



(12) 发明专利

(10) 授权公告号 CN 109661671 B

(45) 授权公告日 2023. 10. 10

(21) 申请号 201780044333.6
 (22) 申请日 2017.06.06
 (65) 同一申请的已公布的文献号
 申请公布号 CN 109661671 A
 (43) 申请公布日 2019.04.19
 (30) 优先权数据
 62/346,065 2016.06.06 US
 (85) PCT国际申请进入国家阶段日
 2019.01.17
 (86) PCT国际申请的申请数据
 PCT/US2017/036174 2017.06.06
 (87) PCT国际申请的公布数据
 W02017/214164 EN 2017.12.14
 (73) 专利权人 莫维迪乌斯有限公司
 地址 荷兰北荷兰省
 (72) 发明人 D·莫洛尼 A·德加尼
 (74) 专利代理机构 永新专利商标代理有限公司
 72002
 专利代理师 张立达 王英

(51) Int.Cl.
 G06V 10/20 (2022.01)
 G06V 10/28 (2022.01)
 G06V 10/44 (2022.01)
 G06V 10/50 (2022.01)
 G06V 10/764 (2022.01)
 (56) 对比文件
 CN 102447901 A, 2012.05.09
 US 2013129233 A1, 2013.05.23
 US 2003012439 A1, 2003.01.16
 STELLA ATKINS* 等. Fully Automatic Segmentation of the Brain in MRI. 《IEEE TRANSACTIONS ON MEDICAL IMAGING》. 1998,
 Navneet Dalal 等. Histograms of Oriented Gradients for Human Detection Navneet Dalal and Bill Triggs. 《IEEE Computer Society Conference on Computer Vision and Pattern Recognition》. 2005,

审查员 康瑞丽

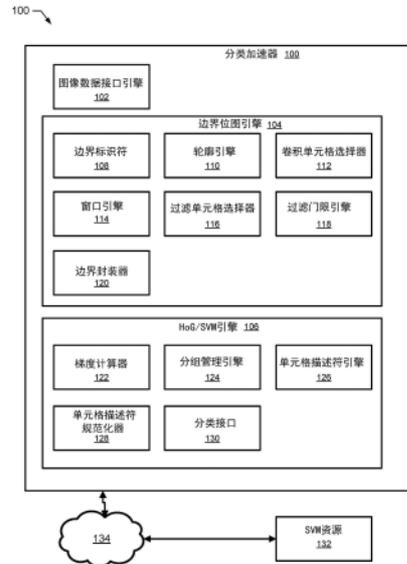
权利要求书4页 说明书17页 附图14页

(54) 发明名称

使用边界位图对图像分类的改善

(57) 摘要

本文公开了用于用边界位图来改善图像分类的方法、系统、装置和制品。示例性公开的装置包括：轮廓引擎(110)，其用于：识别所述图像内的前景轮廓；基于所述前景轮廓的边界来生成边界框；以及生成识别前景的单元格和背景的单元格的编码的轮廓矩阵；卷积单元格选择器(112)，其用于对所述编码的轮廓矩阵进行卷积以生成卷积的位图矩阵；以及过滤单元格选择器(116)，其用于通过以下操作识别所述卷积的位图矩阵的合格块来改善图像分类效率：保留满足单元格保留门限的所述卷积的位图矩阵的第一相应单元格；以及去除不满足所述单元格保留门限的所述卷积的位图矩阵的第二相应单元格。



CN 109661671 B

1. 一种用于分析图像的装置,所述装置包括:
 - 轮廓引擎,其用于:
 - 识别所述图像内的前景轮廓;
 - 基于所述前景轮廓的边界来生成边界框;以及
 - 生成识别前景的单元格和背景的单元格的编码的轮廓矩阵;
 - 卷积单元格选择器,其用于:
 - 对所述编码的轮廓矩阵进行卷积以生成卷积的位图矩阵;以及
 - 用与关联于全一矩阵内核的前景单元格计数相对应的值来对所述卷积的位图矩阵的相应单元格进行编码;以及
 - 过滤单元格选择器,其用于通过以下操作识别所述卷积的位图矩阵的合格块来改善图像分类效率:
 - 保留具有所述前景单元格计数的所述卷积的位图矩阵的满足单元格保留门限的第一相应单元格;以及
 - 去除具有所述前景单元格计数的所述卷积的位图矩阵的不满足所述单元格保留门限的第二相应单元格。
2. 根据权利要求1所述的装置,其中,所述过滤单元格选择器用于通过以下操作来生成贡献块矩阵:用保留指示符来对所述贡献块矩阵的所述第一相应单元格进行编码,以及用丢弃指示符来对所述贡献块矩阵的所述第二相应单元格进行编码。
3. 根据权利要求2所述的装置,还包括边界封装器,其用于识别所述保留指示符中的相应保留指示符的外周边,以生成合格块矩阵。
4. 根据权利要求3所述的装置,还包括分类接口,其用于用所述合格块矩阵来掩蔽检测窗口,所述掩蔽用于减少要在图像分类中处理的块的数量。
5. 根据权利要求1所述的装置,还包括分类接口,其用于将所述卷积的位图矩阵的所保留的第一相应单元格提供给定向梯度分类器的直方图。
6. 根据权利要求1所述的装置,还包括窗口引擎,其用于:
 - 将所述全一矩阵内核卷积到所编码的轮廓的相应单元格;以及
 - 通过计算用于所编码的轮廓的所述相应单元格的所述全一矩阵内核内的前景单元格的总和来确定所述前景单元格计数。
7. 根据权利要求1所述的装置,其中,所述单元格保留门限包括基于矩阵内核大小或重叠步长中的至少一个的值。
8. 根据权利要求2、6或7中任一项所述的装置,还包括分组管理引擎,其用于分配所述图像的定向角以形成图像幅度的直方图。
9. 根据权利要求2、5、6或7中任一项所述的装置,还包括单元格描述符引擎,其用于将图像单元格的直方图聚合到所述图像的相应块。
10. 根据权利要求2、5、6或7中任一项所述的装置,还包括边界封装器,其用于识别所述图像的边界位图区域,所述边界位图区域指示要参与图像分类的减少数量的图像单元格。
11. 一种用于分析图像的方法,所述方法包括:
 - 通过用处理器来执行指令,识别所述图像内的前景轮廓;
 - 通过用所述处理器来执行指令,基于所述前景轮廓的边界来生成边界框;

通过用所述处理器来执行指令,生成用于识别前景的单元格和背景的单元格的编码的轮廓矩阵;

通过用所述处理器来执行指令,对所述编码的轮廓矩阵进行卷积以生成卷积的位图矩阵;

通过用所述处理器来执行指令,用与关联于全一矩阵内核的前景单元格计数相对应的值来对所述卷积的位图矩阵的相应单元格进行编码;以及

通过用所述处理器来执行指令,通过以下操作识别所述卷积的位图矩阵的合格块来改善图像分类效率:(a)保留具有所述前景单元格计数的所述卷积的位图矩阵的满足单元格保留门限的第一相应单元格,以及(b)去除具有所述前景单元格计数的所述卷积的位图矩阵的不满足所述单元格保留门限的第二相应单元格。

12.根据权利要求11所述的方法,还包括通过以下操作来生成贡献块矩阵:用保留指示符来对所述贡献块矩阵的所述第一相应单元格进行编码,以及用丢弃指示符来对所述贡献块矩阵的所述第二相应单元格进行编码。

13.根据权利要求12所述的方法,还包括:识别所述保留指示符中的相应保留指示符的外周边,以生成合格块矩阵。

14.根据权利要求13所述的方法,还包括:用所述合格块矩阵来掩蔽检测窗口,所述掩蔽用于减少要在图像分类中处理的块的数量。

15.根据权利要求11所述的方法,还包括:将所述卷积的位图矩阵的所保留的第一相应单元格提供给定向梯度分类器的直方图。

16.根据权利要求11所述的方法,还包括:

将所述全一矩阵内核卷积到所编码的轮廓的相应单元格;以及

通过计算用于所编码的轮廓的所述相应单元格的所述全一矩阵内核内的前景单元格的总和来确定所述前景单元格计数。

17.根据权利要求11所述的方法,还包括:基于矩阵内核大小或重叠步长中的至少一个来设置所述单元格保留门限的值。

18.根据权利要求12、15、16或17中任一项所述的方法,还包括分配所述图像的定向角以形成图像幅度的直方图。

19.根据权利要求12、15、16或17中任一项所述的方法,还包括将图像单元格的直方图聚合到所述图像的相应块。

20.根据权利要求12、15、16或17中任一项所述的方法,还包括:识别所述图像的边界位图区域,所述边界位图区域指示要参与图像分类的减少数量的图像单元格。

21.一种包括指令的有形计算机可读介质,所述指令在被执行时使得处理器至少执行以下操作:

识别图像内的前景轮廓;

基于所述前景轮廓的边界来生成边界框;

生成用于识别前景的单元格和背景的单元格的编码的轮廓矩阵;

对所述编码的轮廓矩阵进行卷积以生成卷积的位图矩阵;

用与关联于全一矩阵内核的前景单元格计数相对应的值来对所述卷积的位图矩阵的相应单元格进行编码;以及

通过以下操作识别所述卷积的位图矩阵的合格块来改善图像分类效率：(a) 保留具有所述前景单元格计数的所述卷积的位图矩阵的满足单元格保留门限的第一相应单元格，以及(b) 去除具有所述前景单元格计数的所述卷积的位图矩阵的不满足所述单元格保留门限的第二相应单元格。

22. 根据权利要求21所述的计算机可读介质，其中，所述指令在被执行时还使得所述处理器通过以下操作来生成贡献块矩阵：用保留指示符来对所述贡献块矩阵的所述第一相应单元格进行编码，以及用丢弃指示符来对所述贡献块矩阵的所述第二相应单元格进行编码。

23. 根据权利要求22所述的计算机可读介质，其中，所述指令在被执行时还使得所述处理器识别所述保留指示符中的相应保留指示符的外周边，以生成合格块矩阵。

24. 根据权利要求23所述的计算机可读介质，其中，所述指令在被执行时还使得所述处理器用所述合格块矩阵来掩蔽检测窗口，所述掩蔽用于减少要在图像分类中处理的块的数量。

25. 根据权利要求21所述的计算机可读介质，其中，所述指令在被执行时还使得所述处理器将所述卷积的位图矩阵的所保留的第一相应单元格提供给定向梯度分类器的直方图。

26. 根据权利要求21所述的计算机可读介质，其中，所述指令在被执行时还使得所述处理器执行以下操作：

将所述全一矩阵内核卷积到所编码的轮廓的相应单元格；以及

通过计算用于所编码的轮廓的所述相应单元格的所述全一矩阵内核内的前景单元格的总和来确定所述前景单元格计数。

27. 根据权利要求21所述的计算机可读介质，其中，所述指令在被执行时还使得所述处理器基于矩阵内核大小或重叠步长中的至少一个来设置所述单元格保留门限的值。

28. 一种用于分析图像的系统，所述系统包括：

用于识别所述图像内的前景轮廓的单元；

用于基于所述前景轮廓的边界来生成边界框的单元；

用于生成用于识别前景的单元格的编码的轮廓矩阵的单元；

用于对所述编码的轮廓矩阵进行卷积以生成卷积的位图矩阵的单元，该卷积单元用于用与关联于全一矩阵内核的前景单元格计数相对应的值来对所述卷积的位图矩阵的相应单元格进行编码；

用于通过以下操作识别所述卷积的位图矩阵的合格块来改善图像分类效率的单元：(a) 保留具有所述前景单元格计数的所述卷积的位图矩阵的满足单元格保留门限的第一相应单元格，以及(b) 去除具有所述前景单元格计数的所述卷积的位图矩阵的不满足所述单元格保留门限的第二相应单元格。

29. 根据权利要求28所述的系统，还包括用于通过以下操作来生成贡献块矩阵的单元：用保留指示符来对所述贡献块矩阵的所述第一相应单元格进行编码，以及用丢弃指示符来对所述贡献块矩阵的所述第二相应单元格进行编码。

30. 根据权利要求29所述的系统，还包括用于识别所述保留指示符中的相应保留指示符的外周边以生成合格块矩阵的单元。

31. 根据权利要求30所述的系统，还包括用于用所述合格块矩阵来掩蔽检测窗口的单

元,所述掩蔽用于减少要在图像分类中处理的块的数量。

32. 根据权利要求28所述的系统,还包括用于执行以下操作的单元:

将所述全一矩阵内核卷积到所编码的轮廓的相应单元格;以及

通过计算用于所编码的轮廓的所述相应单元格的所述全一矩阵内核内的前景单元格的总和来确定所述前景单元格计数。

使用边界位图对图像分类的改善

[0001] 相关申请

[0002] 本专利要求享有于2016年6月6日提交的题为“Speed Improvement of Object Recognition Using Boundary-Bitmap of Histogram of Oriented Gradients”的序列号为62/346,065的美国临时申请的利益和优先权。序列号为62/346,065的美国临时申请的全部内容在此通过引用方式并入本文。

技术领域

[0003] 概括而言,本公开内容涉及图像分类,并且更具体而言,涉及用边界位图来改善图像分类的方法、系统和装置。

背景技术

[0004] 近年来,已经使用不同的方法实现了图像分类,所述方法包括定向梯度直方图(HoG)分类器。由示例性HoG分类器生成的特征描述符用于例如计算机视觉系统中以检测静态图像和/或视频中的人、动物和/或对象。一般而言,分类器被训练以借助于学习系统(例如,采用支持向量机(SVM)算法的学习系统)来检测这样的对象。因此,分类技术有时被称为HoG/SVM分类系统。

