



(12) 发明专利申请

(10) 申请公布号 CN 115759195 A

(43) 申请公布日 2023. 03. 07

(21) 申请号 202211062950.8

(22) 申请日 2022.09.01

(30) 优先权数据

17/464036 2021.09.01 US

(71) 申请人 辛纳普蒂克斯公司

地址 美国加利福尼亚州

(72) 发明人 O·奥雷菲

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

专利代理师 董婕 陈岚

(51) Int.Cl.

G06N 3/0464 (2023.01)

G06N 3/08 (2006.01)

H04N 5/21 (2023.01)

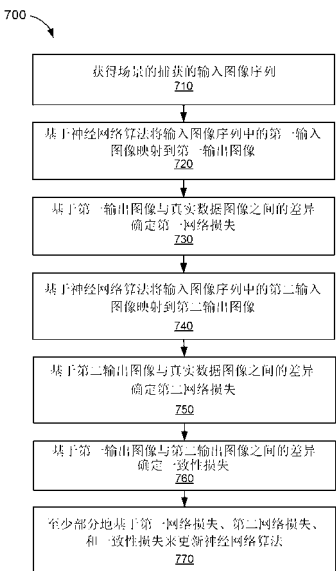
权利要求书2页 说明书14页 附图8页

(54) 发明名称

基于一致性损失的神经网络训练

(57) 摘要

一种用于对图像序列进行去噪同时保持在序列中连续显示的图像之间的一致外观的系统和方法。机器学习系统基于神经网络算法将图像序列中的第一输入图像映射到第一输出图像,并且基于第一输出图像与真实数据图像之间的差异来确定第一网络损失。该系统还基于神经网络算法将图像序列中的第二输入图像映射到第二输出图像,并且基于第二输出图像与真实数据图像之间的差异来确定第二网络损失。系统基于第一输出图像与第二输出图像之间的差异来确定一致性损失,并且基于第一网络损失、第二网络损失和一致性损失来更新神经网络算法。



1. 一种训练神经网络的方法,包括:
获得场景的捕获的输入图像序列;
基于神经网络算法将所述输入图像序列中的第一输入图像映射到第一输出图像;
基于所述第一输出图像与真实数据图像之间的差异确定第一网络损失;
基于所述神经网络算法将所述输入图像序列中的第二输入图像映射到第二输出图像;
基于所述第二输出图像与所述真实数据图像之间的差异确定第二网络损失;
基于所述第一输出图像与所述第二输出图像之间的差异确定一致性损失;以及
至少部分地基于所述第一网络损失、所述第二网络损失、和所述一致性损失来更新所述神经网络算法。

2. 根据权利要求1所述的方法,其中所述第一输出图像和所述第二输出图像中的每一个包括多个像素值,并且其中所述一致性损失是基于与所述第一输出图像的所述多个像素值和所述第二输出图像的所述多个像素值相关联的平均像素值来确定的。

3. 根据权利要求1所述的方法,其中所述第一输出图像和所述第二输出图像中的每一个包括多个像素值,并且其中确定所述一致性损失包括生成包括所述第一输出图像的所述多个像素值和所述第二输出图像的所述多个像素值的矩阵。

4. 根据权利要求3所述的方法,其中所述一致性损失是基于所述矩阵的秩来确定的。

5. 根据权利要求1所述的方法,其中所述真实数据图像具有比所述第一输入图像或所述第二输入图像中的任何高的信噪比SNR。

6. 根据权利要求1所述的方法,其中所述神经网络算法包括递归神经网络算法。

7. 根据权利要求1所述的方法,还包括:

至少部分地基于所述第一网络损失、所述第二网络损失、和所述一致性损失来确定总损失。

8. 根据权利要求7所述的方法,还包括:

使总损失最小化;以及

基于最小化的总损失来确定一个或多个参数,其中所述神经网络算法是基于所述一个或多个参数而更新的。

9. 根据权利要求1所述的方法,还包括:

基于所述神经网络算法将所述输入图像序列中的第三输入图像映射到第三输出图像;
以及

基于所述第三输出图像与所述真实数据图像之间的差异确定第三网络损失,其中所述神经网络算法基于所述第三网络损失被进一步更新。

10. 根据权利要求9所述的方法,其中所述一致性损失还基于所述第三输出图像与所述第二输出图像之间的差异来确定。

11. 一种机器学习系统,包括:

处理系统;以及

存储指令的存储器,所述指令在由所述处理系统执行时使所述机器学习系统:

获得场景的捕获的输入图像序列;

基于神经网络算法将所述输入图像序列中的第一输入图像映射到第一输出图像;

基于所述第一输出图像与真实数据图像之间的差异确定第一网络损失;

基于所述神经网络算法将所述输入图像序列中的第二输入图像映射到第二输出图像；
基于所述第二输出图像与所述真实数据图像之间的差异确定第二网络损失；
基于所述第一输出图像与所述第二输出图像之间的差异确定一致性损失；以及
至少部分地基于所述第一网络损失、所述第二网络损失、和所述一致性损失更新所述神经网络算法。

12. 根据权利要求11所述的机器学习系统，其中所述第一输出图像和所述第二输出图像中的每一个包括多个像素值，并且其中所述一致性损失是基于与所述第一输出图像的所述多个像素值和所述第二输出图像的所述多个像素值相关联的平均像素值来确定的。

13. 根据权利要求11所述的机器学习系统，其中所述第一输出图像和所述第二输出图像中的每一个包括多个像素值，并且其中确定所述一致性损失包括生成包括所述第一输出图像的所述多个像素值和所述第二输出图像的所述多个像素值的矩阵。

14. 根据权利要求13所述的机器学习系统，其中所述一致性损失是基于所述矩阵的秩来确定的。

15. 根据权利要求11所述的机器学习系统，其中所述真实数据图像具有比所述第一输入图像或所述第二输入图像中的任何高的分辨率。

16. 根据权利要求11所述的机器学习系统，其中所述神经网络算法包括递归神经网络算法。

17. 一种训练神经网络的方法，包括：
获得场景的捕获的输入图像序列；
基于神经网络算法将所述输入图像序列中的第一输入图像映射到第一输出图像；
基于所述第一输出图像与第一真实数据图像之间的差异确定第一网络损失；
基于所述神经网络算法将所述输入图像序列中的第二输入图像映射到第二输出图像；
基于所述第二输出图像与第二真实数据图像之间的差异确定第二网络损失；
基于所述第一输出图像与所述第二输出图像之间的差异来确定一致性损失；以及
至少部分地基于所述第一网络损失、所述第二网络损失、和所述一致性损失来更新所述神经网络算法。

18. 根据权利要求17所述的方法，其中所述第一输出图像和所述第二输出图像中的每一个包括多个像素值，并且其中所述一致性损失是基于与所述第一输出图像的所述多个像素值和所述第二输出图像的所述多个像素值相关联的平均像素值来确定的。

19. 根据权利要求17所述的方法，其中所述第一输出图像和所述第二输出图像中的每一个包括多个像素值，并且其中确定所述一致性损失包括生成包括所述第一输出图像的所述多个像素值和所述第二输出图像的所述多个像素值的矩阵。

20. 根据权利要求19所述的方法，其中所述一致性损失是基于所述矩阵的秩来确定的。

基于一致性损失的神经网络训练

技术领域

[0001] 本实施方式总体上涉及神经网络,并且具体地涉及基于一致性损失来训练神经网络。

背景技术

[0002] 当图像捕获设备捕获场景的一系列图像(或视频)时,每个图像可能表现出噪声。当图像由机器学习模型处理并且然后以序列(诸如视频)呈现时,视频可能由于噪声的变化而看起来闪烁。也就是说,由于连续的处理的图像之间的噪声差异,场景中的对象或光可能看起来不稳定或不规则。

发明内容

[0003] 提供本发明内容是为了以简化的形式介绍下面在具体实施方式中进一步描述的概念的选择。本发明内容不旨在标识所要求保护的主题的关键特征或必要特征,也不旨在限制所要求保护的主题的范围。

[0004] 本公开的主题的一个创新方面可以在训练神经网络的方法中实现。所述方法包括以下步骤:获得场景的捕获的输入图像序列;基于神经网络算法将所述输入图像序列中的第一输入图像映射到第一输出图像;基于所述第一输出图像与真实数据图像之间的差异确定第一网络损失;基于所述神经网络算法将所述输入图像序列中的第二输入图像映射到第二输出图像;基于所述第二输出图像与所述真实数据图像之间的差异确定第二网络损失;基于第一输出图像与第二输出图像之间的差异确定一致性损失;以及至少部分地基于第一网络损失、第二网络损失和一致性损失来更新神经网络算法。

[0005] 本公开的主题的另一创新方面可以在包括处理系统和存储器的机器学习系统中实现。所述存储器存储指令,所述指令在由所述处理系统执行时,使所述机器学习系统:获得场景的捕获的输入图像序列;基于神经网络算法将所述输入图像序列中的第一输入图像映射到第一输出图像;基于所述第一输出图像与真实数据图像之间的差异来确定第一网络损失;基于所述神经网络算法将所述输入图像序列中的第二输入图像映射到第二输出图像;基于所述第二输出图像与所述真实数据图像之间的差异来确定第二网络损失;基于第一输出图像与第二输出图像之间的差异确定一致性损失;以及至少部分地基于所述第一网络损失、所述第二网络损失和所述一致性损失来更新所述神经网络算法。

[0006] 本公开的主题的另一创新方面可以在训练神经网络的方法中实现。所述方法包括以下步骤:获得场景的捕获的输入图像序列;基于神经网络算法将所述输入图像序列中的第一输入图像映射到第一输出图像;基于所述第一输出图像与第一真实数据图像之间的差异确定第一网络损失;基于所述神经网络算法将所述输入图像序列中的第二输入图像映射到第二输出图像;基于所述第二输出图像与所述第二真实数据图像之间的差异确定第二网络损失;基于第一输出图像与第二输出图像之间的差异确定一致性损失;以及至少部分地基于第一网络损失、第二网络损失和一致性损失来更新神经网络算法。

