



República Federativa do Brasil
Ministério da Economia
Instituto Nacional da Propriedade Industrial

(11) BR 112015011892-5 B1



(22) Data do Depósito: 14/12/2013

(45) Data de Concessão: 11/01/2022

(54) Título: MÉTODO EM UM SISTEMA DE APRENDIZADO DE MÁQUINA E APARELHO DE APRENDIZADO DE MÁQUINA

(51) Int.Cl.: G06N 7/00.

(30) Prioridade Unionista: 14/02/2012 US 13/714,610.

(73) Titular(es): MICROSOFT TECHNOLOGY LICENSING, LLC.

(72) Inventor(es): REINHARD SEBASTIAN BERNHARD NOWOZIN; PO-LING LOH.

(86) Pedido PCT: PCT US2013075209 de 14/12/2013

(87) Publicação PCT: WO 2014/093949 de 19/06/2014

(85) Data do Início da Fase Nacional: 22/05/2015

(57) Resumo: ALOCAÇÃO DE RECURSOS PARA APRENDIZADO DE MÁQUINA. A presente invenção refere-se à alocação de recursos para aprendizado de máquina que está descrito tal como para selecionar entre muitas possíveis opções, por exemplo, como parte de um eficiente processo de treinamento para treinamento de árvore de decisão randômica, para selecionar qual de muitas famílias de modelos melhor descreve os dados, para selecionar qual de muitas características melhor classifica os itens. Em vários exemplos amostras de informações sobre opções incertas são utilizadas para pontuar as opções. Em vários exemplos, os intervalos de confiança são calculados para as pontuações e utilizados para selecionar uma ou mais das opções. Em exemplos, as pontuações das opções podem ser estatísticas de diferença limitadas as quais mudam pouco conforme qualquer amostra é omitida do cálculo da pontuação. Em um exemplo, o treinamento de árvore de decisão randômica é tornado mais eficiente enquanto retendo precisão para aplicações não limitadas a detecção de pose de corpo humano de imagens de profundidade.

Relatório Descritivo da Patente de Invenção para
**"MÉTODO EM UM SISTEMA DE APRENDIZADO DE MÁQUINA E
APARELHO DE APRENDIZADO DE MÁQUINA".**

ANTECEDENTES

[0001] Muitos sistemas de aprendizado de máquina dos dias atuais sofrem significativos desafios computacionais devido a massivas quantidades de dados. Os sistemas de aprendizado de máquina modernos frequentemente processam milhões de amostras adquiridas a bom preço, gerando uma significativa carga computacional. Para estes sistemas, fazer inferências tão eficientemente quanto possível com base em muito grandes quantidades de dados é um desafio contínuo. Por exemplo, os sistemas de aprendizado de máquina buscam acessar quantos dados precisam ser processados e em qual ordem antes de tomar uma decisão que estão corretos dentro de uma tolerância de erro trabalhável. Recursos de computação finitos estão tipicamente disponíveis e o sistema de aprendizado de máquina precisa alocar os recursos apropriadamente.

[0002] Frequentemente os sistemas de aprendizado de máquina estão preocupados com a alocação eficiente de recursos finitos onde existe incerteza sobre as opções. Por exemplo, encontrar qual de muitos possíveis anúncios mostrarem para um cliente aloca um recurso (recurso de computação, espaço de tela) para uma possível opção (um único anúncio). Mostrar um anúncio e observar a resposta do usuário (clicar / não clicar) revela informações parciais sobre a impressão de anúncio que pode ser utilizada para executar melhores alocações dos recursos no futuro.

[0003] As modalidades abaixo descritas não estão limitadas a implementações as quais resolvem qualquer ou todas as desvantagens dos sistemas de aprendizado de máquina conhecidos

SUMÁRIO

[0004] O seguinte apresenta um sumário simplificado da descrição de modo a prover uma compreensão básica para o leitor. Este sumário não é uma visão geral extensa da descrição e não identifica elementos chave / críticos ou delineiam o escopo da especificação. O seu único propósito é apresentar uma seleção de conceitos aqui descritos em uma forma simplificada como um prelúdio para a descrição mais detalhada que é posteriormente apresentada.

[0005] A alocação de recursos para aprendizado de máquina está descrito tal como selecionar entre muitas possíveis opções, por exemplo, como parte de um eficiente processo de treinamento para treinamento de árvore de decisão randômica, para selecionar qual de muitas famílias de modelos melhor descreve os dados, para selecionar qual de muitas características melhor classifica os itens. Em vários exemplos amostras de informações sobre opções incertas são utilizadas para pontuar as opções. Em vários exemplos, intervalos de confiança são calculados para as pontuações e utilizados para selecionar uma ou mais das opções. Em exemplos, as pontuações das opções podem ser estatísticas de diferença limitadas as quais mudam pouco conforme qualquer amostra é omitida do cálculo da pontuação. Em um exemplo, o treinamento de árvore de decisão randômica é tornado mais eficiente enquanto retendo precisão para aplicações não limitadas a detecção de pose de corpo humano de imagens de profundidade.

[0006] Muitas das características acompanhantes serão mais prontamente apreciadas conforme as mesmas tornam-se melhor compreendidas por referência à descrição detalhada seguinte considerada em conexão com os desenhos acompanhantes.

DESCRÍÇÃO DOS DESENHOS

[0007] A presente descrição será melhor compreendida da

descrição detalhada seguinte lida à luz dos desenhos acompanhantes, em que:

[0008] Fig. 1 é um diagrama esquemático de um seletor de opção o qual pode ser um componente de um sistema de aprendizado de máquina; e também de um seletor de modelo, um seletor de característica e um seletor de função de divisão de árvore, cada um dos quais pode ser um componente de um sistema de aprendizado de máquina;

[0009] Fig. 2 é um gráfico de ganho de informações contra o número de exemplos de treinamento;

[0010] Fig. 3 é um fluxograma de um método em um seletor de opção;

[0011] Fig. 4 é um fluxograma de um método da caixa 308 da Fig. 3 em mais detalhes;

[0012] Fig. 5 ilustra um exemplo de um sistema de aprendizado de máquina baseado em câmera para controlar um jogo de computador;

[0013] Fig. 6 é um diagrama esquemático de um dispositivo de captura de imagem para utilização com o sistema de aprendizado de máquina da Fig. 6;

[0014] Fig. 7 é um fluxograma de um método para predizer posições de junta em uma imagem de profundidade inserida;

[0015] Fig. 8 é um fluxograma de um método para treinar uma floresta de decisão randômica;

[0016] Fig. 9 é um diagrama esquemático de uma floresta de decisão randômica;

[0017] Fig. 10 é um fluxograma de um método de predição de posição junta no tempo de teste;

[0018] Fig. 11 ilustra um dispositivo baseado em computação exemplar no qual as modalidades de um sistema de aprendizado de máquina com um seletor de opção podem ser implementadas.

[0019] Números de referência iguais são utilizados para designar partes iguais nos desenhos acompanhantes.

DESCRIÇÃO DETALHADA

[0020] A descrição detalhada abaixo provida em conexão com os desenhos anexos está destinada como uma descrição dos presentes exemplos e não pretende representar as únicas formas nas quais o presente exemplo pode ser construído ou utilizado. A descrição apresenta as funções do exemplo e a sequência de etapas para construir e operar o exemplo. No entanto, as mesmas ou equivalentes funções e sequências podem ser executadas por diferentes exemplos.

[0021] Apesar dos presentes exemplos serem aqui descritos e ilustrados como sendo implementados em um sistema de aprendizado de máquina para controlar um jogo, o sistema descrito está provido como um exemplo e não uma limitação. Como aqueles versados na técnica apreciarão, os presentes exemplos são adequados para aplicação em uma variedade de diferentes tipos de sistemas de aprendizado de máquina.

[0022] A Fig. 1 é um diagrama esquemático de um seletor de opção 102 o que é um componente de um sistema de aprendizado de máquina; e também de um seletor de modelo 112, um seletor de característica 114 e um seletor de função de divisão de árvore 116, cada um dos quais pode ser um componente de um sistema de aprendizado de máquina.

[0023] O seletor de opção seleciona uma ou mais opções de uma pluralidade de opções 100 para avaliar pontuações das opções utilizando uma lógica de pontuação 106 e amostras 110 de informações sobre as opções. Existe uma incerteza sobre as opções já que as informações sobre as opções estão disponíveis de amostras 110 as quais tornam-se disponíveis em um processo em andamento. A lógica de pontuação utiliza uma lógica de corrida 108 para acelerar o

processo de seleção enquanto mantendo uma tolerância de erro especificada. Como um resultado da seleção de opção, recursos 104 podem ser alocados. Por exemplo, cada opção pode ser um anúncio e os recursos 104 podem ser recursos de computação e espaço de tela para apresentar uma impressão do anúncio para um usuário. Cada amostra 110 pode ser uma impressão de anúncio e uma resposta de usuário observada à impressão de anúncio (por exemplo, resposta de clicar / não clicar). O seletor de opção pode calcular uma pontuação para o anúncio utilizando a lógica de pontuação 106 e as amostras. Após muitos milhares ou mais de amostras terem sido observadas as pontuações para cada anúncio podem atingir um nível estável e estes níveis podem ser utilizados pelo seletor de opção para alocar os recursos 104 selecionando qual anúncio deve ser apresentado. De modo a acelerar o processo, uma lógica de corrida 108 pode ser utilizada para permitir que menos amostras sejam avaliadas antes da seleção ser feita, mas enquanto assegurando uma tolerância de erro especificada do resultado de seleção.

[0024] Cada um do seletor de opção 102, lógica de pontuação 106 e lógica de corrida 108 são implementados por computador utilizando software e/ou hardware. Por exemplo, estes componentes podem ser implementados no todo ou em parte utilizando componentes de lógica de hardware. Isto é, alternativamente, ou além disso, a funcionalidade aqui descrita pode ser executada, pelo menos em parte, por um ou mais componentes de lógica de hardware. Por exemplo, e sem limitação, os tipos ilustrativos de componentes de lógica de hardware que podem ser utilizados incluem Redes de Porta Programáveis no Campo (FPGAs), Circuitos Integrados Específicos de Programa (ASICs), Produtos padrão Específicos de Programa (ASSPs), sistemas de Sistema em um Chip (SOCs), Dispositivos Lógicos Programáveis Complexos (CPLDs), Unidades de Processamento de Gráficos

(GPUs).

[0025] Em um exemplo o seletor de opção é um seletor de modelo 112. Nesta situação as opções são diferentes famílias de modelo modelos 118 os quais podem ter diferentes complexidades. As amostras 110 nesta situação podem ser amostras de um sistema que é requerido modelar. Por exemplo, as amostras podem ser medições empíricas de temperatura, queda de chuva, e outros fatores onde o sistema a ser modelado é um sistema de clima. Em outro exemplo, as amostras podem ser elementos de imagem identificados de imagens de treinamento onde o sistema a ser modelado é um sistema de classificação de imagem. O seletor de modelo produz uma saída que compreende uma ou mais das famílias de modelos 118 para serem utilizadas para modelar os dados (amostras). Os recursos 104 podem ser recursos de computação utilizados pela lógica de pontuação e a lógica de corrida. Os recursos 104 podem ser recursos de computação utilizados processos de computação a jusante os quais utilizam a família de modelos selecionada pelo seletor de modelo. A lógica de pontuação e a lógica de corrida são utilizadas em um modo similar como acima descrito para permitir e acelerar o processo de seleção.

[0026] Em um exemplo o seletor de opção é um seletor de característica 120. Nesta situação as opções são características 120, por exemplo, de um corpo de documentos ou de imagens de objetos. As amostras 110 nesta situação podem ser exemplos de características juntamente com observações de quão bem estas características classificam os documentos ou objetos. O seletor de modelo produz uma saída compreender características selecionadas como sendo boas na classificação dos documentos ou objetos. Os recursos 104 podem ser recursos de computação utilizados pela lógica de pontuação e lógica de corrida. Os recursos 104 podem ser recursos de computação utilizados pelos processos de computação a jusante os

quais utilizam as características selecionadas pelo seletor de característica. A lógica de pontuação e a lógica de corrida são utilizadas em um modo similar como acima descrito para permitir e acelerar o processo de seleção.

[0027] Em um exemplo o seletor de opção é um seletor de função de divisão de árvore 116 o qual faz parte de um sistema de aprendizado de máquina para treinar árvores de decisão randômicas. Uma árvore de decisão randômica é um tipo de estrutura de dados utilizada para armazenar dados acumulados durante uma fase de treinamento de modo que estes possam ser utilizados para fazer previsões sobre exemplos previamente não vistos pela árvore de decisão randômica. A árvore de decisão randômica é usualmente utilizada como parte de um conjunto de árvores de decisão randômicas treinadas para um domínio de aplicação específico de modo a obter generalização (isto é, sendo capaz de fazer boas previsões sobre exemplos os quais são diferentes daqueles utilizados para treinar a floresta). Uma árvore de decisão randômica tem um nodo de raiz, uma pluralidade de nodos de divisão e uma pluralidade de nodos folha. Durante o treinamento a estrutura da árvore (o número de nodos e como estes estão conectados) é aprendida assim como as funções de divisão a serem utilizadas para cada um dos nodos de divisão. Além disso, os dados são acumulados nos nodos folha durante o treinamento.

[0028] As funções de divisão podem ser randomicamente geradas e podem compreender as opções 122 da Fig. 1. Os dados (tal como um elemento de imagem de uma imagem) podem ser empurrados através de árvores de uma floresta de decisão randômica da raiz até uma folha em um processo por meio de que uma decisão é feita em cada nodo de divisão. A decisão pode ser feita de acordo com as características do elemento de imagem e a função de divisão

associada com o nodo de divisão. O elemento de imagem prossegue para o próximo nível da árvore abaixo de um galho escolhido de acordo com os resultados da decisão.

[0029] O seletor de função de divisão de árvore 116 seleciona qual das funções de divisão utilizar em nodos de divisão específicos da árvore de decisão. De modo a fazer a seleção de amostras 110 são utilizadas, as quais são exemplos de dados de treinamento. Por exemplo, os exemplos de dados de treinamento podem ser elementos de imagem os quais estão identificados como pertencendo uma classe específica ou podem ser outros dados de treinamento identificados ou não identificados. A lógica de pontuação 106 calcula uma pontuação para as diferentes opções de função de divisão 122 utilizando as amostras 110. A lógica de corrida 108 pode ser utilizada para acelerar o processo de seleção como abaixo descrito em mais detalhes com referência às Figuras 3, 4, 7 e 8. Os recursos 104 podem ser recursos de computação utilizados pela lógica de pontuação e a lógica de corrida. Os recursos 104 podem ser recursos de computação utilizados pelos processos de computação a jusante os quais utilizam a função de divisão de árvore selecionada pelo seletor de modelo.

