

# (19) 国家知识产权局



## (12) 发明专利



(10) 授权公告号 CN 112805680 B

(45) 授权公告日 2024.08.06

(21) 申请号 202080005517.3

(72) 发明人 K·格里芬

(22) 申请日 2020.06.03

(74) 专利代理机构 中国贸促会专利商标事务所  
有限公司 11038

(65) 同一申请的已公布的文献号

专利代理人 周磊

申请公布号 CN 112805680 A

(51) Int.CI.

G06F 9/455 (2018.01)

G06F 9/50 (2006.01)

G06T 1/00 (2006.01)

G06T 7/50 (2017.01)

H04L 41/0806 (2022.01)

H04L 41/08 (2022.01)

H04L 41/0895 (2022.01)

H04L 41/22 (2022.01)

(43) 申请公布日 2021.05.14

(30) 优先权数据

16/527,921 2019.07.31 US

(85) PCT国际申请进入国家阶段日

2021.04.02

(86) PCT国际申请的申请数据

PCT/US2020/035882 2020.06.03

(87) PCT国际申请的公布数据

W02021/021281 EN 2021.02.04

(73) 专利权人 甲骨文国际公司

(56) 对比文件

US 2018/0373961 A1, 2018.12.27

地址 美国加利福尼亚

审查员 殷婷

权利要求书3页 说明书11页 附图18页

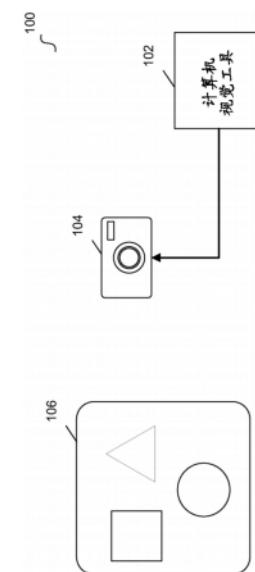
### (54) 发明名称

使用计算机视觉进行供给

### (57) 摘要

本公开的实施例一般而言针对用于使用计算机视觉来供给计算资源的系统和方法。可以接收包括多个视觉元素的图像。图像内的视觉元素可以被识别为表示网络元素的对象，其中识别出的对象表示网络元素类型的网络元素，并且网络元素类型包括数据库、负载均衡器、子网或虚拟机实例中的一种或多种，并且表示网络架构的识别出的对象的布置被识别。通过供给包括网络元素类型中的一种或多种网络元素类型的网络元素可以使得计算资源与识别出的对象对应地被供给，其中所述供给包括基于识别出的布置为被供给的网络元素生成网络架构。

B CN 112805680



1. 一种用于使用计算机视觉来供给计算资源的方法,所述方法包括:  
接收包括多个视觉元素的图像;

在用户界面处接收与所述图像相关联的包括一组网络元素数量的输入,每个网络元素数量与网络元素类型相关联,网络元素类型包括数据库、负载均衡器、子网和虚拟机实例中的一种或多种;

将图像内的视觉元素识别为表示网络元素的对象,其中,

所述对象基于多个对象定义被识别为所述网络元素类型之一,每个对象定义与所述网络元素类型中的至少一种相关联,对象定义至少包括检测到的边的数量,

至少一个对象定义包括边的数量和所述至少一个对象定义的对象被包含在两个检测到的对象内的状态,

作为输入被接收的所述一组网络元素数量用于验证每种网络元素类型的识别出的对象的数量,以及

表示网络架构的识别出的对象的布置被识别;以及

通过供给包括所述网络元素类型中的一种或多种网络元素类型的网络元素,使得计算资源与识别出的对象对应地被供给,其中所述供给包括基于识别出的布置为被供给的网络元素生成网络架构。

2. 如权利要求1所述的方法,其中对象定义包括三角形、矩形、正方形、圆形或五边形中的一个或多个。

3. 如权利要求1所述的方法,其中被供给的网络元素对应于识别出的对象,使得被供给的每种网络元素类型的数量对应于与每种网络元素类型相关联的识别出的对象的数量。

4. 如权利要求3所述的方法,其中图像的第一部分至少包括一个第一视觉元素,所述第一视觉元素至少包含两个第二视觉元素,并且基于图像的第一部分的用于被供给的网络元素的网络架构包括包含有两个子网的网络。

5. 如权利要求4所述的方法,其中与第一视觉元素对应的第一识别出的对象与网络相关联,与第二视觉元素对应的第二识别出的对象与子网相关联,并且第一对象和第二对象之间的识别出的布置与包括子网的网络相关联。

6. 如权利要求5所述的方法,其中图像的第二部分至少包括第三视觉元素,所述第三视觉元素可视地连接到第二视觉元素,并且基于图像的第二部分的用于被供给的网络元素的网络架构包括均衡所述两个子网之间的负载的负载均衡器。

7. 如权利要求6所述的方法,其中与第三视觉元素对应的第三识别出的对象与负载均衡器相关联,并且第三对象和第二对象之间的识别出的布置与负载均衡器和所述两个子网之间的连接相关联。

8. 如权利要求1所述的方法,其中供给计算资源包括根据云基础设施内的网络架构来供给网络元素。

9. 如权利要求1所述的方法,还包括:

基于识别出的对象生成多个网络元素配置文件,其中配置文件用于供给与识别出的对象相关联的网络元素,并且基于识别出的对象的识别出的布置来定义网络元素配置文件中的一个或多个网络元素配置文件。

10. 如权利要求1所述的方法,还包括:

使用一个或多个变换来处理图像,其中基于处理后的图像执行对象识别。

11. 如权利要求1所述的方法,其中所述一组网络元素数量是在对象识别之前作为输入被接收的。

12. 如权利要求11所述的方法,其中第一对象定义包括边的第一数量和第一对象定义的对象被包含在一个检测到的对象内的状态,并且第二对象定义包括边的第一数量和第二对象定义的对象被包含在两个检测到的对象内的状态,第一对象定义和第二对象定义与不同的网元素类型相关联。

13. 如权利要求11所述的方法,其中作为输入被接收的所述一组网络元素数量指示图像中的视觉元素的期望的层次结构,并且基于期望的层次结构使用对象定义来识别对象。

14. 一种用于使用计算机视觉来供给计算资源的系统,所述系统包括:

处理器;以及

存储器,存储由所述处理器执行的指令,所述指令将所述处理器配置为:

接收包括多个视觉元素的图像;

在用户界面处接收与所述图像相关联的包括一组网络元素数量的输入,每个网络元素数量与网络元素类型相关联,网络元素类型包括数据库、负载均衡器、子网和虚拟机实例中的一种或多种;

将图像内的视觉元素识别为表示网络元素的对象,其中,

所述对象基于多个对象定义被识别为所述网络元素类型之一,每个对象定义与所述网络元素类型中的至少一种相关联,对象定义至少包括检测到的边的数量,

至少一个对象定义包括边的数量和所述至少一个对象定义的对象被包含在两个检测到的对象内的状态,

作为输入被接收的所述一组网络元素数量用于验证每种网络元素类型的识别出的对象的数量,以及

表示网络架构的识别出的对象的布置被识别;以及

通过供给包括所述网络元素类型中的一种或多种网络元素类型的网络元素,使得计算资源与识别出的对象对应地被供给,其中所述供给包括基于识别出的布置为被供给的网络元素生成网络架构。

15. 如权利要求14所述的系统,其中所述指令将所述处理器配置为:

基于识别出的对象生成多个网络元素配置文件,其中配置文件用于供给与识别出的对象相关联的网络元素,并且基于识别出的对象的识别出的布置来定义网络元素配置文件中的一个或多个网络元素配置文件。

16. 如权利要求14所述的系统,其中所述一组网络元素数量是在对象识别之前作为输入被接收的。

17. 如权利要求16所述的系统,其中第一对象定义包括边的第一数量和第一对象定义的对象被包含在一个检测到的对象内的状态,并且第二对象定义包括边的第一数量和第二对象定义的对象被包含在两个检测到的对象内的状态,第一对象定义和第二对象定义与不同的网元素类型相关联。

18. 如权利要求16所述的系统,其中作为输入被接收的所述一组网络元素数量指示图像中的视觉元素的期望的层次结构,并且基于期望的层次结构使用对象定义来识别对象。

19.一种非暂态计算机可读介质,其上存储有指令,所述指令在由处理器执行时,使所述处理器使用计算机视觉来供给计算资源,其中,所述指令在执行时,使所述处理器:

接收包括多个视觉元素的图像;

在用户界面处接收与所述图像相关联的包括一组网络元素数量的输入,每个网络元素数量与网络元素类型相关联,网络元素类型包括数据库、负载均衡器、子网和虚拟机实例中的一种或多种;

将图像内的视觉元素识别为表示网络元素的对象,其中,

所述对象基于多个对象定义被识别为所述网络元素类型之一,每个对象定义与所述网络元素类型中的至少一种相关联,对象定义至少包括检测到的边的数量,

至少一个对象定义包括边的数量和所述至少一个对象定义的对象被包含在两个检测到的对象内的状态,

作为输入被接收的所述一组网络元素数量用于验证每种网络元素类型的识别出的对象的数量,以及

表示网络架构的识别出的对象的布置被识别;以及

通过供给包括网络元素类型中的一种或多种网络元素类型的网络元素,使得计算资源与识别出的对象对应地被供给,其中所述供给包括基于识别出的布置为被供给的网络元素生成网络架构。

## 使用计算机视觉进行供给

### 技术领域

[0001] 本公开的实施例一般而言涉及使用计算机视觉来供给计算资源。

### 背景技术

[0002] 由现代服务提供商实现的计算服务经常被要求是鲁棒、灵活和高度可用的。例如，对计算的现代需求通常来自与效率低下或不可用的计算资源相关联的机会成本。一些设置允许按需供给计算资源，诸如基于云的服务提供商，但是这种供给通常涉及多步骤处理、手动工作和其它不便之处。鉴于对现代计算的严格要求，一种高效且快速提供可用计算资源的工具对该工具的用户是有益的。

### 发明内容

[0003] 本公开的实施例通常针对用于使用计算机视觉来供给计算资源的系统和方法。可以接收包括多个视觉元素的图像。图像内的视觉元素可以被识别为表示网络元素的对象，其中识别出的对象表示网络元素类型的网络元素，并且网络元素类型包括数据库、负载均衡器、子网、或虚拟机实例中的一种或多种，并且表示网络架构的识别出的对象的布置被识别。通过供给包括网络元素类型中的一个或多个网络元素类型的网络元素，可以使计算资源与识别出的对象对应地被供给，其中所述供给包括基于识别出的布置为被供给的网络元素生成网络架构。