附图说明

[0007] 本实施例通过示例的方式示出,并且不旨在受附图中的图的限制。

[0008] 图1示出了根据一些实施方式的可以采用神经网络模型的示例图像捕获和显示系统的框图。

[0009] 图2示出了根据一些实施方式的采用神经网络模型的图像处理器的示例操作的框图。

[0010] 图3示出了根据一些实施方式的示例机器学习系统的框图。

[0011] 图4示出了根据一些实施方式的示例损失计算器的框图。

[0012] 图5示出了描绘与金字塔(pyramid)损失相关联的示例金字塔层级的说明性图表。

[0013] 图6示出了根据一些实施方式的示例机器学习系统。

[0014] 图7示出了描绘根据一些实施方式的用于训练神经网络的示例操作的说明性流程图。

[0015] 图8示出了描绘根据一些实施方式的用于训练神经网络的示例操作的说明性流程图。

具体实施方式

[0016] 在以下描述中,阐述了许多具体细节,诸如具体部件、电路和过程的示例,以提供对本公开的透彻理解。如本文所使用的术语“耦合”意指直接连接到或通过一个或多个中间部件或电路连接。术语“电子系统”和“电子设备”可以互换使用,以指代能够电子地处理信息的任何系统。此外,在以下描述中并且出于解释的目的,阐述了具体命名法以提供对本公开的各方面的透彻理解。然而,对于本领域技术人员将显而易见的是,可以不需要这些具体细节来实践示例实施例。在其他实例中,以框图形式示出了公知的电路和设备,以避免使本公开模糊不清。以下详细描述的一些部分是根据对计算机存储器内的数据位的操作的过程、逻辑块、处理和其他符号表示来呈现的。

[0017] 这些描述和表示是数据处理领域的技术人员用来最有效地将他们工作的实质传达给本领域其他技术人员的手段。在本公开中,过程、逻辑块、进程等被认为是导致期望结果的步骤或指令的自洽序列。这些步骤是需要对物理量进行物理操纵的步骤。通常,尽管不一定,这些量采取能够在计算机系统中存储、传送、组合、比较和以其他方式操纵的电信号或磁信号的形式。然而,应当记住,所有这些和类似术语将与适当的物理量相关联,并且仅仅是应用于这些量的方便标签。

[0018] 除非另有明确说明,否则如从以下讨论中显而易见的是,应当理解,在整个本申请中,利用诸如“访问”、“接收”、“发送”、“使用”、“选择”、“确定”、“归一化”、“相乘”、“平均”、“监视”、“比较”、“应用”、“更新”、“测量”、“导出”等术语的讨论是指计算机系统或类似电子计算设备的动作和过程,其将表示为计算机系统的寄存器和存储器内的物理(电子)量的数据操纵和转换成类似地表示为计算机系统存储器或寄存器或其他这样的信息存储、传输或显示设备内的物理量的其他数据。

[0019] 在附图中,单个框可以被描述为执行一个或多个功能;然而,在实际实践中,由该框执行的一个或多个功能可以在单个部件中或跨多个部件执行,和/或可以使用硬件、使用软件或使用硬件和软件的组合来执行。为了清楚地说明硬件和软件的这种可互换性,下面

已经在其功能性方面对各种说明性部件、块、模块、电路和步骤进行了总体描述。此类功能性是实现为硬件还是软件取决于特定应用及强加于整个系统的设计约束。所属领域的技术人员可针对每一特定应用以不同方式实现所描述功能性,但不应将此类实施方式决策解释为导致脱离本发明的范围。此外,示例输入设备可以包括除了所示出的那些部件之外的部件,包括诸如处理器、存储器等公知的部件。

[0020] 本文描述的技术可以以硬件、软件、固件或其任何组合来实现,除非特别描述为以特定方式实现。被描述为模块或部件的任何特征也可以在集成逻辑设备中一起实现,或者单独地实现为分立但可互操作的逻辑设备。如果以软件实现,那么所述技术可至少部分地由包括指令的非暂时性处理器可读存储介质来实现,所述指令在被执行时执行上文所描述的方法中的一个或多个。非暂时性处理器可读数据存储介质可形成计算机程序产品的部分,所述计算机程序产品可包括封装材料。

[0021] 非暂时性处理器可读存储介质可以包括随机存取存储器(RAM)(诸如同步动态随机存取存储器(SDRAM))、只读存储器(ROM)、非易失性随机存取存储器(NVRAM)、电可擦除可编程只读存储器(EEPROM)、闪存、其他已知的存储介质等。另外或替代地,所述技术可至少部分地由处理器可读通信介质实现,所述处理器可读通信介质携带或传达呈指令或数据结构形式的代码且可由计算机或其它处理器存取、读取和/或执行。

[0022] 结合本文公开的实施例描述的各种说明性逻辑块、模块、电路和指令可以由一个或多个处理器(或处理系统)执行。如本文所使用的术语“处理器”可以指代能够执行存储在存储器中的一个或多个软件程序的脚本或指令的任何通用处理器、专用处理器、常规处理器、控制器、微控制器和/或状态机。

[0023] 本公开的方面提供了用于对图像进行去噪同时减少按顺序显示的图像(例如,作为视频)之间的闪烁的系统和方法。在一些实施例中,可以训练神经网络(例如,递归神经网络)以推断场景的图像的序列(也称为“批次”)的去噪表示。更具体地,可以训练神经网络(也称为“神经网络算法”或“神经网络模型”)以减少每个图像中的噪声,同时在序列中连续显示的图像之间保持一致的外观。

[0024] 在训练期间,机器学习系统可以获得场景的捕获的输入图像序列。机器学习系统还可以获得一个或多个真实数据(ground truth)图像,所述一个或多个真实数据图像是场景的可能包含很少噪声或不包含噪声的参考图像。在一些实施例中,真实数据图像可以是具有比序列中的任何输入图像高的信噪比(SNR)的任何图像。在一些其他实施例中,真实数据图像可以是具有比序列中的任何输入图像高的分辨率(例如,像素分辨率)的任何图像。此外,在一些实施例中,机器学习系统可以基于神经网络算法将输入图像序列中的第一输入图像映射到第一输出图像。机器学习系统还可以基于第一输出图像与真实数据图像之间的差异来确定第一网络损失。机器学习系统还可以基于神经网络算法将输入图像序列中的第二输入图像映射到第二输出图像;并且基于第二输出图像和真实数据图像之间的差异来确定第二网络损失。第一网络损失和第二网络损失中的每一个可以分别表示第一输出图像和第二输出图像与真实数据图像之间的差异。例如,这样的差异可以反映每个输出图像中的误差或噪声的量。机器学习系统还可以基于第一输出图像和第二输出图像之间的差异来确定一致性损失。因此,一致性损失可以表示第一输出图像与第二输出图像之间的差异。例如,这样的差异可以反映序列中的连续或不间断图像上的噪声或外观的变化量。

[0025] 在一些实施例中,机器学习系统可以至少部分地基于第一网络损失和第二网络损失以及一致性损失来更新神经网络算法,例如,在训练过程的每次迭代结束时。在一些方面,机器学习系统可以重复训练过程,直到满足某些收敛标准。在一些实施例中,经训练的神经网络模型可以部署在例如图像捕获和显示系统中,并且用于实时推断图像的去噪表示。通过基于一致性损失训练神经网络,神经网络模型可以最小化按顺序显示的图像中的噪声的变化。因此,当去噪图像被按顺序显示时,诸如在视频中,视频看起来平滑且一致,具有最小的闪烁或没有闪烁。

[0026] 图1示出了根据一些实施方式的可以采用神经网络模型的示例图像捕获和显示系统100的框图。系统100包括图像捕获设备110、图像处理器120和显示设备130。图像捕获设备110(例如,相机)捕获来自场景的光101(也称为“场景光101”)的图案的序列,并且将场景光101转换为数字图像捕获数据102(也称为“图像捕获数据102”)。对于序列中的每个光图案,图像捕获数据102可以包括表示场景的数字图像的像素(或像素值)阵列。显示设备130(诸如电视、计算机监视器、智能电话或包括电子显示器的任何其他设备)通过在相关联的显示表面上再现光图案的序列来渲染或显示数字图像。

[0027] 图像处理器120将图像捕获数据102转换为图像渲染数据103,当在显示设备130上显示时,图像渲染数据103可以将由图像捕获设备110捕获的原始场景反映为具有很少或没有闪烁的视频。例如,图像处理器120可以被配置为校正和平滑与图像捕获数据102相关联的各种像素失真。换句话说,图像处理器120可以被配置为减少和最小化与数字图像相关联的噪声的变化。尽管在图1中描绘为独立框,但在实际实施方案中,图像处理器120可并入或以其它方式包含于图像捕获装置110、显示装置130或其组合中。

[0028] 在一些实施例中,图像处理器120可以基于通过机器学习训练的神经网络模型123来创建图像捕获数据102(例如,图像渲染数据103)的去噪表示。机器学习是用于提高计算机系统或应用执行某个任务的能力的技术。在训练阶段期间,可以向机器学习系统提供多个“答案”和要映射到每个答案的一组或多组原始数据。例如,机器学习系统可以被训练为通过向系统提供在低光条件下捕获的场景的短曝光图像序列(其表示原始数据)和包含很少或不包含噪声的同一场景的一个或多个代表性(或真实数据)图像(其表示答案)来对低光图像序列执行去噪操作。然后,机器学习系统可以分析原始数据以“学习”可以用于再现答案的一组规则。例如,机器学习系统可以对原始数据执行统计分析,以确定与关联于原始数据的噪声或噪声变化相关的共同特征集(也称为“规则”)。

[0029] 在一些方面,机器学习系统可以采用深度学习,深度学习是机器学习的特定形式,其中被训练的模型是多层神经网络。由于处理信息的方式(类似于生物神经系统),深度学习架构通常被称为人工神经网络。例如,深度学习架构的每个层可以由多个人工神经元组成。神经元可以跨各种层互连,使得输入数据(或原始数据)可以从一层传递到另一层。更具体地,每个神经元层可以对输入数据执行不同类型的变换,这将最终导致期望的输出。神经元的互连框架可以被称为神经网络模型。因此,神经网络模型123可以包括可以用于“推断”图像捕获数据102的去噪表示的一组规则。因此,图像处理器120可以使用神经网络模型123来减少和最小化与图像捕获数据102相关联的噪声的变化。

