

山崎研究室紹介

2017年10月24日

山崎 勝弘

yamazaki@se.ritsumeai.ac.jp

1. 研究室の目標
 2. 育成したい人材
 3. 指導方針
 4. 研究分野： 並列処理と深層学習
 5. 研究テーマ
 6. 学生交流
 7. 貴君らに提供できること
 8. おわりに
- 付録

1. 研究室の目標

- **社会人としての基本的素養**をつけ、努力を継続して、目標を達成できる人材の育成
- コミュニケーション能力、スケジューリング能力、および知的体力の養成
- **並列処理**と**深層学習**を融合したAI指向の問題解決システムの構築
- ハードウェアとソフトウェアが両方分かる人材の育成

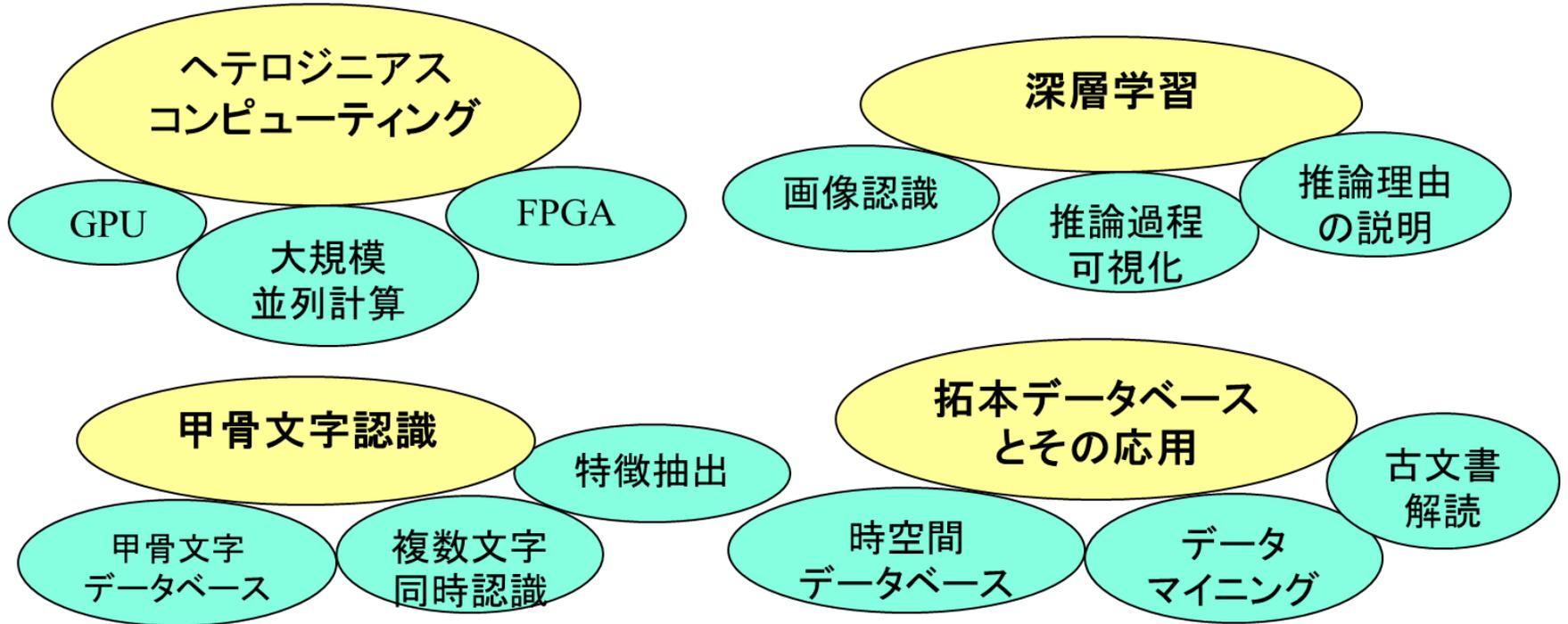
2. 育成したい人材像

- あいさつ
- コミュニケーション能力
 - 日本語で正しく表現、発表、議論
 - 英語能力
- スケジューリング能力
 - 立案、実行、チェック、改善 (Plan, do ,check, act)
- 知的体力
 - 最後まであきらめずにやり遂げること

3. 指導方針

- 明るく前向きに： positive thinking
- 研究テーマの設定、研究環境の整備
- 社会人としての基本的素養をつけて欲しい。
- 社会人になるための実力をつけて欲しい。
- 英会話学習のきっかけをつかんで欲しい。
- 自分の夢を将来にわたって実現して欲しい。

4. 研究分野



FPGA、GPU、CPUを用いた高性能システムの構築

深層学習の各種応用

研究経歴(1976～現在)

FPGA

ハードソフト

アーキテクチャ

並列アプリ

知識工学

プロセッサ4台の並列マシン
マイクロプログラム制御
ハードウェア、システムソフト
アプリケーション

QA-1
MUNAP

電力系統
知的CAI

1988 情報工学科

レイトレーシング
ラジオシティ

事例ベース
並列プログラミング

トランスピュータ
AP1000+

1994 情報学科

KSR1

同期マルチ
メディア

ハード/ソフト・カラー
ニングシステム

ルーティング

JPEG、MPEG
N体、グリッド

2004 電子情報デザイン学科

ハッシュ関
数高速化

OpenMP
動作合成

PCクラスタ
SMPクラスタ

MAP設計

SMPクラスタ

リアルタイム
レイトレーシング

FPGA

TRAX

GPU

2012 電子情報工学科

BlokusDuo

BLOBと前方
車両検出

甲骨文字
認識

拓本の多
書体認識

深層学習

並列処理の目標

- マルチコアプロセッサ
- GPU(Graphics Processing Unit)
- 超並列マルチスレッド処理
- ハイブリッド並列処理
 - 共有メモリ+分散メモリ、OpenMP+MPI
- ヘテロジェニアスコンピューティング

目標

- 深層学習を用いた甲骨文字認識など、大規模問題をGPU上で高速に解く。
- 医療診断、気象予測、地震動予測

並列処理から **超並列処理** の時代へ

名前	コア数	価格(万円)		分類
• MUNAP	4	2000	1982	研究用 自作
• KSR-1	64	Kendahl Square Research		1億円以上
• AP-1000+	64	富士通		1億円以上
• Transputer	64	800	1992	神戸製鋼から購入
• X	16	320	2000	PCクラスタ
• Raptor	16	200	2003	SMPクラスタ
• Diplo	16	184	2006	SMPクラスタ
• Nycto	16	174	2009	SMPクラスタ
• Kepler	2500	68	2013	GPUマシン
• DL-Box	3072	70	2015	深層学習用 TitanX
• DL-Box2	3072 × 2	93	2016	深層学習用 TitanX
• DL-Box3	3584 × 4	99	2017	GTX1080i

ハードとソフトができるメリット

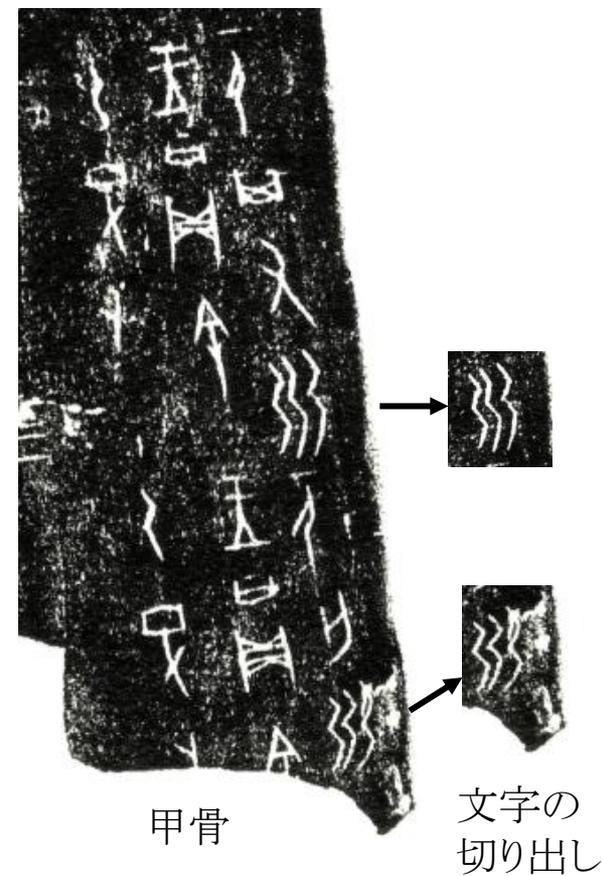
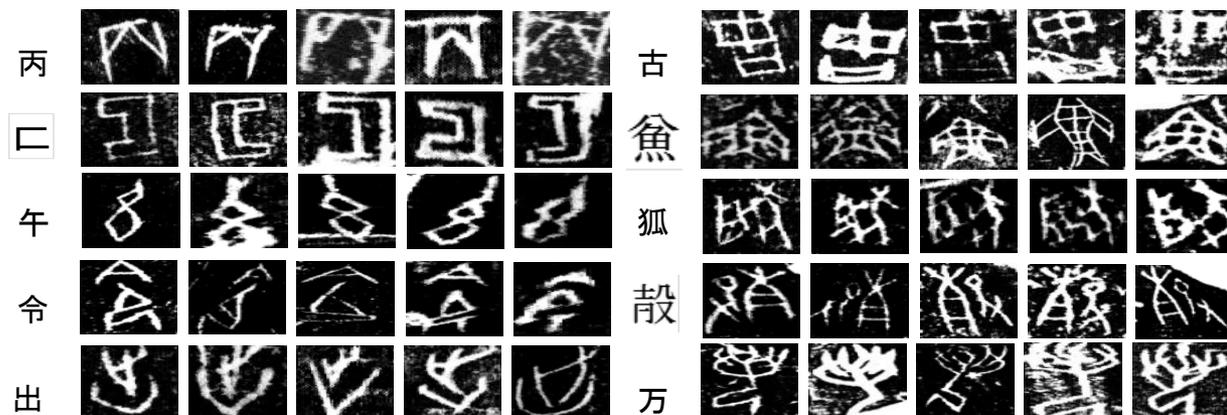
- 人間： 強い肉体(体力) + 知恵(知力)
- コンピュータ： 高い性能(ハードウェア) + 使いやすさ(ソフトウェア)
- ハードウェア： 高い性能 設計とデバッグが難しい。
 - 差別化が難しい。低価格化にさらされる。
- ソフトウェア： 付加価値 設計の自由度が高い。
 - 技術者が多い。Cプログラミングは理工系なら誰でもできる。
- ハードウェアとソフトウェア： できる人材が極めて少ない。
 - ニーズは高い。両方できれば、鬼に金棒
- Cプログラミング：アセンブリ言語やコンパイラのコード生成が分かれば、コンパクトで速いプログラムができる
- プロセッサ設計：命令の使われ方や使用頻度が分かれば、良い命令セットを設計できる。

5. 研究テーマ

- 5. 1 GPU上での深層学習を用いた甲骨文字の自動認識
- 5. 2 深層学習の推論過程の可視化と説明機能の実現
- 5. 3 深層学習を用いた拓本の多書体認識と拓本データベースに基づく古代文献解析
- 5. 4 GPUとFPGAを用いた大規模並列計算

甲骨文字認識の研究

- 研究背景
 - 3000年以上前、劣化、認識困難
 - 文字の起源・変化と未解読資料の解読に重要
 - 中国の古代史学、古典文学の研究
- 研究手法
 - 画像処理: ノイズ除去、細線化、ハフ変換
 - テンプレートマッチング 類似度計算
 - 複数文字の同時認識: GPU並列処理
 - 深層学習: 大量学習 自動認識



