

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号

特許第7007829号

(P7007829)

(45)発行日 令和4年1月25日(2022.1.25)

(24)登録日 令和4年1月12日(2022.1.12)

(51)国際特許分類

F I

G 0 6 N 3/08 (2006.01)

G 0 6 N

3/08

1 4 0

請求項の数 15 (全29頁)

(21)出願番号	特願2017-146800(P2017-146800)	(73)特許権者	000001007
(22)出願日	平成29年7月28日(2017.7.28)		キヤノン株式会社
(65)公開番号	特開2018-26122(P2018-26122A)		東京都大田区下丸子3丁目30番2号
(43)公開日	平成30年2月15日(2018.2.15)	(74)代理人	110003281
審査請求日	令和2年7月22日(2020.7.22)		特許業務法人大塚国際特許事務所
(31)優先権主張番号	特願2016-153212(P2016-153212)	(72)発明者	斎藤 侑輝
(32)優先日	平成28年8月3日(2016.8.3)		東京都大田区下丸子3丁目30番2号
(33)優先権主張国・地域又は機関	日本国(JP)		キヤノン株式会社内
		審査官	多胡 滋

最終頁に続く

(54)【発明の名称】 情報処理装置、情報処理方法及びプログラム

## (57)【特許請求の範囲】

## 【請求項1】

学習データを認識器に入力して出力データを得る処理手段と、

複数の識別器から一部の識別器を選択する選択手段と、前記出力データを前記選択された識別器に入力することにより得られる前記学習データの  
前記選択された識別器による識別結果の誤差を判定する判定手段と、前記識別結果の誤差に基づいて前記選択された識別器の学習処理を行う第2の学習手段  
と、前記識別結果の誤差を前記認識器に逆伝播することにより前記認識器の学習処理を行う  
第1の学習手段と、を備え、前記認識器は特徴ベクトルを前記出力データとして出力し、前記複数の識別器のそれぞれはハッシュ関数であって、前記選択された識別器には共通  
する前記特徴ベクトルが入力されることを特徴とする、情報処理装置。

## 【請求項2】

前記第2の学習手段は、前記選択された識別器のOne-class学習処理を行うこ  
とを特徴とする、請求項1に記載の情報処理装置。

## 【請求項3】

前記判定手段は、前記選択された識別器のそれぞれに前記出力データを入力することによ  
り得られる、前記選択された識別器のそれぞれによる前記学習データの識別結果の誤差の

それぞれに基づいて、前記選択された識別器全体についての識別結果の誤差を判定し、前記第 1 の学習手段は、前記判定された前記選択された識別器全体についての識別結果の誤差を認識器に逆伝播する

ことを特徴とする、請求項 1 又は 2 に記載の情報処理装置。

【請求項 4】

前記選択手段は、前記複数の識別器から第 1 の識別器セットと第 2 の識別器セットとを選択し、

前記第 1 の学習手段は、

学習データを前記認識器に入力して得られた出力データを前記第 1 の識別器セットに入力することにより得られる識別結果の誤差を、前記認識器に逆伝播することにより、前記認識器のパラメータを更新し、

学習データをパラメータ更新後の前記認識器に入力して得られた出力データを前記第 2 の識別器セットに入力することにより得られる識別結果の誤差を、前記認識器に逆伝播することにより、前記認識器のパラメータを再度更新する

ことを特徴とする、請求項 1 乃至 3 のいずれか 1 項に記載の情報処理装置。

【請求項 5】

前記第 1 の学習手段は、前記複数の識別器のうち前記一部の識別器以外の識別器の識別結果の誤差を前記認識器に逆伝播しないことを特徴とする請求項 1 乃至 4 のいずれか 1 項に記載の情報処理装置。

【請求項 6】

前記第 2 の学習手段は、前記複数の識別器のうち前記一部の識別器以外の識別器の識別結果の誤差に基づく前記一部の識別器の学習を行わないことを特徴とする請求項 1 乃至 5 のいずれか 1 項に記載の情報処理装置。

【請求項 7】

前記第 2 の学習手段は、前記複数の識別器間の類似度が低いほど評価値が高くなる目的関数を用いて学習を行うことを特徴とする、請求項 1 乃至 6 の何れか 1 項に記載の情報処理装置。

【請求項 8】

前記第 2 の学習手段は、前記複数の識別器間の直交性が高いほど評価値が高くなる目的関数を用いて学習を行うことを特徴とする、請求項 1 乃至 7 の何れか 1 項に記載の情報処理装置。

【請求項 9】

画像を表す画像データ及び当該画像の種類を示す教師情報を取得し、画像処理により 1 つの前記画像から複数の前記学習データを生成する前処理手段をさらに備え、

前記前処理手段は、前記画像処理により前記画像の種類が変化しない画像を選択し、前記選択された画像に対して前記画像処理を行い、前記画像処理前の前記選択された画像と、前記画像処理後の前記選択された画像と、を前記学習データとして出力することを特徴とする、請求項 1 乃至 8 の何れか 1 項に記載の情報処理装置。

【請求項 10】

前記複数の識別器は One - c l a s s 識別器、弱識別器、線形識別器、又はハッシュ関数を含むことを特徴とする、請求項 1 乃至 9 の何れか 1 項に記載の情報処理装置。

【請求項 11】

前記認識器はニューラルネットワークであることを特徴とする、請求項 1 乃至 10 の何れか 1 項に記載の情報処理装置。

【請求項 12】

前記情報処理装置は、入力された製品の画像に基づいて異常を検知することを特徴とする、請求項 1 乃至 11 の何れか 1 項に記載の情報処理装置。

【請求項 13】

前記識別器は、カーネル法に基づいて非線形にデータを写像することを特徴とする、請求項 1 乃至 12 の何れか 1 項に記載の情報処理装置。

10

20

30

40

50

## 【請求項 1 4】

情報処理装置が行う情報処理方法であって、  
 学習データを認識器に入力して出力データを得る処理工程と、  
複数の識別器から一部の識別器を選択する選択工程と、  
 前記出力データを前記選択された識別器に入力することにより得られる前記学習データの  
 前記選択された識別器による識別結果の誤差を判定する判定工程と、  
前記識別結果の誤差に基づいて前記選択された識別器の学習処理を行う第 2 の学習工程  
 と、  
前記識別結果の誤差を前記認識器に逆伝播することにより前記認識器の学習処理を行う  
第 1 の学習工程と、  
を有し、  
前記認識器は特徴ベクトルを前記出力データとして出力し、  
前記複数の識別器のそれぞれはハッシュ関数であって、前記選択された識別器には共通  
する前記特徴ベクトルが入力されることを特徴とする、情報処理方法。

10

## 【請求項 1 5】

コンピュータを、請求項 1 乃至 1 3 の何れか 1 項に記載の情報処理装置が備える各手段と  
 して機能させるためのプログラム。

## 【発明の詳細な説明】

## 【技術分野】

## 【0 0 0 1】

20

本発明は、情報処理装置、情報処理方法及びプログラムに関する。

## 【背景技術】

## 【0 0 0 2】

映像画像等のデータから、例えば物体及びその状態を認識するために、学習データを用い  
 て認識システムの学習を行い、学習後の認識システムを用いて入力データを判定する装置  
 や方法が知られている。このような方法の一つとして、Neural Network (   
 NN ) を用いる方法が知られている。例えば、非特許文献 1 は、NN を用いて画像中に含  
 まれる物体の種類を自動的に認識する方法を開示している。非特許文献 1 では、NN とし  
 て Convolutional Neural Network ( CNN ) が用いられてい  
 る。CNN は、局所的な畳み込み処理を行う Convolutional Layer を  
 持つという特徴がある。また、非特許文献 1、特許文献 1 及び特許文献 2 は、NN を構成  
 するニューロンを確率的に OFF にする処理を行うことで、過学習が抑制され、NN の汎  
 化能力が向上することを示している。

30

## 【0 0 0 3】

また、近年、NN の出力を他の識別器や確率モデル等への入力として用いる技術が盛んに  
 研究・開発されている。例えば、非特許文献 2 は、CNN と、CNN の出力を受け取る L  
 i n e a r S u p p o r t V e c t o r M a c h i n e ( L i n e a r S V M ) と  
 、の学習を同時に行う方法を開示している。非特許文献 2 では、SVM の学習誤差を CN  
 N の学習に利用するという特徴がある。非特許文献 3 は、CNN と、CNN の出力を受け  
 取る S t r u c t u r e d S V M と、の学習を同時に行う方法を開示している。非特許  
 文献 4 は、CNN と、CNN の出力に基づいてクラスタリングを行う分類器と、の学習を  
 同時に行う方法を開示している。

40

## 【先行技術文献】

## 【特許文献】

## 【0 0 0 4】

【文献】米国特許出願公開第 2 0 1 4 / 0 1 8 0 9 8 6 号明細書  
 国際公開第 2 0 1 4 / 1 0 5 8 6 6 号

## 【非特許文献】

## 【0 0 0 5】

【文献】A. Krizhevsky et al. "ImageNet Classification with Deep Convolutional

50

Neural Networks", Advances in Neural Information Processing Systems 25 (NIPS), 2012.

Y. Tang. "Deep Learning using Linear Support Vector Machines", ICML 2013 Challenges in Representation Learning, 2013.

S. Zhang et al. "Deep neural support vector machines for speech recognition", 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015.

J. Xie et al. "Unsupervised deep embedding for clustering analysis", arXiv:1511.06335, 2015, <http://lib-arxiv-008.serverfarm.cornell.edu/abs/1511.06335>

【発明の概要】

【発明が解決しようとする課題】

【0006】

より高精度で入力データの認識を行うために、認識能力がさらに向上した認識システムが望まれる。

【0007】

本発明は、認識システムのデータ認識精度を向上させることを目的とする。

【課題を解決するための手段】

【0008】

本発明の目的を達成するために、例えば、本発明の情報処理装置は以下の構成を備える。  
すなわち、

学習データを認識器に入力して出力データを得る処理手段と、

複数の識別器から一部の識別器を選択する選択手段と、

前記出力データを前記選択された識別器に入力することにより得られる前記学習データの  
前記選択された識別器による識別結果の誤差を判定する判定手段と、

前記識別結果の誤差に基づいて前記選択された識別器の学習処理を行う第2の学習手段  
と、

前記識別結果の誤差を前記認識器に逆伝播することにより前記認識器の学習処理を行う  
第1の学習手段と、

を備え、

前記認識器は特徴ベクトルを前記出力データとして出力し、

前記複数の識別器のそれぞれはハッシュ関数であって、前記選択された識別器には共通  
する前記特徴ベクトルが入力されることを特徴とする。

【発明の効果】

【0009】

認識システムのデータ認識精度を向上させる。

【図面の簡単な説明】

【0010】

【図1】実施形態1に係る異常検知システムの構成例を示す図。

【図2】実施形態1に係る異常検知システムの動作例を示す図。

【図3】実施形態1に係る学習装置10の動作例を示す図。

【図4】実施形態1に係る認識装置20の動作例を示す図。

【図5】実施形態1に係るNN学習部12の構成例を示す図。

【図6】実施形態1に係る識別器学習部13の構成例を示す図。

【図7】実施形態2に係るハッシュ関数の超平面を説明する図。

【図8】各実施形態で用いられる映像の例を示す図。

【図9】実施形態1に係るNNとハッシュ関数の組み合わせを示す図。

【図10】ハッシュ関数のdropout動作を説明する図。

【図11】実施形態1に係るNN認識部22の構成例を示す図。

【図12】実施形態1に係る識別器23の構成例を示す図。

【図13】実施形態3に係る識別器学習部13の構成例を示す図。

10

20

30

40

50

## 【発明を実施するための形態】

## 【0011】

以下、本発明の実施形態について図面に基づいて説明する。なお、以下に説明する実施形態は、本発明を具体的に実施した場合の一例を示すもので、特許請求の範囲に記載の構成の具体的な実施例の1つである。

## 【0012】

以下の実施形態によれば、検出対象データに対して識別処理を行うことにより、認識対象を認識することができる。例えば、画像データを検出対象データとする場合、画像中の物体認識や、画像の異常認識等を行うことができる。以下の実施形態においては、映像中の物体に異常があるか否かを検出する応用例について主に説明するが、本発明はこのような構成に限定されない。また、検出対象データが1枚の画像である必要はなく、複数の連続する画像がニューラルネットワーク(NN)に入力されてもよい。また、検出対象データは画像には限定されず、例えば音声等であってもよい。

10

## 【0013】

いくつかの実施形態では、認識精度を向上させるために、NNと組み合わせて複数の識別器が用いられる。そして、このような構成を実現するために、複数の識別器及びNNの学習を行う方法が示される。とりわけ、いくつかの実施形態においては、複数の識別器の誤差に基づいてNNの学習が行われる。また、いくつかの実施形態においては、複数の識別器の誤差に基づいて識別器の学習が行われる。また、いくつかの実施形態においては、複数の識別器の類似性を考慮して識別器の学習を行う方法が示される。さらに、いくつかの実施形態では、認識精度を向上させるために、`dropout`と称する学習方法が用いられる。しかしながら、これらの構成の全てを採用することは必須ではないことに注意されたい。

