

ハードウェア・ソフトウェア協調設計による SVM を用いた大腸内視鏡診断支援システム

岡本拓巳^{†1}, 小出哲士^{†1}, 清水達也^{†1}, 杉幸樹^{†1}, Anh-Tuan Hoang^{†1},
玉木徹^{†2}, Bisser Raytchev^{†2}, 金田和文^{†2}, 吉田成人^{†3†1}, 三重野寛^{†3}, 田中信治^{†4}

近年の大腸ガン患者数の増加に対して, 本研究では病状を定量的に評価し医師の診断を支援する大腸 NBI 拡大内視鏡画像診断支援システム (**eCAD** システム) の実現を目指している. 本稿では, 大腸 Narrow Band Imaging (NBI) 拡大内視鏡 Full HD (1920×1080 pixel) 画像に対して, 局所特徴量から病理タイプを識別する, Support Vector Machine (SVM) を用いたタイプ識別部を FPGA 上で実装し, 特徴抽出部および特徴変換部をソフトウェア上に実装した, ハードウェア・ソフトウェア協調設計による診断支援システムについて述べる. 実装評価の結果, Scan Window あたりのタイプ識別部の処理時間は約 0.06 msec とソフトウェア実装と比較して約 2.7 倍の高速化が確認できた.

Hardware Software Co-designed Computer-Aided Diagnosis System for colorectal endoscopy with SVM

TAKUMI OKAMOTO^{†1} TETSUSHI KOIDE^{†1} TATSUYA SHIMIZU^{†1}
KOKI SUGI^{†1} ANH-TUAN HOANG^{†1} HIKARU SATO^{†1}
TORU TAMAKI^{†2} BISSER RAYTCHEV^{†2} KAZUFUMI KANEDA^{†2}
SHIGETO YOSHIDA^{†3†1} HIROSHI MIENO^{†3} SHINJI TANAKA^{†4}

With the increase of colorectal cancer patients in recent years, the needs of quantitative evaluation of colorectal cancer are increased, and the Computer-Aided Diagnosis (CAD) system which supports doctor's diagnosis is essential. Our goal is to develop a CAD system for colorectal endoscopic images with Narrow Band Imaging (NBI) magnification findings. We call the system "eCAD system". In this paper, we introduce a hardware-software co-designed eCAD system for colorectal endoscopy with Support Vector Machine (SVM). The system is implemented the SVM on FPGA, and feature extraction module and feature transformation module are implemented on PC. As an estimation result, we achieved that the type identification module is about 2.7 faster than software implementation.

1. はじめに

近年, 国内外において大腸ガンの罹患数は年々増加傾向にある. しかし, 大腸ガンは早期ステージで発見, 治療ができれば完治はほぼ可能である. そのための大腸内視鏡検査方法として, NBI (Narrow Band Imaging) システムを用いた, 大腸拡大内視鏡による画像検査が行われている. 大腸 NBI 拡大内視鏡診断では, 医師が大腸内壁の血管等の微細模様構造から腫瘍の有無やガン深達度を診断する. この診断を行う医師には専門知識と経験が必要となり, 診断できる医師に限られる.

そこで, コンピュータ画像解析を用いた症状の客観的な判断による診断の正確度向上や, 定量的な指標による若手医師に対する教育支援を行うことが可能なコンピュータ診断支援 (Computer-Aided Diagnosis : CAD) システムが必要とされている. 我々研究グループでは, 広島大学病院の提

唱する NBI 拡大所見分類 [1]に基づき, 大腸内視鏡画像を図 1 に示すように 3 タイプ (Type A, Type B, Type C3) に分類する大腸 NBI 拡大内視鏡画像診断支援システム (**eCAD** システム) の実現を目指している. システムに求められる医療現場からの要求性能として, (I) 高速性と即応性 (スループット 1~5 fps かつ, レイテンシ 1 秒以内), (II) 高識別精度 (腫瘍部位か, あるいは非腫瘍部位であるかの診断結果が, 医師による診断結果と 90%以上一致すること [2]) が求められている.

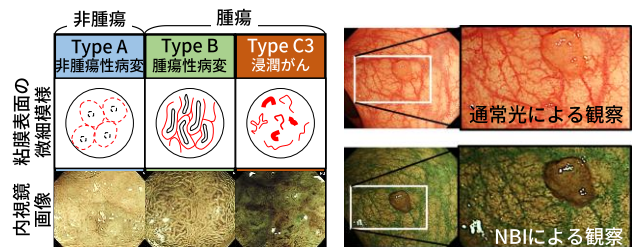


図 1. NBI 拡大所見分類 (広島大学病院提唱) [1].

