

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2008-9893

(P2008-9893A)

(43) 公開日 平成20年1月17日(2008.1.17)

(51) Int. Cl.

G06T 7/00 (2006.01)

F I

G06T 7/00 300F

G06T 7/00 350C

テーマコード (参考)

5 L O 9 6

審査請求 未請求 請求項の数 18 O L (全 30 頁)

(21) 出願番号 特願2006-181896 (P2006-181896)

(22) 出願日 平成18年6月30日 (2006.6.30)

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(74) 代理人 100090538

弁理士 西山 恵三

(74) 代理人 100096965

弁理士 内尾 裕一

(72) 発明者 御手洗 裕輔

東京都大田区下丸子3丁目30番2号キヤ

ノン株式会社内

(72) 発明者 真継 優和

東京都大田区下丸子3丁目30番2号キヤ

ノン株式会社内

最終頁に続く

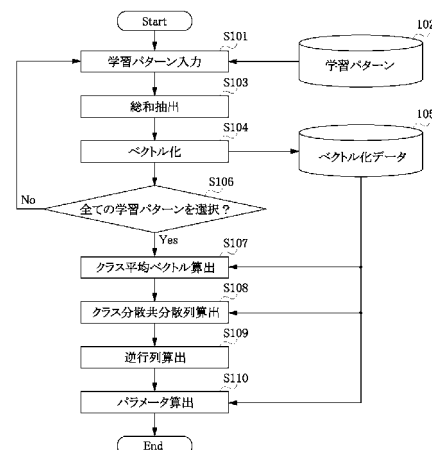
(54) 【発明の名称】 パラメータ学習方法及びその装置、パターン識別方法及びその装置、プログラム

(57) 【要約】

【課題】 高速かつ高精度なパターン識別のためのパラメータ学習方法を提供する。

【解決手段】 それぞれが属するクラスと対応付けられた複数の学習データを入力し(ステップS101)、前記複数の学習データの各々において、特定のk個(k1)の部分について部分ごとに要素の総和値を算出する(ステップS103)。前記複数の学習データの各々を、算出した前記k個の総和値を要素とするベクトルとして、該k個の総和値によって張られるk次元特徴空間に写像し(ステップS104)、写像された各学習データとその属するクラスとに基づいて、当該k次元特徴空間において、入力データを前記複数のクラスに分類するためのパラメータを学習する(ステップS107~110)。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

入力データを複数のクラスに分類するためのパラメータ学習方法であって、
それぞれが属するクラスと対応付けられた複数の学習データを入力する学習データ入力
ステップと、

前記複数の学習データの各々において、特定の k 個 ($k \geq 1$) の部分について部分ごと
に要素の総和値を算出する算出ステップと、

前記複数の学習データの各々を、算出した前記 k 個の総和値を要素とするベクトルとし
て、該 k 個の総和値によって張られる k 次元特徴空間に写像する写像ステップと、

前記 k 次元特徴空間に写像された各学習データとその属するクラスとに基づいて、当該
 k 次元特徴空間において、入力データを前記複数のクラスに分類するためのパラメータを
学習する学習ステップと

を有することを特徴とするパラメータ学習方法。

【請求項 2】

前記パラメータは、前記 k 次元特徴空間における線形識別関数及びクラス間境界を規定
する値を含むことを特徴とする請求項 1 に記載のパラメータ学習方法。

【請求項 3】

前記学習ステップにおいて、前記学習データの持つ重みに応じた学習を行うことを特徴
とする請求項 1 に記載のパラメータ学習方法。

【請求項 4】

前記重みを更新して前記学習ステップを繰り返し実行することで、複数のパラメータを
得ることを特徴とする請求項 3 に記載のパラメータ学習方法。

【請求項 5】

前記写像ステップにおいて、前記 k 個の総和値を、前記学習データの統計量または前記
 k 個の部分のそれぞれの大きさの、少なくとも 1 つに基づいて正規化することを特徴とす
る請求項 1 に記載のパラメータ学習方法。

【請求項 6】

前記学習ステップは、

前記 k 次元特徴空間に写像された学習データを非線形変換により新たな特徴空間に写像
する第 2 の写像ステップと、

前記新たな特徴空間に写像された複数の学習データに基づいて、当該新たな特徴空間に
おける線形識別関数及びクラス間境界を規定するパラメータを求めるパラメータ獲得ステ
ップと

を有することを特徴とする請求項 1 に記載のパラメータ学習方法。

【請求項 7】

前記学習データを x 、前記非線形変換を $\phi(x)$ とすると、関数 $K(x, x')$ を用い
て、 $\phi(x)^T \phi(x') = K(x, x')$ と表されることを特徴とする請求項 6 に記載
のパラメータ学習方法。

【請求項 8】

入力データを複数のクラスに分類するパターン識別方法であって、

請求項 1 乃至 7 に記載のパラメータ学習方法により学習したパラメータを入力するパラ
メータ入力ステップと、

識別対象データを入力する対象データ入力ステップと、

前記識別対象データにおいて、前記 k 個の部分と同位置の部分ごとに要素の総和値を算
出する第 2 の算出ステップと、

前記パラメータ入力ステップで入力されたパラメータと、前記第 2 の算出ステップで算
出された k 個の総和値とを用いて、前記識別対象データが前記複数のクラスの何れに属す
るかを識別する識別ステップと

を有することを特徴とするパターン識別方法。

【請求項 9】

10

20

30

40

50

複数の弱判別方法による判別により、入力データを複数のクラスに分類するパターン識別方法であって、

前記複数の弱判別方法の少なくとも１つが、

請求項１乃至７に記載のパラメータ学習方法により学習したパラメータを入力するパラメータ入力ステップと、

識別対象データを入力する対象データ入力ステップと、

前記識別対象データにおいて、前記 k 個の部分と同位置の部分ごとに要素の総和値を算出する第２の算出ステップと、

前記パラメータ入力ステップで入力されたパラメータと、前記第２の算出ステップで算出された k 個の総和値とを用いて、前記識別対象データが前記複数のクラスの何れに属する可能性が高いかを出力する出力ステップと

を有することを特徴とするパターン識別方法。

【請求項１０】

前記複数の弱判別方法の組み合わせは、ブースティング手法を用いて構築することを特徴とする請求項９に記載のパターン識別方法。

【請求項１１】

前記ブースティング手法による複数の弱判別方法の組み合わせの構築において、前記 k 個の部分を用いる複数の弱判別方法の内、最も識別性能の高い弱判別方法を選択していくことを特徴とする請求項１０に記載のパターン識別方法。

【請求項１２】

前記弱判別方法の１つにおける、前記複数のクラスの何れに属するかの可能性の高さに基づいて前記識別対象データを特定のクラスであると判定し、他の弱判別方法による判別を中断することを特徴とする請求項９に記載のパターン識別方法。

【請求項１３】

前記弱判別方法において、前記複数のクラスの何れに属するかの可能性に応じて、前記複数のクラスの何れに属するかの可能性の高さを示す累積値を更新するステップを有することを特徴とする請求項９に記載のパターン識別方法。

【請求項１４】

前記特定の部分が矩形領域であり、前記識別対象データの累積型データを生成する生成ステップを更に有し、

前記第２の算出ステップにおいて、前記累積型データを用いて前記総和値を算出することを特徴とする請求項８または９に記載のパターン識別方法。

【請求項１５】

入力データを複数のクラスに分類するためのパラメータ学習装置であって、

それぞれが属するクラスと対応付けられた複数の学習データを入力する学習データ入力手段と、

前記複数の学習データの各々において、特定の k 個 ($k \geq 1$) の部分について部分ごとに要素の総和値を算出する算出手段と、

前記複数の学習データの各々を、算出した前記 k 個の総和値を要素とするベクトルとして、該 k 個の総和値によって張られる k 次元特徴空間に写像する写像手段と、

前記 k 次元特徴空間に写像された各学習データとその属するクラスとに基づいて、当該 k 次元特徴空間において、入力データを前記複数のクラスに分類するためのパラメータを学習する学習手段と

を有することを特徴とするパラメータ学習装置。

【請求項１６】

入力データを複数のクラスに分類するパターン識別装置であって、

請求項１５に記載のパラメータ学習装置により学習したパラメータを入力するパラメータ入力手段と、

識別対象データを入力する対象データ入力手段と、

前記識別対象データにおいて、前記 k 個の部分と同位置の部分ごとに要素の総和値を算

10

20

30

40

50

出する第 2 の算出手段と、

前記パラメータ入力手段により入力されたパラメータと、前記第 2 の算出手段により算出された k 個の総和値とを用いて、前記識別対象データが前記複数のクラスの何れに属するかを識別する識別手段と

を有することを特徴とするパターン識別装置。

【請求項 17】

複数の弱判別器を備え、該複数の弱判別器による判別により、入力データを所定のクラスに分類するパターン識別装置であって、

前記弱判別器の少なくとも 1 つが、

請求項 15 に記載のパラメータ学習装置により学習したパラメータを入力するパラメータ入力手段と、

識別対象データを入力する対象データ入力手段と、

前記識別対象データにおいて、前記 k 個の部分と同位置の部分ごとに要素の総和値を算出する第 2 の算出手段と、

前記パラメータ入力手段により入力されたパラメータと、前記第 2 の算出手段により算出された k 個の総和値とを用いて、前記識別対象データが前記複数のクラスの何れに属する可能性が高いかを出力する出力手段とを有する

ことを特徴とするパターン識別装置。

【請求項 18】

入力データを複数のクラスに分類するためのパラメータ学習方法をコンピュータに実行させるコンピュータ読み取り可能なプログラムであって、

それぞれが属するクラスと対応付けられた複数の学習データを入力する学習データ入力ステップと、

前記複数の学習データの各々において、特定の k 個 ($k \geq 1$) の部分について部分ごとに要素の総和値を算出する算出ステップと、

前記複数の学習データの各々を、算出した前記 k 個の総和値を要素とするベクトルとして、該 k 個の総和値によって張られる k 次元特徴空間に写像する写像ステップと、

前記 k 次元特徴空間に写像された各学習データとその属するクラスとに基づいて、当該 k 次元特徴空間において、入力データを前記複数のクラスに分類するためのパラメータを学習する学習ステップとをコンピュータに実行させることを特徴とするコンピュータ読み取り可能なプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、画像認識や音声認識等において、入力信号のパターンを識別するための識別用パラメータの学習、及びそれを用いたパターン識別に関するものである。

【背景技術】

【0002】

従来から、光学的文字認識や自動音声認識等において、入力データを所定のクラスに分類するためのパターン識別方法として多くの手法が考案されている。現在に至っても、処理の高速化や、分類精度の向上等を目的とした様々な新しい手法が提案されている。

【0003】

例えば非特許文献 1 のように、非常に短時間で演算が可能な弱判別方法を用い、AdaBoost による学習法と、弱判別方法のカスケード識別手法の組み合わせにより、高速、且つ高精度なパターン識別方法が提案されている。

【0004】

また、非特許文献 2 のように、上記パターン識別方法の更なる高速化・高精度化を目的として、Real AdaBoost の手法（非特許文献 3 参照）というブースティング手法（非特許文献 4 参照）と、入れ子型カスケード識別手法を導入するという手法も提案

10

20

30

40

50

されている。

【0005】

また、非線形の識別関数を構成できるサポートベクターマシン (Support Vector Machine, SVM) と呼ばれるパターン認識手法が知られている (非特許文献5を参照)。

【0006】

このように、処理装置が高性能となってきた背景もあり、高速、且つ高精度なパターン識別により、リアルタイムにパターン識別を行うことができる技術が望まれている。

【0007】

【非特許文献1】P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Vol. 1, pp. 511 - 518, December 2001 10

【非特許文献2】勞世▲紅▼, 山下隆義, 岡本卓也, 川出雅人, "高速全方向顔検出", MIRU 2004, Vol. 2, pp. 271 - 276, July 2004

【非特許文献3】R. E. Schapire, Y. Singer, "Improved Boosting Algorithms Using Confidence-rated Predictions", Machine Learning, Vol. 37, pp. 297 - 336, December 1999 20