[0030] 图2示出了根据一些实施方式的采用神经网络模型的图像处理器的示例操作200的框图。图像处理器220可以是图1的图像处理器120的实施例。因此,图像处理器220可以被

配置为接收图像202并输出去噪图像203。参考图1,图像202可以是图像捕获数据102的一个示例,并且去噪图像203可以是图像渲染数据103的一个示例。

[0031] 在一些实施例中,图像处理器220可以包括神经处理器221和神经网络模型223。神经处理器221可以包括被配置为加速神经网络推断的一个或多个处理器。例如,神经处理器221的硬件架构可以被设计为比通用处理器快速或有效地遍历神经网络。神经处理器221可以基于神经网络模型223来实现神经网络。在一些实施例中,神经网络模型223可以是图1的神经网络模型123的一个示例。因此,神经网络模型223可以包括可以用于推断捕获图像的去噪表示的一组规则。

[0032] 如图2所示,图像202看起来有噪声或颗粒状。例如,图像202中的像素的分布包含亮度和颜色的“尖峰”(诸如场景的亮区域中的暗像素和场景的暗区域中的亮像素)。相比之下,去噪图像203中的像素的颜色和亮度遵循基本上平滑的分布。例如,在去噪图像203中,场景的暗区域看起来是暗的,并且场景的亮区域看起来是亮的。虽然图2仅示出了由图像处理器220处理的单个有噪声图像202,但是在实际实施方式中,有噪声图像(或视频)的序列可以由图像处理器220处理。此外,图像处理器220可以使用神经处理器221和神经网络模型223来输出去噪图像序列,当在显示屏上显示时,该去噪图像序列表现为平滑或稳定的视频。

[0033] 为了产生去噪图像的序列,可以在被集成在图像处理器220中之前训练神经网络模型223。更具体地,神经网络模型223可以由机器学习系统训练以推断场景的捕获的噪声输入图像序列的去噪表示。在一些实施例中,训练可以包括使用神经网络算法将输入图像序列中的第一输入图像映射到第一输出图像,以及基于第一输出图像与真实数据图像之间的差异来确定第一网络损失。第一网络损失可以表示第一输出图像中相对于真实数据图像的噪声量。在一些实施例中,训练还可以包括使用神经网络算法将输入图像序列中的第二输入图像映射到第二输出图像,以及基于第二输出图像与真实数据图像之间的差异来确定第二网络损失。第二网络损失可以表示第二输出图像中相对于真实数据图像的噪声量。

[0034] 在一些实施例中,可以将第一输出图像与第二输出图像进行比较以确定一致性损失,其可以表示第一输出图像与第二输出图像之间的差异。例如,这样的差异可以反映跨第一输出图像和第二输出图像的噪声或外观的变化量。在一些实施例中,第一网络损失和第二网络损失以及一致性损失可以用于更新神经网络算法的权重。此外,在一些实施例中,机器学习系统可以重复上述训练过程,直到满足某些收敛标准。

[0035] 通过基于第一和第二网络损失更新神经网络算法,神经网络算法可以被配置为减少输出图像序列中的噪声。此外,通过基于一致性损失更新神经网络算法,神经网络算法可以被配置为减少输出图像序列中噪声的变化量,使得当输出图像被显示为视频时,它们可以表现为具有很少或没有闪烁的平滑或稳定的视频。

[0036] 图3示出了根据一些实施方式的示例机器学习系统300的框图。机器学习系统300包括神经网络322、损失计算器326和网络优化器328。在一些方面,机器学习系统300可以用于训练神经网络322以产生神经网络模型323。神经网络模型323可以分别是图1和图2的神经网络模型123和/或223的实施例。

[0037] 如图3所示,机器学习系统300可以获得输入图像302的序列(也称为“输入图像302”)和一个或多个真实数据图像325。在一些实施例中,每个输入图像302的SNR可以小于

一个或多个真实数据图像325的SNR。例如,在一些实施例中,输入图像302中的每一个可以表示低光条件下的场景的有噪声的短曝光图像,并且一个或多个真实数据图像325中的每一个可以表示低光条件下的场景的长曝光图像,其具有很少或没有噪声。此外,在一些实施例中,每个输入图像302的分辨率(例如,像素分辨率)可以小于一个或多个真实数据图像325的分辨率。在这样的实施例中,机器学习系统300可以使用一个或多个真实数据图像325来放大每个输入图像302(或执行超分辨率)。

[0038] 在一些实施例中,输入图像302中的每一个和一个或多个真实数据图像325中的每一个可以具有相应的亮度水平。例如,图像的亮度水平可以反映图像中描绘的场景的光照(或照明度)、图像的曝光和/或用于捕获图像的图像捕获设备的光电传感器的增益(或灵敏度)。在一些实施例中,输入图像302中的每一个可以被预处理,使得每个输入图像302的亮度水平可以等于真实数据图像325的亮度水平。

[0039] 如图3所示,神经网络322可以被配置为接收每个输入图像302并基于所接收的输入图像302产生相应的输出图像303。更具体地,神经网络322可以通过尝试重建一个或多个真实数据图像325来产生输入图像302的去噪表示。在一些实施例中,神经网络322可以是递归神经网络,其被配置为学习输入图像302的序列中的噪声的时间动态。在一些方面,神经网络322可以形成跨多层人工神经元的加权连接的网络,其以输入图像302的序列开始并且导致输出图像的去噪序列303。在一些实施例中,神经网络322可以被配置为将输出图像的去噪序列303中的每个图像提供给损失计算器326。

[0040] 如图3所示,损失计算器326可以被配置为从神经网络322接收输出图像的去噪序列303。损失计算器326还可以被配置为接收一个或多个真实数据图像325。在一些方面,损失计算器326可被配置为确定输出图像303的序列中的每一图像中相对于一个或多个真实数据图像325的网络损失(或误差或噪声)的量。例如,为了确定第一网络损失,损失计算器326可以将第一输出图像303与真实数据图像325进行比较;并且为了确定第二网络损失,损失计算器326可以将第二输出图像303与真实数据图像325进行比较。损失计算器326还可以被配置为通过将输出图像303的序列中的任何剩余输出图像中的每一个与真实数据图像325进行比较来确定附加网络损失。

[0041] 在一些方面,损失计算器326还可以被配置为确定输出图像303的序列之间的一致性损失(或噪声的变化量)。例如,为了确定一致性损失,损失计算器326可以将第一输出图像303与第二输出图像303进行比较,其中第一输出图像303和第二输出图像303是输出图像303的序列中的连续图像。在一些实施例中,损失计算器326还可以被配置为基于输出图像303的序列中的任何剩余的连续图像来确定一致性损失。在一些方面,损失计算器326可以被配置为组合一致性损失、第一网络损失、第二网络损失和任何附加网络损失,以确定总损失305。损失计算器326还可以被配置为将总损失305输出到网络优化器328。

[0042] 在一些方面,网络优化器328可以被配置为接收总损失305并且基于总损失305来调整神经网络322的一个或多个权重(也称为“参数”)307。更具体地,网络优化器328可以减少总损失305的方式调整权重307。机器学习系统300可以在一次或多次迭代中重复上述训练过程,直到满足某些收敛标准。例如,可以将新的噪声图像序列作为输入提供给神经网络322,损失计算器326可以基于神经网络322的输出来确定总损失305,并且网络优化器328可以进一步基于总损失305来更新权重307。

[0043] 图4示出了根据一些实施方式的示例损失计算器400的框图。损失计算器400可以是图3的损失计算器326的实施例。如图4所示,损失计算器400包括网络损失计算器432、一致性损失计算器434和总损失计算器436。

[0044] 网络损失计算器432可以被配置为从神经网络(诸如图3的神经网络322)接收输出图像403的序列(也称为“输出图像403”)中的每个图像。网络损失计算器432还可以被配置为接收一个或多个真实数据图像425。

[0045] 在一些方面,输出图像403的序列可以由以下四个维度表征:(i)批次大小N;(ii)高度H;(iii)宽度W;以及(iv)通道C。批次大小N可以表示输出图像403的序列中的图像总数。高度H可以表示输出图像403的序列中的每个图像的以像素为单位的高度。宽度W可以表示输出图像403的序列中的每个图像的以像素为单位的宽度。通道C可以表示输出图像403的序列中的每个图像中的每个像素的颜色(例如,红色、绿色、蓝色和/或不透明度)或每个像素的值。在一些实施例中,通道C可以由一个或多个数值表示。应注意,如同输出图像403的序列,一个或多个真实数据图像425和输入图像序列(例如,图3的输入图像302的序列)可由四个维度N、H、W和C表征。

[0046] 在一些方面,网络损失计算器432可以被配置为确定总网络损失433(也称为“网络损失433”),其可以表示输出图像403的序列中相对于一个或多个真实数据图像425的误差或噪声量。网络损失计算器432可以基于输出图像403的序列中的每个图像的像素值与一个或多个真实数据图像425的像素值之间的差异来确定网络损失433。例如,在一些实施例中,网络损失计算器432可以被配置为通过将第一输出图像403的像素值与真实数据图像425的像素值进行比较来确定第一网络损失。网络损失计算器432还可以被配置为通过将第二输出图像403的像素值与真实数据图像425的像素值进行比较来确定第二网络损失。在第一和第二输出图像403表示输出图像403的序列中的每个图像的情况下,网络损失计算器432可以组合第一和第二网络损失以生成总网络损失433。

[0047] 在一些实施例中,网络损失计算器432可以被配置为基于一个或多个金字塔损失来确定网络损失433。金字塔损失可表示输出图像403的像素值相对于处于各种金字塔层级(或尺度)的真实数据图像425的像素值的误差总量。金字塔层级(或尺度)可指代图像(诸如输出图像403或真实数据图像425)的像素粒度。例如,较高的金字塔层级可以与较粗糙的像素粒度或较少的像素值相关联。因此,与处于较低金字塔层级的相同图像相比,图像可看起来“缩小”(且因此展现较少细节)。在一些实施例中,网络损失433可以包括至少一个输出图像403的金字塔损失。此外,在一些实施例中,网络损失433可以包括输出图像403的序列中的每个图像的金字塔损失。