[0004] 实施例的特征和优点在下面的描述中阐述，或者将从描述中变得明显，或者可以通过实践本公开而获知。

### 附图说明

[0005] 根据以下结合附图对优选实施例的详细描述，进一步的实施例、细节、优点和修改将变得明显。

[0006] 图1图示了根据示例实施例的用于使用计算机视觉来供给计算资源的系统。

[0007] 图2图示了根据示例实施例的可操作地耦合到系统的计算设备的框图。

[0008] 图3图示了根据示例实施例的计算架构的示图。

[0009] 图4图示了根据示例实施例的计算架构的另一个示图。

[0010] 图5图示了根据示例实施例的计算架构的图像。

[0011] 图6图示了根据示例实施例的用于计算架构的输入。

[0012] 图7图示了根据示例实施例的计算架构的处理后的图像。

[0013] 图8图示了根据示例实施例的在计算架构的图像中识别出的对象。

[0014] 图9A-9B图示了根据示例实施例的用于供给计算架构的配置文件。

[0015] 图10-16图示了根据示例实施例的与供给计算架构相关的用户界面。

[0016] 图17图示了根据示例实施例的用于使用计算机视觉来供给计算资源的流程图。

## 具体实施方式

[0017] 实施例使用计算机视觉来供给计算资源。例如,可以接收包括计算机架构的示图的图像。该示图可以包括表示网络计算元素的视觉元素,网络计算元素诸如虚拟机、负载均衡器、数据库、子网和任何其它合适的计算元素。例如,可以针对接收到的图像内的视觉元素检测多个轮廓,并且可以基于检测到的多个轮廓来识别对象。

[0018] 在一些实施例中,还可以检测用于视觉元素的布置,其中该布置表示计算资源的架构。基于检测到的对象及其布置,可以供给计算资源。例如,可以供给一个或多个虚拟机、负载均衡器、数据库、子网和任何其它合适的计算元素。

[0019] 在一些实施例中,可以使用一个或多个生成的配置文件来供给计算资源。例如,可以基于检测到的对象及其布置来生成配置文件。在一些实施例中,供给工具可以被配置为根据一个或多个生成的配置文件来供给计算资源。例如,检测到的布置可以用于生成配置文件,使得基于检测到的布置的网络架构是由配置文件定义的。基于这些定义,可以根据网络架构组织随后供给的计算资源。

[0020] 现在将详细参考本公开的实施例,其示例在附图中示出。在下面的详细描述中,阐述了许多具体细节以便提供对本公开的透彻理解。但是,对于本领域的普通技术人员来说将明显的是,可以在没有这些具体细节的情况下实践本公开。在其它情况下,未详细描述众所周知的方法、过程、组件和电路,以免不必要地混淆实施例的各个方面。只要有可能,相同的参考数字将用于相同的元素。

[0021] 图1图示了根据示例实施例的用于使用计算机视觉来供给计算资源的系统。系统100包括计算机视觉工具102、图像捕获设备104和示图106。在一些实施例中,示图106可以表示网络架构。例如,可以定义视觉元素和网络计算元素之间的预定对应关系,并且示图106可以包括这些视觉元素的布置。在一些实施例中,这些视觉元素的布置可以表示用于示图106中表示的网络元素的网络架构。

[0022] 在一些实施例中,图像捕获设备104可以用于捕获示图106的图像。例如,示图106可以被描绘在诸如白板、物理纸张、黑板、显示器或任何其它合适的表面之类的表面上。在一些实施例中,可以将示图106作为头脑风暴会话、会议或任何其它讨论(例如,初步讨论或计划讨论)的一部分来生成。例如,团队成员可以讨论项目、服务、产品或其它实现的计算需求,并且示图106可以由团队基于讨论生成(例如,手绘)。然后图像捕获设备104可以用来捕获示图106的图像。其它实施例可以包括用于生成示图106和/或捕获示图106的图像的其它技术。

[0023] 在一些实施例中,捕获的图像可以由计算机视觉工具102接收。例如,图像捕获设备104可以将图像提供给计算机视觉工具102(例如,将图像传输到实现计算机视觉工具102的计算设备)。实施例包括用于接收计算机网络架构的图像的任何其它合适的技术。计算机视觉工具102可以处理图像并识别图像内的对象。例如,可以使用一个或多个过滤器或变换来变换图像,并且计算机视觉工具102可以基于处理后的图像来识别对象。在一些实施例中,识别出的对象可以是示图106内的视觉元素,诸如包括与网络计算元素的预定对应关系的视觉元素。还可以识别对象的布置,其中该布置表示用于网络计算元素的网络架构。实施例实现对象检测和/或对象识别功能和算法,并且这些术语将在整个本公开中互换使用。

[0024] 在一些实施例中,可以供给与检测到/识别出的对象对应的网络计算元素。例如,

计算机视觉工具102可以使得供给这些网络计算元素。在一些实施例中，一个或多个配置文件可以由计算机视觉工具102生成，其中配置文件用于供给网络计算元素。例如，供给工具可以被配置为根据所生成的配置文件来供给计算资源。在一些实施例中，被供给的计算资源可以是云计算环境的一部分，并且供给工具可以供给云中的网络计算元素。例如，网络计算元素可以包括虚拟机、子网、负载均衡器、数据库等中的一个或多个。

[0025] 在一些实施例中，所生成的配置文件可以基于处理后的图像中检测到的对象的检测到的布置。例如，检测到的对象的布置可以表示在两个子网之间均衡负载的负载均衡器。在一些实施例中，可以配置所生成的配置文件，使得被供给的计算元素包括由检测到的布置表示的关系（例如，根据配置文件供给的负载均衡器被配置为在两个子网之间均衡负载）。

[0026] 实施例提供了优于传统计算资源供给技术的许多优点。传统的计算资源供给工具通常需要繁琐的需求定义，诸如明确定义计算资源的形状、虚拟机或数据库的实例、网络元素（诸如负载均衡器）之间的关系、以及其它需求。例如，经常需要用户浏览许多用户界面页面来输入所有所需的信息。在其它时候，要求用户生成包含信息的数据文件。

[0027] 本公开的实施例通过减少用户的工作和需求定义的复杂性提供了提高的可用性速度和效率。例如，视觉元素和网络元素之间的预定对应关系使得能够使用简单示图来定义计算资源需求。在许多情况下，简单图示是与供给分开的流程（诸如用于确定将如何实现服务或产品的业务流程（或其它技术流程））的一部分。实施例可以重用这些简单示图来进行供给，从而减少与用于定义资源需求的传统技术相关联的过多工作。

[0028] 另外，由一些实施例执行的这些视觉元素的检测到的布置进一步降低了复杂性并提高了可用性速度。例如，实施例使得能够供给复杂的计算资源的网络，诸如具有多个子网、连接和异构元素的网络。供给计算机视觉工具的实施例提供了鲁棒的解决方案，即使对于复杂的计算机资源实现，该解决方案也可以改善整个供给处理。

[0029] 传统供给的繁琐性质对于云资源可能尤其麻烦。例如，现代云服务提供商被要求提供高效、快速且灵活的技术来实现计算资源。这通常是由于对组织而言可能是高成本（例如，花费时间、金钱、品牌声誉等）的缓慢、低效或繁琐的技术的机会成本造成的。因此，鉴于对现代云服务提供商的严格要求，用于供给云资源的实施例可能特别有用。

[0030] 图2是根据实施例的计算机服务器/系统200的框图。系统200的全部或部分可以用于实现图1中所示的任何元件。如图2中所示，系统200可以包括总线设备212和/或（一个或多个）其它通信机制，该其它通信机制被配置为在系统200的各种组件（诸如处理器222和存储器214）之间传送信息。此外，通信设备220可以通过对要通过网络（未示出）从处理器222发送到另一个设备的数据进行编码以及通过网络从另一个系统为处理器222接收的数据进行解码来实现处理器222与其它设备之间的连接性。

[0031] 例如，通信设备220可以包括被配置为提供无线网络通信的网络接口卡。可以使用各种无线通信技术，包括红外、无线电、**蓝牙®**、Wi-Fi和/或蜂窝通信。替代地，通信设备220可以被配置为提供（一个或多个）有线网络连接，诸如以太网连接。

[0032] 处理器222可以包括一个或多个通用或专用处理器，以执行系统200的计算和控制功能。处理器222可以包括单个集成电路，诸如微处理设备，或者可以包括协同工作以完成处理器222的功能的多个集成电路设备和/或电路板。此外，处理器222可以执行存储在存储

器214内的计算机程序,诸如操作系统215、计算机视觉工具216和其它应用218。

[0033] 系统200可以包括用于存储信息和由处理器222执行的指令的存储器214。存储器214可以包含用于检索、呈现、修改和存储数据的各种组件。例如,存储器214可以存储在由处理器222执行时提供功能的软件模块。这些模块可以包括为系统200提供操作系统功能的操作系统215。这些模块可以包括操作系统215、实现本文公开的计算机视觉和供给功能的计算机视觉工具216,以及其它应用模块218。操作系统215为系统200提供操作系统功能。在一些情况下,计算机视觉工具216可以被实现为内存(*in-memory*)配置。在一些实现中,当系统200执行计算机视觉工具216的功能时,它实现执行本文公开的功能的非常规专用计算机系统。

[0034] 非暂态存储器214可以包括可以由处理器222访问的各种计算机可读介质。例如,存储器214可以包括随机存取存储器(“RAM”)、动态RAM(“DRAM”)、静态RAM(“SRAM”)、只读存储器(“ROM”)、闪存、高速缓存存储器和/或任何其它类型的非瞬态计算机可读介质。处理器222还经由总线212耦合到显示器224,诸如液晶显示器(“LCD”)。键盘226和光标控制设备228(诸如计算机鼠标)还耦合到通信设备212以使用户能够与系统200接口。

[0035] 在一些实施例中,系统200可以是较大系统的一部分。因此,系统200可以包括一个或多个附加功能模块218以包括附加功能。其它应用模块218可以包括例如**Oracle®**云基础设施、**Oracle®**云平台、**Oracle®**云应用的各种模块。计算机视觉工具216、其它应用模块218和系统200的任何其它合适的组件可以包括HashiCorp的Terraform的各种模块、OpenCV、其它合适的开源工具或其它合适的**Oracle®**产品或服务。

[0036] 数据库217耦合到总线212,以向模块216和218提供集中存储,并存储例如计算机视觉工具216接收到的数据或其它数据源。数据库217可以将数据存储在逻辑上相关的记录或文件的集成集合中。数据库217可以是操作数据库、分析数据库、数据仓库、分布式数据库、最终用户数据库、外部数据库、导航数据库、内存数据库、面向文档的数据库、实时数据库、关系数据库、面向对象的数据库、非关系数据库、NoSQL数据库、**Hadoop®**分布式文件系统(“HDFS”)或本领域已知的任何其它数据库。