【非特許文献4】R. O. Duda, P. E. Hart, D. G. Stork, "パターン識別 Pattern Classification Second Edition", 新技術コミュニケーションズ, pp. 478 - 482, July 2001

【非特許文献5】津田宏治, "Technical Survey「サポートベクターマシンとは何か」", 電子情報通信学会誌, Vol. 83, pp. 460 - 466, June 2000

【発明の開示】

【発明が解決しようとする課題】

【0008】

本発明は、上述した点に鑑みなされたものであり、高速かつ高精度なパターン識別のためのパラメータ学習方法、及び、それを用いたパターン識別方法を提供する。 30

【課題を解決するための手段】

【0009】

上記課題を解決するために、本発明によれば、入力データを複数のクラスに分類するためのパラメータ学習方法に、それぞれが属するクラスと対応付けられた複数の学習データを入力する学習データ入力ステップと、前記複数の学習データの各々において、特定のk個 (k-1) の部分について部分ごとに要素の総和値を算出する算出ステップと、前記複数の学習データの各々を、算出した前記k個の総和値を要素とするベクトルとして、該k個の総和値によって張られるk次元特徴空間に写像する写像ステップと、前記k次元特徴空間に写像された各学習データとその属するクラスとに基づいて、当該k次元特徴空間において、入力データを前記複数のクラスに分類するためのパラメータを学習する学習ステップとを備える。 40

【0010】

また、本発明の他の態様によれば、入力データを複数のクラスに分類するためのパラメータ学習装置に、それぞれが属するクラスと対応付けられた複数の学習データを入力する学習データ入力手段と、前記複数の学習データの各々において、特定のk個 (k-1) の部分について部分ごとに要素の総和値を算出する算出手段と、前記複数の学習データの各々を、算出した前記k個の総和値を要素とするベクトルとして、該k個の総和値によって張られるk次元特徴空間に写像する写像手段と、前記k次元特徴空間に写像された各学習データとその属するクラスとに基づいて、当該k次元特徴空間において、入力データを前 50

記複数のクラスに分類するためのパラメータを学習する学習手段とを備える。

【0011】

また、本発明の他の態様によれば、入力データを複数のクラスに分類するためのパラメータ学習方法をコンピュータに実行させるコンピュータ読み取り可能なプログラムが、それぞれが属するクラスと対応付けられた複数の学習データを入力する学習データ入力ステップと、前記複数の学習データの各々において、特定の k 個($k \geq 1$)の部分について部分ごとに要素の総和値を算出する算出ステップと、前記複数の学習データの各々を、算出した前記 k 個の総和値を要素とするベクトルとして、該 k 個の総和値によって張られる k 次元特徴空間に写像する写像ステップと、前記 k 次元特徴空間に写像された各学習データとその属するクラスとに基づいて、当該 k 次元特徴空間において、入力データを前記複数のクラスに分類するためのパラメータを学習する学習ステップとをコンピュータに実行させることを特徴とする。

【発明の効果】

【0012】

本発明によれば、高速かつ高精度なパターン識別、あるいはそのためのパラメータの学習を実行することが可能となる。

【発明を実施するための最良の形態】

【0013】

以下、添付図面を参照しながら、本発明の実施形態を説明する。

【0014】

(第1の実施の形態)

本発明の第1の実施形態として、入力データとして2次元画像データを用い、その画像データが、ある特定のカテゴリであるかどうかを識別するパターン識別方法に用いるパラメータ学習方法、及びそれを用いたパターン識別方法を示す。このパターン識別方法は、つまり入力された画像データを、特定のカテゴリであるというクラスと、特定のカテゴリではないというクラスの2クラスに分類する方法である。

【0015】

特定のカテゴリとしては、例えば人間の顔であるとか、あるパーツの欠陥であるといったカテゴリが挙げられる。本実施形態では、人間の顔の中心付近が入力画像のほぼ中央に存在するパターン(以下、顔パターン)を、特定のカテゴリとした場合の例を示す。

【0016】

図1に、第1の実施形態におけるパターン識別用パラメータ学習方法の処理の流れを示す。以下、このパターン識別用パラメータ学習方法について図1を用いて説明する。

【0017】

まず、ステップS101において、データベース102に保持してある複数の学習パターンから、1つの学習パターンを入力する。データベース102には、顔パターンと、それ以外の任意のパターン(以下、非顔パターンと称する)を、そのラベル(顔パターンであるのか、非顔パターンであるのか)と共に予め複数保持しておく。本実施形態では、データベース102に保持しておくパターンは、グレースケール画像であり、各画素値が輝度値として表現されたものである。ここでは、全顔パターンは、顔の向きやサイズをある程度正規化しておき、顔・非顔を含めた全パターンを一定の画像サイズにして保持しておく。

【0018】

次に、ステップS103において、ステップS101で入力された学習パターンについて、4つの所定部分の輝度値の総和を求める。この所定部分は任意に設定すれば良いが、分類したい2クラスにおいて、差が顕著に表れるような部分を設定するのが好適である。本実施形態では、顔パターンと非顔パターンとの2クラスに分類するため、顔パターンにとって特徴的である、左右の目の領域、眉間の領域、口の領域を、4つの所定部分として設定する。

【0019】

10

20

30

40

50

具体的には、図2の破線で示すような、全顔パターンの左目が入るような領域21、全顔パターンの右目が入るような領域23、領域21と領域23の中間の眉間の位置に対応する領域22、全顔パターンの口が入るような領域24の4つの矩形領域を、上記所定部分として用いる。なお、本実施形態では、所定部分として4つの矩形領域を用いるが、この所定部分の個数や位置・大きさ・形状は任意であり、少なくとも1つの部分を設定しておき、その部分のデータの総和を求めるようにすればよい。

【0020】

次のステップS104では、ステップS103において求めた各領域の総和を単純に並べたベクトルを生成する。並べ方は任意であるが、本実施形態では、図2の領域21、領域22、領域23、領域24の順に、それぞれの総和値を並べたベクトルを生成する。そして、ここで生成したベクトルをベクトル化データとして、学習パターンのラベルと共にデータベース105に保持しておく。これにより、各学習パターンは、上記4個の総和値によって張られる4次元特徴空間に写像されたことになる。

10

【0021】

ステップS106では、データベース102に保持されている学習パターンの全てを選択したかどうかを判定する。全てを選択してなければ、ステップS101に戻り、データベース102から新たな学習パターンを選択していくようにする。全ての学習パターンが選択された場合、次のステップS107に進む。

【0022】

ステップS107では、データベース105に保持されたベクトルデータを用いて、各クラスの平均ベクトルを求める。

20

【0023】

次いで、ステップS108では、ステップS107で求めたクラス平均ベクトルと、データベース105に保持されたベクトルデータとを用いて、クラスごとの分散共分散行列を求める。

【0024】

次のステップS109では、ステップS108で求めたクラスごとの分散共分散行列に各クラスの事前確率を乗じ、その行列の和を求め、求めた行列の逆行列を算出する。各クラスの事前確率については、事前確率が既知である場合はそれを用いればよい。既知でない場合は、学習パターン数に応じた値に設定するか、学習したい識別器に求める性能に応じて設定すればよい。ここで、逆行列が存在しない、即ち求めた行列の和が正則でない場合は、ステップS103において総和抽出を行う部分の設定が適切でない可能性が高い。そのため、例えば図2に示した各領域の位置や大きさを変える等、所定部分の設定をし直して、もう一度はじめてから学習を開始する必要がある。

30

【0025】

最後に、ステップS110において、まず、ステップS107で求めた各クラスの平均ベクトルと、ステップS109で求めた逆行列とを用いて、パターン識別用パラメータの1つであるクラス分離ベクトルを算出する。クラス分離ベクトルは、顔パターンの平均ベクトルから非顔パターンの平均ベクトルを引いたベクトルを求め、そのベクトルの左側からステップS109の逆行列算出において求めた逆行列をかけることで求められる。

40

【0026】

次いで、もう1つのパターン識別用パラメータである識別閾値を、求めたクラス分離ベクトルと、データベース105に保持されたベクトルデータとを用いて求める。ここでは、まずデータベース105に保持された全ベクトルデータと、求められたクラス分離ベクトルとの内積を求める。この内積値は、顔パターンであるらしさを表す代表値とみなすことができ、例えばこの値が所定値以上であれば、顔パターンのクラスであると分類し、そうでなければ非顔パターンのクラスであると分類するようにする。この識別に用いる所定値が識別閾値となる。

【0027】

識別閾値の決定は、識別閾値を細かく振っていき、失敗パターン総数（顔パターンであ

50

るのに、非顔パターンのクラスに分類された総数と、非顔パターンであるのに顔パターンに分類された総数の和)が最小になる値を検索して決定するようにすればよい。

【0028】

本実施形態では、失敗パターン総数に基づいて識別閾値を決定するが、この方法に限るものではなく、求める識別器の性能に応じて適切に設定すればよい。また、クラス分離ベクトルの方向によっては、所定閾値以下の場合に顔パターンのクラスであると分類した方が、失敗パターン総数をより小さくできる場合がある。その場合は、クラス分離ベクトルの方向を逆にする(全要素の符号を反転する)か、パターン識別の際に、所定値以下のパターンが顔パターンのクラスであるというように取り決めておくようにすればよい。

【0029】

以上の手法により、パターン識別用のパラメータとして、クラス分離ベクトルと、識別閾値を得ることができる。これらはいわゆる線形識別関数とクラス間境界とを与えることになる。本実施形態では、クラス分離ベクトルを得るための手法として、上記のような、いわゆる線形判別分析法を用いたが、本発明はこれに限るものではなく、直交化学習等により、パラメータを逐次更新しながらクラス分離ベクトルを得るような手法でも構わない。

【0030】

続いて、上記手法で求めたパターン識別用のパラメータを用いたパターン識別方法によって、画像中の顔を検出する方法について、図3を用いて説明する。

【0031】

まず、ステップS301で、顔検出を行う画像を入力する。この画像は、上記説明したパターン識別用パラメータ学習時と同様のグレースケール画像である。グレースケール画像以外の画像に対して顔検出を行いたい場合は、まず学習時と同様のグレースケール画像に変換してから入力するようにすれば良い。

【0032】

次に、ステップS302において、矩形領域の総和を高速に求めるために、入力した画像のインテグラルイメージを生成する。インテグラルイメージとは、画像始点(例えば画像の左上の画素)から、注目画素までを対角線とする矩形領域内の画素値の総和を、注目画素の画素値として表現するものである。これを利用すると、任意の矩形領域内の画素値の総和を高速に求めることができる(インテグラルイメージの詳細については、非特許文献1を参照)。本実施形態では、ステップS304において、画素値の総和を抽出する所定部分として4つの矩形領域を利用するため、予めこのインテグラルイメージを生成しておくことで、総和抽出の処理が高速に実行できる。

【0033】

次いで、ステップS303で、画像中の所定の点の中から1つの点を定め、それを中心とする領域を生成したインテグラルイメージから切り出す。上記所定の点は、画像の全画素としても良いが、例えば1画素おきというように、粗くサンプリングするようにしても良い。切り出すサイズは任意だが、本実施形態では、学習時に利用した学習パターンと同じサイズになるように切り出しを行う。

【0034】

ステップS304において、切り出したインテグラルイメージを用いて、学習時にステップS103で総和を求めた部分と同一位置の、矩形領域内の画素値の総和を算出する。

【0035】

次に、ステップS305において、ステップS304で抽出した総和を、学習時のステップS104と同じ順序で並べたベクトルを生成する。

【0036】

そして、ステップS306において、ステップS305で生成したベクトルと、学習により予め生成しておき、識別用パラメータデータベース307に保持しておいたクラス分離ベクトルとの内積を演算する。ここで用いるクラス分離ベクトルは、予め所定値以上を顔パターンと分類するのが好適な方向になるようにしたものを用いる。