†1 広島大学 ナノデバイス・バイオ融合科学研究所
Research Institute for Nanodevice and Bio Systems, Hiroshima University

†2 広島大学 工学研究院
Graduate School of Engineering, Hiroshima University

†3 JR 西日本 広島鉄道病院 消化器内科
Department of Gastroenterology Hiroshima General Hospital of West Japan Railway Company

†4 広島大学大学院 医歯薬保健学研究所 内視鏡医学
Department of Endoscopy and Medicine Graduate School of Biomedical and Health Science, Hiroshima University

2. 大腸 NBI 拡大内視鏡画像診断支援システム

eCAD システムの概要を図 2 に示す. オフラインで行う学習, 学習結果を基にオンラインで実行する識別の 2 つのフェーズが存在する. システムは(1)特徴抽出部, (2)特徴変換部, (3)タイプ識別部の 3 つのモジュールに大きく分けることができる. システムは Bag-of-Features (BoF) と呼ばれる手法に基づいて識別を行っている. BoF とは文書検索を画像に応用した手法で, 画像の局所特徴量によって得られる特徴量ベクトルを 1 つの単語と見做し, その出現頻度によって識別を行う.

学習フェーズでは, 学習用画像から特徴量を抽出し, 各タイプ $2^8=256$ 個のクラスタにクラスタリングを行い, 各クラスタの中心を Visual-Word (VW) として保存しておく. そして, 学習用画像から抽出した特徴量から VW の出現頻度である VW ヒストグラムを作成する. 作成した VW ヒストグラムをタイプ識別部の Support Vector Machine へと入力し, 識別フェーズで必要な Support Vector : SV を決定する. 識別フェーズでは入力された内視鏡映像のフレームから特徴量を抽出し, 得られた特徴量群を VW と照合して出現頻度によるヒストグラムを作成し, ヒストグラムデータに対してタイプ識別を行い, 結果を表示する.

研究グループによりシステムアルゴリズムの有用性に関する評価を行った結果, 腫瘍・非腫瘍の識別に有用性が確認できている[3]. 図 3 にその評価結果を示す. SVM の学習に Type A (非腫瘍) の画像 504 枚, Type B と C3 (腫瘍) の画像 1743 枚を使用し, 病理組織診断が得られた 118 症例の内視鏡観察時の画像をテストとして用いた. 識別結果は SVM の出力値 (SVM output value) によって得られ, 現在のシステムカットオフ値である 0.5 を境に腫瘍・非腫瘍の識別が可能であることが確認できている. また, システムをソフトウェア実装し動作させた場合, 大腸 NBI 拡大内視鏡画像中央の 120×120 pixel 領域に対し処理速度 14.7 fps を達成し, 正診率約 97% を達成している [4].

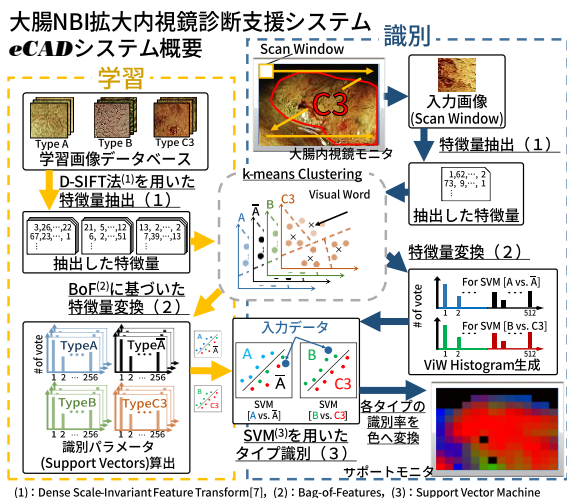


図 2. eCAD システム処理フロー概要.

2.1 複数病理タイプや複雑な形状が存在する 内視鏡画像の全画面識別に見られる課題点

システムの次段階として, 全画面識別への拡張によって画像内に複数タイプが混在する場合や, 画像中央領域外に腫瘍が存在する場合への対応が挙げられる. 内視鏡画像の全画面識別, そして医師への病理部位箇所の提示による診断支援の実現には, Scan Window (SW) と呼ばれる矩形領域による内視鏡画像のラスタスキャンが必要となる. この際, 患部の形状は SW のそれに沿ったものではないため, SW 内に病理タイプが複数混在してしまうことや, 複雑な形状を持つ病理タイプの領域の識別が難しいことがシステムの解決すべき課題として挙げられる.