[0037] 虽然被示出为单个系统,但系统200的功能可以被实现为分布式系统。例如,存储器214和处理器222可以分布在共同表示系统200的多个不同计算机上。在一个实施例中,系统200可以是设备(例如,智能电话、平板电脑、计算机等)的一部分。在实施例中,系统200可以与设备分离,并且可以远程地为设备提供所公开的功能。另外,可以不包括系统200的一个或多个组件。例如,对于作为用户或消费者设备的功能,系统200可以是智能电话或包括处理器、存储器和显示器、不包括图2中所示的一个或多个其它组件、并且包括图2中未示出的附加组件(诸如天线、收发器或任何其它合适的无线设备组件)的其它无线设备。另外,当被实现为执行本文中公开的功能时,系统200是专用于提供需求预测的专用计算机。

[0038] 实施例使用计算机视觉来供给计算资源。再次参考图1,计算机视觉工具102可以接收具有表示网络计算元素的示图的图像。图3图示了根据示例实施例的计算架构的示图。例如,示图302可以包括视觉元素,诸如形状和连接,其表示计算网络元素和计算架构。例如,示图302可以被描绘在诸如白板、物理纸张、黑板、显示器或任何其它合适的表面之类的表面上。

[0039] 在一些实施例中,可以定义视觉元素和网络计算元素之间的预定对应关系。例如,在示图302中表示的视觉元素可以包括形状,其中不同的形状可以对应于不同的网络计算元素。在一些实施例中,视觉元素及其布置可以表示网络架构。

[0040] 图4图示了根据示例实施例的计算架构。例如,示图302可以描绘网络架构400,其可以包括负载均衡器402、网络404、子网406和408、虚拟机410和412、连接414和416以及数据库418。在一些实施例中,这些网络元素由示图302中描绘的形状表示。例如,负载均衡器402可以由三角形表示,网络404可以由矩形表示,子网406和408可以由包含在另一个所描绘的网络元素(例如,网络404)内的矩形表示、虚拟机410可以由包含在另一个所描绘的网络元素(例如,子网406)内的矩形来表示,虚拟机412可以由包含在另一个所描绘的网络元素(例如,子网408)内的五边形来表示,并且数据库418可以由圆形表示。在一些实施例中,这些表示是基于形状(和/或形状的方位,诸如被包含在另一个形状内)和它们表示的特定网络元素之间的预定对应关系来实现的。

[0041] 在一些实施例中,形状可以表示特定的网络计算实体,诸如数据库或虚拟机的类型。例如,表示数据库418的圆形可以对应于自主数据库。其它形状可以表示其它类型的数据库。另外,包含在另一个视觉元素内的矩形可以表示虚拟机410或第一类型的虚拟机,而包含在另一个视觉元素内的五边形可以表示虚拟机412或第二类型的虚拟机。在一些实施例中,取决于系统的配置方式,虚拟机的差异可以从被供给的CPU内核的数量到虚拟机上运行的不同操作系统而变化。例如,虚拟机410可以是VM Standard2.4,而虚拟机412可以是VM Standard2.8。在一些实施例中,虚拟机选项由云供给商确定和/或提供。虽然图3和图4演示了形状和网络元素之间的特定对应关系,但是可以实现任何其它合适的定义/表示。

[0042] 在一些实施例中,示图302包括视觉元素的布置。例如,视觉元素的布置可以描绘架构400。视觉元素可以显示负载均衡器402使用连接414和416连接到子网406和408。该描绘可以表示负载均衡器402在子网406和408之间均衡负载。在所示的实施例中,子网406包括虚拟机410,并且子网408包括虚拟机412。因此,在所示的实施例中,负载均衡器402有效地均衡了虚拟机410和虚拟机412之间的负载。

[0043] 再次参考图1,图像捕获设备104可以用于捕获示图302的图像。在一些实施例中,捕获的图像可以被提供/传输到计算机视觉工具102。可以使用用于在计算机视觉工具102处接收示图的图像的任何其它合适的实现方式。

[0044] 图5图示了根据示例实施例的计算架构的图像。例如,图像502可以是图3中的示图302的图像,图3中的示图302又可以表示图4中的架构400。换句话说,图像502可以包括示图,该示图包括表示网络计算元素的视觉元素和表示网络架构的视觉元素的布置。在一些实施例中,图像502在图1的计算机视觉工具102处被接收/或被传输到图1的计算机视觉工具102。

[0045] 在一些实施例中,计算机视觉工具102可以是接收图像、识别图像中的对象并使得计算机资源被供给的软件。例如,计算机视觉工具可以是接收图5的图像502作为输入的编译代码(例如,可执行文件)或脚本代码。在一些实施例中,可以与图像502一起接收附加信息。

[0046] 图6图示了根据示例实施例的用于计算架构的输入。

[0047] 在一些实施例中,输入参数602可以包括在图像502中表示的网络元素的数量。例

如,输入参数502可以被分解成网络元素类型(例如,网络、子网、虚拟机实例、负载均衡器、连接、数据库等)。在所示的实施例中,图像502的输入参数602是一个网络、两个子网、两个虚拟机实例、一个负载均衡器、两个连接和一个数据库。图6还将计算机视觉工具102描绘为接受输入参数602的脚本(例如,Python脚本)。计算机视觉工具102可以是任何其它形式的软件、硬件或任何组合。

[0048] 在一些实施例中,一旦图像502和输入参数602被计算机视觉工具102接收到,图像502就被处理。例如,可以将一个或多个过滤器或掩模应用于图像502以变换图像。在一些实施例中,可以使用卷积或互相关函数来调整图像502的像素值以生成处理后的图像。在一些实现方式中,可以(例如,基于多个滤波器、掩模、卷积、互相关以及其它图像处理技术)生成多个处理后的图像。一些实施例使用一个或多个开源工具(例如,OpenCV)或其它图像处理工具来实现图像处理。

[0049] 图7图示了根据示例实施例的计算架构的处理后的图像。例如,可以实现一种或多种处理技术以从图像502生成处理后的图像702。在所示的实施例中,已经改变图像502的配色方案(例如,已经翻转图像中的黑色和白色),并且已经执行其它图像质量调整来获得处理后的图像702。

[0050] 在一些实施例中,图像处理算法可以将初始图像转换成黑白图像,并且进一步循环通过多个图像处理参数以生成处理后的图像,诸如本文所述的滤波器、掩模和变换、或任何合适的技术。例如,对于一个或多个处理后的图像,可以使用对象识别算法来尝试检测如用户所提供的(例如,由来自用户的输入定义的)预期的架构。如果/当对象识别算法检测到预期的架构时,可以经由当前的一组图像处理参数将相关的处理后的图像视为被正确处理。如果对象识别算法未检测到预期的架构,那么可以迭代地修改处理参数、可以使用新配置重新处理图像,并且可以在新处理的图像上运行对象识别算法。在一些实施例中,这种迭代技术可以重复直到对象识别算法检测到预期的架构为止。

[0051] 在一些实施例中,对一个或多个处理后的图像执行对象识别。例如,计算机视觉工具102可以对处理后的图像702执行对象识别算法或函数以检测图像内的对象。一些实施例实现一个或多个开源工具(例如,OpenCV)以执行对象识别。例如,可以使用形状对象识别算法或函数(例如,OpenCV的findContours函数)或任何其它合适的识别协议。

[0052] 图8图示了根据示例实施例的在计算架构的图像中识别出的对象。例如,图8包括三角形802、矩形804、正方形806和808、正方形810、五边形812、线814和816以及圆形818。在一些实施例中,可以实现形状对象识别算法或函数,该形状对象识别算法或函数包括用于检测对象的边缘的边缘检测算法或函数,然后是用于识别检测到的对象的对象识别算法或函数。例如,图8中所示的检测到的对象是基于图3的示图302内的视觉元素的,因为处理后的图像基于该示图的捕获图像。在所示的实施例中,示图302中的视觉元素包括形状,因此可以实现容易识别和区分形状的对象识别算法。例如,可以基于示图302中的形状和图8中所示的对象具有的边的数量区分它们(例如,零代表圆形、一代表线、三代表三角形,四代表正方形和矩形,并且五代表五边形)。另外,可以通过比较检测到的对象的边的长度来区分正方形和矩形。

[0053] 在一些实施例中,对象识别算法可以通过比较对象的相对位置来检测对象是否被包含在另一个对象内。例如,可以确定正方形806在矩形804内,以及正方形810在正方形806

内。可以对正方形808和五边形812做出类似的确定。因此，当识别/检测对象时，对象识别算法可以使用对象的边的数量以及它是否被包含在一个或多个其它对象中。

[0054] 在一些实施例中，可以基于对象的位置标识符来确定相对对象位置。例如，图像可以被认为是二维网格，其中第一位置标识符值可以定义网格的第一维度（例如，水平）中的位置，并且第二位置标识符可以定义网格的第二维度（例如，垂直）中的位置。因此，可以通过坐标对（例如， $(x, y)$ ）来定义图像内的位置。可以使用多种不同的约定来定义给定对象的位置，诸如对象的中心点（例如，一个点定义）、对象的顶点（例如，多个点定义）或其它合适的定义。对象识别算法的实施例还可以例如使用这些约定之一来检测所检测到的对象的位置。在一些实施例中，可以容易地确定对象内的面积，诸如当使用顶点来定义对象的位置时（或者当使用指示面积的某个其它多点定义时）。给定多个检测到的对象及其位置，对象识别算法可以确定一个对象是重叠还是包含另一个对象。

[0055] 在一些实施例中，对象识别算法或函数可以使用对象定义来检测/识别对象。例如，可以在定义中使用边的数量，诸如三角形对象被定义为负载均衡器对象，以及圆形对象被定义为数据库对象。在一些实施例中，对象定义还可以包括对象是否在另一个对象内。例如，可以定义不被包含在另一个对象中的矩形对象是网络。另外，可以定义在矩形对象（或被识别为网络的对象）内的正方形对象是子网。可以定义正方形对象（或被识别为子网的对象）内的正方形对象或五边形对象是虚拟机实例。在另一个示例中，可以定义在两个对象（例如，网络和子网）内的正方形对象或五边形对象是虚拟机实例。这些仅是示例对象定义，并且可以实现其它合适的定义。

[0056] 在一些实施例中，可以使用类似的算法来检测所检测到的对象的布置。例如，可以将单边对象（例如，线）定义为对象之间的连接。在一些实施例中，一旦识别出对象并且每个对象的位置已知，就可以分析位置以确定对象的相对位置，并确定任何对象是否被包含在另一个对象内（或多个对象内）。

[0057] 在一些实施例中，可以确定指示用于识别出的对象的布置的层次结构。例如，考虑两个正方形：正方形A和正方形B，其中正方形B在正方形A的内部。对于正方形A，可以确定它没有父项，因为它是最外面的形状。在这种情况下，正方形A可以被赋予0的索引，因为它是最外面的形状，并且其父项的索引可以是-1。然后，可以确定正方形B确实有父项（例如，正方形A-在索引0处），并且它的父项没有父项（例如，父项的父项的索引为-1）。这可以用来确定正方形B是正方形A的子项，并且仅深一个级别被嵌套。因此，可以生成指示对象及其（一个或多个）父项的嵌套级别的层次结构。