5.1 深層学習を用いた甲骨文字の自動認識

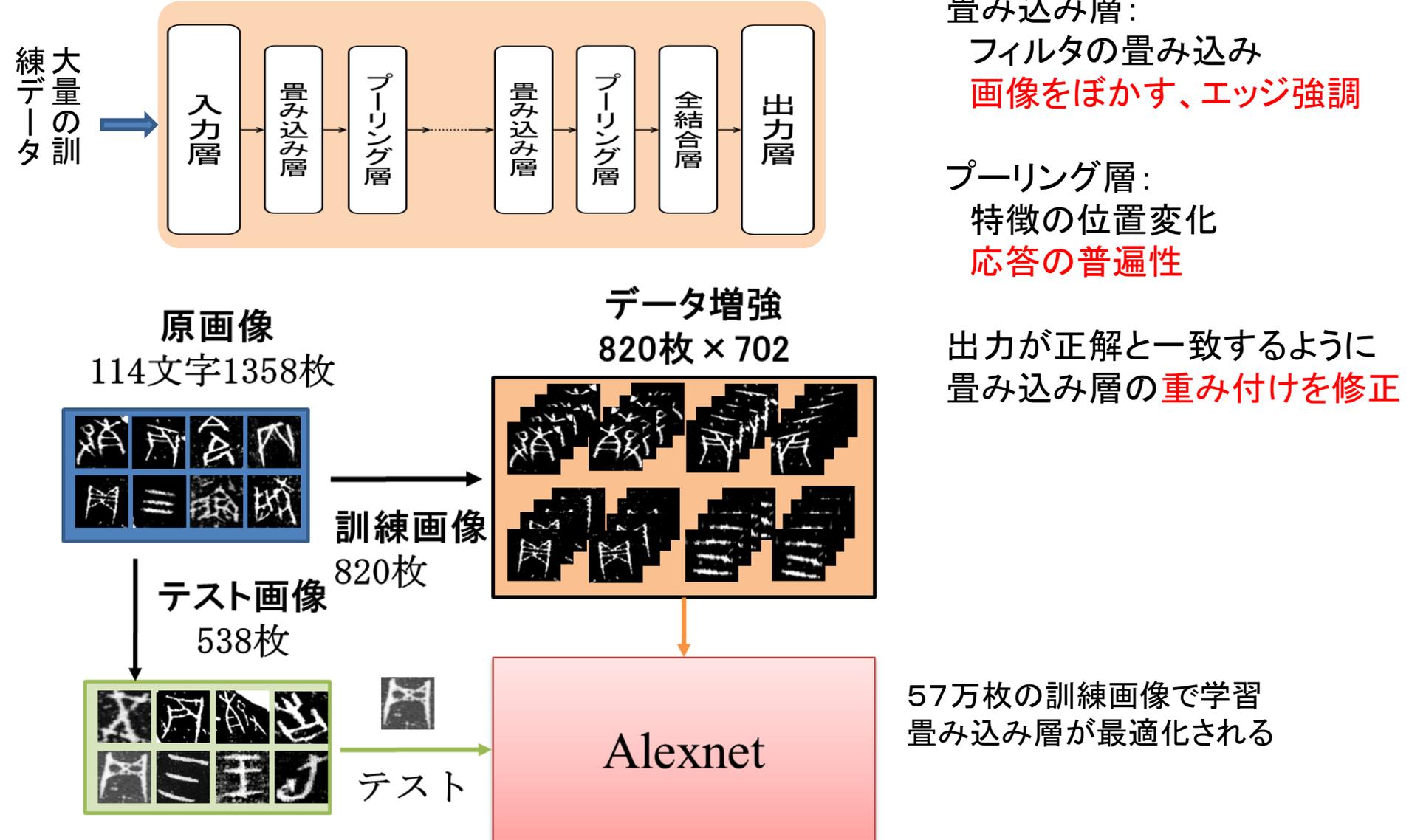
CNN: 畳み込みニューラルネットワーク

畳み込み層:
フィルタの畳み込み
画像をぼかす、エッジ強調

プーリング層:
特徴の位置変化
応答の普遍性

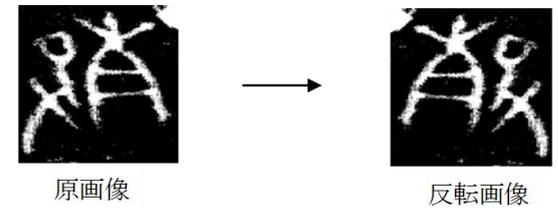
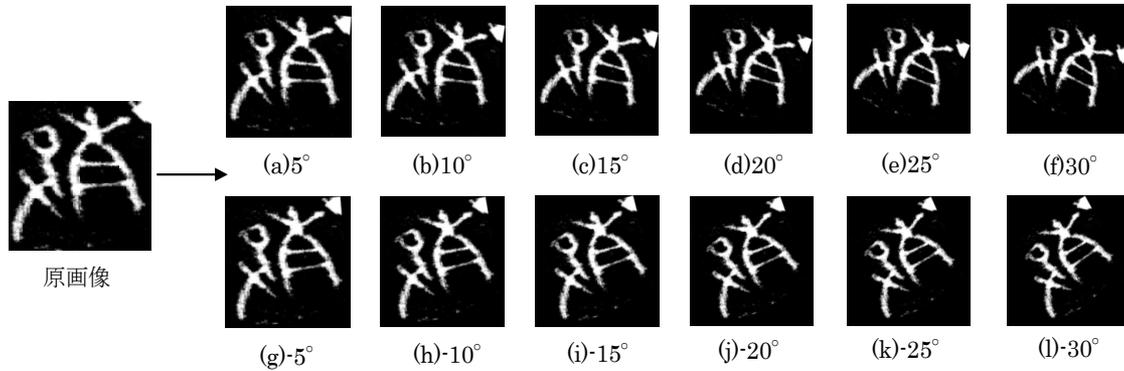
出力が正解と一致するように
畳み込み層の重み付けを修正

57万枚の訓練画像で学習
畳み込み層が最適化される



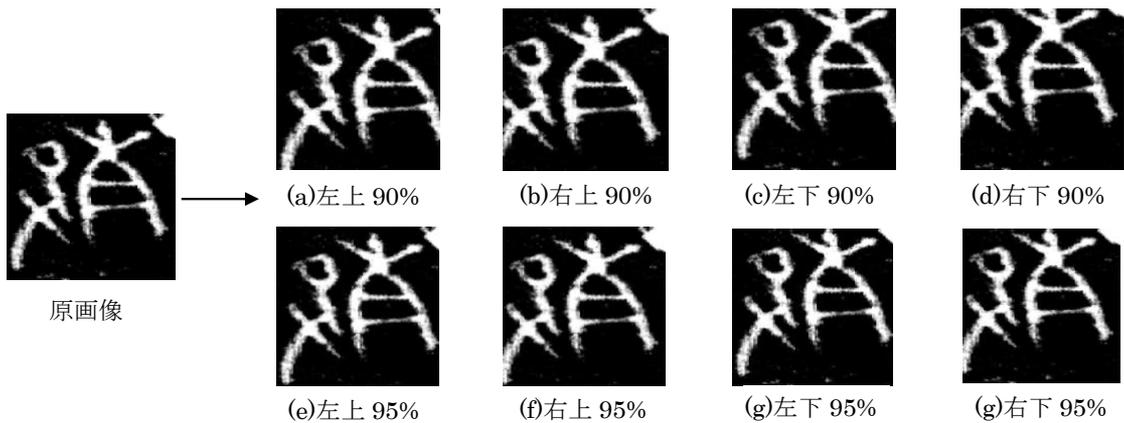
データ増強

- 一回転・切り取り・反転・輝度値 訓練画像を増やす 702倍
- $820 \times 13 \times 9 \times 2 \times 3 = 575640$ 枚の画像生成



回転: $-30^\circ \sim +30^\circ$ 5° 刻み 13通り

反転 2通り



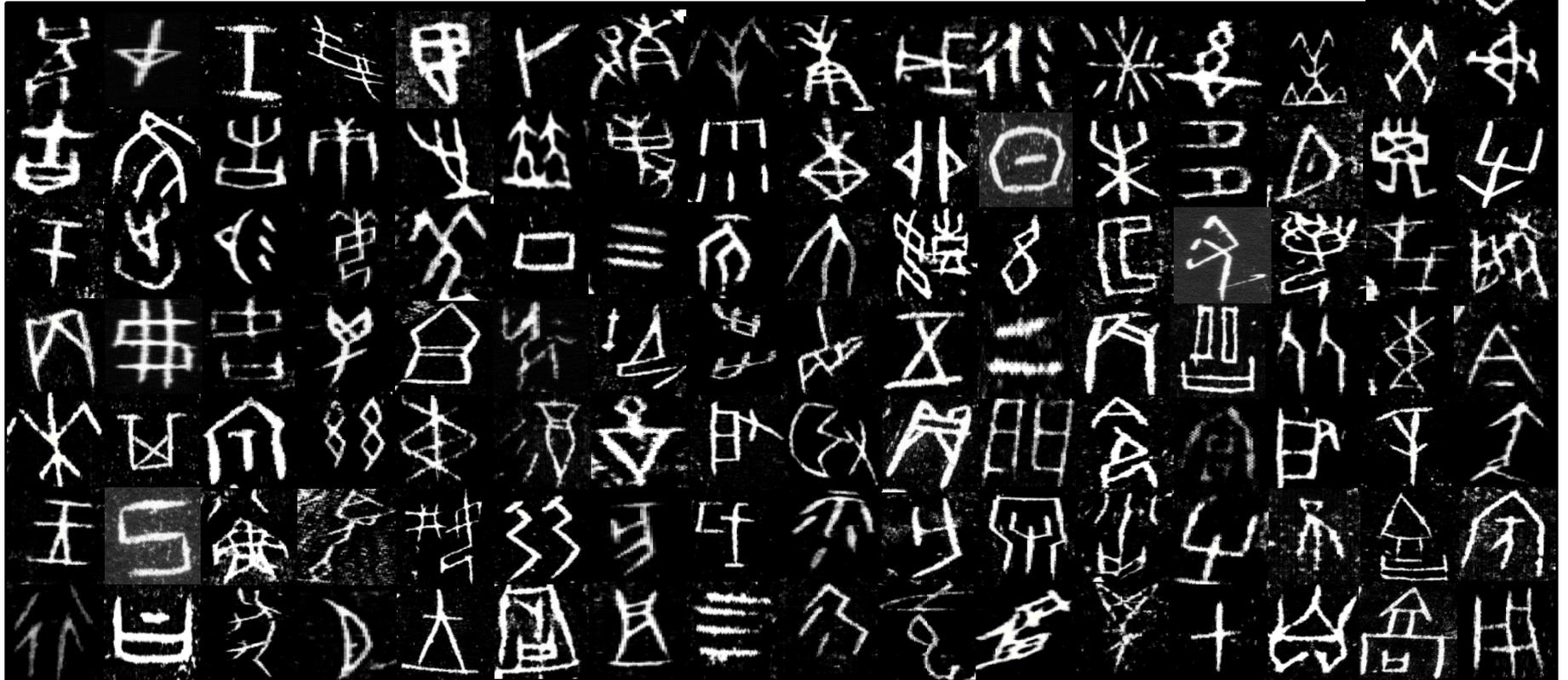
輝度値 3通り

切り取り: 四隅を95%と90% 9通り

実験内容

- ・文字114種類 訓練画像57万枚で学習する
- ・テスト画像538枚で認識実験を行い、認識率を調べる
- ・認識率を向上させるために、データ増強やパラメータを最適化する

対象文字 114



実験結果

・実験内容

文字種類	原画像	訓練画像	テスト画像
114	820	575640	538

・認識率

485枚認識 90% 学習時間 2h37m

切り取り 85%と80%を追加 2%向上
ドロッアウト変更 2%向上

・実験環境

GPUマシン UNIV GU-1100
GeForce TITAN X 3072コア×2
Xeon E5-1620, 64GBメモリ

認識できた画像例



(1) (2) (3) (4)



(5) (6) (7) (8)

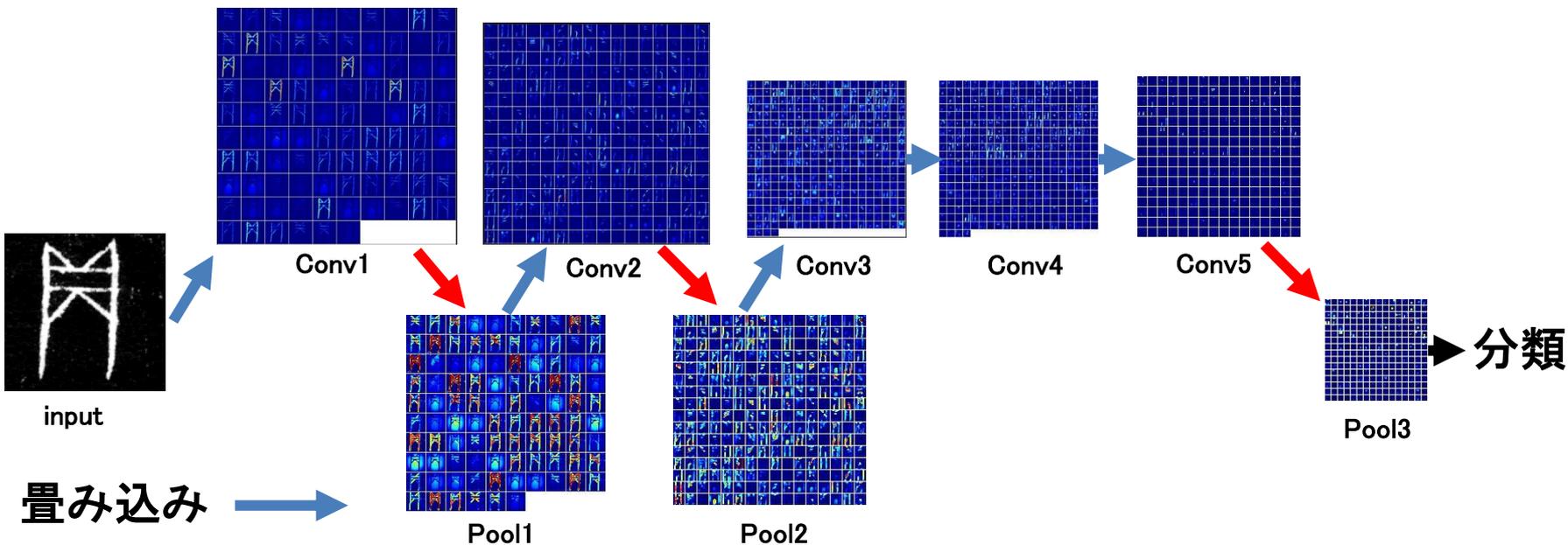
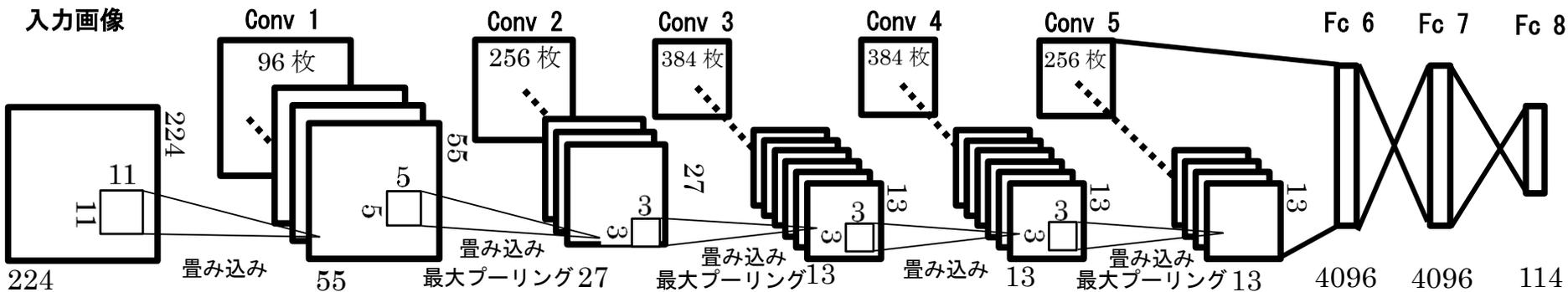


(9) (10) (11) (12)

