(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5368687号
(P5368687)
(45)発行日　平成25年12月18日(2013.12.18)　(24)登録日　平成25年9月20日(2013.9.20)

(51)Int.Cl.　　F I
G06N 3/063 (2006.01)　　G06N 3/063
G06N 3/00 (2006.01)　　G06N 3/00 560C
G06T 7/00 (2006.01)　　G06T 7/00 350C

請求項の数 13　(全 37 頁)

(21)出願番号　特願2007-250063 (P2007-250063)
(22)出願日　平成19年9月26日 (2007.9.26)
(65)公開番号　特開2009-80693 (P2009-80693A)
(43)公開日　平成21年4月16日 (2009.4.16)
審査請求日　平成22年9月27日 (2010.9.27)

(73)特許権者　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　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治
(74)代理人　100134175
弁理士　永川　行光

最終頁に続く

(54)【発明の名称】演算処理装置および方法

(57)【特許請求の範囲】
【請求項１】
入力データに対して演算を行い演算結果データを生成する、複数の論理的な処理ノードを接続したネットワークでネットワーク演算を実行する演算処理装置であって、
前記ネットワークを構成する複数の処理ノードの各々が演算結果データを保持するためのバッファ用の記憶領域をメモリに割り当てるために、１つの処理ノードに対して演算結果データの一部を保持するバッファを割り当てるバンドバッファ方式と、１つの処理ノードに対して前記入力データに対する演算結果データの全てを保持するバッファを割り当てるページバッファ方式とを含む複数種類のバッファ割り当て方法のそれぞれについて、前記ネットワーク演算に必要となる前記記憶領域のメモリ量を、当該ネットワークの構成に基づいて算出する算出手段と、
前記バッファとして割当て可能なメモリ量を取得する取得手段と、
前記複数種類のバッファ割り当て方法のうち、前記算出手段で算出されたメモリ量が前記取得手段で取得されたメモリ量以下になるバッファ割り当て方法を選択する選択手段と、
前記選択手段で選択されたバッファ割り当て方法に応じた実行順で、前記ネットワーク演算における各処理ノードによる演算を実行させる実行手段とを備えることを特徴とする演算処理装置。
【請求項２】
前記ネットワークは、複数の論理的な処理ノードを階層的に接続した階層型ネットワー

クであることを特徴とする請求項１に記載の演算処理装置。
【請求項３】
前記選択手段は、前記算出手段で算出されたメモリ量が前記取得手段で取得されたメモリ量以下になるバッファ割り当て方法が複数存在する場合には、予め定めた優先順位に基づいてそれらのバッファ割り当て方法の中から１つを選択することを特徴とする請求項１に記載の演算処理装置。
【請求項４】
前記算出手段は、前記ネットワーク演算を実行する全ての処理ノードの各々のバッファのサイズを当該処理ノードの後段に接続される処理ノードが必要とするデータ量に設定し、それらバッファのサイズを合計することにより、前記バンドバッファ方式を用いた場合の前記ネットワーク演算に必要なメモリ量を算出することを特徴とする請求項１に記載の演算処理装置。
【請求項５】
前記算出手段は、前記ネットワークの構成に存在する連続する２つの階層のそれぞれの組について、２つの階層に属する全ての処理ノードが前記入力データに対する演算結果データの全てを保持した場合に必要となるバッファの合計サイズを計算し、それら合計サイズの内の最大のサイズを、前記ページバッファ方式を用いた場合の前記ネットワーク演算に必要なメモリ量とすることを特徴とする請求項１に記載の演算処理装置。
【請求項６】
前記実行手段は、前記バンドバッファ方式が選択された場合には、前記複数の処理ノードの各々を予め定められた処理単位で実行させ、
前記複数の処理ノードの各々に割り当てられたバッファを、前記処理単位の演算結果の量に対応したメモリ領域を単位として循環させながら演算結果を書き込むリングバッファとして制御することを特徴とする請求項１乃至５のいずれか１項に記載の演算処理装置。
【請求項７】
前記実行手段は、前記ページバッファ方式が選択された場合には、１つの階層に属する全ての処理ノードの出力が生成された後に次階層に属する処理ノードの演算処理を開始するように制御し、
Ｎ番目の階層に属する全ての処理ノードの出力が生成された後、Ｎ－１番目の階層に属する全ての処理ノードが使用していたバッファ領域を開放し、Ｎ＋１番目以降の階層に属する処理ノードのバッファ領域に割り当てることを特徴とする請求項１乃至６のいずれか１項に記載の演算処理装置。
【請求項８】
入力データに対して演算を行い演算結果データを生成する、複数の論理的な処理ノードを接続したネットワークでネットワーク演算を実行する演算処理装置であって、
前記ネットワークを構成する複数の処理ノードの各々が演算結果データを保持するためのバッファ用の記憶領域をメモリに割り当てる複数種類のバッファ割り当て方法のそれぞれについて、前記ネットワーク演算に必要となる前記記憶領域のメモリ量を、当該ネットワークの構成に基づいて算出する算出手段と、
前記バッファとして割当て可能なメモリ量を取得する取得手段と、
前記複数種類のバッファ割り当て方法のうち、前記算出手段で算出されたメモリ量が前記取得手段で取得されたメモリ量以下になるバッファ割り当て方法を選択する選択手段と、
前記選択手段で選択されたバッファ割り当て方法に応じた実行順で、前記ネットワーク演算における各処理ノードによる演算を実行させる実行手段と、
前記算出手段で算出されたメモリ量が前記取得手段で取得されたメモリ量以下になるバッファ割り当て方法が存在しない場合に、前記入力データを分割する分割手段とを備え、
前記選択手段は、分割された入力データの各々について前記ネットワーク演算に用いるべきバッファ割り当て方法を選択することを特徴とする演算処理装置。
【請求項９】

前記処理ノードが行う演算はコンボリューション演算であり、前記ネットワークは、コンボリューショナル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８のいずれか１項に記載の演算処理装置。

【請求項１０】

入力データに対して演算を行い演算結果データを生成する、複数の論理的な処理ノードを接続したネットワークでネットワーク演算を実行する演算処理方法であって、

前記ネットワークを構成する複数の処理ノードの各々が演算結果データを保持するためのバッファ用の記憶領域をメモリに割り当てるために、１つの処理ノードに対して演算結果データの一部を保持するバッファを割り当てるバンドバッファ方式と、１つの処理ノードに対して前記入力データに対する演算結果データの全てを保持するバッファを割り当てるページバッファ方式とを含む複数種類のバッファ割り当て方法のそれぞれについて、前記ネットワーク演算に必要となる前記記憶領域のメモリ量を、当該ネットワークの構成に基づいて算出する算出工程と、

前記バッファとして割当て可能なメモリ量を取得する取得工程と、

前記複数種類のバッファ割り当て方法のうち、前記算出工程で算出されたメモリ量が前記取得工程で取得されたメモリ量以下になるバッファ割り当て方法のうちの１つを選択する選択工程と、

前記選択工程で選択されたバッファ割り当て方法に応じた実行順で、前記ネットワーク演算における各処理ノードによる演算を実行させる実行工程とを備えることを特徴とする演算処理方法。

【請求項１１】

入力データに対して演算を行い演算結果データを生成する、複数の論理的な処理ノードを接続したネットワークでネットワーク演算を実行する演算処理方法であって、

前記ネットワークを構成する複数の処理ノードの各々が演算結果データを保持するためのバッファ用の記憶領域をメモリに割り当てる複数種類のバッファ割り当て方法のそれぞれについて、前記ネットワーク演算に必要となる前記記憶領域のメモリ量を、当該ネットワークの構成に基づいて算出する算出工程と、

前記バッファとして割当て可能なメモリ量を取得する取得工程と、

前記複数種類のバッファ割り当て方法のうち、前記算出工程で算出されたメモリ量が前記取得工程で取得されたメモリ量以下になるバッファ割り当て方法を選択する選択工程と、

前記選択工程で選択されたバッファ割り当て方法に応じた実行順で、前記ネットワーク演算における各処理ノードによる演算を実行させる実行工程と、

前記算出工程で算出されたメモリ量が前記取得工程で取得されたメモリ量以下になるバッファ割り当て方法が存在しない場合に、前記入力データを分割する分割工程とを備え、

前記選択工程では、分割された入力データの各々について前記ネットワーク演算に用いるべきバッファ割り当て方法を選択することを特徴とする演算処理方法。

【請求項１２】

請求項１０または１１に記載の演算処理方法をコンピュータに実行させるためのコンピュータプログラム。

【請求項１３】

請求項１０または１１に記載の演算処理方法をコンピュータに実行させるためのコンピュータプログラムを格納したコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、例えばパターン識別装置、パターン識別システムや階層的なフィルタ演算処理装置等に適用可能なネットワーク演算を実行する演算処理装置および方法に関するものである。

【背景技術】

【０００２】
パターン識別システムや予測システム、制御システム等への応用として、ニューラルネットワークを利用した信号処理装置が広く利用されている。一般に、ニューラルネットワークはマイクロプロセッサ上で動作するソフトウェアとして実現される事が多く、パーソナルコンピュータやワークステーション等のアプリケーションソフトウェアとして提供されている。

【０００３】
図２は、一般的な階層結合型ニューラルネットワークを利用する画像処理装置の概念的な構成例を示すブロック図である。２１は検出対象のデータであり、例えば、ラスタスキャンされた画像データを示す。２２は画像データ２１中から所定の物体を検出する演算ユニットであり、図示の例では３階層のニューラルネットワークで構成されている。２３は演算結果に相当する出力データ面である。演算ユニット２２は、画像データ２１中の所定の画像領域２４を走査参照しながら処理を行う事で、画像中に存在する検出対象を検出する。出力データ面２３は、検出対象の画像データ２１と同じサイズのデータ面である。出力データ面２３には、画像データ２１の全ての領域を走査しながら処理した演算ユニット２２の検出出力が走査順に格納される。演算ユニット２２は、対象物が検出された位置で大きな値を出力する事から、当該出力データ面２３を走査する事で対象物の画像面内の位置を把握する事ができる。

【０００４】
演算ユニット２２において、２５，２６、２７は夫々ニューラルネットワークの階層を示し、各階層に所定の数のニューロン２８が存在する。第１階層２５は参照画像の画素数と同じ数のニューロン（ノード）２８を有する。各ニューロンは所定の重み係数でフィードフォワード結合する。

【０００５】
図３は１つのニューロン２８の構成例を示すブロック図である。in_1～in_nは入力値であり、第２階層以降では前階層のニューロンの出力値である。累積加算器３２は、当該入力値と学習によって得られた係数w_1～w_nを乗じた結果を累積加算する。非線形変換処理部３３は、累積加算器３２の累積加算結果をロジスティック関数や双曲正接関数（tanh関数）等により非線形変換し、その変換結果を検出結果outとして出力する。なお、階層型ニューラルネットワークにおいて、夫々のニューロンに必要な重み係数w_1～w_nは一般的に知られているバックプロパゲーション等の学習アルゴリズムを使用して、検出する対象物毎に決定されているものである。

【０００６】
このような階層結合型ニューラルネットワークを組み込み機器等へ高性能かつ安価に実装する事を目的として、アナログハードウェアやディジタルハードウェアで実現する手法が提案されている。

【０００７】
特許文献１では、単階層のアナログニューラルネットワークハードウェアを時分割多重利用する事で多階層化を実現する階層構造ニューラルネットのアーキテクチャが開示されている。又、特許文献２ではディジタルハードウェアにより実現する方法が開示されている。

【０００８】
一方、ニューラルネットワークの中でも、Convolutional Neural Networksと呼ばれるニューラルネットワークを用いた演算手法は識別対象の変動に対して頑健なパターン認識を可能にする手法として知られている。以下、Convolutional Neural NetworksはＣＮＮと略記する。例えば、特許文献３及び特許文献４では、ＣＮＮ演算を画像中の対象物識別や検出に適用した例が提案されている。

【０００９】
図４は簡単なＣＮＮの例を示す論理的なネットワーク構成図である。ここでは、第１階層４０６の特徴数が３、第２階層４１０の特徴数が２、第３階層４１１の特徴数が１の３

層ＣＮＮの例が示されている。４０１は画像データであり、ラスタスキャンされた画像データに相当する。４０３ａ～４０３ｃは第１階層４０６の特徴面を示す。特徴面とは、所定の特徴抽出フィルタ（コンボリューション演算の累積和及び非線形処理）で前階層のデータを走査しながら演算した結果を示す画像データ面である。特徴面はラスタスキャンされた画像データに対する検出結果であるため面で表す。特徴面４０３ａ～４０３ｃは４０１から夫々対応する特徴抽出フィルタにより生成される。例えば、特徴面４０３ａ～４０３ｃは、夫々模式的にコンボリューションカーネル４０４ａ～４０４ｃに対応する２次元のコンボリューションフィルタ演算とその演算結果の非線形変換により生成される。なお、４０２はコンボリューション演算に必要な参照画像領域を示す。

【００１０】

例えば、カーネルサイズ（水平方向の長さと垂直方向の高さ）が１１×１１のコンボリューションフィルタ演算は以下に示すような積和演算により処理する。

【００１１】

【数１】

$$output(x,y)=\sum_{row=-RowSize/2}^{rowSize/2}\sum_{column=-columnSize/2}^{columnSize/2}input(x+column,y+row)\times weight(column,row)$$

…（１）

ここで、

input(x,y):座標(ｘ、ｙ)での参照画素値、

output(x,y):座標(ｘ、ｙ)での演算結果、

weight(column,row):座標(ｘ+column、ｙ+row)での重み係数、

columnSize=１１,rowSize=１１:フィルタカーネルサイズ（フィルタタップ数）である。

【００１２】

４０４ａ～４０４ｃは夫々異なる係数のコンボリューションフィルタカーネルである。また、特徴面によってコンボリューションカーネル４０４ａ～４０４ｃのサイズも異なる。

【００１３】

ＣＮＮ演算では、複数のフィルタカーネルを画素単位で走査しながら積和演算を繰り返し、最終的な積和結果を非線形変換する事で特徴面を生成する。特徴面４０３ａを算出する場合は、前階層との結合数が１であるため、１つのコンボリューションカーネル４０４ａが用いられる。一方、特徴面４０７ａ及び４０７ｂを計算する場合、前階層との結合数が３であるため、夫々４０９ａ～４０９ｃ及び４０９ｄ～４０９ｆに相当する３つのコンボリューションフィルタの演算結果を累積加算する。つまり、特徴面４０７ａは、コンボリューションカーネル４０９ａ～４０９ｃの出力を累積加算し、最後に非線形変換処理する事によって得られる。

【００１４】

ここで、４０９ａ～４０９ｆは何れも異なるフィルタ係数のコンボリューションカーネルである。また、コンボリューションカーネル４０９ａ～４０９ｃと４０９ｄ～４０９ｆは、図４に示されるように、それぞれ異なるカーネルサイズを有する。コンボリューションフィルタの累積加算及び非線形変換処理の基本的構成は図３に示すニューロンの構成と同様である。即ち、コンボリューションカーネルの係数が重み係数w_1～w_nに相当する。特徴面４０７ａ、４０７ｂ、４０８の様に複数の前階層の特徴面と結合される場合、複数のコンボリューションカーネルの演算結果が累積加算器３２で蓄積される事になる。即ち、総結合数はコンボリューションカーネルサイズ×前階層の特徴数に相当する。

【００１５】

図５はＣＮＮ演算における図形検出処理の一例を説明する図である。５１ａ～５１ｃは第１階層４０６の特徴抽出対象を模式的に示す図であり、それぞれ水平方向のエッジ及び斜め方向のエッジを抽出する様に学習されたコンボリューションカーネルである。５２ａ、５２ｂは複数の第１階層における特徴抽出結果とその空間的な配置関係から、第２階層

４１０で抽出される図形である。５３は最終的に抽出される図形を示している。５３は複数の第２階層４１０における特徴抽出結果とその空間配置関係から、第３階層４１１で抽出される図形である。コンボリューションカーネルの各フィルタ係数は特徴毎にパーセプトロン学習やバックプロパゲーション学習等の一般的な手法を用いて予め学習により決定されているものとする。物体の検出や認識等においては、１０×１０以上の大きなサイズのフィルタカーネルを使用する事が多い。また、一般的に特徴毎にコンボリューションカーネルのサイズは異なる。

【００１６】

このように、ＣＮＮ演算では、特徴抽出毎に画像面単位で結果を保持しながら階層的に結合する事で、プリミティブな特徴とその空間的な配置関係に基づく頑健なパターン検出を実現する。

【特許文献１】特許第２６７９７３０号明細書
【特許文献２】特開平３－５５６５８号公報
【特許文献３】特開平１０－０２１４０６号公報
【特許文献４】特開２００２－３５８５００号公報

【発明の開示】

【発明が解決しようとする課題】

【００１７】

図２で説明した様に、一般的な階層型ニューラルネットワークを利用した画像中の物体検出装置において、演算処理に必要なメモリサイズは、入出力画像バッファを除くと、各ニューロン出力を保持するためのバッファメモリがあれば十分である。即ち、ニューロン数と等価な数の所定ビット数のメモリがあれば所望の演算処理を実行できる。