10

20

30

40

50

【 0 0 3 7 】

続いて、ステップ S 3 0 8 では、ステップ S 3 0 6 の内積演算で求めた内積値と、学習時に予め求めておき、データベース 3 0 7 に保持しておいた識別閾値との比較を行い、求めた内積値が識別閾値よりも高い場合に、ステップ S 3 0 3 で定めた点に顔が存在すると判定する。

【 0 0 3 8 】

ステップ S 3 0 8 で顔が存在すると判定された場合は、ステップ S 3 0 9 において、ステップ S 3 0 3 で定めた点に顔が存在するという結果を記録しておく。

【 0 0 3 9 】

そしてステップ S 3 1 0 では、ステップ S 3 0 3 で所定の点の全てを中心とする画像切り出しを行ったかどうかを判定して、行ったのであれば処理を終了する。そうでなければステップ S 3 0 3 に戻り、新たな所定の点を 1 つ定め、画像切り出しを行うという処理を繰り返す。 10

【 0 0 4 0 】

以上説明した処理により、入力された画像内の顔を検出することができる。この処理では、学習時に利用した顔パターンと同じような向き、サイズの顔を検出できる。もし、異なるサイズの顔を検出したい場合は、入力画像を縮小または拡大した画像を生成して、同様の処理を行えばよい。あるいは、ステップ S 3 0 4 において総和を求める所定部分を縮小または拡大し、ステップ S 3 0 5 において、縮小または拡大した分の補正を行ったベクトルを生成するようにすればよい。 20

【 0 0 4 1 】

ここでの補正に関しては、縮小または拡大した分の面積比の逆数を、求めた各総和値に乗じてやるようにする。また、顔の向きに関しては、面内回転が学習パターンと異なる顔を検出したい場合は、入力画像を予め回転させたものを入力して同様の処理を行えばよい。また、軸方向の回転が学習パターンと大きく異なる顔の検出をしたい場合は、同一のパターン識別方法で行うのは困難である。そこで、軸方向の回転ごとのパターン識別用パラメータを予め学習により生成しておき、それぞれのパターン識別方法を独立に用いて検出するようにすればよい。

【 0 0 4 2 】

以上、第 1 の実施の形態は、入力データとして 2 次元画像データを用い、その画像データが、ある特定のカテゴリであるかどうかを識別するパターン識別方法に用いるパラメータ学習方法、及びそれを用いたパターン識別方法を説明した。つまり、入力された画像データを、特定のカテゴリであるというクラスと、特定のカテゴリではないというクラスの 2 クラスに分類するパターン識別方法の例について説明した。 30

【 0 0 4 3 】

(第 2 の実施の形態)

第 2 の実施形態では、第 1 の実施形態と同様に、入力データとして 2 次元画像データを用いた 2 クラス分類のパターン識別を行なう。その際、ブースティング手法 (詳細については、非特許文献 4 を参照) を用いて構築する複数の弱判別器の組み合わせにより、さらに高精度なパターン識別に用いるパラメータ学習方法、及びそれを用いたパターン識別方法を示す。 40

【 0 0 4 4 】

図 4 に、第 2 の実施形態におけるパターン識別用パラメータ学習方法の処理の流れを示す。以下、このパターン識別用パラメータ学習方法について図 4 を用いて説明する。

【 0 0 4 5 】

まずステップ S 4 0 1 において、所定の矩形パターンの中から 1 つの矩形パターンを選択する。ここで選択する矩形パターンは、第 1 の実施形態における、目や口の位置に対応する矩形領域と同様のものである。ただし、本実施形態では、予め固定の 1 つの矩形パターンを用いるのではなく、所定の複数の矩形パターンのそれぞれに対して処理を行い、最も識別能力の高いパターン識別結果が得られる矩形パターンを選択する。所定の複数の矩 50

形パターンとしては、任意の位置・大きさの、矩形領域の組み合わせを用いればよい。

【0046】

本実施形態における矩形パターンの設定方法についてさらに詳細に説明する。本実施形態では、学習パターンとして、 20×20 画素の顔パターン、及び非顔パターンを用いるので、その 20×20 画素のパターン内に設定可能な矩形領域を2つ設定し、それを1つの矩形パターンとして用いる。1つの矩形は、その幅・高さ・縦位置・横位置で決まるので、その4つのパラメータを決めることにより、1つの矩形領域が設定される。

【0047】

そこで、まず矩形の幅 w を $1 \sim 20$ 、高さ h を $1 \sim 20$ の中からそれぞれ1つ決定する。次に、矩形の左上の座標位置 x を $0 \sim 20 - w$ 、矩形の左上の座標位置 y を $0 \sim 20 - h$ の中からそれぞれ1つ決定する。これで、1つの矩形領域が設定されることになる。本実施形態では、2つの矩形領域の組み合わせを1つの矩形パターンとして用いるので、もう1つの矩形領域も同様に決定する。ただし、2つ目の矩形領域は、1つ目に決定した矩形領域以外になるようにする。

10

【0048】

選択すべき所定の矩形パターンとしては、これらの組み合わせ全てを用いるようにしてもよいが、その場合、非常に処理に時間がかかる。そこで、例えば幅、高さは何れも奇数にする、矩形の左上の座標位置は、何れも偶数にする、といった制約を付けて、選択すべき所定の矩形パターン数を減らすようにしてもよい。

【0049】

本実施形態では、矩形領域2つから構成されるものを矩形パターンとしたが、本発明はこれに限るものではなく、その他の矩形領域数でも実施可能である。例えば、矩形領域3つから構成されるものを矩形パターンとする場合には、3つ目の矩形領域として、1つ目と2つ目に選ばれた以外の矩形領域を、上記と同様の方法で決定すればよい。矩形領域1つで構成されるものを矩形パターンとすることも可能であるが、その場合は、パターン識別用パラメータの求め方が異なるので、後述する。また、本実施形態では、矩形パターンは全て同じ2つの矩形領域から構成されるものとしているが、矩形領域が2つで構成されるものと、矩形領域3つで構成されるものとが混在するような場合にも適用可能である。

20

【0050】

ステップS401で1つの矩形パターンを選択した後、ステップS402において、データベース403に保持された複数の学習パターンから、1つの学習パターンを入力する。データベース403に保持しておくパターンは、第1の実施の形態と同様であるが、それぞれのパターンに重みを設定しておく点異なる。

30

【0051】

この重みは、学習時の重要度を示しており、この重みが高いパターンほど、それを重要視した学習を行うようにする。初期段階では、通常全てのパターンに同じ重みを設定しておく。つまり、パターンの総数が例えば N 個である場合、それぞれの重みを、 $1/N$ と設定しておけばよい。初期状態の重みの設定に関しては、これに限るものではなく、例えば事前確率によって傾斜した重みを設定するようにしてもよい。また、求める識別器の性能に応じて適当に設定してもよい。例えば非顔パターンの重みを大きく設定しておく、非顔パターンを誤って顔パターンのクラスに分類する確率を小さくすることができる(しかしこの場合、逆に、顔パターンを誤って非顔パターンのクラスに分類する確率は高くなる)。また、この重みの絶対値に関しては、特に限定はしないが、全学習パターンの重みの総和が1であると計算が簡便であるので、そのように設定しておくともよい。

40

【0052】

次に、ステップS404において、ステップS402で入力した学習パターンから、ステップS401で選択した矩形パターンに基づき、そのそれぞれの矩形領域内の輝度値の総和を抽出する。

【0053】

続いて、ステップS405において、ステップS404で求めたそれぞれの矩形領域の

50

総和を用いてベクトルを生成し、そのベクトルをステップ S 4 0 1 で選択した矩形パターンに対応するベクトルデータとして、データベース 4 0 6 に保持する。

【 0 0 5 4 】

ここで、識別能力向上のため、第 1 の実施形態とは異なり、求めた総和値の正規化を行い、それをベクトルの各要素値とするようにする。具体的には、ステップ S 4 0 2 で入力した学習パターンの輝度値の平均と、輝度値の標準偏差を用いる。まず、それぞれの矩形領域の総和から、その矩形領域の面積に輝度値の平均を乗じたものを引き、それに標準偏差の逆数を乗じることによって正規化を行う。このようにして正規化したそれぞれの矩形領域の総和値を、任意の順（1つの矩形パターンで同一の順序）で並べたベクトルを生成して、データベース 4 0 6 に保持しておく。

10

【 0 0 5 5 】

ステップ S 4 0 7 では、データベース 4 0 3 に保持されている学習パターンの全てを選択したかどうかを判定して、全てを選択していなければ、ステップ S 4 0 2 の学習パターン入力に戻り、新たな学習パターンを、データベース 4 0 3 から選択していくようにする。全ての学習パターンが選択された場合、次のステップ S 4 0 8 のクラス平均ベクトル算出に進む。

【 0 0 5 6 】

ステップ S 4 0 8 では、データベース 4 0 6 に保持された、ステップ S 4 0 1 の矩形パターン選択において選択した矩形パターンに対応するベクトルデータを用いて、各クラスの重み付平均ベクトルを求める。具体的には、各ベクトルデータに、先に述べた、対応する学習パターンの重みを乗じ、重みを乗じたベクトルの総和をクラスごとに求める。この時、同時に学習パターンの重みの総和もクラスごとに求めておく。そして、重みを乗じた全ベクトルの総和を算出後、クラスごとに総和を求めたベクトルを、そのクラスの重みの総和で割ることで、各クラスの重み付平均ベクトルが求められる。

20

【 0 0 5 7 】

次いで、ステップ S 4 0 9 では、ステップ S 4 0 8 で求めたクラスの重み付平均ベクトルと、データベース 4 0 6 に保持された、ステップ S 4 0 1 で選択した矩形パターンに対応するベクトルデータとを用いて、クラスごとの重み付分散共分散行列を求める。ここでも、この重み付分散共分散行列を求める際に、学習パターンの重みを利用する。

【 0 0 5 8 】

具体的には、まず、1つのベクトルデータから、そのベクトルデータに対応するクラスの重み付平均ベクトルを引いたベクトルと、そのベクトルの転置ベクトルの積を求める。そして、積を求めたことにより得られる行列に、対応する学習パターンの重みを乗じ、その重みを乗じた行列の総和をクラスごとに求めることによって、クラスごとの重み付分散共分散行列が得られる。もし、ステップ S 4 0 1 で選択した矩形パターンが、1つの矩形で構成されるものである場合は、ベクトルデータが1つの要素しかもたないので、このステップ S 4 0 9 のクラス分散共分散行列算出は省略可能である。このステップ S 4 0 9 を省略した場合は、次のステップ S 4 1 0 の逆行列算出も省略する。

30

【 0 0 5 9 】

次のステップ S 4 1 0 では、ステップ S 4 0 9 で求めたクラスごとの重み付分散共分散行列の和を求め、求めた行列の逆行列を算出する。ここでは第 1 の実施形態とは異なり、ステップ S 4 0 9 で重み付の分散共分散行列を求めており、なおかつ全学習パターンの重みの総和は 1 となるようにしてあるので、事前確率を乗じる必要はない。ここで、逆行列が存在しない、即ち求めた行列の和が正則でない場合は、ステップ S 4 0 1 で選択した矩形パターンが適切でない可能性が高い。そのため、続くステップ S 4 1 1 でのパラメータ算出は飛ばし、ステップ S 4 1 2 の予測誤差記録（詳細は後述）において、その矩形パターンと共に、予測誤差 1 というデータを記録し、次に進むようにすればよい。

40

【 0 0 6 0 】

次に、ステップ S 4 1 1 において、まず、ステップ S 4 0 8 で求めた各クラスの重み付平均ベクトルと、ステップ S 4 1 0 の逆行列算出において求めた逆行列とを用いて、パタ

50

ーン識別用パラメータの1つであるクラス分離ベクトルを算出する。クラス分離ベクトルは、顔パターンの重み付平均ベクトルから非顔パターンの重み付平均ベクトルを引いたベクトルを求め、そのベクトルの左側からステップS 4 1 0で求めた逆行列をかけることによって求められる。