[0058] 在一些实施例中，从用户接收的定义参考图6讨论的网络架构元素的输入可以用于配置对象识别算法。例如，可以向系统通知它应该检测计算实例，并且系统可以被进一步配置为知道这样的计算实例将位于子网中，并且该子网将位于网络中。因此，可以知道，预期识别出被嵌套在级别二的对象（在作为网络的对象和作为子网的对象内）。在一些实施例中，对象的检测到的布置可以基于对象的检测到的层次结构。

[0059] 实施例还包括对连接对象（诸如，线）的检测/识别。在一些实施例中，可以预期线具有两个端点（例如，不与其它线相交）。线可以用于连接各种其它对象，诸如表示负载均衡器连接的线。在一些实施例中，线可以基于到线的端点的接近度而被链接到网络元素（例如，可以连接负载均衡器和要从负载均衡器接收流量的虚拟机/子网）。

[0060] 与对象的层次结构的识别类似,从用户接收的定义网络架构元素的输入可以用于配置对线的识别。例如,如果输入定义应该有两个负载均衡器连接(例如,负载均衡器将流量定向到两个不同的VM),那么对象检测/识别可以被配置为查找包含两条线的处理后的图像,该两条线具有端点:1)靠近(例如,非常接近)负载均衡器对象(例如,外层三角形),以及2)靠近(例如,非常接近)两个不同的计算实例对象(例如,两个不同的嵌套正方形)。与其它对象检测/识别功能(诸如确定层次结构)相似,对象的相对位置可以用于检测哪些对象接近线的端点。在一些实施例中,来自开放源工具(例如,OpenCV)的库可以用于执行这些函数中的一个或多个函数。例如,OpenCV中的findContours()可以用于基于检测到的形状返回层次结构矩阵。

[0061] 在一些实施例中,基于这些对象定义(例如,定义为边的数量,并且可选地,该对象是否被包含在另一个对象中),图8中识别出的对象可以各自与网络元素相关联。例如,三角形802可以与负载均衡器相关联,矩形804可以与网络相关联,并且圆形818可以与数据库相关联。另外,正方形806和808可以与子网(例如,位于与网络相关联的矩形804内)相关联。正方形810和五边形812可以与虚拟机实例(例如,位于与子网相关联的正方形806和808以及与网络相关联的矩形804内)相关联。另外,可以将线814和816识别为由三角形802表示的负载均衡器与由正方形806和808表示的子网之间的连接。

[0062] 在一些实施例中,基于与图像一起接收的输入参数来执行对象识别。例如,可以对照参考图6讨论的输入参数来验证对象识别。在一些实施例中,对照在输入中定义的每个网络元素类型的数量来验证与每个网络元素类型相关联的识别出的对象的数量。例如,图6的输入参数602包括在图5的图像502中表示的网络元素的数量。在所示的实施例中,输入参数602是一个网络、两个子网、两个虚拟机实例、一个负载均衡器、两个连接和一个数据库。可以对照这些数量来验证图8中识别出的对象及其相关联的网络元素类型,以验证图1的计算机视觉工具102已确定准确的网络架构。

[0063] 在一些实施例中,可以类似地实现其它对象定义、识别出的对象和布置。因此,基于图像处理、对象定义和对象识别功能,可以由图1的计算机视觉工具102检测在图3的示图302和图5的图像502中表示的网络元素和网络架构。

[0064] 在一些实施例中,识别出的对象和识别出的布置可以用于生成用于供给实际的网络元素的一个或多个配置文件。例如,可以为与每个识别出的对象相关联的网络元素生成一个或多个数据文件。图9A-9B图示了根据示例实施例的用于供给计算架构的配置文件。在一些实施例中,数据902和904可以是数据文件(例如,JavaScript对象表示(“JSON”)文件)的一部分,该数据文件保有关于要处理的网络元素的配置信息及其之间的关系。

[0065] 例如,数据文件902包括标记为adb0的数据库条目“adbs”和标记为lb0的负载均衡器条目“lbs”,其示出了到instance0/subnet0和instance1/subnet1的连接。数据文件902还示出了虚拟网络条目“vcns”,其由数据文件904进一步详细说明。例如,数据文件904中定义的虚拟网络具有两个子网,分别标记为subnet1和subnet0,每个子网具有虚拟机实例,分别标记为instance0和instance1。数据文件904还将实例0和实例1的形状分别列为矩形和五边形。数据文件904显示虚拟网络被标记为vcn0。

[0066] 实施例将该配置数据文件提供给供给工具,该供给工具供给文件中定义的网络架构。例如,HashiCorp的Terraform可以用于在云基础设施(例如,Oracle®云基础设施)中

供给计算资源。有时被称为“基础设施即代码”的Terraform能够读取格式正确的文件(例如,根据JSON或Hashicorp专有语言、Hashicorp配置语言(HCL)格式化的)并基于文件内的定义供给基础设施。例如,可以生成执行计划,该执行计划描述将执行哪些函数来供给配置文件中定义的计算资源,然后可以执行该计划以构建所描述的基础设施。在一些实施例中,可以构建计算资源的图,并且可以并行地执行非依赖性资源的供给。实施例还可以包括Terraform供给者,其可以运行一组脚本和/或命令以供给基础设施。因此,配置数据文件可以用于启动一组处理、脚本和/或命令(例如,使用Terraform)以供给配置文件中定义的计算资源。可以类似地实现任何其它合适的供给产品、服务或功能。

[0067] 图10-16图示了根据示例实施例的与供给计算架构相关的用户界面。例如,图10描绘了用户界面1002,其列出了虚拟机实例1004和1006作为供给。图11进一步描绘了关于虚拟机实例1006的信息,诸如形状1204、虚拟云网络1206和专用IP地址1208。

[0068] 图12进一步描绘了关于虚拟机实例1004的信息,诸如形状1204、虚拟云网络1206和专用IP地址1208。图12还描绘了用于虚拟机实例1004的公共IP地址1210和子网1212,并且用户界面1102可以包括关于虚拟机实例1006的类似信息。

[0069] 图13的用户界面1302描绘了供给的和可用的虚拟网络1304。图14的用户界面1402描绘了虚拟网络1304内供给的子网和可用的子网,诸如子网1404和1406。另外,图14描绘了子网的无类域间路由块,CIDR 1408和1410。某些默认设置也可以用于供给计算资源。例如,默认设置可以用于为子网供给路由表1412和1416以及安全列表1414和1418。在一些实施例中,还可以默认创建互联网网关,该互联网网关可以向被供给的计算资源提供到互联网的连接性。例如,可以将路由表(例如,路由表1412和1416)自动配置为通过该互联网网关路由流量(例如,传出流量)。

[0070] 图15描绘了用户界面1502,该用户界面列出了供给的和活动的负载均衡器1504以及用于负载均衡器的公共IP地址1506。图16描绘了用户界面1602,该用户界面列出了供给数据库1604。在一些实施例中,网络元素可以花费不同的持续时间进行供给,并且一旦它们被供给,网络架构就可以完全工作并且可用。

[0071] 图17图示了根据示例实施例的用于使用计算机视觉来供给计算资源的流程图。在一些实施例中,图17的功能由存储在存储器或其它计算机可读或有形介质中的软件实现,并由处理器执行。在其它实施例中,每个功能可以由硬件执行(例如,通过使用专用集成电路(“ASIC”)、可编程门阵列(“PGA”)、现场可编程门阵列(“FPGA”)等等),或由硬件和软件的任意组合执行。在实施例中,图17的功能可以由图2的系统200的一个或多个元素执行。

[0072] 在1702处,可以接收包括多个视觉元素的图像。例如,可以接收包括网络架构的示图的图像。该示图可以包括表示网络元素的视觉元素和表示网络架构的视觉元素的布置。

[0073] 在1704处,可以接收与图像相关联的包括网络元素类型的网络元素的数量的输入。例如,可以接收与示图中表示的视觉元素/网络元素对应的输入参数。在一些实施例中,网络元素类型可以包括数据库、负载均衡器、网络、子网和/或虚拟机实例中的一种或多种。输入参数可以包括示图中存在的每种网络元素类型的数量(例如,一个负载均衡器、两个数据库、两个虚拟机等)。

[0074] 在1706,可以使用一个或多个变换来处理图像。例如,可以对图像执行一种或多种图像处理技术以实现一种或多种变换。在一些实施例中,图像处理技术可以调整图像的像

素值。图像处理技术可以包括掩模、滤波器、卷积、互相关以及任何其它合适的图像处理技术。

[0075] 在1708处,基于该处理,可以将图像内的视觉元素识别为表示网络元素的对象,其中识别出的对象表示网络元素类型的网络元素。例如,可以基于多个对象定义来识别对象,并且每个对象定义可以与网络元素类型中的至少一种相关联。

[0076] 在一些实施例中,对象定义至少包括检测到的边的数量。对象定义还可以包括形状,诸如三角形、矩形、正方形、圆形或五边形中的一个或多个。

[0077] 在一些实施例中,基于输入执行对象识别。例如,可以对照输入来验证对象识别,使得对照输入中定义的每种网络元素类型的数量来验证与每种网络元素类型相关联的识别出的对象的数量。

[0078] 在一些实施例中,表示网络架构的识别出的对象的布置被识别。例如,图像的第一部分可以包括包含有至少两个第二视觉元素的至少一个第一视觉元素,并且基于图像的第一部分的用于被供给的网络元素的网络架构可以包括包含有两个子网的网络。在这个示例中,与第一视觉元素对应的第一识别出的对象可以与网络相关联,与第二视觉元素对应的第二识别出的对象可以与子网相关联,并且第一对象和第二对象之间识别出的布置可以与包括子网的网络相关联。

[0079] 在另一个示例中,图像的第二部分可以包括至少第三视觉元素,该至少第三视觉元素可视地连接到第二视觉元素,并且基于图像的第二部分的用于被供给的网络元素的网络架构可以包括均衡两个子网之间的负载的负载均衡器。在这个示例中,与第三视觉元素对应第三识别出的对象可以与负载均衡器相关联,并且第三对象和第二对象之间识别出的布置可以与负载均衡器和两个子网之间的连接相关联。

[0080] 在1710处,可以基于识别出的对象来生成多个网络元素配置文件,其中该配置文件可以用于供给与识别出的对象相关联的网络元素。例如,配置文件可以是根据预定协议生成的数据文件。在一些实施例中,供给工具可以被配置为接收数据文件并实现网络元素的供给。

