修士論文

CNN を用いた静止指文字画像の領域セグメンテーション

平成 28 年度修了

三重大学大学院 工学研究科
博士前期課程 情報工学専攻

小島　広嵩

目次

はじめに 1

第 1 章　序論 2
1.1 研究の背景と目的 . . . 2
1.2 関連研究 . . . 3
1.3 指文字 . . . 4

第 2 章　基本知識 5
2.1 畳込みニューラルネットワーク . . . 5
2.1.1 畳込み処理 . . . 6
2.1.2 プーリング処理 . . . 7
2.1.3 正規化処理 . . . 8
2.1.4 出力層 . . . 8
2.2 勾配降下法 . . . 9
2.3 誤差逆伝播法 . . . 10
2.4 Fully Convolutional Network . . . 13
2.4.1 逆畳込み処理 . . . 13

第 3 章　提案手法 15
3.1 学習画像 . . . 15
3.2 近傍画素を考慮した誤差関数の導入 . . . 16
3.3 学習手法について . . . 18

第 4 章　実験 19
4.1 実験データ . . . 19
4.2 実験概要 . . . 21
4.3 評価方法 . . . 22
4.4 実験結果 . . . 24

4.5 考察 28

おわりに 31

謝辞 32

参考文献 33

付録 34

1 **作成したプログラムおよび実験データについて** 34

はじめに

画像認識の研究において，畳込みニューラルネットワーク (以下，CNN)[1] などのニューラルネットワークを利用した手法が高精度を出し，様々な問題に応用されている．CNN とは，多層ニューラルネットワークの一種で，ネットワーク内の層に画層処理に特化した層を含むネットワークである．ネットワークのはたらきとして，特徴量抽出から識別を自動で行うことが可能である．現在では，画像内に写っている人物や物体名を出力するクラス分類を始め，画素単位で画像から領域を抽出する領域セグメンテーションの研究がされている．CNN の学習は通常のニューラルネットワークと同様に，誤差逆伝播法を用いて行わる．誤差関数の値が最小となるように重みの値を調節し，入力に対して最適な識別結果を出力する．また，誤差関数を変えることで，画像認識の様々な問題に応用することができる．しかし，領域セグメンテーションの研究で利用されている誤差関数は従来のクラス分類で使用する誤差関数と同様であるため，領域セグメンテーションにおいて背景と前景を分ける際に重要となる近傍画素の情報が誤差関数に含まれていない．このため，画像内に写っている人物や物体のエッジが上手く再現されていない可能性がある．この問題に対して，本論文では，ネットワークに近傍画素を考慮した誤差関数を導入し，学習・識別を行うことを提案する．

実験は，静止指文字画像を用いて行う．指文字は手話の標準化された形の一種である．五十音に対応しているため，新出単語や固有名詞などを表現する際に多く利用されている．現在，聴覚障害者と手話を習得していない健常者との間ではコミュニケーションを行うことは難しい．そのため，円滑なコミュニケーションを支援するシステムを開発するために研究が行われている．本研究では，この問題を画像認識の問題として焦点を合わせる．従来の指文字認識の手法では，決められた背景で指文字を撮影し，撮影画像と背景との差分を取り手領域を抽出していた [4]．また，照明の変化などが激しいと手動で背景を画像から取り除いていた [13]．これは，肌色抽出が非常に難しく，有効な手段が確立されていないためである．そのため，本研究では背景などのノイズを含んだ画像に対して，以上で述べた CNN を用いた指文字の種類と形状を示すシステムを開発する．

本論文では，第 1 章で研究の背景と目的について述べる． 次の第 2 章では関連研究を紹介する．第 3 章で提案手法について記述し，第 4 章で提案手法を用いた実験について述べる．

第 1 章
序論

1.1 研究の背景と目的

日本には，難聴で苦しむ人が 600 万人以上いると言われ，その多くは，手話を用いたコミュニケーションを行っている．しかし現在では，手話を習得していない健常者と聴覚障害者の間では，コミュニケーションをとることが難しい．そのため，コンピュータを用いた手話に関する研究が盛んに行われてきた．手話の標準化された形の一種に，指文字がある．指文字とは，五十音を表現することができるため，手話では表現することができない固有名詞や新出単語などを表現することができる．そのため様々な場面でこの指文字が利用されている．手話による研究と比較すると，指文字を含んだシステムの開発は少ない．そこで，健常者と聴覚障害者とのコミュニケーションを円滑に行うための画像認識システムの開発を行う．