このような課題解決のためには, システムの全画面識別において用いる SW サイズの大小に存在する, 識別精度と分解能のトレードオフの関係を考慮する必要がある. 小さい SW サイズによる全画面識別を行った場合, 高い分解能の識別結果を得ることができるが, SW 内の特徴点数が少ないことから, 全画面識別の識別精度は SW サイズが大きなものに比べて低くなってしまふ. SW サイズが大きい場合にはその逆の性質が見られ, SW 内の特徴点数が多いことから識別精度は高くなるが, 全画面識別結果の分解能は低くなってしまふ (図 4). 加えて, SW サイズの大小によって前段モジュールである特徴変換部での処理時間が異なることも分かっている. 処理時間の観点からは, 小さい SW サイズ (例えば 60×60 pixel) に比べて, 大きい SW サイズ (240×240 pixel) は処理時間が 20 倍程度かかってしまふ.

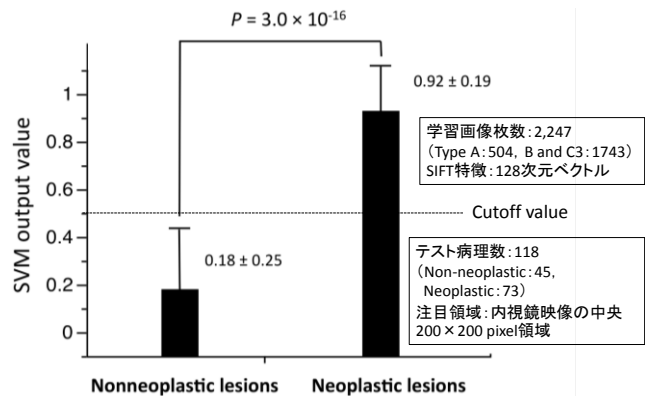


図 3. SVM を用いた大腸腫瘍・非腫瘍の識別有用性評価[3].

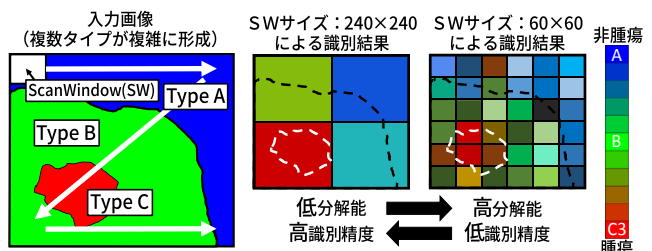


図 4. 全画面識別で用いる SW サイズに見られるトレードオフ.

2.2 階層的タイプ識別による内視鏡画像の全画面識別

このトレードオフ解消のため、単一の SW サイズを用いるのではなく複数の SW サイズを組合せて識別を行う階層的タイプ識別手法の基本コンセプトを提案している[5][6]。その概要を図5に示す。階層的タイプ識別手法では、Scan Grid という任意の SW サイズを基にラスタスキャンし特徴抽出を行う。次に、Scan Grid から得た特徴量を組み合わせて特徴変換することにより、より大きなサイズの SW の VW ヒストグラムを得る。それらを基にタイプ識別部では、SW サイズ毎の大きさに切り取った画像で学習を行い、学習結果を用いてタイプ識別を行う。これによって、上述の識別精度と分解能のトレードオフを解消することができる。図6、図7に実際の内視鏡画像に対して提案手法を適用した場合の全画面識別結果を示す。階層的タイプ識別手法を適用した出力画像の方が、単一階層で識別した結果よりも正確に病理部位を捉えられていることが確認できている。

ソフトウェア実装によるシステムを単純に Full HD (1920 × 1080 pixel) の内視鏡画像全画面識別に適用した場合、120 × 120 pixel の領域 (Scan Window : SW) を 10 pixel 間隔でスキャンすると、1 画像の処理時間は (1SW の処理時間) × (全画面識別に必要な SW 数) = (1/14.7 fps) × (17,557 個) ≒ 20 分となるため、ソフトウェア実装による全画面識別では医師からの要求性能である (I) 高速性と即応性が実現できない。

そこで本研究ではリアルタイムで医師の診断を支援するためシステムのハードウェア実装を目指す。しかしながら、先述の階層的タイプ識別手法における SW サイズや階層数、ラスタスキャンしていく間隔などに依存してハードウェア構成が変わってしまう。一方、タイプ識別部において、SW サイズの違いは学習によって得られるサポートベクトル情報の違いであり、メモリに格納するデータを変更することで異なる SW サイズによる識別を実現することができる。そこで本稿では、SW サイズやラスタスキャン間隔などを議論するため、タイプ識別部をハードウェア実装し、残りの処理をソフトウェアで実装するハードウェア・ソフトウェア協調設計によりシステムを構築する。