附图说明

[0005] 图1是根据本公开内容的教导构造的用于用边界位图来改善图像分类的示例性分类加速器的示意图。

[0006] 图2A是用于生成用于由图1的示例性分类加速器实现的图像分类的描述符的示例性定向梯度直方图(HoG)过程的示意图。

[0007] 图2B是示例性网格的图示。

[0008] 图3是由图1的示例性分类加速器生成的示例性轮廓窗口的图示。

[0009] 图4是具有边界框以在图2的HoG过程期间裁减示例性HoG检测窗口的示例性HoG检测位图的图示。

[0010] 图5是由图1的示例性分类加速器生成的示例性编码的轮廓矩阵的图示。

[0011] 图6是由图1的示例性分类加速器生成的示例性卷积的位图矩阵的图示。

[0012] 图7是由图1的示例性分类加速器生成的示例性贡献块(CB)矩阵的图示。

[0013] 图8是由图1的示例性分类加速器生成的示例性合格块矩阵的图示。

[0014] 图9-12是表示示例性机器可读指令的流程图,所述指令可以被执行以实现图1的示例性分类加速器以用边界位图来改善图像分类。

[0015] 图13是被组织为执行图9-12的示例性机器可读指令以实现图1的示例性分类加速器的示例性处理器平台的框图。

[0016] 附图未按比例绘制。

具体实施方式

[0017] 检测图像、视频和/或实况视频馈送(例如,流式传输视频)中的对象的工作经常用被训练以检测这样的对象的分类器来完成。在一些示例中,对象包括人脸、人类轮廓、动物、车辆和/或其他类型的对象。在一些示例中,定向梯度直方图(HoG)应用训练图像和学习系统以识别对象和/或以其他方式对对象进行分类。在一些示例中,HoG与支持向量机(SVM)结合应用以对图像中的对象进行分类(在本文中被称为HoG/SVM)。这样的方法的示例描述于Navneet Dalal和Bill Triggs的“Histograms of oriented gradients for human detection”,International Conference on Computer Vision and Pattern Recognition(CVPR'05),June 2005,San Diego,United States,IEEE Computer Society, pp.886-893,2005,其全部内容在此通过引用方式并入本文。

[0018] 如以下更详细地描述的,HoG/SVM评估HoG检测窗口中的候选图像的一部分,并且计算特定单元格内的每个像素的梯度值,创建直方图,管理分组(bin),将单元格描述符聚合成块描述符,并规范化与SVM分类器一起使用(馈入SVM分类器)的块描述符。一般而言,HoG促进候选图像的特征提取以产生一个或多个组合特征向量(HoG描述符),其在被提供给传统SVM窗口分类器时,促进对象检测/辨识。

[0019] 图1示出了用于用边界位图来改善图像分类的示例性分类加速器100。在图1所示的示例中,分类加速器100包括图像数据接口102、边界位图引擎104和HoG/SVM引擎106。示例性分类加速器100经由网络134通信地连接到一个或多个SVM资源132。示例性边界位图引擎104包括示例性边界标识符108、示例性轮廓引擎110、示例性卷积单元格选择器112、示例性窗口引擎114、示例性过滤单元格选择器116、示例性过滤门限引擎118和示例性边界封装器120。在图1所示的示例中,HoG/SVM引擎106包括示例性梯度计算器122、示例性分组(bin)管理引擎124、示例性单元格描述符引擎126、示例性单元格描述符规范化器128和示例性分类接口130。

[0020] 响应于示例性图像数据接口引擎102获得和/或以其他方式检索候选图像,HoG/SVM引擎106的示例性梯度计算器122计算梯度值(例如,幅度梯度和定向梯度)。在该示例中,HoG/SVM引擎106实现用于计算梯度值的手段。然而,可以使用等同的结构来实现用于计算梯度值的手段。一般而言,通过评估局部强度梯度和/或边缘方向的分布,可以更好地表征候选图像内的局部对象。梯度 $|G|$ 的幅度由示例性梯度计算器122以与示例性等式1一致的方式计算。

$$[0021] \quad |G| = |G_x| + |G_y| = |G_{x+1} - G_{x-1}| + |G_{y+1} - G_{y-1}|$$

[0022] 等式1

[0023] 在等式1的所示示例中,根据相邻像素的强度值来计算梯度 $|G|$ 的幅度,并且通过示例性梯度计算器122以与示例性等式2一致的方式计算定向角 θ 。

$$[0024] \quad \theta = \arctan(|G_{y+1} - G_{y-1}| / |G_{x+1} - G_{x-1}|)$$

[0025] 等式2

[0026] 在等式1和2的示例中,梯度计算器122实现用于计算梯度和定向角的手段。然而,可以使用等同的结构来实现用于计算梯度和定向角的手段。

[0027] 为了说明,图2A中示出了示例性HoG/SVM过程200。在图2A所示的示例中,示例性图像数据接口引擎102获取用于分析的候选图像,并且选择该图像的一部分作为HoG检测窗口

202。图2A的示例性HoG检测窗口202是64像素宽 \times 128像素高的(默认情况下,在该示例中),但是可以替代地使用其他尺寸。图2A的示例中的示例性HoG检测窗口202的每个单元格204是 $2^n \times 2^n$ (其中, $n > 0$ 且 $2^n < \text{最小窗口宽度/高度}$)的 8×8 像素网格,并且示例性梯度计算器122使用例如如以上描述的等式来计算它们的幅度和定向值。

[0028] 为了创建直方图,示例性分组管理引擎124将计算出的定向角(θ)分解和/或以其他方式分配到目标数量的分组中。在该示例中,分组管理引擎124实现用于分配的手段,但是其可以使用等同的结构来实现。在图2A所示的示例中,通过示例性分组管理引擎124来将 0° 至 180° 的范围分解成九个 20° 的分组。另外,示例性分组管理引擎124将每个强度值添加到与相应定向角相关联的相应分组以创建示例性HoG单元格描述符206(例如,将幅度匹配到定向的直方图)。图2A的示例性HoG单元格描述符206被表示为直方图并且包括九(9)个值,每个所述值具有最小值零,并且最大值是最大像素强度值的128倍。在该示例中,分组表示定向,分组的高度表示对应的定向角的幅度。

[0029] 示例性单元格描述符引擎126聚合 2×2 单元格块208的所有单元格描述符(直方图)206。注意到,因为单元格描述符引擎126使用单元格块208,所以不在示例性HoG检测窗口202的边缘上的任何HoG单元格204都将出现在四个不同的单元格块208中,以使得对应的单元格描述符206将在四个不同的块描述符210中被表示,如虚线箭头209所示。例如,图2B中的块10将出现在以下聚合中的 2×2 网格中:(a) 5、6、9、10(b) 6、7、10、11(c) 9、10、13、14和(d) 10、11、14、15。如以上描述的,因为局部对象外观和形状通过局部梯度的分布被更好地表征,所以相应单元格的局部直方图累积在块描述符中,并且组合的直方图条目形成表示。在该示例中,单元格描述符引擎126实现用于聚合的手段,但是其可以使用等同的结构来实现。

[0030] 为了促进对光照、阴影等的更好的不变性,块描述符在被提供给SVM以用于对象辨识之前被规范化。在一些示例中,局部直方图表示在较大的空间区域(块)上累积,并用于规范化这样的块中的所有单元格。规范化的描述符块在本文中还称为HoG描述符。组合对应的HoG检测窗口202的所有这样的HoG描述符产生组合特征向量(在本文中有时被称为完整的HoG描述符),其最终被提供给SVM以用于检测。

[0031] 示例性单元格描述符规范化器128对其中具有四个HoG单元格204的单元格描述符206的每个块描述符210进行规范化。在该示例中,单元格描述符规范化器128实现用于规范化的手段,但是其可以使用等同的结构来实现。任何类型和/或数量的规范化算法可以由示例性单元格描述符规范化器128应用,例如由Navneet Dalal和Bill Triggs在前述论文中描述的算。在图2所示的示例中,针对对应的单元格块208示出了规范化块描述符212。这包括一组直方图数据,在该示例中其表示每块36个数据元素(例如,块中具有4个单元格的9个分组)。示例性单元格描述符规范化器128聚合用于示例性HoG检测窗口202的规范化的块描述符212以生成完整的HoG描述符214(例如,组合特征向量),并且对于示例性 64×128 像素窗口,包括105个规范化直方图数据块。具体而言,因为对单元格块的分析发生在(重叠的) 2×2 块的大小中,所以这在HoG检测窗口202上重复七(7)次并且在HoG检测窗口202下重复十五(15)次($7 \times 15 = 105$)。示例性分类接口130将示例性完整HoG描述符214馈送和/或以其他方式提供给SVM资源132以执行分类。在一些示例中,SVM资源使用先前评估的训练图像,其通常用于机器学习应用中,例如 Inria® 人物图像数据集和/或 Daimler® 图像数据集。在

该示例中,分类接口130实现将描述符提供给SVM资源的手段,但是其可以使用等同的结构来实现。

[0032] 通过针对候选图像的任何数量的部分重复步进通过示例性HoG检测窗口202来执行前述HoG计算。跨源图像的矩形描述符的该步进涉及分析与匹配任务无关的像素,这反映了浪费的计算资源。与包含待检测对象的图像的一部分无关的像素在本文中被称为背景像素。另一方面,可能与待检测的对象相关的像素被称为前景像素。一般而言,生成完整HoG描述符214(组合特征向量)的HoG计算成本高。例如,对于感兴趣的 42×42 像素候选区域,需要大约11,500个加法运算、1,300个乘法、5,200个除法、16个平方根和5200个反正切运算。一些浪费的计算资源被应用于不是待分析的候选图像的部分的背景像素。

[0033] 为了改善示例性HoG/SVM过程的效率,本文公开的示例建立位图以识别示例性HoG检测窗口202中的像素子集以进行分析。本文公开的示例以与2015年12月17日提交、2017年5月2日授权的题为“Systems and Methods for Providing an Image Classifier”的美国专利第9,639,777号一致的方式建立和/或以其他方式创建位图,所述专利的全部内容通过引用方式并入本文。为了建立和/或以其他方式创建位图,示例性边界标识符108和/或示例性轮廓引擎110首先识别根据应用于待分类图像的训练图像来生成的轮廓(候选图像的具有与HoG检测窗口202相同大小的一部分)。在一些示例中,边界标识符108与训练图像/数据集结合以对候选图像进行平均,并且示例性轮廓引擎110应用门限处理以产生和/或以其他方式计算轮廓窗口300,如图3所示。在该示例中,边界标识符108和轮廓引擎110实现用于创建位图的手段和/或用于识别轮廓的手段,但是其可以使用等同的结构来实现。在图3所示的示例中,INRIA®行人检测图像数据集的平均值用于生成示例性前景轮廓302。示例性轮廓窗口300具有 48×96 像素的大小,其被示例性边界标识符108划分成 4×4 的单元格,总共288个单元格($48/4 \times 96/4 = 12 \times 24 = 288$ 个 4×4 的单元格)。在图3所示的示例中的288个单元格中,示例性轮廓引擎110识别出前景轮廓302中包括92个单元格,其表示图像整个区域的32%。

[0034] 在一些示例中,轮廓引擎110将具有训练图像的每个位置中的像素强度值加在一起,然后将得到的值除以训练图像的数量,如由Jun-Yan Zhu et al.,“AverageExplorer: Interactive Exploration and Alignment of Visual Data Collections,”ACM Transactions on Graphics,Vol.33,No.4,August 2014公开的,其全部内容通过引用方式并入本文。在一些示例中,MATLAB函数imadd()和imdivide()用于对数据集执行这些函数,并且将结果输出到位图文件。然后,示例性轮廓引擎110生成示例性前景轮廓302,作为数据集、计算机视觉和/或深度学习的逐像素平均的结果。示例性前景轮廓302由示例性轮廓引擎110转换以形成HoG检测位图402,如图4所示。

[0035] 在图4所示的示例中,HoG检测位图402显示在HoG检测窗口404旁边,所述HoG检测窗口404类似于图2的示例性HoG检测窗口202。在图4所示的示例中,HoG检测窗口404包括待分析的候选图像。然而,与图2的示例性HoG检测窗口202不同,图4的HoG检测窗口404基于示例性HoG检测位图402用边界框406来约束HoG/SVM过程,以仅处理HoG检测窗口404内的候选图像内的那些待分类的包括示例性前景轮廓302的前景信息的单元格。换言之,当待分类的候选图像与HoG检测位图重叠时,相关像素(前景像素)和不相关像素(背景像素,例如包含与和轮廓无关的场景/对象相关的信息的像素)二者都被仔细检查,并且边界框406防止对

这样的不相关像素的不必要处理。实际上,不是在示例性HoG检测窗口404中处理单元格的整个数量,而是示例性轮廓引擎110应用具有边界框406的示例性HoG检测位图402,以通过减少待分析的单元格和像素的数量来裁减和/或以其他方式约束HoG检测窗口404,从而允许较低数量的定向分组被使用,并减小所得的HoG描述符的总体大小,例如图2的示例性完整HoG描述符214。因此,通过仅将候选图像的相关部分与参考图像的仅相关部分进行比较,通过本文公开的示例实现了工艺成本和复杂性的显著降低。

[0036] 然而,虽然示例性HoG检测位图402的应用通过去除候选图像背景的部分来改善HoG/SVM效率,但是本文公开的示例通过还去除前景的部分来进一步改善计算效率。具体而言,本文公开的示例去除前景的部分以允许对候选图像内的对象进行表示和分类而没有显著的精度损失,如以下更详细地描述的。本文公开的示例以减少执行分类的设备(例如,移动设备)上的功耗的方式去除前景的部分,并且降低执行分类的设备的带宽要求。另外,本文公开的示例减小了最终提供给示例性SVM资源132的描述符的大小,这减少了在一个或多个存储器与一个或多个处理器之间的数据传输。

[0037] 图5示出了由示例性轮廓引擎110生成的示例性编码的轮廓矩阵500。在图5所示的示例中,轮廓引擎110用零(“0”)值来编码背景单元格,并且以一(“1”)值来编码前景单元格。另外,示例性轮廓引擎110识别与过渡单元格相对应的位图区域502表现出从值零到值一的相邻变化(反之亦然),如实线所示。如以上讨论的,虽然轮廓引擎110实现用于生成编码的轮廓矩阵的手段以及用于编码单元格的手段,但是可以使用其他等同的结构来实现这样的手段。示例性位图区域502通常类似于人的轮廓。如以上讨论的,示例性编码的轮廓500在将其应用于示例性检测窗口时实现显著的计算节省,以防止在HoG/SVM过程期间计算不必要的单元格。在该示例中,位图区域502类似于人的轮廓。可以通过传统的HoG/SVM过程来分析并/或以其他方式计算示例性位图区域502内部的所有单元格。然而,前景单元格的很大一部分对于人类对象的分类是不必要的,因此,分析该区域中的所有单元格浪费计算资源。