【００１８】

一方、ＣＮＮ演算の場合、前階層の複数の特徴抽出結果の空間的配置に基いて特徴抽出を行うため、各階層間で所定サイズのデータバッファが必要になる。例えば、図４に示すＣＮＮ演算構成例の場合、入出力画像バッファを除くと画像サイズ×５個の特徴面バッファメモリを用意している（特徴面４０３ａ～４０３ｃ、特徴面４０７ａ～４０７ｂ）。このため、一般的な階層型ニューラルネットに比べ処理に必要なメモリサイズが増大する。

【００１９】

特許文献３及び特許文献４に開示されている手法も、特徴抽出結果を画像面で保持する手法であり、処理に必要なメモリサイズが一般的な階層型ニューラルネットワークによる方式に比べて大きい。

【００２０】

このため、特に、ハードウェアにより実現する場合、ＬＳＩの内部にサイズの大きいＲＡＭ（RandomAccess Memory）を用意する必要があり、回路規模が増大する。ソフトウェアにより実現する場合であっても、組み込み機器に実装する場合、システムに必要なメモリ量が増大する事で同様にコストが上昇する。すなわち、演算に使用可能なメモリ量は、システムにかけることのできるコストによって定まる有限な値となる。

【００２１】

一方、メモリの増大を避ける手法として、入力するデータを領域分割して投入する方法が利用されている。しかしながら、参照領域が広い演算を階層的に処理する場合、分割投入するデータを広い範囲でオーバーラップさせる必要があるため、結果的に処理対象領域が増加してしまい、処理効率及び処理速度が低下する。

【００２２】

本発明はこのような問題点を解決するためになされたものであり、ＣＮＮ演算等のネットワーク構造で接続された複数の処理ノードによって行われる演算処理を、限られたメモリ量で効率良く実現する事を目的とする。

【課題を解決するための手段】

【００２３】

上記課題を解決するために、本発明の一態様による演算処理装置は以下の構成を備える

。すなわち、
入力データに対して演算を行い演算結果データを生成する、複数の論理的な処理ノードを接続したネットワークでネットワーク演算を実行する演算処理装置であって、
前記ネットワークを構成する複数の処理ノードの各々が演算結果データを保持するためのバッファ用の記憶領域をメモリに割り当てるために、１つの処理ノードに対して演算結果データの一部を保持するバッファを割り当てるバンドバッファ方式と、１つの処理ノードに対して前記入力データに対する演算結果データの全てを保持するバッファを割り当てるページバッファ方式とを含む複数種類のバッファ割り当て方法のそれぞれについて、前記ネットワーク演算に必要となる前記記憶領域のメモリ量を、当該ネットワークの構成に基づいて算出する算出手段と、
前記バッファとして割当て可能なメモリ量を取得する取得手段と、
前記複数種類のバッファ割り当て方法のうち、前記算出手段で算出されたメモリ量が前記取得手段で取得されたメモリ量以下になるバッファ割り当て方法を選択する選択手段と、
前記選択手段で選択されたバッファ割り当て方法に応じた実行順で、前記ネットワーク演算における各処理ノードによる演算を実行させる実行手段とを備える。
【００２４】
また上記課題を解決するための、本発明の他の態様による演算処理方法は、
入力データに対して演算を行い演算結果データを生成する、複数の論理的な処理ノードを接続したネットワークでネットワーク演算を実行する演算処理方法であって、
前記ネットワークを構成する複数の処理ノードの各々が演算結果データを保持するためのバッファ用の記憶領域をメモリに割り当てるために、１つの処理ノードに対して演算結果データの一部を保持するバッファを割り当てるバンドバッファ方式と、１つの処理ノードに対して前記入力データに対する演算結果データの全てを保持するバッファを割り当てるページバッファ方式とを含む複数種類のバッファ割り当て方法のそれぞれについて、前記ネットワーク演算に必要となる前記記憶領域のメモリ量を、当該ネットワークの構成に基づいて算出する算出工程と、
前記バッファとして割当て可能なメモリ量を取得する取得工程と、
前記複数種類のバッファ割り当て方法のうち、前記算出工程で算出されたメモリ量が前記取得工程で取得されたメモリ量以下になるバッファ割り当て方法のうちの１つを選択する選択工程と、
前記選択工程で選択されたバッファ割り当て方法に応じた実行順で、前記ネットワーク演算における各処理ノードによる演算を実行させる実行工程とを備える。
【発明の効果】
【００２５】
本発明によれば、ＣＮＮ演算等のネットワーク構造で接続された複数の処理ノードによって行われる演算処理を、限られたメモリ量で効率良く実現することができる。中間演算結果を介して演算部を接続するネットワーク型演算器による演算処理を、限られたメモリ量において最適に処理することが可能となる。すなわち、同じ構成のハードウェアで、より多様なネットワーク型演算処理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】
【００２６】
以下、本発明の好適な実施形態について、添付の図面を用いて説明する。
【００２７】
＜第１実施形態＞
図６は第１実施形態に関する階層的演算処理回路を具備したパターン検出装置の構成例である。当該装置は画像データ中の特定の物体（画像パターン）を検出する機能を有する画像処理装置である。図６において６１は画像入力部であり、光学系、ＣＣＤ（Charge-Coupled Devices）又はＣＭＯＳ（Complimentary Metal OxideSemiconductor）センサ等の光電変換デバイスを有する。また、画像入力部６１は、光電変換間デバイスを制御するド

ライバ回路／ＡＤコンバータ／各種画像補正を司る信号処理回路／フレームバッファ等を具備する。６２は前処理部であり、検出処理を効果的に行うための各種前処理を行う。具体的には、前処理部６２は、色変換処理／コントラスト補正処理等の画像データ変換をハードウェアで処理する。６３はＣＮＮ処理部であり、本実施形態による階層的演算処理回路を含み、特徴検出処理部として機能する。なお、ＣＮＮ処理部６３の詳細は図１を用いて後述する。６６はＤＭＡＣ（DirectMemory Access Controller）であり、画像バス６４上の各処理部間のデータ転送、及び、画像バス６４上のデバイスとＣＰＵバス６７上のＲＡＭ７０間のデータ転送を司る。６５はブリッジであり、画像バス６４とＣＰＵバス６７のブリッジ機能を提供する。６８はＣＰＵであり、本装置全体の動作を制御するものである。６９はＲＯＭ（ReadOnly Memory）であり、ＣＰＵ６８の動作を規定する命令や各種演算に必要なパラメータデータを格納する。例えば、ＣＮＮ処理部６３の動作に必要な重み係数、ネットワーク結合情報、シーケンス情報等もＲＯＭ６９に格納されている。７０はＣＰＵ６８の動作に必要なメモリ（ＲＡＭ：RandomAccess Memory）である。ＲＡＭ７０はＤＲＡＭ（Dynamic ＲＡＭ）等の比較的容量の大きいメモリで構成される。ＣＰＵ６８はブリッジ６５を介して画像バス６４上の各種処理部にアクセスする事が可能である。画像バス６４とＣＰＵバス６７を分離する事により、ハードウェアによる画像入力部６１、前処理部６２、ＣＮＮ処理部６３の各処理部の動作とＣＰＵ６８の動作を同時に並列実行させることができる。

【００２８】

図６の階層的演算処理回路を具備したパターン検出装置は、例えば図１４に示されるようなConvolutionalNeural Networks（以下ＣＮＮと略記する）のような階層的な演算を行うのに用いられる。図１４において処理ノードとは、コンボリューション演算の対象画像とコンボリューションカーネルからコンボリューション演算結果を得る処理を行うブロックを指す。なお、図１４では便宜上第０処理ノードを設けたが、通常第０処理ノードでは特になにも処理は行われず、入力画像が第１～第３処理ノードへ入力される。例えば、図１４の第４処理ノードでは、第１～３処理ノードの出力に対し、それぞれ係数の異なるコンボリューションカーネルを適用してコンボリューション演算を行う。そして、それぞれのコンボリューション演算の結果を加算し、その加算結果に非線形変換を行って第４処理ノードの演算結果を得ている。

【００２９】

ＣＮＮ処理部６３に図１４に示されるＣＮＮを適用する場合、演算処理部を処理ノード間で時分割に使用することで、各処理ノードで規定された演算を実行する。例えば、まず第１処理ノードで規定された演算を行い、その後第２処理ノードで規定された演算を行う、というようにＣＮＮの演算が実行されていく。つまり、ＣＮＮを構成する処理ノードは複数存在し、論理的なネットワークを構成するが、処理ノードで規定された演算を実行する演算処理部は物理的に１つしか存在しない。

【００３０】

図１はＣＮＮ処理部６３の詳細を説明する図である。図１において、１０１は演算部であり、所定のデータ群に対してコンボリューション演算と非線形処理を実行する。図１２に演算部１０１の一例を示す。図１２において、１２０１は乗算器であり、係数選択部１２０４がカーネル選択信号に従って重み係数記憶部１２０５より選択し、出力する重み係数と、カーネル選択信号と同期して入力される入力データとを乗じる。１２０２は累積加算器であり、乗算器１２０１の出力を所定の期間累積加算する。１２０３は非線形変換処理であり、ロジスティック関数やtanh関数を用いて累積加算結果を非線形変換する。非線形変換は、例えば、各入力値に対して所定の関数値を列挙する関数テーブルで実現される。１２０５は重み係数記憶部であり、検出対象と処理ノードに応じた複数の重み係数データが格納されている。重み係数記憶部１２０５は、例えばＲＡＭ等により構成される。１２０４は係数選択部であり、ネットワーク構成管理部１０８が指示するカーネル選択信号に従って、対応する重み係数を順次記憶部から読み出す。

【００３１】

図１に戻り、１０２はワークメモリ（以下、単にメモリという）であり、入力画像／中間層の特徴抽出結果／最終検出結果等を格納する。本実施形態では、コンボリューション演算を高速に実行するため、メモリ１０２として、高速にランダムアクセスが可能なＳＲＡＭ（Static ＲＡＭ）を使用している。

【００３２】

１０３はメモリアクセス制御部であり、メモリ１０２に対するアクセス、アドレスの生成、リード／ライト信号制御及びデータバスの方向制御等を司る。メモリアクセス制御部１０３はリングバッファ設定部１０４-1～１０４-nの出力に従ってメモリ１０２にアクセスする。

【００３３】

１０４-1～１０４-nは複数のリングバッファ設定部であり、それぞれ各処理ノードがメモリ１０２をリングバッファとして利用するために用いられる。以下、リングバッファ設定部１０４-1～１０４-nの任意の１つを指す場合は、リングバッファ設定部１０４と記載する。リングバッファ設定部１０４は、ＣＮＮ演算の論理的な各処理ノード毎に一つずつ用意される。各リングバッファ設定部１０４は、リングバッファのサイズを指定するリングサイズ設定部１０６、リングバッファの動作状況を保持するリングカウンタ１０５及びメモリ１０２上の物理アドレスを決定するためのオフセットアドレス設定部１０７を具備する。リングバッファ設定部１０４の出力はセレクタ１１２１、１１２２で選択されて、メモリアクセス制御部１０３へ提供される。この構成により、メモリ１０２には、ネットワークを構成する複数の処理ノードの各々に対応して、演算結果データを保持するための中間バッファ用の記憶領域が割り当てられることになる。その詳細は、後述する。

【００３４】

１０８はネットワーク構成管理部であり、１つの演算部１０１を利用して論理的な階層ネットワーク処理を実現するために、各動作を制御する。論理的な階層結合関係を指定する構成情報は、ネットワーク構成情報設定部１１０内にテーブルデータ（以下、構成情報テーブルという）として保持される。構成情報テーブルはレジスタやＲＡＭで構成される。ネットワーク構成管理部１０８は、構成情報テーブルに従ってメモリアクセス制御部１０３や演算部１０１の動作を順次制御することにより、後述する所定の単位演算をベースとした階層ネットワークの演算処理を実現する。

【００３５】

シーケンス制御部１０９は、シーケンス情報設定部１１１に記されたシーケンス情報に従って、各処理ノードによる単位演算の実行順を制御する。本実施形態では、１ライン単位の出力を得る演算処理を所定の演算処理単位（単位演算）としている。すなわちライン単位で論理的な処理ノードを切り替えながら時分割で処理を実行することで階層型ネットワーク演算が遂行される。シーケンス情報設定部１１１はシーケンス情報を保持するＲＡＭ等により構成される。

【００３６】

１１３はＣＰＵバスアクセス制御部であり、ＣＰＵ６８がＣＮＮ処理部６３内の各種レジスタやメモリにアクセスするためのバスインターフェースである。例えば、

・リングサイズ設定部１０６のリングバッファサイズ、
・ネットワーク構成情報設定部１１０の構成情報テーブル、
・シーケンス情報設定部１１１のシーケンス情報、
・演算部１０１の重み係数記憶部１２０５内の重み係数データ、

等の各種設定データは、当該インターフェースを介してＣＰＵ６８から書き込むことができる。

【００３７】

ここで、図２２を用いて本実施形態における所定の単位演算について説明する。先に述べたとおり、本実施形態での所定の単位演算とは、演算部１０１を用いて行われるライン単位のコンボリューション演算処理である。ただし、図２２では、簡単のため、一つの処理ノードの演算出力画像（または、ネットワークへの入力画像）を演算対象画像としてコ

ンボリューション演算を行う場合が示されており、非線形変換も省略されている。

【００３８】

図２２の（ａ）において、２２０１は演算対象画像（参照画像）を表している。演算対象画像２２０１において、模式的に示す最小一升が、ラスタスキャン順で示された入力画像又は前階層の処理ノードでの演算結果画像である演算対象画像の画素（input(x, y)、ｘ：水平方向位置、ｙ：垂直方向位置）を示す。

２２０２は演算結果画像を表し、模式的に示す最小一升が、ラスタスキャン順の演算結果画素（output(x,y)、ｘ：水平方向位置、ｙ：垂直方向位置）を示すものとする。

【００３９】

演算対象画像２２０１内の太線で囲まれた領域２２０３は、output(6, 7)位置のコンボリューション演算を処理する場合の参照画像の領域を示す。領域２２０３では、コンボリューションカーネルのサイズが水平方向「１１」、垂直方向「１３」の場合が示されている。

【００４０】

演算結果画像２２０２の太線で囲まれた領域２２０４は、演算対象画像２２０１に対して単位演算（水平方向１行分の演算）を行った場合の結果領域を示す。ここで、領域２２０４内の格子状の網掛け領域２２０６は、コンボリューションカーネルのサイズに依存して発生する周辺領域（演算が行われない領域）の画素である。つまり、output(5, 7)の位置の演算を行うべき参照画像領域は、領域２２０３を左に１画素分ずらしたものとなる。しかしながら、そのような領域は演算対象画像２２０１（参照領域）からはみ出してしまうため、一部の参照画素が存在しないことになる。なお、階層的処理においてこの周辺領域（無効領域）をどう扱うか（削除するか、デフォルト値を埋め込むか等）は、本発明において本質的でないので、ここでは例えば、デフォルト値を埋め込むとする。尚、２２０４より上のラインについても同様に無効領域となる。

【００４１】

図２２から明らかなように、１ラインの単位演算を行うには、演算対象画像２２０１の必要領域として、少なくとも領域２２０５が必要となる。領域２２０５は、図２２において網掛け領域として示されており、水平方向サイズは演算対象画像２２０１と同じサイズ、垂直方向サイズはコンボリューションカーネルの垂直方向サイズが必要となる。説明の都合上、この領域を単位演算対象画像領域２２０５と呼ぶ。領域２２０４で示されるような単位演算を、単位演算対象画像領域２２０５をずらしながら行うことで、演算対象画像２２０１の全領域にわたってコンボリューション演算を行うことができる。例えば、図２２の（ｂ）には、１画素下にずらした単位演算対象画像領域に対して単位演算を行った場合を示している。この時、ある単位演算を実行できるか否かは、その単位演算の単位演算対象画像領域２２０５’の画素データが、前階層の処理ノードによって演算され、その結果が出力されているか否かに依存する。もちろん、複数の参照画像を入力として演算に用いる処理ノードの場合は、全ての参照画像についての単位演算対象画像領域の画素データが出力されている必要がある。

【００４２】

図７は、図４で説明したＣＮＮネットワークに対し、本実施形態を適用した場合の動作の一例を説明する図である。

【００４３】

図７において、７０１は入力層であり、所定サイズの検出対象画像データが入力される。７０３ａ～７０３ｃは第１階層７０６の演算出力を格納するメモリ領域を示す。すなわち、入力層７０１の入力面に対するコンボリューション演算７０４ａ～７０４ｃ及び非線形変換の結果が、メモリ領域７０３ａ～７０３ｃにそれぞれ格納される。第１階層の演算結果である特徴面は論理的には入力層７０１と同じサイズのデータ面となる。しかしながら、ここでは所定高さのリングバッファとして機能するメモリ領域７０３ａ、７０３ｂに特徴面が格納される。このリングバッファは、幅が入力画像と同じであり、ライン単位で循環するバンドバッファである。