[0081] 在一些实施例中,基于识别出的对象的识别出的布置来定义一个或多个网络元素配置文件。例如,可以通过识别出的对象的布置来表示与识别出的对象对应的网络元素之间的关系,并且可以在生成的数据文件中定义这些关系。

[0082] 在1712处,可以通过供给包括一种或多种网络元素类型的网络元素来使得计算资源与识别出的对象对应地被供给。例如,被供给的网络元素可以对应于识别出的对象,使得被供给的每种网络元素类型的数量对应于与每种网络元素类型相关联的识别出的对象的数量。

[0083] 在一些实施例中,供给包括基于识别出的布置为被供给的网络元素生成网络架构。例如,识别出的对象的识别出的布置可以表示网络内的子网、网络或子网内的虚拟机、负载均衡器和网络之间的连接、子网、虚拟机或其它负载均衡器等。供给可以包括根据网络元素之间的相互关系来供给网络元素(例如,如由识别出的布置所表示的)。在一些实施例中,供给计算资源包括根据云基础设施内的网络架构来供给网络元素。

[0084] 贯穿本说明书描述的本公开的特征、结构或特性可以在一个或多个实施例中以任何合适的方式组合。例如,在整个说明书中,“一个实施例”、“一些实施例”、“某个实施例”、

“某些实施例”或其它类似语言的使用是指结合实施例描述的特定特征、结构或特性可以被包括在本公开的至少一个实施例中的事实。因此，贯穿本说明书出现的短语“一个实施例”、“一些实施例”、“某个实施例”、“某些实施例”或其它类似语言不一定都指代相同的一组实施例，并且所描述的特征、结构或特性可以在一个或多个实施例中以任何合适的方式被组合。

[0085] 本领域普通技术人员将容易理解，如上所讨论的实施例可以以不同顺序的步骤和/或在配置中用与所公开的元件不同的元件来实践。因此，虽然本公开考虑了概述的实施例，但是对于本领域技术人员显而易见的是，某些修改、变化和替代构造将是显而易见的，同时保持在本公开的精神和范围内。因此，为了确定本公开的范围和界限，应该参考所附权利要求。