20

## 【0014】

なお、以下の実施形態で説明する異常検知システム1が備える各部は、専用のハードウェアにより実現することもできるし、コンピュータにより実現することもできる。例えば、データ読込部11、NN学習部12、識別器学習部13、データ読込部21、NN認識部22、及び識別部23のうち少なくとも1つをコンピュータにより実現することができる。これらの機能を実現するには、各機能構成をプログラムにより表現し、コンピュータに読み込ませればよい。すなわち、メモリ及びプロセッサを備えるコンピュータにおいて、メモリに格納されたプログラムに従ってプロセッサが処理を行うことにより、これらの機能を実現することができる。また、以下では学習装置10、認識装置20、及び端末装置30は別個の装置であるものとして説明するが、これらの装置が1又は2の装置に統合されていてもよい。さらに、図1では辞書記憶部M1、M2及びデータ記憶部D1、D2が学習装置10及び認識装置20の外部に存在するが、これらは学習装置10又は認識装置20に含まれてもよい。

30

## 【0015】

## [実施形態1]

本発明の1つの実施形態である実施形態1では、検出対象データを入力データとしてニューラルネットワーク(NN)に入力して出力データを得て、この出力データを識別器に入力することにより入力データの識別結果を得ることができる。そして、このような識別を行うために、学習データを用いて、NNと、NNの出力を受け取る識別器と、について同時に学習処理が行われる。本実施形態では、識別のために複数の識別器が用いられ、より具体的には複数のハッシュ関数が用いられる。

40

## 【0016】

本実施形態では、具体例としては、本発明に係る情報処理装置を異常検知システムに適用する場合の構成例を示す。特に本実施形態では、異常検知システムとして、監視カメラによって撮像された映像中の異常を検出する機能を備えたシステムが用いられる。以降では本実施形態について、図面を参照して説明する。

## 【0017】

50

本実施形態に係る異常検知システムは、監視対象をカメラ等の撮像装置で撮影して得られた映像データに基づいて、監視対象に異常があるか否かを判定する。そして、異常があると判定された場合には警告が行われる。例えば、警備室等の監視センタに常駐する監視者に対して警告を行うことができる。この監視対象には、例えば、一般家庭の屋内及び屋外、又は病院及び駅等の公共施設、等が含まれる。

【 0 0 1 8 】

[ 全体構成 ]

図 1 は、実施形態 1 に係る異常検知システムの構成の一例を示す概略ブロック図である。まず、図 1 に示す異常検知システム 1 の構成に関する説明を行う。図 1 の異常検知システム 1 は、学習装置 1 0 と、認識装置 2 0 と、端末装置 3 0 と、を備える。これらの装置内の各部、及びこれらの装置間は、電子回路を介して接続されていても、外部記憶装置を介して接続されていても、またネットワークを介して接続されていてもよい。このネットワークとしては、例えば携帯電話回線網又はインターネット等を利用することができる。

10

【 0 0 1 9 】

本実施形態において、異常検知システム 1 の動作は 2 つのフェーズに分けることができる。1 つ目は学習フェーズであり、2 つ目は認識フェーズである。

【 0 0 2 0 】

学習フェーズは、図 2 のステップ S 2 0 1 に相当する。ステップ S 2 0 1 で、学習装置 1 0 は、学習データを入力データとして NN 及び識別器の学習を行う。

【 0 0 2 1 】

20

認識フェーズは、図 2 のステップ S 2 0 2 及び S 2 0 3 に相当する。ステップ S 2 0 2 で、認識装置 2 0 は、監視対象が写っている映像データ（以下、検出対象データと呼ぶ）を入力データとして、NN 及び識別器を用いた認識・識別を行い、監視対象に異常があるか否かを判定する。ステップ S 2 0 3 で、端末装置 3 0 は、異常の判定結果に基づいて警告処理を行う。

【 0 0 2 2 】

以降では、異常検知システム 1 の動作の概略について、学習フェーズの動作と認識フェーズの動作とに分けて説明を行う。各処理部の詳細な構成及び動作に関しては後述する。

【 0 0 2 3 】

学習フェーズの動作は、学習装置 1 0 により行われる。学習装置 1 0 は、データ読込部 1 1 と、NN 学習部 1 2 と、識別器学習部 1 3 と、を備える。学習装置 1 0 の動作の概要に関して、図 3 に基づいて説明を行う。

30

【 0 0 2 4 】

ステップ S 7 0 0 で、データ読込部 1 1 は、NN 及び識別器を学習するための学習対象データをデータ記憶部 D 1 から読み込み、NN 学習部 1 2 に送る。ステップ S 7 0 1 ~ S 7 1 7 で、NN 学習部 1 2 及び識別器学習部 1 3 は、読み込んだ学習対象データを用いて学習を行う。このとき、NN 学習部 1 2 は、学習対象データに基づいて生成された学習データを入力データとして NN を用いた認識を行い、識別器学習部 1 3 は、NN による認識結果（NN の出力データ）を入力データとして学習を行う。また、このときの識別器学習部 1 3 による学習誤差は、NN 学習部 1 2 に入力され、NN 学習部 1 2 の学習処理に用いられる。ステップ S 7 1 8 で、NN 学習部 1 2 は、学習済みの NN の辞書を辞書記憶部 M 1 に保存する。また、識別器学習部 1 3 は、学習済みの識別器の辞書を辞書記憶部 M 2 に保存する。

40

【 0 0 2 5 】

認識フェーズの動作は、認識装置 2 0 により行われる。認識装置 2 0 は、データ読込部 2 1 と、NN 認識部 2 2 と、識別部 2 3 と、を備える。このうち、NN 認識部 2 2 は検出対象データを NN に入力して出力データを得る。また、識別部 2 3 は NN からの出力データを識別器に入力することにより検出対象データの識別結果を得る。認識装置 2 0 の動作の概要に関して、図 4 に基づいて簡単な説明を行う。

【 0 0 2 6 】

50

ステップ S 1 3 0 0 で、データ読込部 2 1 は、記憶部 D 2 から検出対象データの読み込みを行い、NN認識部 2 2 に送る。ステップ S 1 3 0 1 ~ S 1 3 0 6 で、NN認識部 2 2 は、検出対象データに対する認識処理を行い、その認識結果 (NNの出力データ) を識別部 2 3 に送る。ステップ S 1 3 0 7 ~ S 1 3 0 9 で、識別部 2 3 は、NNによる認識結果を入力データとして、識別処理及び異常判定処理を行い、異常判定結果を端末装置 3 0 に送る。

#### 【 0 0 2 7 】

次に端末装置 3 0 の動作の概要に関して説明を行う。端末装置 3 0 は、表示部 3 1 を備える。端末装置 3 0 としては、例えば P C ( P e r s o n a l C o m p u t e r ) のディスプレイ、タブレット P C、スマートフォン、又はフィーチャーフォン等を利用することができる。

10

#### 【 0 0 2 8 】

表示部 3 1 は、ステップ S 4 0 3 で、検出対象が異常であることを示す異常判定結果が送られた場合、それを受け取り、警告処理を行う。このとき、端末装置 3 0 の備える機能に応じて、どのような警告処理を行ってもよい。例えば、表示部 3 1 は、表示又は音声等を用いた警告を行うことができる。一例として、端末装置 3 0 がサイレンを備える場合は、表示部 3 1 は、端末装置 3 0 に警告音を発させることができる。また、表示部 3 1 は、端末装置 3 0 が備えるディスプレイに点滅表示を行わせることもできる。さらに、端末装置 3 0 が映像確認用のディスプレイを備える場合、監視映像上で異常が検出された箇所を強調表示してもよい。このように異常が検出された箇所を強調表示するには、監視映像上における異常箇所を特定する必要がある。そのためには、例えば NN に入力される入力データに映像中の位置を示す画面座標情報を付与しておき、異常判定結果に応じて、この画面座標情報を利用して警告表示を行ってもよい。

20

#### 【 0 0 2 9 】

##### [ 学習フェーズ ]

以降では、各装置が備える処理部の構成及び動作について、詳細な説明を行う。まずは、NN学習部 1 2 及び識別器学習部 1 3 の詳細な説明を行う。

#### 【 0 0 3 0 】

NN学習部 1 2 の構成に関して、図 5 に基づいて説明を行う。NN学習部 1 2 は、小領域抽出部 1 2 1 と、前処理部 1 2 2 と、NN初期化部 1 2 3 と、学習データ選択部 1 2 4 と、NN認識実行部 1 2 5 と、NN更新部 1 2 6 と、NN辞書保存部 1 2 7 と、を備える。

30

#### 【 0 0 3 1 】

また、識別器学習部 1 3 の構成に関して、図 6 に基づいて説明を行う。識別器学習部 1 3 は、ハッシュ関数生成部 1 3 1 と、ハッシュ関数選択部 1 3 2 と、ハッシュ関数更新部 1 3 3 と、を備える。なお本実施形態の冒頭で述べたように、本実施形態では識別器の具体例としてハッシュ関数を用いた識別器を用いる。ハッシュ関数の詳細は後述する。

#### 【 0 0 3 2 】

次に、NN学習部 1 2 および識別器学習部 1 3 の動作に関して、図 3 に基づいて詳細な説明を行う。ステップ S 7 0 1 で、小領域抽出部 1 2 1 は、データ読込部 1 1 からデータを受け取る。本実施例において、このデータは、カメラ等の撮像装置で撮影して得られた映像データを含むものとする。このデータには、監視対象を撮影して得られた映像データを含めることができる。

40

#### 【 0 0 3 3 】

小領域抽出部 1 2 1 は、映像データの各フレームに写っているオブジェクトを含む小領域画像を抽出する。ここで、ある映像データに含まれる 1 フレームの画像の例を図 8 に示す。図 8 は、フレーム画像の例として、ある交差点において撮像された画像 8 0 1 を示す。画像 8 0 1 には、撮像されたオブジェクト 8 0 2 ~ 8 0 5 と、信号 8 1 0 とが写っている。バウンディングボックス (Bounding Box) 8 0 6 ~ 8 0 9 は、オブジェクト 8 0 2 ~ 8 0 5 のそれぞれを囲む矩形である。小領域抽出部 1 2 1 は、バウンディングボックス 8 0 6 ~ 8 0 9 により囲まれた部分画像をそれぞれ小領域画像として抽出する。

50

## 【 0 0 3 4 】

なお、バウンディングボックスに基づいて小領域画像を抽出する方法は一例にすぎず、小領域抽出部 1 2 1 が用いる方法は限定されない。このような小領域画像を抽出するための方法は複数存在し、例えば背景差分法、物体検出・追尾法、又は領域分割法等を用いることができる。例えば、小領域抽出部 1 2 1 は、後述する背景差分法を用いることにより、オブジェクト 8 0 2 ~ 8 0 5 の輪郭を抽出し、輪郭によって囲まれる部分画像を小領域画像として抽出することもできる。また、監視対象のオブジェクトがあらかじめ既知である場合は、物体検出・追尾法を用いることができる。物体検出・追尾法によれば、監視対象となっているオブジェクトを検出・追尾することができる。物体検出・追尾法としては、例えば H. Grabner, M. Grabner and H. Bischof, "Real-Time Tracking via On-line Boosting", Proceedings of the British Machine Conference, pages 6.1-6.10. B MVA Press, September 2006. に記載の方法を用いることができる。

10

## 【 0 0 3 5 】

さらに小領域抽出部 1 2 1 は、映像データにあらかじめ付与された教師情報を利用して、小領域画像に対して教師情報を付与する。教師情報は、対象（すなわちオブジェクト）をどのように分類すべきかを示すラベルである。本実施形態においては、異常検知システム 1 を使用又は導入するユーザが、ラベルの種類、及び学習データに付与する教師データを予め決定しておく。一般に、どのような対象にどのようなラベルを付与するかは、問題（例えば異常検知システムの用途等）に依存する。

## 【 0 0 3 6 】

より具体的には、ユーザは、映像データを構成する画像上で指定した領域に対して、ラベルを付与する。例えば、ユーザは、フレーム画像上で歩行者 8 0 3 を含む領域を指定し、この領域に対して歩行者を示すラベルを付与することができる。そして、小領域抽出部 1 2 1 は、歩行者を示すラベルが付与された領域から小領域画像を抽出した場合、この小領域画像に対しては歩行者を示す教師情報を付与することができる。小領域画像が、互いに異なるラベルが付与された複数の領域が抽出された場合、小領域画像に占める割合が最も大きい領域に付与されているラベルを、教師情報として小領域画像に付与することができる。