現在，画像認識における多くの問題に対して，CNN を用いた手法が提案され，高精度をあげている．CNN を用いた手法では，画像の特徴量抽出を自動で行うため，人間では特徴量を見つけることが難しい複雑な画像に対しても識別することが可能である．従来では，認識したい物体をクラスに指定して画像内に写っている物体名を識別するクラス分類に応用されていた．現在では，画像内に写っている物体を矩形で囲むバウンディングボックスによる検出 [15] や，物体の領域を画素単位で再現する領域セグメンテーション [16] に応用されている．後者の二つの問題は，画像内の物体の場所や形状をコンピュータが知る必要があるため，クラス分類の問題と比べ出力する情報が多いことで難解である．例えば，コンピュータが物体の位置を取得するには，画像内の物体と背景を画素の集合又は単位で認識する必要がある．そのため，背景の画素が複雑である場合では，背景を除去することが難しく上手く物体と背景を認識することができない．

そこで，本研究では指文字画像に対して CNN を用いた領域セグメンテーションを提案し，指文字認識システムの開発を検討した．

1.2 関連研究

画像認識の研究において，領域セグメンテーションやラベリングの研究は数多く行われている．領域を分割する手法に背景差分法 [2] やグラフカット (Interactive Graph Cuts)[3] がある．背景差分法は高速で領域を分割することが可能であるため，本研究室の指文字認識の研究 [4] においても利用されている．しかし，照明の変化に弱いため，急激な照明変化には対応することができず，画像内の物体を分割することが難しい．また，特定の場所でしか対応することができないため，異なる背景画像では上手く物体を抽出することができない．一方，グラフカットは，エネルギー最小化問題の方法として提案された手法である．グラフカットの代表的なアルゴリズムに min-cut/max-flow[5] がある．グラフカットは，入力画像を有向グラフとみなして，前景と背景を分割する手法である．背景差分法と違い，特定の場所以外でも利用できるため，近年多くの画像認識システムに導入されている．しかし，前景と背景の色分布が似ている複雑な背景の場合は，分割することが難しい．

近年，ディープラーニングの技術が注目され，画像認識では CNN(Convolutional Neural Network，畳込みニューラルネットワーク) が，多クラス分類や，画像復元，バウンディングボックスによる物体検出，ラベリング，領域セグメンテーションの研究に応用されている．CNN は畳込みとプーリングの処理を加えた順伝播型ニューラルネットワークである．学習は従来のニューラルネットワークと同様に誤差逆伝播法 (Backpropagation)[6] を用いて学習を行い，畳込みを行うフィルタの係数を最適化する．そのため，画像に対して様々なパターンのフィルタを構築することができる．また，自動で局所的な特徴量を抽出することができるため，複雑な画像に対しても高い精度を出している．

現在，領域セグメンテーションの分野では一般的な CNN のネットワークに逆畳込み (deconvolution) 層を用いたネットワーク Fully Convolutional network(FCN)[7] が代表的である．このネットワークは畳込みやプーリングを用いて圧縮した特徴ベクトルに逆畳込み処理を利用して，ラベル画像を出力するネットワークである．学習は従来のネットワーク同様に誤差逆伝播法を用いるため，繰り返し学習を行い最適な畳込みフィルタを構築する．このネットワークにより，グラフカットの欠点でもある複雑な背景に対しても領域を分割することが可能であると考えられる．また，画像内の指文字の認識と場所を同時に特定することも可能であると考えられる．

そのため，本研究では CNN を用いた指文字の領域セグメンテーションを提案する．また，従来の CNN を用いた領域セグメンテーションではクラス分類の際に使用する誤差関数を利用しているため，グラフカットなどで利用するエネルギー関数と違い，近傍画素の情報が考慮されていないため物体の細部が上手く再現できず精度が低下している可能性がある．そこで，グラフカットで利用される様なエネルギー関数を参考にして近傍画素の情報を含んだ誤差関数を提案し，従来手法との

精度向上を目指す．

1.3 指文字

指文字は，手の形を文字に対応させた視覚言語の一要素である．
手話には，腕や手の動きを含めた手指動作と顔の一部を用いた非手指動作があるが，指文字は手の動きのみで表現することができる．各国，言語に対して手話や指文字が存在する．例えば，アメリカ英語を基にした American Sign Language(ASL)，イギリス英語を基にした British Sign Language(BSL)，日本では日本語手話が主流である．以下の図 1.1 に日本語の指文字を示す．日本語の指文字では，五十音全てを表現することができる．また，日本語の指文字の手の形には静止指文字と動きを含む文字の 2 種類が存在する．図 1.1 では，動きを含む文字は空白部分である．以下で詳細を述べる．