[0038] 为了通过HoG/SVM过程减少待分析的单元格的数量,图1的示例性卷积单元格选择器112卷积示例性编码的轮廓500以生成卷积的位图矩阵600,如图6所示。在该示例中,卷积单元格选择器112实现用于卷积的手段,但是可以使用其他等同的结构来实现用于卷积的手段。另外,示例性卷积的位图矩阵600指示位图区域502内部的块。每个块包括多个单元格。如以上描述的,每个单元格包括多个像素。然而,示例性编码的轮廓500的一些单元格不包括指示前景像素的任何块,一些单元格可以仅包括指示前景像素的部分块(例如,少于块的四个单元格,但是大于块的一个单元格),并且一些单元格可以完全在前景内(例如,每块4个单元格)。因此,特定单元格是保持以用于进一步分析的更好候选者,这是因为它们与分类过程相关。在图5所示的示例中,卷积单元格选择器一次选择一个单元格(例如,从左上单元格504A开始)。示例性窗口引擎114将所选择的单元格与全一矩阵内核506进行卷积,并且在图5所示的示例中,全一矩阵内核506是 2×2 内核(例如,包含四个单元格的一个块)。在该示例中,窗口引擎114实现用于与内核进行卷积的手段,但是可以使用其他等同的结构来实现用于与内核进行卷积的手段。另外,示例性窗口引擎114对如以上描述的针对前景单元格具有值1并且针对背景单元格具有值0的前景单元格的数目进行计数。在图5所示的示例中,第一所选择单元格504A不包括全一矩阵内核506内的任何前景单元格,并且所得的计数等

于零。示例性卷积单元格选择器112将该计算出的值(例如,0)写入其对应的单元格位置504B中的示例性卷积的位图矩阵600(参见图6)。

[0039] 为了进一步说明,考虑由示例性卷积单元格选择器112选择的第二单元格508A。示例性窗口引擎114将所选择的单元格与另一 2×2 矩阵内核510进行卷积,并且示例性窗口引擎114对两(2)个前景单元格的数量进行计数。换言之,示例性第二单元格508A仅表示块的一部分(例如,块=四个单元格)。如此,示例性卷积单元格选择器112将该计算出的和(例如,2)写入其对应的单元格位置508B中的示例性卷积的位图矩阵600(参见图6)。

[0040] 为了进一步说明,考虑由示例性卷积单元格选择器112选择的第三单元格512A。示例性窗口引擎114将所选择的单元格与另一 2×2 矩阵内核514进行卷积,并且示例性窗口引擎114对四(4)个前景单元格的数量进行计数。如此,示例性卷积单元格选择器112将该计算出的和(例如,4)写入其对应的单元格位置512B中的示例性卷积的位图矩阵600(参见图6)。

[0041] 在图6所示的示例中,卷积的位图矩阵600包括基于特定内核大小和块重叠大小的零(0)和四(4)之间的总和值。为了识别哪些块(单元格组)应当对HoG/SVM过程进行贡献,示例性过滤单元格选择器116选择单元格,并且示例性过滤门限引擎118针对单元格保留门限值评估该单元格。在该示例中,过滤单元格选择器116实现用于选择单元格的手段,但是可以使用其他等同的结构来实现用于选择单元格的手段。另外,在该示例中,过滤门限引擎118实现用于评估单元格的手段,但是可以使用其他等同的结构来实现用于评估单元格的手段。在图6所示的示例中,过滤门限引擎118应用过滤值四(4),并且如果单元格满足单元格保留门限,则示例性过滤单元格选择器116通过在对矩阵单元格中写入/编码保留指示符(例如,值一(1))来保留相应单元格,否则示例性过滤单元格选择器116通过将丢弃指示符(例如,值零(0))写入/编码到对应的矩阵单元格来(从考虑中)去除相应单元格。图7示出了由前述过滤和编码得到的示例性贡献块(CB)矩阵700。换言之,示例性CB矩阵700被编码以识别合格的前景块/单元格。虽然图7所示的示例识别用于保留指示符的值一(1)和用于丢弃指示符的值零(0),但其他示例可以使用任何保留和/或丢弃指示符表示。

[0042] 然而,为了仅识别边界702(CB矩阵700的边缘)的贡献块,示例性边界封装器120识别包含保留指示符(例如,值一(1))802的边界位图区域(例如,单元格的外边缘/周边)806,并且替换丢弃指示符(例如,具有值零(0)的所有其他单元格)804,如图8所示。在该示例中,边界封装器120实现用于识别边界位图区域的手段,但是可以使用其他等同的结构来实现用于识别边界位图区域的手段。在图8所示的示例中,示出了合格块矩阵800,其识别CB矩阵的边界位图区域806。示例性合格块矩阵800被编码以防止HoG/SVM处理或计算对于对象分类不必要的前景单元格(例如,合格单元格用一(1)来编码,并且不合格单元格用零来编码(0))。示例性分类接口130将示例性合格块矩阵800应用于HoG检测窗口,例如图2的示例性HoG检测窗口202,以在HoG/SVM分类过程期间约束和/或以其他方式减少待计算的单元格的数量。在一些示例中,分类接口130结合所应用的合格块矩阵800对HoG检测窗口执行掩蔽操作,以使得在HoG/SVM分类过程期间将仅使用包含相应保留指示符的重叠单元格。如此,用于HoG/SVM处理的计算资源大大减少,由此实现了设备(例如,嵌入式设备、移动设备等)的功率节省和存储器利用率降低。

[0043] 如以上描述的,由本文所公开的示例生成的完整HoG描述符(例如,示例性完整HoG描述符214)具有改善的长度(例如,较短)。作为结果,当完整的HoG描述符被提供给SVM资源

132以用于分类过程时,这样的SVM资源132也可以表现出计算改善。在一些示例中,鉴于本文公开的示例,由示例性SVM资源132计算的检测结果较快。

[0044] 在一些示例中,来自不同数据集的图像具有不同的大小,因此固定大小的边界框可能无法提供足够的准确度。为了解决该缩放环境,可以鉴于边界框和相关联的边界位图区域来应用缩放因子。示例性x-维缩放因子(S_x)和示例性y-维缩放因子(S_y)分别被示出为示例性等式3和4。

$$[0045] \quad S_x = \frac{NI_x}{RI_x}$$

[0046] 等式3

$$[0047] \quad S_y = \frac{NI_y}{RI_y}$$

[0048] 等式4

[0049] 在等式3和4所示的示例中,NI表示新图像(例如,来自INRIA数据集),并且RI表示参考图像(例如,来自Daimler数据集)。例如,如果参考图像的大小是 96×48 并且新图像的大小是 134×70 ,则沿x和y方向的对应缩放因子(S)是:

$$[0050] \quad S_x = \frac{70}{48} = 1.458, \quad S_y = \frac{134}{96} = 1.395$$

[0051] 在一些示例中,虽然边界框和边界位图的缩放,但是对应的大小可以通过向上或向下取整为单元格大小的倍数来调节。在缩放导致位图图案多边形的顶点在单元格内部移位而不是发生在单元格的角上的情况下,示例性边界位图引擎104将位图多边形的任何顶点取整为它所落入的单元格的最近顶点。

[0052] 虽然实现图1的分类加速器100的示例性方式在图2-8中被示出,但图1中示出的元素、过程和/或设备中的一个或多个可以以任何其他方式来组合、划分、重新布置、省略、消除和/或实现。此外,示例性图像数据接口引擎102、示例性边界位图引擎104、示例性边界标识符108、示例性轮廓引擎110、示例性卷积单元格选择器112、示例性窗口引擎114、示例性过滤单元格选择器116、示例性过滤门限引擎118、示例性边界封装器120、示例性Hog/SVM引擎106、示例性梯度计算器122、示例性分组管理引擎124、示例性单元格描述符引擎126、示例性单元格描述符规范化器128、示例性分类接口130和/或更一般地,图1的示例性分类加速器100可以由硬件、软件、固件和/或硬件、软件和/或固件的任何组合实现。因此,例如,示例性图像数据接口引擎102、示例性边界位图引擎104、示例性边界标识符108、示例性轮廓引擎110、示例性卷积单元格选择器112、示例性窗口引擎114、示例性过滤单元格选择器116、示例性过滤门限引擎118、示例性边界封装器120、示例性Hog/SVM引擎106、示例性梯度计算器122、示例性分组管理引擎124、示例性单元格描述符引擎126、示例性单元格描述符规范化器128、示例性分类接口130和/或更一般地,图1的示例性分类加速器100中的任何一个可以由一个或多个模拟或数字电路、逻辑电路、可编程处理器、专用集成电路(ASIC)、可编程逻辑设备(PLD)和/或现场可编程逻辑设备(FPLD)实现。当阅读本专利的用于涵盖纯粹的软件和/或固件实施方式的装置或系统权利要求中的任何一个时,示例性图像数据接口引擎102、示例性边界位图引擎104、示例性边界标识符108、示例性轮廓引擎110、示例性卷积单元格选择器112、示例性窗口引擎114、示例性过滤单元格选择器116、示例性过滤门限引擎

118、示例性边界封装器120、示例性Hog/SVM引擎106、示例性梯度计算器122、示例性分组管理引擎124、示例性单元格描述符引擎126、示例性单元格描述符规范化器128、示例性分类接口130和/或更一般地图1的示例性分类加速器100中的至少一个在此被明确定义为包括非暂时性计算机可读存储设备或存储盘,例如存储器、数字多功能盘(DVD)、光盘(CD)、蓝光盘等,包括软件和/或固件。此外,除了或替代图1-8所示的那些,图1的示例性分类加速器100可以包括一个或多个元素、过程和/或设备,并且/或者可以包括所示元素、过程和设备中的任何一个或全部中的多于一个。

[0053] 在图9-12中示出了表示用于实现图1的分类加速器100的示例性机器可读指令的流程图。在这些示例中,机器可读指令包括用于由诸如以下结合图13讨论的示例性处理器平台1300中示出的处理器1312之类的处理器执行的程序。程序可以体现在存储在非暂时性计算机可读存储介质上的软件中,所述非暂时性计算机可读存储介质例如CD-ROM、软盘、硬盘驱动器、数字多功能盘(DVD)、蓝光盘或者与处理器1312相关联的存储器,但是整个程序和/或其部分可以替代地由除了处理器1312之外的设备执行和/或体现在固件或专用硬件中。此外,虽然参考图9-12所示的流程图描述了示例性程序,但是可以替代地使用实现示例性分类加速器100的许多其他方法。例如,可以改变块的执行次序,并且/或者可以改变、消除或组合所描述的块中的一些。另外地或替代地,块中的任何一个或所有可以由一个或多个硬件电路(例如,分立和/或集成的模拟和/或数字电路、现场可编程门阵列(FPGA)、专用集成电路(ASIC)、比较器、运算放大器(op-amp)、逻辑电路等)实现,其被组织为在不执行软件或固件的情况下执行对应的操作。

[0054] 如上所述,使用存储在非暂时性计算机和/或机器可读介质上的编码的指令(例如,计算机和/或机器可读指令)来实现图9-12的示例性过程,所述非暂时性计算机和/或机器可读介质例如硬盘驱动器、闪存存储器、只读存储器、压缩盘、数字多功能盘、高速缓存、随机存取存储器 and/或在其中信息可以被存储任何持续时间(例如,延长的时间段、永久地;作为简短的实例,用于对信息进行暂时缓冲和/或高速缓存)的任何其他存储设备或存储盘。如本文使用的,术语非暂时性计算机可读介质被明确地定义为包括任何类型的计算机可读存储设备和/或存储盘,并且排除传播信号并排除传输介质。“包括”和“包含”(及其所有形式和时态)在本文中用作开放式术语。因此,每当权利要求在任何形式的“包括(include)”或“包含(comprise)”(例如,包含(comprises)、包括(includes)、包含(comprising)、包括(including)等)之后列出任何内容时,应当理解可以存在另外的元素、术语等而不落在对应权利要求的范围之外。如本文使用的,当短语“至少”被用作权利要求的前序中的过渡术语时,它也是开放式的,与术语“包括”和“包含”是开放式的方式相同。

[0055] 图9的程序900开始于框902处,其中,示例性图像数据接口引擎102获取、接收和/或以其他方式检索图像以用于分类。虽然在所检索的图像内对一个或多个对象进行分类的传统方法包括立即应用HoG/SVM过程,但是本文公开的示例通过首先生成边界位图来拦截HoG/SVM分类过程的发起,以使得任何后续的HoG/SVM分类过程以改善的方式操作(例如,改善的计算效率、较低的存储器需求、较低的功耗等)。示例性边界位图引擎104生成边界位图(框904)以促进HoG/SVM分类的改善方式,如以上讨论和以下更详细讨论的。

[0056] 响应于生成用于通过HoG/SVM分类过程裁减和/或以其他方式减少待分析的像素数量的位图,示例性梯度计算器122计算示例性HoG检测窗口202的单元格204的梯度值(框

906)。另外,示例性分组管理引擎124基于计算出的定向角来计算对应的分组(框908)。以上结合图2公开的示例示出了每个 20° 的分组,但是本文公开的示例不限于此。对应的分组的累积强度值允许示例性分组管理引擎124生成HoG单元格描述符(直方图),例如图2的示例性单元格描述符206。示例性单元格描述符引擎126聚合单元格描述符206以生成块描述符,例如图2的示例性块描述符210(框910)。

[0057] 示例性块描述符210由示例性单元格描述符规范化器128进行规范化(框912)。在一些示例中,单元格描述符规范化器128应用L2范数。由单元格描述符规范化器128执行的示例性规范化产生规范化的块描述符,例如图2的示例性规范化块描述符212。另外,当HoG检测窗口跨越通过其各自的单元格时,示例性单元格描述符规范化器128聚合相应的规范化块描述符(框912),这产生示例性完整的HoG描述符,例如图2的示例性完整的HoG描述符214。示例性分类接口130将示例性完整的HoG描述符214转发、发送和/或以其他方式提供给SVM分类器(框914),例如用于促进分类(对象辨识)的图1的示例性SVM资源132。

[0058] 图10包括与生成图9的位图相关联的另外的细节(框1002)。在图10所示的示例中,示例性边界标识符108识别所选择的图像中的对象的矩形边界,例如图3所示的轮廓窗口300。示例性轮廓引擎110识别和/或以其他方式计算示例性轮廓窗口300内的前景轮廓302(框1004)。示例性轮廓引擎110还生成和/或以其他方式识别对应的边界框(例如,图4的示例性边界框406),并且生成轮廓的二进制矩阵以识别和/或以其他方式显示边界框是否还包括背景单元格(框1006)。在一些示例中,轮廓引擎110基于轮廓的边界来生成边界框(区域),这去除了示例性矩形边界(轮廓窗口300)的背景单元格的部分。由示例性轮廓引擎110生成的示例性二进制矩阵在图5中示出,其中,在前景轮廓302外部的单元格用零(0)来编码,并且在前景轮廓302内部的单元格用一(1)来编码。