【００４４】
図１３は、本実施形態によるリングバッファの動作を模式的に説明する図である。ここでは説明のためリングバッファの高さ（循環数）を６とする。また、ここでは、入力画像１３００の８ライン分の画像データが、Ｌ１～Ｌ８としてラスタスキャン順に入力された場合に、６ライン分のリングバッファにどのように保持され、参照されるかを説明する。
【００４５】
ここでリングバッファに付随するリングカウンタは０～５の値を循環する。また、リングカウンタの初期値は５であり、１ライン分のデータが投入されるときに１インクリメントされるものとする。ただし、リングバッファの循環数と同じ値になると、リングカウンタのカウンタ値は０に戻る。例えば、本リングバッファでは循環数は６であるので、カウンタ値は５の次は０に戻ることになる。
【００４６】
１３０１はリングバッファに入力画像１３００の先頭から６ライン分のデータ（Ｌ１～Ｌ６）がフルに充填された状態を表し、リングカウンタの値は「５」となっている。次のラインを格納するとき、リングカウンタはインクリメントされ０に戻り、リングバッファの先頭行にＬ７が充填される。すなわちリングカウンタの値は、最新のラインを格納したリングバッファ中の行を示す（０基準）。この状態を、図１３の１３０２に示す。
【００４７】
１３０２の状態では、リングバッファからＬ２～Ｌ７のバンドを参照する事が可能となり、その開始行はリングカウンタの値＋１の行である。更に次のラインＬ８を格納する場合は、１３０３に示すように２行目位置にＬ８が充填され、リングカウンタの値は１となる。この場合、Ｌ３～Ｌ８を参照する事が可能となり、先頭行はやはりリングカウンタの値＋１の行となっていることが分かる。
【００４８】
尚、リングバッファの循環数を入力画像データのライン数に一致させると、そのバッファにおいては、１ページ分の処理中において前のラインが上書きされなくなる。すなわち、リングバッファは、バンドバッファとしてだけでなく、ページバッファとしても機能させることができる。
【００４９】
ここで、各処理ノードのリングバッファに最低限必要な高さ（循環数）は、次階層の処理ノードが単位演算処理を行う際の、単位演算対象画像領域（図２２の２２０５）の高さに一致する。すなわち、各リングバッファの高さ（循環数）は、その処理ノードの後段に接続される全処理ノードのコンボリューションカーネルのサイズに基づいて決定することができる。
【００５０】
例えば図７において、メモリ領域７０３ａの場合、コンボリューションカーネル７０９ａと７０９ｄの単位演算対象画像領域の高さのうち大きい方の値をメモリ領域７０３ａのリングバッファの最低限必要な高さとする。このように定めると、メモリ領域７０３ａによって形成されたリングバッファに納められた画素データを用いて、コンボリューションカーネル７０９ａと７０９ｄのどちらのコンボリューション演算も可能となる。同様にメモリ領域７０３ｂ、７０３ｃに最低限必要な高さは、夫々コンボリューションカーネル７０９ｂ／７０９ｅ、コンボリューションカーネル７０９ｃ／７０９ｆのカーネルサイズから決定できる。同様に、メモリ領域７０７ａ、７０７ｂは、夫々コンボリューションカーネル７１２ａ、７１２ｂの単位演算対象画像領域の高さから決定できる。
【００５１】
なお、ここで決定しているのは、あくまで次階層が演算する上で最低限必要な高さであるので、他の必要性があればさらに大きい高さのリングバッファを用いてももちろんよい。例えば、図７では、メモリ領域７０３ａ、７０３ｂ、７０７ａ、７０７ｂはカーネルサイズから規定される次階層演算のために最低限必要な高さをリングバッファ高さとしている。しかし、メモリ領域７０３ｃは、次階層演算に使われるのみならず、ＣＰＵ６８が判

定処理に使用する特徴検出データでもあるため、入力画像データと同じサイズのページバッファを割り当てている。すなわちＣＰＵ６８は、最終階層の特徴データ７１３だけでなく、メモリ領域７０３ｃに格納された第１階層の特徴データも参照して検出対象画像の存在を判定することができる。

【００５２】

この様に、本実施形態のＣＮＮ処理部を用いると、中間層の特徴面をネットワークの結合状態及び目的に応じて最適なサイズのバッファ（メモリ領域７０３ａ、７０３ｂ、７０３ｃ、７０７ａ、７０７ｂ）にアサインする事が可能となる。

【００５３】

ところで、図７に示した本実施形態のＣＮＮ演算処理では、中間処理ノードの演算結果を保持するメモリ領域７０３ａ、７０３ｂ、７０７ａ、７０７ｂをバンドバッファとしている。このように、中間処理ノードの出力バッファ（中間バッファ）をバンドバッファとすることにより、図４で説明した従来のＣＮＮ演算処理よりも、メモリ使用容量を減らしている。しかしながら、ネットワーク構成と演算シーケンスによっては、必ずしもバンドバッファによる中間バッファがトータルのメモリ使用量を最小にするとは限らない。以下、そのようなケースについて説明する。

【００５４】

図２３は、Ｎ個の階層からなるＣＮＮに対し、中間バッファとしてバンドバッファを割り当てた場合の例を示す図である。図２３において、各円は論理的な処理ノードを表し、各円の右に付属する四角がそれぞれの処理ノードに割り当てられたバッファを示している。ここでは入力画像である第０階層と出力層である第Ｎ階層の処理ノードには、１ページ分のフレームバッファ（ページバッファ）を割当て、他の中間層のノードにはバンドバッファを割り当てている。なお、本明細書では、中間層の出力結果保持バッファを特に中間バッファと呼んでいる。また、一部の円の左側に付属する四角は、その処理ノードのコンボリューションカーネルを例示するものであるが、図が複雑になるので一部のみしか記述していない。

【００５５】

第０階層のノードは入力層であり、上述した通り便宜上処理ノードを割り当てているが、実際に演算処理を行うわけではなく、単に入力画像データが入力バッファメモリに格納されている状態を示す。

【００５６】

また、最終層（第Ｎ階層）に割り当てられているバッファは出力バッファであって、この例では２つの処理ノードに対し入力画像サイズと同等のサイズのバッファがそれぞれ割り当てられている。もちろんこれら出力バッファの大きさは、このＣＮＮによる演算結果を使用する、例えばＣＰＵ６８による後処理部の都合によって定めればよいもので、コンボリューション演算の都合にはよらない。例えば、最終出力結果からある範囲の重心を取ってその座標に検出対象が存在するとする後処理を行うならば、出力バッファの大きさは重心を取る範囲の演算結果のみを保持できる大きさで十分である。

【００５７】

図２３のＣＮＮでは、中間バッファ（第１階層～第Ｎ－１階層の出力バッファ）を、図７で説明したのと同様のバンドバッファとして割り当てている。それぞれのバンドバッファの横幅は入力画像幅と同じであり、縦高さは先に説明した通り、接続される次階層の処理ノード（隣接上層処理ノード）のコンボリューションカーネルのサイズによって定まる最低限必要な高さとしている。尚、ここでは説明を簡単にするため、中間層の結果は後処理には用いないものとする。

【００５８】

このように中間バッファを全てバンドバッファとして割り当てるとき、ＣＮＮ全体の処理シーケンスは、ライン単位の演算処理を、処理ノードを切り替えて順に行っていくことになる。例えば第１階層の第１処理ノードのバンドバッファの高さが５だったとすると、５ライン分の演算処理が終わりその結果が格納されると、この処理ノードの演算結果を参

照する第２階層の処理ノードが、１ライン分の単位演算処理を行うことができる。逆に第２階層の処理ノードの処理が終わらない限り、バンドバッファ内の５ラインの内の１ラインも破棄することはできない。そのため、新たな演算結果の格納場所がないため、第１階層の第１処理ノードは次の単位演算を行うことができない。こうして、第１階層の第１処理ノードは、その演算結果を必要とする全ての第２階層の処理ノードが１ライン分の単位演算処理を完了したら、第１階層の第１処理ノードは次の１ラインの単位演算処理を行うことができるということになる。

【００５９】

従って、ＣＮＮ全体で処理が１ライン単位ずつ進んでいくことになるので、全ての中間バッファは、基本的に同時に存在している必要がある。ここで、

全階層数（入力層除く）：　Ｎ

階層番号変数（中間層）：　ｌ＝１，２，…，Ｎ－１

各階層特徴数：　$Ｆ_{l}$

特徴番号（注目階層の注目特徴）：　$ｆ＝ｆ_{l}＝１，…，Ｆ_{l}$

前階層の特徴番号：　$ｆ’＝ｆ_{l-1}$

入力画像サイズ（水平方向，垂直方向）：$Ｉ_{X}$，$Ｉ_{Y}$

カーネルサイズ（水平方向，垂直方向）：$W_{X}(l,f,f’)$、$W_{Y}(l,f,f’)$

バンドバッファ高さ：$B_{Y}(l,f)$

バンド中間バッファ必要容量：$Ｓ_{B}$

とすると、全ての中間バッファを最低限必要なサイズのバンドバッファとして割り当てるときに必要となるトータルのサイズ$Ｓ_{B}$は、

【数２】

$$B_Y(l,f)=\max\left(W_Y(l+1,f_{l+1},f_l)\middle|1\le l\le N-1,1\le f_{l+1}\le F_{l+1}\right)$$

$$S_B=I_x\times\sum_{l=1}^{N-1}\sum_{f=1}^{F_l}B_Y(l,f)$$

…（２）

として求めることができる。

【００６０】

上記計算では、

・ネットワーク演算を構成する全ての処理ノードの各々に必要な中間バッファのサイズを当該処理ノードの後段に接続される処理ノードが必要とするデータ量に設定し、

・それら中間バッファのサイズを合計することにより、ネットワーク演算に必要なメモリ量を算出している。

【００６１】

一方、図２４は、図２３で示したＣＮＮと同じネットワーク構成のＣＮＮに対し、中間バッファとしてページバッファを用いた場合の例を示している。入力層および出力層のバッファは図２３と同様、ネットワーク構成や演算シーケンスによらず定まるもので、ここではやはりページバッファとしてある。

【００６２】

図２４では、第１階層から第Ｎ－１階層までの夫々の処理ノードに入力画像サイズと同等のサイズのページバッファが割り当てられている。しかしながら、これらのページバッファは、全て同時に存在している必要は無い。

【００６３】

まず第１階層の各ノードに注目すると、それぞれのノードにおける演算処理は、第０階層すなわち入力画像のみを参照して行われる。本実施形態において入力画像データは全てメモリ上に格納されているので、各ノードは必要なときに必要な範囲のデータを参照可能である。また、各ノードの出力格納バッファとして、１ページ分のバッファが用意されているので、バンドバッファの場合と異なりラインの上書きを気にすることなくいつでも演算結果を格納可能である。

【００６４】

次に第２階層の各処理ノードに注目すると、これらのノードは第１階層の処理結果を参照する。従って、第１階層の各処理ノードが、１ページ分の処理を終えた後であれば、必要な参照データは入力画像と同様ページで保持された状態となるので、所望のときに所望の範囲を参照することができる。以下の層も同様で、基本的に前段の階層の処理が完了しており参照データがページ分揃っていれば処理可能である。

【００６５】

逆に、各階層の処理ノードは演算処理を行うに当たって、前々階層の処理結果は必要としない。つまり、ある処理ノードの演算処理を開始するに当たって、前階層の全処理ノードの演算が１ページ分完了しているならば、更にその前の階層の処理ノードの中間バッファは開放してしまって構わないことになる。

【００６６】

従って、各処理ノードでライン単位の演算処理を順次行うに当たって、まずは第１階層に属する処理ノードでのみ順次単位演算を行う。そして、第１階層の全ての処理ノードで１ページ分の演算処理が完了した後に、第２階層の処理ノードでの単位演算処理を開始する。これを順に次の階層にたいしても繰り返してゆく、というように、単位演算処理シーケンスを組むと、中間バッファは１時期に連続する２階層のみで存在していれば、最終層までの演算が可能となる。より一般化して言うと、

・１つの階層に属する全ての処理ノードの出力が生成された後に次階層に属する処理ノードの演算処理を開始するように制御し、

・Ｎ番目の階層に属する全ての処理ノードの出力が生成された後、Ｎ－１番目の階層に属する全ての処理ノードが使用していた中間バッファ領域を開放してＮ＋１番目以降の階層に属する処理ノードの中間バッファ領域に割り当て可能にする。

こうして、１階層の演算処理が完了する毎に、その前の階層のページバッファを、次階層の中間バッファとして再利用するようにすることにより、トータルの中間バッファ必要サイズを減らすことが可能となる。以下、この方式をページバッファ方式という。

【００６７】

ここで、ページ中間バッファ必要容量：S_P

とすると、

【数３】

$$S_P = (I_X \times I_Y) \times \max\left(F_l, F_{l+1} \middle| 1 \le l \le N-1\right) \quad \cdots (3)$$

として、ページで中間バッファを構成した際（ページバッファ方式を採用した際）のトータルの必要サイズを求めることができる。この計算は、ネットワーク構成において連続する２つの階層の組に属する全ての処理ノードが生成する演算結果データのサイズの合計を全ての組について計算し、その内の最大となるサイズを必要なメモリ量とするものである。

【００６８】

（２）、（３）式から分かる通り、バンド中間バッファ必要容量S_Bとページ中間バッファ必要容量S_Pのどちらが小さくなるかは、ＣＮＮネットワークの各論理的処理ノードの接続構造と、各ノードのコンボリューションカーネルサイズに依存する。一般的に、ネットワークの階層数が少なくカーネルサイズが小さければ、バンドバッファ方式の割当てが有利となるが、階層数が多くかつ各階層に属する処理ノード数が比較的少なければページバッファ方式の割当てが有利となる。

【００６９】

通常、システムとして、中間バッファに用いることのできるメモリの上限サイズは一定である。この条件サイズをＭとする。特に本実施形態のような専用のパターン検出装置においては、中間バッファに割り当てるメモリサイズは、小さければ小さいほど良いという訳ではなく、割り当て可能なサイズＭ以下であれば全く問題ない。

【００７０】

また、上述のようにバンドバッファ方式の場合と、ページバッファ方式の場合では、単

位演算の処理シーケンスが異なる。特にページバッファ方式では、一つの処理ノードでの単位演算を１ページ分連続して行うことも可能である。このようなシーケンスにすると、実装によっては、処理ノード切り替えのオーバーヘッドを省略でき、バンドバッファ割当て方式よりトータル演算時間を若干短くできる可能性がある。従って、S_BとS_PのどちらもM以下になる場合は、より小さい方を選ぶのではなく、ページバッファ方式を優先する方がよい。

【００７１】

ところで、図１１で示すように、ページバッファ割当て方式の場合、最終出力層の出力バッファも他の中間バッファと領域を兼用することが可能である。そこで、（３）式の変わりに、以下の（４）式を用いる。

【００７２】

【数４】

$$S_P' = (I_X \times I_Y) \times \max\left(F_l, F_{l+1} \middle| 1 \le l \le N\right)$$

…（４）

【００７３】

また、中間バッファだけでなく最終出力層も含めて割当て可能なサイズをＭ’とすると、以下の（５）式のように表される。

【００７４】

【数５】

$$M' = M + (I_X \times I_Y) \times F_N$$

…（５）

【００７５】

すなわち、ページバッファ方式での割当てが可能かどうかを判断するのに、（３）式の代わりに（４）式を用いて、Ｍ’と比較するようにしてもよい。つまりページバッファ方式の場合は、実質割当て可能なサイズを増やすことができる。このようにすると、さらにメモリ使用効率を上げられる。

【００７６】

以下、図８および図２６、２７に示すフローチャートを用いて、本実施形態の階層型ネットワークによる演算処理動作を詳細に説明する。図８はＣＰＵ６８での一連の検出処理動作を示すフローチャートである。ＣＰＵ６８におけるソフトウェアによる処理は、所定の設定処理等を行った後、画像入力部６１やＣＮＮ処理部６３等のハードウェア処理回路部をドライブする。

【００７７】

まず、ステップＳ８０１において、ＣＰＵ６８は、検出処理の開始に先立ち、変数やレジスタ等の各種初期化処理を実行する。各処理部はＣＰＵ６８の初期化指示に従って内部のレジスタ、メモリ等を初期化する。またここでは、入力画像サイズ全域を処理領域とする初期設定も行う。本処理を含め、以降ＣＰＵ６８は、ブリッジ６５及び画像バス６４を介してＣＮＮ処理部６３や他のハードウェア回路部にアクセスし、ＣＰＵバスアクセス制御部１１３を介して所定のレジスタやメモリにデータを設定することが可能となる。また、本初期化処理においては、演算部１０１内の重み係数記憶部１２０５（図１２）へ、本パターン検出装置で対応する全ての検出対象に対する全ての処理ノードのコンボリューションカーネルの係数データをロードする。本パターン検出装置は、コンボリューションカーネルの重み係数、シーケンス情報、ネットワーク構成情報等を入れ替える事で、同一のハードウェアで様々な検出対象に対応することができるが、そのためのカーネルをここで一通りロードしておく。

【００７８】

次に、ステップＳ８０２において、ＣＰＵ６８は、ユーザからの検出対象の選択を受け付ける。ここでは、本パターン検出装置が対応する全ての検出対象の中から、画像中より

検出したい所望の検出対象がユーザにより選択される。

【００７９】

検出対象が決定したら、ステップＳ８０３でネットワーク構成管理部１０８のネットワーク構成情報設定部１１０にネットワーク構成情報を設定する。ネットワーク構成情報はネットワークの結合関係を指定するテーブルであり、レジスタファイル或いはＲＡＭ等により構成される。ネットワーク構成情報も重み係数と共に検出対象に応じて異なる値となる。