【0061】

次いで、もう1つのパターン識別用パラメータである識別閾値を、求めたクラス分離ベクトルと、データベース406に保持された、ステップS 4 0 1で選択した矩形パターンに対応するベクトルデータとを用いて求める。ここでは、まず第1の実施形態と同様に、データベース406に保持された、ステップS 4 0 1で選択した矩形パターンに対応する全ベクトルデータと、求めたクラス分離ベクトルとの内積を求める。この内積値について

10

【0062】

この識別に用いる識別閾値の決定は、第1の実施形態とは異なり、学習パターンの重みに応じて決定するようにする。具体的には、まず識別閾値を細かく振っていき、識別に失敗したパターンの重みを、全学習パターンについて、誤差の累積値として累積していき、その誤差の累積値が最小になる識別閾値を検索して決定する。ここで、識別に失敗したパターンとは、顔パターンであるのに、非顔パターンのクラスであると分類された学習パターンと、非顔パターンであるのに顔パターンであると分類された学習パターンである。上述した誤差の累積値が小さいということは、現状重視すべき学習パターンにおける識別の失敗が少ないということを示しており、ここにおいても、学習パターンの重みに基づいた学習を行っているといえる。

20

【0063】

ステップS 4 0 1で選択した矩形パターンが、1つの矩形で構成されるものであり、ステップS 4 0 9のクラス分散共分散行列算出を省略した場合は、上記とはパラメータ算出方法が異なる。まず顔パターンの重み付平均ベクトル（要素数は1なのでスカラー）から非顔パターンの重み付平均ベクトル（スカラー）を引いた時の符号のみを求める。次いで、その符号をデータベース406に保持された、ステップS 4 0 1で選択した矩形パターンに対応する全ベクトルデータ（スカラー）に乘じ、これを上記手法の内積値と同等のものとして扱い、識別閾値と、誤差の累積値を算出するようにすればよい。

30

【0064】

続いて、ステップS 4 1 2において、ステップS 4 1 1で求めたパターン識別用のパラメータと共に、識別閾値を決定した際の誤差の累積値を予測誤差として記録しておく。

【0065】

そして、ステップS 4 1 3において、ステップS 4 0 1で所定の矩形パターンの全てを選択したかどうかを判定する。全てを選択していなければ、学習パターンの選択履歴をクリアして、ステップS 4 0 1に戻り、新たな矩形パターンを選択していくようにする。所定の全ての矩形パターンが選択された場合、ステップS 4 1 4に進む。

【0066】

ステップS 4 1 4では、ステップS 4 1 2において記録した、ステップS 4 0 1で選択した全ての矩形パターンごとの予測誤差に基づき、予測誤差が最小である矩形パターンを1つ選択する。

40

【0067】

そして、ステップS 4 1 5において、ステップS 4 1 4で選択された矩形パターンと、それに対応するパターン識別用パラメータ、予測誤差から算出する識別スコア α を、弱判別方法に用いるパラメータとして順番に記録していく。識別スコア α は、予測誤差を ε とした時、 $\alpha = 1 / 2 \times \ln \{ (1 - \varepsilon) / \varepsilon \}$ として求められる。ここで、前式中の $\ln$ は自然対数を表わしている。

【0068】

次に、ステップS 4 1 6において、データベース403に保持されている学習パターン

50

が持つ重みを更新する。ここでは、データベース406に保持された、直前にステップS415で記録された矩形パターンに対応する全ベクトルデータに対し、同じく直前にステップS415で記録したパターン識別用パラメータ（クラス分離ベクトル、及び識別閾値）を用いて実際に識別を行う。

【0069】

そして、識別に成功したパターンに対しては、そのパターンの重みに $\exp(-\alpha)$ を乗じ、失敗したパターンに対しては、そのパターンの重みに $\exp(\alpha)$ を乗じる。ここで、識別に成功したパターンとは、顔パターンであって、正しく顔パターンのクラスであると分類された学習パターンと、非顔パターンであって、正しく非顔パターンであると分類された学習パターンである。また、失敗したパターンとは、顔パターンであるのに、非顔パターンのクラスであると分類された学習パターンと、非顔パターンであるのに、顔パターンであると分類された学習パターンである。前式中の α は、直前のステップS415において求めた識別スコアである。上記のような重み更新後、データベース403内の、全ての学習パターンが持つ重みの総和が1となるように正規化を行う。

10

【0070】

最後に、ステップS417で、ステップS415で記録したパラメータの個数が、予め定めた個数に達しているかどうかを判定して、達していれば処理を終了する。達していなければ、矩形パターン選択履歴、学習パターンの選択履歴、データベース406のベクトル化データをクリアして、ステップS401に戻り、もう1度最初から矩形パターンを選択していくようにする。以上の手法により、パターン識別用のパラメータとして、所定個数の矩形パターンと、その矩形パターンに対応したそれぞれのパターン識別用パラメータを、各弱判別方法に用いるパラメータとして得ることができる。

20

【0071】

本実施形態においても、第1の実施形態と同様に、重み付きのパターンを利用した線形判別分析法を用いてクラス分離ベクトルを求めているが、本発明はこれに限るものではなく、その他の手法を用いても構わない。

【0072】

続いて、上記手法により求めた弱判別方法のパターン識別用のパラメータを用いた、複数の弱判別方法の組み合わせによるパターン識別方法を利用した、画像中の顔を検出する顔検出方法について、図5を用いて説明する。

30

【0073】

まず、ステップS501で、顔検出を行う画像を入力する。この画像は、上記説明したパターン識別用パラメータ学習時と同様のグレースケール画像である。グレースケール画像以外の画像に対して顔検出を行いたい場合は、まず学習時と同様のグレースケール画像に変換してから入力するようにすれば良い。

【0074】

次に、ステップS502において、第1の実施形態の顔検出方法で説明したインテグラルイメージを生成すると共に、2乗のインテグラルイメージも生成する。2乗のインテグラルイメージとは、画像始点（例えば画像の左上の画素）から、注目画素までを対角線とする矩形領域内の、各画素値の2乗の総和を、注目画素の画素値として表現するものである。これを利用すると、高速に任意の矩形領域内の、各画素値の2乗の総和を求めることができ、この2乗のインテグラルイメージと、通常のインテグラルイメージを用いることにより、高速に任意の矩形領域の標準偏差を求めることができる。

40

【0075】

次いで、ステップS503で、画像中の所定の点の中から、点を1つ定め、それを中心とする矩形領域を、生成したインテグラルイメージから切り出す。上記所定の点は、画像の全画素としても良いが、例えば1画素おきというように、粗くサンプリングするようにしても良い。切り出すサイズは任意だが、本実施形態では、学習時に利用した学習パターンと同じサイズになるように切り出しを行う。ここで、1つの点を定めた段階で、その点に顔が存在するかの判定に用いるスコアを0に設定する。

50

【 0 0 7 6 】

次に、ステップ S 5 0 4 において、ステップ S 5 0 1 で入力した画像の、ステップ S 5 0 3 で切り出した矩形領域に対応する領域内の平均画素値、及び標準偏差を算出する。平均は、ステップ S 5 0 2 で生成したインテグラルイメージを用いて、矩形領域内の画素値の総和を求めて、それを矩形領域の面積で割ることによって求められる。標準偏差は、ステップ S 5 0 2 で生成した 2 乗のインテグラルイメージを用いて矩形領域内の各画素値の 2 乗の総和を求め、それを矩形領域の面積で割ったものから、先に求めた平均の 2 乗を引くことで矩形領域内の画素の分散を求め、その分散の平方根を算出することで求められる。

【 0 0 7 7 】

10

次いで、ステップ S 5 0 5 では、識別用パラメータ群が保持されたデータベース 5 0 6 から、所定の順番で矩形パターンを 1 つ選択する。データベース 5 0 6 には、上記説明したパターン識別用パラメータ学習時の、ステップ S 4 1 5 の学習結果記録において記録した順の通りに、矩形パターンと、それに対応する識別用パラメータを予め保持しておき、ステップ S 5 0 5 では、その順に従って矩形パターンを 1 つ選択する。

【 0 0 7 8 】

ステップ S 5 0 7 では、ステップ S 5 0 3 で切り出したインテグラルイメージを用いて、ステップ S 5 0 5 で選択した矩形パターンに基づく矩形領域の総和を抽出する。

【 0 0 7 9 】

次に、ステップ S 5 0 8 において、ステップ S 5 0 7 で抽出した矩形領域の総和と、ステップ S 5 0 4 で求めた平均と標準偏差とを用い、それぞれの矩形領域の総和から、その矩形領域の面積に平均を乗じたものを引く。その値に標準偏差の逆数を乗じることによって正規化を行い、ステップ S 5 0 5 で選択した矩形パターンにおける、上記説明したパターン識別用パラメータ学習時のステップ S 4 0 5 と同じ順序で並べたベクトルを生成する。

20

【 0 0 8 0 】

そして、ステップ S 5 0 9 において、ステップ S 5 0 8 で生成したベクトルと、学習により予め生成してデータベース 5 0 6 に保持された、ステップ S 5 0 5 で選択した矩形パターンに対応するクラス分離ベクトルとの内積を演算する。ここで用いるクラス分離ベクトルは、予め所定値以上を顔パターンと分類するのが好適な方向になるようにしたものを用いる。ここで、ステップ S 5 0 5 で選択された矩形パターンが 1 つの矩形で構成され、上記説明したパターン識別用パラメータ学習時にステップ S 4 0 9 のクラス分散共分散行列算出を省略した場合を考える。その場合は、ステップ S 4 1 1 のパラメータ算出において求めた符号を、ステップ S 5 0 8 で生成したベクトル（スカラー）に乗じたものを、上記内積値とみなすようにすればよい。

30

【 0 0 8 1 】

続いてステップ S 5 1 0 では、ステップ S 5 0 9 で求めた内積値と、学習時に予め求めてデータベース 5 0 6 に保持しておいた、ステップ S 5 0 5 で選択した矩形パターンに対応する識別閾値との比較を行う。求めた内積値が識別閾値よりも高い場合には、ステップ S 5 0 3 で定めた点に顔が存在する可能性が高いと判定する。ここで、処理の高速化のために、ステップ S 5 0 9 で求めた内積値に基づいて、ステップ S 5 0 3 で定めた点に顔が存在しないと決定して処理を打ち切ってしまうもよい。

40

【 0 0 8 2 】

その場合、予めデータベース 5 0 6 に、ステップ S 5 0 5 で選択した矩形パターンに対応する打ち切り閾値を保持しておく。そして、ステップ S 5 0 9 で求めた内積値が、その打ち切り閾値以下であるならば、処理を打ち切り、ステップ S 5 1 4 に進むようにする。この打ち切り閾値は、上記説明したパターン識別用パラメータ学習時のステップ S 4 1 1 のパラメータ算出において、識別閾値とは別に求めておく。例えば学習パターンの内、顔パターンであるのに、非顔パターンのクラスであるとみなされて処理を打ち切られてしまう割合が、所定値以下になるといった条件に基づいて求めておくようにすればよい。

50

【 0 0 8 3 】

次のステップ S 5 1 1 では、ステップ S 5 1 0 において顔が存在する可能性が高いと判定された場合は、ステップ S 5 0 3 で 0 に設定したスコアに、予めデータベース 5 0 6 に保持しておいた、ステップ S 5 0 5 で選択した矩形パターンに対応する識別スコアを加算する。逆に、ステップ S 5 1 0 で顔が存在する可能性が高いと判定されなかった場合は、識別スコアを減算する。ここで、更なる処理の高速化のために、現状のスコアに基づいて、ステップ S 5 0 3 で定めた点に顔が存在しないと決定して処理を打ち切ってしまうてもよい。