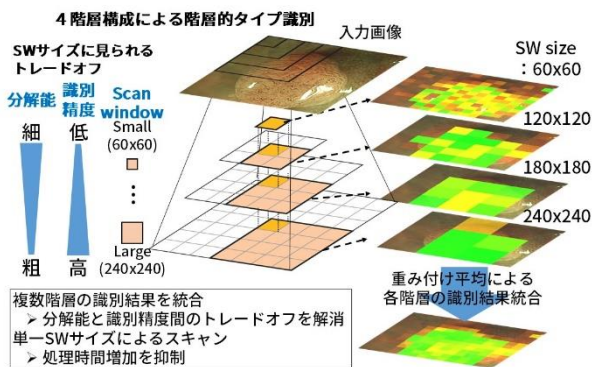


図5. 階層的タイプ識別手法コンセプト。

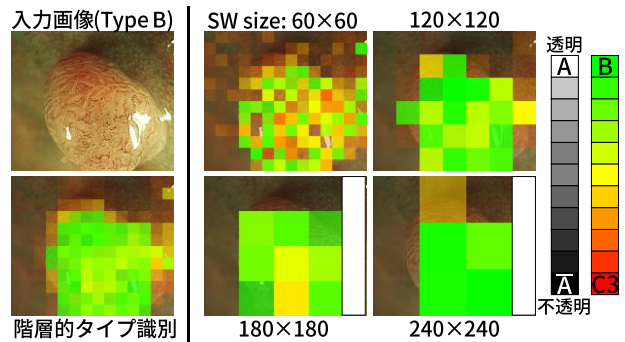


図6. 階層的タイプ識別手法適用例 (Type B)。

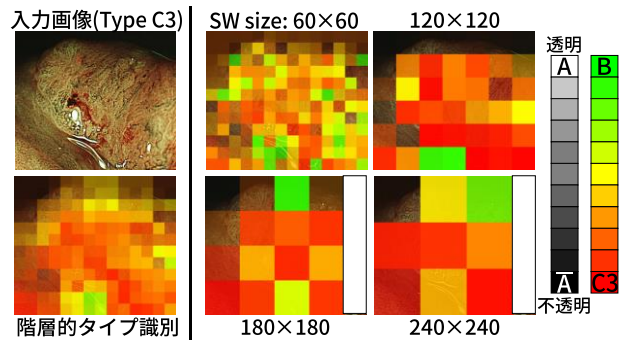


図7. 階層的タイプ識別手法適用例 (Type C3)。

3. SVM を用いた病理タイプ識別

3.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) は、2タイプ識別を行う分類アルゴリズムである[7]。SVM は教師あり学習により、識別空間において正と負の2タイプの距離 (マージン) が最大となるような識別超平面を生成する。入力データが識別超平面のどちら側に位置するかどうかを識別関数値の正負によって判断する。

本研究における SVM は2クラス識別器であるため、識別対象となる3クラス (3タイプ) の識別には複数の SVM が必要となる。本研究における SVM での識別関数を式(1)に示す。式(1)は入力データである特徴変換部の出力である VW ヒストグラムデータ \vec{x} がクラス Y とクラス Z のどちらに近いかに計算する (Y vs. Z) クラス識別器の例である。

$$d_{Y:Z}(\vec{x}) = \sum_{i=1}^{N_Y+N_Z} coef_i \times (\overline{sv}_i \cdot \vec{x}) + \rho_{Y:Z} \quad (1)$$

ここで、 \overline{sv}_i は学習時に求められるサポートベクトル (Support Vector : SV) であり、識別超平面を構成する。本研究において、 \overline{sv}_i 及び \vec{x} は 512 次元ベクトルである。また、 $coef_i$ は各 sv の係数、 $\rho_{Y:Z}$ は識別器の係数であり、学習時に求められる。また、 N_Y 、 N_Z はクラス Y とクラス Z の SV の数である。 $(\overline{sv}_i \cdot \vec{x})$ はカーネル関数と呼ばれ、入力空間を高次元特徴空間に写像することで線形分離不可能な問題にも適応が可能になる関数である (カーネルトリック) [7]。大腸 NBI 拡大内視鏡画像認識において、SVM のカー

ネル関数の違いによる認識率の差は2~3%であることが検証されている [4]. そこで、本研究ではハードウェア実装において計算コストの小さな、積和演算を行う Linear kernel (線形カーネル) を採用している. 2 クラス識別器である SVM を用いて3クラス (3 タイプ) 識別を行うため、本研究では2段階判定手法による3タイプ識別を提案している. 1段階目で Type A と Type B 及び C3 の識別を行う. 1段階目は言い換えれば、非腫瘍と腫瘍の識別に相当する. 次に2段階目で Type B と Type C3 の識別を行う.