[0030] Nos exemplos descritos com referência à Fig. 1 a lógica de pontuação 106 é utilizada para avaliar uma pontuação para cada uma das opções 100. Por exemplo, a pontuação pode estar baseada em uma estatística que descreve o comportamento de uma opção, tal como uma média, mediana, modo ou outra estatística. Em vários exemplos a pontuação está baseada em uma estatística a qual é uma estatística de diferença limitada. Uma estatística de diferença limitada é um valor numérico o qual descreve o comportamento de uma entidade (tal como uma opção nos exemplos da Fig. 1) de uma pluralidade de observações (amostras) do comportamento; e a qual muda o seu valor número somente um pouco se somente uma da

pluralidade de observações for mudada. Em alguns exemplos, a mudança de uma observação pode ser conseguida omitindo a observação. Uma lista não exaustiva de exemplos de estatísticas de diferença limitada é: variância, entropia, ganho de informações, índice Gini.

[0031] A Fig. 2 é um gráfico de ganho de informações em relação ao número de exemplos de treinamento. O ganho de informações é um exemplo de uma estatística de diferença limitada a qual pode ser medida pela lógica de pontuação conforme os exemplos de treinamento são recebidos. No exemplo da Fig. 2 existem duas opções as quais são característica A e característica B. Apesar de que na prática muito mais opções podem estar presentes (duas estão mostradas para clareza). As pontuações de ganho de informações para a característica A são registradas como uma linha sólida e as pontuações de ganho de informações para a característica B são registradas como uma linha pontilhada. Suponha que o seletor de opção seja o seletor de característica 114 da Fig. 1 e precisa encontrar a característica com o mais alto ganho de informações. Como as amostras são inicialmente recebidas a pontuação de ganho de informações é ruidosa porque somente poucas amostras estão disponíveis. Uma vez que mais de 500 amostras foram recebidas as pontuações de ganho de informações estabilizaram e estão claramente separadas para as características A e B com a característica B tendo uma mais alta pontuação de ganho de informações. O seletor de característica 114 está projetado para tomar uma decisão quanto a se a característica A ou a característica B tem a mais alta pontuação de ganho de informações em um ponto ótimo. Isto é, após apenas exemplos de treinamento suficientes foram recebidos para dar uma resposta a qual é precisa dentro de uma tolerância de erro especificada. Isto permite que recursos computacionais sejam

alocados apropriadamente tanto para o próprio processo de seleção de característica quanto aos processos a jusante os quais utilizam os resultados de seleção de característica.

[0032] O seletor de característica 114 pode ter uma lógica de corrida 108 a qual calcula barras de erro também conhecidas como intervalos de confiança para as pontuações de ganho de informações (ou pontuações de outra estatística de diferença limitada). Um intervalo de confiança é uma faixa de valores numéricos que representa possíveis valores de uma pontuação com uma precisão especificada. Na Fig. 2 um intervalo de confiança 200 está mostrado para a pontuação de ganho de informações da característica A quando o número de exemplos de treinamento está logo abaixo de 500. Quando o número de exemplos de treinamento está acima de 500 o intervalo de confiança 206 para a pontuação da característica A tem uma menor faixa porque existe mais certeza sobre o ganho de informações. No mesmo modo um intervalo de confiança 202 para a pontuação da característica B é grande quando o número de exemplos de treinamento está ao redor 500 e é um menor intervalo de confiança 204 quando o número de exemplos de treinamento aumenta. A lógica de corrida pode utilizar os intervalos de confiança para decidir se exemplos de treinamento suficientes foram recebidos para dar uma seleção de opção a qual é precisa dentro de uma tolerância de erro especificada. Por exemplo, as opções onde o intervalo de confiança não está sobrepondo com o intervalo de confiança para a mais alta opção de pontuação podem ser rejeitadas. Gradualmente o número de opções reduz conforme cada vez mais opções são rejeitadas da corrida conforme o número de amostras aumenta. Isto está agora explicado com referência à Fig. 3 a qual é um fluxograma de um método em um seletor de opção o qual pode ser qualquer um do seletor de opção 102, seletor de modelo 112, seletor de característica

114 ou seletor de função de divisão de árvore 116 da Fig. 1.

[0033] O seletor de opção ajusta 300 uma tolerância de erro a qual deve ser utilizada pela lógica de corrida quando este decide ser exemplos de treinamento suficientes foram recebidos para dar uma seleção de opção a qual é preciso dentro da tolerância de erro ajustada. A tolerância de erro pode ser ajustada de acordo com uma entrada do usuário, pode ser pré-configurada, ou pode ser automaticamente selecionada de acordo com o domínio de aplicação, os recursos computacionais disponíveis, e outros fatores.

[0034] O seletor de opção acessa 302 uma lista ou outras informações que identificam as opções potências e recebe 304 um ou mais exemplos de treinamento para cada uma das opções. Por exemplo, as opções podem ser funções de divisão de árvore de decisão randomicamente geradas de um processo de treinamento de árvore de decisão. Os exemplos de treinamento podem ser acessados de um armazenamento de dados, fluxo de dados ou outra fonte. O seletor de opção utiliza a lógica de pontuação 106 para calcular 306 uma pontuação para cada uma das opções utilizando os exemplos de treinamento recebidos. Por exemplo, a pontuação é uma estatística de diferença limitada como acima descrito. A lógica de corrida 108 é utilizada para calcular 308 um intervalo de confiança para a pontuação para cada opção. A lógica de corrida 108 identifica estas opções as quais atendem as condições de intervalo de confiança. Por exemplo, as opções cujos intervalos de confiança não sobrepõem com o intervalo de confiança da mais alta opção de pontuação são identificadas. As opções identificadas são removidas 314 da lista de opções potencias na etapa 302 e o processo pode repetir até que somente uma opção permaneça 312 ou somente um número especificado de opções permaneça.

[0035] A Fig. 4 é um fluxograma de parte do método da Fig. 3 em

mais detalhes. A lógica de corrida executa o método da Fig. 4 durante o processo de calcular os intervalos de confiança das pontuações para cada opção (caixa 308 da Fig. 3). A lógica de corrida estima 400 uma variância das pontuações para cada opção que foi calculada até agora. Uma variância é um valor numérico que descreve quanto uma quantidade varia. Qualquer método de estimar a variância pode ser utilizado. Em alguns exemplos a estimativa de variância Jack-Knife é utilizada como abaixo descrito em mais detalhes. Uma estimativa de variância Jack-Knife é feita recalculando a pontuação uma pluralidade de vezes removendo as diferentes das amostras de cada vez. As pontuações recalculadas são então utilizadas para estimar a variância avaliando quanto estas variam umas das outras.

[0036] A lógica de corrida pode calcular 402 um intervalo de confiança para uma das opções como a pontuação para esta opção, mais ou menos uma constante vezes a raiz quadrada dos tempos de variância estimados, um logaritmo de dois divididos pela tolerância de erro. A constante pode tomar um valor $5/2$ ou um valor similar.

[0037] A variância estimada pode ser uma estimativa de variância Jack-Knife como acima descrito e pode ser calculada em qualquer modo adequado. Em exemplos onde as opções são candidatas de função de divisão em uma árvore de decisão randômica que executa a classificação, e a pontuação é um ganho de informações ou um ganho de Gini, a estimativa de variância Jack-Knife pode ser calculada como posteriormente descrito em mais detalhes neste documento de modo a reduzir o tempo de computação enquanto mantendo a precisão.

[0038] Em alguns exemplos a lógica de corrida pode ajustar o intervalo de confiança calculado para tendência onde é conhecido que a lógica de pontuação calcula a pontuação utilizando um estimador plug-in o qual introduz a tendência. No entanto, ajustar a tendência é uma etapa opcional. Bons resultados de trabalho podem ser

conseguidos quando a etapa de ajuste de tendência é omitida.

[0039] Mais formalmente, a lógica de corrida pode calcular um intervalo de confiança para uma das opções como segue:

$$Z - \frac{5}{2} \sqrt{\left(V_n^J + o\left(E(V_n^J)\right) \right) \log\left(\frac{2}{\delta}\right)}$$

$$Z + \left(\frac{5}{2} \sqrt{\left(V_n^J + o\left(E(V_n^J)\right) \right) \log\left(\frac{2}{\delta}\right)} + bias \right)$$

[0040] O qual pode ser expresso em palavras como uma faixa da pontuação Z da opção menos 5/2 vezes a raiz quadrada de uma estimativa de variância Jack-Knife da variância do número de

amostras n mais uma correção $o\left(E(V_n^J)\right)$ vezes um logaritmo de 2 dividido pela tolerância de erro δ para a pontuação Z da opção mais 5/2 a raiz quadrada de uma estimativa de variância Jack-Knife da

variância do número de amostras n mais uma correção $o\left(E(V_n^J)\right)$ vezes um logaritmo de 2 dividido pela tolerância de erro δ mais um termo de tendência.

[0041] Na prática foi descoberto que a convergência de uma estimativa de Jack-Knife da variância é extremamente rápida e os

termos de correção $o(E(V_n^J))$ podem ser omitidos da lógica de corrida.

[0042] A lógica de corrida pode ser utilizada para calcular os intervalos de confiança desta forma para pontuações as quais estão baseadas em estatística de diferença limitadas tal como, por exemplo, onde as pontuações são calculadas utilizando um histograma de contagens de bin. Isto permite que a lógica de corrida forneça resultados de trabalho para muitas formas de pontuações e permite que uma faixa muito mais ampla de aplicações de aprendizado de

máquina utilize a lógica de corrida do que anteriormente possível. Aumentando ou diminuindo o tamanho da tolerância de erro é possível negociar um nível de precisão garantido do resultado para o tempo de computação.

[0043] A lógica de corrida pode ser disposta para aplicar uma correção de tendência no intervalo de confiança como acima descrito, por exemplo, onde a pontuação é calculada pela lógica de pontuação utilizando um estimador plug-in tal como um estimador de entropia, um estimador de ganho de informações, um estimador de índice.

[0044] No caso que a pontuação é relativa ao ganho de informações, foi descoberto que a tendência pode ser dada por:

$$\frac{2(K-1)}{n}$$

[0045] A qual pode ser expressa em palavras como a razão de: duas vezes, o número de bins de um histograma utilizado para calcular o ganho de informações menos 1, para o número de amostras.

[0046] No caso que a pontuação está relacionada um índice Gini, foi descoberto que a tendência pode ser dada por

$$\frac{2(K-1)}{nK}$$

[0047] A qual pode ser expressa em palavras como a razão de: duas vezes, o número de bins de um histograma utilizado para calcular o índice Gini menos 1, para o número de amostras vezes o número de bins de histograma.

[0048] No caso que a pontuação é um ganho de variância, foi descoberto que nenhum termo de tendência é necessário.

[0049] Como acima mencionado uma estimativa de variância Jack-Knife é feita recalculando a pontuação uma pluralidade de vezes removendo as diferentes das amostras de cada vez (referido como uma estimativa de saída). As pontuações recalculadas são então utilizadas para estimar a variância avaliando quanto estas variam

umas das outras.

[0050] No caso em que a pontuação é um ganho de informações a estimativa de variância Jack-Knife da pontuação pode ser calculada pela lógica de corrida como:

$$\hat{V}_n^J = \frac{n-1}{n} \sum_{b \in \{L, R\}} \sum_{j=1}^K h_j^b (\hat{I}^{(b,j)} - \bar{I})^2$$

[0051] Onde o somatório é sobre termos com $h_j^b > 0$. A variância pode ser computada utilizando atualizações online ponderadas. h_j^b é a contagem de histograma na ramificação b e bin j. Este é um número o qual é um inteiro e maior do que ou igual a zero e menor do que ou igual a n, o número total de amostras.

[0052] A igualdade acima pode ser expressa em palavras como, uma estimativa Jack-Knife da variância de uma estimativa de plug-in de ganho de informações para n amostras de treinamento as quais alcançam um nodo de divisão é igual ao número de amostras de menos um, dividido pelo número de amostras de treinamento, vezes a soma sobre os nodos filhos esquerdo e direito do nodo de divisão, e a soma sobre todos os bins de histograma da contagem de histograma no bin j e ramificação b vezes o quadrado da diferença entre a estimativa de plug-in do ganho de informações quando uma amostra é removida da ramificação b e bin j menos a média empírica do ganho de informações. Onde a média empírica do ganho de informações é representada pelo símbolo \bar{I} e é dada por:

$$\bar{I} = \frac{1}{2K} \sum_{b \in \{L, R\}} \sum_{j=1}^K \hat{I}^{(b,j)}$$

[0053] A qual pode ser expressa em palavras como uma média empírica do ganho de informações é igual a um dividido por, duas vezes o número de bins de histograma, vezes a soma sobre os nodos filhos esquerdo e direito da soma sobre todos os bins de histograma

de uma estimativa de plug-in de ganho de informações com uma amostra na ramificação b e bin j removido.

[0054] No caso em que a pontuação é um ganho de Gini a estimativa de variância Jack-Knife da pontuação pode ser calculada pela lógica de corrida primeiro computando uma média empírica do ganho de Gini como segue:

$$\bar{G} = \frac{1}{2K} \sum_{b \in \{L, R\}} \sum_{j=1}^K \hat{G}^{(b,j)}$$

[0055] A qual pode ser expressa em palavras como uma média empírica do ganho de Gini é igual a um dividido por, duas vezes o número de bins de histograma, vezes a soma sobre os nodos filhos esquerdo e direito da soma sobre todos os bins de histograma de uma estimativa de plug-in de ganho de Gini.

[0056] Uma vez que a média empírica é computada a lógica de corrida é capaz de computar:

$$\hat{V}_n^J = \frac{n-1}{n} \sum_{b \in \{L, R\}} \sum_{j=1}^K h_j^b (\hat{G}^{(b,j)} - \bar{G})^2$$

[0057] Onde o somatório é sobre termos com $h_j^b > 0$.

[0058] A igualdade acima pode ser expressa em palavras como, uma estimativa de Jack-Knife de variância de uma estimativa de plug-in de ganho de Gini para n amostras de treinamento as quais alcançam um nodo de divisão igual ao número de amostras de treinamento menos um, dividido pelo número de amostras de treinamento, vezes a soma sobre os nodos filhos esquerdo e direito do nodo de divisão, da soma sobre todos os bins de histograma da contagem de histograma na ramificação b e bin j vezes o quadrado da diferença entre a estimativa de plug-in do ganho de Gini quando removendo uma única amostra na ramificação b e bin j menos uma média empírica dos ganhos de Gini obtidos removendo amostras individuais.

[0059] Um exemplo será agora descrito no qual o seletor de opção é utilizado como um seletor de função de divisão de árvore decisão para treinamento de árvore de decisão em um sistema de aprendizado de máquina para controlar um jogo. A árvore de decisão pode ser treinada utilizando o seletor de opção para ou uma tarefa de regressão ou a classificação dependendo da tarefa de aprendizado de máquina.