20

## 【 0 0 3 7 】

もっとも、上記の方法で小領域画像に教師情報を付与することは必要ではない。例えば、あらかじめフレーム画像から小領域画像を抽出しておき、それぞれの小領域画像に対してユーザが教師情報を付与してもよい。また、小領域抽出部 1 2 1 が抽出した小領域画像のそれぞれに対して、ユーザが教師情報を付与してもよい。以上の処理により、小領域抽出部 1 2 1 は、映像データを構成するフレーム画像から、教師情報が付された小領域画像を抽出する。最後に、小領域抽出部 1 2 1 は、抽出した小領域画像を学習画像として教師情報とともに前処理部 1 2 2 に送る。

30

## 【 0 0 3 8 】

なお、説明を簡略化するために、ステップ S 7 0 1 で小領域抽出部 1 2 1 は、使用するすべてのデータを一度に読み込み、それぞれのフレーム画像から小領域画像を抽出し、すべての小領域画像を前処理部 1 2 2 に送るものとする。しかしながら、学習データのデータ量が非常に大きい場合、すべてのデータを一度に通信することは難しいため、一部ずつ学習データを通信してもよい。以降の説明では、特に注意書きを記載する場合を除いては、すべての学習データについて一度に処理及び通信を行うものとする。

40

## 【 0 0 3 9 】

ステップ S 7 0 2 で、前処理部 1 2 2 は、入力データの増しを行う。データの増し処理は NN の学習の前処理として一般的に行われており、学習データの見掛け上のバリエーションを増大させる効果がある。ここで、どのような増し処理を行うかは、問題に依存して決定することができる。例えば、画像データに対して一般的に行われている増し処理として、入力画像を水平方向に左右反転させる画像処理が挙げられる。この方法は、入力画像が水平方向に反転しても、教師情報が不変であるときに用いることができる。一方

50



、図 8 のように交差点の映像から車の逆走を検知したい場合には、車の左右の向きが重要であることから、別の水増し処理を採用しなければならない。本実施形態では、異常検知の対象は映像中の人体であり、それぞれの人体を示す小領域画像に対して付される教師情報は、画像を左右反転させても不変であるものとする。したがって、ステップ S 7 0 2 で、前処理部 1 2 2 は、入力された学習画像のそれぞれに対して左右反転処理を行い、処理前の画像と、処理後の画像との双方を、以降で学習データ（学習画像）として扱う。

#### 【 0 0 4 0 】

一般に、前処理部 1 2 2 は、小領域画像を表す画像データ及びこの画像の種類を示す教師情報を取得し、画像処理により 1 つの画像から複数の画像を生成することができる。この際、前処理部 1 2 2 は、画像処理により画像の種類が変化しない画像を選択し、選択された画像に対して画像処理を行う。そして、前処理部 1 2 2 は、画像処理前の選択された画像と、画像処理後の選択された画像と、を学習データとして出力する。

10

#### 【 0 0 4 1 】

ステップ S 7 0 3 で、前処理部 1 2 2 は、学習画像全体の平均画像を算出し、すべての学習画像から平均画像を減算する処理を行う。なお、平均画像とは、各画素が画素値として、各学習画像の対応する画素の平均画素値を有している画像のことを指す。また、学習画像から平均画像を減算する処理とは、学習画像の各画素の画素値から、平均画像の対応する画素の画素値を減算する処理のことを指す。このように学習画像から平均画像を引くことは、学習データに対して一種のセンタリング処理を行うことに対応する。このような処理によれば、学習画像ごとに画素値が大きく異なる場合であっても、NN の学習処理を収束させやすくすることができる。このようにセンタリング処理済みの学習画像を用いる本実施形態においては、後述する NN の初期化処理において、結合重みの初期値として平均 0 のガウス分布を用いることができる。

20

#### 【 0 0 4 2 】

なお、本実施形態においては、平均画像を算出するために、前処理部 1 2 2 は、学習画像の解像度（縦と横のサイズ）を一致させる前処理をさらに行う。具体的には、後述する NN の初期化を行う際に用いる NN のネットワーク構造を参照して、その入力層のサイズに合わせる形で、前処理部 1 2 2 は、それぞれの学習画像に対して拡大又は縮小処理を行うことができる（ネットワーク構造については後述する）。

#### 【 0 0 4 3 】

以上の前処理部 1 2 2 による処理により、学習画像と付与された教師データとのセットが複数用意され、これらは学習データ選択部 1 2 4 へと入力される。そして、学習画像と付与された教師データとのセットを用いて、NN の学習が行われる。もっとも、目標とする検出精度に応じて、前処理部 1 2 2 による水増し処理及び平均画像を減算する処理の一方又は双方を省略することも可能である。

30

#### 【 0 0 4 4 】

ステップ S 7 0 4 で、NN 初期化部 1 2 3 は、NN のパラメータ（結合重みおよびバイアス項）を初期化する。本実施形態では、NN のネットワーク構造は予め決定されており、NN のネットワーク構造を参照して初期化が行われる。本実施形態において用いられるネットワーク構造及び初期パラメータは特に限定されない。例えば非特許文献 1 と同じ手法を用いてよいし、独自に定義したネットワーク構造及び初期パラメータを用いてもよい。

40

#### 【 0 0 4 5 】

図 9 は、本実施形態で用いられる NN 9 2 0 と、識別器 9 2 1 と、の一例を模式的に示す。なお、識別器 9 2 1 に関しては、その詳細を後述する。NN 9 2 0 は、本実施形態で用いられる NN のネットワーク構造の一例を示す。図 9 には、ネットワークが、入力層 9 0 1、convolution 1 層 9 0 2、pooling 1 層 9 0 3、convolution 2 層 9 0 4、pooling 2 層 9 0 5、Inner product 層 9 0 6、及び出力層 9 0 7 を含むことが示されている。また、2 つの階層間の処理方法として、convolution 処理 9 1 0、pooling 処理 9 1 1、及び Inner Product 処理 9 1 2 が設定されていることが示されている。それぞれの処理の具体的な内容は

50

非特許文献 1 と同様であるため、ここでは省略する。簡単に説明すると、convolution 処理 910 では畳み込みフィルタを用いたデータ処理が行われ、pooling 処理 911 では、例えば max pooling であれば局所的な最大値を出力する処理が行われる。また、Inner Product 処理 912 では、内積処理が行われる。

#### 【0046】

また、図 9 からは、convolution 層及び pooling 層には複数の特徴マップが存在し、入力層における画像上のピクセルに対応する位置には、複数のニューロンが存在することがわかる。例えば、学習画像が RGB 形式である場合、RGB チャンネルのそれぞれに対応する 3 つのニューロンが存在しうる。また、学習画像が、撮像映像における動き情報を持つ Optical Flow 画像である場合、画像の横軸方向の動きと縦軸方向の動きとをそれぞれ表現する 2 つのニューロンが存在しうる。また、複数の画像を入力として用いる場合は、入力される画像の数に従って入力層のニューロンを増やすことができる。本実施形態では、標準的な RGB 画像を学習画像として用いる例について説明する。

#### 【0047】

NN 初期化部 123 が NN のパラメータを初期化する方法は特に限定されない。初期化方法としては様々な方法が知られており、例えば、用意されたデータを用いて初期化する場合と、データを用いずに初期化する場合と、がある。本実施形態では、一例として、データを用いずに NN のパラメータを初期化する方法について説明する。具体的には、本実施形態では、NN 初期化部 123 は、平均 0、分散 1 の正規分布からランダムにサンプリングした値を用いて重みパラメータを初期化し、バイアス項パラメータをすべて 0 で初期化するものとする。なお、パラメータに関する詳細は非特許文献 1 を参照のこと。なお、データを用いて初期化する場合の一例についても後述する。最後に NN 初期化部 123 は、初期化により得られた辞書を、NN 認識実行部 125 に送る。

#### 【0048】

ステップ S705 で、ハッシュ関数生成部 131 は、識別器の初期化を行う。具体的には、ハッシュ関数生成部 131 は、ハッシュ関数を生成する。本実施形態においては、一個一個のハッシュ関数を、それぞれ線形識別器として扱う。生成されるハッシュ関数の数は特に限定されないが、ハッシュ関数の数を増やすことにより、近似近傍探索の精度を向上させることができる。本実施形態では、例として、ハッシュ関数生成部 131 は 100 個のハッシュ関数を生成するものとする。最後にハッシュ関数生成部 131 は、生成したハッシュ関数をハッシュ関数更新部 133 に送る。

#### 【0049】

本実施形態において、識別器 921 は、入力データを NN に入力して得られる出力データに基づいて、NN がある分類に属するか属さないかを判定する One-class 識別器である。具体例として、識別器 921 は、図 9 に示すように、NN の出力（ここでは出力層からの出力値）に基づいて、ハッシュ関数生成部 131 により生成されたハッシュ関数を用いた Hashing 処理 913 によりハッシュ値 908 を求める。ハッシュ値 908 は、それぞれのハッシュ関数を用いて得られた識別結果を表すバイナリ値で構成されるベクトルであり、詳細は後述する。

#### 【0050】

Hashing 処理 913 で用いられるハッシュ関数は特に限定されない。本実施形態では、一例として、Hachiya, H., Masakazu M., "NSH: Normality Sensitive Hashing for Anomaly Detection", 2013 IEEE International Conference on Computer Vision Workshops (ICCVW) (以下、Hachiya と呼ぶ) 及び特開 2014-203289 号公報に示されるハッシュ関数を用いることとする。これらの文献に示されるハッシュ関数は、Normality Sensitive Hashing (NSH) と呼ばれ、正常学習ベース異常検知技術の構成の一例において用いられる。正常学習ベース異常検知技術は、正常データを含むが異常データを含まない学習データを用いて異常を検知するための検出器を学習するという特徴を有している。本実施形態では、説明を具体的に

10

20

30

40

50

ため、N S Hを用いて正常学習を行う例について説明する。なお、正常データと異常データとを用いてハッシュ関数を学習する場合は、特開 2 0 1 4 - 2 0 3 2 8 9 号公報に示されるような異常データに対応したハッシュ関数の目的関数を用いればよい。ここでは詳細な説明は行わない。なお、正常学習は、一般には One - c l a s s 学習の一種として知られている。

【 0 0 5 1 】

ハッシュ関数の具体的な説明は H a c h i y a に示されているため、ここでは詳細を省くが、本実施形態においては以下の超平面の式 ( 1 ) によりハッシュ関数が定義されるものとする。このハッシュ関数を用いると、特徴ベクトルが、超平面の式 ( 1 ) を境界として、法線ベクトルの方向側にあるか、その反対側にあるかを表すバイナリ値が得られる。本実施形態においては、超平面の式 ( 1 ) を境界として、法線ベクトルの方向側にある特徴ベクトル ( 入力データ ) のハッシュ値を「 0 」、その反対側にある特徴ベクトル ( 入力データ ) のハッシュ値を「 1 」というバイナリ値で表すものとする。本実施形態においては、入力データが正常である場合にハッシュ関数がバイナリ値「 0 」を返し、入力データが異常である場合にハッシュ関数がバイナリ値「 1 」を返すように、複数のハッシュ関数それぞれの学習が行われる。

【数 1】

$$\mathbf{w}_m^T \mathbf{x} - b_m = 0 \quad \cdots \cdots (1)$$

式 ( 1 ) において、 $T$  はベクトルの転置、 $x$  は入力データ、 $w$  は超平面の法線ベクトル又は重みベクトル、 $b$  はバイアス項、 $m$  はハッシュ関数に関するインデックスを表す。上記の通り、本実施形態においては 1 0 0 個のハッシュ関数が用いられ、 $m$  は 1 ~ 1 0 0 のいずれかの値をとる。また、入力データ  $x$  は、N N の出力 ( すなわち出力層 9 0 7 の出力値 ) である。

【 0 0 5 2 】

ここで、式 ( 1 ) の左辺を以下の式 ( 2 ) のように  $z$  で表す。

【数 2】

$$z = \mathbf{w}_m^T \mathbf{x} - b_m \quad \cdots \cdots (2)$$

【 0 0 5 3 】

ある入力データ  $x$  が法線ベクトル  $w$  の方向側にある場合、 $z$  は正の値をとり、反対側にある場合は、 $z$  は負の値を取る。この性質を利用することにより、本実施形態では、H a c h i y a に示すように、次の評価式 ( 3 ) を用いてそれぞれのハッシュ関数を評価する。

【数 3】

$$E = \frac{1}{N} \sum_{i=1}^N L(\mathbf{w}^T \mathbf{x}_i - b) - \lambda b \quad \cdots \cdots (3)$$

式 ( 3 ) において、 $N$  は正常データの数 ( 後述する M i n i b a t c h のサイズ )、 $\lambda$  はバイアス項の重みパラメータである。また、 $L(\mathbf{w}^T \mathbf{x}_i - b)$  は、入力されたある正常データ  $\mathbf{x}_i$  を異常と判定した場合における誤差を出力する関数であり、例えば次の式 ( 4 ) を用いて定義することができる。

【数 4】

$$L(z) \equiv \begin{cases} 0 & (z \geq 0) \\ z^2 & (z < 0) \end{cases} \quad \cdots \cdots (4)$$