【００８０】

図９はネットワーク構成情報の一例を示す図であり、図１４に示したＣＮＮの構成情報を表現している。図９において、「対象処理ノード」は図１４に示すネットワークの論理的な第０から第８処理ノードに対応する。なお、論理的な処理ノードとは、演算部１０１を時分割利用する事により実現する論理的な演算処理の単位である。ここで第０処理ノードは入力画像データ面に対応する処理ノードであって便宜上第０処理ノードとしているが、実際には演算は実行しない。

【００８１】

「隣接下層処理ノード数」とは、処理ノードが演算実行時に必要とする下位層の接続数を示す。例えば第４処理ノードの場合、３つの下位層に接続される。ネットワーク構成管理部１０８では、当該隣接下層処理ノード数に応じてメモリアクセスと演算を制御する。

【００８２】

「隣接下層処理ノード」は処理ノードの演算時に必要とする下位層の処理ノードを指定する情報である。例えば第４処理ノードは、第１処理ノード、第２処理ノード、第３処理ノードに接続される。つまり、第４処理ノードの演算時は、第１～３処理ノードの演算結果が参照データとして使用される。

【００８３】

「演算種別」は実行する演算の種別を示す情報であり、ＣＮＮ演算の場合、演算種別に応じて重み係数を選択することになる。すなわち、演算部１０１が図１２に示す構成の場合、ここでの演算種別番号が係数を選択するための「カーネル選択信号」に相当する。ＣＮＮ処理の場合、各処理ノードは、「カーネル選択信号」に応じて選択した、それぞれ異なる重み係数を用いて、コンボリューション演算を実行する。

【００８４】

「参照データ幅」は「演算種別」に対応するコンボリューションカーネルの幅に相当し、「参照データ高さ」はコンボリューションカーネルの高さに相当する。

【００８５】

また、「処理開始ライン」は、該処理ノードにおいて有効な演算出力が可能な画像位置を表す。先に説明したように、コンボリューション演算はカーネルサイズの範囲の周辺画素を掃き寄せる演算であるので、カーネル参照範囲が参照画像データの有効範囲外にはみ出す場合には、有効な演算結果を得ることができない領域となる。横方向には図２２の領域２２０６が有効範囲外の領域（無効領域）に相当するが、同様に縦方向にも無効領域が存在する。これは処理ノードのカーネルサイズに依存し、階層を経ることによって蓄積される。本実施形態のパターン検出装置のＣＮＮでは、ライン単位に処理を行うので、無効領域のラインは計算をスキップした方が全体の処理を高速化できる。つまり、「処理開始ライン」までは、その処理ノードにおける演算の無効領域となるので、処理をスキップすることができる。ちなみに、本実施形態では入力画像の最初のラインの番号を０として開始している。また、同様に開始ラインのみならず終了ラインも存在するが、これは入力画像のライン数I_Yから「処理開始ライン」を引いてさらに１を減じたライン番号となる。

【００８６】

尚、本実施形態では、入力画像に対して各処理ノードは演算可能な最大範囲を演算するように設定しているが、最終的に演算結果を利用する処理ノードの利用したい演算範囲から、前階層に向かって順に演算範囲を逆算してももちろんかまわない。この場合は終了ラインも合わせて情報として持つようにすると、より演算を無駄無く行える。

【００８７】
次にステップＳ８０４において、ＣＰＵ６８は、中間バッファ割当て方式を選択する。この処理の詳細を示すのが、図２６のフローチャートである。中間バッファ割当て方式の選択処理では、まずステップＳ２６０１において、ＣＰＵ６８は、設定されている入力処理画像サイズI_X，I_Yを取得する。そしてステップＳ２６０２において、ＣＰＵ６８は、中間バッファとして利用可能なサイズＭを計算する。このサイズＭは、ワークメモリ１０２で利用可能な容量から、入力画像バッファサイズや出力バッファサイズその他の処理に必要なサイズを差し引いた値となる。
【００８８】
そして、ステップＳ２６０３において、ＣＰＵ６８は、まずページバッファ方式で中間バッファを割り当てた場合に必要な容量S_Pを計算する。この容量S_Pは、上述した式（３）で求めることができる。ステップＳ２６０４において、このS_PがＭ以下であるかどうかを確認する。Ｍ以下であれば、ステップＳ２６０５において、ＣＰＵ６８は、ページバッファ方式を今回の中間バッファ割当て方式として選択し、本処理終了する。尚、ここで使用する値として、上述したように、S_PとＭの代わりにS_P'とＭ'を用いてももちろん良く、このようにするとよりページバッファ方式の選択される率が高まる。
【００８９】
一方、ＭよりS_Pが大きい場合には、処理はステップＳ２６０４からステップＳ２６０６へ進む。ステップＳ２６０６において、ＣＰＵ６８は、バンドバッファ方式で割り当てた場合の中間バッファ容量S_Bを計算する。これは上述の式（２）により求めることができる。次に、ステップＳ２６０７において、S_BがＭ以下であるかどうかを確認する。Ｍ以下であれば、ステップＳ２６０８において、ＣＰＵ６８は、バンドバッファ方式を今回の中間バッファ割当て方式として選択し、本処理を終了する。このように、本実施形態では、複数種類のバッファ割り当て方法のうち、必要となるバッファの合計サイズが中間バッファとして利用可能なメモリ量以下（Ｍ以下）になるバッファ割り当て方法が選択される。
【００９０】
S_P、S_Bが共にＭより大きい場合には、複数種類のバッファ割り当て方法のうち、必要となるバッファの合計サイズが中間バッファとして利用可能なメモリ量以下（Ｍ以下）になるバッファ割り当て方法がない。従って、そのままでは本パターン検出装置で演算処理を行うことができない。この場合、ステップＳ２６０９において、ＣＰＵ６８は、元の入力画像サイズを２分割したサイズを新たな処理画像サイズとして再設定する。この分割処理は、図２５に示すように、カーネルのオーバーヘッドを考慮して行われる。図２５において（ａ）は元の入力画像を示し、太線枠は最終層における累積カーネル領域２５０１を示す。なお、ここで累積カーネル領域とは、最終層処理ノードの参照範囲が有効になるように、順に隣接下層処理ノードを遡って入力画像まで行き着いたときの参照範囲のことである。
【００９１】
これを図２５の（ｂ）の点線領域に示すように、左右に２分割することを考える。左右それぞれの点線領域を有効とするには、累積カーネル領域２５０１を考慮すると、網掛けで示した領域まで入力する必要があることが分かる。
【００９２】
従って、オーバーヘッドを考慮した２分割とは、図２５の（ｃ）に示すものとなる。すなわち、入力画像サイズ（ａ）の単純な２分割の領域に、累積カーネル領域２５０１の１／２（奇数時切捨て）の幅を加えたものとなる。このように分割して演算した結果の両点線枠内の領域を合成すれば、図２５の（ｄ）に示すように、元の入力画像にＣＮＮによる演算を行ったのと同等の演算結果を得ることができる。なお、このように入力データを分割して処理する方式をタイル処理と呼ぶこともある。
【００９３】
本実施形態においては、図２５に示したような横方向の分割処理を優先的に行う。なぜ

ならば、縦方向の分割処理では、ページ方式の中間バッファ使用容量は減るが、バンド方式の中間バッファ方式は変わらないからである。横方向の分割では、どちらの方式であっても一時的なトータル使用容量も削減される。分割可能なサイズは、累積カーネル領域の大きさによって決まる（領域以上のサイズが必要）ので、横方向がこれ以上分割できなくなったら、縦方向の分割を行う。尚、本処理では、分割をするたびに初期値１の分割カウンタをインクリメントするとともに、分割方向も合わせて記憶している。

【００９４】

ステップＳ２６１０において、分割後のサイズが累積カーネルサイズより大きいことを確認し、再度ステップＳ２６０１からの処理を繰り返す。もし、上述のように縦方向分割を駆使しても、分割後サイズが累積カーネルサイズよりも小さくなってしまった場合、その検出処理は本ＣＮＮ演算装置で実行することができない。よって、処理はステップＳ２６１１に進み、ＣＰＵ６８は、処理不可能の判断を行う。これは全処理ノードのカーネルサイズと割当て可能な中間バッファ容量最大値によって決まるので、通常はこのようなことにならない構成を取る。

【００９５】

図８に戻り、ステップＳ８０５において、ＣＰＵ６８は、ステップＳ８０４の選択処理において処理不可能と判別されていないことを確認する。処理不可能と判別されていた場合、処理はステップＳ８０５からステップＳ８１９へ進み、終了判定処理（後述）が実行される。但し、通常はそのようなことのない構成をとる。

【００９６】

ステップＳ８０６において、ＣＰＵ６８は、ステップＳ８０４の選択処理で選択された方式がページバッファ方式であったかどうかを判定する。ページバッファ方式の場合、処理はステップＳ８０７のページバッファ方式シーケンス設定およびステップＳ８０８のページバッファ方式中間バッファ設定へ進む。一方、バンドバッファ方式が選択されていた場合、処理はステップＳ８０９のバンドバッファ方式シーケンス設定およびステップＳ８１０のバンドバッファ方式中間バッファ設定へと進む。

【００９７】

ステップＳ８０７のページバッファ方式シーケンス設定処理と、ステップＳ８０９のバンドバッファ方式シーケンス設定処理において、ＣＰＵ６８は、シーケンス制御部１０９のシーケンス情報設定部１１１にシーケンス情報を設定する。シーケンス情報とは、時分割処理する演算処理単位（本実施形態ではライン単位）のシーケンス動作を規定するテーブル情報であり、ＲＡＭ等に保持される。なお、上述したように、シーケンス情報はＲＡＭ６９に格納され、ＣＰＵ６８からＣＮＮ処理部６３ないに書き込まれる。

【００９８】

図１０はシーケンス情報テーブルの例であって、図１４に示したＣＮＮネットワークで処理を行う場合の、（ａ）がバンドバッファ方式、（ｂ）がページバッファ方式の場合のシーケンスを示している。ここで、「シーケンス番号」はライン単位での処理の順序を示す番号で、テーブルの配列Indexに相当するので実際に数字を保持する必要はない。「処理ノード番号」は図１４の処理ノード番号であって、シーケンス番号に対応する論理的な実行処理ノードを示す。即ち、シーケンス番号１のとき、第１処理ノードに対して演算処理を実行し、シーケンス番号２においては第２処理ノードのライン単位演算処理を実行する。処理ラインは、該シーケンス番号の処理のとき該処理ノードが出力するラインの位置を示している。各処理ノードの処理ラインは、図９で説明した「処理開始ライン」より順にスタートし、１ライン処理が完了する毎にインクリメントするものである。従って、本実施形態のように、テーブル上にシーケンス番号に対応する番号として明記しなくとも、各処理ノード毎に現在の処理ラインをレジスタ等で記憶しておくようにしてもよい。

【００９９】

図１０の（ａ）は、バンドバッファ方式の場合のシーケンス情報テーブルの例であり、１ライン単位処理するごとに、処理ノードを切り替えている。例えば、シーケンス番号“２１”までは、処理ノード番号“１”～“３”までの演算を行っており、シーケンス番号

“２２”において処理ノード番号４の処理を開始している。これは、シーケンス番号“２１”までで、処理ノード番号４の１ライン分の演算処理に必要な参照画像が演算されたためである。このようにバンドバッファ方式では、演算可能になった処理ノードから速やかにライン単位演算を行っていくことにより、参照される側のバッファをライン単位で順次開放可能にしている。これにより、中間バッファを必要最小限の循環数のリングバッファとして構成することが可能となる。

【０１００】

図１０の（ｂ）は、ページバッファ方式の場合のシーケンス情報テーブルの例であり、特定処理ノードの単位演算を連続して行い、該特定処理ノードの演算が有効領域全域について完了した後に、次の処理ノードの演算を開始している。このようにすると、図９および図１４のＣＮＮネットワークにおいて、例えば処理ノード番号“４”と“５”の全有効領域演算が完了すると、処理ノード番号“１”～“３”の出力に割り当てられていた中間バッファ領域は不用となる。従って、最終階層まで演算の完了していないこの時点で、このバッファ領域を開放して同領域を処理ノード番号７、８、９の出力用として割り当てることができるようになる。

【０１０１】

ステップＳ８０８のページバッファ方式による中間バッファ設定およびステップＳ８１０のバンドバッファ方式による中間バッファの設定では、各処理ノードに必要な中間バッファ領域が割り当てられる。当該処理は、論理的な処理ノードの数に対応する数のレジスタセット（リングバッファ設定部１０４-1～１０４-n）に値を設定することにより行われる。図１４および図９に示したネットワーク構造の場合、８個のリングバッファ設定部（１０４-1～１０４-8）に所定の値を設定する必要がある。

【０１０２】

リングサイズ設定部１０６には対応する論理的な処理ノードのリングバッファの高さ（リングカウンタの循環数）が設定される。バンドバッファ方式の場合、この循環数は当該処理ノードの出力を参照する次階層の処理ノード（隣接上層処理ノード）のコンボリューションカーネルのうちの最大の高さに相当する。一方、ページバッファ方式の場合は、リングサイズ設定部１０６に設定されるリングバッファの高さは、入力画像のライン数（もしくは各処理ノードの有効領域ライン数）と同じになり、当該リングバッファは実質循環させずに使用されることになる。オフセットアドレス設定部１０７には、対応する処理ノードのリングバッファ先頭アドレスが設定される。

【０１０３】

図１１は図１４に示すネットワークを実現する場合の処理ノードとオフセットアドレス及びリングバッファの高さの関係の一例を示すメモリマップである。図１１において、（ａ）がバンドバッファ方式の場合のメモリマップを、（ｂ）がページバッファ方式の場合のメモリマップを示す。ADRｘ（ｘ：0～8）はオフセットアドレス、BHｘ（ｘ：０～３）がリングバッファの高さ（循環数）に相当する。I_xは入力画像データの幅を示す。図中、第０処理ノードのための領域とは、入力画像データを保持するページバッファ領域である。つまりBH0は入力画像データの高さI_Yに等しい。また本実施形態においてBH1は、次階層（第２階層）の第４処理ノード及び第５処理ノードのカーネルサイズのうちの大きい方、すなわち図９の構成情報テーブルから９ライン分のサイズが設定される。同様にBH2には１３、BH3には１７ライン分のサイズが設定される。一方、ページバッファ方式の場合は、全ての領域の高さはBH0となり、またいくつかの領域は複数の処理ノードによって利用される。本実施形態ではこのように、メモリ１０２を所定の領域に分割して各領域をサイズの異なるリングバッファ或いはフレームバッファとして利用する。

【０１０４】

以上の各種設定処理により、後述のステップＳ８１４で実行されるＣＮＮ演算処理におけるネットワーク演算処理の実行手順が決定されることになる。すなわち、選択された中間バッファ割り当て方式（上記例ではバンドバッファ方式かページバッファ方式のいずれか）に応じてネットワーク演算処理における処理ノードの実行順が決定される。以上の各

種設定を終了すると、ステップＳ８１１から、画像入力部６１、前処理部６２、ＣＮＮ処理部６３に対して各処理の開始を指示する。

【０１０５】

まずステップＳ８１１において、ＣＰＵ６８より処理開始の指示を受けた画像入力部６１は、１フレーム分の画像データを取得し、図示しない内部バッファに格納する。画像入力部６１は、画像データの格納が終了するとＣＰＵ６８に対して画像取得終了割り込みを発生する。ＣＰＵ６８はこの割り込みを検知すると、ＤＭＡＣ６６を起動して取得した画像データを前処理部６２の内部メモリ（図示しない）に転送する。前処理部６２は画像データの転送が終了すると、前処理を開始する。前処理部６２は、例えば、予め指定するコントラスト補正情報に従って画像データのコントラストを補正する。前処理部６２は補正処理を終了するとＣＰＵ６８に対して割り込みを発生する。ＣＰＵ６８はこの割り込みを検知すると、再びＤＭＡＣ６６を起動し、前処理部６２によって補正された画像データをＣＮＮ処理部６３内のメモリ１０２の入力画像バッファ（図１１の第０処理ノード領域に相当）に転送する。

【０１０６】

次に、ステップＳ８１２において、ＣＰＵ６８は、分割処理カウンタを設定する。これは上述のステップＳ８０４において、中間バッファ割当て方式の選択処理を行った際に、入力画像（処理画像）が分割指定された場合に１よりも大きい値となる。分割無しの場合は１が設定される。そしてステップＳ８１３において、ＣＰＵ６８は、分割無しの場合は画像全域を、分割有りの場合は初回の処理領域（例えば図２５の左右どちらか）を、処理領域として設定する。

【０１０７】

次に、ステップＳ８１４において、ＣＰＵ６８がＣＮＮ処理部６３に対し演算開始トリガを送ることにより、検出処理が開始される。ＣＮＮ処理部６３におけるハードウェア処理について、図２７のフローチャートを参照して、以下に説明する。