[0060] A Fig. 5 mostra um exemplo de sistema de controle baseado em câmera 500 para controlar um jogo de computador. A Fig. 5 mostra um usuário 502 jogando, neste exemplo ilustrativo, um jogo de boxe. Em alguns exemplos, o sistema de controle baseado em câmera 500 pode ser utilizado para, entre outras coisas, determinar a pose do corpo, conectar, reconhecer, analisar, rastrear, associar a alvo humano, prover retorno, interpretar gestos, e/ou adaptar a aspectos de um alvo humano tal usuário 502.

[0061] O sistema de controle baseado em câmera 500 comprehende um dispositivo de computação 504. O dispositivo de computação 504 pode ser um computador de uso geral, um sistema ou console de jogos, ou um dispositivo de processamento de imagem dedicado. O dispositivo de computação 504 pode incluir componentes de hardware e/ou componentes de software de modo que o dispositivo de computação 504 pode ser utilizado para executar aplicações tais como aplicações de jogos e/ou aplicações não de jogos. A estrutura do dispositivo de computação 504 está daqui em diante discutida com referência à Fig. 11.

[0062] O sistema de controle baseado em câmera 100 ainda comprehende um dispositivo de captura 506. O dispositivo de captura 506 pode ser, por exemplo, um sensor ou detector de imagem que pode ser utilizado para visualmente monitorar um ou mais usuários (tal como o usuário 502) de modo que os gestos executados pelos um ou mais usuários podem ser capturados, analisados, processados, e

rastreados para executar um ou mais controles ou ações dentro de um jogo ou aplicação, como abaixo descrito em mais detalhes.

[0063] O sistema de controle baseado em câmera 500 pode ainda compreender um dispositivo de display 508 conectado no dispositivo de computação 504. O dispositivo de computação pode ser uma televisão, um monitor, uma televisão de alta definição (HDTV), ou similares que podem prover visuais de jogo ou aplicação (e opcionalmente áudio) para o usuário 502.

[0064] Em operação, o usuário 502 pode ser rastreado utilizando o dispositivo de captura 506 de modo que as posições de junta, movimentos e tamanho do usuário 502 possam ser interpretados pelo dispositivo de computação 504 (e/ou o dispositivo de captura 506) como controles que podem ser utilizados para afetar a aplicação que está sendo executada pelo dispositivo de computação 504. Como um resultado, o usuário 502 pode mover o seu corpo para controlar um jogo ou aplicação executado.

[0065] No exemplo ilustrativo da Fig. 5, a aplicação que executa no dispositivo de computação 504 é um jogo de boxe que o usuário 502 está jogando. Neste exemplo, o dispositivo de computação 504 controla o dispositivo de display 508 para prover uma representação visual de um oponente de boxe para o usuário 502. O dispositivo de computação 504 também controla o dispositivo de display 508 para prover uma representação visual de um avatar de usuário que o usuário 502 pode controlar com os seus movimentos. Por exemplo, o usuário 502 pode dar um soco no espaço físico para fazer com que o avatar de usuário dê um soco no espaço de jogo. Assim, de acordo com este exemplo, o dispositivo de computação 504 e o dispositivo de captura 506 do sistema de controle baseado em câmera 500 podem ser utilizados para reconhecer e analisar o soco do usuário 502 no espaço físico de modo que o soco possa ser interpretado como um

controle de jogo do avatar de usuário no espaço de jogo.

[0066] Mais ainda, alguns movimentos podem ser interpretados como controles que correspondem a ações outras que controlar o avatar. Por exemplo, o usuário pode utilizar movimentos para entrar, sair, ligar ou desligar o sistema, pausar, salvar um jogo, selecionar um nível, perfil, ou menu, ver pontuações altas, comunicar com um amigo, etc. Além disso, o movimento do usuário 502 pode ser utilizado e analisado em qualquer modo adequado para interagir com aplicações outras que jogos, por exemplo inserir texto, selecionar ícones ou itens de menu, controlar reprodução de mídia, navegar websites ou operar qualquer outro aspecto controlável de um sistema de operação ou aplicação.

[0067] Referência é agora feita à Fig. 6, a qual ilustra um diagrama esquemático do dispositivo de captura 506 que pode ser utilizado no sistema de controle baseado em câmera 500 da Fig. 5. No exemplo da Fig. 6 o dispositivo de captura 506 está configurado para capturar imagens de vídeo com informações de profundidade. Tal dispositivo de captura pode ser referido com uma câmera de profundidade. As informações de profundidade podem estar na forma de uma imagem de profundidade que inclui valores de profundidade, isto é, um valor associado com cada elemento de imagem da imagem de profundidade que está relacionado com a distância entre a câmera de profundidade e um item ou objeto colocado no elemento de imagem.

[0068] As informações de profundidade podem ser obtidas utilizando qualquer técnica adequada incluindo, por exemplo, tempo de voo, luz estruturada, imagem estéreo, ou similares. Em alguns exemplos, o dispositivo de captura 506 pode organizar as informações de profundidade em "camadas Z," ou camadas que podem ser perpendicular ao eixo geométrico Z que estende da câmera de profundidade ao longo de sua linha de visão.

[0069] Como mostrado na Fig. 6, o dispositivo de captura 506 compreende pelo menos um sensor de formação de imagem 600. No exemplo mostrado na Fig. 6, o sensor de formação de imagem 600 compreende uma câmera de profundidade 602 disposta para capturar da imagem de profundidade de uma cena. A imagem de profundidade capturada pode incluir uma área bidimensional (2-D) da cena capturada onde cada elemento de imagem na área 2-D representa um valor de profundidade tal como um comprimento ou distância de um objeto na cena capturada da câmera de profundidade 602.

[0070] O dispositivo de captura pode também incluir um emissor 604 disposto para iluminar a cena de tal modo que as informações de profundidade possam ser verificadas pela câmera de profundidade 602. Por exemplo, no caso em que a câmera de profundidade 602 é uma câmera de tempo de voo de infravermelho (IR), o emissor 604 emite luz IR por sobre a cena, e a câmera de profundidade 602 está disposta para detectar a luz retrodifundida da superfície de um ou mais alvos e objetos na cena. Em alguns exemplos, uma luz infravermelha pulsada pode ser emitida do emissor 604 de modo que o tempo entre o pulso de luz que sai e um pulso de luz que chega correspondente possa ser detectado pela câmera de profundidade e medido e utilizado para determinar uma distância física do dispositivo de captura 506 para uma posição sobre os alvos ou objetos em cena. Além disso, em alguns exemplos, a fase da onda de luz que sai do emissor 604 pode ser comparada com a fase da onda de luz que chega na câmera de profundidade 602 para determinar um deslocamento de fase. O deslocamento de fase pode então ser utilizado para determinar uma distância física do dispositivo de captura 506 para uma posição sobre os alvos ou objetos. Em um exemplo adicional, a análise de tempo de voo pode ser utilizada para indiretamente determinar uma distância física do dispositivo de captura 506 para uma posição sobre os alvos

ou objetos analisando a intensidade do feixe de luz refletido ao longo do tempo através de várias técnicas incluindo, por exemplo, formação de imagem de pulso de luz com obturador.

[0071] Em outro exemplo, o dispositivo de captura 506 pode utilizar uma luz estruturada para capturar as informações de profundidade. Em tal técnica, uma luz padronizada (por exemplo, luz exibida como um padrão conhecido tal como um padrão de grade ou padrão de tiras) pode ser projetada por sobre a cena utilizando o emissor 604. Quando atingindo a superfície de um ou mais alvos ou objetos na cena, o padrão torna-se deformado. Tal deformação do padrão pode ser capturada pela câmera de profundidade 602 e então ser analisada para determinar uma distância física do dispositivo de captura 506 para uma posição sobre os alvos ou objetos na cena.

[0072] Em outro exemplo, a câmera de profundidade 602 pode estar na forma de duas ou mais câmeras fisicamente separadas que veem uma cena de diferentes ângulos, de modo que das estéreo visuais são obtidos que podem ser resolvidos para gerar as informações de profundidade. Neste caso o emissor 604 pode ser utilizado para iluminar a cena ou pode ser omitido.

[0073] Em alguns exemplos, além da câmera de profundidade 602, o dispositivo de captura 506 pode compreender uma câmera de vídeo regular, a qual é referida como uma câmera RGB 606. A câmera RGB 606 para capturar sequências de imagens da cena em frequências de luz visível, e pode com isto prover imagens que podem ser utilizadas para aumentar as imagens de profundidade. Em exemplos alternativos, a câmera RGB 606 pode ser utilizada ao invés da câmera de profundidade 602.

[0074] O dispositivo de captura 506 mostrado na Fig. 6 ainda compreende pelo menos um processador 608, o qual está em comunicação com o sensor de formação de imagem 600 (isto é,

câmera de profundidade 602 e câmera RGB 606 no exemplo da Fig. 6) e o emissor 604. O processador 608 pode ser um microprocessador de uso geral, ou um processador de sinal / imagem especializado. O processador 608 está disposto para executar instruções e controlar o sensor de formação de imagem 600 e o emissor 604 para capturar imagens de profundidade e/ou imagens RGB. O processador 608 pode também opcionalmente estar disposto para executar um processamento sobre estas imagens, como daqui em diante delineado em mais detalhes.

[0075] Em alguns exemplos o sensor de formação de imagem é utilizado para prover imagens de silhueta as quais são imagens binárias bidimensionais que identificam as regiões de frente e fundo da profundidade e/ou imagens RGB capturadas pelo sensor de formação de imagem. As imagens de silhueta podem ser formadas no sensor de formação de imagem e/ ou processador 608 da profundidade capturada e imagens RGB. As imagens de silhueta podem ser processadas utilizando os métodos aqui descritos para predizer posições de junta bidimensionais. Neste caso as imagens de silhueta podem ser imaginadas como imagens de profundidade achatadas a uma profundidade fixa. As imagens de profundidade capturadas podem ser utilizadas para predizer posições de junta tridimensionais como abaixo descrito em mais detalhes.

[0076] O dispositivo de captura 506 mostrado na Fig. 6 ainda inclui uma memória 610 dispostas para armazenar as instruções para execução pelo processador 608, imagens ou quadros de imagens capturados pela câmera de profundidade 602 ou câmera RGB 606, ou quaisquer outras informações, imagens, ou similares. Em alguns exemplos, a memória 610 pode incluir uma memória de acesso randômico (RAM), uma memória somente de leitura (ROM), um cache, uma memória instantânea, um disco rígido, ou qualquer outro

componente de armazenamento adequado. A memória 610 pode ser um componente separado em comunicação com o processador 608 ou integrada no processador 608.

[0077] O dispositivo de captura 506 também compreende uma interface de saída 612 em comunicação com o processador 608 e está disposta para prover dados para o dispositivo de computação 604 através de uma conexão de comunicação. A conexão de comunicação pode ser, por exemplo, uma conexão com fio (tal como USB, Firewire, Ethernet ou similar) e/ou uma conexão sem fio (tal como WiFi, Bluetooth ou similar). Em outros exemplos, a interface de saída 612 pode interfacear com uma ou mais redes de comunicação (tal como a internet) e prover dados para o dispositivo de computação 504 através destas redes.

[0078] A Fig. 7 é um fluxograma de predição de posições de junta em uma imagem de profundidade inserida 708 que apresenta um ou mais humanos ou animais ou partes de humanos ou animais. Uma pluralidade de imagens de treinamento 700 as quais podem ser imagens de profundidade que especificaram as posições de junta está disponível. As imagens podem ser bidimensionais, tridimensionais ou imagens dimensionais mais altas ou sequências de tais imagens. Os exemplos aqui descritos utilizam imagens de profundidade, mas são também aplicáveis a imagens de coloridas, imagens médicas ou outros tipos de imagem.

[0079] Uma floresta de decisão randômica é treinada 702 para permitir que os elementos de imagem das imagens de treinamento 700 voltem para as posições de uma ou mais junta apresentadas nestas imagens. O processo de treinamento pode compreender utilizar um seletor de função de divisão de árvore como acima descrito com referência à Fig. 1 de modo a acelerar o treinamento enquanto mantendo a precisão. O resultado do processo de treinamento é uma

floresta de decisão randômica 704 a qual foi treinada utilizando um seletor de função de divisão de árvore.

[0080] Elementos de imagem podem ser pixels, grupos de pixels, voxels, grupos de voxels, blobs, remendos ou outros componentes de uma imagem. Uma floresta de decisão randômica compreende uma ou mais árvores de decisão cada um tendo um nodo de raiz, uma pluralidade de nodos de divisão e uma pluralidade de nodos folha. Os elementos de imagem de uma imagem podem ser empurrados através de árvores de uma floresta de decisão randômica da raiz até um nodo folha em um processo por meio de que uma decisão é feita em cada nodo de divisão. A decisão de acordo com as características do elemento de imagem e características elementos de imagem de teste deslocados destes por deslocamentos espaciais especificados pelos parâmetros no nodo de divisão. O processo de decisão pode ser expresso utilizando uma função de divisão. Em um nodo de divisão o elemento de imagem prossegue para o próximo nível das árvores por galho abaixo escolhido de acordo com os resultados da decisão. A floresta de decisão randômica pode utilizar regressão ou classificação como abaixo descrito em mais detalhes. Durante o treinamento, valores de parâmetros (também referidos como características) são aprendidos para utilização nos nodos de divisão e os dados são acumulados nos nodos folha. O processo de aprender os parâmetros compreende selecionar uma função de divisão para utilização no nodo de divisão. Um seletor de opção pode ser utilizado para este processo de seleção como descrito com referência à Fig. 1 e utilizando os processos das Figuras 3 e 4.

[0081] Os votos de posição de junta podem ser acumulados nos nodos folha durante o treinamento. Um voto de posição de junta é uma posição de imagem (ou região) onde uma junta é predita estar relativa a um elemento de imagem que faz o voto. Um voto de posição de

junta pode ser especificado em qualquer modo adequado. Por exemplo, como um vetor que expressa a distância relativa e a direção de uma posição de junta predita de um elemento de imagem que faz o voto. Não é essencial utilizar um vetor, outros formatos podem ser utilizados.

[0082] Armazenar todos os votos de posição de junta nos nodos folha durante o treinamento pode ser muito intensivo de memória já que grandes quantidades de dados de treinamento são tipicamente utilizadas para aplicações práticas. Em algumas modalidades os votos são agregados de modo que estes possam ser armazenados em um modo compacto. Vários diferentes processos de agregação podem ser utilizados como abaixo descrito em mais detalhes.