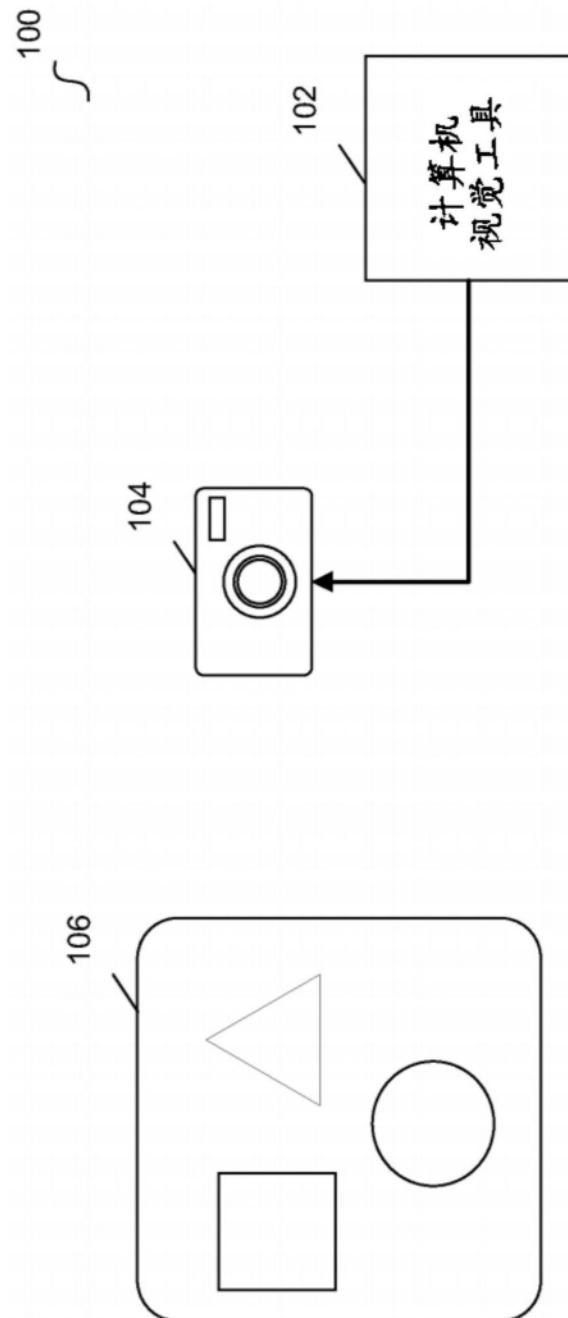


图1

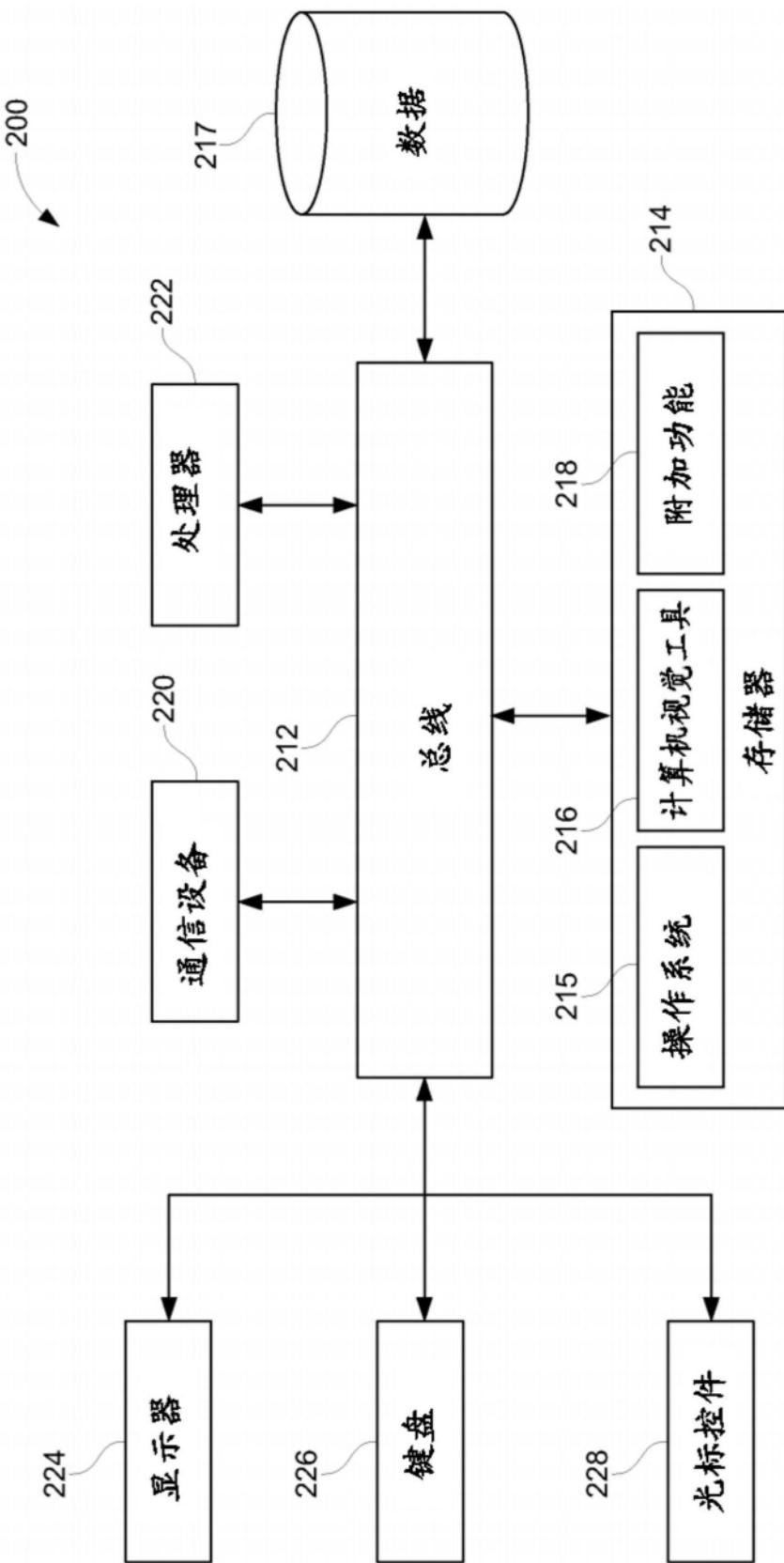


图2

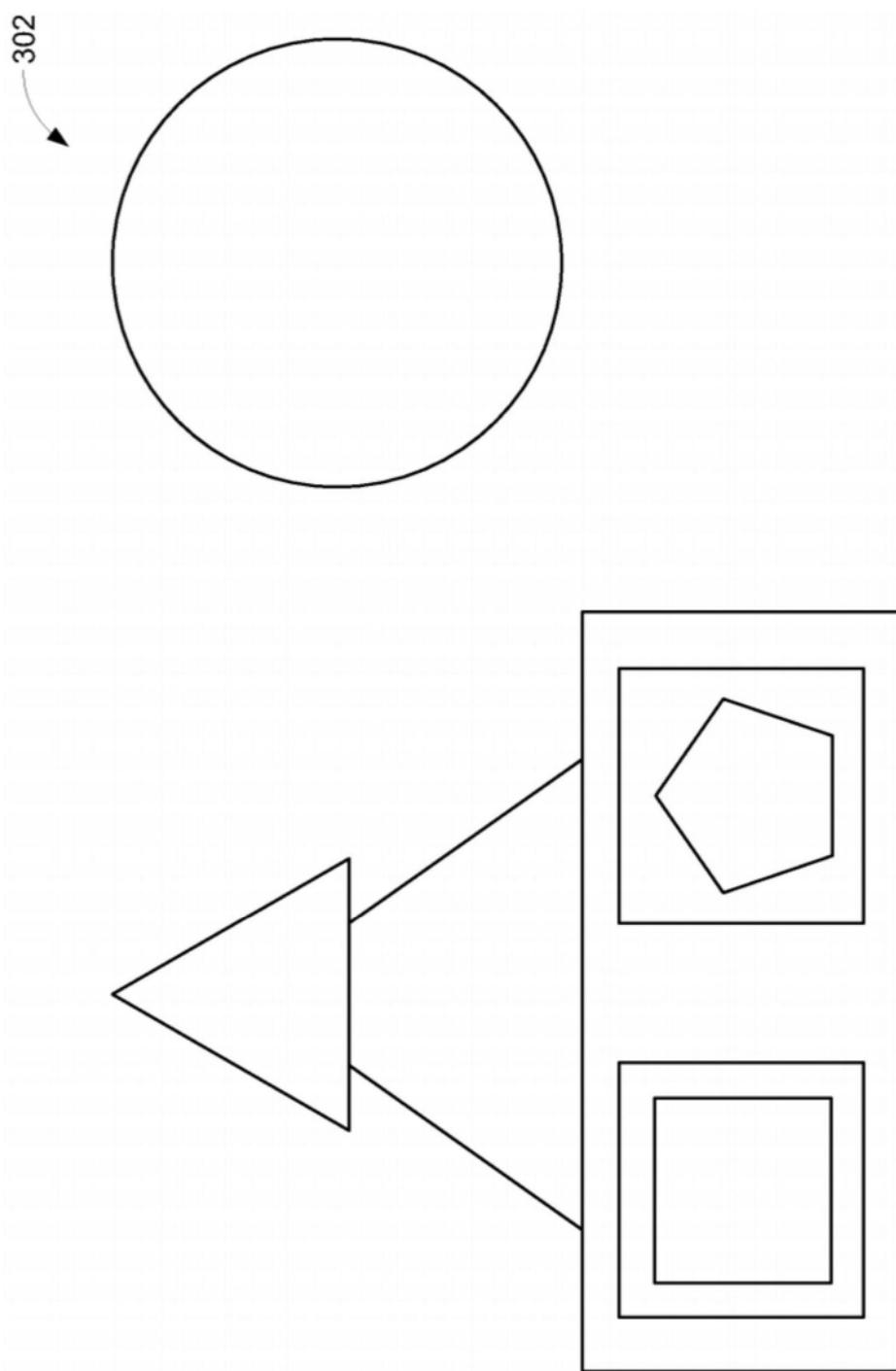


图3

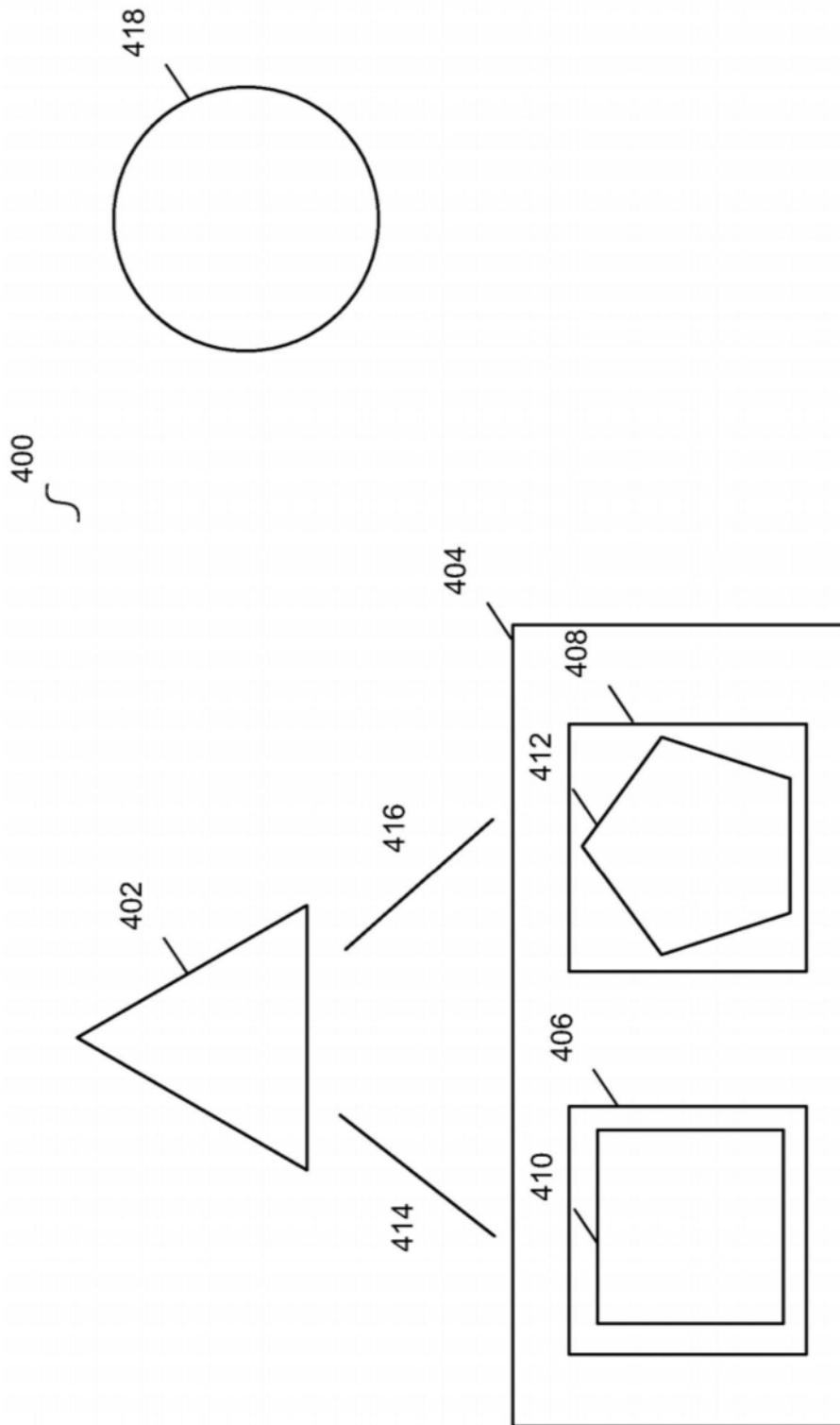


图4

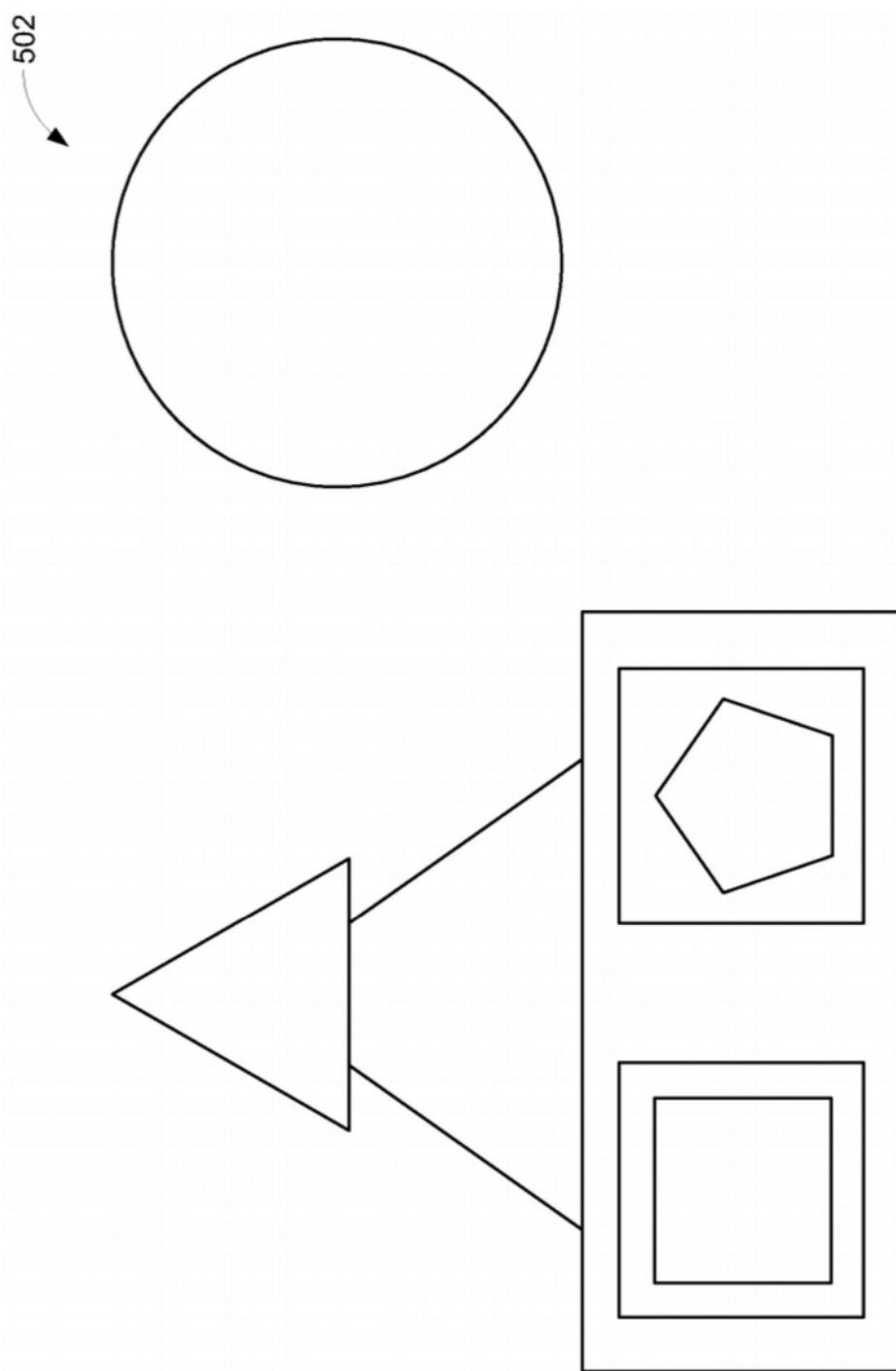


图5

602

```
~/D/c/autoprovision >>> python3 main.py --image ./images/demo.jpg
您想要创建多少个网络？ 1 个
您想要创建多少个子实例？ 2 个
您想要创建多少个均衡器？ 1 个
您想要创建多少个负载均衡器连接（即，线）？ 1 个 []
您想要创建多少个主数据仓库？ 1 个 []
```