【０１０８】

先ず、ステップＳ２７０１において、シーケンス制御部１０９は、処理ノードを決定する。シーケンス制御部１０９は、上述したシーケンス情報設定部１１１に記載されたシーケンス情報テーブルを、ライン単位演算毎に上から辿り、毎回の処理ノードを決定する。図１０の（ａ）に示す例の場合、初回のシーケンスでは処理ノード番号１を選択する。シーケンス制御部１０９はシーケンス回数をカウントするシーケンスカウンタを有し、シーケンス単位（この場合ライン単位の処理毎）でカウントアップする。このカウンタは初期値１であって、図１０のシーケンス情報テーブルのIndexとして使用できる。つまり、シーケンス制御部１０９は、シーケンスカウンタをアドレスとしてシーケンス情報テーブル（図１０）を参照する事により処理対象処理ノードを決定する。

【０１０９】

ステップＳ２７０２において、処理ノードの演算に必要な参照データをメモリ１０２から読み出す。より具体的には、先ず、ネットワーク構成管理部１０８がシーケンス制御部１０９の出力するシーケンス指示情報に従って参照データに対応するリングバッファ設定部１０４を選択する。すなわち、リングバッファ設定部１０４-1～１０４-nの何れかを選択する。例えば、ステップＳ２７０１で第１処理ノードが選択された場合、図９に示すネットワーク構成情報テーブルの内容に従って、「接続ノード数が１」「接続先ノードが第０処理ノード」「演算種別１」が決定される。ネットワーク構成管理部１０８はこのネットワーク構成情報テーブルの内容に従ってノード選択信号をセレクタ１１２１，１１２２に出力し、参照すべきリングバッファに対応したリングバッファ設定部の出力を選択する。例えば初回のシーケンスでは対象処理ノードが第１処理ノードであるので、その隣接下層ノードである第０処理ノードに対応する選択信号を出力する。選択されたリングバッファ設定部の情報（この場合、第０処理ノードに対応したリングカウンタ値、オフセットアドレス値）に従ってメモリアクセス制御部１０３は読み出すメモリの先頭アドレスを生成する。

【０１１０】
図１６はメモリアクセス制御部１０３の内部を説明する図である。また、図１５はメモリアクセス制御部１０３の参照データの読み出し動作を説明する図である。
【０１１１】
図１５において、１５０１はバンドバッファ方式におけるリングバッファ、１５０２は演算するコンボリューションカーネルの参照ウインドウに相当する大きさ、１５０３はコンボリューションカーネルの重み係数列の様子を説明する図である。ここではコンボリューションカーネルサイズが６×６の場合について例示している。重み系数列１５０３において、W00～W05は１行目のデータ列に対する重み系数列、W10～W15は２行目のデータ列に対する重み係数列であり、以下同様に各データ列に対する重み係数列を示している。コンボリューション演算時は当該係数値と対応する位置の参照データの積和演算処理が実行される。
【０１１２】
WIDTHは特徴面の幅（即ち本実施形態の場合、入力画像データの幅I_Xに相当）、L3～L8は特徴面の３行目から８行目のラインデータであることを示す。またA1～A6は夫々対応するラインの先頭メモリアドレスである。
【０１１３】
メモリアクセス制御部１０３において、制御部１６０１はネットワーク構成管理部１０８の出力する動作制御信号に従って各処理部及びメモリに対するコマンド信号（Read/Write制御信号）を生成する。１６０２は列カウンタであり、今回の演算で使用するカーネルサイズに等しい参照ウインドウ１５０２（図１５の点線枠で示された範囲）のデータバッファ上の位置（横方向の位置）を示す。アドレス変換部１６０５が生成する行先頭アドレスA1～A6と、列カウンタ１６０２及び後述のウインドウカウンタ１６０７のカウンタ値を加算器１６０３で加算する事で、リングバッファ内の各行内のデータをアクセスするためのメモリアドレスが生成される。１６０７はウインドウカウンタであり、参照範囲の各行に対して横方向に連続する参照画素を取り出すための参照ウインドウ幅（カーネル幅に相当）カウンタである。ウインドウカウンタ１６０７は、参照ウインドウ幅分の画素をカウントすると０にリセットされる。尚、列カウンタ１６０２やウインドウカウンタ１６０７はネットワーク構成管理部１０８が保持するネットワーク構成情報（図９）の内容に従って演算種別の変更毎に設定される。
【０１１４】
すなわち、メモリアドレスは、アドレス変換部１６０５の生成する行先頭アドレス、参照ウインドウの列位置を指定する列カウンタ１６０２のカウンタ値、参照ウインドウ内の画素位置を指定するウインドウカウンタ１６０７の各出力を加算した値である。メモリアクセス制御部１０３は、このようにして生成されたメモリアドレスをメモリ１０２に対して逐一出力する。
【０１１５】
例えば、アドレス変換部１６０５には、オフセットアドレス設定部１０７からの各処理ノードに割り当てられたオフセットアドレスと、リングカウンタ１０５からの最終行カウンタ値が入力される。そして最終行カウンタ値の示す行を縦方向の最後の位置として、カーネル高さ分のバッファ内の各行の先頭アドレスA1～A6を順次出力する。ここで、最終行カウンタ値は、図１５に示すバンドバッファ方式の場合、リングバッファの値と一致し、リングバッファ内で最新の行が入っている位置を示すものとなる。最終行カウンタ値から逆算することにより、図１５の例ではL3の入っている行の先頭アドレスすなわちA3から、順にA4,A5,A6,A1,A2と出力される。尚、特に図示はしないがページバッファ方式の場合は、各シーケンスに対し図１０に示した処理ライン番号に、カーネル高さ（参照データ高さ）の１／２（切捨て）を加えた値を最終行カウンタ値とする。なお、１ライン分のデータ投入によりリングカウンタがインクリメントされ、アドレス変換部１６０５には、リングカウンタからの最終行カウンタ値が入力される。したがって、アドレス変換部１６０５は、カーネル幅のカウントを行う毎に次の行の先頭アドレスを発生することになる。

【０１１６】
双方向制御部１６０４はメモリデータバスの双方向制御を司るバッファであり、制御部１６０１の出力する制御信号に従ってデータバスの方向制御をする。１６０６はコンボリューション演算に必要な参照データを一時的に保持するキャッシュメモリである（以下、参照データキャッシュという）。上記のアドレス変換結果に基づいて得られた参照ウインドウ内の参照データは、参照データキャッシュ１６０６に格納される。制御部１６０１はウインドウカウンタ１６０７を更新しながら参照データキャッシュ１６０６を制御する事で、列方向に連続する参照データをキャッシュする。アドレス変換部１６０５による先頭アドレスの出力順に従って、参照データキャッシュ１６０６には元の正しいライン順でデータが格納される。

【０１１７】
以上のようにして、メモリアクセス制御部１０３によってメモリ１０２からキャッシュへの参照データ群の読み出しを終了すると、処理はステップＳ２７０３に進む。ステップＳ２７０３において、演算部１０１は、コンボリューション演算処理を開始する。ネットワーク構成管理部１０８は、構成情報テーブルに記録された「演算種別」情報（カーネル選択信号に対応する）に従って演算部１０１の重み係数を指定し、演算部１０１を駆動する。演算部１０１の乗算器１２０１は、メモリアクセス制御部１０３の参照データキャッシュ１６０６に格納された参照データを読み出し、演算種別情報で指定された重み係数を用いてコンボリューション演算処理を実行する。そして、ステップＳ２７０４において、演算部１０１の累積加算器１２０２は、コンボリューション演算処理の演算結果を累積加算する。

【０１１８】
次に、ステップＳ２７０５において、全ての接続先処理ノードの参照データについてコンボリューション演算処理を実施したか否かを判定し、コンボリューション演算が未実施の参照データがあれば処理をステップＳ２７０４に戻す。例えば、図９に示す例において、処理対象ノードが第４処理ノードの場合、接続先ノード数は３である。この場合、構成情報テーブルの内容に従って、第１処理ノードの結果、第２処理ノードの結果、第３処理ノードの結果に対するコンボリューション演算処理が順次実行され、累積加算器１２０２に累積、保持される。各処理ノードに対する参照データの読み出しと演算のシーケンスは前述した方法と同じである。即ち、メモリアクセス制御部１０３は処理ノード毎に異なるリングカウンタ値、オフセットアドレス値等の情報に従って、必要な参照データ群をメモリ１０２から参照データキャッシュ１６０６に読み出す。そして、演算部１０１は当該キャッシュデータに対してコンボリューション演算を実行する。

【０１１９】
全ての接続先ノードに対する演算を終了すると処理はステップＳ２７０５からステップＳ２７０６に進む。ステップＳ２７０６において、非線形処理部１２０３は、累積加算器１２０２の出力を非線形変換する。

【０１２０】
次に、ステップＳ２７０７において、ＣＮＮ処理部６３は、演算部１０１で得られた非線形変換結果をメモリ１０２に格納する。より具体的には、まずネットワーク構成管理部１０８は自身の処理ノードに関するリングバッファ設定部１０４を選択する。例えば、第１処理ノードを演算している場合、第１処理ノードに対応するリングバッファ設定部１０４を選択する。メモリアクセス制御部１０３はここで指定されたリングバッファ設定部１０４のリングカウンタ１０５が示す行の次の行を先頭アドレスとしてメモリアドレスを生成する。ページバッファ方式であっても、リングカウンタには常に最新の格納済み行番号が示されている。なお、ライト動作時は制御部１６０１によって、ウインドウカウンタ１６０７は０に初期化されている。メモリアクセス制御部１０３は生成した先頭アドレスに演算結果を書き込む。書き込みを終了すると列カウンタ１６０２の値を１インクリメントする。列カウンタ１６０２は１つの演算結果書き込み毎にインクリメントする。従って、次の処理時は１列分（１画素分）ずれた領域の参照データ群が読み出される。

【０１２１】
図１７はここで説明した演算の様子をネットワーク構成管理部１０８、メモリアクセス制御部１０３、演算部１０１別に模式的にタイムチャート化した図である。上段がネットワーク構成管理部１０８の動作を示し、メモリアクセス制御部１０３及び演算部１０１はネットワーク構成管理部１０８の指示に従って各処理を実行する。
【０１２２】
上述したように、ステップＳ２７０１において、ネットワーク構成管理部１０８は、シーケンス制御部１０９からのシーケンス制御指示情報に従って処理ノードを選択する（１７０１）。そして、ネットワーク構成情報テーブルを参照して、接続ノード数を設定する（１７０２）。続いて、ネットワーク構成管理部１０８は、選択された参照ノードに関する情報（リングカウンタ値、オフセットアドレス値等）をメモリアクセス制御部１０３に通知し、参照データの読み出しを指示する（１７０３）。メモリアクセス制御部１０３は、通知されたリングカウンタ値、オフセットアドレス値を用いてメモリ１０２から参照データを読み出し、参照データキャッシュ１６０６にキャッシュする（１７０４，１７０５）。メモリアクセス制御部１０３による参照データの読み出しが完了すると、ネットワーク構成管理部１０８は、演算部１０１に対して、演算の開始を指示する。演算部１０１は、参照データキャッシュ１６０６にキャッシュされた参照データを読み出してコンボリューション演算処理を実行する（１７０６，１７０７）。演算部１０１におけるコンボリューション演算処理が完了すると、ネットワーク構成管理部１０８は、次の参照ノードについて同様の処理（１７０９～１７１３）を繰り返す。全ての参照ノードについてコンボリューション演算を完了すると、ネットワーク構成管理部１０８は、演算部１０１に非線形変換処理を実行させ（１７１４）、特徴面における１画素の演算結果を得る。この演算結果をメモリ１０２に格納するために、ネットワーク構成管理部１０８は、上記処理ノードに関する情報（リングカウンタ値、オフセットアドレス値等）をメモリアクセス制御部１０３に通知し、演算結果の書込みを指示する。メモリアクセス制御部１０３は、通知されたリングカウンタ値、オフセットアドレス値を用いてメモリ１０２に1行分の演算結果を書き込む（１７１５，１７１６，１７１７）。そして、列カウンタを１６０２をインクリメントする（１７１８）。
【０１２３】
以上の処理を１ライン分繰り返し（Ｓ２７０８）、処理を終了すると、処理はステップＳ２７０８からステップＳ２７０９へ進む。ステップＳ２７０９において、ネットワーク構成管理部１０８は処理中の演算ノードに対応するリングバッファ設定部１０４のリングカウンタ１０５をインクリメントする。リングカウンタ１０５の更新は１ラインの処理終了毎に行われる。リングカウンタ１０５はカウント値がリングサイズ設定部１０６の値に等しくなった場合０に初期化される。つまり、リングカウンタ１０５はリングサイズを基準にして循環する。但し、ページバッファ方式の場合はリングサイズ設定部１０６の値が入力画像サイズ高さと等しくなるので、実質０に戻ることはない。この様に、メモリ１０２に対するアクセスを論理的な処理ノード毎にリングカウンタ１０５の動作に伴って処理する事でメモリ１０２上の所定の領域をサイズ（循環数）の異なる複数のリングバッファとして独立に使用する事ができる。即ち図１１で示すメモリマップ上の領域を夫々リングバッファとして利用する事になる。
【０１２４】
次にステップＳ２７１０において、ＣＮＮ処理部６３は、全ての処理ノードが演算を終了したか否かを判定する。ここではシーケンス情報テーブル（図１０）に記された最後のシーケンスまでを終了したか否かが判定される。なお、シーケンス制御部１０９は図示しないシーケンスカウンタを予め設定されたシーケンス数と比較する事で終了判定を行っても良い。或いは、シーケンス制御部１０９は、テーブルの最後に付加された、予め定められたTerminationデータを検出することによっても終了判定を行うようにしてもよい。演算が終了していない場合、処理はステップＳ２７１０からステップＳ２７０１に戻る。そして、ＣＮＮ処理部６３は、シーケンスカウンタを更新し、カウンタ値に対応するテーブ

ルを参照する事で次に処理する処理ノード番号を取得する。処理ノードを決定すると、シーケンス指示情報に従ってネットワーク構成管理部１０８は次の処理ノードに対する処理を開始する。異なる処理ノードを処理する場合も、リングバッファ及び演算に関する各種パラメータが異なるだけであり、前述した処理と同様の動作が繰り返される。

【０１２５】

尚、図１０に示した様に、演算処理は下位層から順次リングバッファに特徴データを格納しながら処理を進めるが、バンドバッファ方式の場合とページバッファ方式の場合では処理の順序が異なる。

【０１２６】

以上、ステップＳ２７０１～Ｓ２７１０の処理を繰り返す事で、所定のＣＮＮネットワークに基づく各特徴面の演算が、ライン単位で時分割処理しながら実行される。そして、ＣＮＮ処理部６３は全てのシーケンスを終了すると、ステップＳ２７１１において、ＣＰＵ６８に対して割り込みを発生する。

【０１２７】

図８に戻り、ＣＰＵ６８は割り込みを検知すると、ステップＳ８１４のＣＮＮ演算処理が完了したと見なし、ステップＳ８１５において、出力画像の取得処理を行う。この処理において、ＣＰＵ６８は、ＤＭＡＣ６６を起動してＣＮＮ処理部６３から必要な演算結果をＲＡＭ７０に転送する。本実施形態では、最終層の第８処理ノードの出力結果を吸い上げている。尚、ステップＳ８１２で設定した分割処理カウンタが１よりも大きい場合（入力画像が分割されて処理された場合）は、ＲＡＭ７０に用意する出力データ格納領域の中で、ステップＳ８０４で定めた領域に対応する位置へとデータを転送する。

【０１２８】

ステップＳ８１６において、ＣＰＵ６８は、分割カウンタをデクリメントする。そして、ステップＳ８１７において、分割カウンタが０になるまでステップＳ８１３からの一連の処理を繰り返す。この結果、最終的に入力画像一面分に対応するＣＮＮ演算結果がＲＡＭ７０上に格納される。

【０１２９】

演算結果がＲＡＭ７０上に格納されると、ステップＳ８１８において、ＣＰＵ６８は、判定処理を実行する。この判定処理では、ＲＡＭ７０上に吸い上げた最終層処理ノードの出力である特徴データを利用して対象物の検出状況を判定する。例えば、所定のしきい値で特徴データを２値化しその重心を取得する等の方法で対象物の有無を判定するという判定処理が行われる。

【０１３０】

以上で入力画像に対する１検出対象の検出処理が完了する。次の入力画像や検出対象を変更しての処理を行わないならば、本処理を終了する（ステップＳ８１９）。一方、検出対象を変更する場合、処理はステップＳ８２０からステップＳ８０２に戻り、各種パラメータが再設定される。そして、上述の処理を繰り返し、検出対象に応じた重み係数／ネットワーク構成情報／シーケンス情報をそれぞれ更新する。更にリングバッファ設定部１０４のリングカウンタも新たな重み係数及びネットワーク構成情報に応じて再設定する。これにより、論理的な処理ノードは、検出対象に応じて、メモリ１０２を異なるサイズのリングバッファとしてマッピングし処理を行う。

【０１３１】

一方、ステップＳ８２０において検出対象を変更しない場合は、ステップＳ８２１に処理を進める。ステップＳ８２１とＳ８２２において、ＣＰＵ６８は、リングバッファ設定部１０４-1～１０４-nのリングカウンタ１０５及びシーケンス制御部１０９の内部カウンタ等を初期化する。そして、処理をステップＳ８１１に戻し、画像データの取得から再開する。即ち次のフレーム画像に対して同じ検出処理を実行する。