【 0 0 8 4 】

その場合、予めデータベース 5 0 6 に、ステップ S 5 0 5 で選択した矩形パターンによるスコアリング終了後に対応するスコア打ち切り閾値を保持しておく。現状のスコアがそのスコア打ち切り閾値以下ならば、処理を打ち切り、ステップ S 5 1 4 に進むようにする。このスコア打ち切り閾値は、上記説明したパターン識別用パラメータ学習時に、データベース 4 0 3 に保持してある学習パターンに、それぞれスコア（初期状態では 0）を設定しておく。ステップ S 4 1 6 での学習パターンの重み更新において、ステップ S 5 1 1 と同様にスコアに識別スコアを加算もしくは減算していき、そのスコアを用いて、例えば上記打ち切り閾値と同様の条件に基づいて求めるようにすればよい。本実施形態では、現状のスコアに基づいて処理の打ち切りを行うが、非特許文献 1 で開示されているように、数個分のスコアの累積値に基づいて処理の打ち切りを行うようにしてもよい。

10

【 0 0 8 5 】

次に、ステップ S 5 1 2 では、ステップ S 5 0 5 において、データベース 5 0 6 に保持されている、全ての矩形パターンを選択したかどうかを判定する。全ての矩形パターンを選択したのであれば、ステップ S 5 1 3 に進み、そうでなければステップ S 5 0 5 に戻り、新たな矩形パターンを、所定の順で 1 つ選択していくという処理を繰り返す。

20

【 0 0 8 6 】

ステップ S 5 1 3 では、ステップ S 5 1 1 で加算もしくは減算していったスコアが所定値以上であれば、ステップ S 5 0 3 で定めた点に顔が存在すると判定して、その結果を記録しておく。判定に用いる所定値としては通常 0 を用いるが、求める顔検出方法の性能に応じて適切に定めればよい。

【 0 0 8 7 】

そしてステップ S 5 1 4 では、ステップ S 5 0 3 において、所定の点の全てを中心とする画像切り出しを行ったかどうかを判定して、行ったのであれば処理を終了する。そうでなければステップ S 5 0 3 に戻り、新たな所定の点を 1 つ定め、画像切り出しを行うという処理を繰り返す。

30

【 0 0 8 8 】

上記説明した処理により、入力された画像内の顔を検出することができる。この処理でも、第 1 の実施形態と同じように、学習時に利用した顔パターンと同じような向き、サイズの顔を検出することになるが、異なるサイズや方向の顔を検出したいような場合は、第 1 の実施形態で述べたような手法を用いればよい。

【 0 0 8 9 】

以上、第 2 の実施の形態は、上記第 1 の実施形態の変形として、第 1 の実施形態と同様に、入力データとして 2 次元画像データを用いた 2 クラス分類のパターン識別を行う。その際、ブースティング手法を用いて構築する複数の弱判別方法の組み合わせによる、さらに高精度なパターン識別に用いるパラメータ学習方法、及びそれを用いたパターン識別方法の例として説明した。

40

【 0 0 9 0 】

（第 3 の実施の形態）

第 3 の実施形態は、上記第 2 の実施形態の変形として、第 2 の実施形態と同様に、入力データとして 2 次元画像データを用いた 2 クラス分類のパターン識別を行う。その際に、ブースティング手法を用いて構築する複数の弱判別方法の組み合わせによる、さらに高精

50

度なパターン識別に用いるパラメータ学習方法、及びそれを用いたパターン識別方法を示す。

【0091】

第2の実施形態では、抽出した総和により張られる特徴空間において、線形識別を行うようにしていた。これに対し、本実施形態では、抽出した総和により張られる特徴空間内の点を、非線形変換により新たな特徴空間に写像し、その新たな特徴空間において線形識別を行う、いわゆる非線形識別を行う。そのような例として、サポートベクターマシン（以下、SVM）を用いた例を説明する（SVMの詳細については、非特許文献5を参照）。

【0092】

図6に、第3の実施形態におけるパターン識別用パラメータ学習方法の処理の流れを示す。以下、このパターン識別用パラメータ学習方法について図6を用いて説明する。

【0093】

ステップS601～607の処理と、第2の実施形態におけるステップS401～S407までの処理とは、ステップS605とステップS405とでベクトル化が異なる以外は、同様の処理である。よってステップS605のベクトル化以外の処理の説明は省略する。なお、本実施形態では、第2の実施形態において説明した、累積スコアによる演算の打ち切りに用いるスコア打ち切り閾値も同時に学習する。そのため、データベース603に保持してある全学習パターンに累積スコアを持たせ、最初にそれを0に初期化しておくようにする。

【0094】

ステップS605では、第2の実施形態のステップS405と同様に、ステップS604で求めたそれぞれの矩形領域の総和を用いてベクトルを生成する。それをステップS601で選択した矩形パターンに対応するベクトルデータとして、データベース606に保持する。ただし、第2の実施形態では、学習パターンの輝度値の平均と標準偏差のみを用いて、求めたそれぞれの矩形領域の総和を正規化してからベクトル化するようにしていたが、ここではさらに、それらを各矩形領域の面積で割ってからベクトル化するようにする。

【0095】

次に、ステップS608では、データベース606に保持された、ステップS601で選択した矩形パターンに対応する、顔パターンと非顔パターンとのそれぞれから得られたベクトルデータ群を分離するように、SVMの学習を行う。

【0096】

本実施形態では、SVMにおいて、ガウシアンカーネル $\exp(-|x - x_k|^2 / \sigma^2)$ を用いた非線形SVMを利用する。ガウシアンカーネルを用いた非線形SVMの学習では、ステップS601で選択した矩形パターンに対応するベクトルデータと同次元の、少なくとも1つのサポートベクター x_k と、それに対応する係数 a_k 、及びバイアス値 b が得られる。即ち、ステップS601で選択された矩形パターンを構成する矩形領域の個数分の次元数となる。

【0097】

通常のSVMでは、 n 個のサポートベクターが得られた場合には、識別時に $\sum [a_k \cdot \exp(-|x - x_k|^2 / \sigma^2)] + b$ （ $\sum$ は1から n までの総和を意味する）を算出し、その符号に応じて2値的な判定を行う。しかしながら、本実施形態では、 $\sum [a_k \cdot \exp(-|x - x_k|^2 / \sigma^2)]$ の値に応じた連続的な確信度を出力する手法を用いるため、ここで得られるバイアス値 b は無視しても構わない。

【0098】

また、SVMの学習時に必要となる、ソフトマージンの程度を決める定数や、ガウシアンカーネルの σ に関しては、SVMの学習を行うベクトルデータ群のレンジや、問題の複雑さに応じて適当に定めてやればよい。この学習では、第2の実施形態での学習のように、学習パターンが持つ重みを、学習に直接的に用いることが困難である。そこで、学習時

10

20

30

40

50

に学習パターンの重みを反映させたい場合は、例えばその重みに比例する確率に応じて、SVM学習用のデータとして利用するか否かを決めるというようにすればよい。

【0099】

次のステップS609では、第2の実施形態の顔検出方法におけるステップS510の閾値判定において説明した打ち切り閾値と同様の、後述の顔検出方法において処理の高速化を行うために用いる打ち切り閾値の算出を行う。具体的には、データベース606に保持された、ステップS601で選択した矩形パターンに対応する全ベクトルデータ x を対象とする。

【0100】

これに対して、ステップS608で得られた、 n 個のサポートベクター x_k と、それに対応する係数 a_k 、及び学習時に用いたパラメータ σ を用いて、 $\sum [a_k \cdot \exp(-|x - x_k|^2 / \sigma^2)]$ を算出する。この値を中間出力値と呼ぶ。そして、全ベクトルデータから得られた中間出力値に基づいて、この中間出力値が所定値以下であるならば、そのパターンを非顔パターンのクラスであるとみなして処理を打ち切るための、打ち切り閾値を求める。

10

【0101】

この打ち切り閾値は、第2の実施形態におけるステップS510で説明した打ち切り閾値と同様に求められる。例えば、学習パターンの内、顔パターンであるのに非顔パターンのクラスであるとみなされて、処理を打ち切られてしまう割合が所定値以下であるといった条件に基づいて求めればよい。

20

【0102】

本実施形態では、上記条件の他に、学習パターンの内、非顔パターンであって、正しく非顔パターンであるとみなされ、処理が打ち切られる割合が第2の所定値以上であるという第2の条件も同時に用いる。そして、両方の条件を同時に満たし、第2の条件における打ち切られる割合が最も大きい値を探索して決定するようにしている。もし両方の条件を同時に満たす値が存在しない場合は、識別時に打ち切りを行わないようにすればよい。

【0103】

ステップS610では、ステップS608により得られたパラメータを用いて、予測誤差を算出しそれを記録しておく。具体的には、ステップS609と同様に、まずデータベース606に保持された、ステップS601で選択した矩形パターンに対応する全ベクトルデータ x を対象とする。これに対して、ステップS608において得られた、 n 個のサポートベクター x_k と、それに対応する係数 a_k 、及び学習時に用いたパラメータ σ を用いて、全中間出力値を算出する。

30

【0104】

そして次に、この全中間出力値と各学習パターンの重みを用い、クラスごとに、重み付ヒストグラムを作成する。このクラスごとの重み付ヒストグラムは、1つのピンに対応する中間出力値範囲内である当該クラスのパターンが持つそれぞれの重みを累積したものが、そのピンの値として表現されたものである。そして、作成したクラスごとの重み付ヒストグラムを用いて、予測誤差を算出し、それを記録しておく。予測誤差は、作成したクラスごとの重み付ヒストグラムの、 k 番目のピン（ピンに対応する中間出力値範囲は、クラス間で同一でなければならない）の値を $S_{k,c}$ （ c はクラスを表す）、全ピン数を n とした時、 $2 \cdot \sum [(S_{k,Face} \cdot S_{k,NonFace})^{1/2}]$ として求められる。

40

【0105】

そして、ステップS611では、第2の実施形態において説明したステップS413と同様に、ステップS601で所定の矩形パターンの全てを選択したかどうかを判定する。全てを選択していなければ、学習パターンの選択履歴をクリアして、ステップS601の矩形パターン選択に戻り、新たな矩形パターンを選択していくようにする。所定の矩形パターンが全て選択された場合、ステップS612に進む。

【0106】

50

ステップ S 6 1 2 では、ステップ S 6 1 0 において記録した、ステップ S 6 0 1 で選択した全ての矩形パターンごとの予測誤差から、予測誤差が最小である矩形パターンを 1 つ選択する。

【 0 1 0 7 】

そして、ステップ S 6 1 3 において、パターン識別用のパラメータを、弱判別方法に用いるパラメータとして順番に記録していく。まず、ステップ S 6 1 2 で選択された矩形パターンと、その矩形パターン選択時に、ステップ S 6 0 8 で得られたサポートベクター x_k と、それに対応する係数 a_k 、及び学習時に用いたパラメータ σ を記録する。更に、ステップ S 6 0 9 で得られた打ち切り閾値、ステップ S 6 1 0 で生成した、クラスごとの重み付ヒストグラムから生成する識別スコアテーブル T を記録する。

10

【 0 1 0 8 】

識別スコアテーブル T は、中間出力値が、クラスごとの重み付ヒストグラムのビンに対応する中間出力値範囲である時に、各クラスの重み付ヒストグラムの、そのビンの値に対して、出力値を出力するテーブルである。ビンの値、 $S_{i, Face}$ 、及び $S_{i, Non Face}$ を用いて、出力値 $\alpha = 1 / 2 \cdot \ln \{ (S_{i, Face} + \varepsilon) / (S_{i, Non Face} + \varepsilon) \}$ という値を出力する。

【 0 1 0 9 】

次に、ステップ S 6 1 4 において、データベース 6 0 3 に保持されている学習パターンが持つ重みを更新する。また同時に、第 2 の実施形態の顔検出方法において説明した、スコアによる演算打ち切りのための、スコア打ち切り閾値を算出する。そのため、各学習パターンが持つ累積スコアの更新を行い、それに基づいて、スコア打ち切り閾値の算出も行う。

20

【 0 1 1 0 】

重みの更新では、データベース 6 0 6 に保持された、直前にステップ S 6 1 3 の学習結果記録において記録された矩形パターンに対応する全ベクトルデータを用いる。これに対し、同じく直前にステップ S 6 1 3 の学習結果記録において記録したパターン識別用パラメータ (サポートベクター x_k と、それに対応する係数 a_k 、及び学習時に用いたパラメータ σ 、識別スコアテーブル T) を用いる。まず中間出力値である $\sum [a_k \cdot \exp (- | x - x_k |^2 / \sigma^2)]$ を算出し、その値を基に、識別スコアテーブル T を参照して出力値 α を求める。