5.2 深層学習の推論過程の可視化と説明機能の実現

特徴抽出

分類

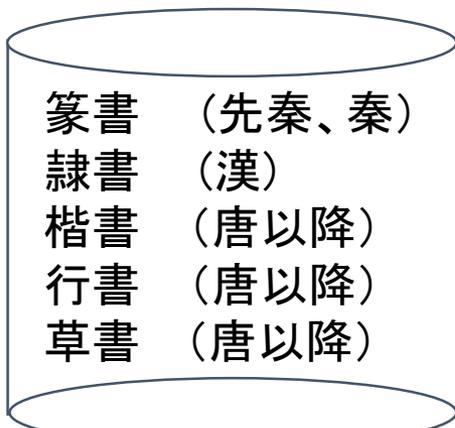


Alexnet

5.3 拓本の多書体認識と古代文献分析

- ・ 深層学習を用いて拓本内の多書体文字を認識
- ・ 時空間データベースの作成、情報の解析

1. 拓本データベースの作成



拓本データベース

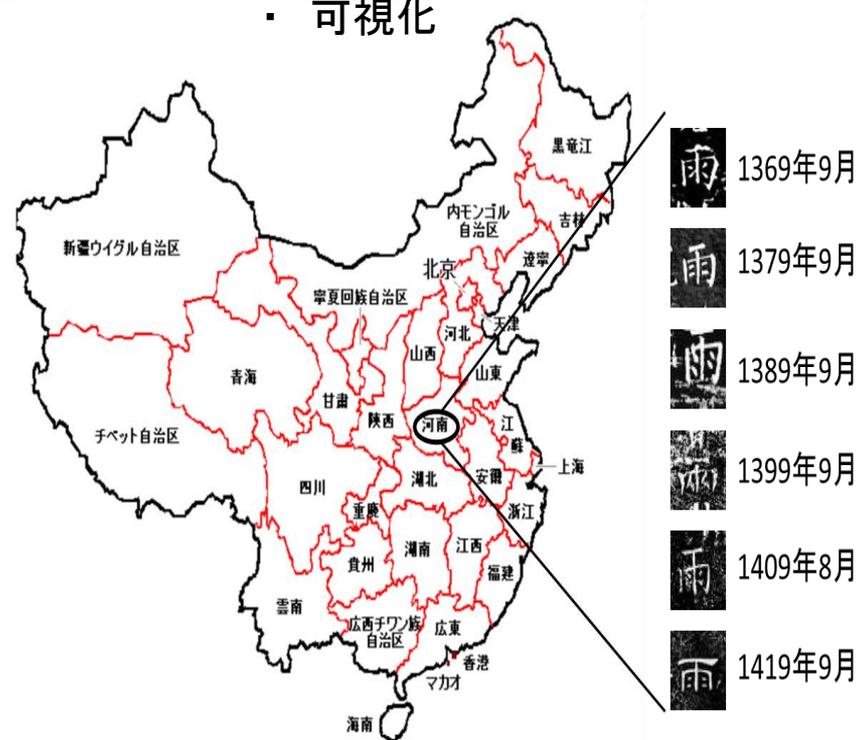
2. 拓本文字認識、データベースの更新

- ・ 文字認識
- ・ データベース更新



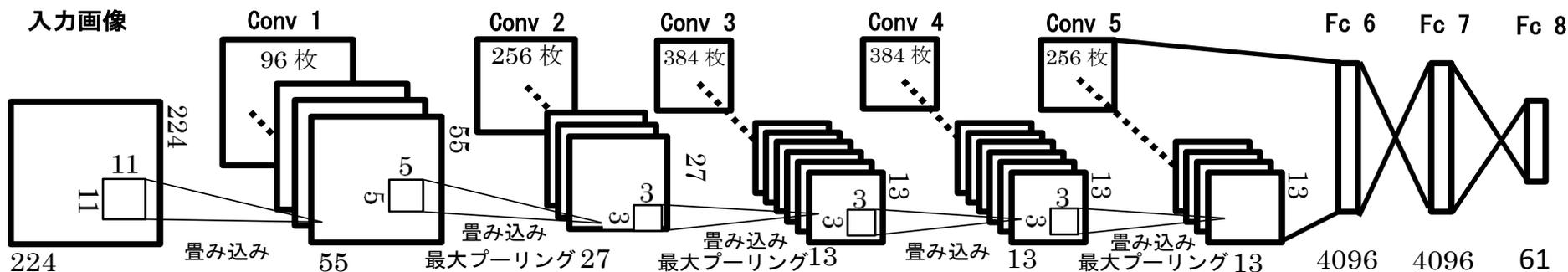
3. 解析処理と

- ・ 時空間情報の追加
- ・ 可視化



深層学習を用いた拓本の多書体認識

- 畳み込みニューラルネットワーク(CNN)を使用
 - AlexnetとGoogLenet
- 61種計79054枚の画像をデータ増強して学習
- 61種各20枚(5書体含む)の画像でテスト



Alexnet

Alexnet :8層

GoogLenet :38層

訓練用データセット

- 収集した画像を教師画像と訓練画像に分ける
- 教師画像には、5書体が必ず含まれるようにする



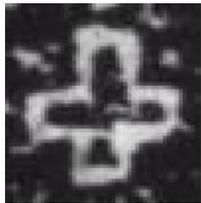
実験結果

- 各文字20枚計1220枚のテスト画像

	学習時間	正解数	認識率
Alexnet	9時間34分	1147	94%
GoogLenet	1日4時間	1172	96%

Deep Learning BOX
GeForceGTX1080Ti
3584コア×4
Xeon E5-1620 4コア
64GBメモリ

認識できた画像例



亜



青



池



巖



員



牛



歌



鬱

誤認識した画像例



入力(印)



出力(卯)



入力(雲)



出力(色)



入力(牛)



出力(午)



入力(馬)



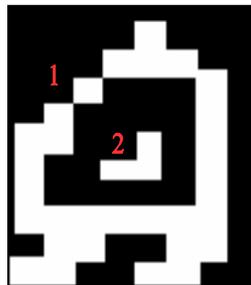
出力(馬)

FPGA上でのマルチパイプラインを用いたBLOB検出の高速化

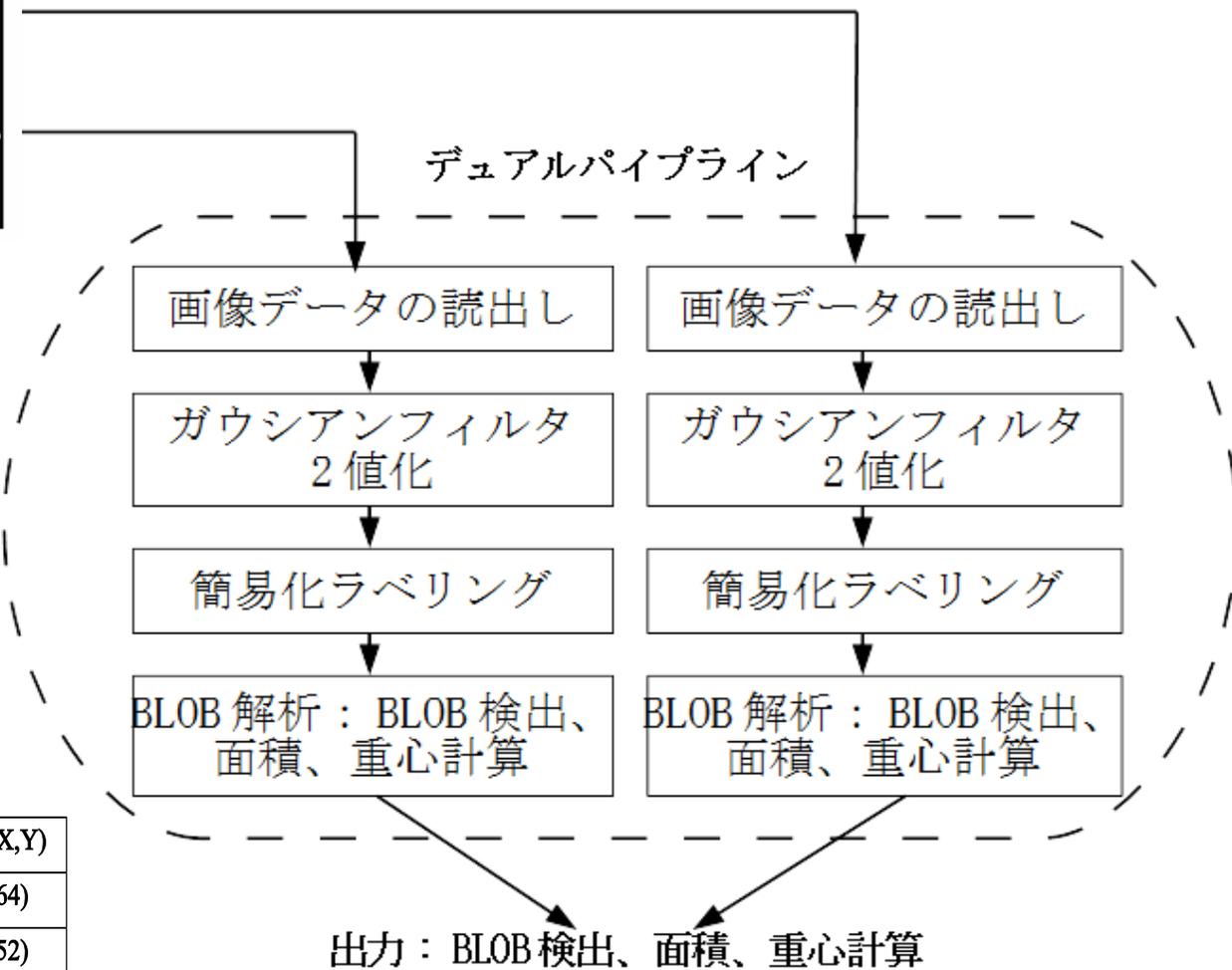
入力画像



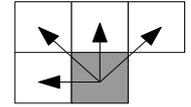
BLOB	面積	重心(X,Y)
1	320	(30,26)
2	630	(42,52)
3	320	(84,66)
4	320	(52,88)



BLOB	面積	重心(X,Y)
1	3643	(58,64)
2	275	(48,52)

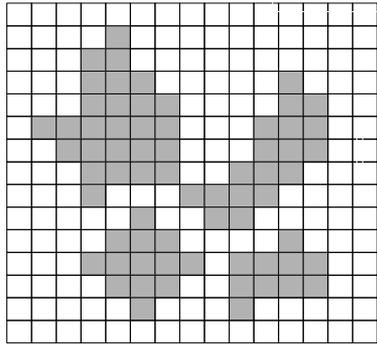


簡略化ラベリング

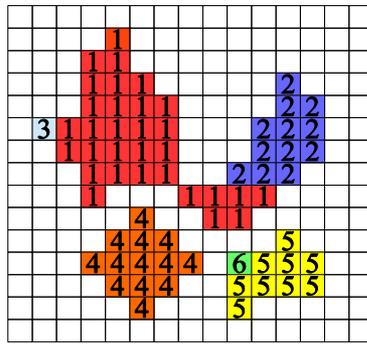


仮ラベル生成マスクパターン

一般的なラベリング



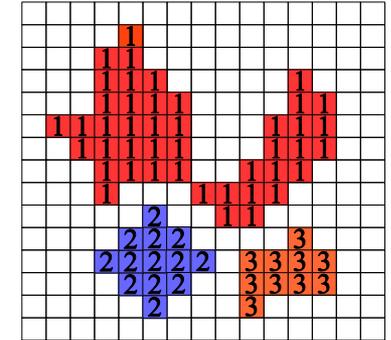
2値化画像



仮ラベル生成

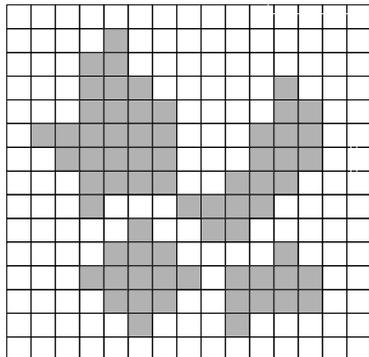
仮ラベル	仮ラベル 連結成分
1	3
1	2
5	6

LUT

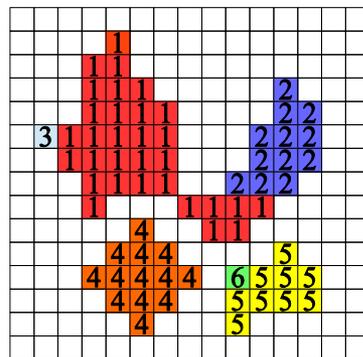


ラベル補正

簡略化ラベリング(並列化に応用)



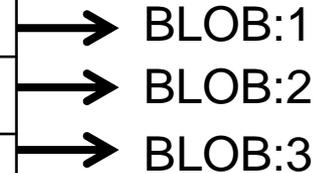
2値化画像



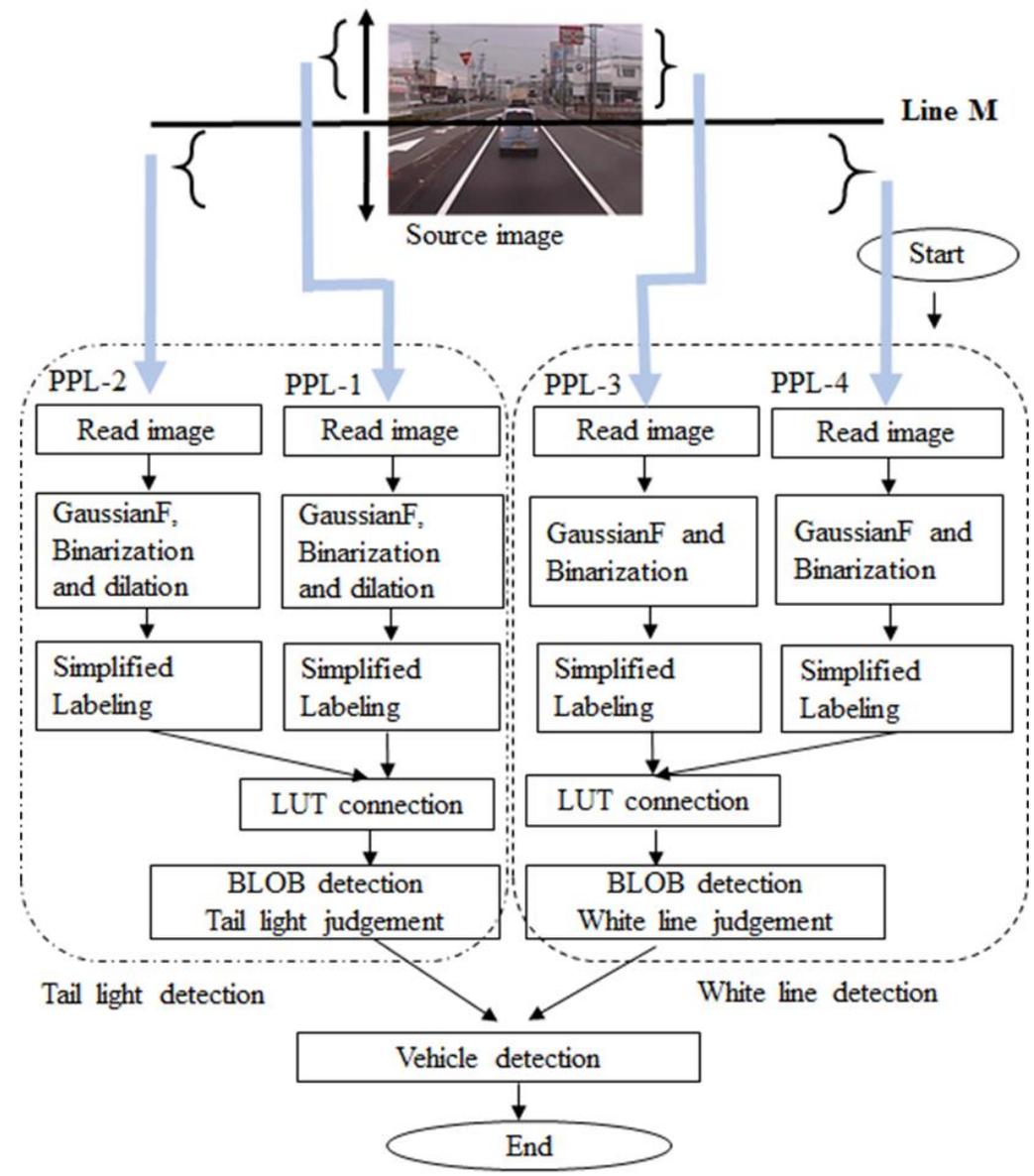
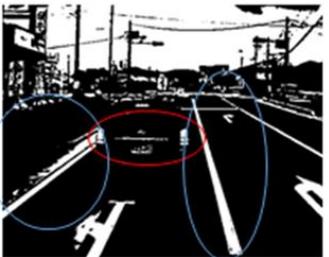
仮ラベル生成