[0083] No tempo de teste uma imagem de profundidade 708 anteriormente não vista é inserida no sistema para ter posições de junta preditas. Esta é aplicada na floresta de decisão randômica treinada 706 para obter posições de junta preditas 701. Cada elemento de imagem da imagem de profundidade inserida 708 pode ser enviado através de cada árvore da floresta de decisão randômica treinada e votos de posição de junta obtidos das folhas. Deste modo os votos podem ser feitos comparando cada elemento de imagem com elementos de imagem de teste deslocados destes por deslocamentos espaciais aprendidos. Cada elemento de imagem pode fazer uma pluralidade de votos para cada junta. Estes votos podem ser agregados de acordo com vários diferentes métodos de agregação para fornecer as posições de junta preditas 710. O processo de tempo de teste é, portanto um processo de estágio único de aplicar a imagem de profundidade de inserida na floresta de decisão randômica treinada para diretamente obter as posições de junta preditas. O processo de estágio único não envolve resultados intermediários tal como as partes do corpo que está sendo preditas. Este processo de estágio único

pode ser executado em um modo rápido e efetivo para fornecer resultados em tempo real e com resultados de alta qualidade.

[0084] A Fig. 8 é um fluxograma de um processo para treinar uma floresta de decisão para identificar posições de junta em uma imagem. Isto também pode ser imaginado como gerar votos de posição de junta de elementos de imagem de uma imagem. A floresta de decisão é treinada utilizando um conjunto de imagens de treinamento. O conjunto de imagens de treinamento compreende uma pluralidade de imagens cada uma mostrando pelo menos um humano ou animal ou parte de um corpo de humano ou animal. Os corpos ou partes de corpo apresentados nas imagens de treinamento estão em várias diferentes poses (tal como sentados ou em pé). As posições de junta nas imagens de treinamento estão especificadas. Portanto, o conjunto de treinamento forma um banco de dados preciso.

[0085] Em um exemplo, ao invés de capturar as imagens de profundidade de muitos diferentes exemplos de poses de corpo, o conjunto de treinamento pode compreender imagens geradas por computador sintéticas. Tais imagens sintéticas modelam realisticamente o corpo humano ou de animal (ou parte do corpo humano ou de animal) em diferentes poses interagindo com diferentes objetos, e podem ser geradas para serem vistas de qualquer ângulo ou posição. Estas podem ser produzidas muito mais rapidamente do que as imagens de profundidade reais, e podem prover uma variedade mais ampla de imagens de treinamento.

[0086] Referindo à Fig. 8, para treinar as árvores de decisão, o conjunto de treinamento acima descrito é primeiro recebido 800. O número de árvores de decisão a serem utilizadas em uma floresta de decisão randômica é selecionado 802. Uma floresta de decisão randômica é uma coleção de árvores de decisão determinísticas. As árvores de decisão podem ser utilizadas em algoritmos de

classificação ou regressão, mas podem sofrer de excesso de ajuste, isto é, má generalização. No entanto, um conjunto de muitas árvores de decisão randomicamente treinadas (uma floresta randômica) gera uma generalização aperfeiçoada. Durante o processo de treinamento, o número de árvores é fixo.

[0087] Um exemplo de floresta de decisão randômica está mostrado ilustrado na Fig. 9. A floresta de decisão ilustrativa da Fig. 9 compreende três árvores de decisão: uma primeira árvore 900; uma segunda árvore 902; e uma terceira árvore 904. Cada árvore de decisão compreende um nodo de raiz (por exemplo, nodo de raiz 906 da primeira árvore de decisão 900), uma pluralidade de nodos internos, denominados nodos de divisão (por exemplo, o nodo de divisão 908 da primeira árvore de decisão 900), e uma pluralidade de nodos folha (por exemplo, o nodo folha 910 da primeira árvore de decisão 900).

[0088] Em operação, cada nodo de raiz e divisão de cada árvore executa um teste binário sobre os dados inseridos e com base no resultado direciona os dados para o nodo filho esquerdo ou direito. Os nodos folha não executam nenhuma ação; estes armazenam os votos de posição de junta acumulados (e opcionalmente outras informações). Por exemplo, as distribuições de probabilidade podem ser armazenadas representando os votos de posição de junta acumulados. Em alguns exemplos os nodos folha opcionalmente armazenam as informações tal como as distribuições de probabilidade sobre partes do corpo. Neste caso a floresta é capaz de atuar tanto como uma floresta de regressão para predizer posições de junta quanto como uma floresta de classificação para predizer as partes de corpo de superfície.

[0089] O modo no qual os parâmetros utilizados para cada nodos de divisão são escolhidos e como as probabilidades de nodo folha

podem ser computadas será agora descrito. Uma árvore de decisão da floresta de decisão é selecionada 804 (por exemplo, a primeira árvore de decisão 800) e o nodo de raiz 806 é selecionado 806. Pelo menos um subconjunto dos elementos de imagem de cada uma das imagens de treinamento é então selecionado 808. Por exemplo, a imagem pode ser segmentada de modo que os elementos de imagem nas regiões de primeiro plano são selecionados. Cada elemento de imagem selecionado de cada imagem de treinamento está associado com uma pluralidade de posições de junta conhecidas. Por exemplo, um elemento de imagem no torso pode estar associado com uma posição de junta conhecida para o quadril esquerdo e uma posição de junta conhecida para o quadril direito. Em outro exemplo, um elemento de imagem no torso pode estar associado com todas as posições de junta conhecidas na imagem.

[0090] Um conjunto randômico de parâmetros de teste (os quais juntamente com os testes formam um conjunto randômico de funções de divisão) é então gerado 810 para utilização pelo teste binário executado no nodo de raiz como características candidatas. Em um exemplo, o teste binário é uma função aplicada no elemento de imagem x com parâmetros y, e com a saída da função comparada com os valores limite. Se o resultado do teste estiver dentro da faixa entre os valores limite então o resultado do teste binário é verdadeiro. De outro modo, o resultado do teste binário é falso. Em outros exemplos, somente um dos valores limite pode ser utilizado, de modo que o resultado do teste binário é verdadeiro se o resultado for maior do que (ou alternativamente menor do que) um valor limite. No exemplo aqui descrito, o parâmetro pode definir uma característica da imagem.

[0091] O processo para gerar o parâmetro pode compreender gerar valores de deslocamento espacial randômicos na forma de um deslocamento bi ou tridimensional. O resultado da função de divisão é

então computado observando o valor de profundidade para um elemento de imagem de teste o qual está deslocado do elemento de imagem de interesse x na imagem pelo deslocamento espacial. Os deslocamentos espaciais são opcionalmente feitos variantes de profundidade escalando por 1/profundidade do elemento de imagem de interesse. Os valores limite podem ser utilizados para decidir se o elemento de imagem de teste é uma posição de junta.

[0092] O resultado do teste (de divisão) binário executado em um nodo de raiz ou nodo de divisão determina para qual nodo filho um elemento de imagem é passado. Por exemplo, se o resultado do teste binário for verdadeiro, o elemento de imagem é passado para um primeiro nodo filho, enquanto que se o resultado for falso, o elemento de imagem é passado para um segundo nodo filho.

[0093] O conjunto randômico de parâmetros de teste gerados comprehende uma pluralidade de valores randômicos para o parâmetro de função de divisão e os valores limite. De modo a injetar randomicidade nas árvores de decisão, os parâmetros de função de cada nodo de divisão são otimizados somente sobre um subconjunto randomicamente amostrado de todos os parâmetros possíveis. Este é um modo efetivo e simples de injetar randomicidade nas árvores, e aumenta a generalização.

[0094] Um seletor de função de divisão de árvore pode ser utilizado 814 para selecionar qual das funções de divisão randomicamente geradas deve ser utilizada em um dado nodo de divisão. Isto acelera o processo de treinamento porque não é mais necessário aplicar todas as combinações de parâmetros de teste a cada elemento de imagem no conjunto de imagens de treinamento. Isto é porque a lógica de corrida é capaz de omitir opções de função de divisão de acordo com os intervalos de confiança calculados como acima descrito com referência às Figuras 3 e 4. Deste modo é possível

significativamente reduzir os tempos de treinamento enquanto mantendo a precisão em uma quantidade especificada utilizando o valor de tolerância de erro.

[0095] A lógica de pontuação calcula uma pontuação, também referida como critérios ou objetivos. Em um exemplo, a pontuação calculada compreende o ganho de informações (também conhecido como a entropia relativa). Em outro exemplo, a pontuação calculada compreende um ganho de Gini. A função de divisão que otimiza a pontuação (tal como maximizando o ganho de informações) é selecionada 814 e armazenadas no nodo corrente para utilização futura. Como uma alternativa ao ganho de informações, outros critérios podem ser utilizados tal como variância, ou outras estatísticas de diferença limitadas.

[0096] É então determinado 816 se o valor para a pontuação calculada é menor do que (ou maior do que) um limite. Se o valor para a pontuação calculada for menor do que o limite, então isto indica que uma expansão adicional da árvore não provê um benefício significativo. Isto dá origem a árvores assimétricas as quais naturalmente param de crescer quando nenhum nodo adicional é benéfico. Em tais casos, o nodo corrente é ajustado 818 como um nodo folha. Similarmente, a profundidade corrente da árvore é determinada (isto é, quantos níveis de nodos estão entre o nodo de raiz e o nodo corrente). Se isto for maior do que um valor máximo predefinido, então o nodo corrente é ajustado 818 como um nodo folha. Cada nodo folha tem votos de posição de junta os quais acumulam no nodo folha durante o processo de treinamento como abaixo descrito.

[0097] É também possível utilizar outros critérios de parada em combinação com aqueles já mencionados. Por exemplo, avaliar o número de exemplos de elementos de imagem que atingem a folha.

Se existirem muito poucos exemplos (comparado com um limite, por exemplo) então o processo pode ser disposto para parar para evitar um sobreajuste. No entanto, não é essencial utilizar este critério de parada.

[0098] Se o valor para o critério calculado for maior do que ou igual ao limite, e a profundidade de árvore for menor do que o valor máximo, então o nodo corrente é ajustado 820 como um nodo de divisão. Como o nodo corrente é um nodo de divisão, este tem nodos filhos, e o processo então move para treinar estes nodos filhos. Cada nodo filho é treinado utilizando um subconjunto dos elementos de imagem de treinamento no nodo corrente. O subconjunto de elementos de imagem enviado para um nodo filho é determinado utilizando os parâmetros que otimizaram o critério. Estes parâmetros são utilizados no teste binário, e o teste binário executado 822 em todos os elementos de imagem no nodo corrente. Os elementos de imagem que passam no teste binário formam um primeiro subconjunto enviado para um primeiro nodo filho, e os elementos de imagem que falham no teste binário formam um segundo subconjunto enviado para um segundo nodo filho.

[0099] Para cada um dos nodos filhos, o processo como delineado nos blocos 810 a 822 da Fig. 8 é recursivamente executado 824 para o subconjunto de elementos de imagem direcionados para o respectivo nodo filho. Em outras palavras, para cada nodo filho, novos parâmetros de teste randômicos são gerados 810, aplicados 812 ao respectivo subconjunto de elementos de imagem, parâmetros que otimizam o critério selecionado 814, e o tipo de nodo (divisão ou folha) determinado 816. Se este for um nodo folha, então a ramificação corrente de recursão cessa. Se este for um nodo de divisão, os testes binários são executados 822 para determinar subconjuntos adicionais de elementos de imagem e outra ramificação de recursão inicia.

Portanto, este processo recursivamente move através da árvore, treinamento cada nodo até que os nodos folha são alcançados em cada ramificação. Conforme os nodos folha são alcançados, o processo aguarda 826 até que os nodos em todas as ramificações tenham sido treinados. Note que, em outros exemplos, a mesma funcionalidade pode ser obtida utilizando técnicas alternativas para recursão.

[0100] Uma vez que todos os nodos na árvore foram treinados para determinar os parâmetros para o teste binário otimizando o critério m cada nodo de divisão, os nodos folha foram selecionados para terminar cada ramificação, então os votos podem ser acumulados 828 nos nodos folha da árvore. Este é o estágio de treinamento e assim elementos de imagem específicos os quais atingem um dado nodo folha especificaram votos de posição de junta relativos conhecidos dos dados de treinamento de precisão. Uma representação dos votos acumulados pode ser armazenada 830 utilizando vários diferentes métodos. Opcionalmente uma amostragem pode ser utilizada para selecionar votos a serem acumulados e armazenados de modo a manter uma baixa pegada de memória. Por exemplo, uma amostragem de reservatório pode ser utilizada por meio de que uma amostra de votos de tamanho máximo fixo é tomada. A seleção pode ser randômica ou em qualquer outro modo.

[0101] Uma vez que os votos acumulados foram armazenados é determinado 832 se mais árvores estão presentes na floresta de decisão. Se assim, então a próxima árvore na floresta de decisão é selecionada, e o processo repete. Se todas as árvores na floresta foram treinadas e nenhuma outra resta, então o processo de treinamento está completo e o processo termina 834.

[0102] Portanto, como um resultado do processo de treinamento, uma ou mais árvores de decisão são treinadas utilizando imagens de

treinamento sintetizadas ou empíricas. Cada árvore compreende uma pluralidade de nodos de divisão que armazenam parâmetros de teste otimizados, e nodos folha que armazenam votos de posição de junta associados ou representações de votos de posição de junta agregados. Devido à geração randômica de parâmetros de um subconjunto limitado utilizado em cada nodo, as árvores da floresta são distintas (isto é, diferentes) umas das outras.

[0103] O processo de treinamento pode ser executado com antecedência de utilizar o sistema de predição treinado para identificar as posições de junta em uma imagem de profundidade real. A floresta de decisão e os parâmetros de teste otimizados podem ser armazenados em um dispositivo de armazenamento para utilização na identificação de posições de junta em imagens de profundidade posteriormente.

[0104] A Fig. 10 ilustra um fluxograma de um processo para predizer posições de junta em uma imagem de profundidade previamente não vista utilizando uma floresta de decisão que foi treinada como aqui acima descrito. Primeiramente, uma imagem de profundidade na vista é recebida 1000. Uma imagem é referida como 'não vista' para distingui-la de uma imagem de treinamento a qual tem as posições de junta já especificadas. Note que a imagem de profundidade não vista pode ser pré-processada a um certo grau, por exemplo, para identificar as regiões de primeiro plano, o que reduz o número de elementos de imagem a serem processadas pela floresta de decisão. No entanto, um pré-processamento para identificar as regiões de primeiro plano não é essencial. Em alguns exemplos a imagem de profundidade na vista é uma imagem de silhueta como acima descrito onde a imagem de profundidade é efetivamente achatada para uma profundidade fixa.

[0105] Um elemento de imagem da imagem não vista é

selecionado 1002. Uma árvore de decisão treinada da floresta de decisão é também selecionada 1004. O elemento de imagem selecionado é empurrado 1006 através da árvore de decisão selecionada, de modo que este é testado contra os parâmetros treinados em um nodo, e então passado o filho apropriado na dependência do resultado do teste, e o processo é repetido até que o elemento de imagem atinja um nodo folha. Uma vez que o elemento de imagem atinge um nodo folha, os votos acumulados (do estágio de treinamento) associados com este nodo folha são armazenados 1008 para este elemento de imagem.