3.2 臨床データからの内視鏡画像データベースの構築

システムの学習には、共同研究機関である広島大学病院から提供される臨床データである、大腸 NBI 拡大内視鏡画像を画像データベースとして用いている. 画像データベースは、図 8 に示すような順序を経て構築される. まず、専門医による大腸内視鏡による診察が行われ、患者の同意を得た上で、病理タイプに該当する箇所の画像が臨床データとして収集される. 次に、熟練した医師により、収集された臨床データから3タイプそれぞれの特徴が明確に確認できる部位を切り出す. このようにして切り出された画像を蓄積し、画像データベースとして構築する. 画像データベースはシステムにおいて学習、及びシステムのテストを行う際に用いられる.

5. ハードウェア・ソフトウェア協調によるシステム評価プラットフォーム構築

FPGA に実装したタイプ識別 (識別関数計算) アーキテクチャを組み込んだ **eCAD** テストシステムを、ハードウェア・ソフトウェア協調設計により評価プラットフォームとして構築する. システムの概要を図 8 に示す. 広島大学病院で撮影された内視鏡診断時の実際の動画を入力として、SDI 信号で外部から PC へ入力する. PC での入力信号の受け取りには Blackmagic Design の DeckLink Mini Recorder を使用する. FPGA 実装には GiDEL 社 Proce IV 530-A 開発ボードを使用した.

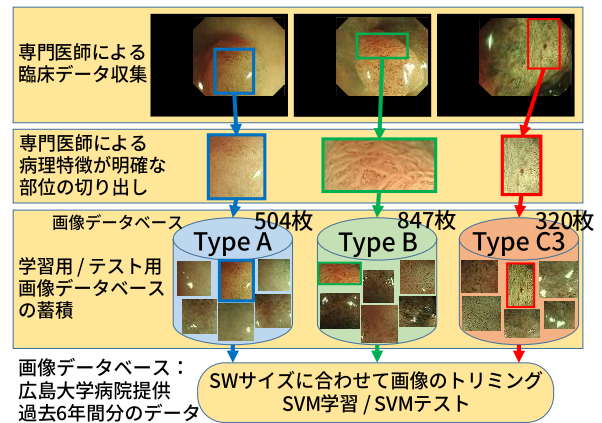


図 8. 内視鏡画像データベースの構築フロー.

4. SVM の識別関数計算アーキテクチャ

SVM によるタイプ識別では式 (1) に示した識別関数を計算し、その正負により識別を行う. その際に使用する学習時に得たサポートベクトルはそれぞれ、512 次元ベクトルで表現される. 識別関数計算アーキテクチャを図 9 に示す. サポートベクトルと、システムオンライン時に作成した VW ヒストグラム (512 次元) との積和演算は DSP ユニットにより並列に計算 (積和演算部)、係数 $coef_i$, $\rho_{A,Z}$ との積和演算を経て (係数計算部)、識別関数計算終了となる.

識別関数計算のタイミングチャートを図 10 に示す. 積和演算部が並列に実行され、サポートベクトル \vec{sv}_i と \vec{x} との積和演算が 256 クロックで完了する. 残り全てのサポートベクトルも同様にメモリから読み込まれ、積和演算が行われる. 同時に係数計算部では、積和演算部からの結果 $(\vec{sv}_i \cdot \vec{x})$ と係数 $coef_i$ の計算を順次行う. この演算は 256 クロック毎に到着する $(\vec{sv}_i \cdot \vec{x})$ の処理結果を 2 つずつ処理するため、計算に要するクロック数は $n/2$ となる. 最後に係数 ρ を加算して (1 clock)、識別関数計算は終了する. この一連の処理を連続的に処理し、システム全体でストリーム処理を行うことで、リアルタイム全画面識別を行う. 本稿において評価に用いた識別関数アーキテクチャ積和演算部の並列度 P は 64 に設定した.

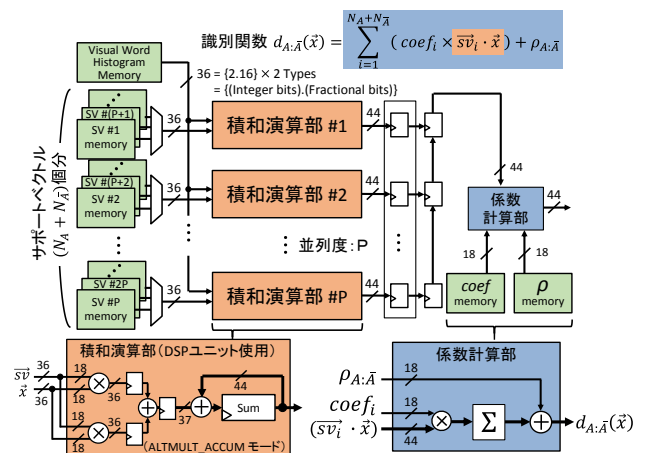


図 9. SVM 識別関数計算アーキテクチャ.