## 【0054】

すなわち  $L(w^T x_i - b)$  はハッシュ関数による学習データに対する識別結果の誤差を表しており、式(3)は、入力された複数の正常データに対する、あるハッシュ関数のロスを示している。そして、式(3)は、値が小さいほどよいと考えることができる。なお、上記の例において、 $w$  及び  $b$  はハッシュ関数のパラメータであり、 $\lambda$  はハイパーパラメータである。本実施形態において、 $\lambda$  はあらかじめ手で与えるものとし(例えば  $Hachiy a$  と同じ値を用いてもよい)、 $w$  及び  $b$  はそれぞれランダムに初期化するものとする。

## 【0055】

ステップS706で、学習データ選択部124は、変数  $Iteration$  を0に初期化する。 $Iteration$  は、最適化処理の繰り返し回数を示し、その詳細は後述する。ステップS707で、学習データ選択部124は、学習データの部分集合群 ( $Minibatch$ ) を作成する。 $Minibatch$  は、NNのパラメータを  $Stochastic Gradient Descent$  (SGD、確率的勾配降下法) によって繰り返し最適化する際に用いる、学習データの集合である。詳細は後述するが、本実施形態では、NNのパラメータを最適化する方法の例として、SGDを用いる。なお、 $Minibatch$  のサイズは限定されないが、本実施形態では、同じラベルを示す教師情報を持つ50個の学習画像をひとまとめにしたものを  $Minibatch$  として用いる。 $Minibatch$  は、例えば、全ての学習画像が複数の  $Minibatch$  のいずれかに含まれるように、かつ1つの学習画像が2以上の  $Minibatch$  に含まれないように、作成することができる。

## 【0056】

説明を簡略化するために、本実施形態で用いる学習データは、その教師情報が示すラベルの種類に関して均一にバランスがとれているものとする。すなわち、ステップS707で学習データ選択部124は、複数のラベルのそれぞれについて、同数の  $Minibatch$  を生成するものとする。学習データがラベルの種類に関して均一でない場合は、例えば、均一になるように学習データをサンプリングすることができる。

## 【0057】

以上の処理が、NNと識別器とを学習する(すなわち、パラメータを最適化する)ための準備に相当する。すなわち、以上の処理により、学習対象データから学習データ(学習画像)が得られ、また初期状態のNN及びハッシュ関数が得られる。ステップS708以降では、学習データを用いて実際にNNと識別器の学習が行われ、以下ではこの動作について説明する。本実施形態では、NNの学習を行うための方法として、誤差逆伝播法とSGDとを組み合わせた方法を用いる。この方法は非特許文献1において詳細に説明されているため、ここでは詳しく説明することはしない。簡単に説明すると、確率的に  $Minibatch$  を選択し、その勾配を計算して、勾配方向にパラメータを更新する手順を繰り返すことで、学習が行われる。

## 【0058】

ステップS708で、学習データ選択部124は、 $Minibatch$  集合からランダムに1つの  $Minibatch$  を選択する。このとき、学習データ選択部124は、 $Iteration$  の値が同じである間にまだ選択されていない  $Minibatch$  を選択する。本実施形態では、ステップS707で学習データ選択部124はあらかじめすべての  $Minibatch$  が作成した。しかしながら、ステップS716において  $Iteration$  値がインクリメントされる度に、学習データ選択部124は新たな  $Minibatch$  群を作成することにより  $Minibatch$  を初期化してもよい。このような構成によれば、それぞれの  $Minibatch$  が含む学習データの組み合わせが繰り返しごとに変化するため、過学習のリスクが低減することができる。

## 【0059】

ステップS709で、NN認識実行部125は、学習データをNNに入力して出力データを得る。具体的には、NN認識実行部125は、 $Minibatch$  を入力データとしてNNの出力を得る。例えば、図9に示すNNを用いる場合、入力層901から出力層90

10

20

30

40

50

7まで階層的に順伝播するようにNNを用いたデータ処理が行われる。ステップS710で、NN認識実行部125は、識別器学習部13にNNの出力（出力層907の値）を送る。

【0060】

ステップS711で、ハッシュ関数選択部132は、複数の識別器から一部の識別器を選択する。本実施形態において、ハッシュ関数選択部132は、ハッシュ関数生成部131が予め生成したハッシュ関数のうち、更新するハッシュ関数をMini batchごとを選択する。そして、ハッシュ関数選択部132により選択されたハッシュ関数に対してのみ、ハッシュ関数更新部133によるパラメータ更新処理と、NNに対して逆伝播される識別器誤差の計算と、が行われる。このように使用するハッシュ関数を選択する処理を以下ではハッシュ関数のdropout処理と呼ぶ。この処理は、非特許文献1で示されるNNに対するdropout処理とは異なるものの、複雑な目的関数(3)に対するハッシュ関数の過学習を抑制する効果を有する。また、異なるハッシュ関数のセットについての識別器誤差が順次NNに逆伝播されることも、最終的な認識結果の精度の向上につながると考えられる。式(2)は全てのハッシュ関数に関する表現であるが、ステップS711で選択されたハッシュ関数に対しては、式(2)を以下の式(5)として表すことができる。

【数5】

$$z = \{ \mathbf{w}_d^T \mathbf{x} - b_d \mid d \in D \} \quad \cdots \cdots (5)$$

式(5)において、Dは選択されたハッシュ関数の集合であり、dはインデックスであり、 $d \in D$ はdがDの要素であることを示す。

【0061】

ハッシュ関数選択部132は、すべてのハッシュ関数の集合から、選択されたハッシュ関数の集合Dを、確率的に選択することができる。例えば、ハッシュ関数選択部132は、各々のハッシュ関数に関して、50%の確率で選択するか否かを決定してもよい。このとき、選択しなかったハッシュ関数は、dropoutした(OFFになった)と捉えることができる。すなわち、この処理は、ハッシュ関数に関するdropout処理であると考えることができる。

【0062】

ステップS711の処理を分かりやすく説明するために、図10に、上記で説明したハッシュ関数の構成及び動作の一例を示す。図10においてニューロン1001は、NNのニューロンを示す。また、NNの出力空間1002は、NNの出力空間の模式図（ここでは出力層907の値がなす空間）を示す。ハッシュ関数の超平面1003は、複数のハッシュ関数が超平面としてNNの出力空間1002を分離する様子を示す。Hash dropout1004は、一部のハッシュ関数を選択する（選択されなかったハッシュ関数はdropoutされる）処理を示す。dropoutされたハッシュ値1005は、ハッシュ関数がdropoutされることで、識別結果を表すベクトルからバイナリ値が欠落する様子を示す。dropoutされたハッシュ関数の超平面1006は、ハッシュ関数がdropoutされることで分離超平面が一時的に無くなることを示している。

【0063】

このようなハッシュ関数のdropout処理を、ハッシュ関数選択部132は、ステップS711が実行されるたびに行う。以降では、ハッシュ関数更新部133が、選択されたハッシュ関数の集合Dに対して、勾配の計算及び誤差の計算を行う。すなわち、ハッシュ関数更新部133は、まず学習データをNNに入力して得られた出力データを第1の識別器セットに入力することにより得られる識別結果の誤差を判定する。そして、NN更新部126は、この識別結果の誤差をNNに逆伝播することにより、NNのパラメータを更新する。さらに、ハッシュ関数更新部133は、学習データをパラメータ更新後のNNに入力して得られた出力データを第2の識別器セットに入力することにより得られる識別結

10

20

30

40

50

果の誤差を判定する。そして、NN更新部126は、この識別結果の誤差をNNに逆伝播することにより、NNのパラメータを再度更新する。ハッシュ関数選択部132は、複数の識別器から第1の識別器セットと第2の識別器セットとを選択する。ここで、第1の識別器セットに含まれる識別器の少なくとも1つは、第2の識別器セットには含まれてもよい。

【0064】

ステップS712で、ハッシュ関数更新部133は、NN認識実行部125からの出力データを複数の識別器に入力することにより得られる学習データの識別結果の誤差を判定する。本実施形態ではdropout処理が行われる。このため、ハッシュ関数更新部133は、学習データをNNに入力して得られた出力データを、ハッシュ関数選択部132により選択された一部の識別器に入力することにより得られる識別結果の誤差を判定する。この際、ハッシュ関数選択部132により選択されなかった識別器は、識別結果の誤差の判定のためには用いられない。本実施形態ではNNの学習処理が行われるため、こうして得られた識別結果の誤差は、ステップS714でNNに逆伝播される。

【0065】

また、本実施形態では複数の識別器の学習処理を行うため、ハッシュ関数更新部133はさらに学習データの識別結果の誤差に基づいて複数の識別器の学習処理を行う。具体的には、ハッシュ関数更新部133は、ハッシュ関数選択部132により選択されたハッシュ関数を、NN認識実行部125から受け取ったNNの出力に基づいて更新する。ハッシュ関数の更新方法としては任意の方法を採用することができるが、本実施形態においては勾配法に基づいてハッシュ関数が更新される。例えば、ハッシュ関数のパラメータwの更新式としては式(6)を用いることができ、ハッシュ関数のパラメータbの更新式としては式(7)を用いることができる。

【数6】

$$\mathbf{w}_{new} = \mathbf{w}_{old} + \alpha \Delta_{\mathbf{w}} \quad \cdots \cdots (6)$$

【数7】

$$b_{new} = b_{old} + \beta \Delta_b \quad \cdots \cdots (7)$$

式(6)において、 $\mathbf{w}_{new}$ は更新後のパラメータwを、 $\mathbf{w}_{old}$ は更新前のパラメータwを、 $\alpha$ はwに関する学習率を、 $\Delta_{\mathbf{w}}$ はパラメータwの勾配を表す。また、式(7)において、 $b_{new}$ は更新後のパラメータbを、 $b_{old}$ は更新前のパラメータbを、 $\beta$ はbに関する学習率を、 $\Delta_b$ はパラメータbの勾配を表す。

【0066】

$\mathbf{w}$ は式(3)を偏微分することで求めることができ、具体的には式(8)を用いて求めることができる。

【数8】

$$\Delta_{\mathbf{w}} = -\frac{\partial E}{\partial \mathbf{w}} = -\frac{1}{N} \sum_{i=1}^N \frac{\partial L(\mathbf{w}^T \mathbf{x}_i - b)}{\partial \mathbf{w}} \quad \cdots \cdots (8)$$

【0067】

bも式(3)を偏微分することで求めることができ、具体的には式(9)を用いて求めることができる。

【数9】

10

20

30

40

50

$$\Delta_b = -\frac{\partial E}{\partial b} = -\frac{1}{N} \sum_{i=1}^N \frac{\partial L(\mathbf{w}^T \mathbf{x}_i - b)}{\partial b} + \lambda \quad \dots\dots (9)$$

【 0 0 6 8 】

式 ( 8 ) ( 9 ) において、 $L(\mathbf{w}^T \mathbf{x}_i - b)$  は、既に式 ( 3 ) について説明したように識別器 ( ハッシュ関数 ) による学習データの識別誤差を表す。そして、この  $L(\mathbf{w}^T \mathbf{x}_i - b)$  に基づいて、式 ( 8 ) ( 9 ) を用いて識別器 ( ハッシュ関数 ) の学習処理が行われる。関数  $L$  の勾配を計算する方法は特に限定されない。また、学習率  $\lambda$  及び  $\lambda$  は任意に決めることができ、例えば実験的に調整を行うことにより決定することができる。

10

【 0 0 6 9 】

識別器の学習処理を行うための方法として、ハッシュ関数を勾配法により更新する代わりに、*Hachiya* のようにハッシュ関数を新たに生成する方法を採用してもよい。なお、上記ではハッシュ関数の更新を行う場合の例を示したが、実際には、 $NN$  の更新だけが必要で、ハッシュ関数の更新は必要がない場合があると考えられる。当然ながら、その際はハッシュ関数を更新しなくてもよい。なお、その際には、ハッシュ関数の勾配を計算せず、式 ( 3 ) の値を誤差として  $NN$  に逆伝播してもよい。

【 0 0 7 0 】

ステップ  $S713$  で、ハッシュ関数更新部 133 は、 $NN$  更新部 126 にハッシュ関数の誤差を送る。本実施形態においては、ハッシュ関数更新部 133 は、式 ( 10 ) に従ってハッシュ関数全体の誤差を求め、 $NN$  更新部 126 に送る。

20

【 数 1 0 】

$$\Delta_x = \frac{\partial E}{\partial \mathbf{x}} = \frac{1}{N_D} \frac{1}{N} \sum_{d \in D} \sum_{i=1}^N \frac{\partial L(\mathbf{w}_d^T \mathbf{x}_i - b_d)}{\partial \mathbf{x}_i} \quad \dots\dots (10)$$