BLOB番号	仮ラベル 連結成分
1	1, 2, 3
2	4
3	5, 6

BLOB



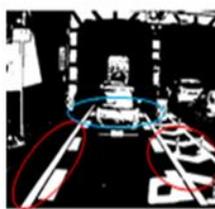
マルチパイプライン前方車両検出



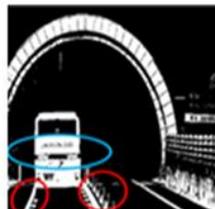
Original images

Detection results

(a) Tunnel 1



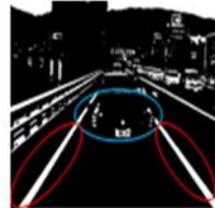
(b) Tunnel 2



(c) Long distance



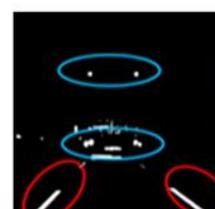
(d) Short distance



(e) Bus



(f) Truck



検出結果と実行時間

BLOB検出の実行時間

100 × 100画像

Virtex5

	クアドパイプライン	デュアルパイプライン	シングルパイプライン
動作周波数(MHz)	92.05	99.18	
実行時間/ピクセル(ns)	10.9	10.08	
全実行時間(μs)	31.6	57.7	112.13
速度向上率(倍)	3.55	1.94	1

前方車両検出の実行時間

128 × 128画像

Artix7

	クアド	デュアル
動作周波数 (MHz)	73.5	76.1
実行時間ピクセル (ns)	13.6	13.1
全実行時間 (μs)	124.8	238.6
速度向上	1.91	1

6. 学生交流

- **宇都宮大学** 馬場・横田・大津・大川研究室
 - 2012年11月29日～30日 7名参加
- **広島市立大学** 北村・川端・窪田研究室
 - 2013年11月14日 7名参加
- **熊本大学** 末吉・久我・飯田研究室
 - 2014年12月4日 8名参加
- **東北大学** 佐野研究室
 - 2016年3月9日 8名参加
- **北陸先端大学院大学** 井口研究室
 - 2016年9月6日 12名参加
- **電気通信大学** 吉永研究室
 - 2017年9月16日 8名参加

7. 貴君らに提供できること

- **問題解決**の仕方

- 卒論、進路、就職、…

- **スケジューリング**の仕方

- Plan, do, check, act

- **研究発表**の仕方

- 日本語文章、スライド作成、発表、…

- **英会話勉強**の仕方

- マンチェスター大学 客員研究員 1992年～1993年

- IEEE student branch カウンセラー 2006年から5年間

- 英語プレゼン大会 2004年から14回実施

MUNAP

トランスピュータ

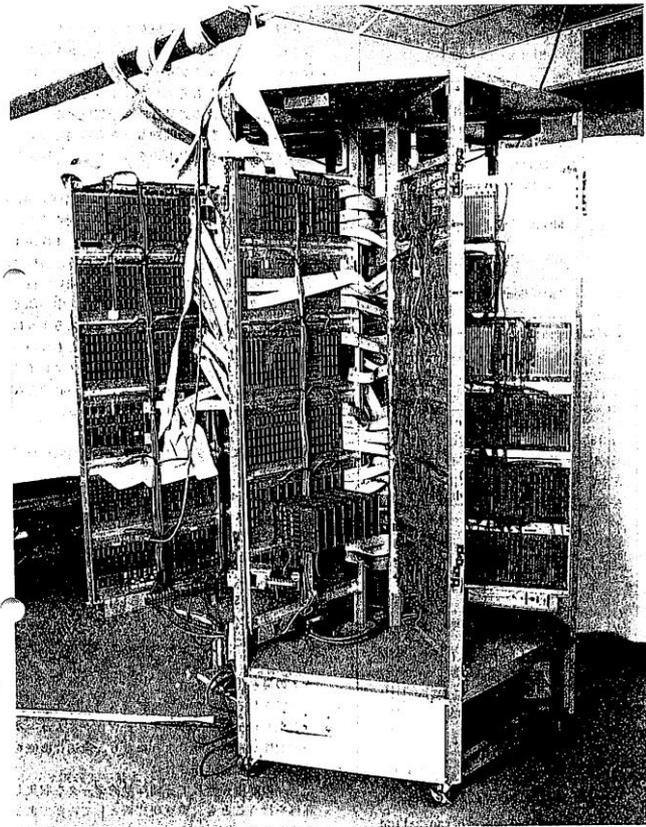


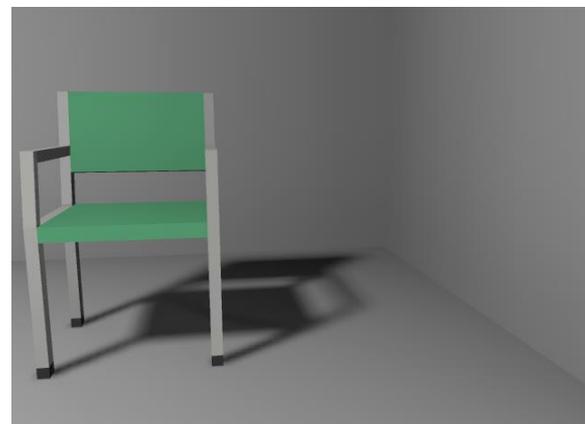
図2 MUNAP概観
Fig. 2 General View of MUNAP

= 6 =

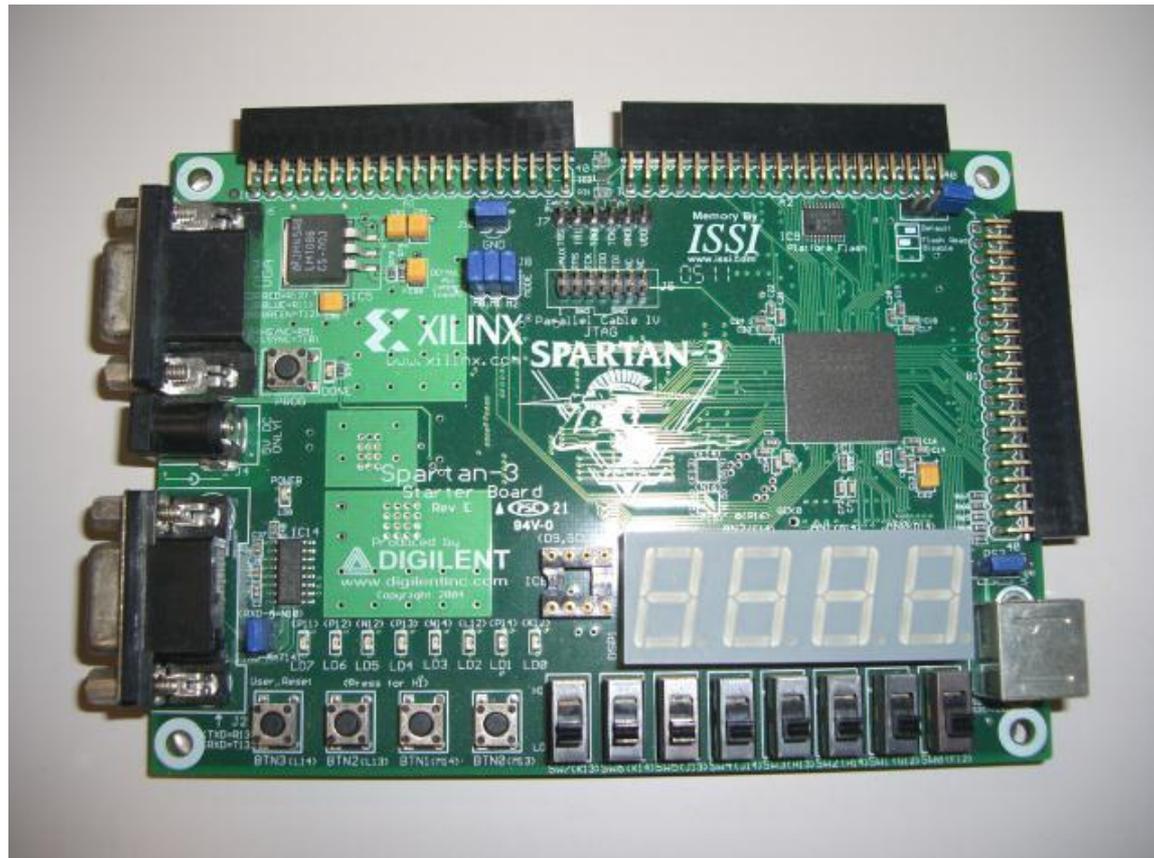


ラジオシティ法 サンプル画像

上嶋 明氏 作成、富士通 AP1000+ 64プロセッサを使用、1997.



入門 SPARTAN-3 Starter Kit 3(20万ゲート), 3E(50万), 3A(70万)



VIRTEX5



DL-Box



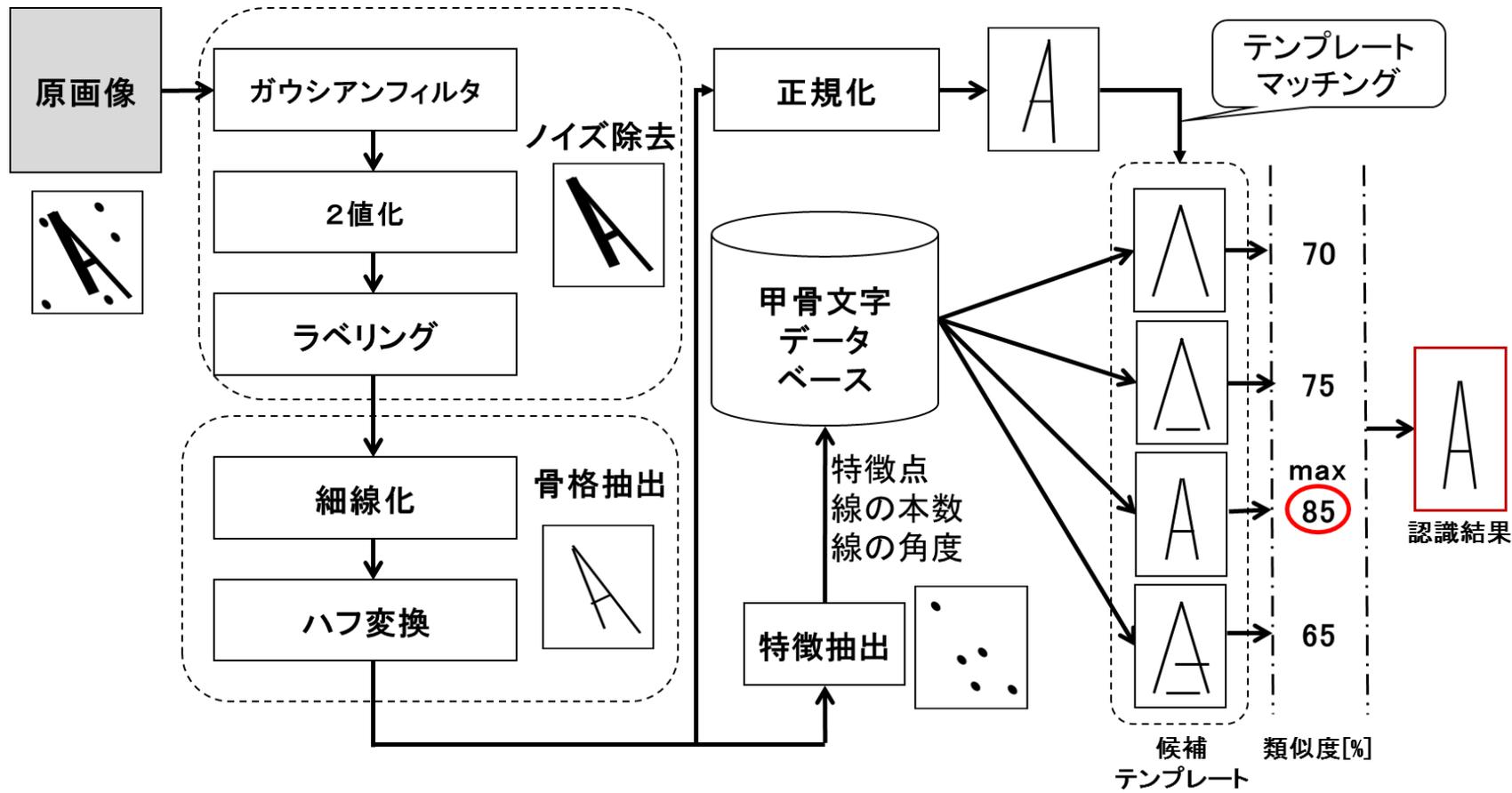
ラズベリーパイ



8. おわりに

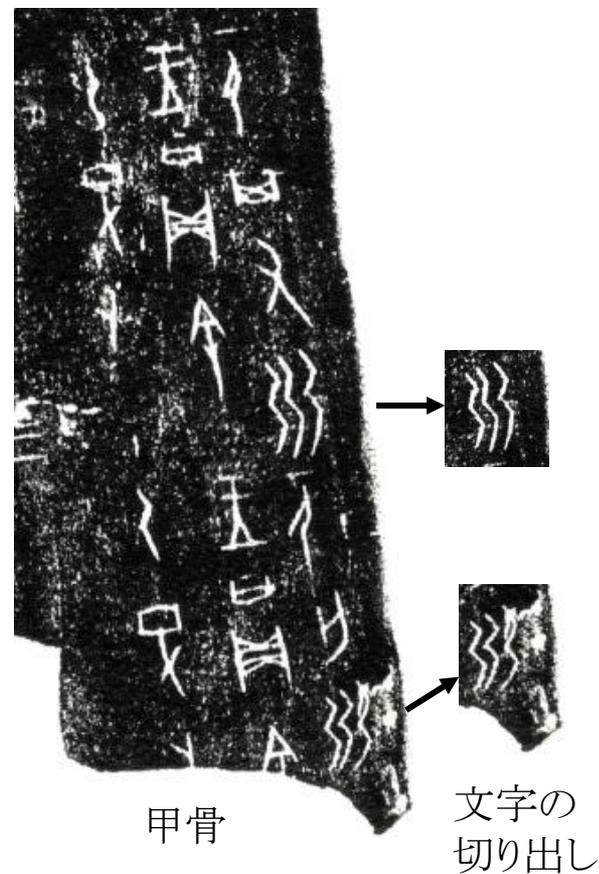
- 今日のスライドを高性能計算研究室のHPにアップしてあります。
- 興味のある人は、ぜひ研究室見学に来て下さい。
- 気力、体力、知力のある人 歓迎！
- 研究室デモ 10月31日(火) 12時～

