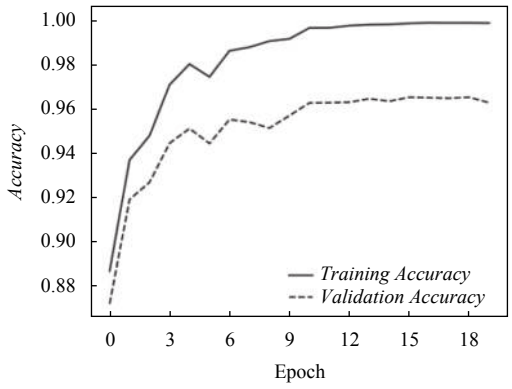


图 5 模型在训练集和验证集上准确率变化曲线图

4 结语

结合深度可分离卷积和注意力机制, 本文提出了一种端到端的轻量化半导体晶圆缺陷检测模型. 其中深度可分离卷积模块大大降低模型的参数量和运算量, 提高了模型的运算速度. 而注意力机制模块则提高了模型对缺陷分类的精度. 该方法在本次实验所运行的数据集上获得了不错的效果, 与其他模型相比大大降低了参数量和运算量, 但仍然保持较高的分类准确度.

参考文献

1 Wu MJ, Jang JSR, Chen JL. Wafer map failure pattern recognition and similarity ranking for large-scale data sets. IEEE Transactions on Semiconductor Manufacturing, 2015, 28(1): 1–12. [doi: 10.1109/TSM.2014.2364237]

2 Yu JB, Lu XL. Wafer map defect detection and recognition using joint local and nonlocal linear discriminant analysis. IEEE Transactions on Semiconductor Manufacturing, 2016, 29(1): 33–43. [doi: 10.1109/TSM.2015.2497264]

3 Piao MH, Jin CH, Lee JY, *et al*. Decision tree ensemble-based wafer map failure pattern recognition based on Radon transform-based features. IEEE Transactions on Semiconductor Manufacturing, 2018, 31(2): 250–257. [doi: 10.1109/TSM.2018.2806931]

4 Nakazawa T, Kulkarni DV. Wafer map defect pattern classification and image retrieval using convolutional neural network. IEEE Transactions on Semiconductor Manufacturing, 2018, 31(2): 309–314. [doi: 10.1109/TSM.2018.2795466]

5 Kim J, Kim H, Park J, *et al*. Bin2Vec: A better wafer bin map coloring scheme for comprehensible visualization and effective bad wafer classification. Applied Sciences, 2019, 9(3): 597. [doi: 10.3390/app9030597]

6 Saqlain M, Jargalsaikhan B, Lee JY. A voting ensemble classifier for wafer map defect patterns identification in semiconductor manufacturing. IEEE Transactions on Semiconductor Manufacturing, 2019, 32(2): 171–182. [doi: 10.1109/TSM.2019.2904306]

7 Ishida T, Nitta I, Fukuda D, *et al*. Deep learning-based wafer-map failure pattern recognition framework. Proceedings of the 20th International Symposium on Quality Electronic Design (ISQED). Santa Clara: IEEE, 2019. 291–297.

8 Nakazawa T, Kulkarni DV. Anomaly detection and segmentation for wafer defect patterns using deep convolutional encoder-decoder neural network architectures in semiconductor manufacturing. IEEE Transactions on Semiconductor Manufacturing, 2019, 32(2): 250–256. [doi: 10.1109/TSM.2019.2897690]

9 Kyeong K, Kim H. Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks. IEEE Transactions on Semiconductor Manufacturing, 2018, 31(3): 395–402. [doi: 10.1109/TSM.2018.2841416]

10 Cheon S, Lee H, Kim CO, *et al*. Convolutional neural network for wafer surface defect classification and the detection of unknown defect class. IEEE Transactions on Semiconductor Manufacturing, 2019, 32(2): 163–170. [doi: 10.1109/TSM.2019.2902657]

11 Yu NG, Li X, Xu Q, *et al*. Research on wafer surface defect pattern detection method based on incremental learning. Journal of Physics: Conference Series, 2021, 2078: 012046. [doi: 10.1088/1742-6596/2078/1/012046]

12 Gómez-Sirvent JL, de la Rosa FL, Sánchez-Reolid R, *et al*. Optimal feature selection for defect classification in semiconductor wafers. IEEE Transactions on Semiconductor Manufacturing, 2022, 35(2): 324–331. [doi: 10.1109/TSM.2022.3146849]

13 Chauhan KK, Joshi G, Kaur M, *et al*. Semiconductor wafer defect classification using convolution neural network: A binary case. IOP Conference Series: Materials Science and Engineering, 2022, 1225: 012060. [doi: 10.1088/1757-899X/1225/1/012060]

14 Wei YX, Wang H. Mixed-type wafer defect recognition with multi-scale information fusion transformer. IEEE Transactions on Semiconductor Manufacturing, 2022, 35(2): 341–352. [doi: 10.1109/TSM.2022.3156583]

15 Wei YX, Wang H. Mixed-type wafer defect pattern recognition framework based on multifaceted dynamic convolution. IEEE Transactions on Instrumentation and Measurement, 2022, 71: 2511211.

16 Sifre L, Mallat S. Rigid-motion scattering for texture classification. arXiv:1403.1687, 2014.

17 Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 7132–7141.

18 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv: 1409.1556, 2014.

19 Sandler M, Howard A, Zhu ML, *et al*. MobileNetv2: Inverted residuals and linear bottlenecks. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 4510–4520.

20 Cortes C, Vapnik V. Support-vector networks. Machine Learning, 1995, 20(3): 273–297.

(校对责编: 孙君艳)