式 ( 10 ) において、 $N_D$  はハッシュ関数選択部 132 により選択されたハッシュ関数の数である。式 ( 10 ) の関数  $L$  の勾配を計算する方法は特に限定されない。式 ( 10 ) において、 $L(\mathbf{w}_d^T \mathbf{x}_i - b)$  はそれぞれの識別器 ( ハッシュ関数 ) による学習データの識別誤差を表し、目的関数  $E$  は、選択されたハッシュ関数全体 (  $d \in D$  ) の誤差を表している。このように、ハッシュ関数更新部 133 は、複数のハッシュ関数全体についての識別結果の誤差を判定している。また、この誤差は、複数のハッシュ関数のそれぞれに  $NN$  からの出力データを入力することにより得られる、複数のハッシュ関数のそれぞれによる学習データの識別結果の誤差に基づいて判定されている。特に、本実施形態においては、複数のハッシュ関数のそれぞれによる学習データの識別結果の誤差の線形結合として、複数のハッシュ関数全体についての識別結果の誤差が判定される。

30

【 0 0 7 1 】

ステップ  $S714$  で、 $NN$  更新部 126 は、判定された複数の識別器全体についての識別結果の誤差を  $NN$  に逆伝播する。具体的には、 $NN$  更新部 126 は、ハッシュ関数による識別結果の誤差を  $NN$  に逆伝播することにより  $NN$  の学習処理を行う。具体的には、 $NN$  更新部 126 は、受け取ったハッシュ関数の誤差を用いて、誤差逆伝播法によって  $NN$  のパラメータを更新する。式 ( 10 ) に従って目的関数の勾配を求めて  $NN$  のパラメータを更新することは、選択されたハッシュ関数全体の誤差を逆伝播することに相当する。 $NN$  のパラメータを更新する方法としては、 $NN$  技術における公知の方法を用いることができる。本実施形態で  $NN$  更新部 126 は、非特許文献 1 と同様の方法を用いて  $NN$  のパラメータを更新する。

40

【 0 0 7 2 】

ステップ  $S715$  で、学習データ選択部 124 は、*Iteration* の値が同じである間に、全ての *Minibatch* が学習に利用されたか否かを判定する。全ての *Mini*

50

batchが利用されていない場合、処理はステップS708に戻り、まだ学習に用いられていないMinibatchを入力データとして学習が行われる。全てのMinibatchが学習に利用されている場合、処理はステップS716に移り、学習データ選択部124はIterationをインクリメントする。

#### 【0073】

ステップS717で、学習データ選択部124は、NN及び識別器の学習が完了しているか否かを判定する。具体的には、学習データ選択部124は、所定の学習停止条件が満たされている場合に、NN及び識別器の学習が完了したと判定する。学習停止条件の設定方法は特に限定されず、例えばユーザが設定することもできるし、学習データ選択部124が自動的に設定することもできる。ユーザが学習停止条件を設定する場合の具体例としては、ユーザが経験的に学習停止条件を設定する方法、例えばIteration回数が所定回数に達することを学習停止条件として設定する方法が挙げられる。また、自動的に学習停止条件を設定する場合の具体例としては、学習曲線の勾配を参照する方法が挙げられる。具体例としては、全てのハッシュ関数について式(3)に従って求められる誤差の総和の、Iteration回数に対する勾配が所定値未満となった際に、NN及び識別器の学習が完了していると判定することができる。

#### 【0074】

本実施形態においては、Iteration回数が所定回数に達した時に、学習停止条件が満たされるものとする。所定回数はユーザが経験的に設定することができ、その値は特に限定されず、例えば所定回数を20000回とすることができる。なお、具体的なNNの学習方法としては、例えば非特許文献1に記載されているように、Iteration回数が増えるほどNNの学習率を低下させる方法等が知られている。本実施形態においては、このような方法をさらに採用することもできる。ステップS717で学習停止条件を満たしていないと判定された場合、処理はステップS708に戻る。条件を満たしていると判定された場合、処理はステップS718に進む。

#### 【0075】

最後に、ステップS718で、NN辞書保存部127は、辞書記憶部M1に学習済みのNNのパラメータ(NNの辞書)を保存する。また、ハッシュ関数更新部133は、辞書記憶部M2に学習済みのハッシュ関数のパラメータ(識別器の辞書)を保存する。ハッシュ関数更新部133は、さらに、学習済みのハッシュ関数に基づいて分離される正常データの情報(正常情報)を保存することができる。ここで、正常データの情報とは、各々のハッシュ関数によって識別される正常データ(学習データ)の数を示す。例えば、あるハッシュ関数についての正常データの情報は、ハッシュ値が「0」となる学習データの数でありうる。この正常データの情報は、正常な度合を示す指標となる。例えば、この正常データの情報は、それぞれのハッシュ関数により得られる入力データのハッシュ値が「0」である場合に、入力データが正常である確からしさを示す。なお、正常データの情報を作成する具体的な方法としては、例えば特開2014-203289号公報に記載されている方法を採用することができる。

#### 【0076】

##### [認識フェーズ]

次に、NN認識部22及び識別部23の詳細な説明を行う。まず、NN認識部22の構成を、図11を参照して説明する。NN認識部22は、小領域抽出部121と、前処理部122と、辞書読込部221と、NN認識実行部125と、を備える。小領域抽出部121、前処理部122、及びNN認識実行部125は、NN学習部12が備える処理部と同様の機能を持つ。すなわち、小領域抽出部121は検出対象の映像データ(画像データ)から小領域画像を抽出し、前処理部122は抽出された小領域画像に対して前処理を行う。そして、NN認識実行部125は前処理された小領域画像データを入力データとしてNNに入力して出力を得る。また、辞書読込部221は、辞書記憶部M1からNNの辞書を読み込み、パラメータをNN認識実行部125が用いるNNに設定する。

#### 【0077】



次に、識別部 2 3 の構成を、図 1 2 を参照して説明する。識別部 2 3 は、辞書読込部 2 3 1 と、識別実行部 2 3 2 と、異常判定部 2 3 3 と、を備える。辞書読込部 2 3 1 は、辞書記憶部 M 2 から識別器の辞書を読み込み、ハッシュ関数のパラメータを識別実行部 2 3 2 に送る。識別実行部 2 3 2 は、画像データを学習処理後の NN に入力して得られた出力データを NN 認識実行部 1 2 5 から受け取り、この出力データを複数の識別器に入力することにより識別結果を得る。例えば、識別実行部 2 3 2 は、複数のハッシュ関数のそれぞれを用いて NN 認識実行部 1 2 5 の出力結果に対する処理を行うことにより、バイナリ値のセットを得る。異常判定部 2 3 3 は、識別実行部 2 3 2 の出力に基づいて、画像データ中の異常を検知する。検出対象の映像に異常があるか否かを判定する。

#### 【 0 0 7 8 】

本実施形態においては、先に説明した学習フェーズにより得られた NN 及びハッシュ関数が、検出対象データの識別のために用いられる。しかしながら、NN と複数の識別器とを組み合わせ検出対象データの識別を行う構成において、NN 及び複数の識別器の学習を先に説明した方法で行うことは必須ではなく、別の方法を用いてもよいことに注意されたい。一実施形態においては、複数の識別器は、複数の識別器の類似度が低いほど評価が高くなるように学習により最適化されている。また、一実施形態において、NN は、学習データを NN に入力して得られた出力データを複数の識別器に入力することにより得られる識別結果の誤差を、NN に逆伝播する学習処理によって得られている。

#### 【 0 0 7 9 】

次に、NN 認識部 2 2 及び識別部 2 3 の動作に関して、図 4 を参照して詳細な説明を行う。ステップ S 1 3 0 1 で、辞書読込部 2 2 1 は、NN の辞書を辞書記憶部 M 1 から読み込み、NN 認識実行部 1 2 5 に送る。ステップ S 1 3 0 2 で、辞書読込部 2 3 1 は、識別器の辞書を辞書記憶部 M 2 から読み込み、識別実行部 2 3 2 に送る。本実施形態において辞書読込部 2 3 1 は、さらに正常データの情報を辞書記憶部 M 2 から読み込む。

#### 【 0 0 8 0 】

ステップ S 1 3 0 3 で、小領域抽出部 1 2 1 は検出対象データを受け取り、検出対象の映像から小領域を抽出する。ステップ S 1 3 0 4 で、前処理部 1 2 2 は、小領域画像から平均画像を減算する処理を行い、得られたデータを NN 認識実行部 1 2 5 に入力データとして送る。ステップ S 1 3 0 5 で、NN 認識実行部 1 2 5 は、前処理部 1 2 2 から得たデータを NN に入力することにより認識処理を実行し、出力を得る。

#### 【 0 0 8 1 】

ステップ S 1 3 0 6 で、識別実行部 2 3 2 は、NN 認識実行部 1 2 5 の出力を受け取る。ステップ S 1 3 0 7 で、識別実行部 2 3 2 は、学習済みのハッシュ関数を用いて NN 認識実行部 1 2 5 の出力（すなわち NN の認識結果）に対して識別処理を行う。ここで、識別実行部 2 3 2 は、式（ 1 ）に関して上述した方法に従い、ハッシュ関数に基づいて、NN の認識結果に対するバイナリ値を得る。識別実行部 2 3 2 は、複数のハッシュ関数のそれぞれを用いて NN の認識結果を処理するため、ステップ S 1 3 0 7 ではバイナリ値のセットが得られる。

#### 【 0 0 8 2 】

ステップ S 1 3 0 8 で、異常判定部 2 3 3 は、ステップ S 1 3 0 7 で得られたバイナリ値のセットに従って、小領域画像が異常か否か、すなわち検出対象の映像に異常が含まれるか否かを判定する。ステップ S 1 3 0 8 の処理は、例えば特許文献 3 に記載されている方法を用いることができる。簡単に説明すれば、正常データに対して認識処理を行った際に得られるバイナリ値と、ステップ S 1 3 0 7 で得られたバイナリ値と、が一致するような正常データの数に基づいて、入力データが異常か否かを判定することができる。本実施形態において、辞書読込部 2 3 1 は、このような正常データの数を正常情報として読み込んでいる。もっとも、ステップ S 1 3 0 8 の判定において正常情報を参照することは必須ではない。例えば、ステップ S 1 3 0 7 で得られた複数のバイナリ値が、全て、小領域画像が正常であることを示す（すなわちバイナリ値が「 0 」である）場合に、小領域画像が正常であると判断することができる。また、ステップ S 1 3 0 7 で得られた複数のバイナリ

10

20

30

40

50

値のうち、小領域画像が正常であることを示すバイナリ値が一定割合以上である場合に、小領域画像が正常であると判断することもできる。ステップS 1 3 0 9で、識別実行部2 3 2は、検出対象の映像に異常が含まれる場合に、このことを示す情報を端末装置3 0に送る。

#### 【0083】

本実施形態では、NNのパラメータをランダムに初期化してからNNの学習を行った。一方で、学習済みのNNを再利用し、所望のデータを用いて追加学習してもよい(Fine-tuning)。例えば、非特許文献1で得られたNNのパラメータを初期値として、さらに学習を行ってもよい。この際、非特許文献1で用いられたデータセットの教師情報と、本実施形態における学習データの教師情報とが異なる場合は、再利用するNNの出力層のパラメータのみを新たに定義又は初期化してから、追加学習を行うことができる。また、Fine-tuning以外の方法として、例えばAuto encoderを用いて教師なし学習を行い、学習により得られたパラメータを初期値として利用してもよい。その場合の具体的な方法としては、例えば非特許文献4に記載されている方法を用いることができる。

10

#### 【0084】

本実施形態では、一例として、図9のようにNNとハッシュ関数とを組み合わせる例について説明した。しかしながら、図9に示す構成を採用することは必須ではなく、異なる構成を用いてもよい。例えば、図9は出力層907の出力値をハッシュ関数の入力データとして用いる例を示すが、さらにConvolution2層904の出力をハッシュ関数の入力データとして用いてもよい。この場合、ステップS 7 1 0及びS 1 3 0 6では、Convolution2層904の出力と、出力層907の出力と、の双方がハッシュ関数に入力される。また、この場合、ステップS 7 1 3で、ハッシュ関数の誤差を、Convolution2層904と、出力層907と、の双方に対して逆伝播させることができる。なお、複数の誤差が伝播されるNNの学習動作は、Christian Szegedy et al. "Going Deeper with Convolutions" Conference on Computer Vision and Pattern Recognition, 2015に示されており、このような方法を用いて学習を行うことができる。

20

#### 【0085】

本実施形態では一例として、図9のようなNNのネットワーク構造を示したが、ネットワーク構造は図示されたものに限定されない。例えば、Inner Product1層906及びInner Product2層907を用いなくてもよい。また、これら2つの層の代わりにハッシュ関数を導入することにより、階層的なハッシュ関数処理を行ってもよい。

30

#### 【0086】

本実施形態では一例として、検出対象の映像から切り出された小領域画像のデータをNNの入力データとして用いる場合について説明したが、入力データはこれに限定されない。例えば、Optical Flow画像を映像から抽出してNNへの入力データとして用いてもよい。さらに、Optical Flow画像とRGB画像とをそれぞれ異なるNNに入力し、それぞれのNNの出力をハッシュ関数に対して入力してもよい。