甲骨文字認識システム



5. 1 画像処理とアフィン変換による正規化を用いた甲骨文字認識の研究

- 研究背景
 - 3000年以上前、劣化、認識困難
 - 文字の起源・変化と未解読資料の解読に重要
 - 中国の古代史学、古典文学の研究
- 研究手法
 - 画像処理: ノイズ除去、細線化、ハフ変換
 - テンプレートマッチング 類似度計算
 - 複数文字の同時認識: GPU並列処理
 - 深層学習: 大量学習 自動認識



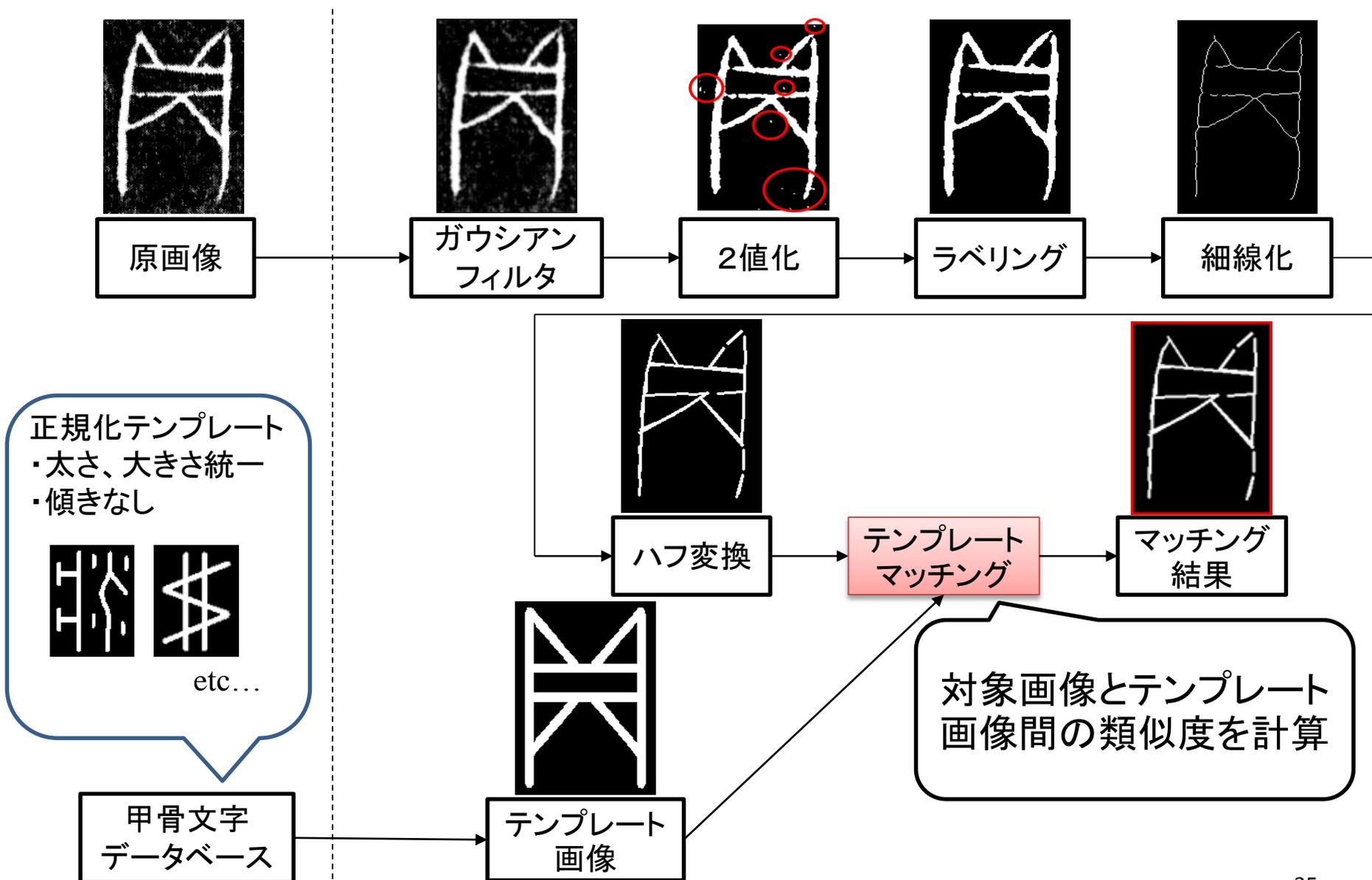
テンプレート



原画像 6枚



テンプレートマッチングによる甲骨文字認識



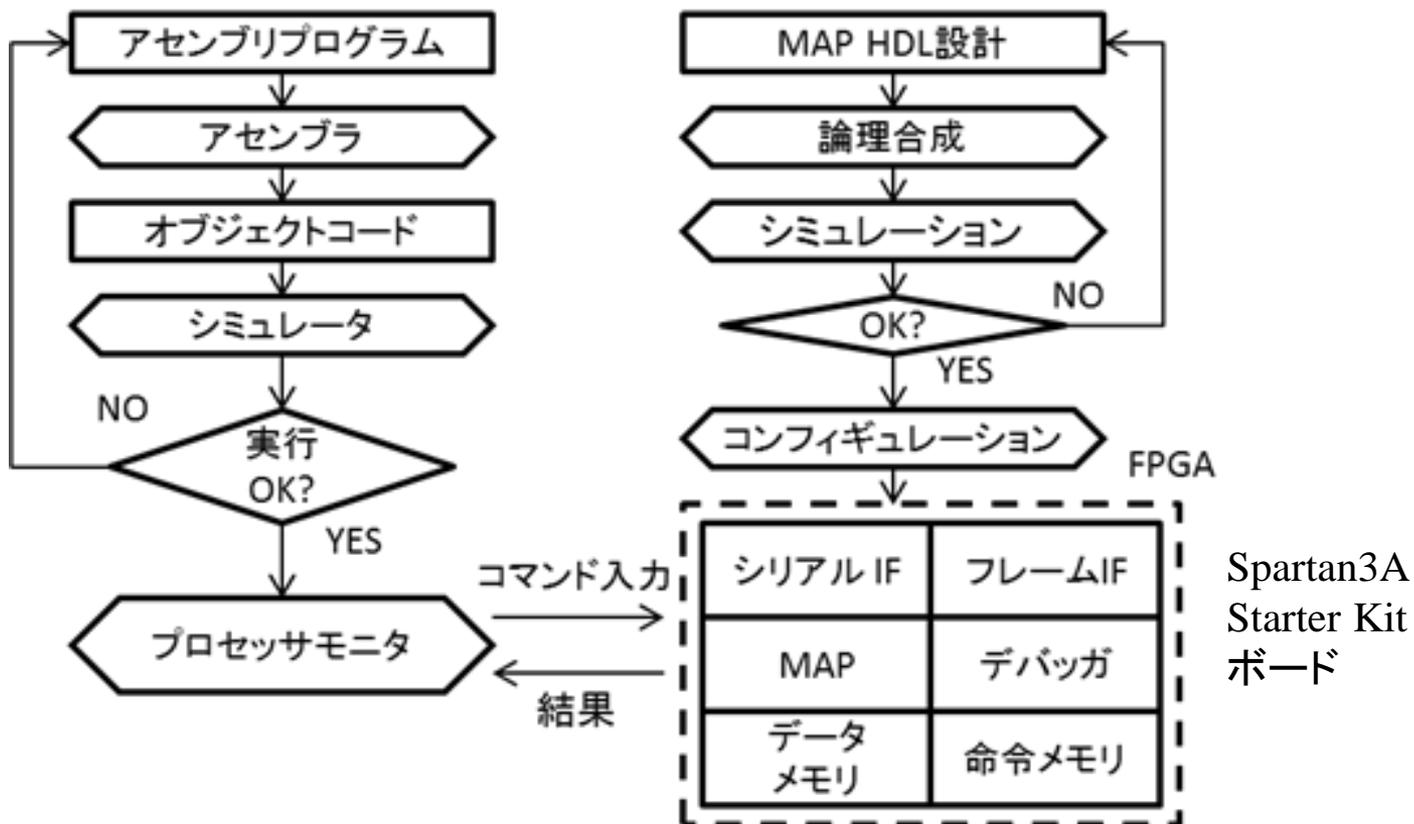
アフィン変換による正規化

回転 拡大 平行移動

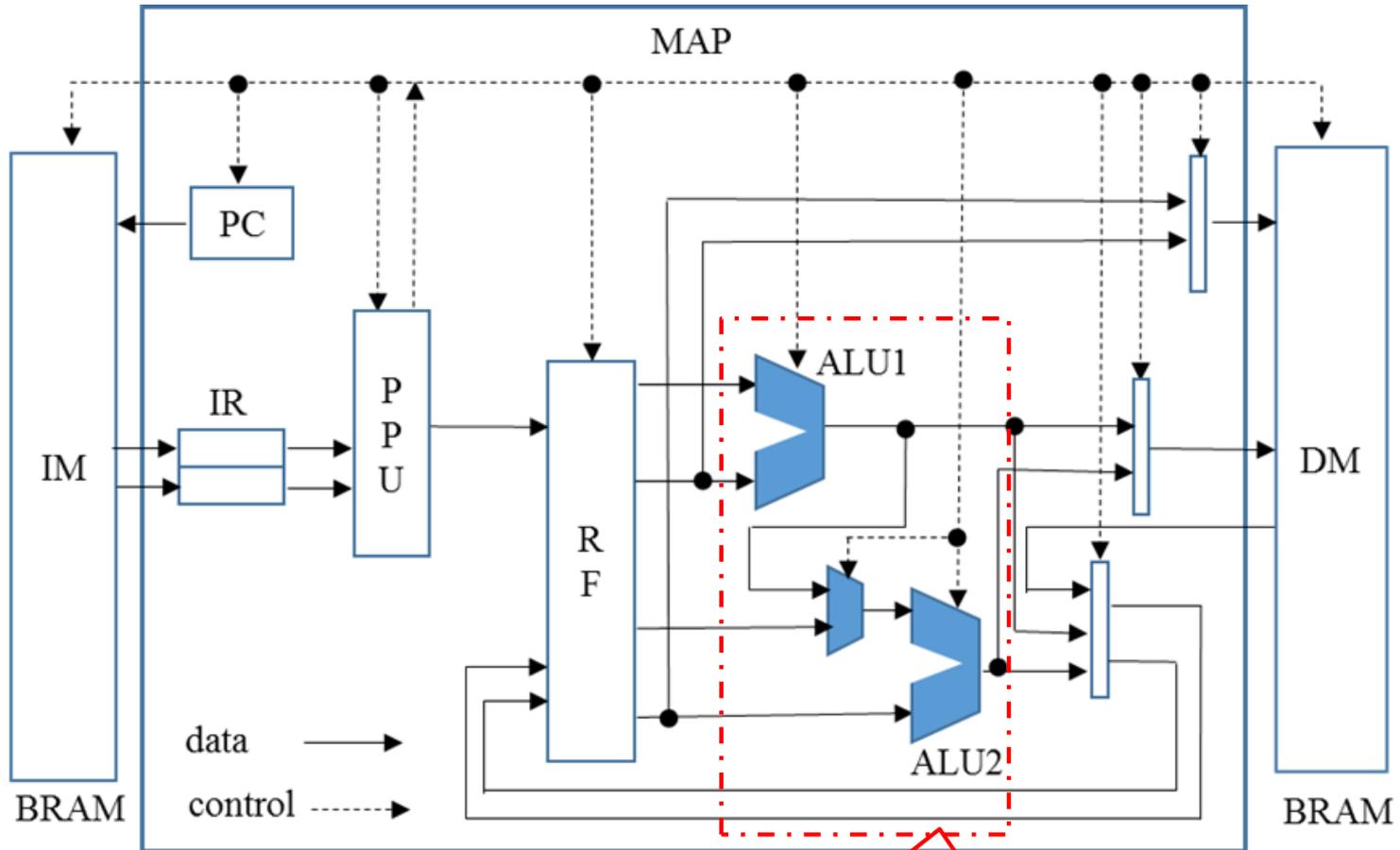
Original Image	Hough Transform	Normalization	Template Image	Angle	Power (x, y)	Position (x, y)	Similarity	
							Before	Improvement Rate
							After	
				5	0.53, 0.92	4, 4	0.16	1.78
							0.29	
				5	1.15, 0.91	1, 0	0.15	2.88
							0.43	
				-5	1.04, 0.50	-3, 2	0.15	3.03
							0.44	
				-5	1.09, 0.80	1, 0	0.28	1.85
							0.51	

5.4 マルチALUプロセッサMAPの設計

- 複数ALUによる並列処理 ALU数: 2, 4, 8, 16
- 演算レベル並列処理: ALUで並列演算と連鎖演算



2ALUのMAPのデータパス



Numbers of ALUs : 2, 4, 8, 16

ハード/ソフト・コデザインの目標

- プロセッサ設計
- 命令セットアーキテクチャ
- アセンブリ言語、C言語、コンパイラ
- ハードウェア設計言語(HDL)
- ハードウェアとソフトウェアの最適バランス
- FPGA: プログラム可能なLSI

目標

- マルチALUプロセッサを設計してFPGA上で動かす。
- 画像処理、パイプライン処理などをFPGA上で動かす。
- FPGA上で動作させて、感激を味わう。

MAPの命令セットアーキテクチャ

1命令32ビット 4命令形式 MIPSサブセット

命令語長		32						命令種類
		6	5	5	5	5	6	
命令形式	R形式	Op	Rs	Rt	Rd	Shamt	Fn	ADD,SUB,AND,OR,XOR,NOT,NOP,SLT SGT,SLE,SGE,SEQ,SNE,SLL,SRL,SRA,JR
	I形式	Op	Rs	Rd	imm			ADDI,SUBI,ANDI,ORI,XORI,SLTI,SGTI, SLEI,SGEI,SEQI,SNEI,LDHI,LDLI
	L形式	Op	Rs	Rd	address/immediate			BEQZ,BNEZ,LD,ST,JAL
	J形式	Op	address					JUMP,HALT