[0059] 为了识别位图区域(例如,图5的其中仅包括前景单元格的示例性位图区域502),示例性分类加速器100调用示例性卷积单元格选择器112来对编码的轮廓进行卷积(框1008)。图11示出了与卷积相关联的另外的细节(框1008)。在图11所示的示例中,示例性卷积单元格选择器112从示例性编码的轮廓矩阵500中选择第一单元格(框1102)。示例性窗口引擎114将分析窗口应用于所选择的第一单元格(框1104)。如以上描述的,分析窗口可以是全一矩阵内核(例如,图5的示例性全一矩阵506),其是 2×2 内核。示例性窗口引擎114对所应用的矩阵内的前景单元格的数目进行计数(框1106),其在图5所示的示例中对于第一所选择的单元格504A不包括任何前景单元格。如此,在该示例中,示例性窗口引擎114计算总和为零。示例性卷积单元格选择器112将该总和值写入卷积的位图矩阵(框1108)(例如,图6的示例性卷积的位图矩阵600),并写入在其相应的单元格位置中。如果示例性卷积单元格选择器112确定在示例性编码的轮廓500中存在待分析的另外的单元格(框1110),则卷积单元格选择器移动和/或以其他方式跨越到下一可用单元格(例如,逐单元格跨越到右侧,然后绕回到新的行并继续从左到右跨越/选择),并且控制返回到框1102。在另一方面,如果示例性卷积单元格选择器112确定示例性编码的轮廓500的所有单元格已经被卷积(框1110),则示例性程序(框1108)退出并且控制返回到图10的框1010。

[0060] 如以上结合图5和图6讨论以生成示例性卷积的位图矩阵600,基于所选择的特定内核大小和重叠步幅导致在零(0)和四(4)之间的总和值,但是本文公开的示例不限于此。图6的示例性卷积的位图矩阵600示出了是HoG/SVM分类过程的候选者的那些单元格,其中,

现在通过识别其中的零 (0) 值来从考虑中去除所有背景单元格。然而,为了经由应用HoG/SVM来进一步缩小应当对分类进行贡献的单元格的选择,示例性过滤单元格选择器116对示例性卷积的位图矩阵600进行过滤以识别其中的候选贡献块(框1010)。

[0061] 图12示出了与块1010的过滤相关联的另外的细节。在图12所示的示例中,示例性过滤单元格选择器116从卷积的位图矩阵600中选择单元格(框1202)。示例性过滤门限引擎118将所选择的单元格的值与门限值进行比较(框1204)。在该示例中,门限值是四(4),并且如果单元格值满足该门限测试,则示例性过滤单元格选择器116将单元格值一(1)写入示例性贡献块(CB)矩阵的相应单元格(框1206),例如图7的示例性CB矩阵700。另一方面,如果示例性过滤门限引擎118确定单元格值不满足门限测试(例如,值小于四(4)),则示例性过滤单元格选择器116将单元格值零(0)写入到示例性CB矩阵的相应单元格(框1208)。示例性过滤单元格选择器116确定示例性卷积的位图矩阵600是否包括用于针对门限进行测试/分析的另外的单元格(框1210),并且如果存在待分析的剩余单元格,则示例性过滤单元格选择器116移动和/或以其他方式跨越到示例性卷积的位图矩阵600内的下一可用单元格(框1212),并且控制返回到框1202。否则,框1010的示例性程序退出并且控制返回到图10的框1012。

[0062] 作为框1010的前述过滤的结果,示例性CB矩阵700被生成以仅识别足以促进经由HoG/SVM的分类的那些单元格(合格的前景块)。然而,内部前景单元格中的一些能够在不牺牲分类准确性的情况下被去除。示例性边界封装器120封装和/或以其他方式识别具有值一(1)的单元格的外边界(框1012),并且用值零(0)来盖写不是外边界的部分的任何其它单元格(框1014)。作为结果,由边界封装器120生成合格块矩阵800,以在HoG/SVM分类过程期间约束、引导和/或以其他方式掩蔽HoG检测窗口,以针对分类工作减少待分析的单元格的數量。

[0063] 图13是能够执行图9-12的指令以实现图1的示例性分类加速器100的示例性处理器平台1300的框图。处理器平台1300可以例如是服务器、个人计算机、移动设备(例如,手机、智能电话、诸如iPad™之类的平板计算机)、个人数字助理(PDA)、互联网设备、DVD播放器、CD播放器、数字视频记录器、蓝光播放器、游戏终端、个人视频记录器、机顶盒、嵌入式设备或任何其他类型的计算设备。

[0064] 所示示例的处理器平台1300包括处理器1312。所示示例的处理器1312是硬件。例如,处理器1312可以由来自任何期望的系列或制造商的一个或多个集成电路、逻辑电路、微处理器或控制器实现。硬件处理器可以是基于半导体的(例如,基于硅的)设备。在该示例中,处理器1312实现示例性图像数据接口引擎102、示例性边界位图引擎104、示例性HoG/SVM引擎106,并且在一些示例中实现示例性SVM资源132。在图15所示的示例中,处理器1312包括经由示例性指令1332配置的一个或多个示例性处理核心1315,所述示例性指令1332包括图9-12的示例性指令。

[0065] 所示示例的处理器1312包括本地存储器1313(例如,高速缓存)。所示示例的处理器1312经由总线1318与包括易失性存储器1314和非易失性存储器1316的主存储器通信。易失性存储器1314可以由以下各项实现:同步动态随机存取存储器(SDRAM)、动态随机存取存储器(DRAM)、RAMBUS动态随机存取存储器(RDRAM)和/或任何其他类型的随机存取存储器设备。非易失性存储器1316可以由闪速存储器和/或任何其他期望类型的存储器设备实现。对

主存储器1314、1316的访问由存储器控制器进行控制。

[0066] 所示示例的处理器平台1300还包括接口电路1320。接口电路1320可以由任何类型的接口标准实现,所述接口标准例如以太网接口、通用串行总线(USB)和/或PCI快速接口。

[0067] 在所示示例中,一个或多个输入设备1322连接到接口电路1320。输入设备1322允许用户将数据和/或命令录入到处理器1312中。输入设备可以由例如以下各项实现:音频传感器、麦克风、(静止或视频)相机、键盘、按钮、鼠标、触摸屏、跟踪板、轨迹球、isopoint和/或语音辨识系统。

[0068] 一个或多个输出设备1324还连接到所示示例的接口电路1320。输出设备1324可以例如由显示设备(例如,发光二极管(LED)、有机发光二极管(OLED)、液晶显示器、阴极射线管显示器(CRT)、触摸屏、触觉输出设备、打印机和/或扬声器)实现。因此,所示示例的接口电路1320通常包括图形驱动卡、图形驱动芯片和/或图形驱动处理器。

[0069] 所示示例的接口电路1320还包括通信设备,例如发射机、接收机、收发机、调制解调器和/或网络接口卡,所述通信设备用于促进经由网络1326(例如,以太网连接、数字订户线(DSL)、电话线、同轴电缆、蜂窝电话系统等)与外部机器(例如,任何种类的计算设备)的数据交换。

[0070] 所示示例的处理器平台1300还包括用于存储软件和/或数据的一个或多个大容量存储设备1328。这样的大容量存储设备1328的示例包括软盘驱动器、硬盘驱动器、压缩盘驱动器、蓝光盘驱动器、RAID系统和数字多功能盘(DVD)驱动器。

[0071] 图9-12的编码的指令1332可以被存储在大容量存储设备1328中、易失性存储器1314中、非易失性存储器1316中,和/或可移除的有形计算机可读存储介质(例如,CD或DVD)上。

[0072] 本文公开的示例可以应用于通过链路耦合到相应网关的物联网(IoT)网络。物联网(IoT)是其中大量计算设备互连到彼此和互联网以提供非常低级别的功能和数据获取的概念。因此,如本文使用的,IoT设备可以包括与其他IoT设备和较宽的网络(例如,互联网)通信的半自主设备,其执行例如感测、图像分类(例如,静止图像、流式传输视频等)或控制等的功能。

[0073] 通常,IoT设备在存储器、大小或功能方面是有限的,允许较大数量以与较小数量的较大设备类似的成本部署。然而,IoT设备可以是智能电话、膝上型计算机、平板计算机或PC、或其他较大的设备。此外,IoT设备可以是虚拟设备,例如智能电话、嵌入式设备或其他计算设备上的应用。IoT设备可以包括IoT网关,其用于将IoT设备耦合到其他IoT设备和云应用,以用于数据存储、过程控制等。

[0074] IoT设备的网络可以包括商业和家庭自动化设备,例如配水系统、电力分配系统、管道控制系统、工厂控制系统、灯开关、恒温器、锁、相机、警报器、运动传感器等。IoT设备可以是可通过远程计算机、服务器和其他系统访问的,例如,以控制系统或访问数据。

[0075] 互联网和类似网络的未来增长可以涉及非常大量的IoT设备。因此,在本文讨论的技术的上下文中,用于这样的未来联网的多个创新将解决所有这些层的需要,以不受阻碍地增长、发现连接的资源并使其可访问、并且支持隐藏和划分连接的资源的能力。可以使用任何数量的网络协议和通信标准,其中,每个协议和标准被设计为解决特定目标。此外,协议是支持人类可访问服务的结构的部分,所述服务不论位置、时间或空间地操作。创新包括

服务交付和相关联的基础设施,例如硬件和软件;安全增强;以及基于服务级别和服务交付协议中规定的服务质量(QoS)条款的服务供应。

[0076] 骨干链路可以包括任何数量的有线或无线技术,包括光网络,并且可以是局域网(LAN)、广域网(WAN)或互联网的部分。另外,这样的通信链路促进IoT设备和网关二者之中的光信号路径,包括使用促进各种设备的互连的MUXing/deMUXing组件。

[0077] 网络拓扑可以包括任何数量的类型的IoT网络,例如被提供有使用蓝牙低能量(BLE)链路的网络的网状网络。可能存在的其他类型的IoT网络包括用于通过IEEE 802.11(Wi-Fi®)链路和IoT设备通信的无线局域网(WLAN)网络、用于通过LTE/LTE-A(4G)或5G蜂窝网络与IoT设备通信的蜂窝网络、以及低功率广域(LPWA)网,例如,与由LoRa联盟颁布的LoRaWan规范兼容的LPWA网络、或与由互联网工程任务组(IETF)颁布的规范兼容的低功率广域网(LPWAN)上的IPv6。另外,相应的IoT网络可以使用任何数量的通信链路和外部网络提供商(例如,层2或层3提供商)通信,所述通信链路例如LTE蜂窝链路、LPWA链路或基于IEEE802.15.4标准(例如,Zigbee®)的链路。相应的IoT网络还可以使用各种网络和互联网应用协议(例如,约束应用协议(CoAP))来操作。相应的IoT网络还可以与协调器设备集成,所述协调器设备提供形成链接的设备和网络的集群树的链路链。

[0078] 这些IoT网络中的每一个可以为新技术特征提供机会,所述新技术特征例如本文描述的那些。改善的技术和网络可以实现设备和网络的指数增长,包括将IoT网络用于雾设备或系统中。随着这样的改善技术的使用的增长,可以开发IoT网络用于自我管理、功能演进和协作,而无需直接的人为干预。改善的技术甚至可以使得IoT网络在没有集中控制系统的情况下运行。因此,本文描述的改善的技术可以用于远远超出当前实施方式的自动化和增强网络管理和操作功能。

[0079] 在示例中,例如在骨干链路上在IoT设备之间的通信可以由用于认证、授权和计费(AAA)的分散系统保护。在分散AAA系统中,可以跨互连的异构网络基础设施实现分布式支付、信用、审计、授权和认证系统。这允许系统和网络朝向自主操作移动。在这些类型的自主操作中,机器甚至可以签订人力资源合同并与其他机器网络协商合作伙伴关系。这可以允许实现共同目标和平衡的服务交付,而不是概述的、计划的服务水平协议,以及实现提供计量、测量、可追溯性和可跟踪性的解决方案。创建新的供应链结构和方法可能使得大量服务被创建、针对值而被挖掘并在没有任何人类参与的情况下崩溃。

[0080] 通过将诸如声音、光、电子交通、面部和模式辨识、图像分类、气味、振动之类的感测技术集成到IoT设备之间的自主组织中,可以进一步增强这样的IoT网络。感测系统的集成可以允许相对于基于合同服务目标、协调和服务质量(QoS)的资源的集群和融合的服务交付的系统的和自主的通信和协调。

[0081] 可以配备IoT设备的集群以与其他IoT设备以及云网络通信。这可以允许IoT设备在设备之间形成自组织网络,以允许它们充当单个设备,其可以被称为雾设备。

[0082] 与IoT设备的网状网络通信的云计算网络可以充当云计算网络边缘处的雾设备。IoT设备的网状网络可以被称为雾,其在云的边缘处操作。

[0083] 雾可以被认为是大规模互连网络,其中,多个IoT设备例如通过无线电链路和彼此通信。作为示例,可以使用由Open Connectivity Foundation™(OCF)发布的互连规范来促

进该互连网络。该标准允许设备发现彼此并建立用于互连的通信。还可以使用其他互连协议,包括例如优化的链路状态路由(OLSR)协议、用于移动自组织联网的更好的方法(B.A.T.M.A.N.)路由协议、或OMA轻量级M2M(LWM2M)协议等。

[0084] 三种类型的IoT设备包括网关、数据聚合器和传感器,但是可以使用IoT设备和功能的任何组合。网关可以是在云和雾之间提供通信的边缘设备,并且还可以为从传感器获得的数据(例如,运动数据、流数据、温度数据等)提供后端处理功能。数据聚合器可以从任何数量的传感器收集数据,并且执行后端处理功能以进行分析。结果、原始数据或二者可以通过网关向前传递到云。传感器可以是完全的IoT设备,例如,其既能够收集数据又能够处理数据。在一些情况下,传感器在功能方面可以更受限,例如,收集数据并允许数据聚合器或网关处理数据。

[0085] 来自任何IoT设备的通信可以沿着在任何IoT设备之间的方便路径(例如,最方便路径)向前传递以到达网关。在这些网络中,互连的数量提供了显著的冗余,以允许即使丢失了多个IoT设备,也可以保持通信。另外,使用网状网络可以允许使用功率非常低或位于距基础设施一定距离的IoT设备,这是因为连接到另一IoT设备的范围可以远小于连接到网关的范围。