30

【 0 1 1 1 】

そして、その学習パターンが顔パターンであれば、その重みに $\exp (- \alpha)$ を乗じ、非顔パターンであれば $\exp (\alpha)$ を乗じる。累積スコアの更新では、上記求めた出力値 α を、学習パターンの持つ累積スコアに加算していく。スコア打ち切り閾値の算出は、ステップ S 6 0 9 と同様の条件に基づいて、条件を満たすスコア打ち切り閾値を検索し、条件を満たすスコア打ち切り閾値が存在するならば、その値をステップ S 6 1 3 で記録したパターン識別用パラメータに追加して記録するようにする。ここでも第 2 の実施形態において説明したのと同様に、上記のような重み更新後、データベース 6 0 3 の学習パターン内の、全ての学習パターンが持つ重みの総和が 1 となるように正規化を行う。

【 0 1 1 2 】

40

最後に、ステップ S 6 1 5 では、第 2 の実施形態のステップ S 4 1 7 と同様に、ステップ S 6 1 3 で、弱判別方法のパラメータを記録した個数が、予め定めた所定個数に達しているかどうかを判定する。所定個数に達していれば処理を終了する。達していなければ、矩形パターン選択履歴、学習パターンの選択履歴、データベース 6 0 6 のベクトル化データをクリアして、ステップ S 4 0 1 に戻り、もう 1 度最初から矩形パターンを選択していくようにする。

【 0 1 1 3 】

以上の手法により、パターン識別用のパラメータとして、所定個数の矩形パターンと、その矩形パターンに対応したそれぞれのパターン識別用パラメータを、各弱判別方法のパラメータとして得ることができる。

50

【 0 1 1 4 】

このように、本手法では、第 1、第 2 の実施形態において説明したような線形識別の手法ではなく、非線形識別の手法にも拡張可能である。本実施形態では、ガウシアンカーネルを用いた非線形 SVM による手法を説明したが、これに限るものではなく、例えば多項式カーネルを利用した SVM や、非線形判別分析といった、その他の非線形手法や、k 最近傍法による手法にも適用可能である。

【 0 1 1 5 】

続いて、上記手法により求めた弱判別方法のパターン識別用のパラメータを用いた、複数の弱判別方法の組み合わせによるパターン識別方法を利用した、画像中の顔を検出する顔検出方法について、図 7 を用いて説明する。

10

【 0 1 1 6 】

ステップ S 7 0 1 ~ 7 0 7 は、第 2 の実施形態の顔検出方法において説明した、ステップ S 5 0 1 ~ 5 0 7 と同様の処理であるので説明を省略する。

【 0 1 1 7 】

ステップ S 7 0 8 では、ステップ S 7 0 7 で抽出した矩形領域の総和と、ステップ S 7 0 4 で求めた平均と標準偏差を用い、それぞれの矩形領域の総和から、その矩形領域の面積に平均を乗じたものを引き、それに標準偏差の逆数を乗じることによって正規化を行う。その値をさらに矩形領域の面積で割ったものを、ステップ S 7 0 5 で選択した矩形パターンにおける、上記説明したパターン識別用パラメータ学習時のステップ S 6 0 5 と同じ順序で並べたベクトルを生成する。

20

【 0 1 1 8 】

そして、ステップ S 7 0 9 において、中間出力値 $\sum [a_k \cdot \exp(-|x - x_k|^2 / \sigma^2)]$ を算出する。ここで、ステップ S 7 0 8 で生成したベクトルと、学習により予め生成して保持しておいた、ステップ S 7 0 5 で選択した矩形パターンに対応するサポートベクター x_k と、それに対応する係数 a_k 、及び学習時に用いたパラメータ σ を用いる。

【 0 1 1 9 】

続いて、ステップ S 7 1 0 では、ステップ S 7 0 9 で求めた中間出力値と、学習時に予め求めておき、データベース 7 0 6 に保持しておいた、ステップ S 7 0 5 で選択した矩形パターンに対応する打ち切り閾値との比較を行う。求めた中間出力値が打ち切り閾値よりも低い場合に、ステップ S 7 0 3 で定めた点に顔が存在しないと決定して処理を打ち切り、ステップ S 7 1 4 に進むようにする。

30

【 0 1 2 0 】

次のステップ S 7 1 1 では、ステップ S 7 0 9 で求めた中間出力値と、データベース 7 0 6 に保持しておいた、ステップ S 7 0 5 で選択した矩形パターンに対応する識別スコアテーブル T を用いて出力値 α を求める。そして、その値 α を、ステップ S 7 0 3 で 0 に設定したスコアに加算する。また、この加算を行った後のスコアと、学習時に予め求めておいた、ステップ S 7 0 5 で選択した矩形パターンによるスコアリング終了時に対応するスコア打ち切り閾値との比較を行う。その結果、加算を行った後のスコアが、スコア打ち切り閾値よりも低い場合に、ステップ S 7 0 3 で定めた点に顔が存在しないと決定して処理を打ち切り、ステップ S 7 1 4 に進むようにする。

40

【 0 1 2 1 】

ステップ S 7 1 2 から、ステップ S 7 1 4 までの処理は、第 2 の実施形態の顔検出方法において説明した、ステップ S 5 1 2 から、ステップ S 5 1 4 までの処理と同様の処理であるので説明を省略する。

【 0 1 2 2 】

上記説明した処理により、入力された画像内の顔を検出することができる。この処理でも、第 1 の実施形態における顔検出方法と同じように、学習時に利用した顔パターンと同じような向き、サイズの顔を検出することになる。異なるサイズや方向の顔を検出したいような場合は、第 1 の実施形態で述べたような手法を用いればよい。

50

【 0 1 2 3 】

以上、第3の実施の形態は、上記第2の実施形態の変形として、第2の実施形態と同様に、入力データとして2次元画像データを用いた2クラス分類のパターン識別する例を説明した。その際、ブースティング手法を用いて構築する複数の非線形識別を行う弱判別方法の組み合わせによる、さらに高精度なパターン識別に用いるパラメータ学習方法、及びそれを用いたパターン識別方法を示した。

【 0 1 2 4 】

(第4の実施の形態)

第4の実施形態では、上記第2の実施形態の変形として、入力データとして2次元画像データを用い、その画像データが、所定のカテゴリの何れに属するか、若しくは何れにも属さないか識別する。つまり入力された画像データを、複数の所定のカテゴリであるというクラス、もしくは何れのカテゴリでもないクラスに分類する、多クラス分類のパターン識別を行う。そこで、ブースティング手法を用いて構築する複数の弱判別方法の組み合わせによる、パターン識別に用いるパラメータ学習方法、及びそれを用いたパターン識別方法を示す。

10

【 0 1 2 5 】

複数の所定のカテゴリとしては、第3の実施形態までと同様の人物の顔や、その他、車両・草花といった様々なものが考えられるが、本実施形態では、ある個人の顔を1つのカテゴリとみなし、所定の登録人数(例えば3人)分のカテゴリを用いた場合の例を示す。

【 0 1 2 6 】

図8に、第4の実施形態におけるパターン識別用パラメータ学習方法の処理の流れを示す。以下、このパターン識別用パラメータ学習方法について図8を用いて説明する。

20

【 0 1 2 7 】

まず、ステップS801において、所定の領域パターンの中から、1つの領域パターンを選択する。第3の実施形態まででは、パターン識別をコンピュータ上で高速に実行するために、いくつかの矩形領域から構成されている矩形パターンを用いた。この領域パターン、つまり総和を抽出する、いくつかの所定部分の組み合わせとしては、例えば所定同心円状上のデータの総和を求めるといった、その他の任意の部分から構成されているものであっても構わない。

【 0 1 2 8 】

本実施形態では、図10に示すように、3つの領域1001~1003から構成される領域パターンを、予め複数用意しておき、ここでその中から1つを選択して用いるようにする。ここで、領域1001は任意のサイズ・位置の矩形領域であり、領域1002及び領域1003は、任意のサイズ・位置の円形領域(2つの領域は同一の領域にはならない)である。

30

【 0 1 2 9 】

次のステップS802では、複数の学習パターンが保持されたデータベース803から、学習パターンを入力する。データベース803には、登録者であるA、B、Cの3名の、それぞれ複数の顔パターンと、登録者でない人物の複数の顔パターンを、そのラベル(A、B、Cの何れであるか、もしくは登録者でないという情報)、及びパターンの重みと共に予め保持しておく。パターンの重みは、第2の実施形態と同様に、初期状態の重みを設定しておく。

40

【 0 1 3 0 】

この初期状態の重みの設定では、A、B、Cそれぞれの学習パターンの重みを同一にしておくことが好ましい。本実施形態でも、データベース803に保持しておく顔パターンは、グレースケール画像であり、各画素値が輝度値として表現されたものである。また、全顔パターンは、顔の向きやサイズをある程度正規化しておき、一定の画像サイズにして保持しておく。

【 0 1 3 1 】

続いて、ステップS804において、ステップS802で入力した学習パターンから、

50

ステップS 8 0 1で選択した領域パターンに基づき、その領域パターンを構成する各領域内の、輝度値の総和を抽出する。

【0 1 3 2】

次に、ステップS 8 0 5において、ステップS 8 0 4で求めたそれぞれの領域の総和を単純に並べたベクトルを生成し、そのベクトルを、ステップS 8 0 1で選択した領域パターンに対応するベクトルデータとして、データベース8 0 6に保持する。本実施形態では、第2, 3の実施形態のような、学習パターンの平均値や標準偏差を用いた総和の正規化を行わないが、正規化を行ってからベクトル化するようにしても構わない。

【0 1 3 3】

ステップS 8 0 7では、データベース8 0 3に保持されている学習パターンの全てを選択したかどうかを判定して、全てを選択していなければ、ステップS 8 0 2に戻り、新たな学習パターンを、データベース8 0 3から選択していくようにする。全ての学習パターンが選択された場合、ステップS 8 0 8に進む。

【0 1 3 4】

ステップS 8 0 8では、データベース8 0 6に保持された、ステップS 8 0 1で選択した領域パターンに対応するベクトルデータと、そのラベルとを用いて、ニューラルネットワーク（以下、NN）の学習を行う。NNとしては、多クラス識別が可能であればどのようなタイプのネットワークでも構わないが、本実施形態では、図1 1に示すような、3層フィードフォワード型であるNNを用いる。

【0 1 3 5】

このNNは、入力層1 1 0 1の入力ノード1 1 0 2が3つ（領域パターンを構成する領域数によって決まる）、出力層1 1 0 5の出力ノード1 1 0 6も3つ（分類すべきカテゴリ数によって決まる）である。中間層1 1 0 3の中間ノード1 1 0 4の数は、予め所定の個数にしても良いが、問題に応じて実験的に適切な個数にするのが好ましい。

【0 1 3 6】

図1 1に示すように中間ノード1 1 0 4は入力ノード1 1 0 2に全結合しており、出力ノード1 1 0 6も中間ノード1 1 0 4に全結合している。NNの学習では、入力層1 1 0 1の3つのノード1 1 0 2に、データベース8 0 6に保持された、ステップS 8 0 1で選択した領域パターンに対応するベクトルデータの、それぞれの要素の値を入力する。そして、入力したベクトルデータのラベルに対応する出力ノード1 1 0 6の出力が1、それ以外の出力ノード1 1 0 6の出力は0となるように、誤差逆伝播法を用いて、いわゆる教師有り学習を繰り返し行う。