2ALUによる並列処理

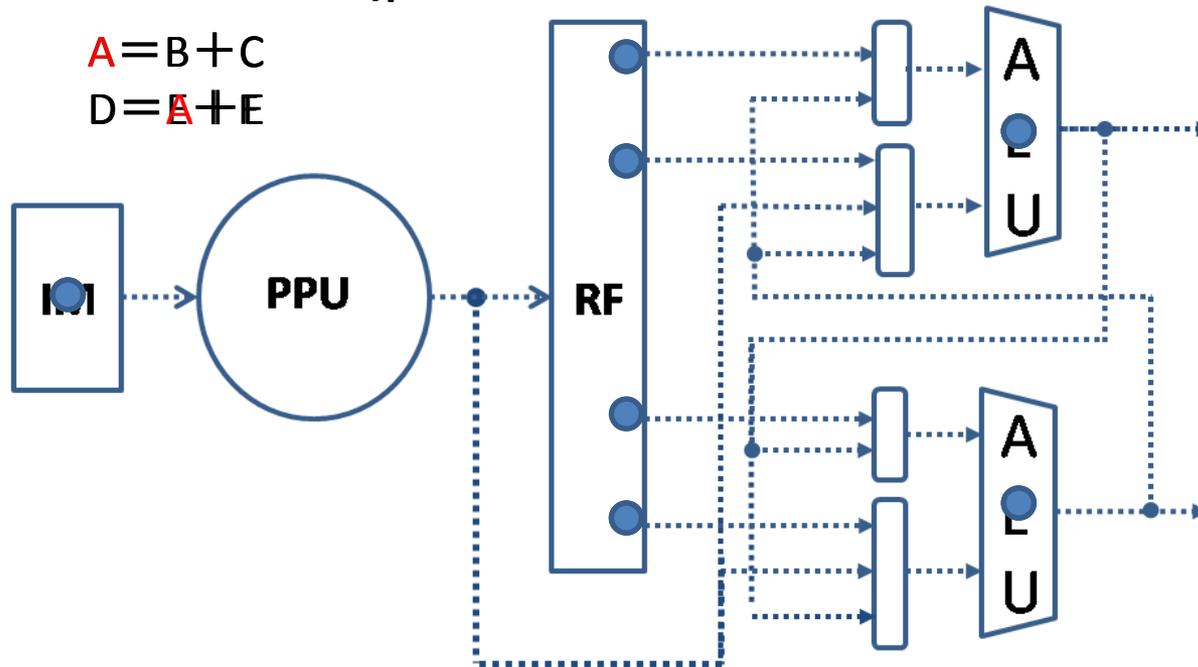
ハードウェアによる並列性の検出

PPUで並列演算、連鎖演算、単一演算を判定

連鎖演算 Parallel Operations: PPO

$$A = B + C$$

$$D = A + E$$



MAPプログラミング例

プログラマは並列演算や連鎖演算を考慮せずに1命令ずつ記述する。

整数同士の乗算

$$c = a \times b$$

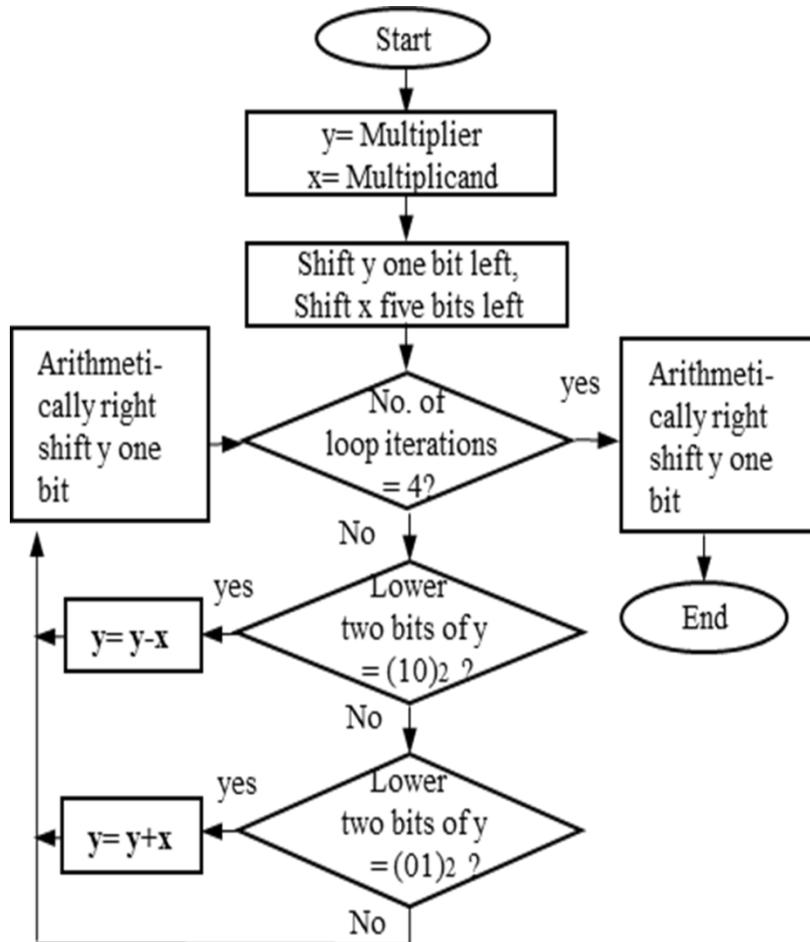
	SUB	\$0	\$0	\$0	} 連鎖演算
	LD	\$1	0[\$0]		
	SUB	\$3	\$3	\$3	} 並列演算
	LD	\$2	4[\$0]		
LOOP:	SUBI	\$2	\$2	1	} 連鎖演算
	SGTI	\$4	\$2	0	
	ADD	\$3	\$3	\$1	} 並列演算
	BNEZ	\$4	LOOP		
	ST	\$3	8[\$0]		... 単一実行
	HALT				

FPGAボード上への実装

- Spartan-3A Starter Kitと、論理合成ツールISE13.2を使用
- プロセッサデバッガを設計・開発し、実動作を確認

コマンド	ターゲット	意味
send	dm/im/rf	メモリ、レジスタの書き込み
read	dm/im/rf/pc	メモリ、レジスタの読み出し
save	dm/im/rf/pc/bp	メモリ、レジスタの内容を保存
load	dm/im/rf/bp	ファイルからロード
set	pc/bp	PC、ブレイクポイントの設定
del	bp	ブレイクポイントの削除
list/init	dm/im/rf/pc/bp	メモリ、レジスタの表示/初期化
run		通常実行
run clk N		Nクロック実行
run bp		ブレイク実行

ブース乗算 並列演算と連鎖演算



	LD	\$1	0[\$0]		①	PO
	LD	\$2	4[\$0]		②	PO
	ANDI	\$2	\$2	15		
	SLL	\$1	\$1	5	③	PO
	SLL	\$2	\$2	1		
LOOP:	ADDI	\$15	\$15	1	④	CO
	SEQI	\$16	\$15	5	⑤	CO
	BNEZ	\$16	LAST		⑥	SO
	ANDI	\$3	\$2	3	⑦	CO
	SEQI	\$4	\$3	2	⑧	CO
	BNEZ	\$4	SKIP1		⑨	SO
	SEQI	\$5	\$3	1	⑩	CO
	BNEZ	\$5	SKIP2		⑪	SO
	SRA	\$2	\$2	1	⑫	CO
	JUMP	LOOP				
SKIP1:	SUB	\$2	\$2	\$1	⑬	CO
	SRA	\$2	\$2	1	⑭	SO
	JUMP	LOOP				
SKIP2:	ADD	\$2	\$2	\$1	⑮	CO
	SRA	\$2	\$2	1	⑯	SO
	JUMP	LOOP				
LAST:	SRA	\$2	\$2	1		
	ST	\$2	12[\$0]			
	HALT					

CO: Chaining Operations
PO: Parallel Operations
SO: Single Operation

ブース乗算の並列性の評価

		Instrs	2ALU with Chaining				4ALU with Chaining					Non-Chaining		
			PO	CO	SO	Cyc.	PO2	CO2	CO3	SO	Cyc.	PO	SO	Cyc.
1st Order	Number	24	4	15	12	31	5	8	9	1	18	7	36	43
	Ratio(%)	-	13	48	39	-	28	44	50	6	-	16	84	-
2nd Order	Number	20	3	14	7	24	3	11	5	1	16	5	31	36
	Ratio(%)	-	13	58	29	-	19	69	31	6	-	14	86	-
3rd order	Number	39	4	30	6	40	2	25	5	2	33	5	64	69
	Ratio(%)	-	10	75	15	-	6	76	15	6	-	7	93	-

2ALU: 連鎖演算が48%~75%

4ALU: 2連鎖演算と3連鎖演算 有効

連鎖演算なし: 単一演算が84%以上

MAP 現状と今後の研究内容

- 2ALUのMAPをHDLで設計して、MAP本体のSpartan3A上での実動作を確認
- MAPアセンブラ、シミュレータを作成、MAPプログラミング
- Booth乗算(1次、2次、3次)のFPGAボード上での実行
- 並列演算と連鎖演算の有効性を評価し、HEART2014で発表

現在の研究内容

- 2ALUMAPのパイプライン設計と動作確認
- 4ALUMAPの並列・連鎖検出ユニットの設計

ガウシアンフィルタと2値化

- ガウシアンフィルタ

- ガウス関数を用いて、画像の平滑化を行う

$$\text{Gaussian_Filter}(x,y) = \begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix}$$



- 2値化

- 画像を白黒にする

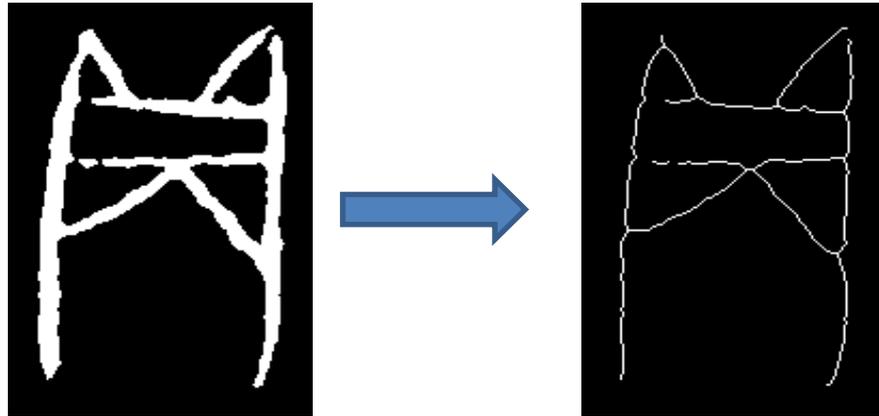
$$\text{Binarization}(x,y) = \begin{cases} 255 & (\text{Pixel}(x,y) > \text{Threshold}) \\ 0 & (\text{Pixel}(x,y) < \text{Threshold}) \end{cases}$$



細線化とハフ変換

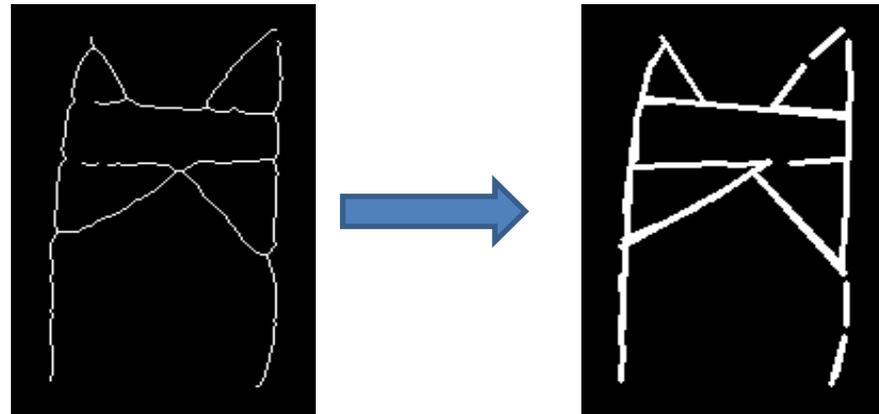
- 細線化

- パターンに基づいて、2値化画像を線幅が1画素の画像に変換
(田村法)



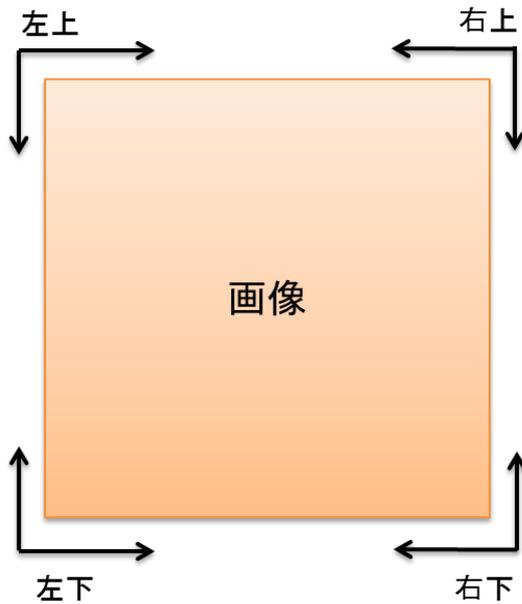
- ハフ変換

- 直線を表す代数方程式を用いて、x-y空間座標から ρ - θ 極座標に変換し、直線を抽出する。

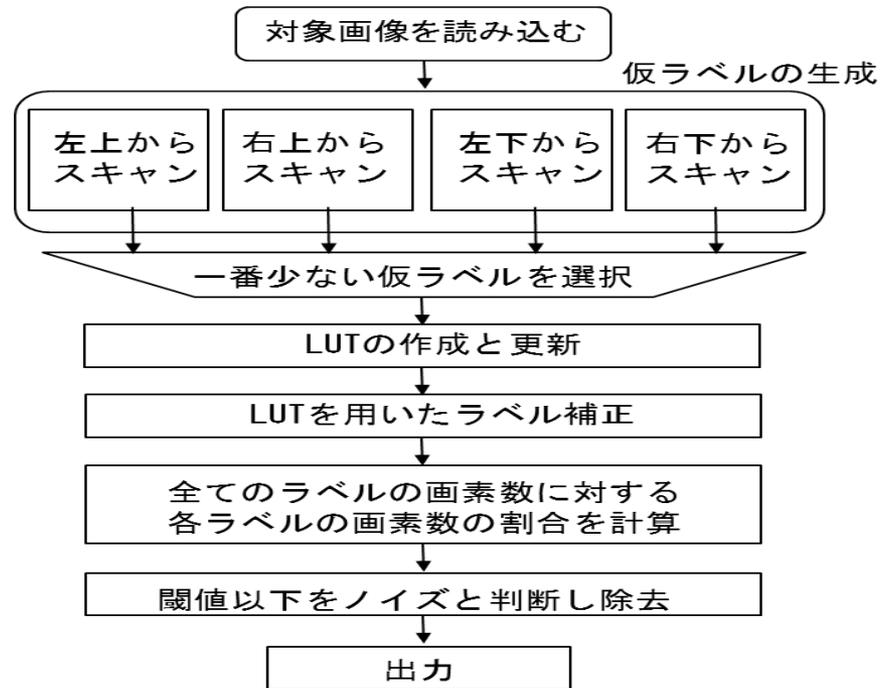


4方向ラベリング

- LUT(ルックアップテーブル)を用いたラベリングの問題
 - 仮ラベル数が多い場合、LUTの更新とラベル補正にかかる時間が長い
- 提案手法
 - 画像の4隅からスキャンを行い、仮ラベルが最少の方向を選択し、ラベル補正を行う