#### 【0087】

本実施形態では、データ中の物体やその状態を認識するNNとハッシュ関数とを用い、NNの出力をハッシュ関数に対する入力データとする構成において、ハッシュ関数の学習誤差がNNに伝播するように動作して学習を行う。とりわけ、本実施形態では、複数のハッシュ関数を識別器として用い、選択された一部のハッシュ関数だけを用いて学習を行うことを繰り返すこと、すなわち学習を行う際にハッシュ関数の一部をdropoutすることについて説明した。しかしながら、dropout処理を行うことは複数の識別器を用いる上で必須ではない。一方で、dropout処理を行うことにより、NNの出力を受け取る識別器の過学習を抑制し、最終的な認識結果の精度を向上させることができる。dropout処理の具体的な方法については特に限定されない。すなわち、識別器の識別能力を低下させ、学習データをNNに入力して得られた出力データを識別能力が低下した識別器に入力することにより得られる識別結果の誤差をNNに逆伝播すればよい。複数の

40

50

識別器の一部を `dropout` することは、複数の識別器による識別能力を低下させることに対応する。

【 0 0 8 8 】

また、`dropout` 処理の適用例は、ハッシュ関数を用いる場合に限定されないし、複数の識別器を利用する場合にも限られない。例えば、サポートベクタマシン等の他の識別器を用いる場合に同様の `dropout` 処理を行うこともできる。サポートベクタマシンを用いる場合、`dropout` 処理は以下のように行うことができる。すなわち、サポートベクタマシンを用いて識別を行う場合、サポートベクタマシンのパラメータと、NNの出力との内積計算が行われる。この際に、サポートベクタマシンのパラメータの一部を脱落させることにより、識別器の識別能力を低下させ、`dropout` の効果を得ることができる。このような場合、脱落させたパラメータの分だけ得られる内積の値が増減するため、`dropout` 後に得られる内積の値を補正してもよい。具体例としては、脱落させたパラメータの数と、全てのパラメータの数とに基づいて、内積を重み付けすることができる。このような構成により、`dropout` 後に得られる内積を、全てのパラメータを用いた場合に得られると想定される大きさにすることができる。

10

【 0 0 8 9 】

[ 変形例 ]

実施形態 1 では、一例として、ハッシュ関数の学習に用いる目的関数として式 ( 3 ) を用いたが、異なる目的関数を用いてもよい。例えば、複数の識別器間の直交性が高いほど評価値が高くなる目的関数を用いて学習を行うことができる。一例として、目的関数として式 ( 1 1 ) を用いることができる。

20

【 数 1 1 】

$$E = \sum_{i=1}^M \sum_{j=1}^M \mathbf{w}_i^T \mathbf{w}_j \quad \dots\dots (11)$$

*subject\_to* :  $z \geq 0$

式 ( 1 1 ) は、ハッシュ関数全体の目的関数を示している。 *subject\_to* は、式 ( 1 1 ) の制約を示している。  $z$  の定義は式 ( 5 ) と同様であり、すなわち式 ( 1 1 ) の *subject\_to* は、全てのハッシュ関数が、すべての学習データに対して  $z \geq 0$  を満たすという意味を持つ。  $M$  はハッシュ関数の数を表す。

30

【 0 0 9 0 】

式 ( 1 1 ) において、目的関数  $E$  はハッシュ関数のパラメータ  $w$  に関する内積により表される。すなわち、複数のハッシュ関数間の直交性が高いほど、目的関数  $E$  は小さくなり、すなわち評価値は高くなる。このため、式 ( 1 1 ) の  $E$  を最小化することで、制約を満たしながら、ハッシュ関数同士ができるだけ直交に近くなる。また、ハッシュ関数自身のノルムが小さくなる。このように、式 ( 1 1 ) を用いて学習及び最適化を行うことにより、ハッシュ関数同士が直交に近くなり、ハッシュ関数同士が似る程度が低減される。

【 0 0 9 1 】

40

このように式 ( 1 1 ) を用いる場合には、それぞれのハッシュ関数がgreedyに最適化される可能性が低くなる。このため、式 ( 3 ) を用いる場合と比較して、学習の結果得られた複数のハッシュ関数に、互いに類似したハッシュ関数が含まれる可能性が低減される。したがって、式 ( 1 1 ) を用いることにより、識別器におけるハッシュ関数全体の表現能力が向上することが期待される。なお、上記の例では、類似度の表現に直交性という基準を用いて、式 ( 1 2 ) を設計した。当然ながら、直交性以外の類似度を用いてもよい。例えば、ハッシュ関数同士の距離を定義し、その距離を用いて式 ( 1 2 ) を変更してもよい。式 ( 1 2 ) は値が小さいほどよいとすると、距離のマイナス倍を式 ( 1 2 ) に利用することが考えられる。ここで、ハッシュ関数同士の距離はどのように定義してもよいが、例えば、異なるハッシュ関数間のパラメータ (例えば、重みベクトルおよび切片) の差の二乗

50

を用いてもよい。また、直交性をとらえるその他の尺度として、コサイン類似度を用いてもよい。コサイン類似度は、ハッシュ関数間の角度をとらえることが可能であり、ノルムの大きさに依存しないという性質がある。

【 0 0 9 2 】

式 ( 1 1 ) では、全てのハッシュ関数に関して  $z$  が 0 以上になるという制約を用いたが、このような構成には限定されない。例えば、 $z$  に関して一定の誤差を許容してもよい。このための方法としては、例えばサポートベクタマシンで一般に使われるように、 $z$  に関するソフトマージンを導入してもよい。このような制約が緩められた構成を用いることにより、学習データにノイズが含まれる場合であっても識別能力を維持できることが期待される。

10

【 0 0 9 3 】

別の方法として、 $z$  に関する制約を用いる代わりに、 $z$  を含む項を目的関数に加えてもよい。例えば、 $z$  を含む項が目的関数  $E$  の計算式に追加された式 ( 1 2 ) を用いて、ハッシュ関数の最適化を行うことができる。

【 数 1 2 】

$$E = \sum_{i=1}^M \sum_{j=1}^M \mathbf{w}_i^T \mathbf{w}_j + \sum_k (-\min(z_k, 0))^p \quad \cdots \cdots (12)$$

式 ( 1 2 ) において、 $p$  は  $p$ -norm である。例えば、二乗ノルムを用いることができ、この場合  $p = 2$  である。式 ( 1 2 ) の第 2 項は、式 ( 1 1 ) の  $z$  に関する制約に似た役割を果たす。

20

【 0 0 9 4 】

なお、本実施例では異常検知システムを用いて映像中の異常を検知する例を示したが、当然ながら本実施例で示した構成の一部又は全部を利用して、その他の応用に用いてもよい。例えば、工場のライン上の製品を外観検査する場合に、本実施例で例示した方法を用いて、異常検知を行ってもよい。その場合、例えば良品の画像データを正常データとして認識部 ( NN 及びハッシュ関数 ) を学習してもよい。これにより、製品の画像データが入力された際に、認識器の出力結果に基づいて、当該製品が異常か否かを検知することができる。なお、異常か否かを判定するための処理の詳細は、特許文献 3 に記載されている方法を用いてもよく、詳細は省略する。

30

【 0 0 9 5 】

なお、本実施例では識別器として線形識別器を用いる例を示したが、データをカーネル空間に写像する ( カーネル法に基づく非線形写像 ) ことで、非線形識別器の特性を利用できるように拡張してもよい。カーネルを利用する方法として、例えば、"Learning with kernels: support vector machines, regularization, optimization, and beyond", B Scholkopf, AJ Smola, MIT press, 2001 や、"Random Features for Large-Scale Kernel Machines", Ali R., et al, Advances in Neural Information Processing Systems 20 (NIPS 2007) に記載の方法を用いてもよい。

【 0 0 9 6 】

以上では、ハッシュ関数を最適化する際に用いる目的関数の例を示した。特に、式 ( 1 1 ) ( 1 2 ) を用いることにより、複数のハッシュ関数同士が類似しないように学習を行うことができる。しかしながら、目的関数がこれらのものに限定されるわけではない。例えば、式 ( 1 1 ) ( 1 2 ) では、ハッシュ関数同士が直交に近くなるように目的関数を定義した。しかしながら、ハッシュ関数同士の類似度をその他の方法で定義し、複数の識別器間の類似度が低いほど評価値が高くなる目的関数を用いて、類似度が最小化されるように、ハッシュ関数の学習を行ってもよい。例えば、類似度の一種として、ハッシュ関数同士の、パラメータの平均に関する差等を用いてもよい。

40

【 0 0 9 7 】

[ 実施形態 2 ]

50

実施形態 1 では、NN と組み合わせるハッシュ関数として NSH を用いる例を示した。実施形態 2 では、NN と組み合わせるハッシュ関数として、p-stable hash を用いる例について説明する。p-stable hash は、例えば M. Datar et al. "Locality-Sensitive Hashing Scheme Based on p-Stable Distributions", SoCG, 2004. (以下、Datar と呼ぶ) に記載されている。

【0098】

本実施形態における異常検知システムの構成及び動作は実施形態 1 と大部分が同一であり、以下では異なる点について説明する。本実施形態と実施形態 1 との大きな違いは、ハッシュ関数の生成方法と、その学習方法にある。

【0099】

本実施形態において、ハッシュ関数生成部 131 は、Datar に記載の方法に従って動作する。本実施形態で用いるハッシュ関数は、式 (13) で定義される。

【数 13】

$$\left\lfloor \frac{\mathbf{w}^T \mathbf{x} + c}{r} \right\rfloor \dots\dots (13)$$

式 (13) において、 $r$  は定数であり任意の値をとることができる。 $c$  は例えば  $[0, r]$  の範囲の値をとる実数である。式 (13) には  $\text{floor}$  関数が用いられており、式 (13) により得られる値は、 $(\mathbf{w}^T \mathbf{x} + c) / r$  の小数点以下を切り捨てた値である。このとき、 $c$  の増減に応じて、式 (13) の分子は傾きが同じで切片だけが移動する線形関数の集合として定義される。

【0100】

このように定義されたハッシュ関数を用いるとき、 $c$  の値として、任意の定数に従って  $[0, r]$  の区間を等分する値を用いることができる。例えば、 $r = 9$ 、刻み幅 1 の場合、 $c = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$  となり、計 10 個の等間隔に並ぶ平行なハッシュ関数が得られる。さらに、式 (13) のパラメータ  $x$  が異なるハッシュ関数を複数用意することにより、互いに平行なハッシュ関数の集合を複数含むハッシュ関数群を得ることができる。

【0101】

このように得られたハッシュ関数群を用いる場合、ハッシュ関数選択部 132 が行う  $\text{drop out}$  処理の方法としては様々なバリエーションが存在する。第 1 の方法として、実施形態 1 と同様に、全てのハッシュ関数を平等に扱い、すべてのハッシュ関数に対して確率的に  $\text{drop out}$  するか否かの判定を行うことができる。

【0102】

第 2 の方法として、 $c$  の値を変えることで得られた互いに平行なハッシュ関数の集合のそれぞれについて、集合に含まれる互いに平行なハッシュ関数を丸ごと  $\text{drop out}$  するか否かを判定することができる。この方法では、複数の集合のそれぞれについて、 $\text{drop out}$  するか否かを確率的に判定する処理を行うことができる。

【0103】

また、ハッシュ関数更新部 133 が行うパラメータ更新処理の方法にも様々なバリエーションが存在する。全てのハッシュ関数について独立にパラメータを更新することができる。一方で、互いに平行なハッシュ関数の集合について、各ハッシュ関数が平行かつ等間隔に並ぶ状態を維持するように、集合に含まれる互いに平行なハッシュ関数のパラメータを同様に更新することもできる。

【0104】

例えば、互いに平行なハッシュ関数の集合のうち 1 つのハッシュ関数のパラメータを更新する際は、この集合に含まれる他のハッシュ関数に関しても同様にパラメータを更新する

10

20

30

40

50

ことができる。すなわち、あるハッシュ関数のパラメータに対しての変更は、このハッシュ関数が属するハッシュ関数の集合に対して同じように反映する、という処理を行うことができる。

#### 【 0 1 0 5 】

ここで、パラメータの更新量をどのハッシュ関数に基づいて算出するかという点において、パラメータの更新方法には様々なバリエーションがある。以下では、その一例について説明する。図 7 は、共通のパラメータ  $w$  を持つハッシュ関数の集合と、学習データとの関係を例示する模式図である。図 7 には、共通したパラメータ  $w$  を持つハッシュ関数が表す超平面 1 4 0 1 ~ 1 4 0 5 と、学習データ 1 4 0 6 と、が示されている。学習データ 1 4 0 6 は、NN の出力を、ハッシュ関数に対する入力データとして可視化したものである。

10

#### 【 0 1 0 6 】

ここで、学習データ 1 4 0 6 を用いてパラメータの更新量を求める場合、ハッシュ関数 1 4 0 1 ~ 1 4 0 5 のそれぞれについて異なる更新量が算出される。この際、全てのハッシュ関数 1 4 0 1 ~ 1 4 0 5 についてパラメータの更新量を求め、それぞれの更新量に基づいてハッシュ関数の集合に共通に適用する更新量を決定してもよい。一方で、ハッシュ関数 1 4 0 1 ~ 1 4 0 5 のうち一部だけについてパラメータの更新量を求めてもよい。