- 静止指文字
  動きを含まず手の形だけで表現できる．41 文字存在する．
- 動作指文字
  上記の静止指文字以外．清音文字では，「の」「も」「り」「を」「ん」が存在し，濁音，半濁音，長音も動きを含む文字に含まれる．

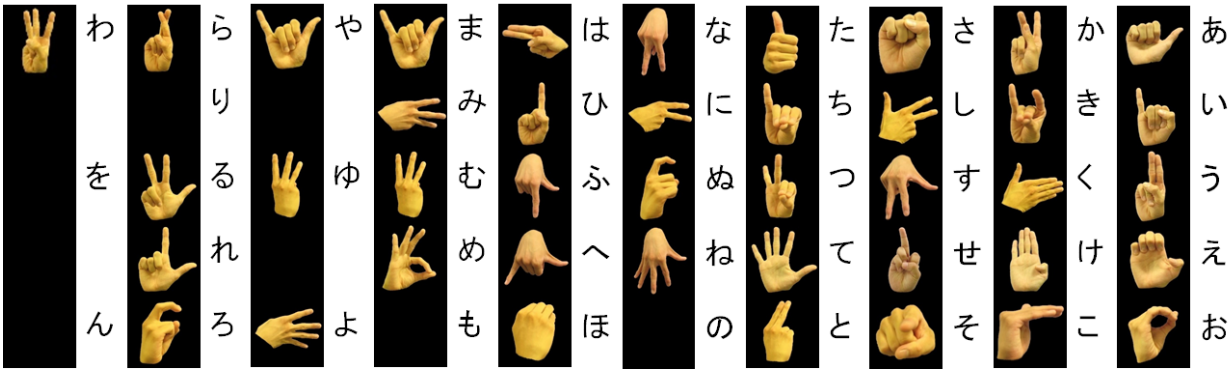


図 1.1　日本語指文字一覧

第 2 章
基本知識

本章では，基本知識に本研究で利用する畳込みニューラルネットワーク（CNN）と FCN について述べる．CNN の説明に関しては著書 [14] を参考にした．

2.1 畳込みニューラルネットワーク

畳込みニューラルネットワーク（Convolutional neural network, CNN）は, 畳込み (convolution) とプーリング (pooling) の処理を複数含んだ画像認識に用いられるニューラルネットワークである．CNN のルーツは Y.LeCun ら [8][9] の研究である．通常の順伝播型ニューラルネットワークと同様に，誤差逆伝播法を用いた勾配降下法で学習を行う．学習では，識別データと正解データの誤差が最小となるように重みを調節する．全体の構造は以下の図 2.1 に示す．ネットワークの前半で特徴ベクトルを算出し，後半で識別を行う．前半では入力画像に対して畳込みとプーリングの処理を行い，識別に最適な特徴ベクトルを計算する．一般的なネットワークでは，畳込み処理の後にプーリング処理が行われる様に設置されている．後半は，従来の順伝播型ニューラルネットワーク同様，全結合層により前半で計算した特徴ベクトルを全て結び付け識別を行う．ネットワークの最後は，各クラスの精度を計算するための出力層が設置される．例えば，10 クラスの問題に対して分類を行うのであれば，出力層のノード数は 10 となる．

また，近年のディープラーニングの研究で開発された革新的なネットワークに，Alex-Net[10], VGG16[11]，Google-Net[12] が挙げられる．以下で畳込みとプーリングなどの CNN を構成する層と学習のアルゴリズムについて詳細を述べる．