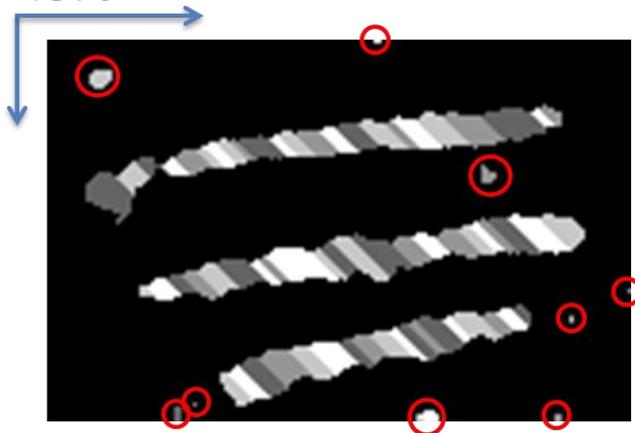
(a) 4方向スキャン



(b) 処理フロー

ラベリングの実験結果

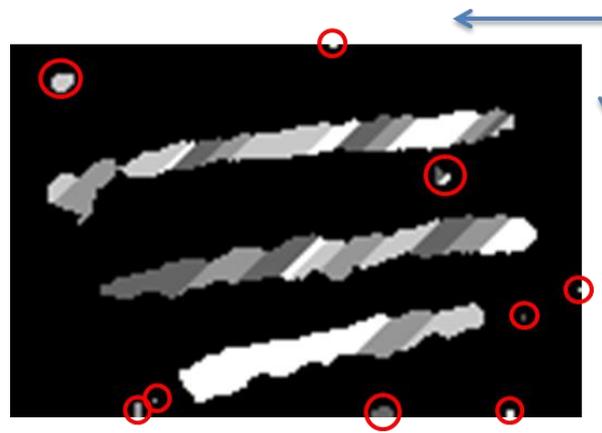
スキャン方向



仮ラベル数	75
-------	----

1方向ラベリングでの仮ラベル生成

スキャン方向



仮ラベル数	37
-------	----

4方向ラベリングでの最少の仮ラベル生成

	1方向	4方向
除去したラベル数	9	
補正後ラベル数	3	
処理時間(ms)	1.110	0.613

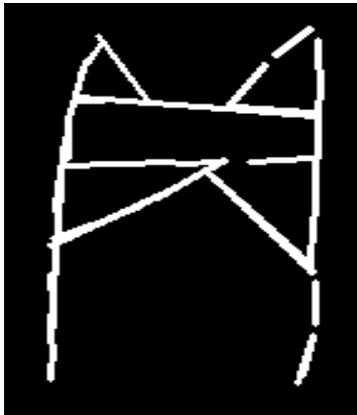


ノイズ除去と処理時間比較

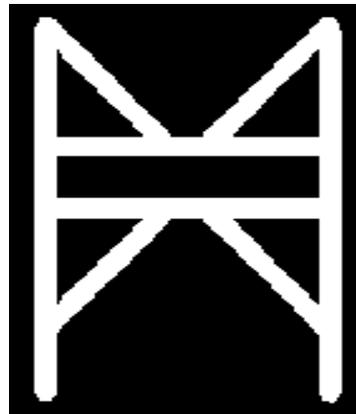
テンプレートマッチング

- 予め用意した既知の甲骨文字のテンプレートと、検出対象の画像を比較し、2つの画像の類似度を計算する
 - 正規化相互相関関数を使用し、2つの画像をベクトル表現し、ベクトル間の内角でマッチングするかどうかを判断する
 - 閾値 ($\cos\theta$) は0.7に設定する

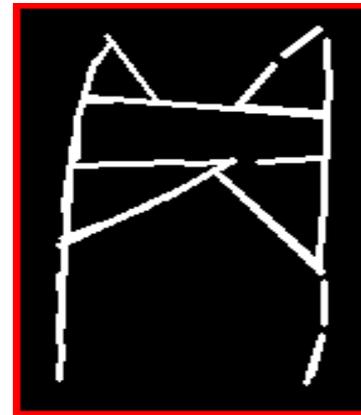
$$R = \cos\theta = \frac{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} I(i,j)T(i,j)}{\sqrt{\sum_{j=0}^{N-1} \sum_{i=0}^{M-1} I(i,j)^2 * \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} T(i,j)^2}}$$



ハフ変換後画像



テンプレート画像

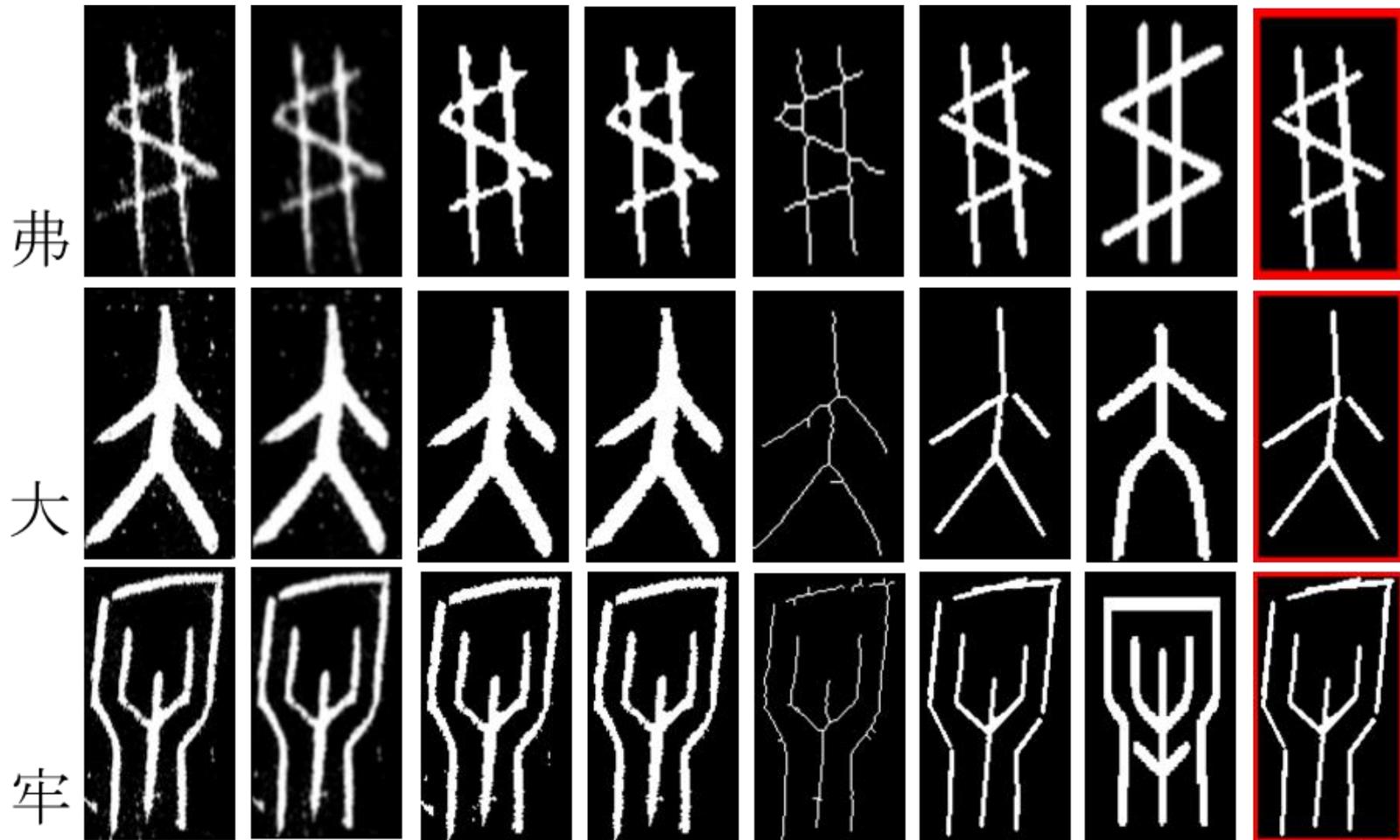


マッチング結果

実験環境

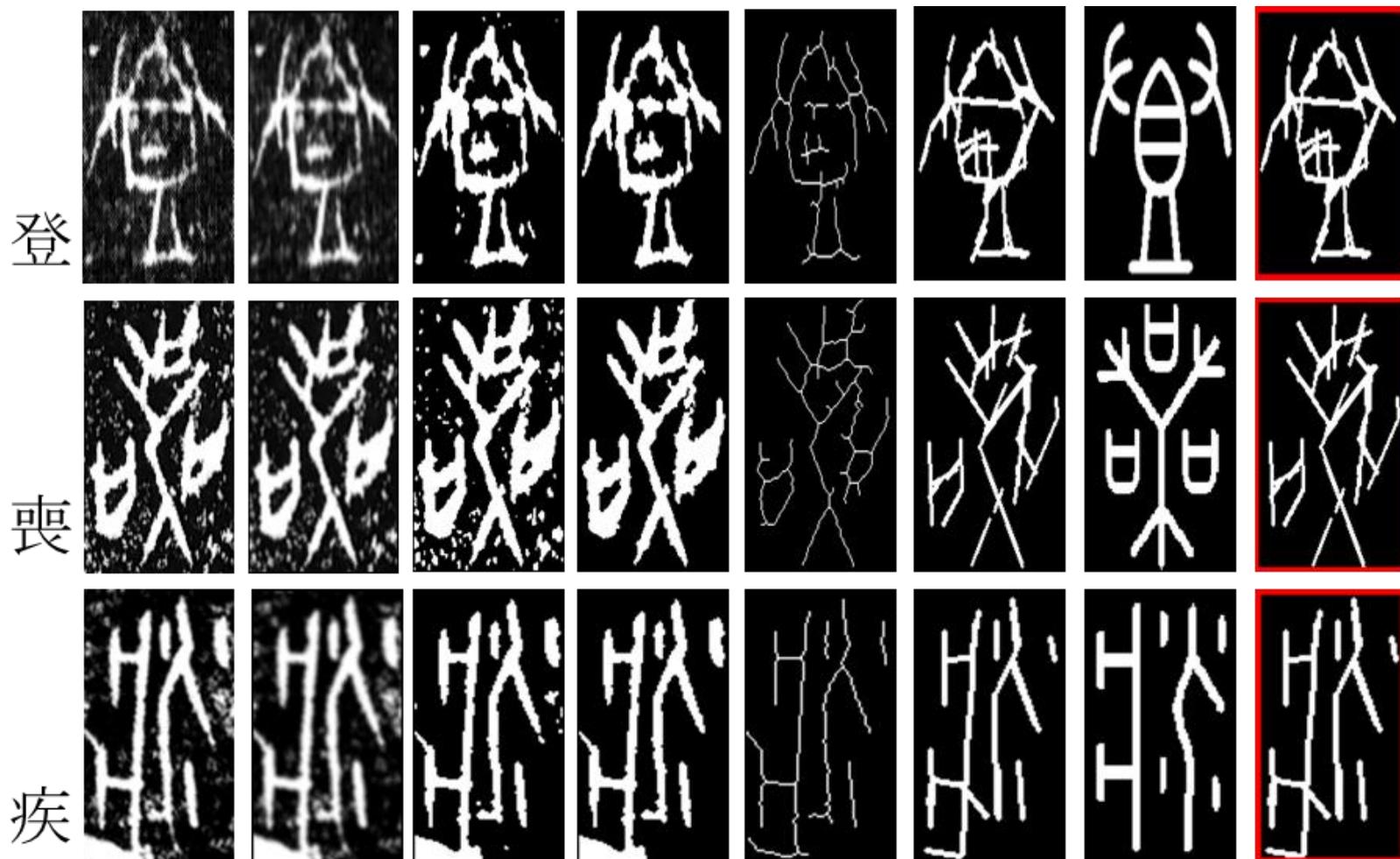
- Intel(R) Core(TM)i7-3820Processor 3.60GHz
 - 実装メモリ : 8.00GB
- Ubuntu13.10
- プログラミング言語
 - C(GCC4.8) : ガウシアンフィルタ、2値化、ラベリング
 - OpenCV 2.4.9 : ハフ変換、細線化、テンプレートマッチング
- 実験対象文字 : 91文字