#### 【 0 1 0 7 】

例えば、1 つのハッシュ関数についてパラメータの更新量を求め、得られた更新量をハッシュ関数の集合に共通に適用することができる。この場合、パラメータの更新量を求めるために用いるハッシュ関数の選択方法として、学習データをよりよく表現しているハッシュ関数を選択する方法が挙げられる。具体的な選択方法は特に限定されないが、例えば、それぞれのハッシュ関数について超平面から学習データまでの距離（法線の距離）を算出し、距離が最も小さいハッシュ関数を選択することができる。図 7 の例であれば、学習データ 1 4 0 6 に最も距離が近いのは、ハッシュ関数の超平面 1 4 0 1 である。実施形態 1 で説明したように `Mini batch` に含まれる複数の学習データを用いてパラメータの更新量を求める場合も、同様にそれぞれの学習データまでの距離に基づいてハッシュ関数を選択することができる。

20

#### 【 0 1 0 8 】

なお、例えばすべてのハッシュ関数に対して確率的に `dropout` 処理を行っている場合等、`dropout` しているハッシュ関数がハッシュ関数の集合に含まれている場合、`dropout` していないものからハッシュ関数を選択することができる。例えば、図 7 の例において超平面 1 4 0 1 に対応するハッシュ関数が `dropout` している場合、これを除いたハッシュ関数の中から、学習データ 1 4 0 6 に最も距離が近い超平面に対応するハッシュ関数を選択することができる。

30

#### 【 0 1 0 9 】

##### [ 実施形態 3 ]

実施形態 1 , 2 では、NN の出力を用いて識別を行う識別器としてハッシュ関数を用いる場合について説明した。実施形態 3 では、アダブーストを用いて識別器を構成する場合の例を示す。ここで、アダブーストとは、複数の弱識別器を用いて 1 つの強識別器を構成する方法であり、ブースティング手法の一種であるとされる。アダブーストの一例は、P. Viola et al. "Rapid Object Detection using a Boosted Cascade of Simple Features", CVPR 2001. (以降、Viola) に示されている。しかしながら、アダブーストを行うための具体的な手法には様々なバリエーションが存在している。上記の文献に示される手法、及び本実施形態で説明する方法は、様々なアダブーストの方法のうち 1 つを用いた場合の具体例にすぎず、使用するブースティング方法に特に制限はないことに注意すべきである。

40

#### 【 0 1 1 0 】

特に、本実施例において重要なのは、実施形態 1 , 2 における複数のハッシュ関数のアナロジーとして例示されるアダブーストの構成と動作である。より具体的には、本実施形態においては、NN の出力を入力データとして受け取る複数の弱識別器の集合が存在し、こ

50

これらの識別器の誤差がNNに逆伝播される。なお、本実施形態でも弱識別器の一部が確率的にdrop outされるが、既に説明したように、drop out処理は必須ではない。

#### 【0111】

本実施形態における異常検知システムの構成及び動作は実施形態1, 2と大部分が同一であり、以下では異なる点について説明する。本実施形態と実施形態1との大きな違いは、本実施形態では複数の弱識別器を識別器として用いる点、及び識別器の学習方法にある。すなわち、本実施形態に係る異常検知システムの構成は、識別器学習部13及び識別部23においてハッシュ関数の代わりに弱識別器を用いることを除き、図1に示す実施形態1の構成と同様である。

#### 【0112】

図13は、本実施形態における識別器学習部13の構成例を示す。識別器学習部13は、弱識別器生成部131aと、弱識別器選択部132aと、弱識別器更新部133aと、を備える。

#### 【0113】

次に、識別器学習部13の動作について説明する。アダブースト自体は公知であり、例えばViolaに記載の構成を用いることができるため、以下では動作の概要及び本実施形態における特徴的な処理について説明する。

#### 【0114】

まず、弱識別器生成部131aは、複数の弱識別器の生成を行う。このとき、生成する弱識別器の種類は特に限定されないが、具体例として線形識別器を生成するものとする。また、生成する弱識別器の数も特に限定されないが、具体例として100個の弱識別器を生成するものとする。なお、アダブーストのバリエーションの1つには、1個の弱識別器を最初に生成し、順次弱識別器を追加していく方法もあり、このような方法を採用することもできる。弱識別器を追加する方法としては、例えばViolaに記載の方法が挙げられる。

#### 【0115】

こうして生成された弱識別器を用いて、弱識別器生成部131aは、それぞれの弱識別器の信頼度を計算する。信頼度の計算方法は特に限定されず、例えばViolaに記載の方法を用いることができる。なお、最初に弱識別器を生成した際に、弱識別器生成部131aは、各学習データに正規化された重みを設定する。その具体的な方法としては、例えばViolaに記載の方法を用いることができる。

#### 【0116】

次に、弱識別器更新部133aは、NN認識実行部125から、それぞれの学習データをNNに入力して得られるNNの認識結果（出力データ）を受け取る。例えば、弱識別器更新部133aは、実施形態1と同様に出力層907の出力を受け取ることができる。もっとも、受け取るNNの認識結果は特に限定されず、例えば他の層の出力を受け取ってもよい。

#### 【0117】

次に、弱識別器選択部132aは、生成された弱識別器のうち一部を、学習に用いる弱識別器として選択する。この処理は、実施形態1, 2と同様に、学習のIterationごとに行うものとする。この際に、弱識別器を選択する基準は特に限定されない。例えば、実施形態1と同様に、弱識別器選択部132aは、各弱識別器を確率的に選択することができる。

#### 【0118】

次に、弱識別器更新部133aは、弱識別器選択部132aが選択した弱識別器の集合から、NNからの出力データを用いた識別処理について、最も正解率の高い弱識別器を選択する。そして、弱識別器更新部133aは、選択された弱識別器について信頼度を再計算し、各学習データを再度重み付けする。信頼度の計算及び重み付け方法は特に限定されないが、例えばViolaに記載の方法を用いることができる。具体例としては、正解率が高いほど選択された弱識別器の信頼度を上げることができ、また選択された弱識別器が学

10

20

30

40

50

習データの識別に失敗している場合にこの学習データの重みを上げることができる。Violaの方法によれば、選択された複数の弱識別器全体についての識別結果の誤差として、最も正解率の高い弱識別器について識別誤差が得られるが、この誤差をNNに逆伝播することができる。別の方法として、信頼度に従って弱識別器を組み合わせて得られる強識別器についての識別誤差をNNに逆伝播してもよい。

【0119】

その後、弱識別器更新部133aは、再び、NN認識実行部125から、それぞれの学習データを更新後のNNに入力して得られるNNの認識結果（出力データ）を受け取ることができる。以後は実施形態1と同様であり、弱識別器更新部133aは、学習に用いる弱識別器の選択、識別誤差の算出、及び誤差のNNへの逆伝播を繰り返すことができる。

10

【0120】

一方、弱識別器更新部133aは、以上の処理によって弱識別器のそれぞれに信頼度を設定することができる。そして、この信頼度に従って弱識別器を組み合わせることにより、強識別器を構成することができる。こうして得られた強識別器のパラメータは、辞書記憶部M2に格納される。認識装置20の動作において、弱識別器を用いた認識処理を行う際に用いる方法は特に限定されず、例えばViolaに記載の方法を用いることができる。その際には、異常と正常とを判別する識別器として弱識別器の学習処理が行われる。

【0121】

本実施形態では、データ中の物体やその状態を認識するNNと弱識別器とがあり、NNの出力を弱識別器に対する入力データとする構成において、弱識別器の誤差がNNに伝播するように動作して学習を行う。このとき、選択された一部の弱識別器だけを用いて学習を行うことを繰り返すことで、弱識別器の過学習が抑制され、最終的な認識結果（異常検知結果）の精度を向上させることができる。

20

【0122】

（その他の実施例）

本発明は、上述の実施形態の1以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける1つ以上のプロセッサがプログラムを読み出し実行する処理でも実現可能である。また、1以上の機能を実現する回路（例えば、ASIC）によっても実現可能である。

【符号の説明】

30

【0123】

10：学習装置、11：データ読込部、12：NN学習部、13：識別器学習部、20：認識装置、21：データ読込部、22：NN認識部、23：識別部、30：端末装置、31：表示部