[0086] 从这些IoT设备提供的雾可以被呈现给云中的设备(例如,服务器),作为位于云边缘处的单个设备(例如,雾设备)。在该示例中,来自雾设备的警报可以被发送而不被识别为来自雾内的特定IoT设备。以该方式,雾可以被认为是分布式平台,其提供计算和存储资源以执行处理或数据密集型任务,例如数据分析、数据聚合和机器学习等。

[0087] 在一些示例中,可以使用命令式编程风格来配置IoT设备,例如,其中每个IoT设备具有特定功能和通信伙伴。然而,形成雾设备的IoT设备可以以声明式编程风格进行配置,以允许IoT设备重新配置它们的操作和通信,例如响应于条件、查询和设备故障来确定所需的资源。作为示例,来自位于服务器处的用户关于由IoT设备监测的装备子集的操作的查询可以引起雾设备选择回答查询所需的IoT设备(例如,特定传感器)。然后,在由雾设备发送到服务器以回答查询之前,可以通过传感器、数据聚合器或网关的任何组合来聚合和分析来自这些传感器的数据。在该示例中,雾中的IoT设备可以基于查询来选择使用的传感器,例如添加来自流传感器或温度传感器的数据。此外,如果IoT设备中的一些不可操作,则雾设备中的其他IoT设备可以提供类似数据(如果可用)。

[0088] 根据前述内容可以领会,已经公开了改善图像分类工作的示例性方法、装置、系统和制品。HoG/SVM分类过程的传统方法已经为图像分类和辨识工作带来了显著改善。然而,这样的传统方法对处理设备施加了计算负荷。这样的负荷可能过于严格,特别是对于不享受桌上型平台和服务器的相对丰富的存储器、带宽和/或功率可用性的嵌入式设备而言。因为传统的HoG/SVM分类过程通过跨测试图像重复步进HoG窗口来执行检测,所以随着测试图像大小、分辨率和缩放调整的增加,对应的计算成本增加。

[0089] 如此,本文公开的示例通过去除需要对象检测的图像的背景单元格和前景单元格来减少计算负担。在一些示例中,将位图应用于HoG检测窗口以掩蔽和/或以其他方式裁减HoG检测窗口以仅馈送在测试图像内成功分类对象所需的那些内部单元格。对HoG/SVM分类过程的输入的这样的减少使得待分析的图像像素/单元格/块的数量对应减少、待计算的HoG单元格描述符的数量的减少、待生成的块描述符的数量的减少、要对块描述符执行的规

范化计算量的减少、以及要馈送到SVM中的聚合/完成的HoG描述符的数量的减少。

[0090] 本文公开了用于用边界位图来改善图像分类的示例性方法、系统和装置。一些这样的示例及其组合包括以下示例。

[0091] 示例1是一种用于分析图像的装置,所述装置包括:轮廓引擎,其用于:识别所述图像内的前景轮廓;基于所述前景轮廓的边界来生成边界框;以及生成用于识别前景的单元格的编码的轮廓矩阵;卷积单元格选择器,其用于对所述编码的轮廓矩阵进行卷积以生成卷积的位图矩阵;以及过滤单元格选择器,其用于通过以下操作识别所述卷积的位图矩阵的合格块来改善图像分类效率:保留满足单元格保留门限的、所述卷积的位图矩阵的第一相应单元格;以及去除不满足所述单元格保留门限的、所述卷积的位图矩阵的第二相应单元格。

[0092] 示例2包括示例1中定义的装置,其中,所述过滤单元格选择器用于通过以下操作来生成贡献块矩阵:用保留指示符来对所述贡献块矩阵的所述第一相应单元格进行编码,以及用丢弃指示符来对所述贡献块矩阵的所述第二相应单元格进行编码。

[0093] 示例3包括示例2中定义的装置,还包括边界封装器,其用于识别所述保留指示符中的相应保留指示符的外周边,以生成合格块矩阵。

[0094] 示例4包括示例3中定义的装置,还包括分类接口,其用于用所述合格块矩阵来掩蔽检测窗口,所述掩蔽用于减少要在图像分类中处理的块的数量。

[0095] 示例5包括示例1中定义的装置,还包括分类接口,其用于将所述卷积的位图矩阵的所保留的第一相应单元格提供给定向梯度分类器的直方图。

[0096] 示例6包括示例1中定义的装置,还包括窗口引擎,其用于:将全一矩阵内核卷积到所编码的轮廓的相应单元格;以及计算用于所编码的轮廓的所述相应单元格的所述全一矩阵内核内的前景单元格的总和。

[0097] 示例7包括示例6中定义的装置,其中,所述卷积单元格选择器要用所述卷积的位图矩阵的相应单元格中的所述总和来对所述卷积的位图矩阵进行编码。

[0098] 示例8包括示例1中定义的装置,其中,所述单元格保留门限包括基于矩阵内核大小或重叠步长中的至少一个的值。

[0099] 示例9包括示例2、5、6或8中任一项中定义的装置,还包括分组管理引擎,其用于分配所述图像的定向角以形成图像幅度的直方图。

[0100] 示例10包括示例2、5、6或8中任一项中定义的装置,还包括单元格描述符引擎,其用于将图像单元格的直方图聚合到所述图像的相应块。

[0101] 示例11包括示例2、5、6或8中任一项中定义的装置,还包括边界封装器,其用于识别所述图像的边界位图区域,所述边界位图区域指示要参与图像分类的减少数量的图像单元格。

[0102] 示例12是一种用于分析图像的方法,所述方法包括:通过用处理器来执行指令,识别所述图像内的前景轮廓;通过用所述处理器来执行指令,基于所述前景轮廓的边界来生成边界框;通过用所述处理器来执行指令,生成用于识别前景的单元格的编码的轮廓矩阵;通过用所述处理器来执行指令,对所述编码的轮廓矩阵进行卷积以生成卷积的位图矩阵;通过用所述处理器来执行指令,通过以下操作识别所述卷积的位图矩阵的合格块来改善图像分类效率:(a)保留满足单元格保留门限的、所述卷积的位图矩阵的第

一相应单元格,以及(b)去除不满足所述单元格保留门限的、所述卷积的位图矩阵的第二相应单元格。

[0103] 示例13包括示例12中定义的方法,还包括通过以下操作来生成贡献块矩阵:用保留指示符来对所述贡献块矩阵的所述第一相应单元格进行编码,以及用丢弃指示符来对所述贡献块矩阵的所述第二相应单元格进行编码。

[0104] 示例14包括示例13中定义的方法,还包括:识别所述保留指示符中的相应保留指示符的外周边,以生成合格块矩阵。

[0105] 示例15包括示例14中定义的方法,还包括:用所述合格块矩阵来掩蔽检测窗口,所述掩蔽用于减少要在图像分类中处理的块的数量。

[0106] 示例16包括示例12中定义的方法,还包括:将所述卷积的位图矩阵的所保留的第一相应单元格提供给定向梯度分类器的直方图。

[0107] 示例17包括示例12中定义的方法,还包括:将全一矩阵内核卷积到所编码的轮廓的相应单元格;以及计算用于所编码的轮廓的所述相应单元格的所述全一矩阵内核内的前景单元格的总和。

[0108] 示例18包括示例17中定义的方法,还包括:用所述卷积的位图矩阵的相应单元格中的所述总和来对所述卷积的位图矩阵进行编码。

[0109] 示例19包括示例12中定义的方法,还包括:基于矩阵内核大小或重叠步长中的至少一个来设置所述单元格保留门限的值。

[0110] 示例20包括示例13、16、17或19中任一项定义的方法,还包括分配所述图像的定向角以形成图像幅度的直方图。

[0111] 示例21包括示例13、16、17或19中任一项所述的方法,还包括将图像单元格的直方图聚合到所述图像的相应块。

[0112] 示例22包括示例13、16、17或19中任一项定义的方法,还包括:识别所述图像的边界位图区域,所述边界位图区域指示要参与图像分类的减少数量的图像单元格。

[0113] 示例23是一种包括指令的有形计算机可读介质,所述指令在被执行时使得处理器至少执行以下操作:识别图像内的前景轮廓;基于所述前景轮廓的边界来生成边界框;生成用于识别前景的单元格和背景的单元格的编码的轮廓矩阵;对所述编码的轮廓矩阵进行卷积以生成卷积的位图矩阵;通过以下操作识别所述卷积的位图矩阵的合格块来改善图像分类效率:(a)保留满足单元格保留门限的、所述卷积的位图矩阵的第一相应单元格,以及(b)去除不满足所述单元格保留门限的、所述卷积的位图矩阵的第二相应单元格。

[0114] 示例24包括示例23中定义的计算机可读介质,其中,所述指令在被执行时还使得所述处理器通过以下操作来生成贡献块矩阵:用保留指示符来对所述贡献块矩阵的所述第一相应单元格进行编码,以及用丢弃指示符来对所述贡献块矩阵的所述第二相应单元格进行编码。

[0115] 示例25包括示例24中定义的计算机可读介质,其中,所述指令在被执行时还使得所述处理器识别所述保留指示符中的相应保留指示符的外周边,以生成合格块矩阵。

[0116] 示例26包括示例25中定义的计算机可读介质,其中,所述指令在被执行时还使得所述处理器用所述合格块矩阵来掩蔽检测窗口,所述掩蔽用于减少要在图像分类中处理的块的数量。

[0117] 示例27包括示例23中定义的计算机可读介质,其中,所述指令在被执行时还使得所述处理器将所述卷积的位图矩阵的所保留的第一相应单元格提供给定向梯度分类器的直方图。

[0118] 示例28包括示例23中定义的计算机可读介质,其中,所述指令在被执行时还使得所述处理器执行以下操作:将全一矩阵内核卷积到所编码的轮廓的相应单元格;以及计算用于所编码的轮廓的所述相应单元格的所述全一矩阵内核内的前景单元格的总和。

[0119] 示例29包括示例28中定义的计算机可读介质,其中,所述指令在被执行时还使得所述处理器用所述卷积的位图矩阵的相应单元格中的所述总和来对所述卷积的位图矩阵进行编码。

[0120] 示例30包括示例23中定义的计算机可读介质,其中,所述指令在被执行时还使得所述处理器基于矩阵内核大小或重叠步长中的至少一个来设置所述单元格保留门限的值。

[0121] 示例31包括示例24、27、28或30中任一项中定义的计算机可读介质,其中,所述指令在被执行时还使得所述处理器分配所述图像的定向角以形成图像幅度的直方图。

[0122] 示例32包括示例24、27、28或30中任一项所述的计算机可读介质,其中,所述指令在被执行时还使得所述处理器将图像单元格的直方图聚合到所述图像的相应块。

[0123] 示例33包括示例24、27、28或30中任一项所述的计算机可读介质,其中,所述指令在被执行时还使得所述处理器识别所述图像的边界位图区域,所述边界位图区域指示要参与图像分类的减少数量的图像单元格。

[0124] 示例34是一种用于分析图像的系统,所述系统包括:用于识别所述图像内的前景轮廓的单元;用于基于所述前景轮廓的边界来生成边界框的单元;用于生成用于识别前景的单元格和背景的单元格的编码的轮廓矩阵的单元;用于对所述编码的轮廓矩阵进行卷积以生成卷积的位图矩阵的单元;用于通过以下操作识别所述卷积的位图矩阵的合格块来改善图像分类效率的单元:(a)保留满足单元格保留门限的、所述卷积的位图矩阵的第一相应单元格;以及(b)去除不满足所述单元格保留门限的、所述卷积的位图矩阵的第二相应单元格。

[0125] 示例35包括示例34中定义的系统,还包括用于通过以下操作来生成贡献块矩阵的单元:用保留指示符来对所述贡献块矩阵的所述第一相应单元格进行编码,以及用丢弃指示符来对所述贡献块矩阵的所述第二相应单元格进行编码。

[0126] 示例36包括示例35中定义的系统,还包括用于识别所述保留指示符中的相应保留指示符的外周边以生成合格块矩阵的单元。

[0127] 示例37包括示例36中定义的系统,还包括用于用所述合格块矩阵来掩蔽检测窗口的单元,所述掩蔽用于减少要在图像分类中处理的块的数量。

[0128] 示例38包括示例34中定义的系统,还包括用于将所述卷积的位图矩阵的所保留的第一相应单元格提供给定向梯度分类器的直方图的单元。

[0129] 示例39包括示例34中定义的系统,还包括用于执行以下操作的单元:将全一矩阵内核卷积到所编码的轮廓的相应单元格;以及计算用于所编码的轮廓的所述相应单元格的所述全一矩阵内核内的前景单元格的总和。

[0130] 示例40包括示例39中定义的系统,还包括用于用所述卷积的位图矩阵的相应单元格中的所述总和来对所述卷积的位图矩阵进行编码的单元。

[0131] 示例41包括示例34中定义的系统,还包括:用于基于矩阵内核大小或重叠步长中的至少一个来设置所述单元格保留门限的值的单元。

[0132] 示例42包括示例35、38、39或41中任一项定义的系统,还包括:用于分配所述图像的定向角以形成图像幅度的直方图的单元。

[0133] 示例43包括示例35、38、39或41中任一项所述的方法,还包括用于将图像单元格的直方图聚合到所述图像的相应块的单元。

[0134] 示例44包括示例35、38、39或41中任一项定义的方法,还包括用于识别所述图像的边界位图区域的单元,所述边界位图区域指示要参与图像分类的减少数量的图像单元格。

[0135] 示例45包括示例34中定义的系统,其中,用于识别前景轮廓的所述单元包括轮廓引擎。

[0136] 示例46包括示例34中定义的示例,其中,用于生成边界框的所述单元包括轮廓引擎。

[0137] 示例47包括示例34中定义的系统,其中,用于生成编码的轮廓矩阵的所述单元包括轮廓引擎。

[0138] 示例48包括示例34中定义的系统,其中,用于对所述编码的轮廓矩阵进行卷积的所述单元包括卷积单元格选择器。

[0139] 示例49包括示例34中定义的系统,其中,用于通过识别合格块来改善图像分类效率的所述单元包括过滤单元格选择器。

[0140] 虽然本文已经公开了某些示例性方法、装置和制品,但是本专利的覆盖范围不限于此。相反,本专利覆盖了很大程度上落入本专利权利要求范围内的所有方法、装置和制品。