[0048] 一致性损失计算器434,如网络损失计算器432,可以被配置为接收输出图像403的序列中的每个图像。在一些方面,一致性损失计算器434可以被配置为确定一致性损失435,一致性损失435可以表示输出图像403的序列中的两个或更多个连续图像之间的噪声变化量。换句话说,一致性损失435可以表示输出图像403的序列中的两个或更多个连续图像之间的不一致性的量或程度。在一些方面,一致性损失435可以基于标准偏差,并且可以表示如下:

$$\text{一致性损失} = \sum_{n=0}^N \sum_{h=0}^H \sum_{w=0}^W \sum_{c=0}^C (p_{cwhn} - m_{cwh})^2 \quad (\text{等式 1A})$$

在等式1A中，N、H、W和C表示输出图像403的序列的四个维度，并且n、h、w和c是分别对应于维度N、H、W和C的索引。 p_{cwhn} 表示通道c、宽度w、高度h和批次索引（也称为“图像索引”）n处的像素值。 m_{cwh} 表示跨批次维度（例如，从图像索引n=0到n=N）计算的通道c、宽度w和高度h处的像素的均值（或平均值）。此外， m_{cwh} 可以表示如下：

$$m_{cwh} = \frac{\sum_{n=0}^N p_{cwhn}}{N} \quad (\text{等式 1B})$$

注意，当输出图像403的序列（或对应的输入图像的序列）的批次大小N在机器学习系统（例如，图3的机器学习系统300）经历训练过程的多次迭代时保持恒定时，上面的等式1A和1B可以适合于使用。

[0049] 在一些其他实施例中，一致性损失435可以表示基于输出图像403的序列的矩阵(B)（也称为“矩阵B”）的秩。更具体地，矩阵B可以包括一系列向量，其中每个向量对应于输出图像403的序列中的图像。此外，对于输出图像403的序列中的给定图像，每个向量可以被平坦化或减小到表示维度H、W和C中的每一个的单个维度。在一些方面，矩阵B可以表示如下：

$$\text{矩阵 B} = [\text{vec}(\text{输出图像}_1), \text{vec}(\text{输出图像}_2), \dots, \text{vec}(\text{输出图像}_N)]$$

矩阵B具有两个维度，其中一个维度包括H、W和C，并且另一个维度包括N。本公开的方面认识到，矩阵B的秩可以表示矩阵B中的线性独立输出图像403的最大数量，并且矩阵B的秩越高，输出图像403的序列之间存在的（或不一致性的变化）的量越大。

[0050] 虽然一致性损失435可以基于矩阵B或平均像素值（如上面的等式1A和1B所示），但是在一些实施例中，一致性损失435可以基于输出图像403的序列中的图像之间的某些关系。例如，在一些实施例中，输出图像403的序列可以描绘动态场景，诸如描绘运动中的对象的场景。输出图像403的序列中的第一图像可以包括与场景中的对象的第一位置相对应的一个或多个像素，并且输出图像403的序列中的第二图像可以包括与场景中的对象的第二位置相对应的一个或多个像素。在一些方面，可基于对应于第一图像中的对象的位置的一个或多个像素和对应于第二图像中的对象的位置的一个或多个像素来定义对象的移动。一旦定义了移动，就可以修改或扭曲输出图像403的序列中的第一和/或第二图像，并且可以基于定义的移动和/或修改的第一和第二图像来确定一致性损失435。虽然上面已经描述了基于运动（或定义的移动）的关系，但是在一些实施例中，该关系可以基于输出图像403的序列之间的像素值（例如，像素强度）的变化。

[0051] 如图4所示，网络损失433和一致性损失435被提供给总损失计算器436。总损失计算器436可以被配置为组合网络损失433和一致性损失435以确定总损失405，其可以表示如下：

$$\text{总损失} = \alpha * \text{网络损失} + (1 - \alpha) * y * \text{一致性损失} \quad (\text{等式 2})$$

在等式2中， α 是用于平衡网络损失433和一致性损失435的标量，并且y可以被设置为0或1的值，这取决于输出图像403的序列的内容。在一些实施例中，输入图像序列（诸如图3的输入图像302的序列）可以被标记（或分类）为描绘静态场景、动态场景、或静态场景和动态场景两者。标记可以通过任何合适的手动或自动方法进行。此外，标记可以在训练操作期间从输入图像序列转移到对应的输出图像序列（例如，分别是图3和图4的输出图像303和/

或403的序列)。

[0052] 在一些实施例中,在输出图像403的序列描绘静态场景的情况下,y可以被设置为1。在一些其他实施例中,在输出图像403的序列描绘动态场景的情况下,y可以被设置为0。此外,在一些其他实施例中,在输出图像403的序列描绘静态场景和动态场景两者的情况下,y可以是大小为N(输出图像403的序列的批次大小)的向量。也就是说,y可以是N个分量的向量,其中每个分量对应于输出图像403的序列中的相应图像。此外,如果分量对应于作为静态场景的一部分的输出图像403,则分量可以被设置为1,并且如果分量对应于作为动态场景的一部分的输出图像403,则分量可以被设置为0。

[0053] 一旦确定了总损失405,网络优化器(诸如图3的网络优化器328)就可以使用总损失405来确定是否满足某些收敛标准。例如,在总损失405下降到阈值水平以下的情况下,和/或在预定数量的训练迭代已经完成的情况下,网络优化器可以确定神经网络(诸如图3的神经网络322)被优化。因此,网络优化器可以不更新神经网络的权重(诸如图3的权重307)。作为另一示例,在总损失405高于阈值水平的情况下,和/或在预定数量的训练迭代尚未完成的情况下,网络优化器可以确定神经网络尚未被优化。因此,网络优化器可以确定使总损失405最小化的一个或多个权重,以便更新神经网络的权重。

[0054] 在一些方面,网络优化器可以使用随机梯度下降方法来最小化总损失405并确定一个或多个权重。除了随机梯度下降方法之外,网络优化器可以使用附加技术来最小化总损失405的一致性损失分量。例如,在使用上面的等式1A和1B计算一致性损失435的情况下,可以使用Tensorflow或Keras软件库来最小化一致性损失435。作为另一示例,在一致性损失435由矩阵B的秩表示的情况下,可以通过用其非凸代理项(surrogate)、核范数替换秩并使用L1-范数稀疏分解和增广拉格朗日乘数(ALM)方法来最小化秩(其是非凸的)。

[0055] 图5示出了描绘与金字塔损失相关联的示例金字塔层级(或尺度)的说明性图表500。在一些方面,金字塔损失可表示处于不同金字塔层级(例如,图5的金字塔层级0和1)的输出图像(例如,图5的输出图像503或504)与真实数据图像(例如,图5的真实数据图像525)之间的差异。在一些实施例中,金字塔损失可以由网络损失计算器(例如,图4的网络损失计算器432)确定。

[0056] 金字塔层级(或尺度)可指代图像(诸如输出图像或真实数据图像)的像素粒度。例如,较高的金字塔层级可以与较粗糙的像素粒度或较少的像素值相关联。因此,与处于较低金字塔层级的相同图像相比,图像可看起来缩小(或展现较少细节)。例如,与较高金字塔层级相关联的每一像素值可表示与较低金字塔层级相关联的两个或更多个像素值(例如,两个或更多个邻近像素)的平均值。相比之下,较低金字塔层级可与较精细像素粒度或较多像素值相关联。因此,图像可看起来比处于较高金字塔层级的相同图像放大(或展现更多细节)。举例来说,与最低金字塔层级(例如,图5中的金字塔层级0)相关联的像素值可表示输出图像或真实数据图像的原始(或未经缩放)像素值。

[0057] 在一些实施例中,网络损失计算器可以被配置为使用池化(也称为“下采样”),诸如平均池化(如上所述)、最大池化或最小池化,以产生与除了最低金字塔层级之外的每个金字塔层级相关联的缩放数量的像素值。例如,网络损失计算器可以池化与最低金字塔层级(例如,图5的金字塔层级0)相关联的像素值,以产生与下一最低金字塔层级(例如,图5的金字塔层级1)相关联的缩放像素值。在一些实施例中,网络损失计算器可以被配置为基于

预定池化大小来计算图像的金字塔损失,其中池化大小包括两个或更多个像素。例如,参考图5,网络损失计算器可以基于四个像素值的预定池化大小来计算输出图像503的金字塔损失。也就是说,网络损失计算器可以池化与输出图像503的金字塔层级0相关联的四个像素值,以产生与输出图像503的金字塔层级1相关联的缩放像素值。此外,在一些实施例中,网络损失计算器可以被配置为基于预定数量的金字塔层级(例如,两个金字塔层级,诸如图5中所示的金字塔层级0和1)来计算图像的金字塔损失。此外,在一些实施例中,网络损失计算器可以被配置为至少部分地基于最高金字塔层级的预定(或目标)数量的像素值来计算图像的金字塔损失。例如,参考图5,网络损失计算器可以至少部分地基于与输出图像503的金字塔层级1相关联的单个像素值来计算输出图像503的金字塔损失。

[0058] 如图5中所展示,金字塔层级0和金字塔层级1可用于确定输出图像503及504中的每一个的金字塔损失。如上文所描述,金字塔层级0是图5中所示的最低金字塔层级。因此,金字塔层级0与表示输出图像503、输出图像504和真实数据图像525的相应像素的原始(或未缩放)像素值的像素值相关联。换句话说,图5描绘了处于金字塔层级0的输出图像503、输出图像504和真实数据图像525中的每一个的四个像素的子集。即,在金字塔层级0处,图5示出输出图像503的像素(或像素值) P_{01} 、 P_{02} 、 P_{03} 和 P_{04} ;输出图像504的像素(或像素值) P_{01}' 、 P_{02}' 、 P_{03}' 和 P_{04}' ;以及真实数据图像525的像素(或像素值) P_{01}'' 、 P_{02}'' 、 P_{03}'' 和 P_{04}'' 。虽然图5中未示出,但在一些实施例中,两个像素值、三个像素值、或五个像素值或更多像素值的子集可表示处于金字塔层级0的输出图像503、输出图像504和真实数据图像525的相应像素的原始像素值。