[0106] Se for determinado 1010 que existem mais árvores de decisão na floresta, então uma nova árvore de decisão é selecionada 1004, o elemento de imagem empurrado 1006 através da árvore e os votos acumulados armazenados 1008. Isto é repetido até que tenha sido executado para todas as árvores de decisão na floresta. Note que o processo para empurrar um elemento de imagem através da pluralidade de árvores na floresta de decisão pode também ser executado em paralelo, ao invés de em sequência como mostrado na Fig. 10.

[0107] É então determinado 1012 se elementos de imagem não analisados adicionais estão presentes na imagem de profundidade não vista, e se assim outro elemento de imagem é selecionado e o processo repetido. Uma vez que todos os elementos de imagem na imagem não vista foram analisados, então os votos de posição de junta são obtidos para todos os elementos de imagem.

[0108] Conforme os elementos de imagem são empurrados através das árvores na floresta de decisão, votos acumulam. Estes votos acumulados são agregados 1014 para formar uma agregação de votos total para cada elemento de imagem. Opcionalmente uma amostra de votos pode ser tomada para agregação. Por exemplo, N

votos podem ser escolhidos randomicamente, ou tomando os N votos ponderados superiores, e então o processo de agregação aplicado somente a estes N votos. Isto permite que a precisão seja negociada em relação à velocidade.

[0109] Pelo menos um conjunto de posições de junta pode então ser emitido 1016 onde as posições de junta podem ser ponderadas em confiança. Isto ajuda qualquer algoritmo de rastreamento subsequente (ou outro processo) avaliar se a proposta é boa ou não. Mais do que um conjunto de posições de junta podem ser emitidos; por exemplo, onde existe incerteza. Também, um conjunto posições de junta pode compreender nulos para uma ou mais juntas. Por exemplo, onde uma posição de junta está fora do campo de visão do sensor de formação de imagem e/ou onde nenhum elemento de imagem volta para uma posição de junta.

[0110] A Fig. 11 ilustra vários componentes de um dispositivo baseado em computação 504 exemplar o qual pode ser implementado como qualquer forma de um dispositivo de computação e/ou eletrônico, e no qual as modalidades de um sistema de aprendizado de máquina podem ser implementadas. Neste exemplo, o dispositivo baseado em computação 504 está disposto para treinar uma floresta de decisão randômica para predição de localização de junta (ou outras tarefas) de imagens de profundidade, por exemplo, para controlar um jogo utilizando movimentos de corpo humano. No entanto isto não é essencial; outras aplicações de aprendizado de máquina podem ser providas as quais envolvem selecionar uma opção de uma pluralidade de opções incerta.

[0111] O dispositivo baseado em computação 504 compreende um ou mais processadores 1100 os quais podem ser microprocessadores, controladores ou qualquer outro tipo adequado de processadores para processar instruções executáveis por computador para controlar a

operação do dispositivo, de modo a operar um sistema de aprendizado de máquina, por exemplo, para treinar uma floresta de decisão randômica para predição de localização de junta ou outras tarefas. Em alguns exemplos, por exemplo onde um sistema em uma arquitetura de chip é utilizado os processadores 1100 podem incluir um ou mais blocos de função fixos (também referidos como aceleradores) os quais implementam uma parte do método de qualquer das Figuras 1, 3, 4, 7, 8 e 10 em hardware (ao invés de software ou firmware). Um software de plataforma que compreende um sistema de operação 1114 ou qualquer outro software de plataforma adequado pode estar provido no dispositivo baseado em computação para permitir que o software de aplicação 1116 seja executado no dispositivo. Uma lógica de treinamento de árvore 1118 pode estar provida para treinar uma ou mais árvores de decisão randômicas utilizando pelo menos parte do processo das Figuras 7 e 8. Um seletor de opção 1124 pode estar provido o qual pode ser qualquer um do seletor de opção, seletor de modelo, seletor de característica e seletor de função de divisão de árvore da Fig. 1. Um armazenamento de dados 1122 pode armazenar um valor de tolerância de erro, árvores de decisão randômicas treinadas, exemplos de treinamento, opções selecionadas e outros dados. No caso em que o dispositivo baseado em computação está disposto para executar uma detecção de localização de junta de profundidade ou outras imagens uma lógica de predição de localização de junta 1120 está provida. A lógica de predição de localização de junta 1120 pode implementar pelo menos parte do método da Fig. 10.

[0112] As instruções executáveis por computador podem ser providas utilizando qualquer meio legível por computador que é acessível pelo dispositivo baseado em computação 504. O meio legível por computador pode incluir, por exemplo, um meio de armazenamento de computador tal como uma memória 1112 e meios

de comunicações. O meio de armazenamento de computador, tal como a memória 1112, inclui meio volátil e não volátil, removível e não removível implementado em qualquer método ou tecnologia para armazenamento as informações tal como instruções executáveis por computador, estrutura de dados, módulos de programa ou outros dados. O meio de armazenado de computador inclui, mas não está limitado a RAM, ROM, EPROM, EEPROM, memória instantânea ou outra tecnologia de memória, CD-ROM, discos versáteis digitais (DVD) ou outro armazenamento ótico, cassete magnéticos, fita magnética, armazenamento de disco magnético ou outros dispositivos de armazenamento magnéticos, ou qualquer outro meio de não transmissão que pode ser utilizado para armazenar informações para acesso um dispositivo de computação. Em contraste, o meio de comunicação pode incorporar instruções executáveis por computador, estruturas de dados, módulos de programa, ou outros dados em um sinal de dados modulado, tal como uma onda portadora, ou outro mecanismo de transporte. Como aqui definido, meio de armazenamento de computador não inclui meios de comunicação. Portanto, um meio de armazenamento de computador não deve ser interpretado ser um sinal propagante por si. Os sinais propagados podem estar presentes em um meio de armazenamento de computador, as os sinais propagados por si não exemplos de meio de armazenamento de computador. Apesar do meio de armazenamento de computador (memória 1112) ser mostrado dentro do dispositivo baseado em computação 504 será apreciado que o armazenamento pode ser distribuído ou localizado remotamente e acessado através de uma rede ou outra conexão de comunicação (por exemplo, utilizando a interface de comunicação 1113).

[0113] O dispositivo baseado em computação 504 também compreender uma interface de entrada 1102 e uma interface de saída

1110 disposta para emitir informações de display para um dispositivo de display 508 o qual pode separado do ou integral com o dispositivo baseado em computação 504. As informações de display podem prover uma interface gráfica de usuário. A interface de entrada 1102 está também disposta para receber e processar a entrada de um ou mais dispositivos, tal como o dispositivo de entrada de usuário (por exemplo, um mouse 1108, um teclado 1106, controlador de jogo 1104, câmera 506, microfone ou outro sensor). Em alguns exemplos a interface de entrada 1102 pode detectar uma entrada de voz, gestos de usuário ou outras ações de usuário e pode prover uma interface de usuário natural (NUI). Esta entrada de usuário pode ser utilizada para controlar um jogo ou outra aplicação. Em uma modalidade o dispositivo de display 508 pode também atuar como um dispositivo de entrada de usuário se este for um dispositivo de display sensível ao toque. A interface de saída 1110 pode também emitir dados para dispositivos outros que o dispositivo de display, por exemplo, um dispositivo de impressão localmente conectado.

[0114] A interface de entrada 1102, a interface de saída 1110 e opcionalmente os dispositivos de entrada de usuário (controlador de jogo 1104, teclado 1106, mouse 1108, dispositivo de captura 506) podem compreender a tecnologia NUI a qual permite um usuário interagir com o dispositivo baseado em computação em um modo natural, livre de restrições artificiais impostas por dispositivos de entrada tais como mouses, teclados, controles remotos e similares. Exemplos de tecnologia NUI que podem ser providos incluem, mas não estão limitados àqueles que se baseiam em reconhecimento de voz e/ou fala, reconhecimento de toque e/ou caneta (displays sensíveis ao toque), reconhecimento de gestos, tanto na tela quanto adjacentes à tela, gestos no ar, rastreamento de cabeça e olhos, voz e fala, visão, toque, gestos, e inteligência de máquina. Outros exemplos

tecnologia NUI que podem ser utilizados incluem sistemas de compreensão de intenção de objetivo, sistemas de detecção de gesto de movimento utilizando câmeras de profundidades (tal como sistemas de câmera estereoscópicos, sistema de câmera infravermelho, sistema de câmeras RGB e combinações destes), detecção de gesto e movimento utilizando acelerômetros / giroscópio, reconhecimento facial, displays 3D, rastreamento de cabeça, olhos e olhar, sistemas de realidade aumentada imersiva e de realidade virtual e tecnologias para detectar atividades cerebrais utilizando eletrodos de detecção de campo elétrico (EEG e métodos relativos).

[0115] O termo 'computador' ou 'dispositivo baseado em computação' é aqui utilizado para referir a qualquer dispositivo com capacidade de processamento de modo que este possa executar instruções. Aqueles versados na técnica perceberão que tais capacidade de processamento estão incorporadas em muitos diferentes dispositivos e, portanto os termos 'computador' e 'dispositivo baseado em computação' cada um inclui PCs, servidores, telefones móveis (incluindo smartphones), computadores tablet, decodificadores, media players, consoles de jogos, assistentes digitais pessoais e muitos outros dispositivos.

[0116] Os métodos aqui descritos podem ser executados por software em forma legível por máquina em um meio de armazenamento tangível, exemplo, na forma de um programa de computador que compreende um meio de código de programa de computador adaptado para executar todas as etapas de qualquer um dos métodos aqui descritos em que quanto o programa é executado em um computador e onde o programa de computador pode ser incorporado em um meio legível por computador. Exemplos de meios de armazenamento tangíveis incluem dispositivos de armazenamento de computador que compreender um meio legível por computador tal

como discos, pen drives, memória etc. e não incluem sinais propagados. Os sinais propagados podem estar presentes em um meio de armazenamento tangível, mas os sinais propagados por si não exemplos de meios de armazenamento tangíveis. O software pode ser adequado para execução em um processador paralelo ou um processador seria de modo que as etapas de método possam ser executadas em qualquer ordem adequada, ou simultaneamente.

[0117] Isto reconhece que o software pode ser uma mercadoria valiosa, separadamente comerciável. É pretendido abranger um software o qual executa ou controla um hardware "mudo" ou padrão, para executar as funções desejadas, este também pretende abranger um software o qual "descreve" ou define a configuração de hardware, tal como um software HDL (linguagem de descrição de hardware), como é utilizado para designar chips de silício, ou para configurar chips programáveis universais, para executar funções desejadas.

[0118] Aqueles versados na técnica perceberão que os dispositivos de armazenamento utilizados para armazenar instruções de programa podem estar distribuídos através de uma rede. Por exemplo, um computador remoto pode armazenar um exemplo do processo descrito como software. Um computador local ou terminal pode acessar o computador remoto e fazer o download de uma parte ou todo o software para executar o programa. Alternativamente, o computador local pode fazer o download de partes do software conforme necessário, ou executar algumas instruções de software no terminal local e algumas no computador remoto (ou rede de computador). Aqueles versados na técnica também perceberão que utilizando as técnicas convencionais conhecidas daqueles versados na técnica que todas, ou uma porção das instruções de software podem ser executadas por um circuito dedicado, tal como um DSP, rede de lógica programável, ou similares.

[0119] Qualquer faixa ou valor de dispositivo aqui fornecido pode ser estendida ou alterada sem perder o efeito buscado, como será aparente para a pessoa versada na técnica.

[0120] Apesar do assunto ter sido descrito em linguagem específica para características estruturais e/ou atos metodológicos, deve ser compreendido que o assunto definido nas reivindicações anexas não está necessariamente limitado às características ou atos específicos acima descritos. Ao invés, as características e atos específicos acima descritos estão descritos como formas exemplares de implementar as reivindicações.

[0121] Será compreendido que os benefícios e vantagens acima descritos podem referir a uma modalidade ou podem referir as diversas modalidades. As modalidades não estão limitadas àquelas que resolvem qualquer ou todos os problemas apresentados ou àquelas que têm qualquer ou todos os benefícios e vantagens declarados. Será ainda compreendido que referência a 'um' item refere-se a um ou mais destes itens.

[0122] As etapas e métodos aqui descritos podem ser executados em qualquer ordem adequada, ou simultaneamente onde apropriado. Além disso, blocos individuais podem ser apagados de qualquer dos métodos sem afastar do espírito e escopo do assunto aqui descrito. Aspectos de quaisquer dos exemplos acima descritos podem ser combinados com aspectos de qualquer dos outros exemplos descritos para formar exemplos adicionais sem perder o efeito buscado.

[0123] O termo 'compreendendo' é aqui utilizado para significar incluindo os blocos de método ou elementos identificados, mas que tais blocos ou elementos não compreendem uma lista exclusiva e um método ou aparelho pode conter blocos ou elementos adicionais.

[0124] Será compreendido que a descrição acima é dada como exemplo somente e que várias modificações podem ser feitas por

aqueles versados na técnica. A especificação acima, exemplos e dados proveem uma completa descrição da estrutura e utilização de modalidades exemplares. Apesar de várias modalidades terem sido acima descritas com um certo grau de particularidade, ou com referência a uma ou mais modalidades individuais, aqueles versados na técnica poderiam fazer numerosas alterações nas modalidades descritas sem afastar do espírito ou escopo desta especificação.

REIVINDICAÇÕES

1. Método, implementado em computador, em um sistema de aprendizado de máquina, **caracterizado pelo fato de que** comprehende as etapas de:

em um processador, receber para cada uma de uma pluralidade de opções (100) às quais os recursos do sistema de aprendizado de máquina podem ser alocados, uma pluralidade de amostras (110) descrevendo o comportamento da opção (100), o número de amostras com base em pelo menos em parte, em uma tolerância de erros determinada automaticamente;

para cada opção (100), calcular (308) uma pontuação usando as amostras, a pontuação compreendendo uma estatística de diferença limitada;

para cada opção (100), calcular um intervalo de confiança da pontuação usando as amostras calculando a pontuação para essa opção, mais ou menos um tempo constante da raiz quadrada de, uma variação estimada dos tempos de pontuação, um logaritmo de dois dividido por a tolerância a erros;

selecionar uma ou mais das opções (100) com base nos intervalos de confiança e nas pontuações, selecionar uma ou mais opções (100) com base nos intervalos de confiança, incluindo:

determinar uma opção de pontuação mais alta da pluralidade de opções (100);

comparar o intervalo de confiança da opção de pontuação mais alta com o intervalo de confiança de uma opção diferente da pluralidade de opções (100); e

excluir a opção diferente de uma análise mais aprofundada quando o intervalo de confiança da opção de pontuação mais alta e o intervalo de confiança da opção diferente não se sobrepõem; e

alocar recursos do sistema de aprendizado de máquina

para uma ou mais opções selecionadas.