图6

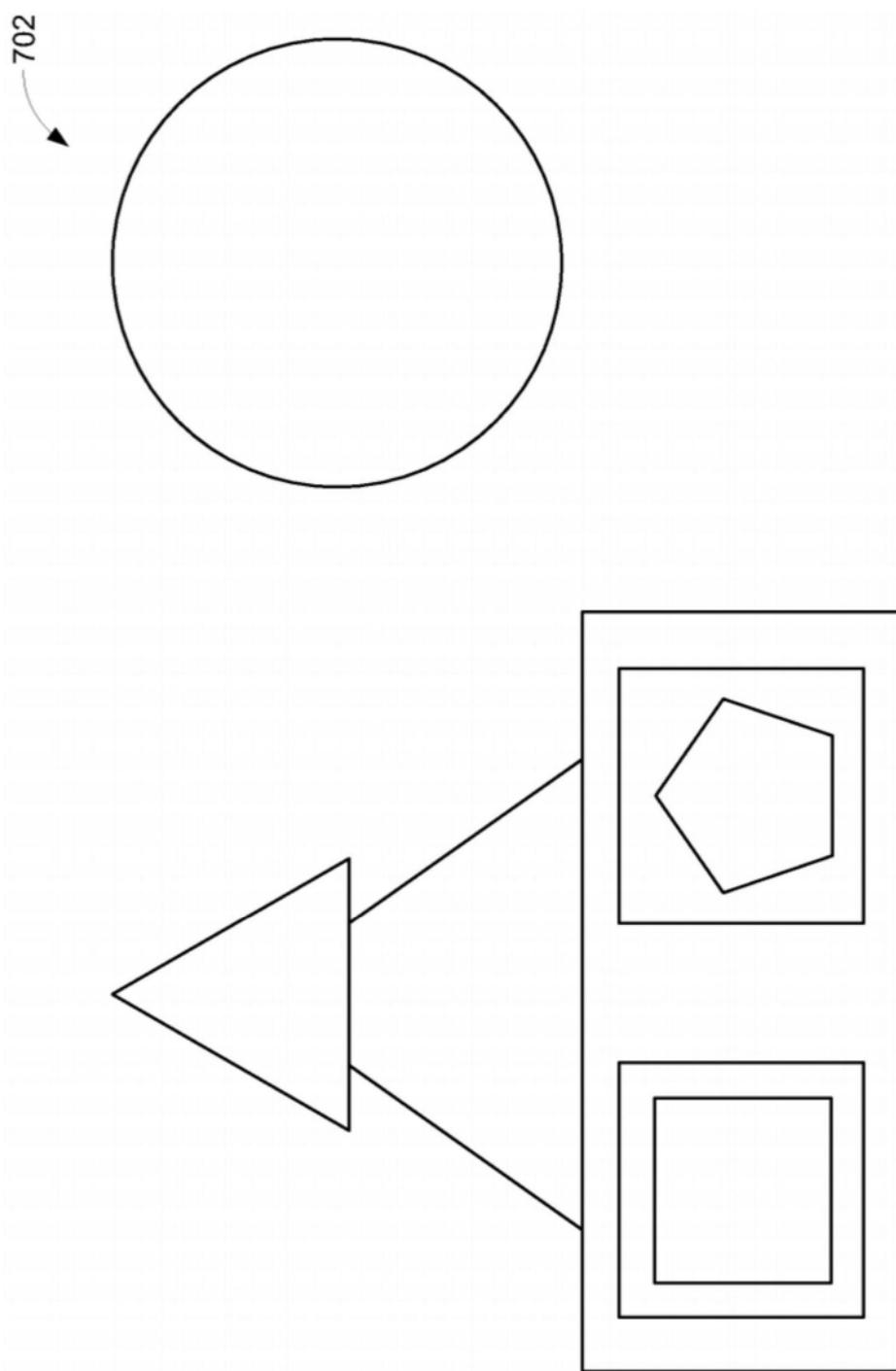


图7

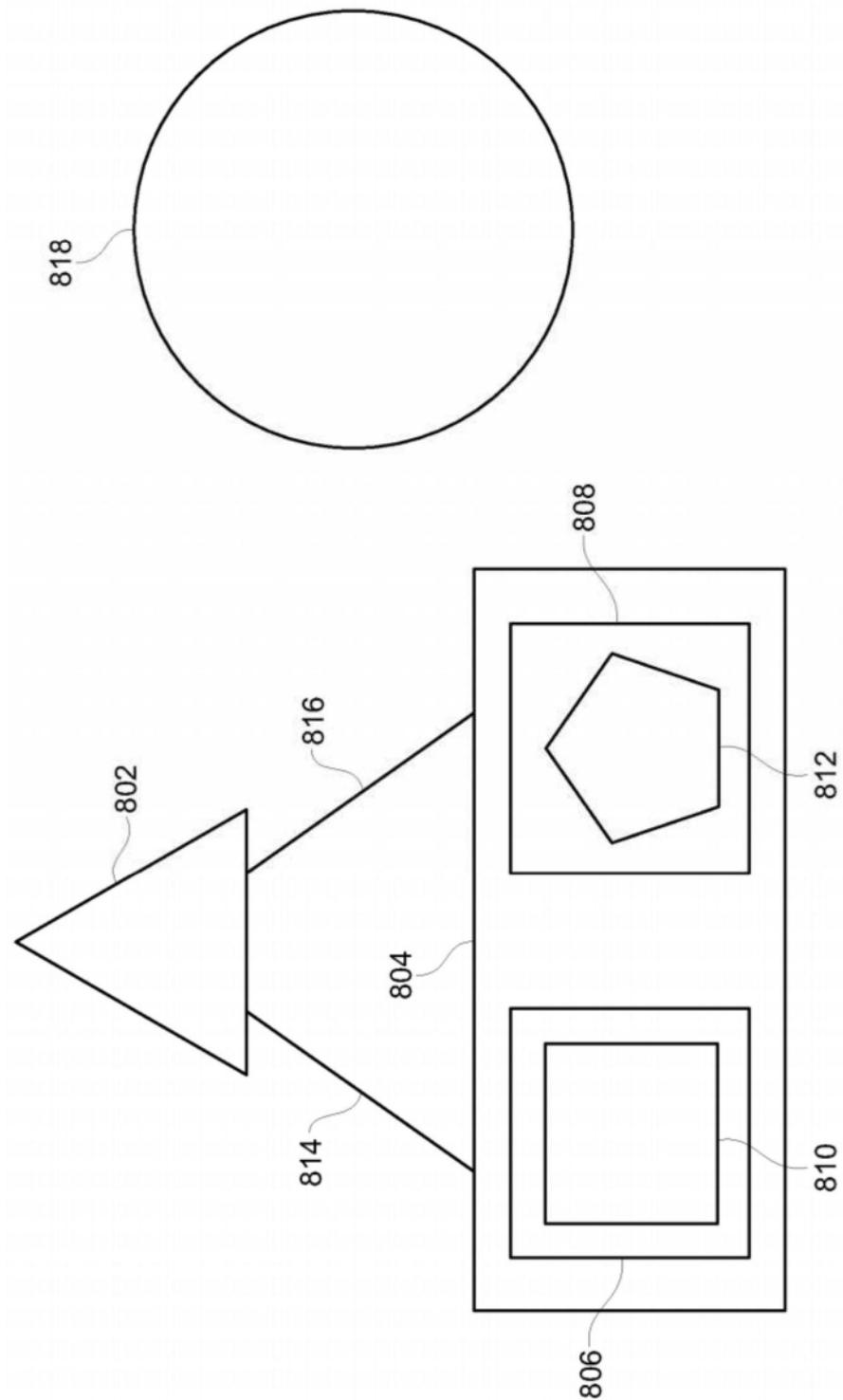


图8

902

```
Success! Counts: [1, 2, 2, 1, 2, 1]
{'adbs': [[{'countourIndex': 12, 'dbname': 'adb0', 'display_name': 'adb0'}],
'connections': [[8, 14], [4, 14]],
'lbs': [[{'conns': [[4, 'instance0', 'subnet0'], [8, 'instance1', 'subnet1']],
'contourIndex': 14,
'lname': 'lbo '}},
{'vcns': [{'cidr': '10.0.0.0/16'}]}]
```

图9A

904

```

'contourIndex': 0,
'subnets': [{availability_domain': '${lookup(data.oci_identity_availability_domains.ads.availability_domains[var.availability_
domain ,
' + 0], "name")}',
'cidr': '10.0.0.0/24',
'contourIndex': 2,
'instances': [{availability_domain':
'$lookup(data.oci_identity_availability_domains.ads.availability_domains[var.availability_domain
' + 0], "name")},
'contourIndex': 4,
'inst_name': 'instance0',
'shape': 'rectangle'],
'sn_name': 'subnet0'},
{availability_domain': '${lookup(data.oci_identity_availability_domains.ads.availability_domains[var.availability_
domain ,
' + 1], "name")}',
'cidr': '10.0.1.0/24',
'contourIndex': 6,
'instances': [{availability_domain: '${lookup(data.oci_identity_availability_domains.ads.availability_domains
[var.availability_domain
' + 1], "name")},
'contourIndex': 8,
'inst_name': 'instance1',
'shape': 'pentagon'],
'sn_name': 'subnet1'},
'vecn_name': 'vcn0'}]}

```

图9B

排序方式:	创建日期(描述) ^
<input type="checkbox"/> 实例 1   OCID: ...q5atba 显示 复制 供给 ...	1004 形状: VM.Standard2.8 区域: iad 可用性域: PHSI:US-ASHBURN-AD-2 故障域: FAULT-DOMAIN-3
<input type="checkbox"/> 实例 0   OCID: ...oijx7q 显示 复制 供给 ...	1006 形状: VM.Standard2.4 区域: iad 可用性域: PHSI:US-ASHBURN-AD-1 故障域: FAULT-DOMAIN-1