[0059] 为了确定输出图像503的金字塔损失,网络损失计算器可以首先计算输出图像503的与金字塔层级0相关联的损失量(也称为“ L_0 损失”)和输出图像503的与金字塔层级1相关联的损失量(也称为“ L_1 损失”)。为了计算输出图像503的 L_0 损失,网络损失计算器可以对处于金字塔层级0的输出图像503和真实数据图像525的对应像素值之间的差的绝对值求和,如下面的等式3所示:

$$L_0 \text{ 损失} = |P_{01} - P_{01}''| + |P_{02} - P_{02}''| + |P_{03} - P_{03}''| + |P_{04} - P_{04}''| \quad (\text{等式 3})$$

如图5所示,处于金字塔层级0的输出图像503的像素具有以下像素值: $P_{01}=0$ 、 $P_{02}=0.5$ 、 $P_{03}=0.5$ 和 $P_{04}=0$ 。此外,真实数据图像525的像素 P_{01}'' 、 P_{02}'' 、 P_{03}'' 和 P_{04}'' 中的每一个具有0.25的像素值。当将处于金字塔层级0的输出图像503和真实数据图像525的像素值代入等式3时,获得为1的 L_0 损失,如下所示:

$$L_0 \text{ 损失} = |0 - 0.25| + |0.5 - 0.25| + |0.5 - 0.25| + |0 - 0.25| = 1$$

在可以计算输出图像503的 L_1 损失之前,网络损失计算器可以使用池化来生成输出图像503和真实数据图像525的处于金字塔层级1的像素值。例如,如图5所示,网络损失计算器可以通过跨四个相邻像素(例如, P_{01} 、 P_{02} 、 P_{03} 和 P_{04})对处于金字塔层级0的输出图像503的像素值求平均来采用平均池化,以生成处于金字塔层级1的输出图像503的相应像素值 P_1 。可以如下所示计算 P_1 的值:

$$P_1 = \frac{P_{01} + P_{02} + P_{03} + P_{04}}{4} = \frac{0 + 0.5 + 0.5 + 0}{4} = 0.25$$

类似地,网络损失计算器可以使用平均池来平均处于金字塔层级0的真实数据图像525的像素值,以确定处于金字塔层级1的真实数据图像525的相应像素值 P_1'' 。可以如下

所示计算P1”的值：

$$P1'' = \frac{P0_1'' + P0_2'' + P0_3'' + P0_4''}{4} = \frac{0.25 + 0.25 + 0.25 + 0.25}{4} = 0.25$$

为了计算输出图像503的 L_1 损失，网络损失计算器可以确定在处于金字塔层级1的输出图像503的像素P1与真实数据图像525的像素P1”之间的差的绝对值，如下面的等式4所示：

$$L_1 \text{ 损失} = |P1 - P1''| \quad (\text{等式 4})$$

当将处于金字塔层级1的输出图像503和真实数据图像525的像素值代入等式4时，获得为0的 L_1 损失，如下所示：

$$L_1 \text{ 损失} = |P1 - P1''| = |0.25 - 0.25| = 0$$

更一般地，网络损失计算器可以使用下面的等式5A确定具有任何数量(N')像素的输出图像的任何数量(i)的金字塔层级上的金字塔损失：

$$\text{金字塔损失 (输出图像)} = \sum_i \frac{|\text{池化}_i(\text{输出图像}) - \text{池化}_i(\text{地面实况图像})|}{N'_i} \quad (\text{等式 5A})$$

在等式5A中，池化 $_i$ (图像 x)表示通过缩放因子 i 对图像 x (例如，输出图像或真实数据图像)的池化或下采样。与第 i 个金字塔层级相关联的损失量可以表示如下：

$$L_i = |\text{池化}_i(\text{输出图像}) - \text{池化}_i(\text{地面实况图像})|$$

当将以上针对 L_i 的表达式代入等式5A中时，获得针对金字塔损失的以下替代表达式：

$$\text{金字塔损失 (输出图像)} = \frac{L_0}{N'_0} + \frac{L_1}{N'_1} + \dots + \frac{L_{z-1}}{N'_{z-1}} \quad (\text{等式 5B})$$

在等式5B中，(z)表示金字塔层级的总数，并且 i 被表示为范围从0到 $z-1$ 的整数值(例如， $0 \leq i \leq z-1$)。当输出图像503的损失量和与金字塔层级0和1相关联的像素数量被代入等式5B中时，获得为0.25的金字塔损失，如下所示：

$$\text{金字塔损失 (输出图像 503A)} = \frac{L_0}{N'_0} + \frac{L_1}{N'_1} = \frac{1}{4} + \frac{0}{1} = 0.25$$

使用上述过程，可以为输出图像504计算0.5的金字塔损失。在一些实施例中，当计算金字塔损失时，可以将不同的权重应用于与不同金字塔层级相关联的损失量(也称为“损失项”) L_0, L_1, \dots, L_{z-1} 。例如，在一些应用中，与较低金字塔层级相关联的损失项可以比与较高金字塔层级相关联的损失项更重地加权。在一些其它应用中，与较高金字塔层级相关联的损失项可比与较低金字塔层级相关联的损失项更重地加权。此外，在一些实施例中，输出图像序列中的一个或多个输出图像的金字塔损失可以用于更新神经网络算法的权重。

[0060] 图6示出了根据一些实施方式的示例机器学习系统600。在一些实施例中，机器学习系统600可以是图3的机器学习系统300的一个示例。因此，机器学习系统600可以被配置为基于输入图像602(也称为输入图像602)的序列和一个或多个真实数据图像625来产生神经网络模型623。在一些实施例中，机器学习系统600可以包括处理系统640和存储器650。

[0061] 存储器650可以包括非暂时性计算机可读介质(包括一个或多个非易失性存储元件，诸如EPROM、EEPROM、闪存、硬盘驱动器等)，其可以存储至少以下软件(SW)模块：

- 神经网络SW模块652,其被配置为通过尝试重建一个或多个真实数据图像625来创建输入图像602的序列(例如,输出图像序列)的去噪表示,以训练神经网络模型623;

- 损失计算器SW模块654,其被配置为基于输出图像序列和一个或多个真实数据图像625来确定总损失量,损失计算器SW模块654还包括:

- o网络损失子模块655,用于基于输出图像序列和一个或多个真实数据图像625来确定总网络损失量;以及

- o第二一致性损失子模块656,用于基于输出图像序列确定一致性损失量;以及

- 网络优化器SW模块658,被配置为基于总损失量来确定神经网络SW模块652的一个或多个经更新的权重。

[0062] 每个软件模块包括当由处理系统640执行时使机器学习系统600执行对应功能的指令。

[0063] 处理系统640可以包括能够执行存储在机器学习系统600中(诸如在存储器650中)的一个或多个软件程序的脚本或指令的任何合适的一个或多个处理器。例如,处理系统640可以执行神经网络SW模块652,以通过尝试重建一个或多个真实数据图像625来创建输入图像602的序列的去噪表示。换句话说,处理系统640可以执行神经网络SW模块652以将输入图像602的序列映射到输出图像序列。

[0064] 处理系统640还可以执行损失计算器SW模块654以确定总损失。在执行损失计算器SW模块654时,处理系统640还可以执行网络损失子模块655和一致性损失子模块656。例如,处理系统640可以执行网络损失子模块655以确定输出图像序列中相对于一个或多个真实数据图像625的网络损失(或噪声)的总量。此外,处理系统640可以执行一致性损失子模块656以确定输出图像序列中噪声的变化量。在一些实施例中,处理系统640还可以执行网络优化器SW模块658,以基于总损失来确定神经网络SW模块652的经更新的权重。

[0065] 图7示出了描绘根据一些实施方式的用于训练神经网络的示例操作700的说明性流程图。示例操作700可以由机器学习系统(诸如分别是图3和图6的机器学习系统300和/或600)执行,以训练神经网络来推断输入图像序列的去噪表示。

[0066] 机器学习系统可以获得场景的捕获的输入图像序列(710)。机器学习系统还可以基于神经网络算法将输入图像序列中的第一输入图像映射到第一输出图像(720)。在一些实施例中,输入图像序列可以表示静态场景的图像。此外,在一些实施例中,神经网络算法可以包括递归神经网络算法。

[0067] 机器学习系统还可以基于第一输出图像与真实数据图像之间的差异来确定第一网络损失(730)。机器学习系统还可以基于神经网络算法将输入图像序列中的第二输入图像映射到第二输出图像(740)。在一些实施例中,真实数据图像的SNR可以大于第一输入图像的SNR和第二输入图像的SNR。在一些其他实施例中,真实数据图像的分辨率可以大于第一输入图像的分辨率和第二输入图像的分辨率。

[0068] 机器学习系统还可以基于第二输出图像与真实数据图像之间的差异来确定第二网络损失(750)。机器学习系统还可以基于第一输出图像与第二输出图像之间的差异来确定一致性损失(760)。在一些方面,第一输出图像和第二输出图像中的每一个可以包括多个像素值。在一些实施例中,机器学习系统可以基于与第一输出图像的多个像素值和第二输出图像的多个像素值相关联的平均像素值来确定一致性损失。在一些其他实施例中,机器

学习系统可以通过生成包括第一输出图像的多个像素值和第二输出图像的多个像素值的矩阵并且通过确定矩阵的秩来确定一致性损失。

[0069] 机器学习系统还可以至少部分地基于第一网络损失、第二网络损失和一致性损失来更新神经网络算法(770)。在一些实施例中,机器学习系统还可以至少部分地基于第一网络损失、第二网络损失和一致性损失来确定总损失。机器学习系统还可以使总损失最小化,并且基于最小化的总损失确定一个或多个参数以更新神经网络算法。

[0070] 图8示出了描绘根据一些实施方式的用于训练神经网络的示例操作800的说明性流程图。在一些实施例中,示例操作800可以由机器学习系统(诸如分别是图3和图6的机器学习系统300和/或600)执行,以训练神经网络来推断图像序列的去噪表示。