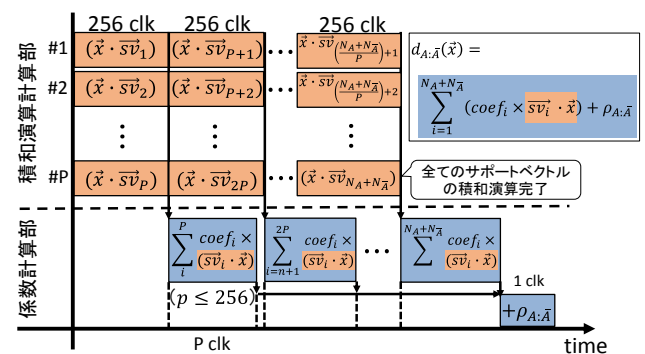


図 10. 識別関数計算タイミングチャート.

その後、映像の1フレームからSW領域の切り出しを行い、特徴量抽出へ入力し領域の特徴量を抽出する。特徴量データをVWヒストグラムへ変換し、それをFPGA上に実装したタイプ識別部へ入力することで識別処理を行う。PCとFPGAはPCI Express Gen3 x4で接続し、データ転送はDMA (Direct Memory Access) 転送で行う。ここで、タイプ識別部、評価プラットフォームインタフェース部を含めた全体での最適化は行っていない。PCとFPGA間のインタフェース開発には、GiDEL社が提供する統合開発環境ツールProcWizard v.9.4.2を用いた。FPGAからの結果を識別結果としてPCで受け取り、画面に出力する。1フレームの処理が終了すると、再度取り込んでいる映像から1フレーム画像を取り出し、同様の処理を繰り返し続ける。以上のようにして、FPGAに実装したタイプ識別部を使用したハードウェア・ソフトウェア協調設計によるテストシステムで評価を行う。

PC側の開発環境を表1に、インタフェース部を含んだFPGAリソース使用率を表2にそれぞれ示す。また、この評価プラットフォームを用いて図8におけるFPGAでのタイプ識別処理を、PC側で行うソフトウェア実装によるタイプ識別処理の時間と比較する。FPGAでの処理の時間計測方法は、PC側からFPGAへのDMAデータ転送の開始か

表1. PC開発環境.

OS	Windows7 Enterprise SP1 64bit
CPU	AMD FX-8120 8Core @3.10GHz
Memory	DDR3-1333 8GB x 4
使用言語	C++
開発環境	Microsoft VisualStudio 2012
使用ライブラリ	OpenCV 3.0

表2. FPGAリソース使用率.

Altera Stratix VI Available (EP4SE530)		積和演算部 並列度	
		64	
# ALUTs	424,960	9,114	2%
# registers	424,960	11,031	3%
# Logic Array Blocks	21,248	1047	5%
# M9K RAM Blocks	1,280	200	16%
# 18 x 18 DSPs	1,024	260	25%
Max Frequency		110.2 MHz	

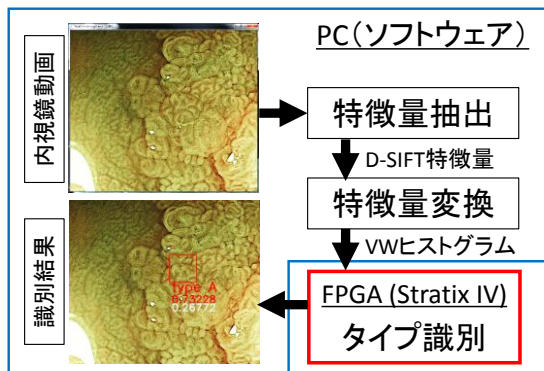


図11. ハードウェア・ソフトウェア協調設計による評価プラットフォーム.

ら、PC側でFPGAの処理結果を受け取り終わるまでの時間をPC側で計測した。

6. 評価結果

5章で述べた評価プラットフォームを用いて、SWサイズを60×60, 120×120, 180×180, 240×240 pixelとした場合のFPGAで処理した場合とソフトウェアで処理した場合との処理時間比較結果を表3に示す。処理時間は10,000個のSWを処理した平均値である。SWあたりの処理時間はSWサイズにほぼ影響されず、ソフトウェアの場合約0.16 msec、ハードウェアの場合約0.06 msecとなった。

表3. SWあたりの処理時間比較.