【０１３２】

以上、第１実施形態によれば、論理的な処理ノードに毎にリングバッファを制御するリングバッファ設定部１０４が設けられ、ネットワーク構成と目的に応じてリングバッファ

のサイズ（循環数）が設定される。そして、この構成において、複数の中間バッファ割当て方式（ページバッファ方式、バンドバッファ方式）から最適なものをネットワーク構成に基づいて選択可能としている。この構成により、同一のハードウェアでより多くの種類のコンボリューショナルニューラルネットワーク等の階層的な演算処理を処理する事が可能になる。更に、当該複数の中間バッファ割当て方式に優先順位を設けたことにより、同一条件化でより高速な演算の可能な方式を選択することができる。また、どの中間バッファ割当て方式でも所定の範囲内の容量に収まらない場合に、入力画像を分割してタイル処理することにより、より多くの種類の階層演算に対応可能としている。

【０１３３】

＜第２実施形態＞

第１実施形態では全ての論理的な処理ノード毎にリングバッファのサイズを設定可能な構成について説明したが、本発明はこれに限るわけではない。例えば、階層毎にリングバッファのサイズを設定する構成とすることも可能である。第２実施形態では、そのような構成について説明する。

【０１３４】

図１９は階層毎にリングバッファのサイズを規定する場合のＣＮＮ処理部６３の構成を示す。図１８は図１９に示すＣＮＮ処理部６３で実現されるＣＮＮネットワーク構成の一例を示す図である。図１８では、階層毎にのみリングバッファのサイズが異なっている様子が示されている。即ち、第１階層の演算結果を格納するためのメモリ領域１８０３ａ～１８０３ｃと、第２階層の演算結果を格納するためのメモリ領域１８０７ａ～１８０７ｂを夫々同じサイズのリングバッファで構成する。図１８は図７と比較して、メモリ領域１８０３ｃのバッファサイズが異なっていることが分かる。

【０１３５】

以下、第１実施形態との違いについて説明する。第２実施形態のＣＮＮ処理部６３は、論理的な処理ノード毎にリングバッファ設定部１９４-1～１９４-nを有する。以下、リングバッファ設定部１９４-1～１９４-nの任意の１つを指す場合は、リングバッファ設定部１９４と記載する。各リングバッファ設定部１９４は、第１実施形態のリングカウンタ１０５とオフセットアドレス設定部１０７に対応するリングカウンタ１９５１とオフセットアドレス設定部１９７１を有する。但し、第２実施形態のリングバッファ設定部１９４は、第１実施形態のリングバッファ設定部１０４が有していたリングサイズ設定部１０６を有していない。その代わりに、第２実施形態によるＣＮＮ処理部６３は、階層型ネットワーク演算の論理的な階層毎に、リングサイズ設定部を有する。図１９の例では、階層数が３までに対応するべく、２つのリングサイズ設定部１９６１ａ，ｂが設けられている。

【０１３６】

リングサイズ設定部１９６１ａ，ｂは夫々複数のリングバッファ設定部１９４-1～１９４-nに接続されている。本例では、リングサイズ設定部１９６１ａは第１階層１８０６の処理ノードに対応した複数のリングバッファ設定部１９４に接続され、リングサイズ設定部１９６１ｂは第２階層１８１０の処理ノードに対応した複数のリングバッファ設定部１９４に接続される。即ち、リングバッファ設定部１９４-1～１９４-nがリングサイズ設定部１９６１ａ，ｂによってグルーピングされている。

【０１３７】

図２０にリングサイズ設定部１９６１ａ，ｂとリングバッファ設定部１９４の関係を図示する。第１階層１８０６のためのリングバッファの制御に利用するリングバッファ設定部１９４として、リングサイズ設定部１９６１ａが接続されたリングバッファ設定部が選択される。一方、第２階層のリングバッファの制御に利用するリングバッファ設定部１９４には、リングサイズ設定部１９６１ｂが接続されたリングバッファ設定部が選択される。演算時、処理ノードに対応するリングバッファ設定部１９４の選択は、ネットワーク構成管理部１０８が保持する管理テーブル情報に従って行われる。

【０１３８】

以下、処理フローに関して、図８を用いて第１実施形態との違いを説明する。第２実施

形態では、ステップＳ８０８およびステップＳ８１０において、リングサイズ設定部１９６１ａ、ｂへの設定が階層毎に行われる。また、階層毎にグルーピングされたリングバッファ設定部１９４の中から構成するネットワークに対応するリングバッファ設定部１９４を選択し、オフセットアドレスを設定する。リングサイズ設定部１９６１ａ、ｂには、図１８のネットワーク構成では、第１階層、第２階層のリングバッファ高さに相当する値が設定される。ページバッファ方式の場合、この値は入力画像のサイズと同じである（ステップＳ８０８）。バンドバッファ方式の場合、この値は次階層の処理ノードのコンボリューションカーネルの内の最大のものの高さとなる。

【０１３９】

他の処理は第１実施形態と同じであり説明を省略する。図１８に示すネットワークを処理する場合、以上の設定でネットワーク構成管理部１０８が所定の論理処理ノードに対応するリングバッファ設定部１９４を選択しながら処理を進める事で第１実施形態と同様にライン単位で処理が実行される。

【０１４０】

以上のように、第２実施形態によれば、リングサイズ設定部を各リングバッファ設定部に設けず、階層毎に設けるようにしたので、リングサイズ設定部を構成するレジスタの数を削減する事が可能になる。

【０１４１】

＜第３実施形態＞

上記第１、第２実施形態ではＣＮＮ処理部６３をハードウェアで実現する場合について説明したが、本発明はソフトウェアにより実現する場合にも適用することが可能である。図２１にソフトウェアで実現する場合の構成例を示す。図２１に示す構成は図６に示す構成からＣＮＮ処理部６３を取り除き、ランダムアクセス可能な高速メモリであるＲＡＭ２１０１を追加したものであるためその違いについて説明する。

【０１４２】

ＣＰＵ６８は前処理部６２の終了割り込みを受け付けるとＤＭＡＣ６６を起動して前処理部６２内のメモリに格納された補正後の画像データをＲＡＭ２１０１に転送する。ＣＰＵ６８はＲＡＭ２１０１に格納した画像データに対して、図２７に示したステップＳ２７０１～ステップＳ２７１０の処理をソフトウェアにより実行する。その場合、ＣＮＮ演算処理の動作に必要なワークメモリとしてＲＡＭ２１０１を使用する。即ち、ＣＰＵ６８は、ＲＡＭ２１０１上に図１１で示すメモリマップを構成し、処理ノードに対応する各メモリ領域をリングバッファとして使用する。もちろんリングバッファはハード構成の場合と同様、バンドバッファ方式とページバッファ方式のいずれかの割当て方式を選択して使用できる。

【０１４３】

尚、第１実施形態のＣＮＮ処理部６３に存在するリングバッファ設定部１０４等はソフトウェア上の変数として構成され、具体的にはＲＡＭ７０上にアサインされる。

【０１４４】

以上の第３実施形態によれば、ＣＮＮ処理部６３をハードウェアにより構成する場合と同様に、処理に必要なバッファメモリを削減する事が可能になる。図２１に示す構成の場合、ＲＡＭ２１０１を少ないメモリで実現する事が出来る。また、ＲＡＭ２１０１を用意せずにＲＡＭ７０をワークメモリとして利用する場合であっても同様である。

【０１４５】

＜他の実施形態＞

上記実施形態では、リングカウンタを使用して、メモリ１０２の所定の連続領域をライン単位で循環しながら使用する方法について説明したが、本発明はこのようなメモリの使用方法に限るわけではない。例えば、リングカウンタに対応するメモリアドレステーブルを有し、当該テーブルを参照する事で、不連続な領域を所定の処理単位に割り当てながら処理する等の方法でも良い。即ち、本発明で規定するリングバッファとは狭義のリングバッファ或いは循環バッファに限定するものではない。

【０１４６】
上記実施形態では、シーケンス情報テーブルの内容に従って論理的な処理ノードをライン単位で時分割処理する場合について説明したが、他の方法を適用しても良い。例えば、読み取り側バッファと書き込みバッファの利用状況から適応的にスケジューリングする等の方法でも良い。特にページ割当て方式を選択した場合、処理単位自体をライン単位からページ単位に切り替えるような実装としてももちろんかまわない。
【０１４７】
また実施形態ではページバッファ方式をバンドバッファ方式より優先的に選択する例を紹介したが、もちろん優先順はこれに限るものではない。例えば、演算結果データ群の最初のラインが出力されるまでのレイテンシを重視する場合には、バンドバッファ方式を優先することも可能である。このようにすると、例えば、全演算結果が出力される前に途中で演算を打ち切るような場合に有利にできることは明らかである。なお、バンドバッファ方式を優先する場合は、図２６において、Ｓ２６０３～Ｓ２６０５の処理と、Ｓ２６０６～Ｓ２６０８の処理を入れ替えれば良い。
【０１４８】
また、中間バッファの割り当て方法としては、上述のバッファ割当て方式に限らず、他のバッファ割当て方式や、複数の方式を混在させるようにしても良いことは明らかである。すなわち、
・各処理ノードの演算結果データを保持するための中間バッファを各処理ノードへの割り当てるための複数種類のバッファ割り当て方法を用意しておき、
・これらのバッファ割り当て方法のそれぞれについて、ネットワーク演算に必要なメモリ量を当該ネットワーク演算の構成に基づいて算出し、
・算出されたメモリ量に基づいて、複数種類のバッファ割り当て方法のうちの１つを選択するように構成することが可能である。
【０１４９】
ここで、算出された必要なメモリ量が、メモリ１０２の中間バッファの割り当てに利用可能なメモリ容量以下となるバッファ割り当て方法が複数存在した場合は、予め定められた優先順位に従って使用すべきバッファ割り当て方法が選択される。また、算出された必要なメモリ量が、上記割り当てに利用可能なメモリ容量以下となるバッファ割り当て方法が存在しない場合には、上述したように入力データを分割して処理する。
【０１５０】
また、上記各実施形態では、特徴抽出結果を入力層と同じ解像度で保持するが、特徴面を入力面に対してサブサンプリングする場合に対しても同様に適用可能である。
【０１５１】
また、上記各実施形態では、最も効率の良い処理単位としてライン単位でシーケンス制御する場合について説明したが本発明はこれに限るわけではない。１ライン以下の単位やブロック単位でシーケンス制御する場合にも適用可能であることは明らかであり、その構成も当業者には明らかである。
【０１５２】
また、上記各実施形態では、本発明をコンボリューショナルニューラルネットワーク演算に適用する場合について説明したが本発明はこれに限るわけではない。所定の参照領域を必要とする様々な階層的な演算処理に対して適用することが可能である。更に、２次元演算に限るわけでもない。
【０１５３】
また、上記実施形態ではリングバッファ設定部１０４-1～１０４-nをレジスタとして提供する場合について説明したが、メモリとして他のパラメータメモリやワークメモリ１０２と共有する構成にしても良い。その場合、回路リソースをより有効に利用できる。即ち、より柔軟なネットワーク構成を実現する事が可能になる。
【０１５４】
また、上記実施形態では、参照データキャッシュ１６０６に２次元の参照データを全て

取り込んだ後に演算部１０１を起動する場合について説明したが、これに限られるものではない。例えば、コンボリューション演算等の場合、参照ウインドウ内の１行単位で演算部１０１を駆動する様に制御する事も可能である。この場合、ウインドウカウンタ１６０７のデクリメントに伴う連続する列方向参照データの読み出しを終了すると、次のラインの参照データ読み出し処理開始前に演算部１０１を駆動し、読み出したデータに対するコンボリューション演算を実行する。演算が終了すると次のラインの参照データ読み出しを開始する。以上の処理を繰り返す。この場合、参照データキャッシュのサイズが参照データ幅分のみで良いため、少ないキャッシュ容量で実現する事が出来る。

【０１５５】

また、第２実施形態では、リングサイズ設定部１９６１ａ，ｂを階層毎に有する場合について説明したが、リングバッファ設定部１０４（１９４）を階層毎に有する構成とすることも可能である。その場合は階層処理単位でリングカウンタを更新する様に制御する。論理的な処理ノードの動作シーケンスに制限が生じる（必ず階層単位でシーケンスする必要が生じる）が、より回路規模を削減する事が出来る。

【０１５６】

また、上記実施形態では、リングサイズ設定部が任意に設定可能な場合（レジスタやＲＡＭにより構成する場合）について説明したが、全て或いは一部が固定的な値として指定される構成とすることもできる。

【０１５７】

以上、実施形態を詳述したが、本発明は、例えば、システム、装置、方法、プログラムもしくは記憶媒体等としての実施態様をとることが可能である。具体的には、複数の機器から構成されるシステムに適用しても良いし、また、一つの機器からなる装置に適用しても良い。

【０１５８】

尚、本発明は、ソフトウェアのプログラムをシステム或いは装置に直接或いは遠隔から供給し、そのシステム或いは装置のコンピュータが該供給されたプログラムコードを読み出して実行することによって前述した実施形態の機能が達成される場合を含む。この場合、供給されるプログラムは実施形態で図に示したフローチャートに対応したコンピュータプログラムである。

【０１５９】

従って、本発明の機能処理をコンピュータで実現するために、該コンピュータにインストールされるプログラムコード自体も本発明を実現するものである。つまり、本発明は、本発明の機能処理を実現するためのコンピュータプログラム自体も含まれる。

【０１６０】

その場合、プログラムの機能を有していれば、オブジェクトコード、インタプリタにより実行されるプログラム、ＯＳに供給するスクリプトデータ等の形態であっても良い。

【０１６１】

コンピュータプログラムを供給するためのコンピュータ読み取り可能な記憶媒体としては以下が挙げられる。例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、ＭＯ、ＣＤ－ＲＯＭ、ＣＤ－Ｒ、ＣＤ－ＲＷ、磁気テープ、不揮発性のメモリカード、ＲＯＭ、ＤＶＤ（ＤＶＤ－ＲＯＭ，ＤＶＤ－Ｒ）などである。

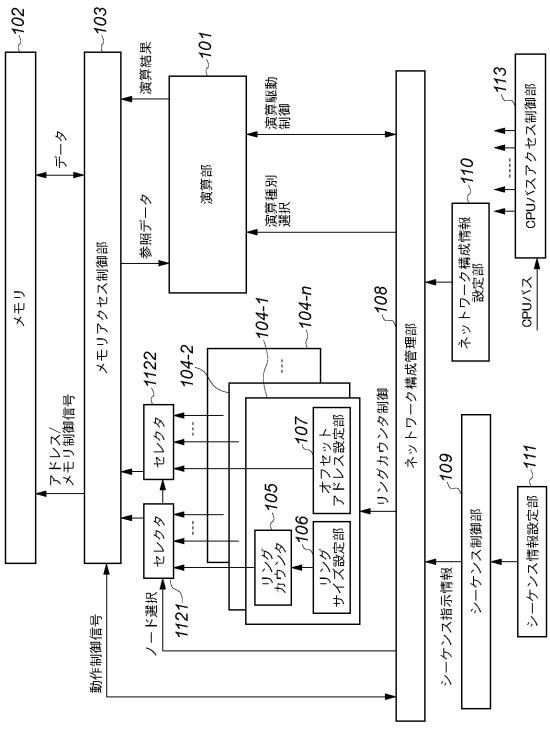
【０１６２】

その他、プログラムの供給方法としては、クライアントコンピュータのブラウザを用いてインターネットのホームページに接続し、該ホームページから本発明のコンピュータプログラムをハードディスク等の記録媒体にダウンロードすることが挙げられる。この場合、ダウンロードされるプログラムは、圧縮され自動インストール機能を含むファイルであってもよい。また、本発明のプログラムを構成するプログラムコードを複数のファイルに分割し、それぞれのファイルを異なるホームページからダウンロードすることによっても実現可能である。つまり、本発明の機能処理をコンピュータで実現するためのプログラムファイルを複数のユーザに対してダウンロードさせるＷＷＷサーバも、本発明に含まれる