【0 1 3 7】

ちなみに、A、B、C何れのカテゴリにも属さないベクトルデータの場合は、3つの出力ノードの出力が、全て0になるような教師有り学習をすればよい（多層フィードフォワード型NNと、誤差逆伝播法等の、NNの教師有り学習方法に関しては、S. Haykin, "Neural Networks A Comprehensive Foundation 2nd Edition", Prentice Hall, pp. 156 - 255, July 1998を参照）。第2の実施形態等と同様に、学習パターンが持つ重みを学習結果に反映させるためには、学習の際にパターンを提示する確率を、学習パターンが持つ重みに比例した確率となるようにしてやればよい。

【0 1 3 8】

次のステップS 8 0 9では、ステップS 8 0 8での学習により得られたNNのパラメータを用いて、実際にNNでの演算を行い、その結果から得られる予測誤差を算出しそれを記録しておく。具体的には、まずデータベース8 0 6に保持された、ステップS 8 0 1で選択した領域パターンに対応する全ベクトルデータxに対して、ステップS 8 0 8で得られたNNのパラメータを用いて、実際にNNでの演算を行う。そして、NNの全出力結果（3つの出力ノードの出力値）を算出する。

【0 1 3 9】

次に、この全出力結果を用い、それぞれの判定閾値（出力値がこの値以上であれば、当

10

20

30

40

50

該ノードに対応するクラスである確率が高いと判定する閾値)を細かく振りながら、実際の判定を行っていく。実際の判定では、まずそれぞれの出力ノードの出力値と判定閾値とを比較する。もし出力値が判定閾値を超えたノードが無い場合は、何れのクラスにも属しないと判定する。また、出力値が判定閾値を超えたノードが1つである場合は、そのノードに対応するクラスであると判定する。そして、出力値が判定閾値を超えたノードが多数である場合は、それらの内で最大の出力値であるノードに対応するクラスであると判定する。最大の出力値の比較においては、出力値の絶対値で比較してもよいし、判定閾値からの出力値の差で比較してもよい。

【0140】

この判定により、誤った判定をしてしまった(Aのクラスに属するパターンであるのに、A以外であると判定してしまった等)学習パターンの重みを、予測誤差として累積していく。そして、上記予測誤差を、様々な判定閾値の組み合わせにおいて算出し、得られた予測誤差が最小となる3つの判定閾値と、その時の予測誤差を記録しておく。

【0141】

そして、ステップS810では、第2の実施形態におけるステップS413と同様に、ステップS801で所定の領域パターンの全てを選択したかどうかを判定する。全てを選択していなければ、学習パターンの選択履歴をクリアして、ステップS801に戻り、新たな領域パターンを選択していくようにする。所定の全ての領域パターンが選択された場合、ステップS811に進む。

【0142】

ステップS811では、ステップS809で記録した、ステップS801で選択した全ての領域パターンごとの予測誤差から、予測誤差が最小である領域パターンを1つ選択する。

【0143】

そして、ステップS812において、ステップS811で選択された最小誤差領域パターンと、その領域パターン選択時にステップS808で得られたNNのパラメータを、弱判別方法のパラメータとして順番に記録していく。ここで、NNのパラメータは、NNの結合係数と、中間層のノード数である。更に、ステップS809で得られた各判定閾値と、予測誤差から算出する識別スコア α を、弱判別方法のパラメータとして順番に記録していく。識別スコア α は、第2の実施形態と同様に、予測誤差を ε とした時、 $\alpha = 1 / 2 \times \ln \{ (1 - \varepsilon) / \varepsilon \}$ として求められる。

【0144】

ステップS813では、データベース803に保持されている学習パターンが持つ重みを更新する。重みの更新では、データベース806に保持された、直前のステップS812で記録された最小誤差領域パターンに対応する全ベクトルデータを対象とする。これに対し、同じく直前のステップS812の学習結果記録において記録したパターン識別用パラメータ(NNの結合係数、中間層のノード数、各判定閾値)を用い、実際にNNによる判定を行う。そして、その学習パターンが正しい識別をされたなら、識別スコア α を用いて、その重みに $\exp(-\alpha)$ を乗じ、誤った識別をされたなら $\exp(\alpha)$ を乗じる。

【0145】

最後に、ステップS814では、第2の実施形態のステップS417と同様に、ステップS812で弱判別方法のパラメータを記録した個数が、予め定めた所定個数に達しているかどうかを判定する。所定個数に達していれば処理を終了し、達していなければ、領域パターン選択履歴、学習パターンの選択履歴、データベース806内のベクトル化データをクリアして、ステップS801に戻り、もう1度最初から矩形パターンを選択していくようにする。以上の手法により、パターン識別用のパラメータとして、所定個数の領域パターンと、その領域パターンに対応したそれぞれのパターン識別用パラメータを、各弱判別方法のパラメータとして得ることができる。

【0146】

続いて、上記手法により求めた弱判別方法のパターン識別用のパラメータを用いた、複

10

20

30

40

50

数の弱判別方法の組み合わせによるパターン識別方法を利用した個人識別方法について、図9を用いて説明する。この個人識別方法によれば、入力された顔画像に対し、それが登録されたA、B、Cの何れの人物であるのか、若しくは登録された人物ではないのかを検出する。

【0147】

まず、ステップS901において、識別処理対象となる顔画像として1つの顔パターンを入力する。この時、識別対象となるクラス(ここでは、A、B、Cの3クラス)のそれぞれのスコアを0に初期化しておく。ここで入力する画像は、上記説明したパターン識別用パラメータ学習時と同様のグレースケール画像であり、予め学習時と同じ大きさに正規化されたものである。グレースケール画像以外の顔画像による個人識別を行いたい場合は、まず学習時と同様の画像サイズであるグレースケール画像に変換してから入力するようにすればよい。

10

【0148】

続いて、ステップS902では、識別用パラメータ群データベース903から、所定の順番で領域パターンを1つ選択する。データベース903には、上記説明したパターン識別用パラメータ学習時の、ステップS812で記録した順の通りに、領域パターンと、それに対応する識別用パラメータとを予め保持しておく。ステップS902では、その順に従って領域パターンを1つ選択する。

【0149】

ステップS904では、ステップS901の画像入力で入力された画像の、ステップS902の領域パターン選択において選択した領域パターンに基づく領域の輝度値の総和を抽出する。

20

【0150】

次に、ステップS905において、ステップS904で抽出した各領域の総和を、上記説明したパターン識別用パラメータ学習時のステップS805と同じ順序で並べたベクトルを生成する。

【0151】

そして、ステップS906において、出力層の全ノードの出力値を演算する。その際、ステップS905で生成したベクトルと、学習により予め生成して、データベース903に保持しておいた、ステップS902で選択した領域パターンに対応するNNの結合係数と、その中間層のノード数であるNNとを用いる。

30

【0152】

続いて、ステップS907では、まずステップS906で求めた全ノードの出力値と、学習時に予め求めておいた、ステップS902で選択した領域パターンに対応する、各出力ノードの判別閾値との比較を行う。上記説明したパターン識別用パラメータ学習時のステップS809の予測誤差記録で説明したのと同様の判定を行う。

【0153】

そして、判定されたクラスのスコアに、予めデータベース903に保持しておいた、ステップS902で選択した領域パターンに対応する識別スコアを加算し、判定されなかったクラスのスコアからは、識別スコアを減算していく。何れのクラスでもない判定された場合は、全てのスコアから、識別スコアを減算するようにする。

40

【0154】

次に、ステップS908では、ステップS902において、データベース903に保持されている、全ての領域パターンを選択したかどうかを判定して、全ての領域パターンを選択したのであれば、ステップS909の結果出力に進む。そうでなければステップS902に戻り、新たな領域パターンを、所定の順で1つ選択していくという処理を繰り返す。

【0155】

ステップS909では、ステップS907で加算もしくは減算していった、それぞれのクラスのスコアを用い、ステップS901で入力された顔パターンがどのクラスに属する

50

ものであるのかを判定し、その結果を出力する。この判定では、まずそれぞれのスコアの内、最大の出力値であるクラスの出力値を1つ選択する。そして、そのスコアが所定値以上であれば、当該クラスに属すると判定し、そうでなければ、何れのクラスにも属しないと判定する。所定値は、通常0をその値として用いるが、求める顔画像による個人識別方法の性能に応じて適切に定めればよい。

【0156】

以上、第4の実施の形態は、上記第2の実施形態の変形として、入力データとして2次元画像データを用い、その画像データが、所定のカテゴリの何れに属するか、若しくは何れにも属さないかを識別する。つまり入力された画像データを、複数の所定のカテゴリであるというクラス、もしくは何れのカテゴリでもないクラスに分類する、多クラス分類の

10

【0157】

以上説明した第4の実施形態のように、多クラス分類のパターン識別において、多クラスの識別方法を複数の弱判別方法の1つとして用いることにより、多クラス分類のパターン識別を実現してもよい。また、第2の実施形態や、第3の実施形態に示したような、2クラス識別のパターン識別方法を複数用いることにより、多クラスの識別を行っても構わない。

【0158】

20

また、上記第2から第4の実施形態においては、同一の弱判別方法を複数用いたパターン識別方法を説明したが、それぞれの弱判別方法が混載するような場合でも実現可能である。また、これとは異なる弱判別方法との組み合わせにおいても実現可能である。

【0159】

図12は、上述した各実施形態に係るパラメータ学習装置およびパターン識別装置のハードウェア構成を示すブロック図である。

【0160】

同図において、CPU1201は、フローチャートにつき説明した処理手順を含む各種プログラムを実行し、システムバス1208により接続された装置各部を制御する。RAM1202は、データやプログラムを一時記憶する。ROM1203は、固定的なデータやプログラムを記憶する。キーボード1204は、文字を入力したり、機能を選択するためのキーを備えている。更に画面上の位置を指示するためのポインティングデバイスとしてマウスが備えられている。

30

【0161】

ディスプレイ1205は、識別対象のパターンやその識別結果を表示したり、ユーザインターフェース(UI)を提供する。ハードディスクドライブ(HDD)1206は、プログラムやデータを永続的に記憶する装置である。更にDVD、CD、FDなどの記憶媒体を用いる記憶装置を設けてもよい。通信I/F1207は、LANやインターネットなどを經由して装置外部と通信するためのインタフェースである。システムバス1208は、上述した装置各部を接続し、装置間でデータやアドレス、制御信号を送受信する媒体として利用される。

40

【0162】

フローチャートにつき説明した処理手順を含む各種プログラムは、ROM1203に記憶されていてもよいし、HDD1206からRAM1202へ、処理に先立ってあるいは処理の実行中に必要に応じてロードされるようにしてもよい。

【0163】

また、識別対象パターンは、HDD1206などの記憶装置から読み出したり、通信I/F1207を介して外部の撮像装置から受信するなどして、得ることができる。

【0164】

なお、本発明は、複数の機器から構成されるシステムに適用しても、一つの機器からな

50

る装置に適用してもよい。

【0165】

また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体が供給されたシステムあるいは装置のコンピュータが、そのプログラムコードを読み出し実行することでも達成される。

【0166】

この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステムなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【0167】

さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれる場合がある。そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行うことで前述した実施形態の機能が実現される場合も本発明に含まれる。