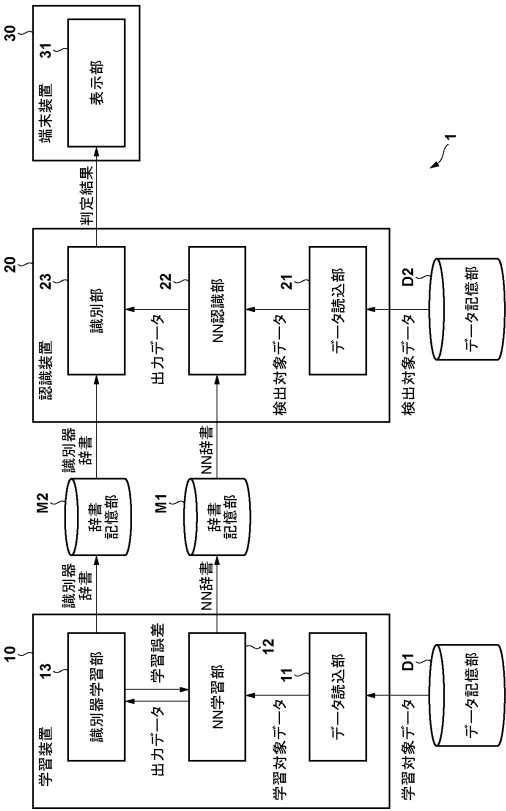
40

50

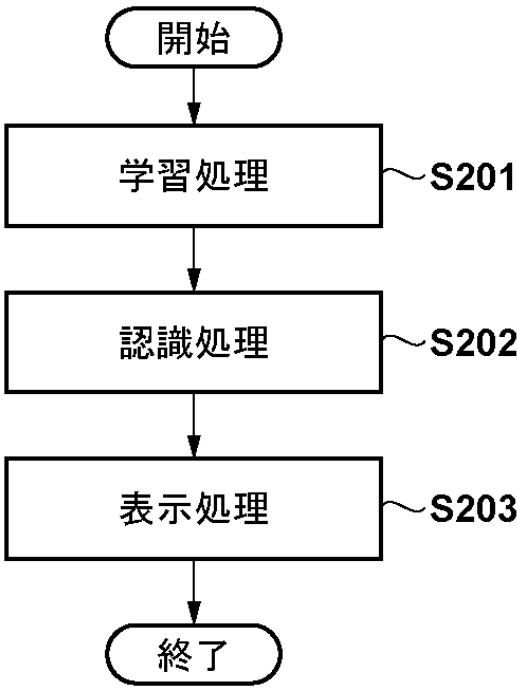


【図面】

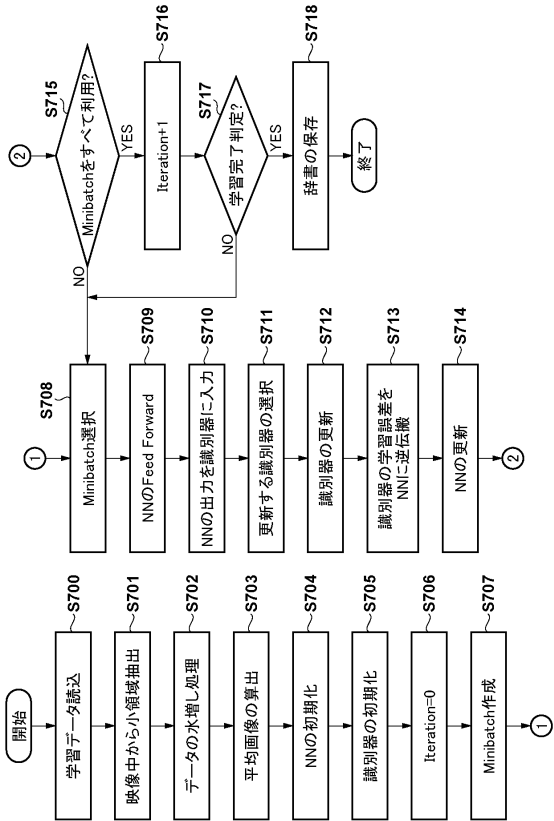
【図 1】



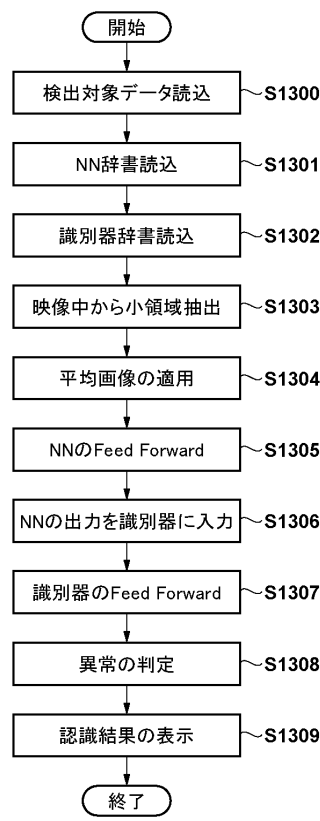
【図 2】



【図 3】



【図 4】



10

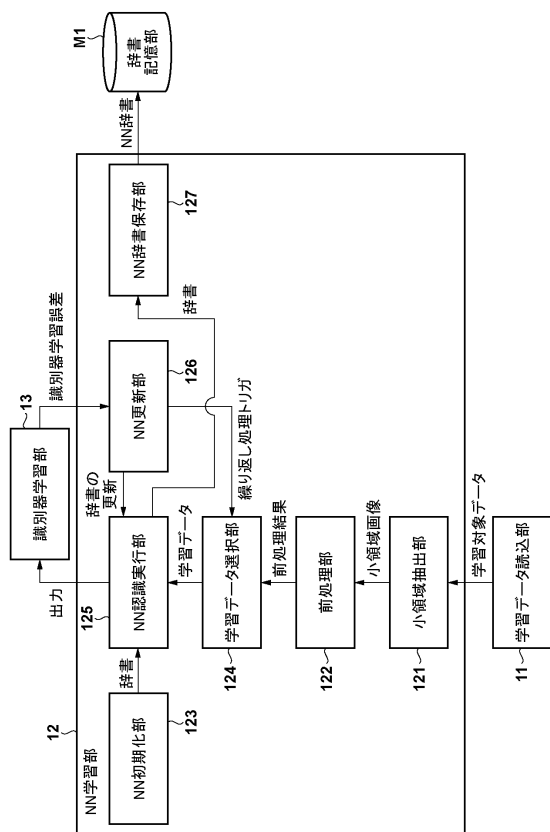
20

30

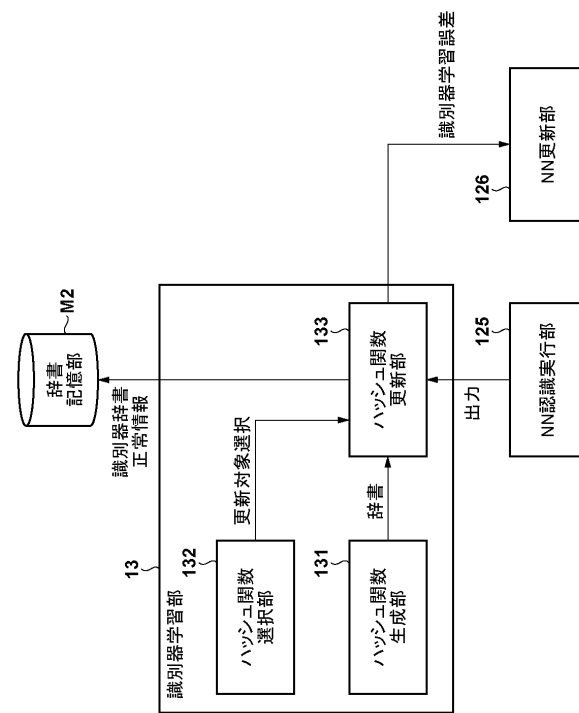
40

50

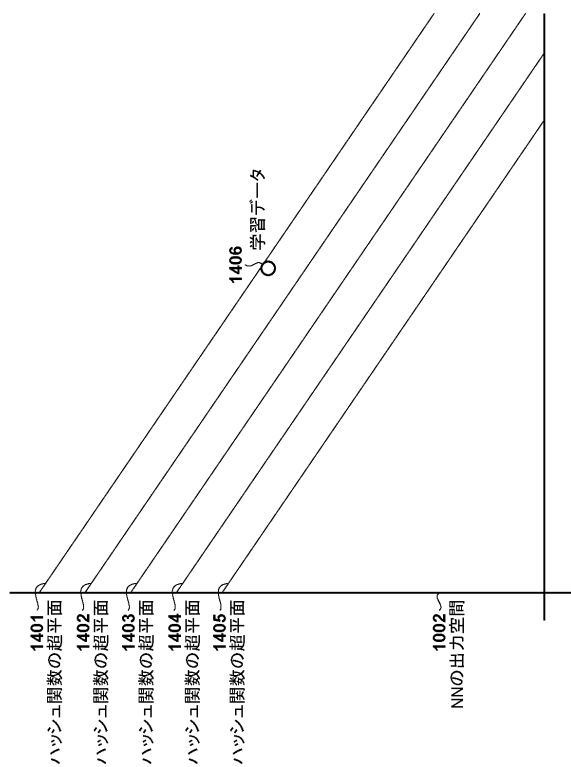
【 図 5 】



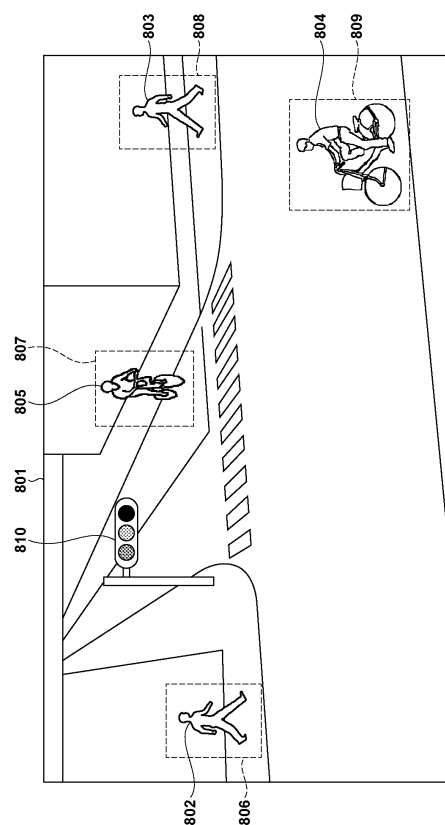
【 図 6 】



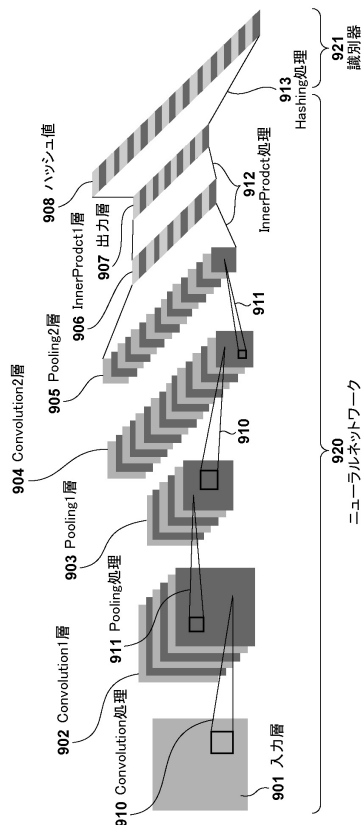
【圖 7】



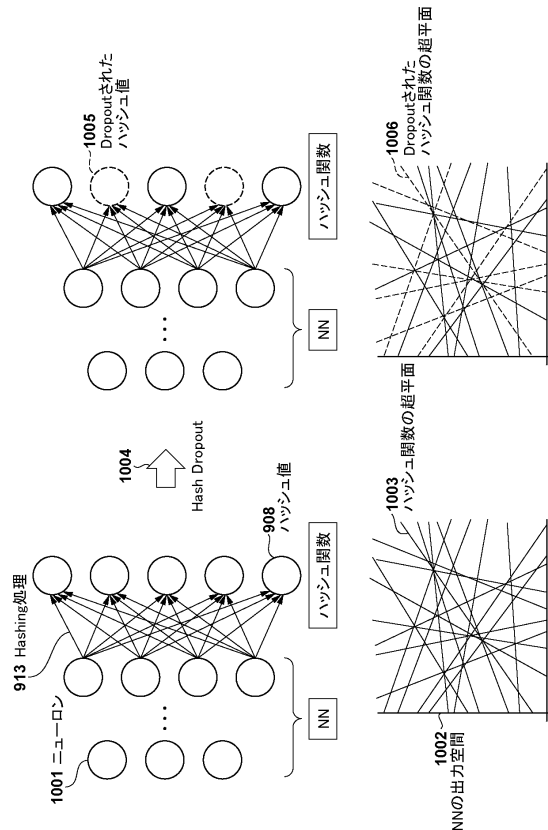
【図 8】



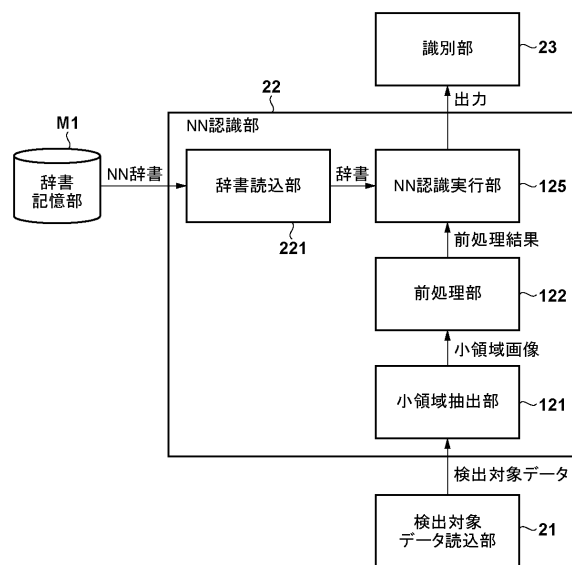
【図 9】



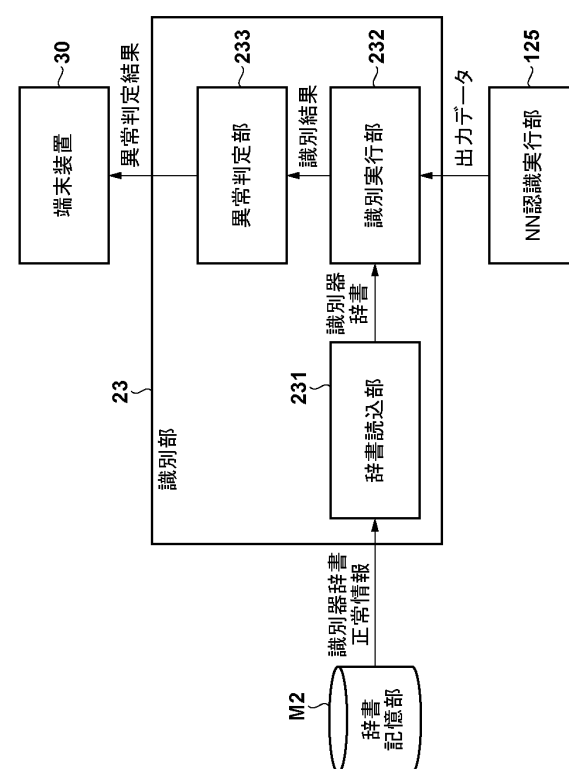
【図 10】



【図 11】



【図 12】



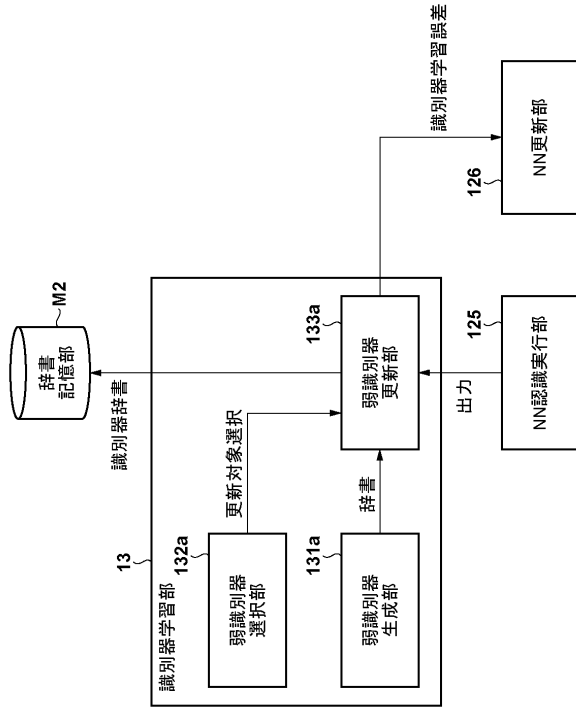
10

20

30

40

50



10

20

30

40

50

---

フロントページの続き

- (56)参考文献      特開 2 0 1 6 - 0 0 4 5 5 0 ( J P , A )  
                    特開 2 0 1 3 - 1 8 2 6 2 1 ( J P , A )  
                    特表 2 0 1 7 - 5 2 0 8 6 4 ( J P , A )  
                    特開平 0 4 - 3 6 4 5 6 4 ( J P , A )  
                    特開 2 0 0 5 - 2 5 8 8 0 1 ( J P , A )
- (58)調査した分野 (Int.Cl. , D B 名)  
                    G 0 6 N      3 / 0 0