簡易な文字の認識結果



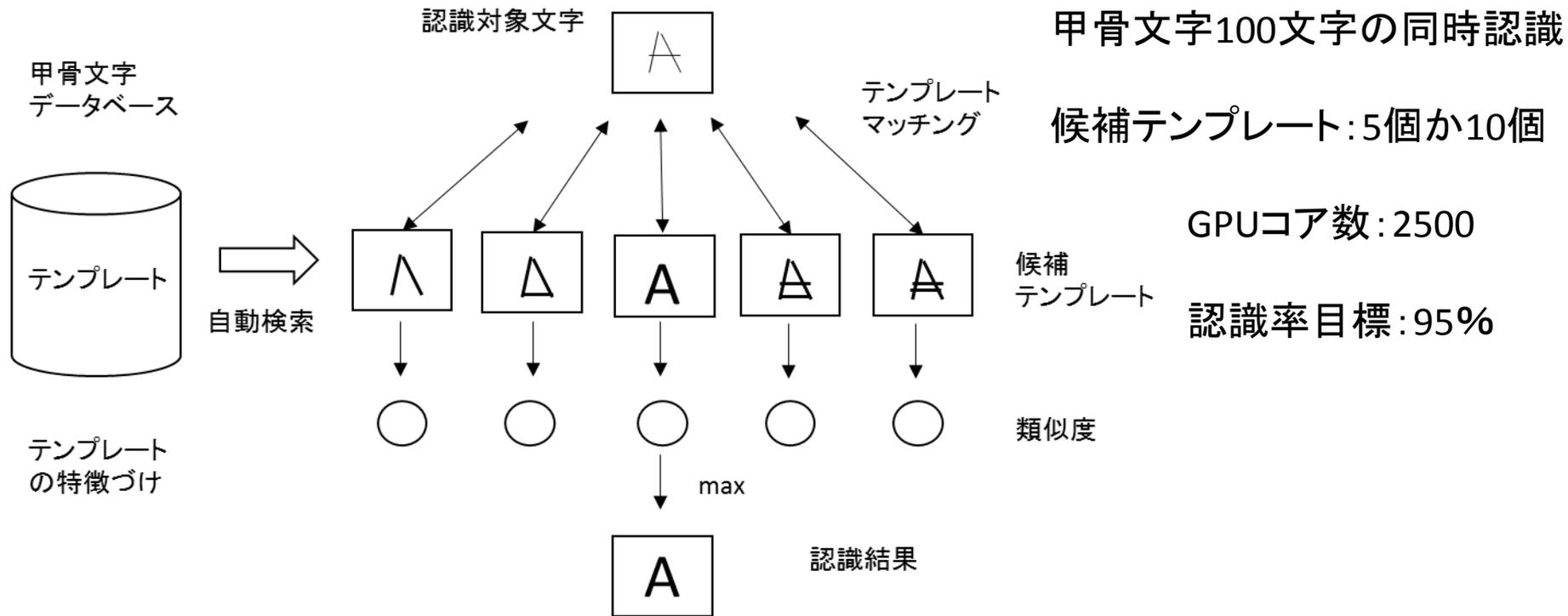
(a)原画像 (b)ガウシアンフィルタ (c)2値化 (d)ラベリング (e)細線化 (f)ハフ変換 (g)テンプレート画像 (h)マッチング結果

複雑な文字の認識結果



(a)原画像 (b)ガウシアン (c)2値化 (d)ラベリング (e)細線化 (f)ハフ変換 (g)テンプレート (h)マッチング
フィルタ 画像 結果

GPUによる超並列処理を用いた 甲骨文字認識の高速化



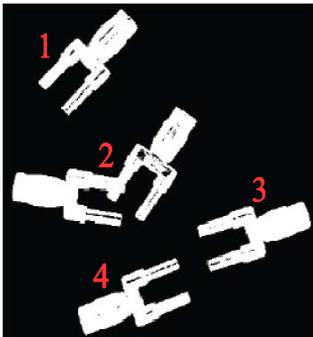
BLOB解析

- 重心の算出

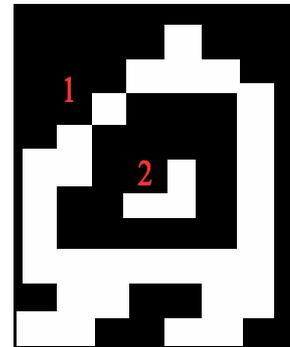
$$Xの重心 = \frac{\text{全ての}X座標の合計}{\text{検出対象}BLOBの面積} \quad Yの重心 = \frac{\text{全ての}Y座標の合計}{\text{検出対象}BLOBの面積}$$

- BLOB解析

– BLOBの個数、各BLOBの面積と重心を算出(面積:ピクセル数)



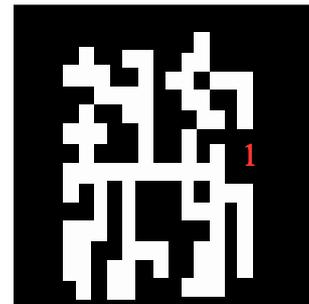
BLOB	面積	重心(X,Y)
1	320	(30,26)
2	630	(42,52)
3	320	(84,66)
4	320	(52,88)



BLOB	面積	重心(X,Y)
1	3643	(58,64)
2	275	(48,52)



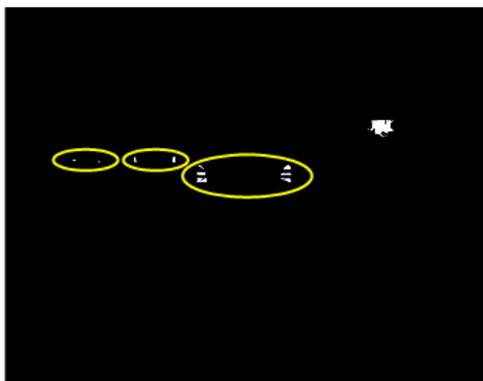
BLOB	面積	重心(X,Y)
1	214	(67,7)
2	134	(10,24)
3	235	(24,20)
4	127	(14,60)
5	111	(71,77)
6	213	(44,81)



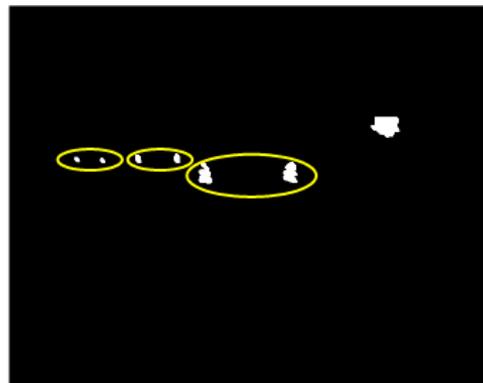
BLOB	面積	重心(X,Y)
1	2647	(44,59)

テールライト抽出

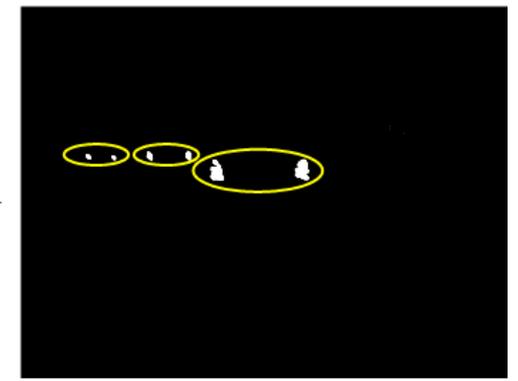
- 赤色成分の抽出と肥大化
 - 左上から右下に向かってラスタ走査を行い、赤色成分であれば、周囲8近傍を赤色成分に変換
- ラベリング
 - 各赤色成分にラベル番号の割り当て
- BLOB解析
 - 面積と重心を算出し、左右対称性評価



赤色成分抽出



赤色成分の肥大化



左右対称性評価 57

車両エリア検出

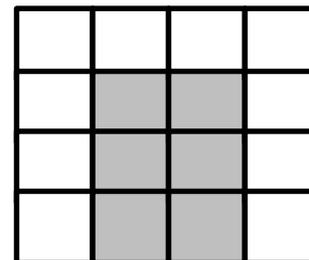
- 領域内分割を用いたソーベルフィルタと2値化
 - 画像中央部に対して水平と垂直方向のマスクパターンを適用し、エッジを強調

-1	0	1
-2	0	2
-1	0	1

水平方向

-1	-2	-1
0	0	0
1	2	1

垂直方向



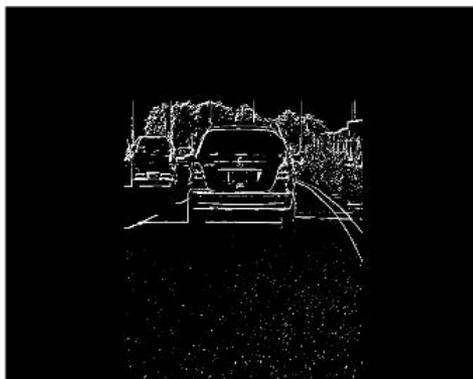
領域分割

- BLOB解析

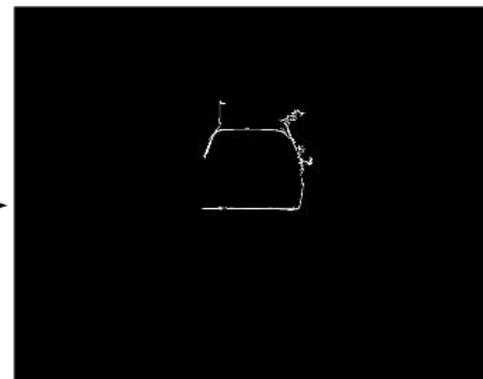
- 車両エリアの輪郭のX,Y座標の最小値と最大値の算出



原画像

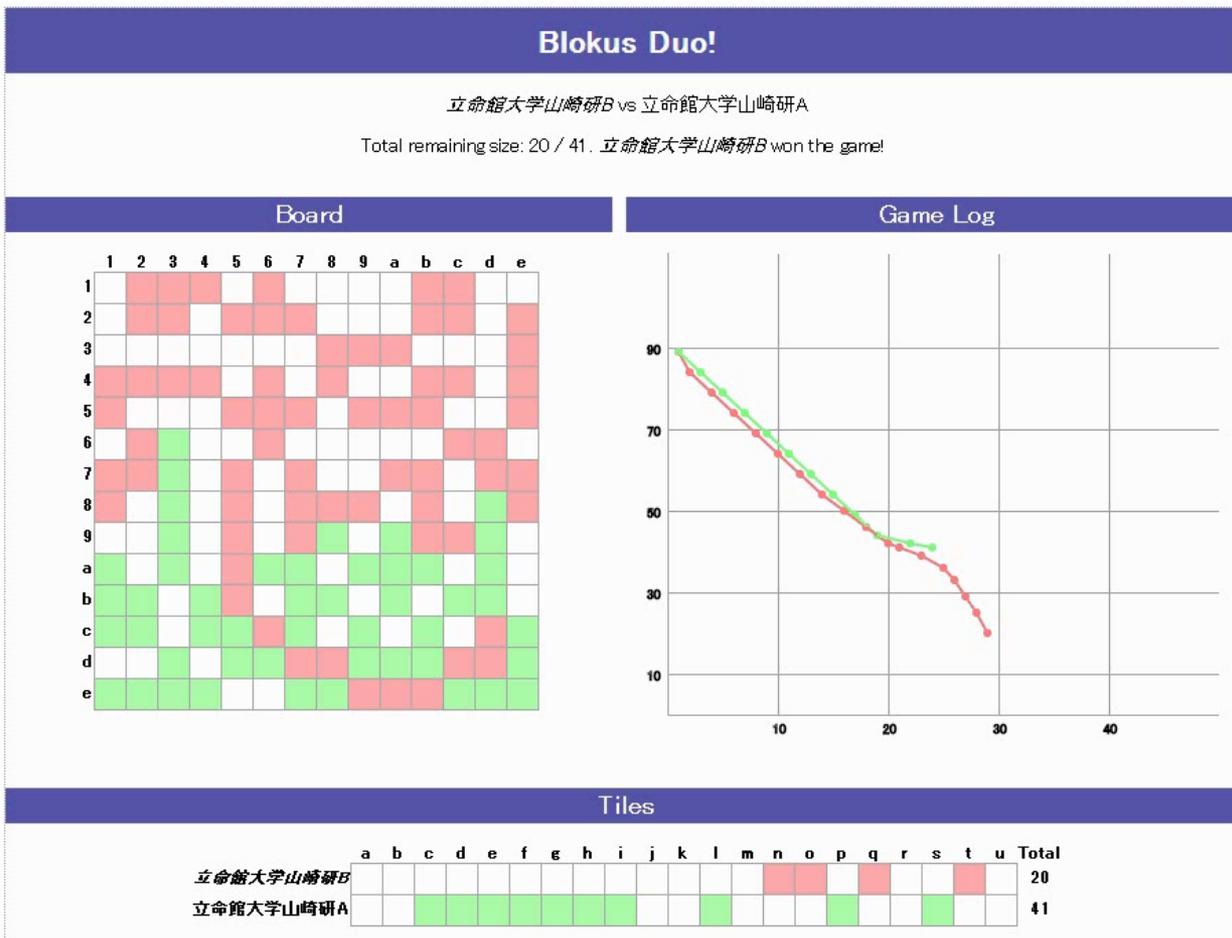


ソーベルフィルタと2値化



周囲長の算出 58

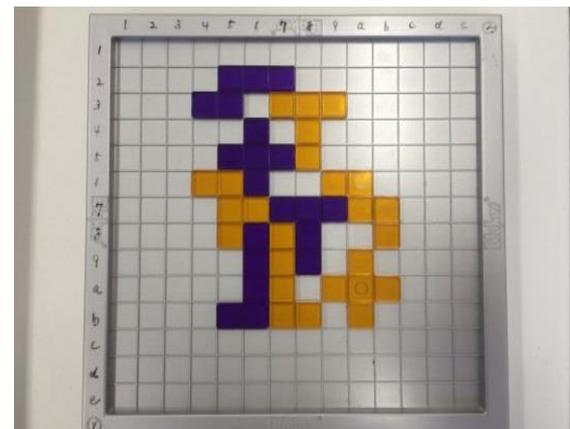
FPGA設計コンテスト: BlokusDuoの設計



第3回 相磯秀夫杯 FPGAデザインコンテスト

2013年9月18日 北陸先端科学技術大学院大学
電子情報通信学会 リコンフィギャラブル研究会

21チーム参加
予選4グループ Bグループ 1位
決勝リーグ 3位入賞



BlokusDuoボード



FPGAデザインコンテスト 表彰式



2台のFPGAでの対戦

チェス、将棋、囲碁の比較

	チェス	将棋	囲碁
局面数	10^{120}	10^{220}	10^{360}
コンピュータ の強さ	チャンピオン に勝利	プロ棋士以上	トッププロ以上
人間に勝った のはいつ	1997年 Kasparov	2012年 前名人 2014年	2016年3月
トピック	IBM DeepBlue	評価関数の 機械学習	深層学習 アルファ碁
電王戦	第1回(2012年1月)~第4回(2015年3月) 第1回:ボンクラーズ 米長邦雄元名人に勝利 第2回:ソフトの3勝1敗1分け プロ棋士 4段から8段 第3回:ソフトの4勝1敗 プロ棋士 5段から9段 第4回:プロ棋士の3勝2敗 第2局はソフトのバグ		2016年1月 Fan Hui2段 (欧州チャンピオン)に 5-0で勝利 2016年3月 イセドル9段 (韓国トッププロ)に 4-1で勝利
情報処理学会 2015年10月	コンピュータ将棋プロジェクト終了宣言		

Nyctoクラスタ: サーバー(8プロセッサ) × 2