100 ↘

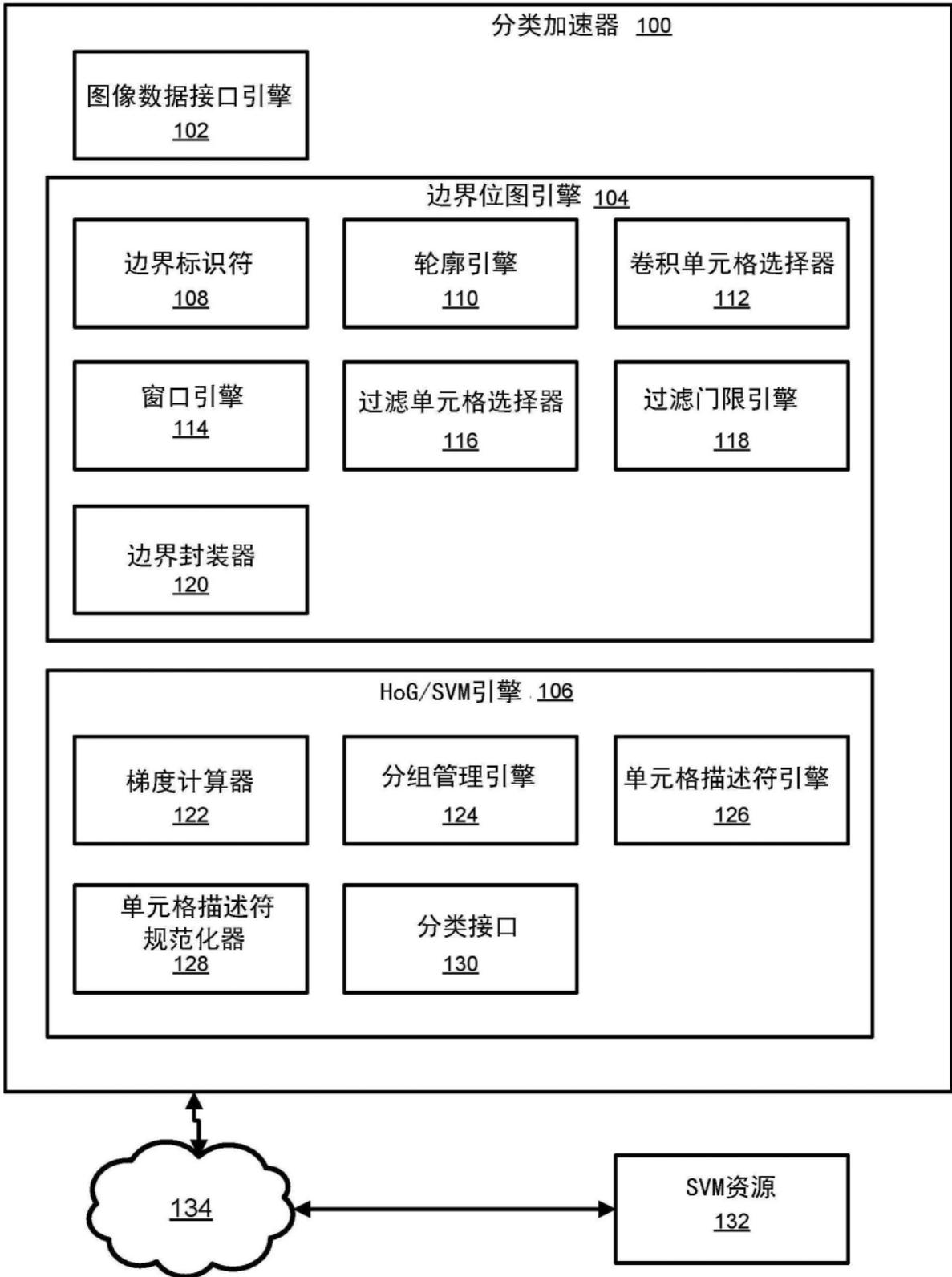


图1

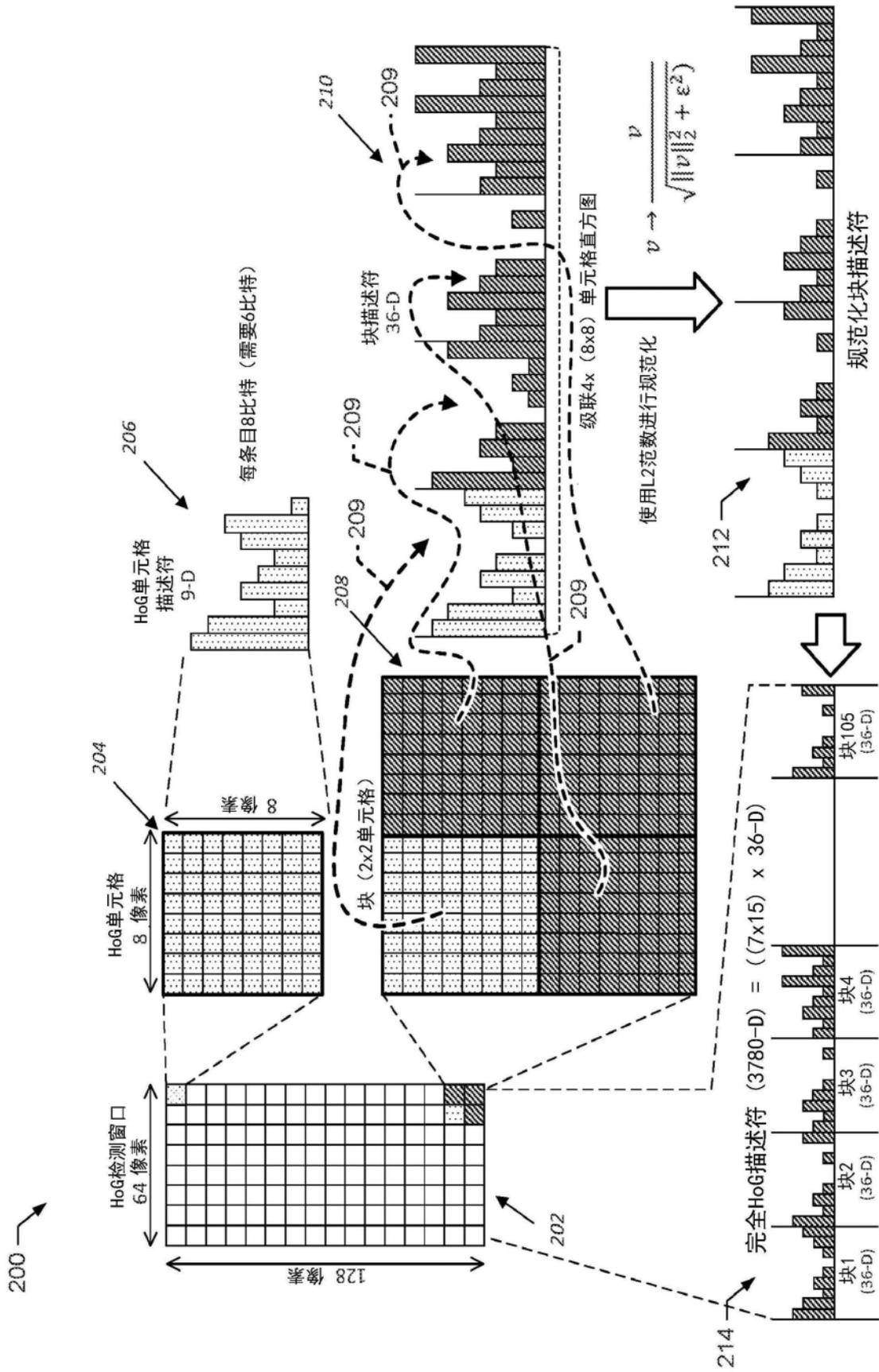


图2

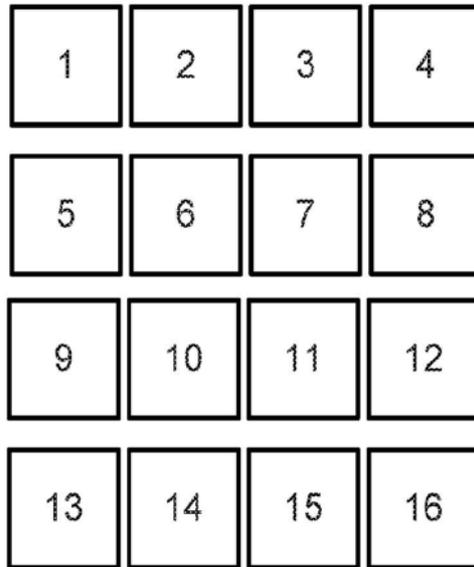


图2B

300 ↘

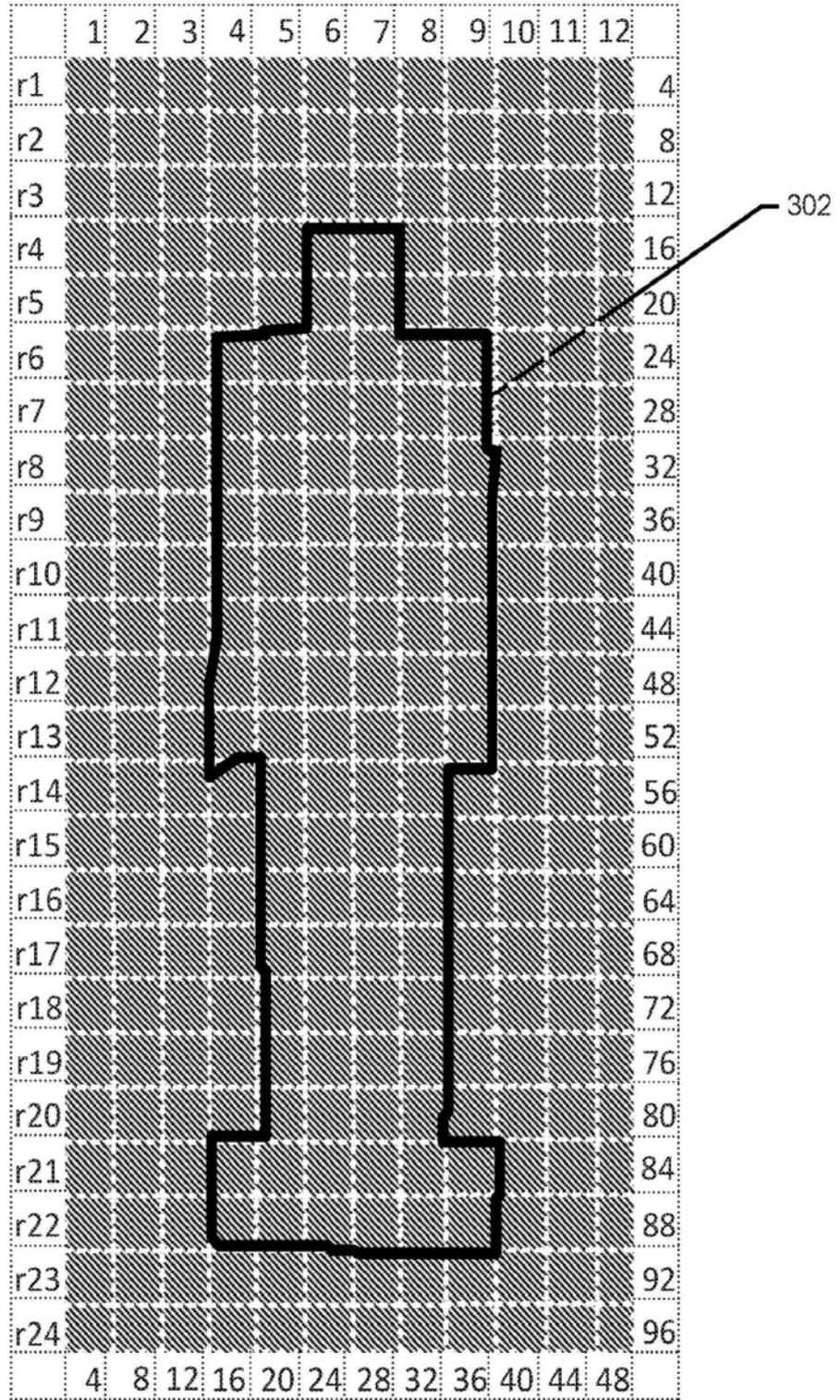


图3

400 →

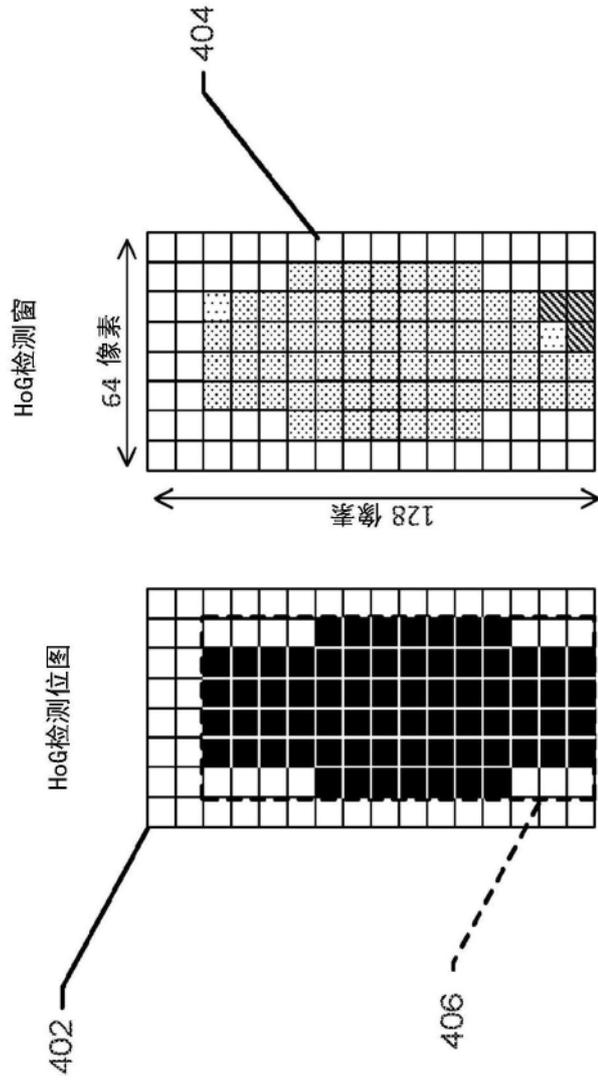