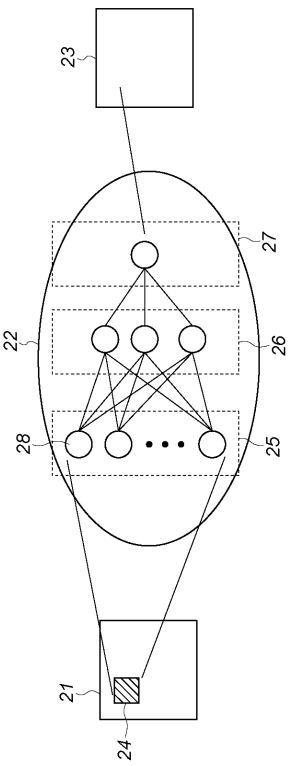
ものである。
【０１６３】
また、本発明のプログラムを暗号化してＣＤ－ＲＯＭ等の記憶媒体に格納してユーザに配布するという形態をとることもできる。この場合、所定の条件をクリアしたユーザに、インターネットを介してホームページから暗号を解く鍵情報をダウンロードさせ、その鍵情報を使用して暗号化されたプログラムを実行し、プログラムをコンピュータにインストールさせるようにもできる。
【０１６４】
また、コンピュータが、読み出したプログラムを実行することによって、前述した実施形態の機能が実現される他、そのプログラムの指示に基づき、コンピュータ上で稼動しているＯＳなどとの協働で実施形態の機能が実現されてもよい。この場合、ＯＳなどが、実際の処理の一部または全部を行ない、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。
【０１６５】
さらに、記録媒体から読み出されたプログラムが、コンピュータに挿入された機能拡張ボ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き込まれて前述の実施形態の機能の一部或いは全てが実現されてもよい。この場合、機能拡張ボードや機能拡張ユニットにプログラムが書き込まれた後、そのプログラムの指示に基づき、その機能拡張ボ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なう。
【図面の簡単な説明】
【０１６６】
【図１】第１実施形態の階層的演算処理装置の構成を説明するブロック図である。
【図２】階層結合型ニューラルネットワークの構成例を説明する図である。
【図３】ニューロンの構成を示す図である。
【図４】ＣＮＮのネットワーク構成例を説明する図である。
【図５】ＣＮＮの特徴抽出の例を説明する図である。
【図６】実施形態による、階層的演算処理を利用した画像処理装置の構成を説明するブロック図である。
【図７】実施形態に関するＣＮＮのネットワークの構成を説明する図である。
【図８】第１実施形態の画像処理装置の動作を説明するフローチャートである。
【図９】ネットワーク構成情報テーブルのデータ構成例を示す図である。
【図１０】シーケンス情報テーブルのデータ構成例を示す図である。
【図１１】メモリの割り当てに関する例を示す図である。
【図１２】演算処部１０１の構成例を説明するブロック図である。
【図１３】リングバッファの例を説明する図である。
【図１４】処理ノードの論理的な接続構成を説明する図である。
【図１５】参照データ群の読み出しを説明する図である。
【図１６】メモリアクセス制御部１０３の構成を説明する図である。
【図１７】ＣＮＮ演算単位の動作タイミングを説明する図である。
【図１８】第２実施形態のＣＮＮネットワークの構成を説明する図である。
【図１９】第２実施形態の演算処理装置の構成を説明するブロック図である。
【図２０】第２実施形態のリングバッファ設定部とリングサイズ設定部の関係を説明する図である。
【図２１】第２実施形態の演算処理装置の構成を説明するブロック図である。
【図２２】ライン単位で演算部１０１が処理を実行する場合の様子を模式的に説明する図である。
【図２３】バンドバッファ方式による中間バッファ割当ての様子の一例を示す図である。
【図２４】ページバッファ方式による中間バッファ割当ての様子の一例を示す図である。
【図２５】実施形態による分割処理の一例を示す図である。

【図２６】中間バッファ割当て方式の選択処理を説明するフローチャートである。
【図２７】ＣＮＮ演算処理を説明するフローチャートである。

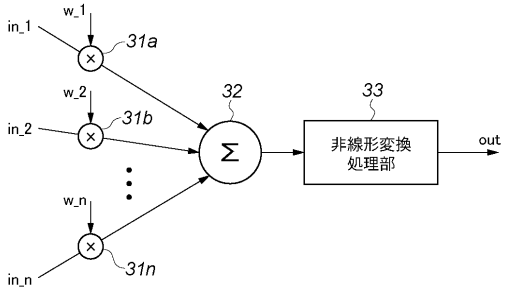
【図１】



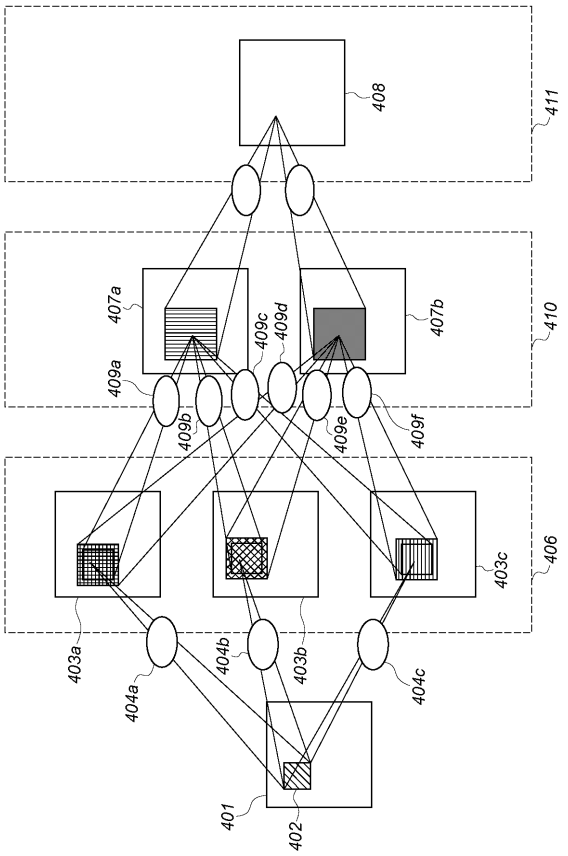
【図２】



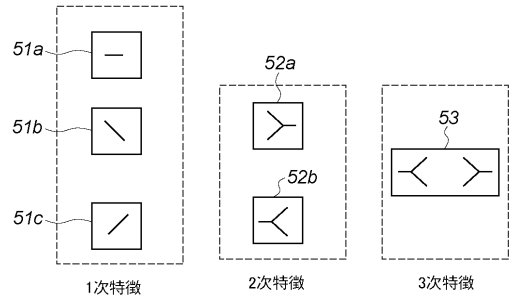
【図３】


in_1
w_1
31a
in_2
w_2
31b
w_n
in_n
31n
32
33
非線形変換
処理部
out

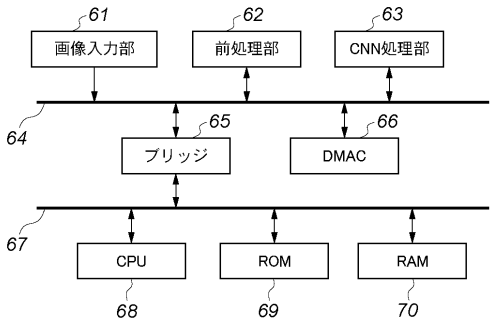
【図４】


408
411
407a
407b
409c
409d
410
409a
409b
409e
409f
403c
406
403a
404b
403b
404c
404a
401
402

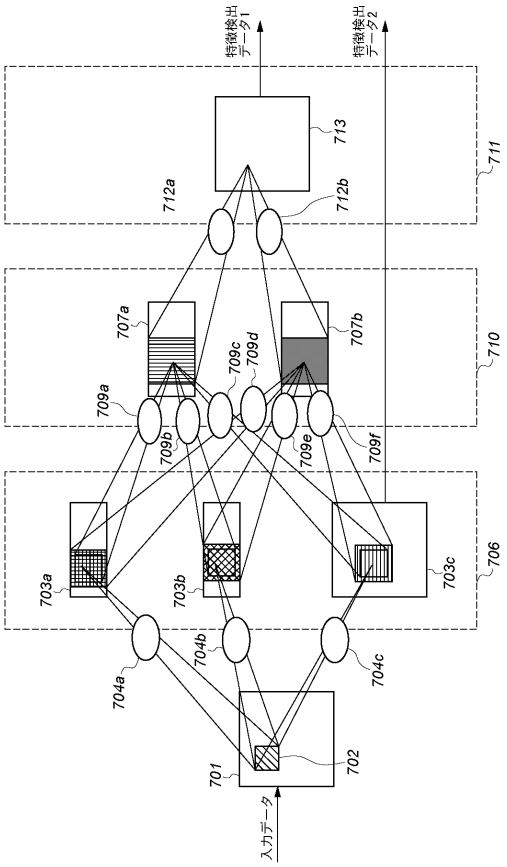
【図５】


51a
51b
51c
52a
52b
53
1次特徴
2次特徴
3次特徴

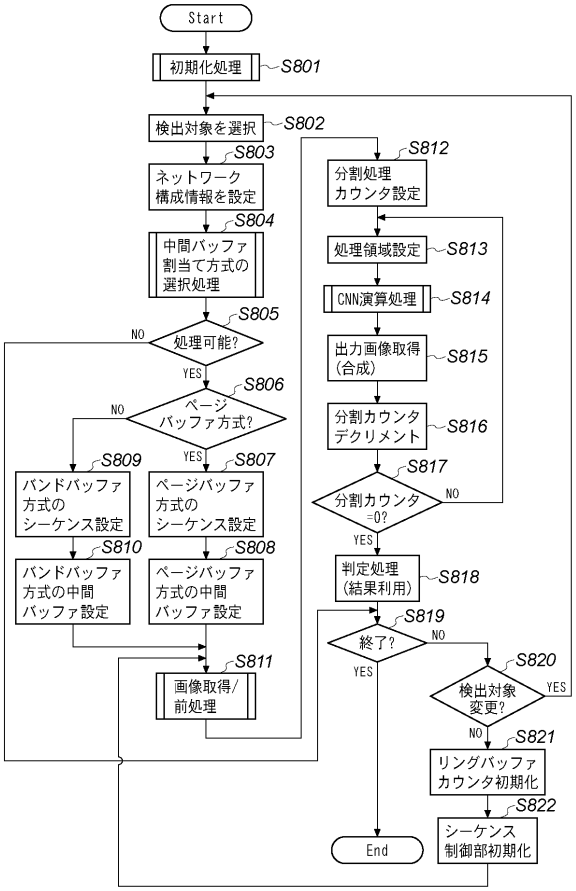
【図６】


61
画像入力部
62
前処理部
63
CNN処理部
64
65
ブリッジ
66
DMAC
67
CPU
ROM
RAM
68
69
70

【図７】


特徴検出
データ1
特徴検出
データ2
713
711
712a
712b
707a
707b
709c
709d
710
709a
709b
709e
709f
703a
703b
703c
706
704b
704a
704c
701
702
入力データ

【図８】



【図９】

対象処理ノード	隣接下層処理ノード数	隣接下層処理ノード	演算種別	参照データ幅	参照データ高さ	処理開始ライン(有効領域オフセット)
第0処理ノード	1	第0処理ノード	演算1	1	1	0
第1処理ノード	1	第0処理ノード	演算2	3	3	1
第2処理ノード	1	第0処理ノード	演算3	3	3	1
第3処理ノード	1	第0処理ノード	演算4	3	3	1
第4処理ノード	3	第1処理ノード	演算4	7	7	4
		第2処理ノード	演算5	7	7	
		第3処理ノード	演算6	7	7	
第5処理ノード	3	第1処理ノード	演算7	9	9	5
		第2処理ノード	演算8	9	9	
		第3処理ノード	演算9	9	9	
第6処理ノード	2	第4処理ノード	演算10	11	11	10
		第5処理ノード	演算11	11	11	
第7処理ノード	2	第4処理ノード	演算12	13	13	11
		第5処理ノード	演算13	13	13	
第8処理ノード	2	第6処理ノード	演算14	17	17	19
		第7処理ノード	演算15	17	17	

【図１０】

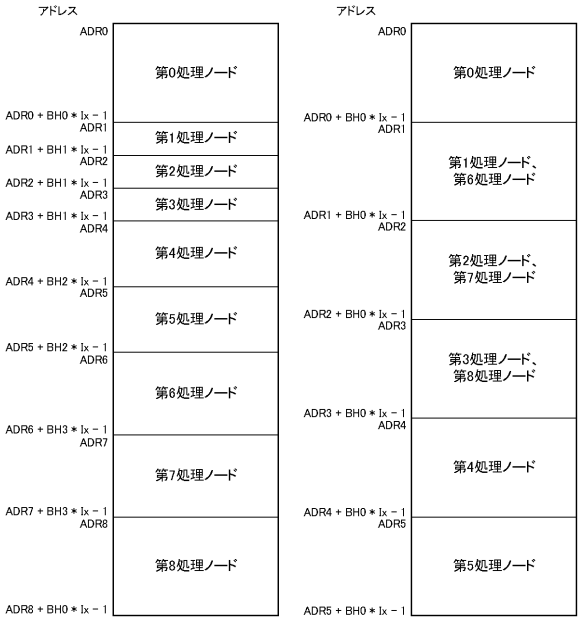
シーケンス番号	処理ノード番号	処理ライン
1	1	1
2	2	1
3	3	1
4	1	2
5	2	2
6	3	2
・	・	・
・	・	・
19	1	7
20	2	7
21	3	7
22	4	4
・	・	・
・	・	・
27	1	9
28	2	9
29	3	9
30	4	6
31	5	5
・	・	・
・	・	・
・	3	19
	4	16
	5	15
	6	10
	・	・
	・	・
	3	21
	4	18
	5	17
	6	12
	7	11
	・	・
	・	・
	3	37
	4	34
	5	33
	6	28
	7	27
	8	19
	・	・
	・	・
	・	・
	1	N-2
	2	N-2
	3	N-2
	4	N-5
	5	N-6
	6	N-11
	7	N-12
	8	N-20

(a)

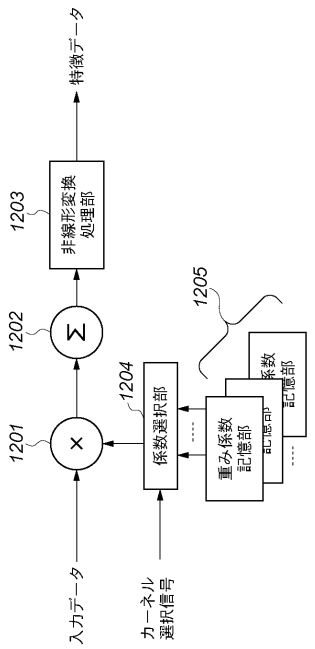
シーケンス番号	処理ノード番号	処理ライン
1	1	1
2	1	2
3	1	3
4	1	4
・	・	・
・	・	・
N-4	1	N-4
N-3	1	N-3
N-2	1	N-2
N-1	2	1
N	2	2
N+1	2	3
・	・	・
・	・	・
2N-3	2	N-3
2N-4	2	N-2
・	3	1
・	3	2
・	・	・
・	・	・
・	3	N-2
・	4	4
	4	5
	4	6
	・	・
	・	・
	4	N-4
	4	N-5
	5	5
	5	6
	・	・
	・	・
	5	N-6
	6	10
	6	11
	・	・
	・	・
	6	N-11
	7	11
	7	12
	・	・
	・	・
	7	N-11
	7	N-12
	8	19
	8	20
	8	21
	・	・
	・	・
	8	N-19
	8	N-20

(b)