[0071] 机器学习系统可以获得场景的捕获的输入图像序列(810)。机器学习系统还可以基于神经网络算法将输入图像序列中的第一输入图像映射到第一输出图像(820)。在一些实施例中,输入图像序列可以表示静态和/或动态场景的图像。此外,在一些实施例中,神经网络算法可以包括递归神经网络算法。

[0072] 机器学习系统还可以基于第一输出图像与第一真实数据图像之间的差异来确定第一网络损失(830)。在一些实施例中,第一真实数据图像的SNR可以大于第一输入图像的SNR。在一些其他实施例中,第一真实数据图像的分辨率可以高于第一输入图像的分辨率。

[0073] 机器学习系统还可以基于神经网络算法将输入图像序列中的第二输入图像映射到第二输出图像(840)。机器学习系统还可以基于第二输出图像与第二真实数据图像之间的差异来确定第二网络损失(850)。在一些实施例中,第二真实数据图像的SNR可以大于第二输入图像的SNR。在一些其他实施例中,第二真实数据图像的分辨率可以高于第二输入图像的分辨率。

[0074] 机器学习系统还可以基于第一输出图像和第二输出图像之间的差异来确定一致性损失(860)。在一些方面,第一输出图像和第二输出图像中的每一个可以包括多个像素值。在一些实施例中,机器学习系统可以基于与第一输出图像的多个像素值和第二输出图像的多个像素值相关联的平均像素值来确定一致性损失。在一些其他实施例中,机器学习系统可以通过生成包括第一输出图像的多个像素值和第二输出图像的多个像素值的矩阵并且通过确定矩阵的秩来确定一致性损失。

[0075] 机器学习系统还可以至少部分地基于第一网络损失、第二网络损失和一致性损失来更新神经网络算法(870)。

[0076] 本领域技术人员将理解,信息和信号可以使用各种不同的技术和技艺中的任何一种来表示。例如,可以通过电压、电流、电磁波、磁场或磁粒子、光场或光粒子、或其任何组合来表示可以贯穿以上描述引用的数据、指令、命令、信息、信号、比特、符号和码片。

[0077] 另外,所属领域的技术人员将了解,结合本文中所公开的方面所描述的各种说明性逻辑块、模块、电路和算法步骤可实现为电子硬件、计算机软件或两者的组合。为了清楚地说明硬件与软件的此可互换性,上文已大体上在其功能性方面描述了各种说明性部件、块、模块、电路和步骤。这种功能性是实现为硬件还是软件取决于特定应用及强加于整个系统的设计约束。所属领域的技术人员可针对每一特定应用以不同方式实现所描述的功能性,但此类实施方式决策不应被解释为导致脱离本公开的范围。

[0078] 结合本文中所公开的方面描述的方法、序列或算法可直接体现于硬件中、由处理

器执行的软件模块中、或两者的组合中。软件模块可以驻留在RAM存储器、闪存、ROM存储器、EPROM存储器、EEPROM存储器、寄存器、硬盘、可移动盘、CD-ROM、或本领域已知的任何其他形式的存储介质中。示例性存储介质耦合到处理器,使得处理器可以从存储介质读取信息和向存储介质写入信息。在替代方案中,存储介质可以集成到处理器。

[0079] 在前述说明书中,已经参考实施例的具体示例描述了实施例。然而,将显而易见的是,在不脱离如所附权利要求中阐述的本公开的更宽范围的情况下,可以对其进行各种修改和改变。因此,说明书和附图被认为是说明性的而不是限制性的。