2. Método, de acordo com a reivindicação 1, **caracterizado** pelo fato de que a estatística de diferença limitada é um valor numérico obtido das amostras o qual descreve o comportamento da opção (100); e o valor numérico muda ligeiramente quando qualquer uma das amostras é omitida ou mudada.

3. Método, de acordo com a reivindicação 1 ou 2, **caracterizado** pelo fato de que a pontuação é selecionada de qualquer um ou mais de: um ganho de informações, um ganho de Gini, uma variância, uma entropia.

4. Método, de acordo com qualquer uma das reivindicações 1 a 3, **caracterizado** pelo fato de que comprehende selecionar uma opção (100) com base nos intervalos de confiança e nas pontuações de modo que a opção selecionada tem, dentro da tolerância de erro, a melhor pontuação de todas as opções (100).

5. Método, de acordo com qualquer uma das reivindicações 1 a 4, **caracterizado** pelo fato de que calcular o intervalo de confiança comprehende estimar uma variância da pontuação da opção.

6. Método, de acordo com qualquer uma das reivindicações 1 a 5, **caracterizado** pelo fato de que calcular (308) o intervalo de confiança comprehende estimar uma variância da pontuação da opção calculando uma estimativa de variância Jack-Knife.

7. Método, de acordo com qualquer uma das reivindicações 1 a 6, **caracterizado** pelo fato de que a tolerância de erros é baseada, pelo menos em parte, em um domínio de aplicativo ou em recursos de computação disponíveis.

8. Método, de acordo com qualquer uma das reivindicações 1 a 7, **caracterizado** pelo fato de que comprehende ajustar o intervalo de confiança calculado para a tendência.

9. Método, de acordo com qualquer uma das reivindicações

1 a 8, **caracterizado** pelo fato de que este é pelo menos parcialmente executado utilizando uma lógica de hardware selecionada de qualquer um ou mais de: uma rede de portas programáveis no campo, um circuito integrado específico de programa, um produto padrão específico de programa, um sistema em um chip, um dispositivo lógico programável complexo, uma unidade de processamento gráfico.

10. Método, de acordo com qualquer uma das reivindicações 1 a 9, **caracterizado** pelo fato de que para cada opção (100), o cálculo de um intervalo de confiança da pontuação inclui o cálculo de um intervalo de confiança usando as seguintes fórmulas: $Z - 5/2 V n J \delta + EV n J \log 2 \delta$

Imagen disponível em "Documento original"

$Z - 5/2 V n J \delta + EV n J \log 2 \delta$ biasImage disponível em "Documento original"

, que pode ser expresso em palavras como um intervalo da pontuação Z da opção menos $5/2$ vezes a raiz quadrada de, uma estimativa da variância de Jack-Knife $V n J$ Image disponível em "Documento original" do número de amostras n mais uma correção δ $V n J$ Image disponível em "Documento original" vezes um logaritmo de 2 dividido pela tolerância a erros δ ao escore Z da opção mais $5/2$ vezes a raiz quadrada de, uma estimativa da variância de Jack-Knife $V n J$ Image disponível em "Documento original" da variação do número de amostras n mais uma correção δ $V n J$ Image disponível em "Documento original" vezes um logaritmo de 2 dividido pela tolerância de erros δ mais um termo de tendência.

11. Aparelho de aprendizado de máquina **caracterizado pelo fato de que** comprehende:

uma lógica de pontuação (106) disposta para receber, para cada uma de uma pluralidade de opções (100) às quais os recursos do sistema de aprendizado de máquina podem ser alocados, uma

pluralidade de amostras que descrevem o comportamento da opção, o número de amostras com base pelo menos em parte de uma tolerância a erros determinada automaticamente;

a lógica de pontuação (106) sendo disposta para calcular, para cada opção (100), uma pontuação usando as amostras, a pontuação compreendendo uma estatística de diferença limitada;

uma lógica de corrida (108), disposta a calcular, para cada opção (100), um intervalo de confiança da pontuação usando as amostras calculando a pontuação para essa opção, mais ou menos vezes constantes a raiz quadrada de, uma variação estimada dos tempos de pontuação, um logaritmo de dois dividido pela tolerância de erros, para determinar a melhor opção de pontuação da pluralidade de opções (100), para comparar o intervalo de confiança da melhor opção de pontuação com o intervalo de confiança de uma opção diferente da pluralidade das opções (100) e excluir a opção diferente de uma análise mais aprofundada quando o intervalo de confiança da melhor opção de pontuação e o intervalo de confiança da opção diferente não se sobrepõem;

um processador (1100) disposto para selecionar uma ou mais das opções (100) com base nos intervalos de confiança e nas pontuações e para alocar recursos do aparelho de aprendizado de máquina às uma ou mais opções selecionadas.

12. Aparelho de aprendizado de máquina, de acordo com a reivindicação 11, **caracterizado** pelo fato de que a lógica de corrida (108) que calcula o intervalo de confiança estimando uma variação da pontuação da opção ao calcular uma estimativa de variação de Jackknife.

13. Aparelho de aprendizado de máquina, de acordo com a reivindicação 11 ou 12, **caracterizado** pelo fato de que a lógica de corrida (108) é disposta para calcular para cada opção (100), um

intervalo de confiança da pontuação usando as seguintes fórmulas: $Z - 5/2 V_n J \circ EV_n J \log 2 \delta$

Imagen disponível em "Documento original"

$Z 5/2 V_n J \circ EV_n J \log 2 b_{bias}$ Image disponível em "Documento original"

, que pode ser expresso em palavras como um intervalo da pontuação Z da opção menos $5/2$ vezes a raiz quadrada de, uma estimativa da variância de Jack-Knife da variância $V_n J$ Image disponível em "Documento original"

do número de amostras n mais uma correção o $EV_n J$ Image disponível em "Documento original"

vezes um logaritmo de 2 dividido pela tolerância de erros δ ao escore Z da opção mais $5/2$ vezes a raiz quadrada de, uma estimativa da variância de Jack-Knife $V_n J$ Image disponível em "Documento original"

da variação do número de amostras n mais uma correção o $EV_n J$ Image disponível em "Documento original"

vezes um logaritmo de 2 dividido pela tolerância de erros δ mais um termo de tendência.