【図面の簡単な説明】

【0168】

【図1】第1の実施形態におけるパターン識別用パラメータ学習処理を示すフローチャートである。

【図2】総和抽出範囲の例を示す図である。

【図3】第1の実施形態における顔検出処理を示すフローチャートである。

【図4】第2の実施形態におけるパターン識別用パラメータ学習処理を示すフローチャートである。

【図5】第2の実施形態における顔検出処理を示すフローチャートである。

【図6】第3の実施形態におけるパターン識別用パラメータ学習処理を示すフローチャートである。

【図7】第3の実施形態における顔検出処理を示すフローチャートである。

【図8】第4の実施形態におけるパターン識別用パラメータ学習処理を示すフローチャートである。

【図9】第4の実施形態における顔検出処理を示すフローチャートである。

【図10】領域パターンの例を示す図である。

【図11】ニューラルネットワークの例を示す図である。

【図12】パラメータ学習装置およびパターン識別装置のハードウェア構成を示すブロック図である。

【符号の説明】

【0169】

102 学習パターンのデータベース

105 ベクトル化データのデータベース

307 識別用パラメータのデータベース

403 学習パターンのデータベース

406 ベクトル化データのデータベース

506 識別用パラメータのデータベース

603 学習パターンのデータベース

606 ベクトル化データのデータベース

706 識別用パラメータのデータベース

803 学習パターンのデータベース

806 ベクトル化データのデータベース

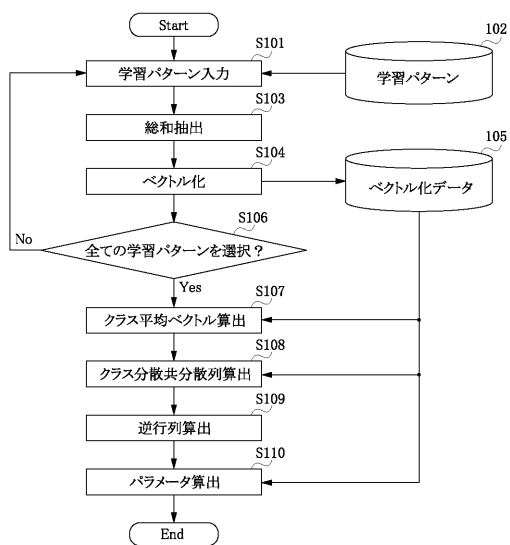
903 識別用パラメータのデータベース

40

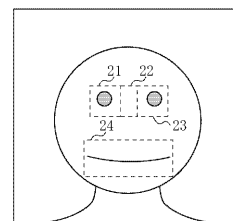
50

1 2 0 1 C P U
1 2 0 2 R A M
1 2 0 3 R O M
1 2 0 4 キーボード
1 2 0 5 ディスプレイ
1 2 0 6 H D D
1 2 0 7 通信 I / F
1 2 0 8 バス

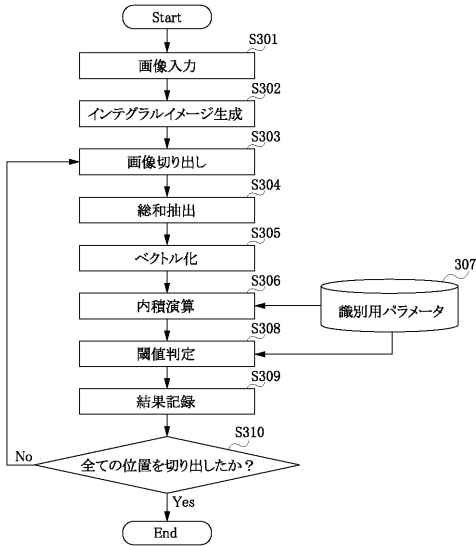
【図 1】



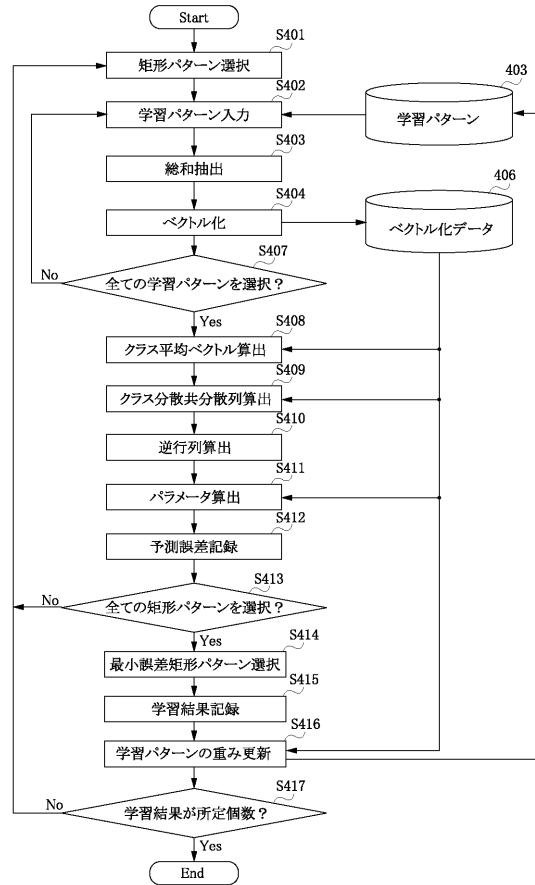
【図 2】



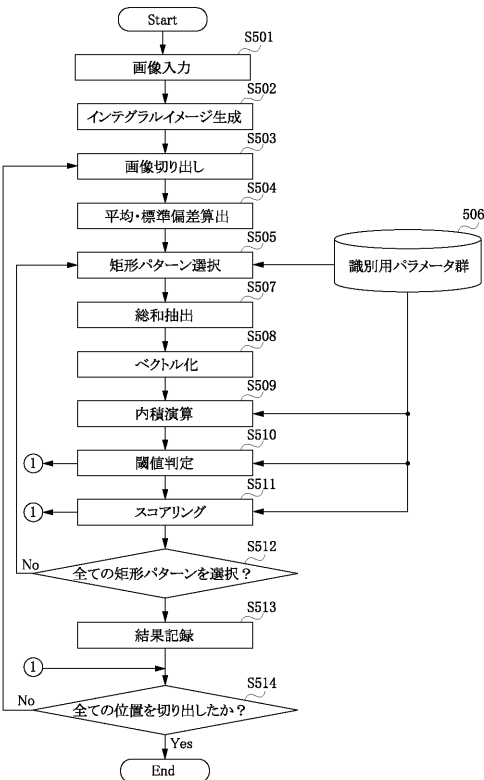
【 図 3 】



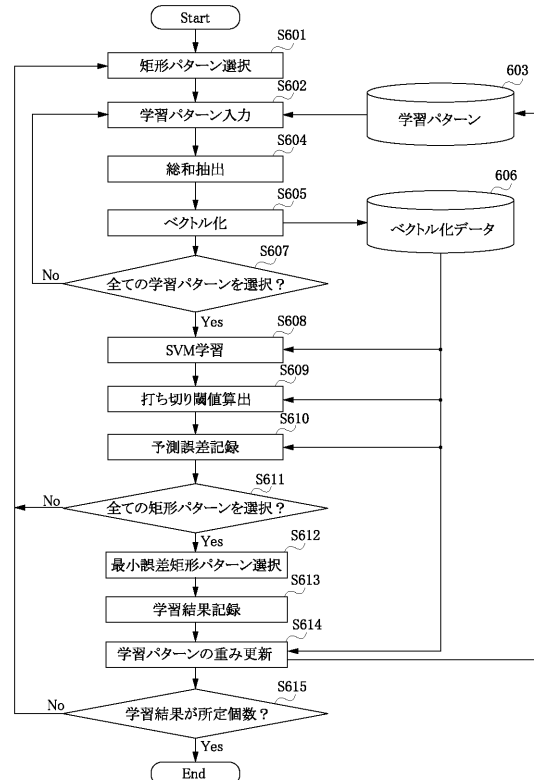
【 図 4 】



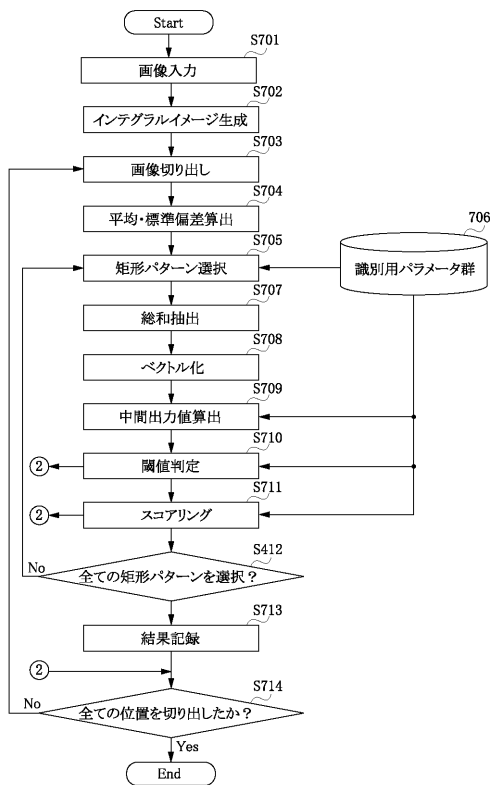
【 図 5 】



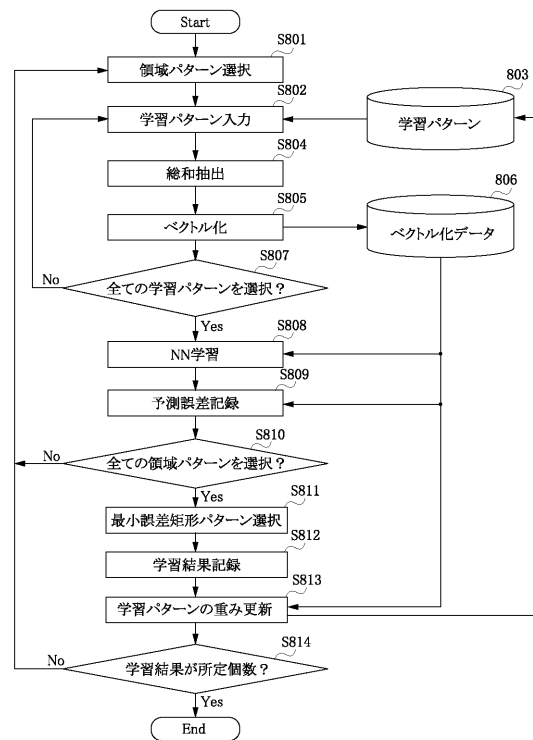
【 図 6 】



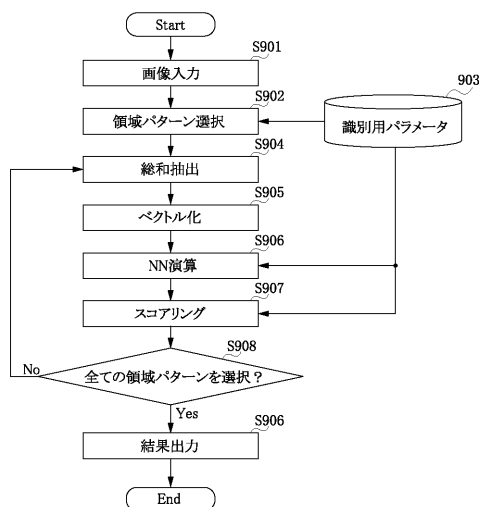
【図 7】



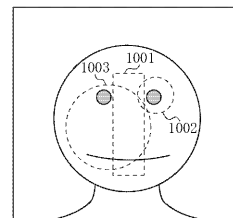
【図 8】



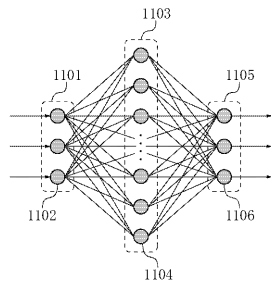
【図 9】



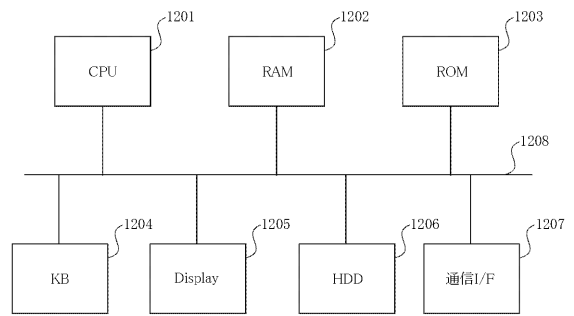
【図 10】



【図 1 1】



【図 1 2】



フロントページの続き

- (72)発明者 森 克彦
東京都大田区下丸子3丁目30番2号キヤノン株式会社内
- (72)発明者 鳥居 寛
東京都大田区下丸子3丁目30番2号キヤノン株式会社内
- (72)発明者 佐藤 博
東京都大田区下丸子3丁目30番2号キヤノン株式会社内
- Fターム(参考) 5L096 AA06 FA26 HA09 JA11 JA22 KA04 MA07