SVM実装	SW60	SW120	SW180	SW240
Hardware	0.06 msec	0.06 msec	0.06 msec	0.06 msec
Software	0.16 msec	0.15 msec	0.16 msec	0.16 msec

次に、Full-HD 画像を全画面識別した際の処理時間の比較を行う。この場合の処理時間は、Full-HD 画像内のSWの切り取りから測定を開始し、特徴抽出、特徴変換処理を経て、タイプ識別を実行し、結果が得られるまでを測定している。スキャン間隔として、30, 60 pixelの2通りの場合を評価した。

図12, 図13にスキャン間隔が30, 60 pixelだった場合の、ハードウェア実装、ソフトウェア実装の処理時間比較結果を示す。結果として、ハードウェア実装の場合と、ソフトウェア実装の場合とで大きな差は確認できなかった。これは、タイプ識別部以外の処理時間がタイプ識別部のそれと比べて支配的であるため、ハードウェア実装とソフトウェア実装とで差が顕著になって確認できなかったことが原因として挙げられる。図14にSWサイズを60×60から300×300まで、20 pixel毎に処理時間を評価した結果を示す。この場合も、ソフトウェア実装とハードウェア実装で大きな変化は見られず、処理時間が一様に増加した。

評価結果より、SVMのみをハードウェア実装しただけでは、ひとつのSWサイズでの全画面識別の処理時間はソフトウェア実装の場合と比べて大きな変化は見られなかった。しかしながら、我々が開発を目指している全画面識別による診断支援システムでは、2章で述べた階層的タイプ識別手法の採用を想定している。階層的タイプ識別手法では複数のSWサイズを並列に実行し、各SWサイズによる識別結果を統合させる。このことから、逐次処理によるソフトウェア実装の場合、構成した階層分だけ処理時間が単純増加してしまう。一方でハードウェア実装の場合、表2に示したアーキテクチャを用いると、4階層分を並列に処理することが可能であることがわかる。また、システム全体のハードウェア実装に向け、ハードウェア向けアルゴリズム

を特徴抽出部および特徴変換部で提案しており[8][9], 特に特徴変換部では, ソフトウェアアルゴリズムを用いると 2 分木探索を階層数分だけ実行する必要がある. 我々研究グループでは, これに対して VW ヒストグラムを小さな SW サイズで作成したヒストグラムを足し合わせるにより構築する手法を提案している.

本稿で提案したハードウェア・ソフトウェア協調設計による評価プラットフォーム上での診断支援システムの構築, およびシステム全体として SW サイズやスキャンステップ間隔などのさらなる検討のため, ハードウェア向けアルゴリズムを実装した特徴抽出部および特徴変換部でのソフトウェア部を構築し, 評価を続けて行っていく必要がある.

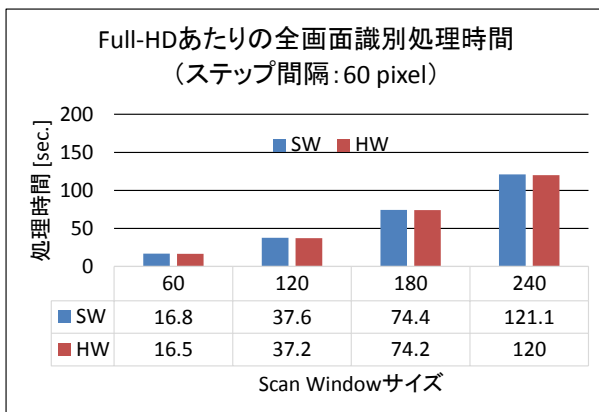


図 12. Full-HD 画像を入力とした時の全画面識別処理時間比較 (ステップ間隔 60 px).

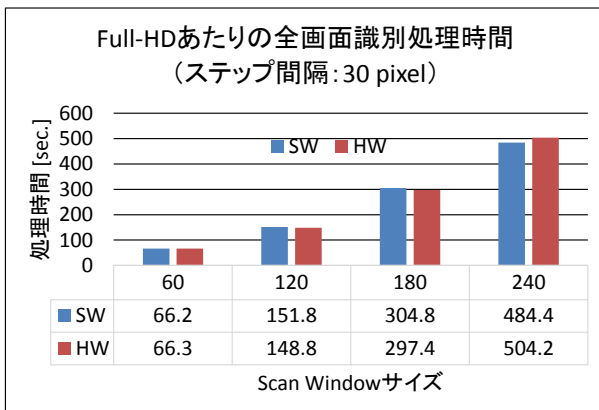


図 13. Full-HD 画像を入力とした時の全画面識別処理時間比較 (ステップ間隔 30 px).