图4

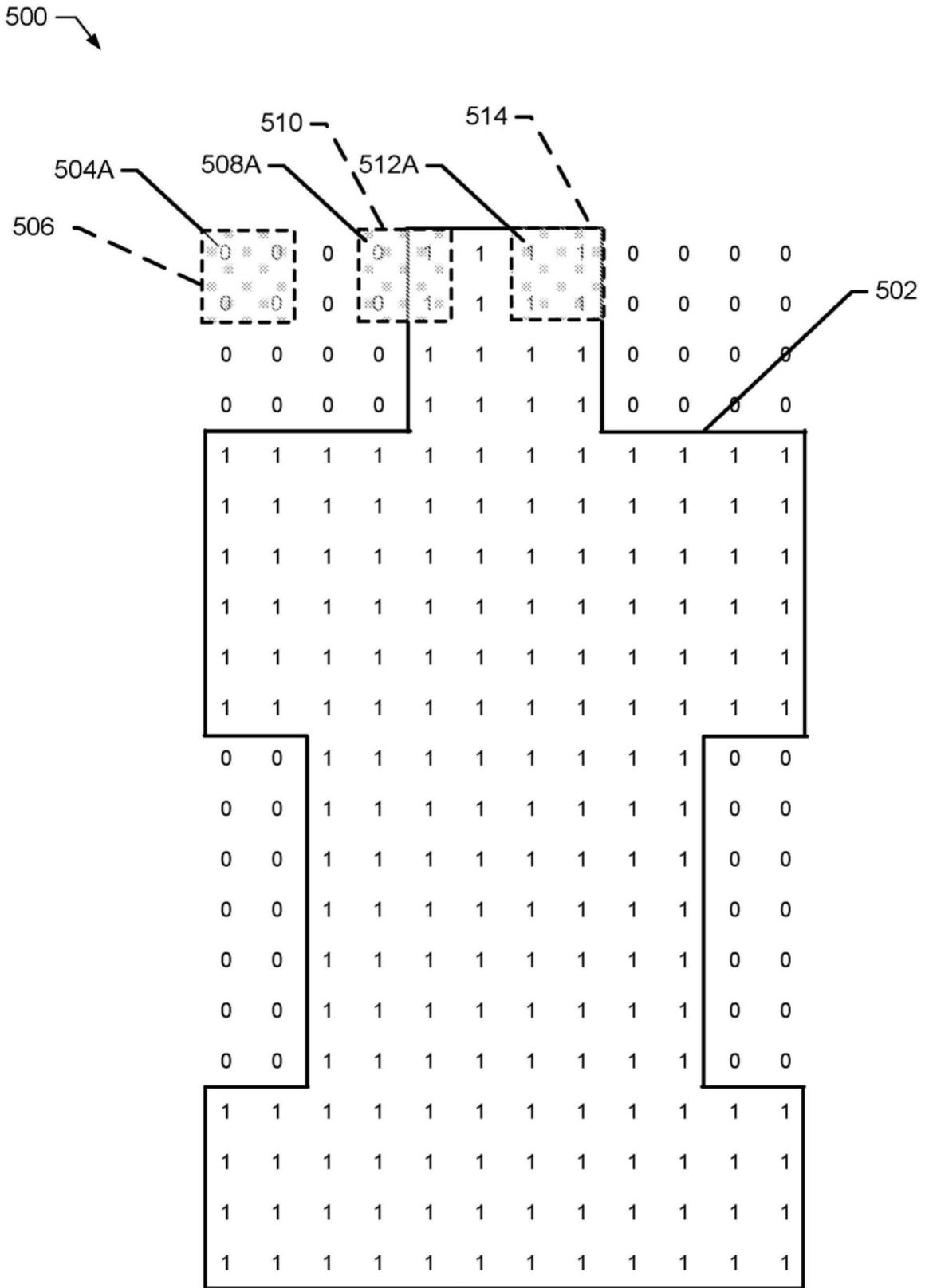


图5

600 ↘

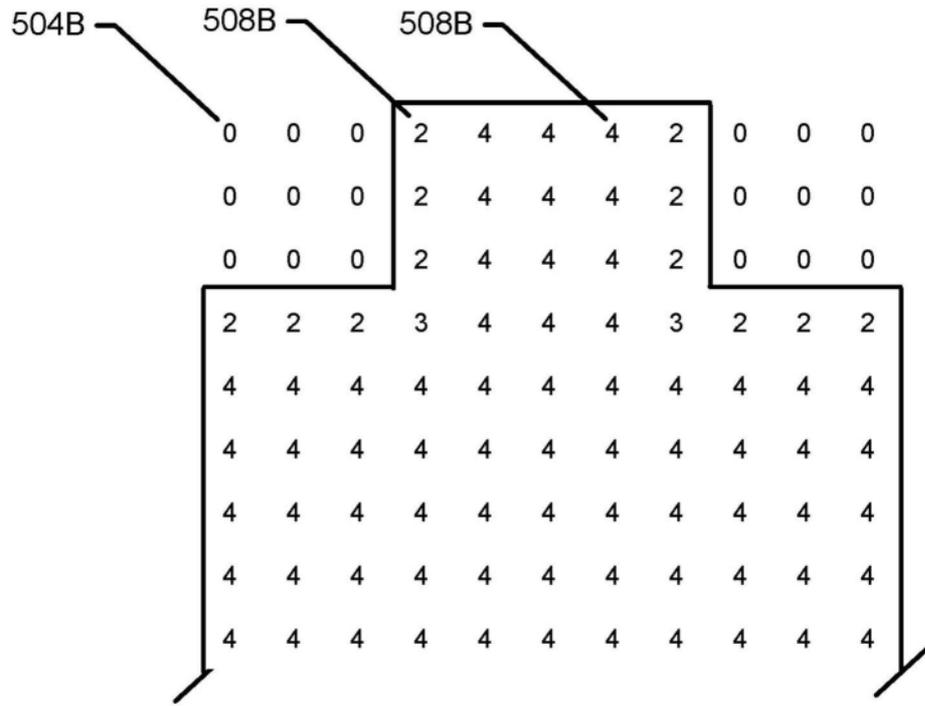


图6

700 ↘

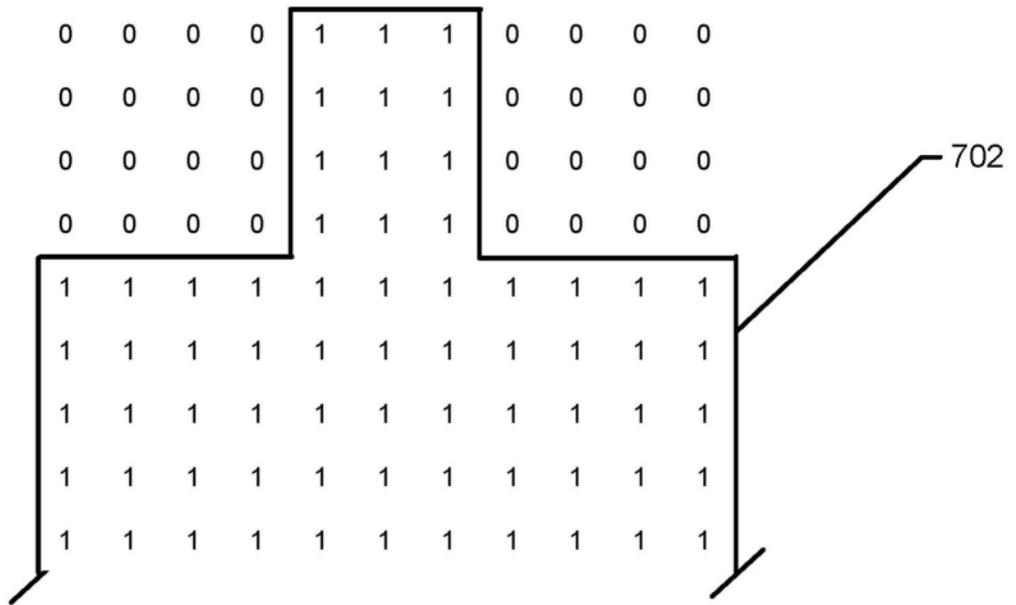


图7

800 ↘

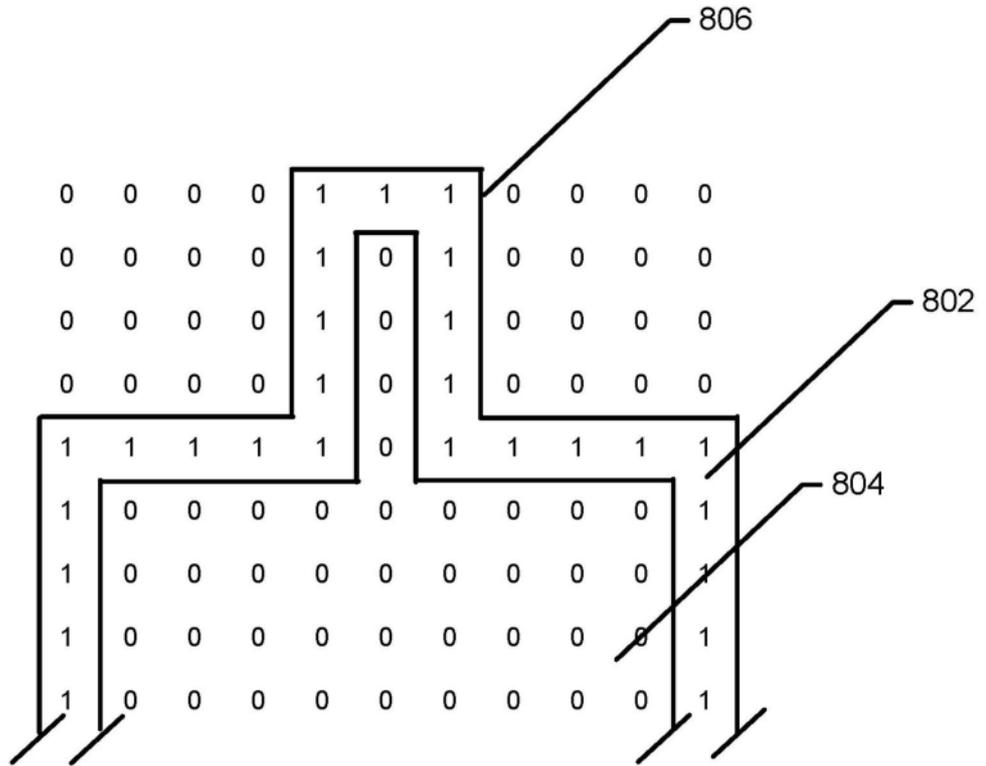


图8

900 ↘