图10

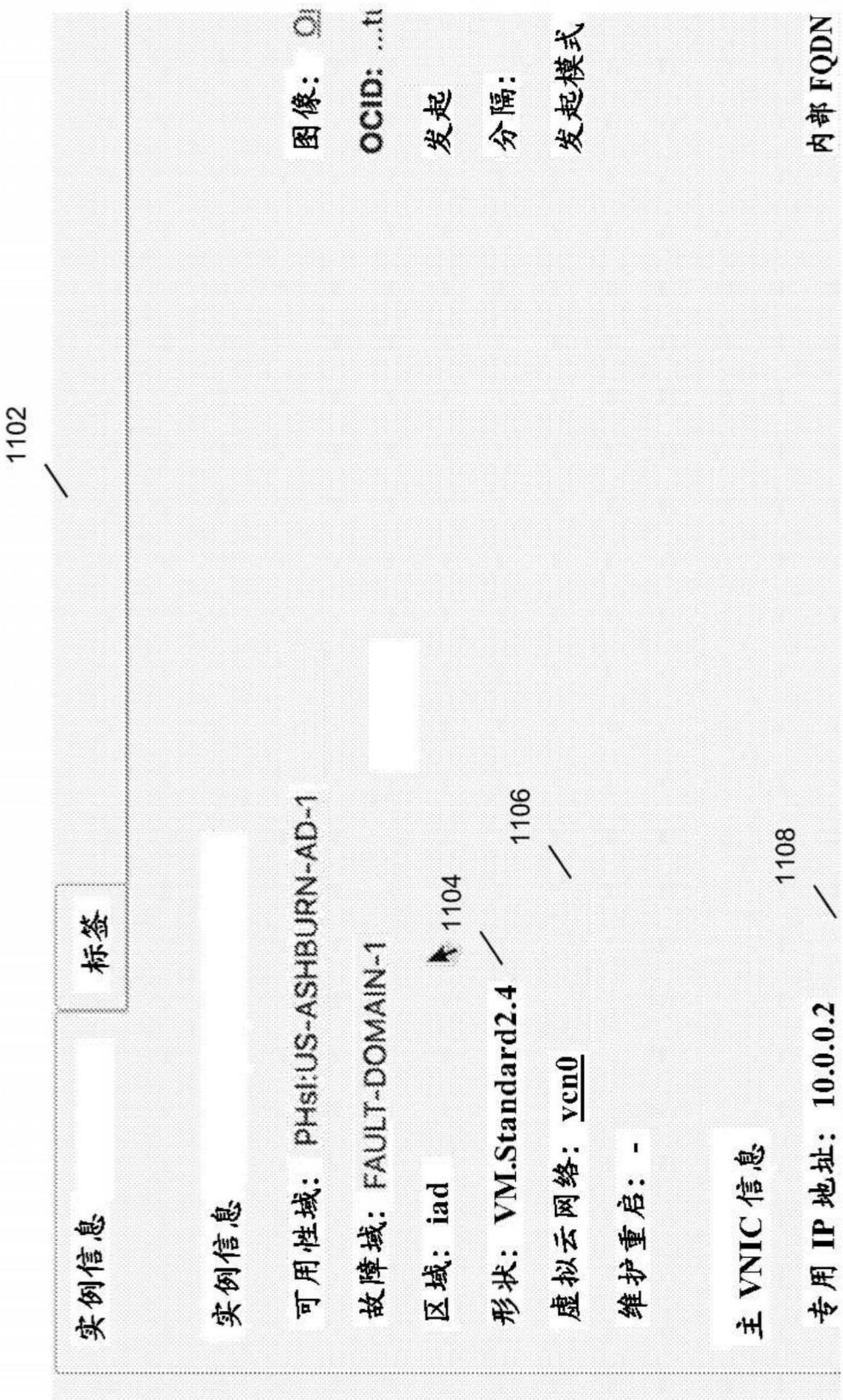


图11

1202

**实例信息****标签****实例信息**

可用性域: PHSI-US-ASHBURN-AD-2

故障域: FAULT-DOMAIN-3

区域: iad 1204

形状: VM.Standard2.8 1206

虚拟云网络: vcn0

维护重启: -

**主 VNIC 信息**

专用 IP 地址: 10.0.0.2 1208

公共 IP 地址: 132.145.167.34 1210

子网: subnet1

1212

图像: Oracle-Linux-7.6-2019.01.17-0

OCID: ...nw2jq 显示 复制

发起: 2019 年 3 月 20 日, 星期三, 21:41:05

分隔: natsammonhubtraining

发起模式: 本机

内部 FQDN: 实例 1... 显示 复制

子网: subnet1

1212

图12

1302

cv_component 组件中的虚拟云网络			
创建的虚拟云网络			
排序方式: 创建日期 (描述) ^ v			
 VCN	<u>vcn0</u>	1304 CIDR 块:	10.0.0.0/16
	OCID: ...54snqq	默认路由表: vcn0 的默认路由表	DNS 域名: Vcn0...显示 复制
可用	显示 复制		
 VCN	代理 - 原型	CIDR 块:	10.0.0.0/16
	OCID: ...som4fa	默认路由表: 代理 原型的默认路由表	DNS 域名: Pp...显示 复制
可用	显示 复制		

图13

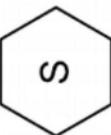
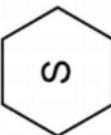
cv_compartment 分隔中的子网	
创建的子网	
排序方式:	创建日期 ( 描述 )
 可用	<p>subnet1 CIDR 块: 1408  <b>OCID:</b> 10.0.0.0/24  <u>bo4ila</u> <u>显示</u>  <u>复制</u></p> <p>虚拟路由器  <b>MAC 地址:</b> 00:00:17:21:BA:EB</p>
	<p>子网类型: 可用的特定于域  <b>可用性域:</b> PhsI:US-ASHBURN-AD-2  <b>DNS 域名:</b> Subnet1...<u>显示</u> <u>复制</u></p> <p>子网接入:            公共子网</p>
 可用	<p>subnet0 CIDR 块: 1410  <b>OCID:</b> 10.0.0.0/24  <u>ldwimq</u> <u>显示</u>  <u>复制</u></p> <p>虚拟路由器  <b>MAC 地址:</b> 00:00:17:21:BA:EB</p>
	<p>子网类型: 可用的特定于域  <b>可用性域:</b> PhsI:US-ASHBURN-AD-1  <b>DNS 域名:</b> Subnet0...<u>显示</u> <u>复制</u></p> <p>子网接入:            公共子网</p>

图14

1502	
创建负载均衡器	
排序方式: 创建日期 ( 描述 )	^
lb0	1504
OCID: ...irqha	<u>显示</u> <u>复制</u>
活动	IP 地址: 129.213.143.41 ( 公共 ) 1506 创建: 2019 年 3 月 20 日, 星期三, 20:02:26 GMT

图15

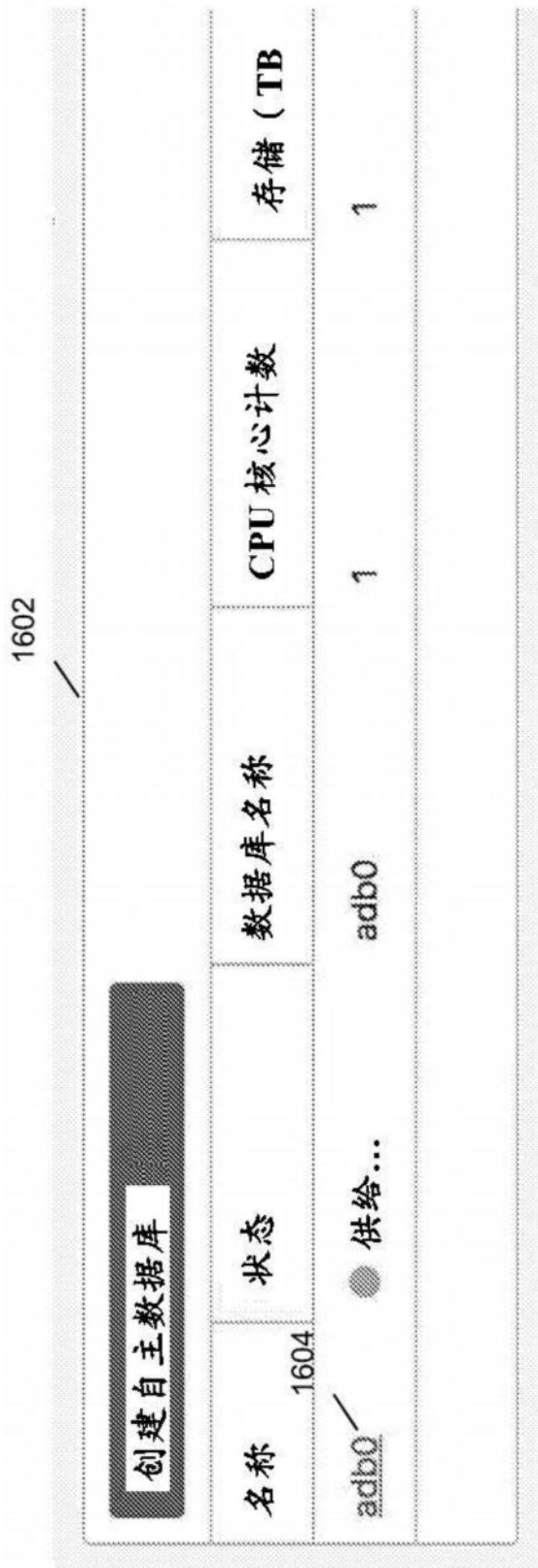


图16

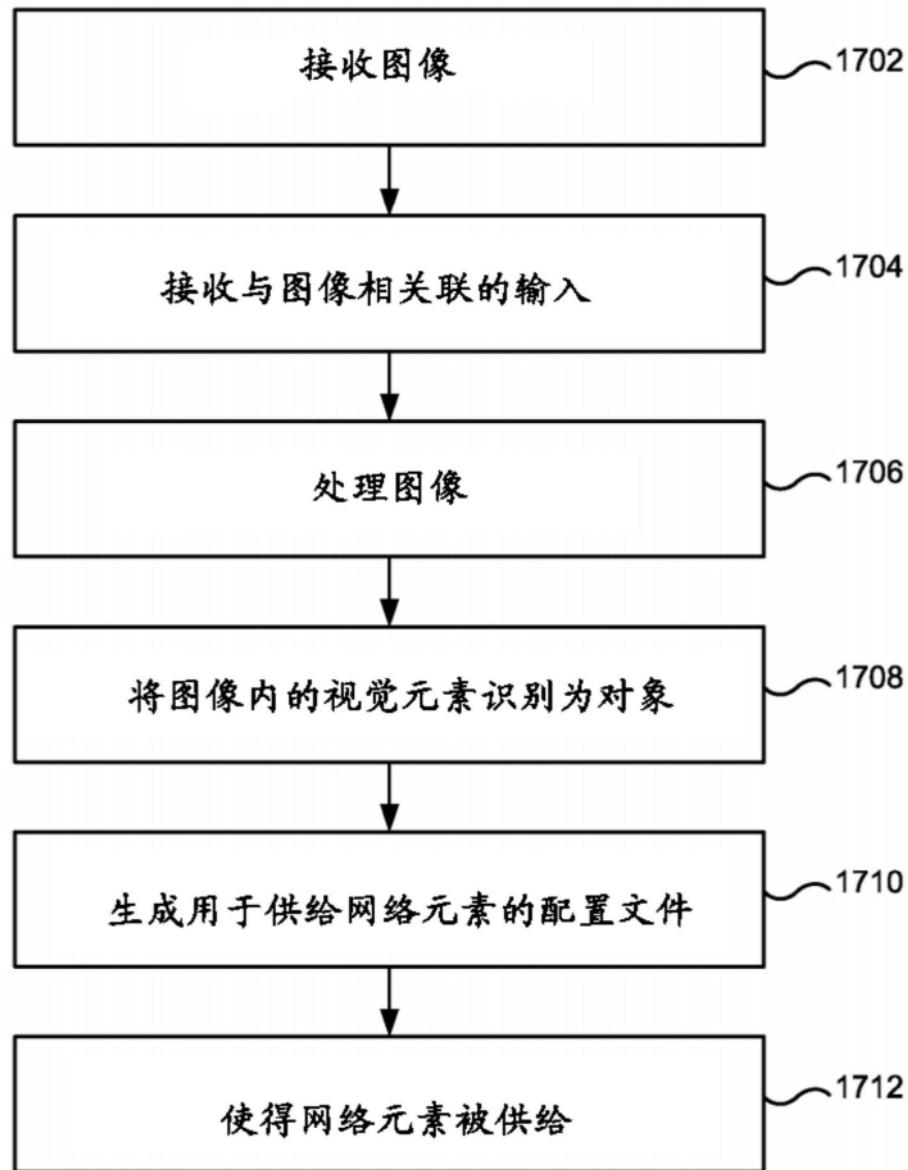


图17