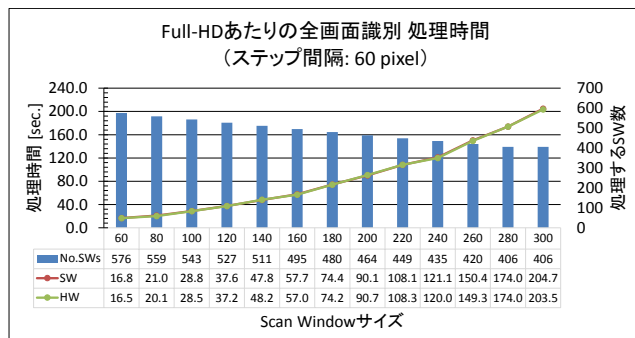


図 14. SW サイズが異なる場合の全画面識別の処理時間比較 (ステップ間隔 60 px).

7. まとめ

本稿では, **eCAD** システムの処理モジュールの一つであるタイプ識別部に関して, ハードウェア・ソフトウェア協調設計による評価プラットフォームを提案し, ソフトウェア実装とハードウェア実装との場合について, 処理時間の比較を行った. タイプ識別部の DSP ユニット数を 64 とした場合, SW ひとつを処理する時間はハードウェア実装の場合約 0.06 msec と, ソフトウェア実装と比較して約 2.7 倍の高速化が確認できた. Full-HD 画像の全画面識別のため, SW を 30, 60 pixel 間隔でスキャンして評価を行った結果, 特徴抽出部および特徴変換部の処理時間が支配的であることから, ハードウェアとソフトウェアとの間に大きな差は確認できなかつた. 今後の課題として, さらなるシステム改良に向け, 現在研究グループで提案している, ハードウェア向けに改良した特徴抽出および特徴変換をソフトウェア状に実装し, システムに適した SW サイズ階層数やステップ間隔などを検討していく必要がある.

謝辞

本研究の一部は, 基盤研究(B)26280015 および特別研究員奨励費(16J06130)の助成を受けたものです. FPGA 開発ツールは Altera University Program の支援を受けたものです.

参考文献

- 1) H. Kanao, et al., "Narrow-band imaging magnification predicts the histology and invasion depth of colorectal tumors," Journal of Gastrointestinal Endoscopy, vol. 69, no.3, pp. 631-636, 2009.
- 2) D. K. Rex, et al., "The American Society for Gastro- intestinal Endoscopy PIVI (Preservation and Incorporation of Valuable Endoscopic Innovations) on real-time endoscopic assessment of the histology of diminutive colorectal polyps," Gastrointestinal Endoscopy, Vol.73, Issue 3, pp.419-422 March 2011.
- 3) Y. Kominami, et al., "Computer-aided diagnosis of colorectal polyp histology by using a real-time image recognition system and narrow-band imaging magnifying colonoscopy," Gastrointestinal Endoscopy, Vol. 83, Issue 3, pp. 643-649, 2016.
- 4) T. Tamaki, et al., "Computer-aided colorectal tumor classification in NBI endoscopy using local features", medical image analysis, Vol. 17, No. 1, pp. 78-100, 2013.
- 5) T. Okamoto, et al., "Effective Diagnostic Image Segmentation with Pyramid Style Support Vector Machine for Colorectal Endoscopic Images," Proc. of the 30th International Conference on Circuits / Systems, Computers and Communications, 2015.
- 6) T. Okamoto, et al., "A Hierarchical Type Segmentation Algorithm Based on Support Vector Machine for Colorectal Endoscopic Images with NBI Magnification," Proc. of the 19th Workshop on Synthesis And System Integration of Mixed Info. Technologies, pp. 374-379, 2015.
- 7) V. Vapnik, "Statistical learning theory," John Wiley & Sons Inc., New York, 1998.
- 8) T. Koide, et al., "Compact and High-Speed Hardware Feature Extraction Accelerator for Dense Scale-Invariant Feature Transform", Proc. of the International Technical Conference on Circuits/Systems, Computers and Communications, pp. 387-390, July 2016.
- 9) T. Koide, et al., "A hardware accelerator for bag-of-features based visual word transformation in computer aided diagnosis for colorectal endoscopic images", Proc. of the 31th International Technical Conference on Circuits/Systems, Computers and Communications 2016, pp. 233-236, July 2016