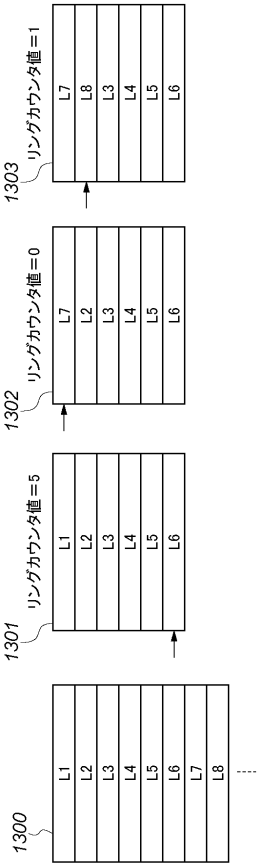
【図１１】



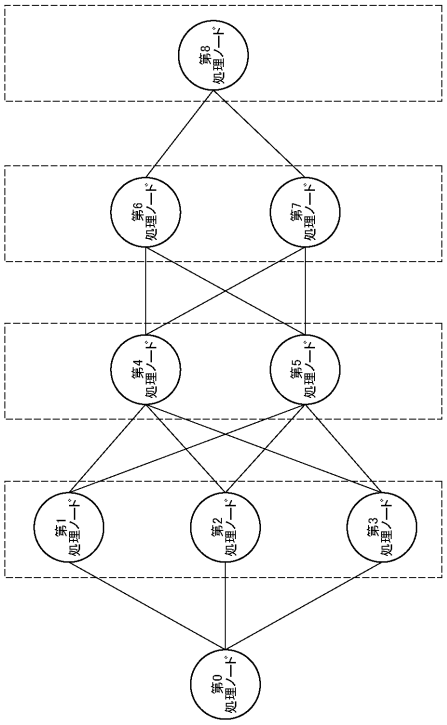
【図１２】



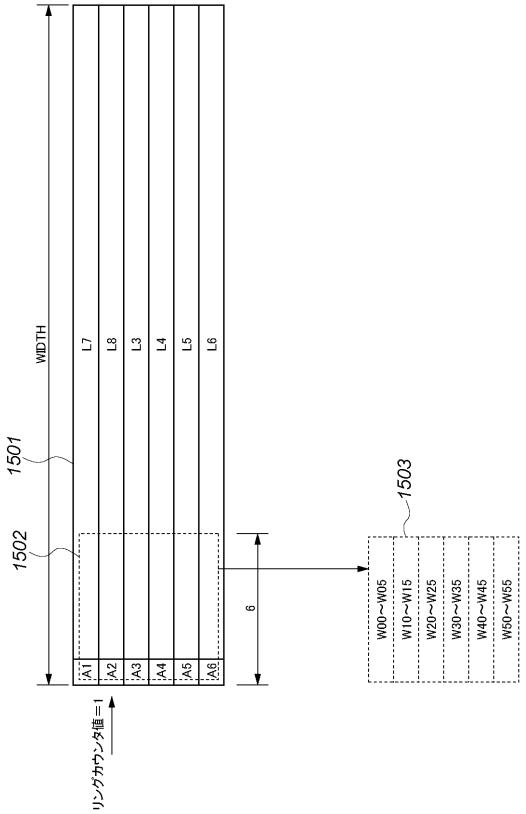
【図１３】



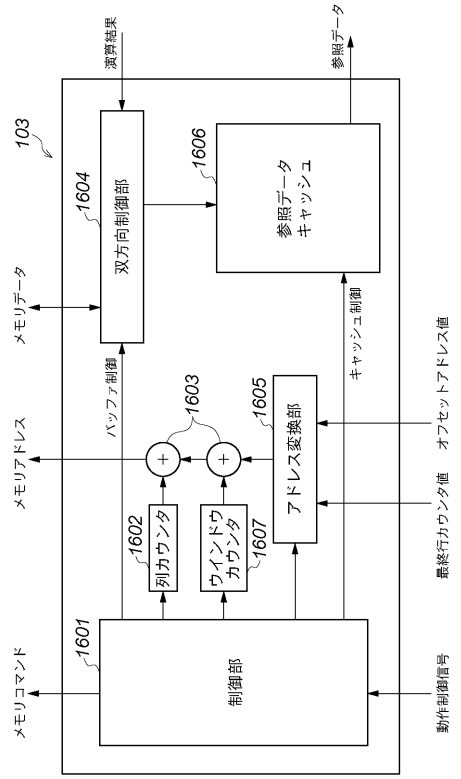
【図１４】



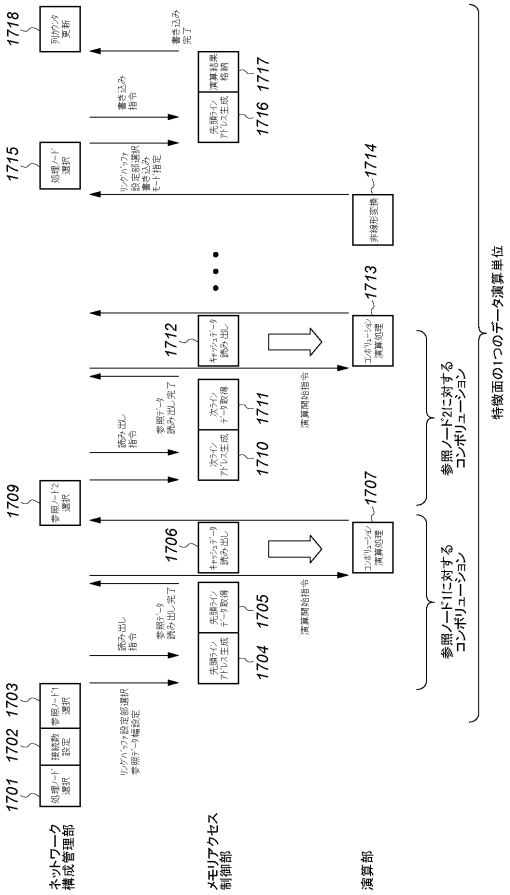
【図１５】



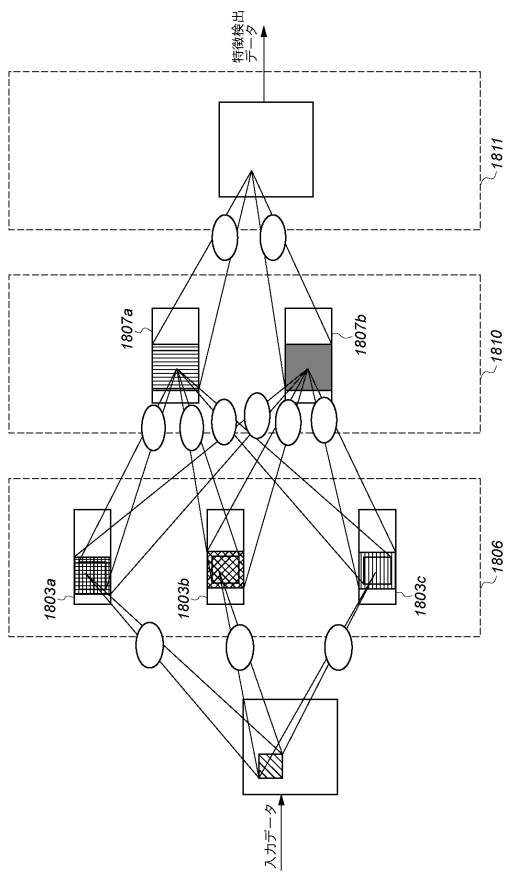
【図１６】



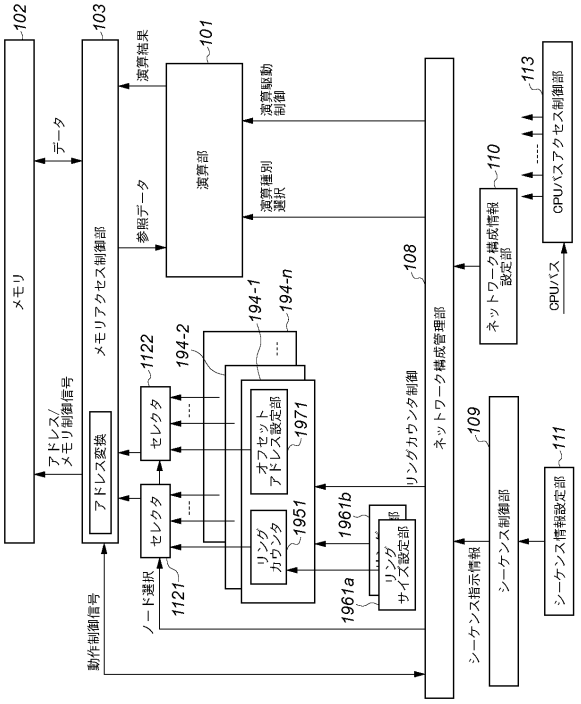
【図１７】



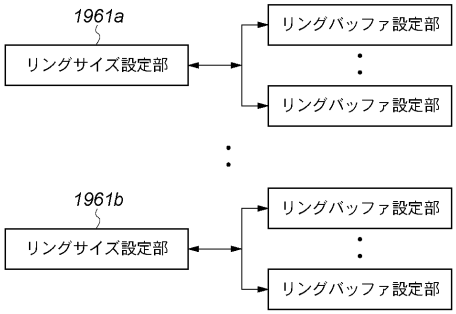
【図１８】



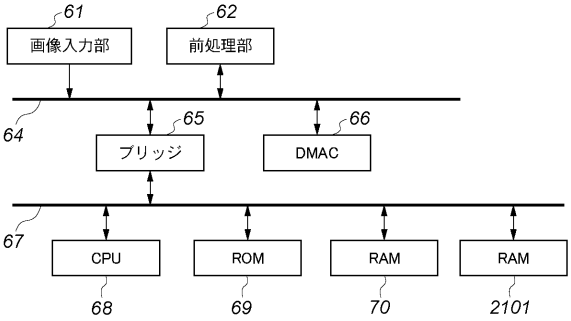
【図１９】



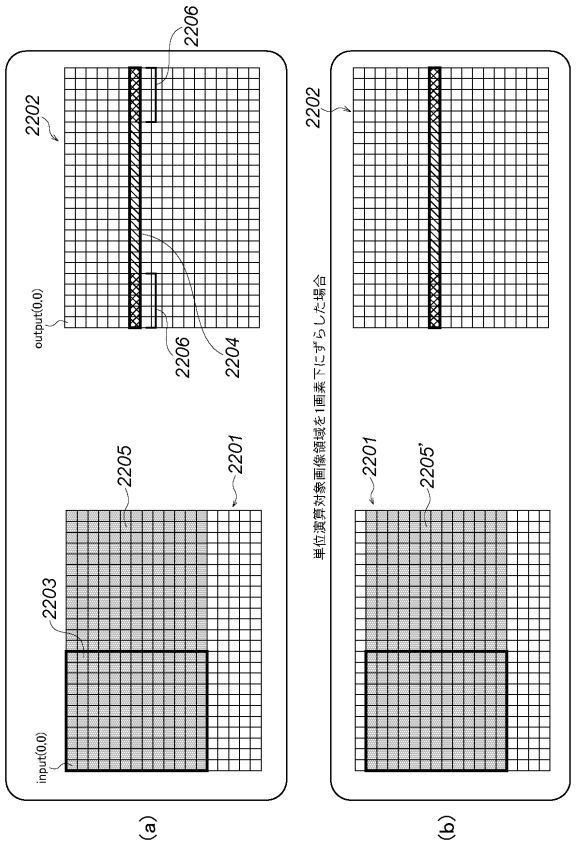
【図２０】



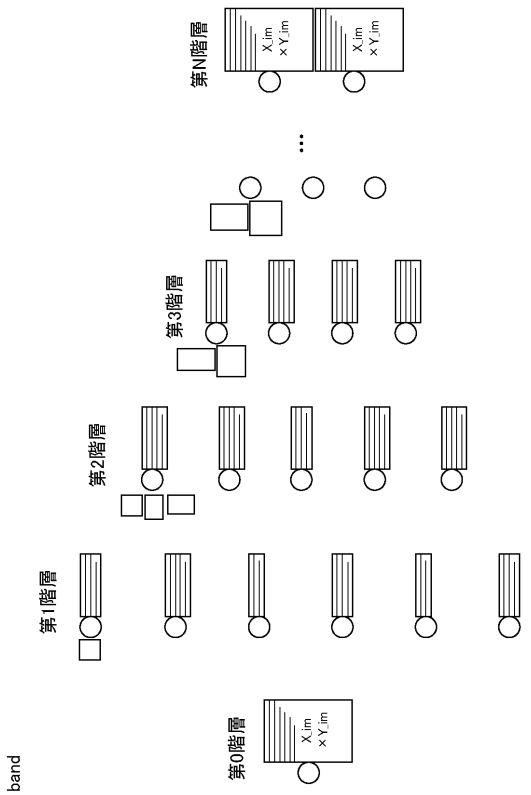
【図２１】



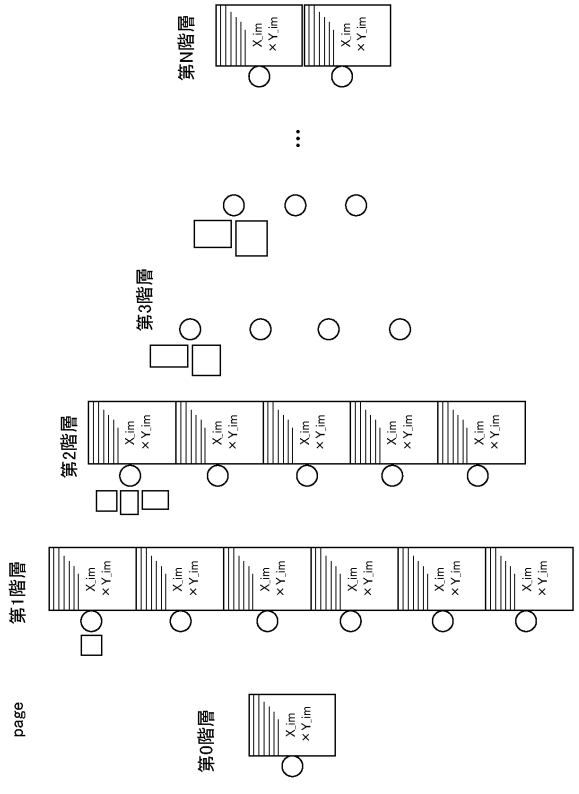
【図２２】



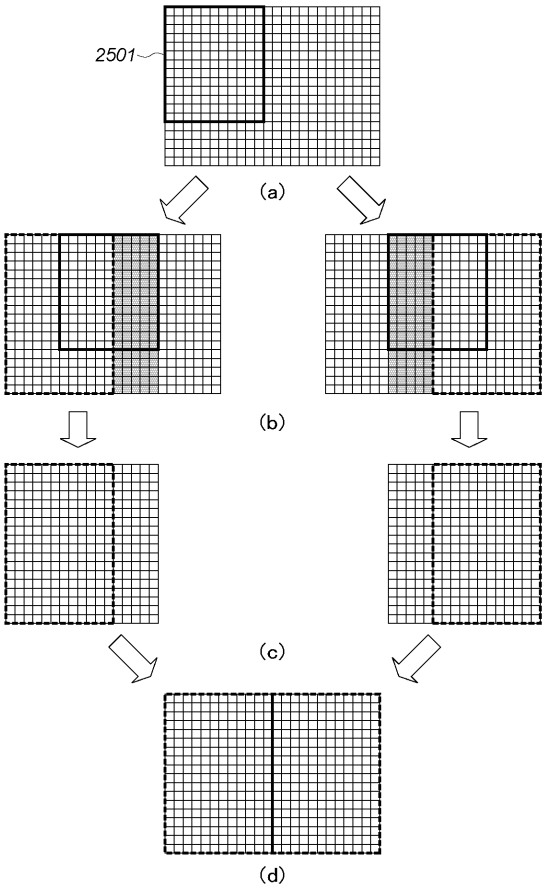
【図２３】



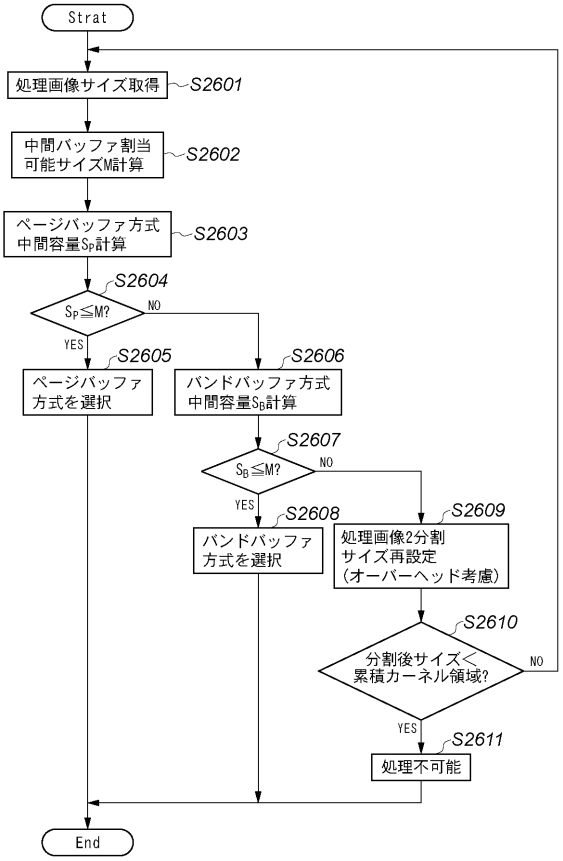
【図２４】



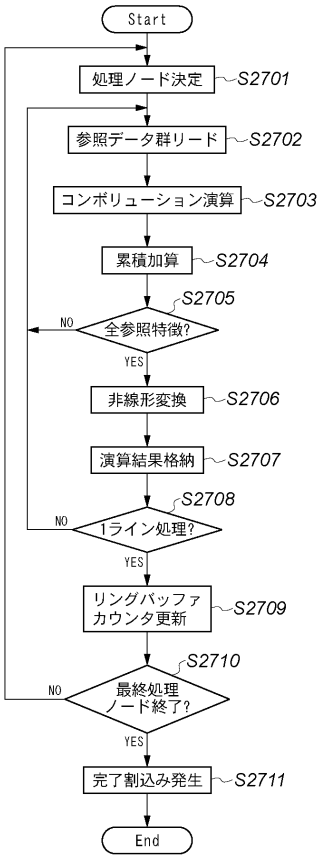
【図２５】


2501
(a)
(b)
(c)
(d)

【図２６】


Strat
処理画像サイズ取得
S2601
中間バッファ割当
可能サイズM計算
S2602
ページバッファ方式
中間容量Sp計算
S2603
S2604
Sp≦M?
NO
YES
S2605
ページバッファ
方式を選択
S2606
バンドバッファ方式
中間容量Sb計算
S2607
Sb≦M?
NO
YES
S2608
バンドバッファ
方式を選択
S2609
処理画像2分割
サイズ再設定
(オーバーヘッド考慮)
S2610
分割後サイズ＜
累積カーネル領域?
NO
YES
S2611
処理不可能
End

【図２７】


Start
処理ノード決定
S2701
参照データ群リード
S2702
コンボリューション演算
S2703
累積加算
S2704
S2705
全参照特徴?
NO
YES
非線形変換
S2706
演算結果格納
S2707
S2708
1ライン処理?
NO
YES
リングバッファ
カウンタ更新
S2709
S2710
最終処理
ノード終了?
NO
YES
完了割込み発生
S2711
End

フロントページの続き

(72)発明者　伊藤　嘉則
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　加藤　政美
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　山本　貴久
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

審査官　塚田　肇

(56)参考文献　特開平０５－１０８５９３（ＪＰ，Ａ）
特許第２７２９９８７（ＪＰ，Ｂ２）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｎ　　３／０６３
Ｇ０６Ｎ　　３／００
Ｇ０６Ｔ　　７／００