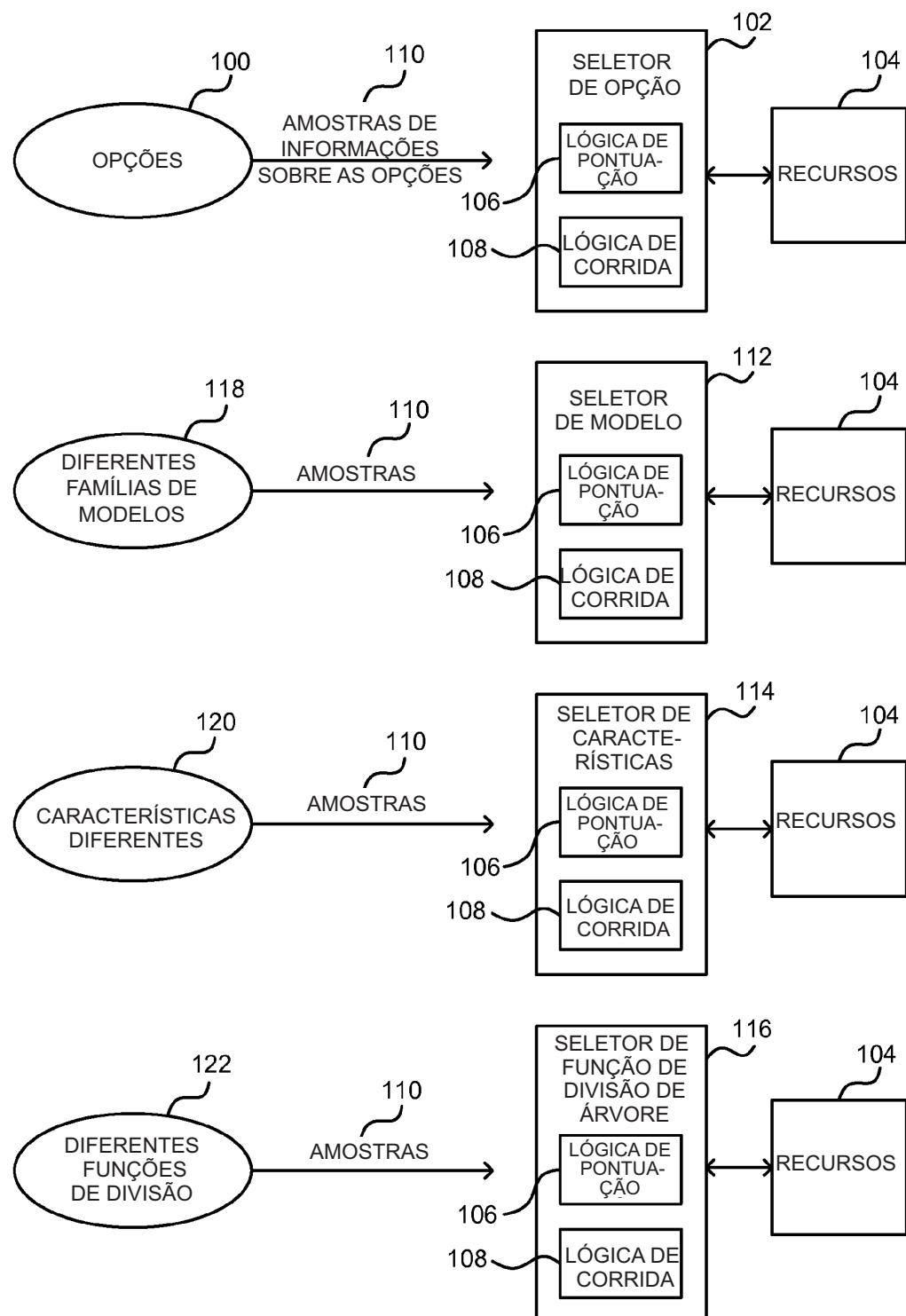


Fig. 1

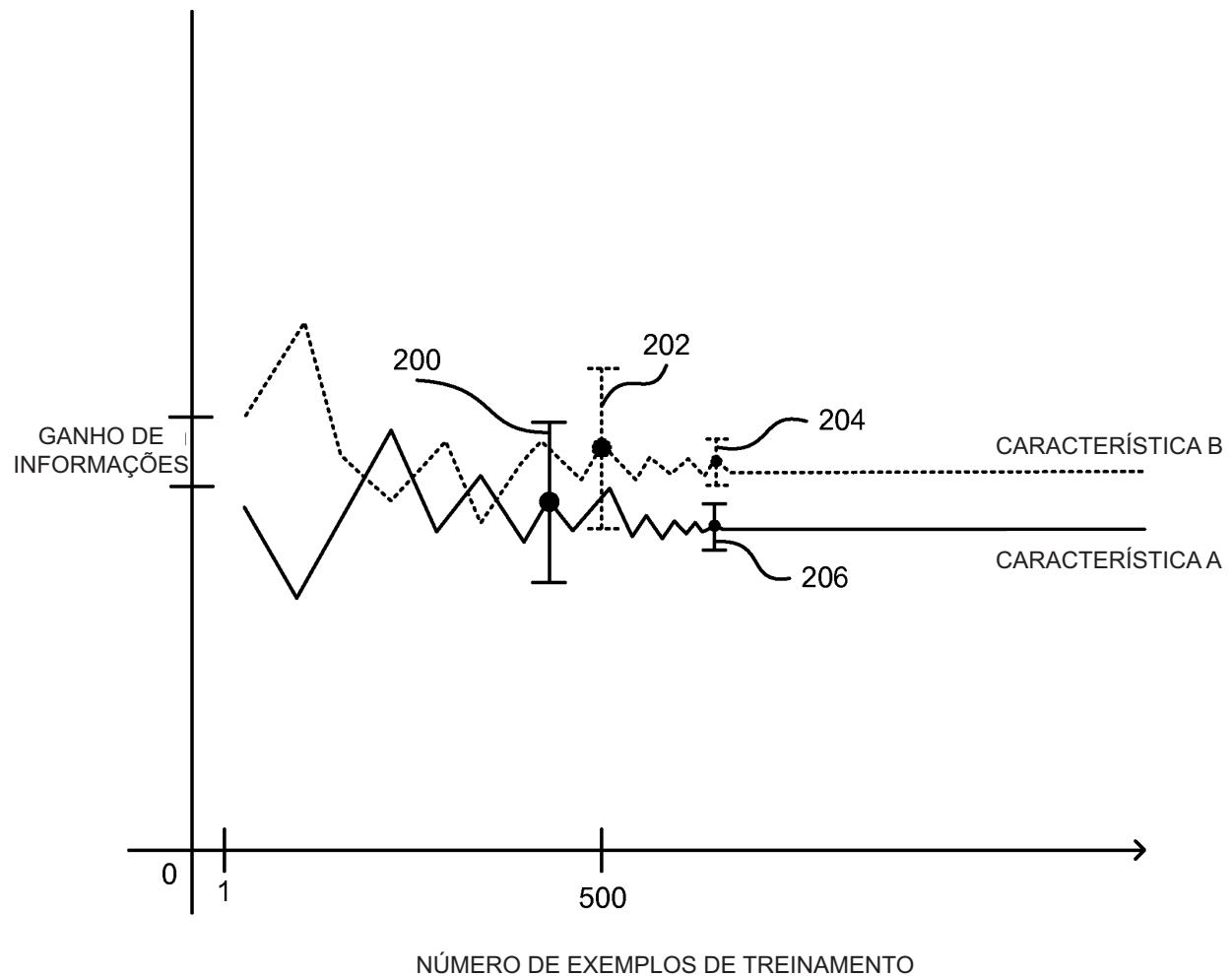


Fig. 2

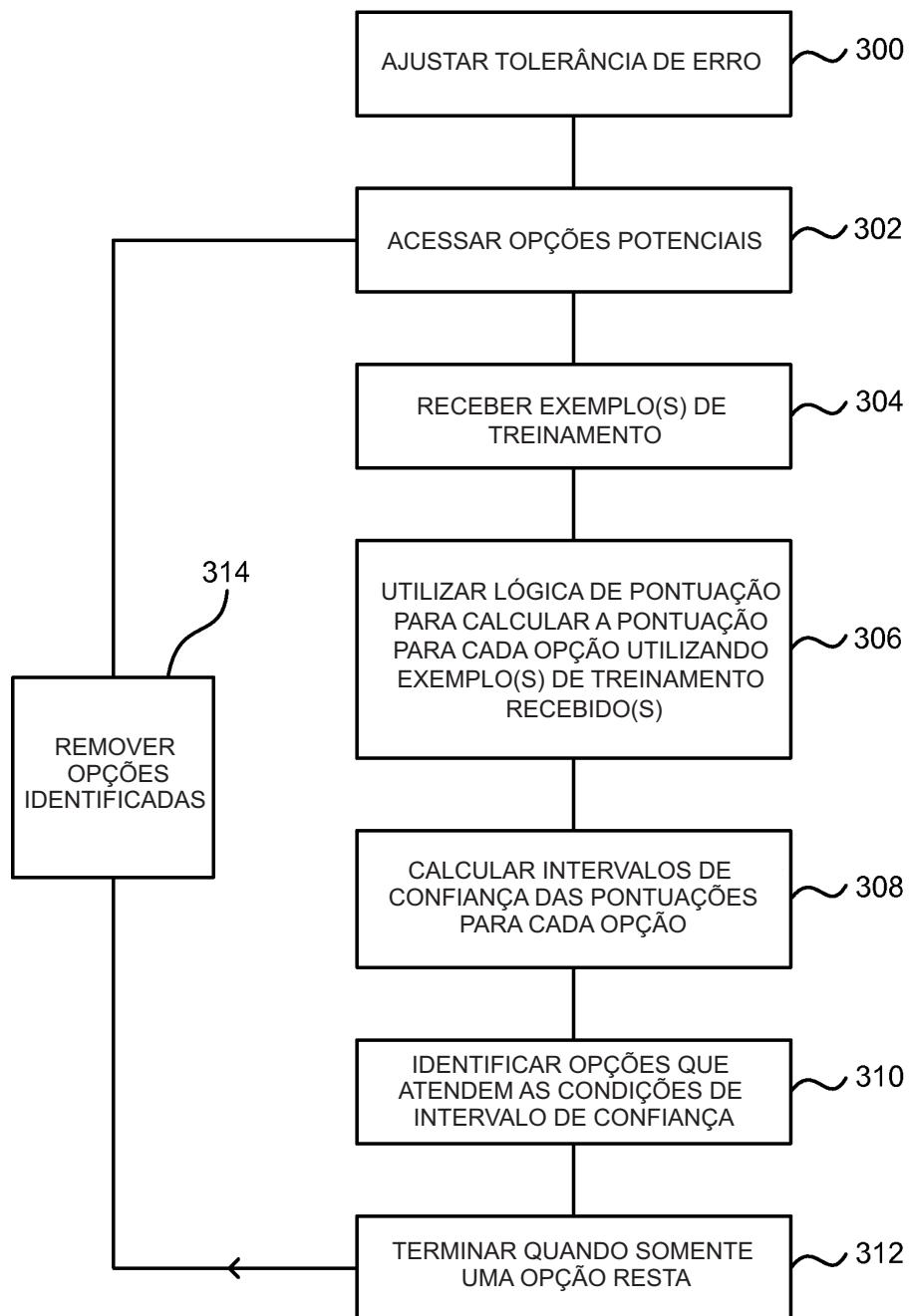


Fig. 3

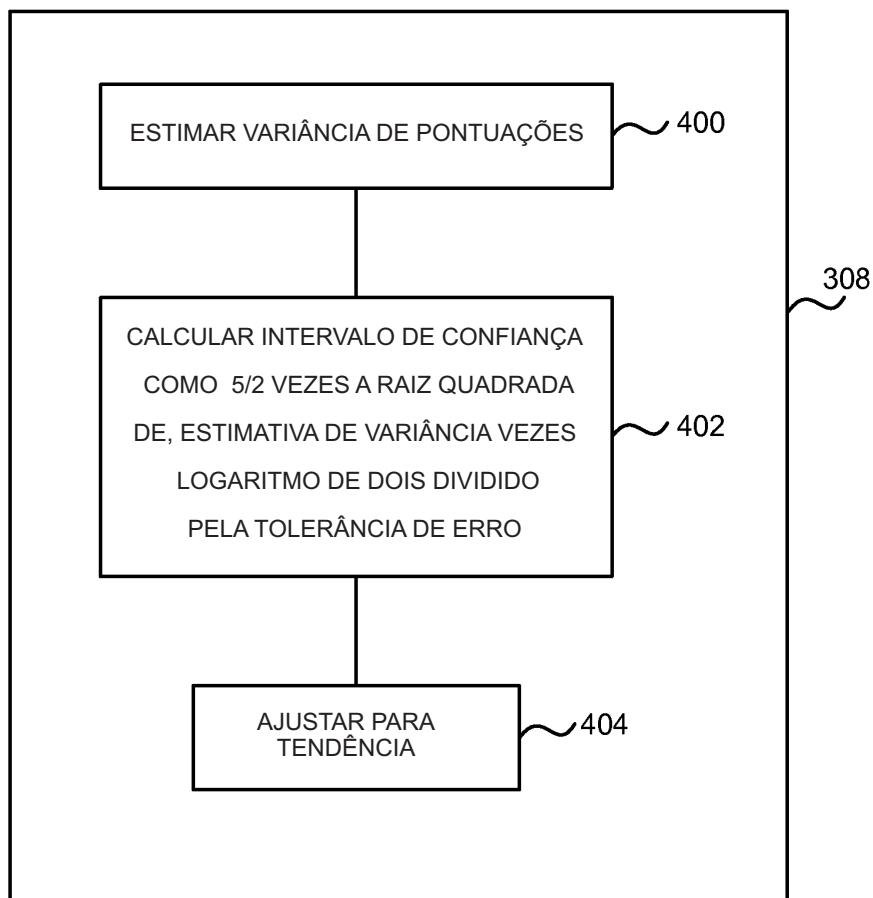


Fig. 4

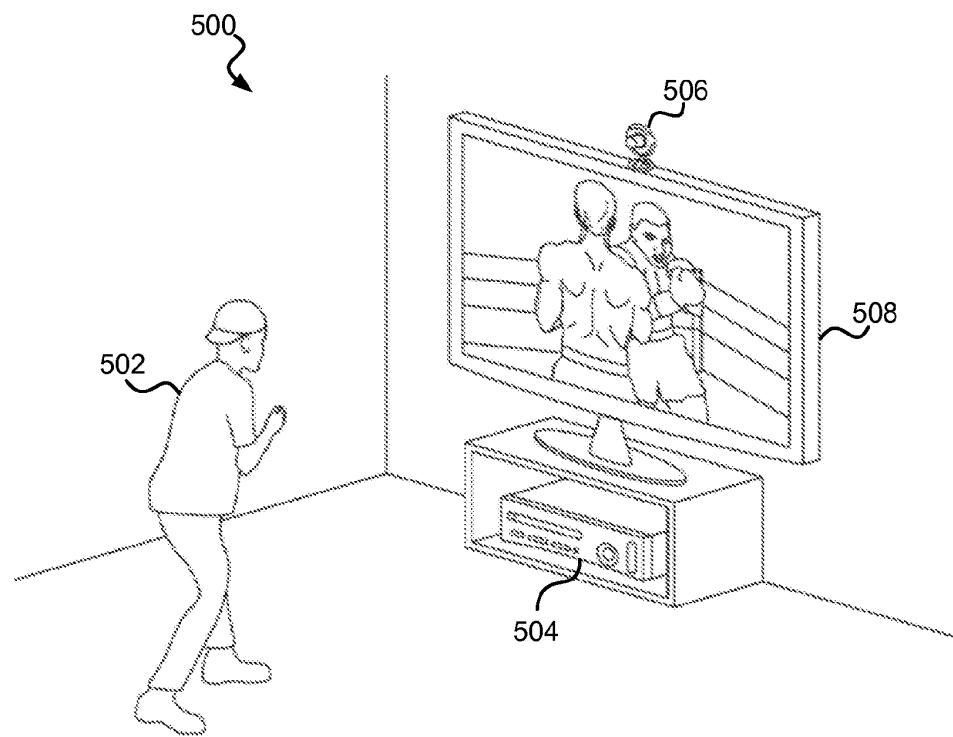


Fig. 5

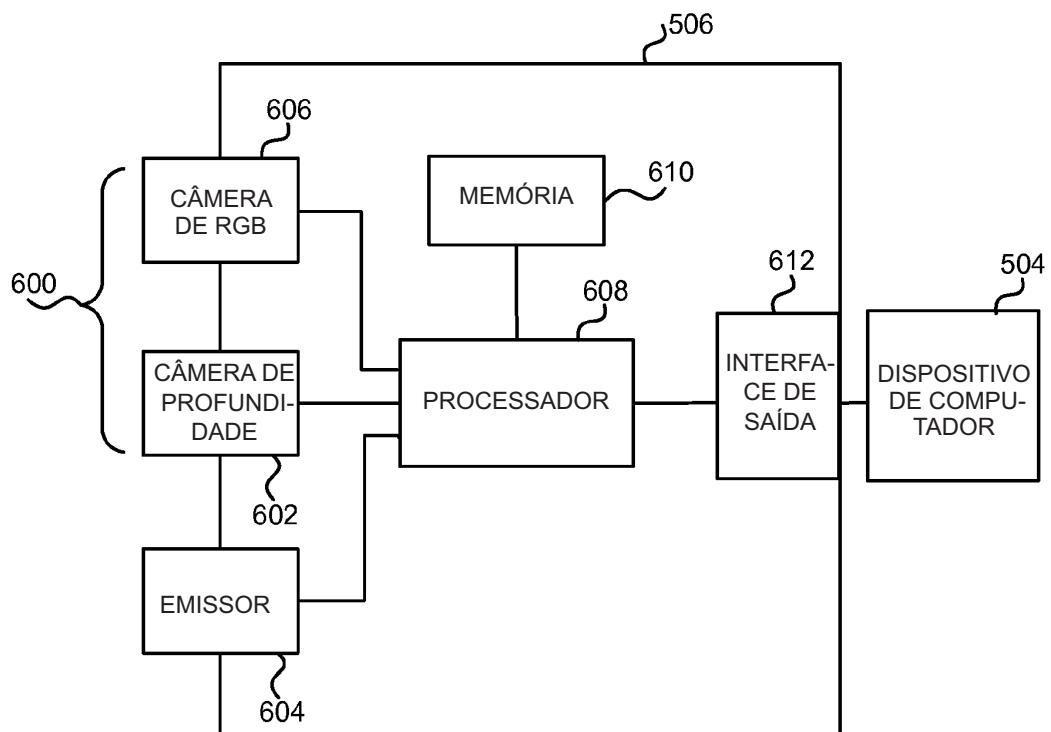