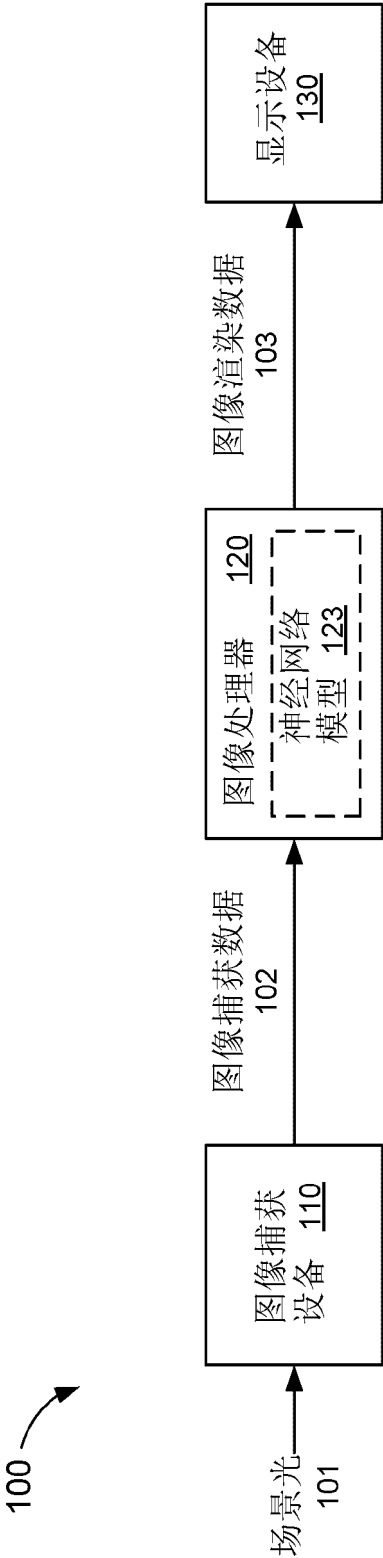


图 1

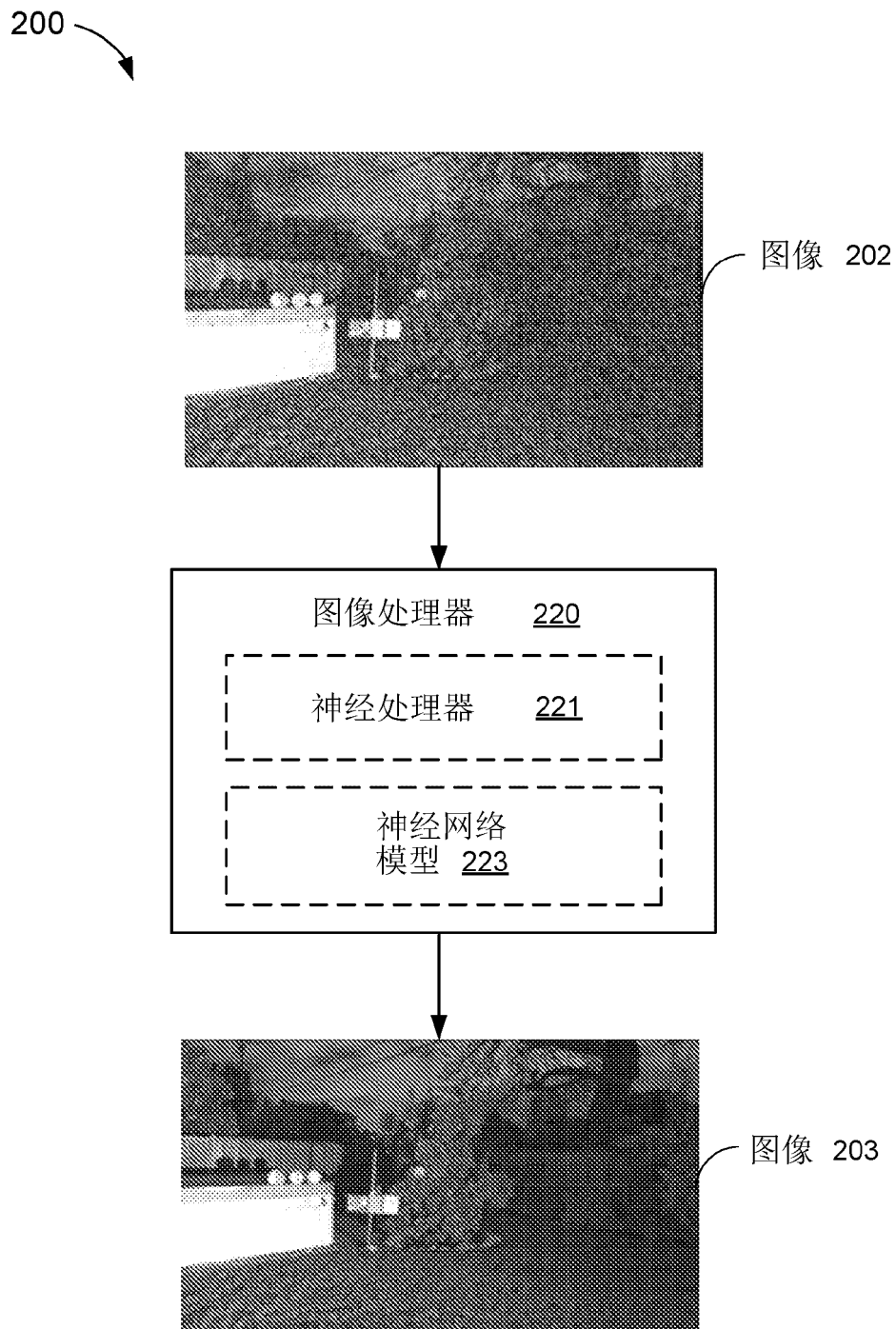


图 2

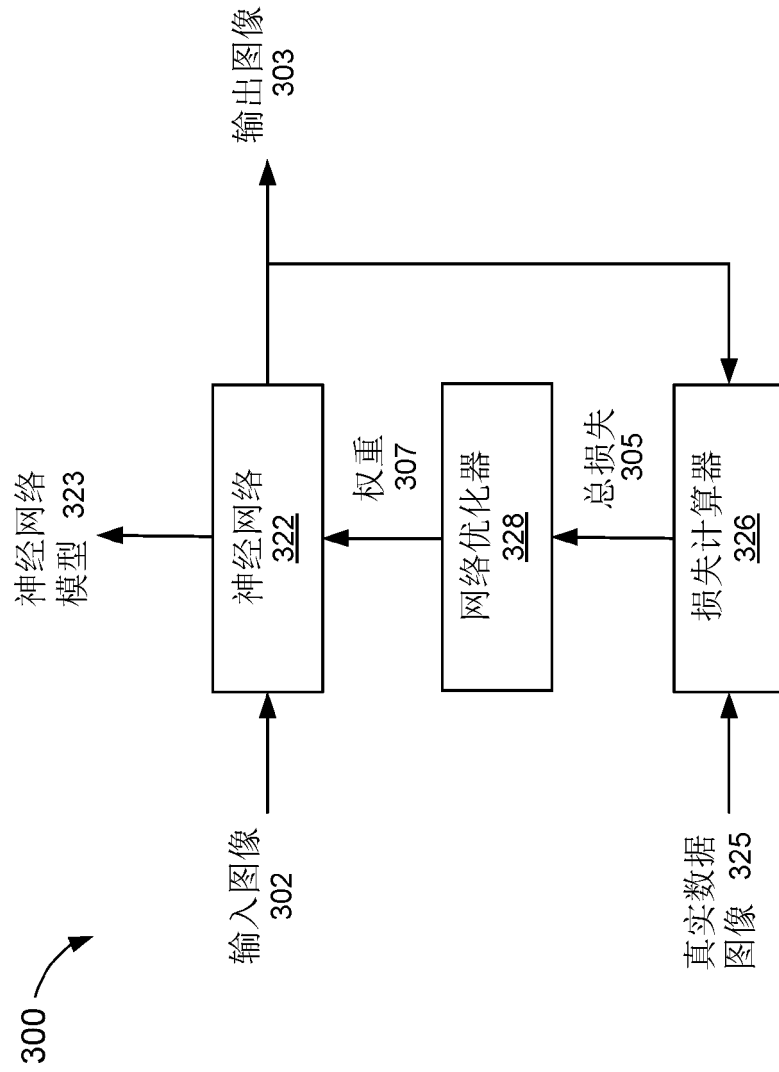


图 3

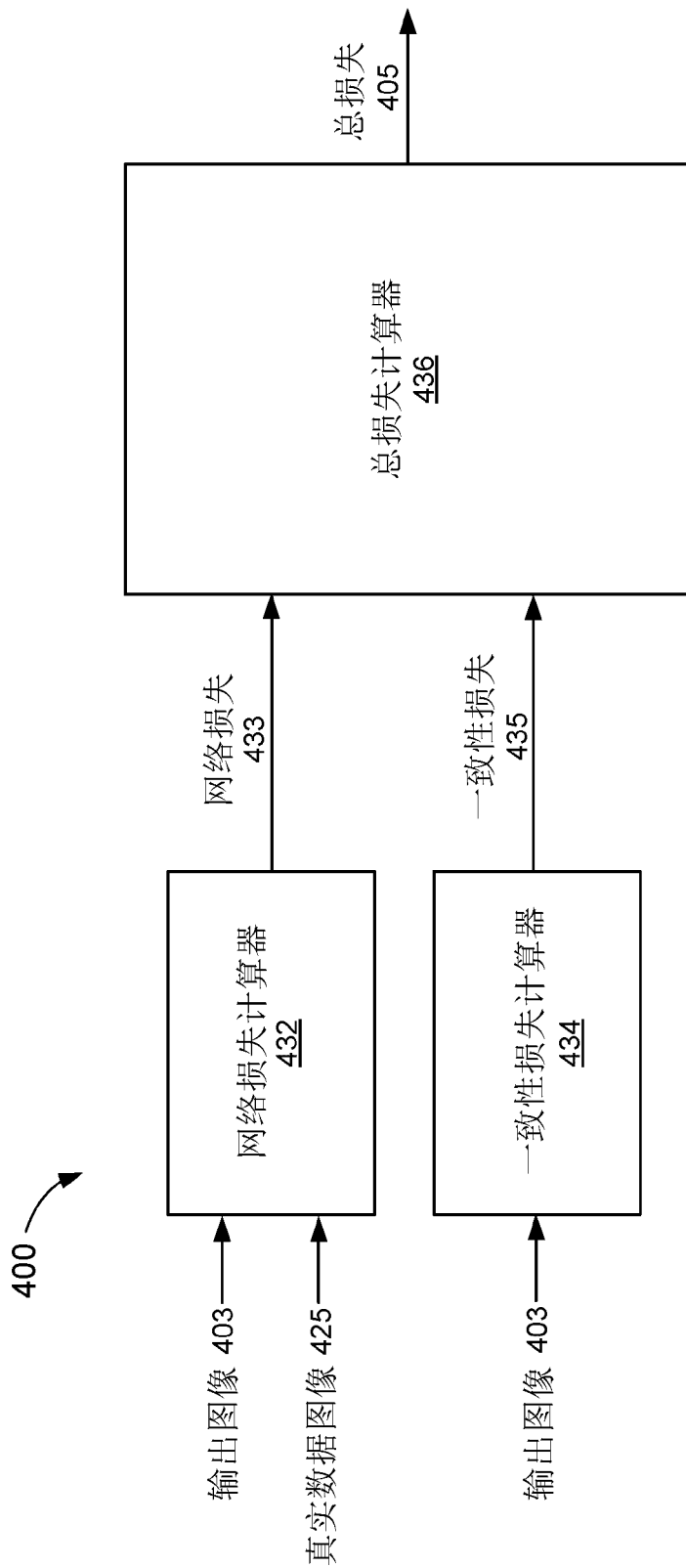


图 4