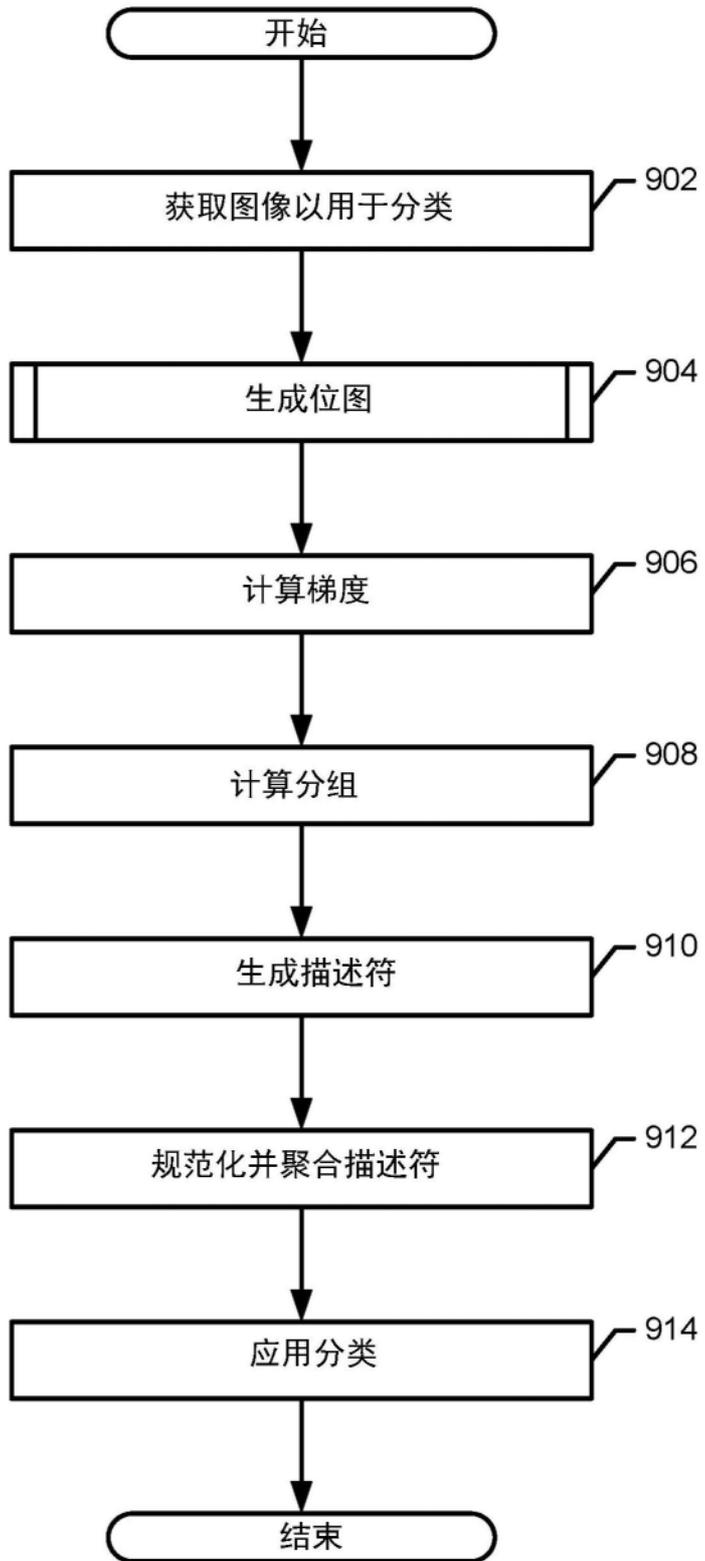


图9

904 ↘

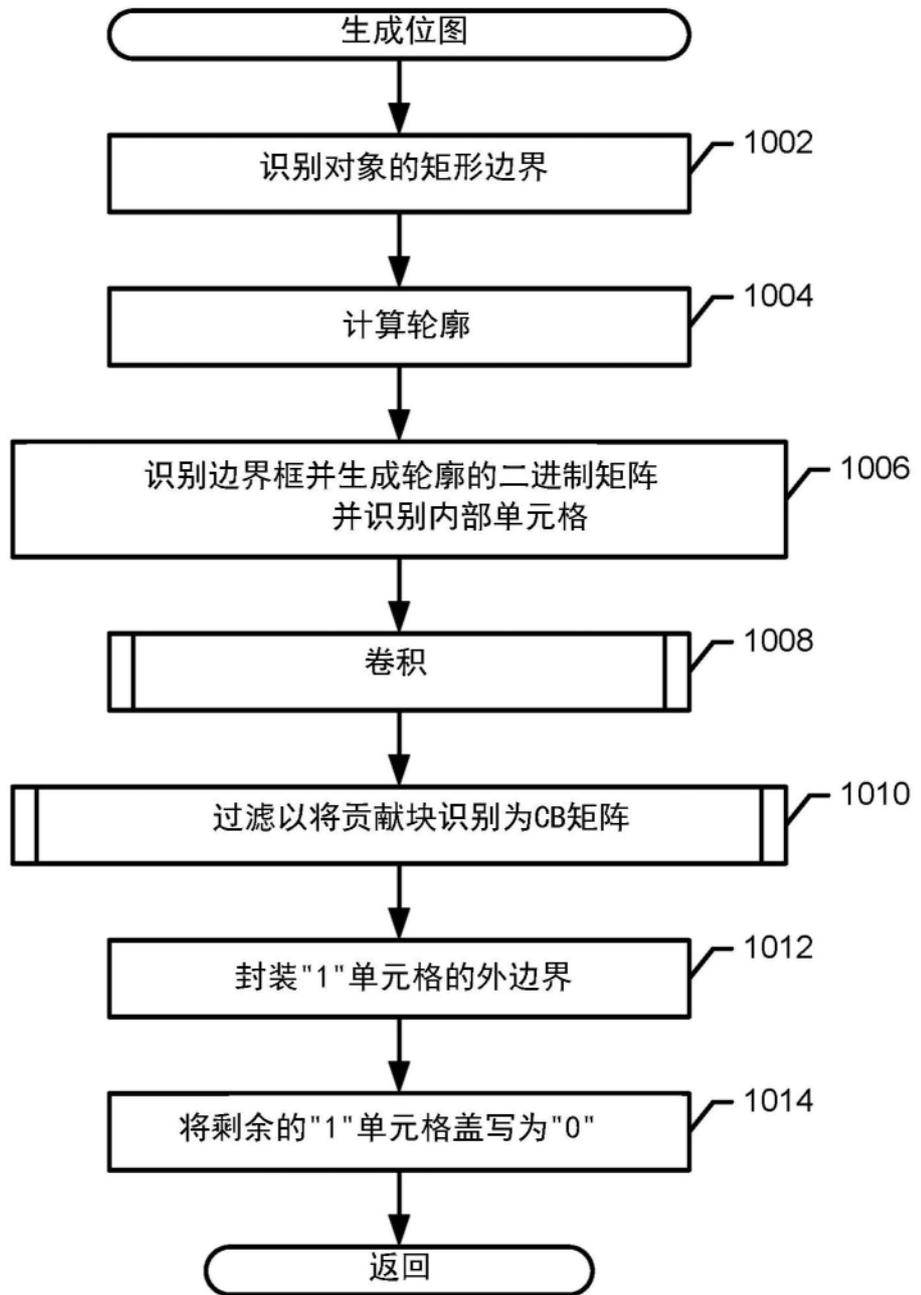


图10

1008 ↘

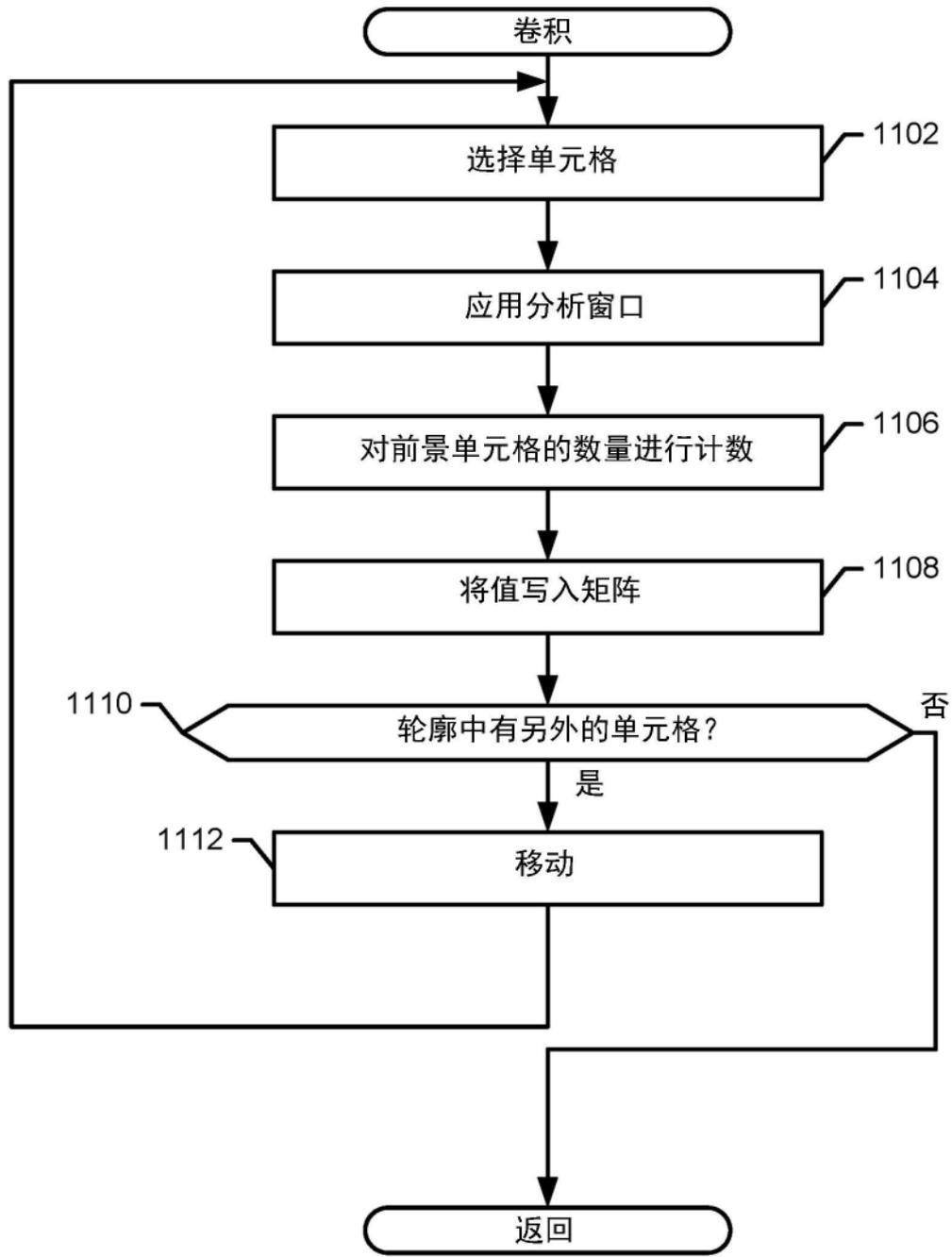


图11

1010 ↘

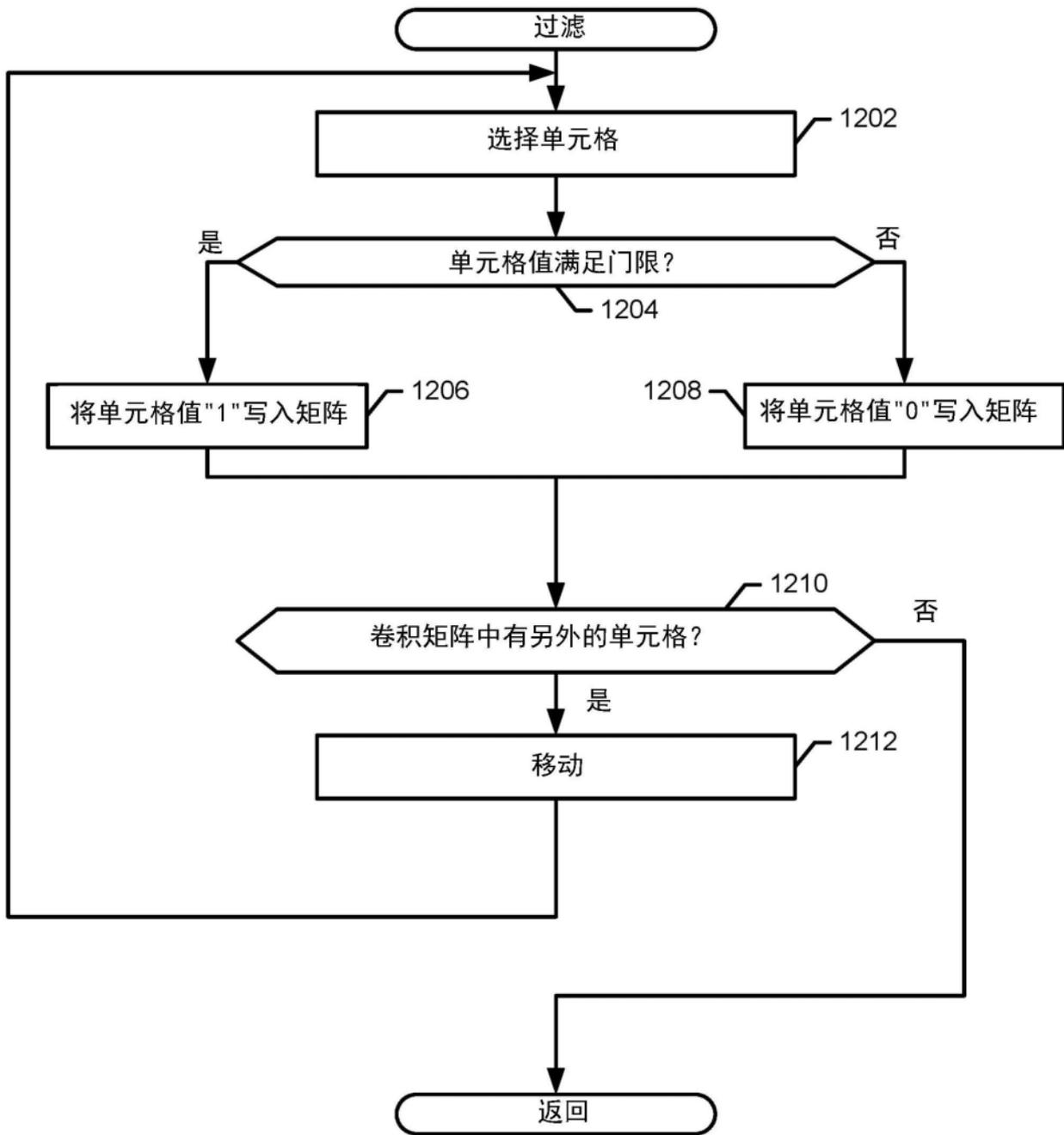


图12

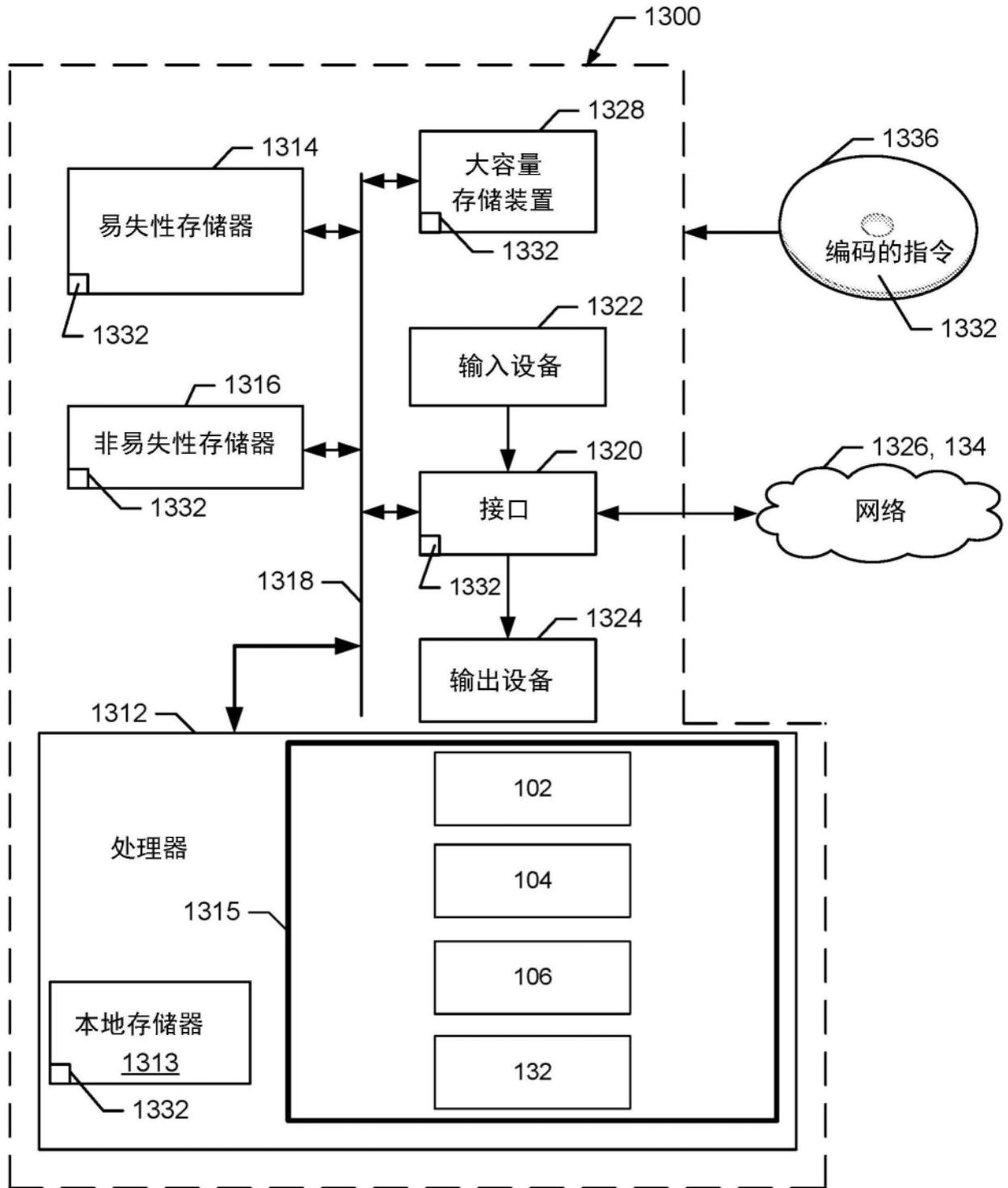


图13