Fig. 6

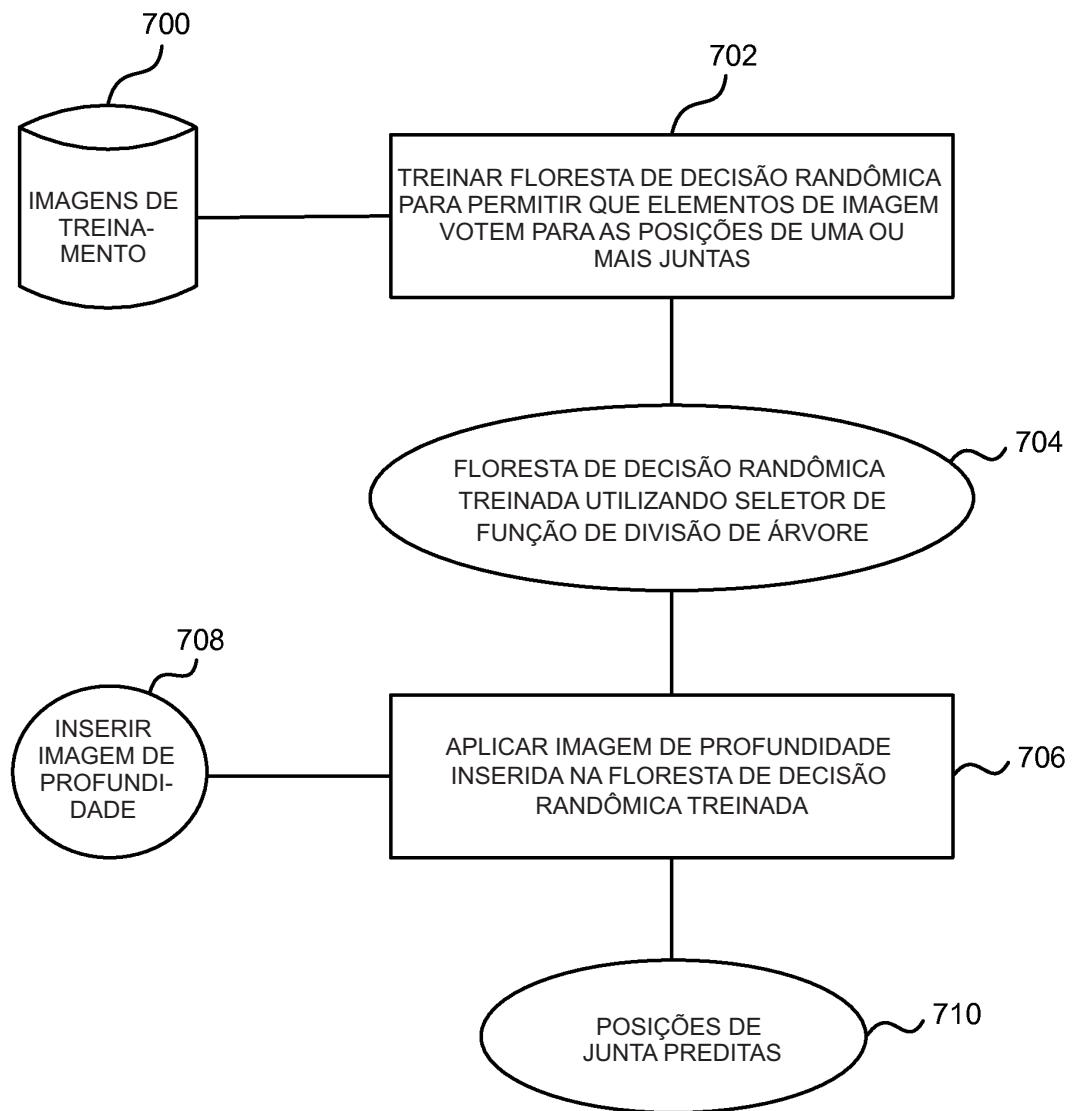


Fig. 7

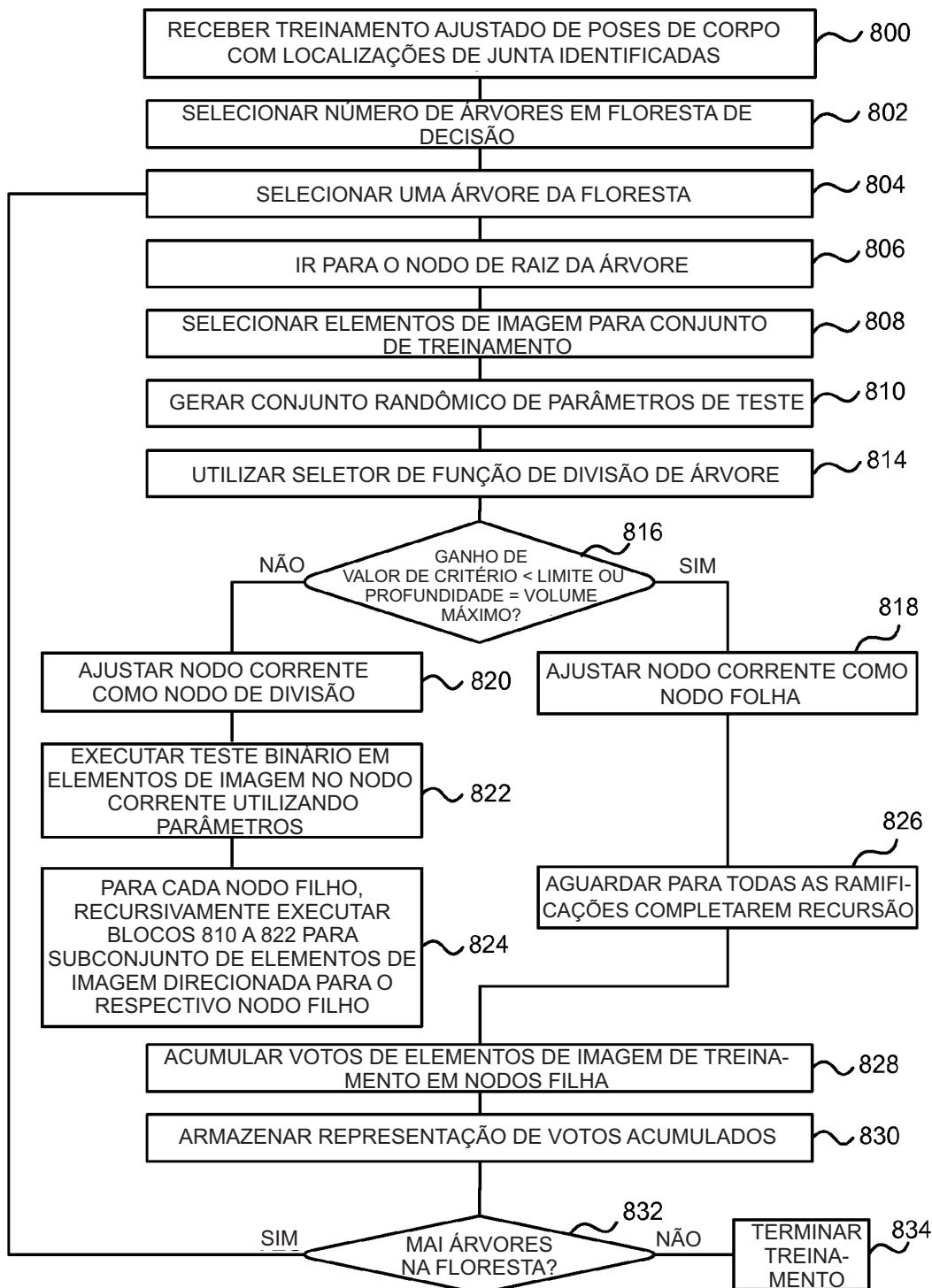


Fig. 8

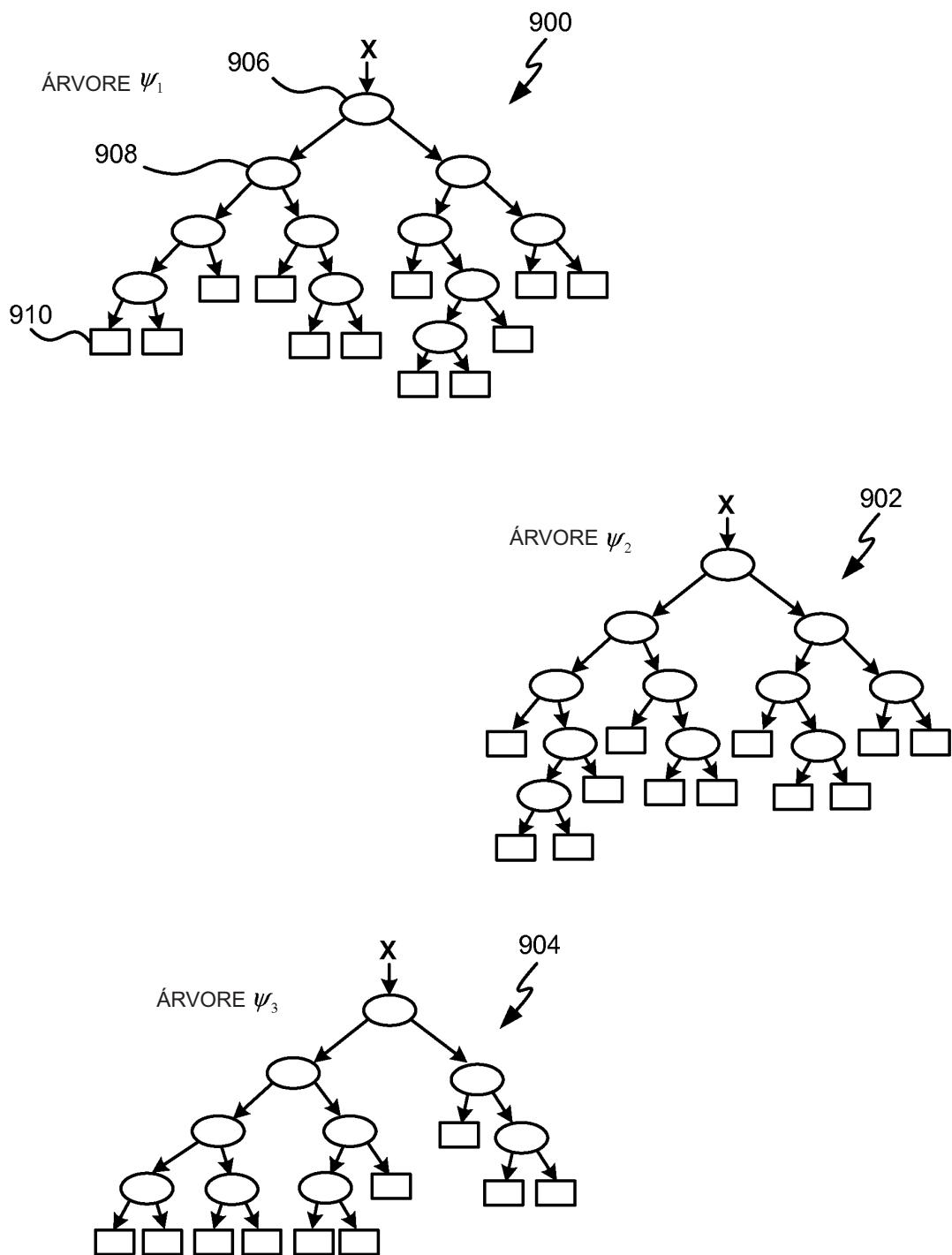


Fig. 9

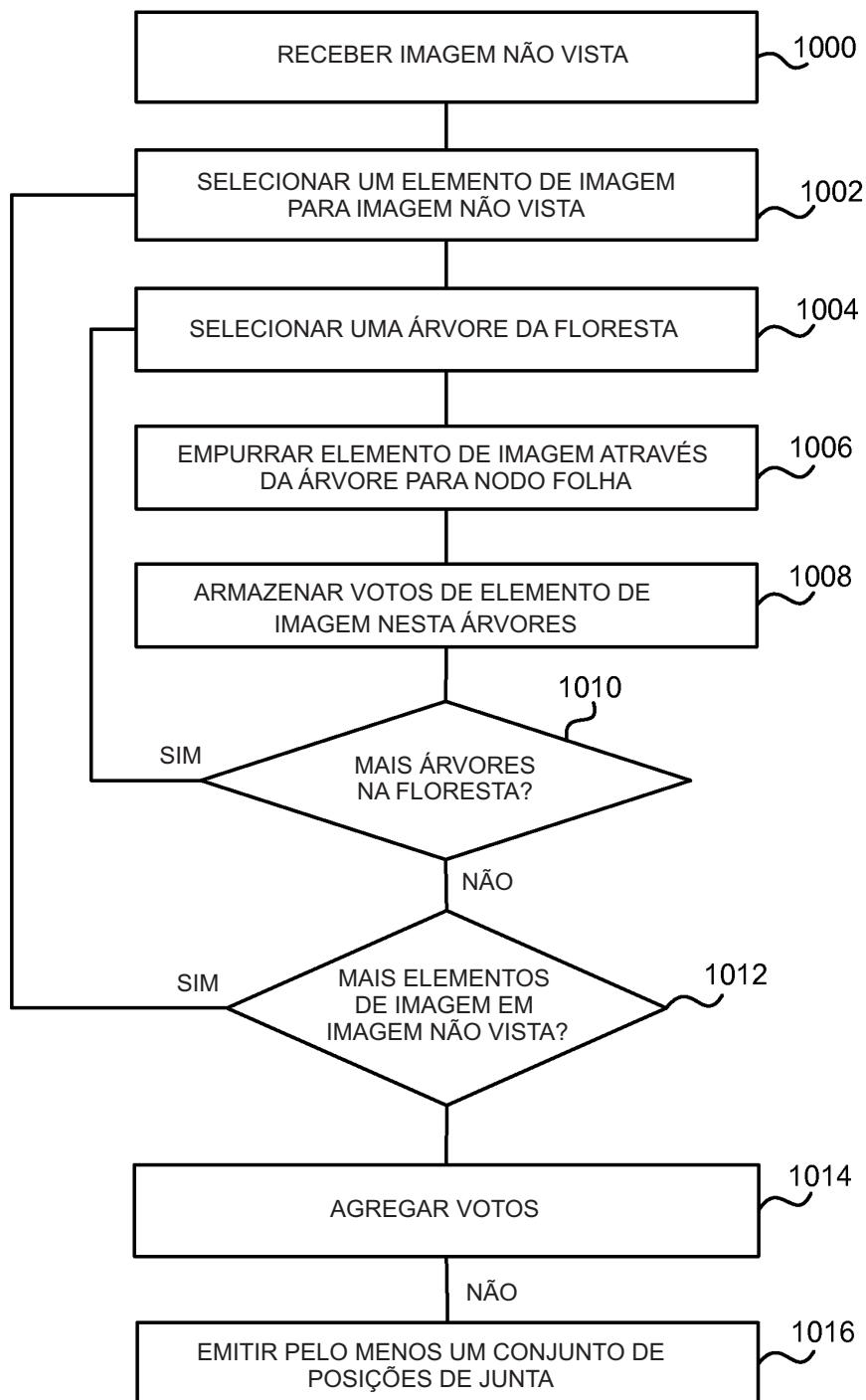


Fig. 10

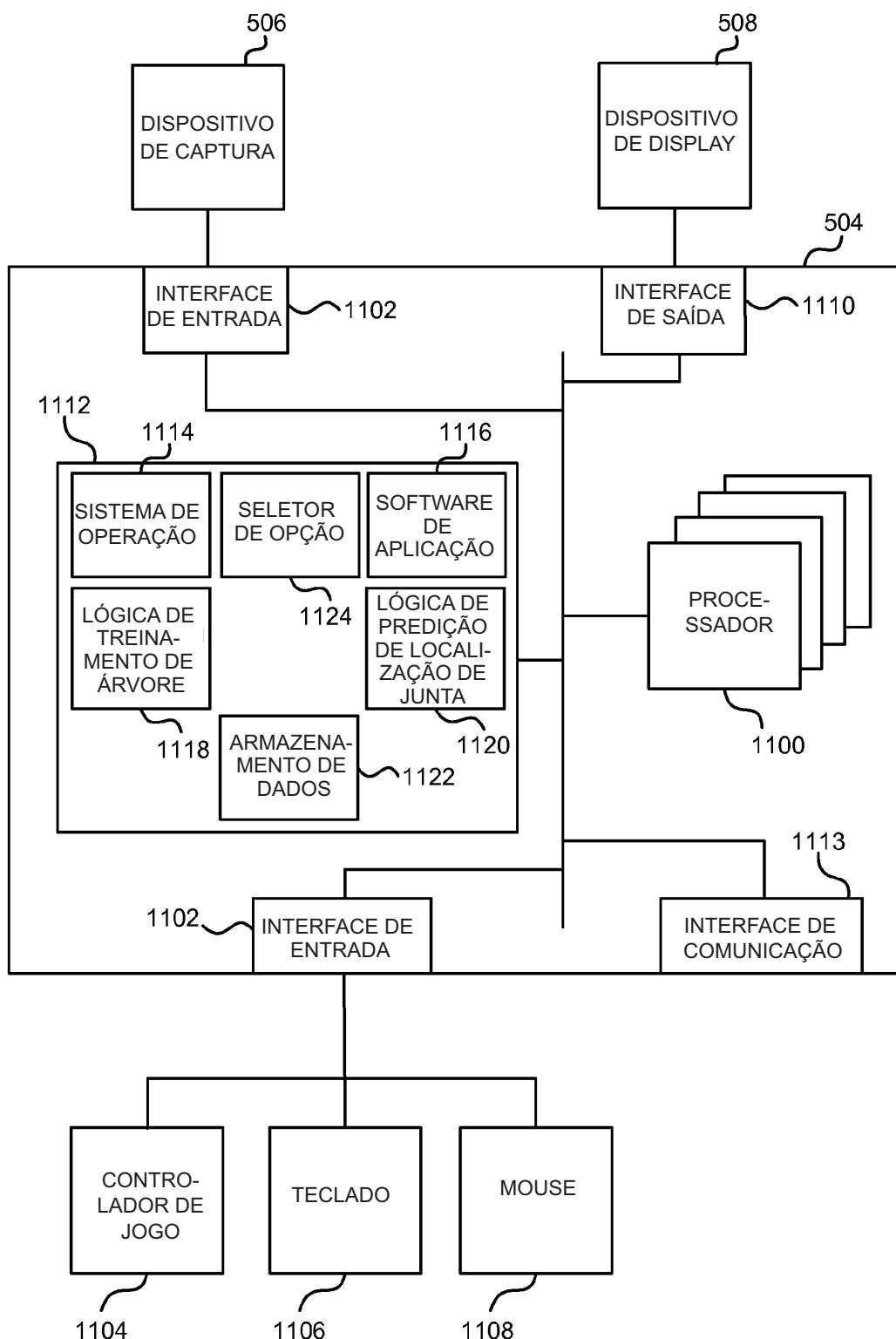


Fig. 11