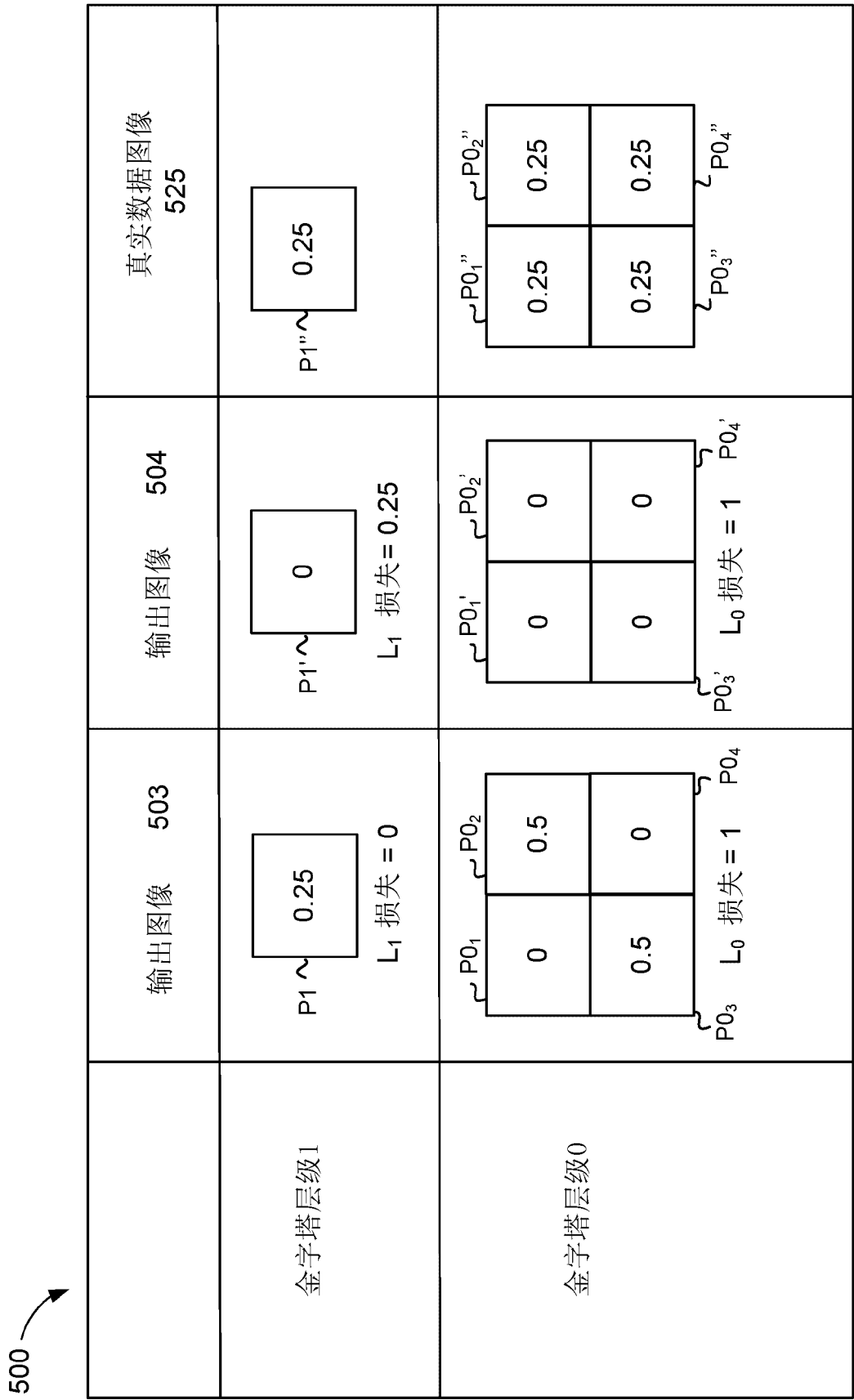


图 5

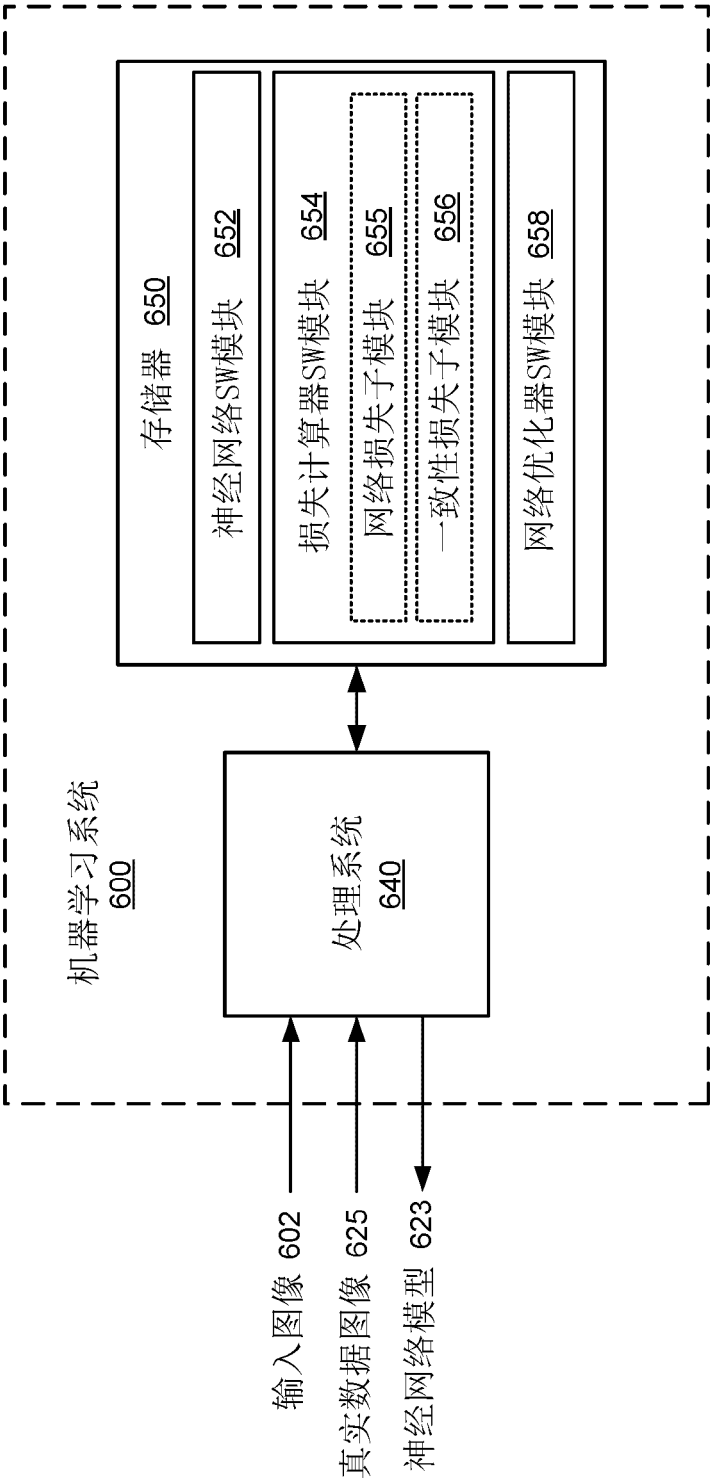


图 6

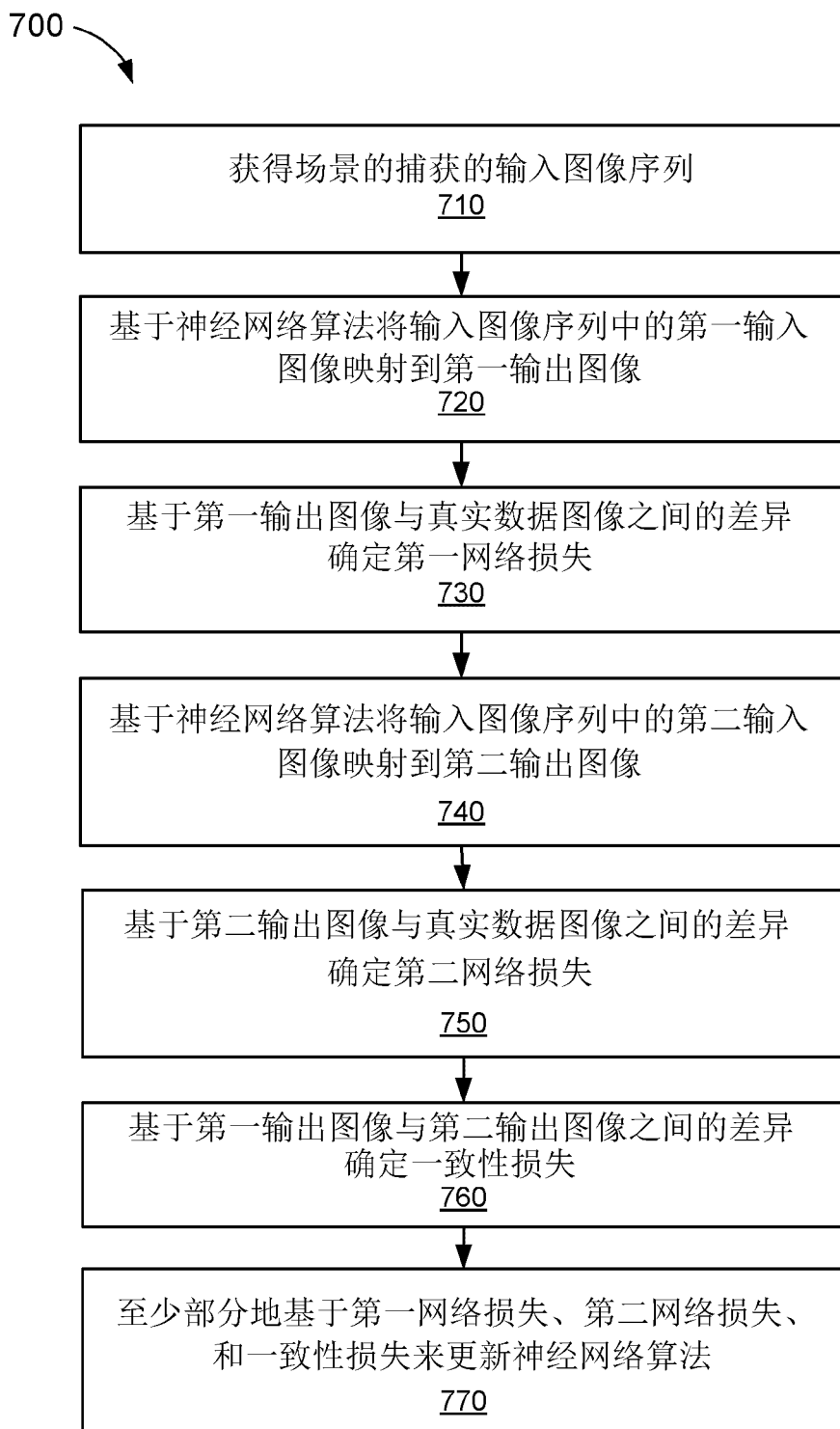


图 7

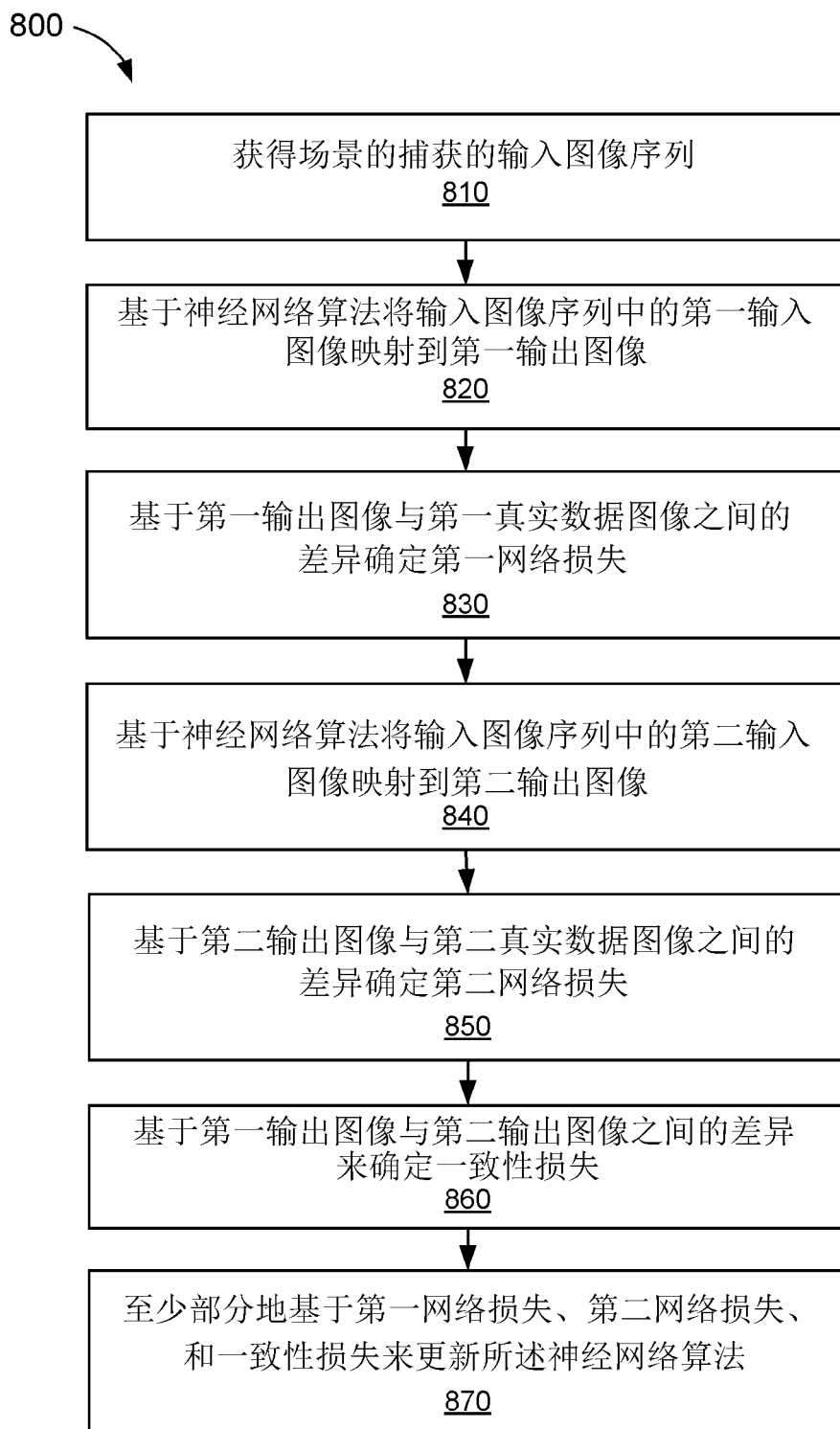


图 8