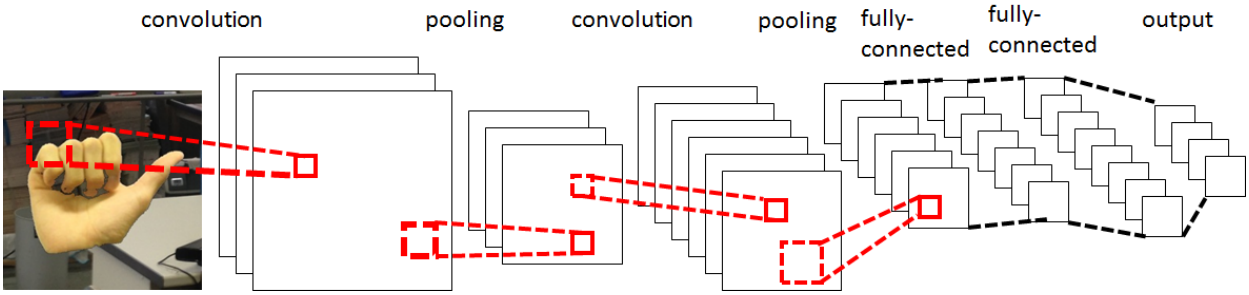


図 2.1　CNN

2.1.1 畳込み処理

畳込み層で行う処理について述べる．

畳込み層では，画像にフィルタを掛け合わせる畳込み演算を行う．1 チャネルの濃淡画像を考える．$W \times W$ の画像サイズに対して，画素値をインデックス $(i,\ j)$ を用いて x_{ij} と表す．$H \times H$ のフィルタを考え，フィルタの画素値をインデックス $(p,\ q)$ を用いて h_{pq} と表す．畳込みは，x_{ij} と h_{pq} を用いて，積和計算を行う．畳込みの式を式 (2.1) に示す．

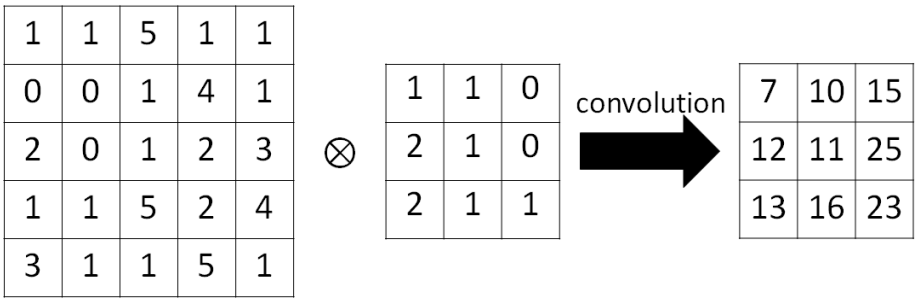


図 2.2　畳込み処理

$$u_{ij} = \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p,j+q} h_{pq} \tag{2.1}$$

次に，チャネルを K 個に拡張した場合を考える．

チャネル番号 $k(= 0, 1, 2, 3, ..., K-1)$ とおく．RGB の入力画像では，チャネルは $K = 3$ である．

また，CNN の中間層では $K = 96$，$K = 256$ の場合を扱う．式 (2.2) にチャネル数を K とした式

を示す．

$$u_{ij} = \sum_{k=0}^{K-1} \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} x_{i+p,j+q,k} h_{pqk} \tag{2.2}$$

次に，式 (2.2) を用いて，畳込み層で行われる計算の詳細を述べる．

第 l 層の畳込み層を考える．前層の $l-1$ 層の出力を $z_{ijk}^{(l-1)}$ に M 種類のフィルタ $h_{pqkm}(m = 0, 1, 2, 3, ..., M-1)$ を適用する．また，各フィルタ共通のバイアス b_m を考慮すると，以下の式 (2.3) になる．

$$u_{ijm} = \sum_{k=0}^{K-1} \sum_{p=0}^{H-1} \sum_{q=0}^{H-1} z_{i+p,j+q,k}^{(l-1)} h_{pqkm} + b_m \tag{2.3}$$

計算で得た u_{ijm} を活性化関数 f で計算した値が，l 層の出力 $z_{ijm}^{(l)} = f(u_{ijm})$ となる．活性化関数の種類は様々であり，一般的には ReLU(rectified linear function，正規化線形関数) が用いられる．この関数は以下の様な式で表され，他の活性化関数と比べ計算が単純であるため高速に学習ができ，最終的に高い精度が得られるためよく利用されている．

$$f(u) = \max(0, u) = \begin{cases} 0 & (u < 0) \\ u & (u \geq 0) \end{cases} \tag{2.4}$$

2.1.2 プーリング処理

プーリング層で行う処理について述べる．

多くの CNN のネットワークでは，プーリング層は畳込み層の後層に配置される．プーリング層では，畳込み層で得た出力から局所的な代表値を算出する．例えば，$W \times W \times K$ の入力に対して画素 (i, j) を中心とする正方領域 $H \times H$ を考える．各正方領域の画素の集合を P_{ij} とおき，各 P_{ij} 毎に代表値を出力する．プーリングの方法には様々な手法があり，画像認識では最大プーリング (Max Pooling) がよく利用されている．これは，集合 P_{ij} 内から画素の最大値 u_{ijk} を出力する．以下に最大プーリングの式 (2.5) を示す．

$$u_{ijk} = \max_{(p,q) \in P_{ij}} z_{pqk} \tag{2.5}$$

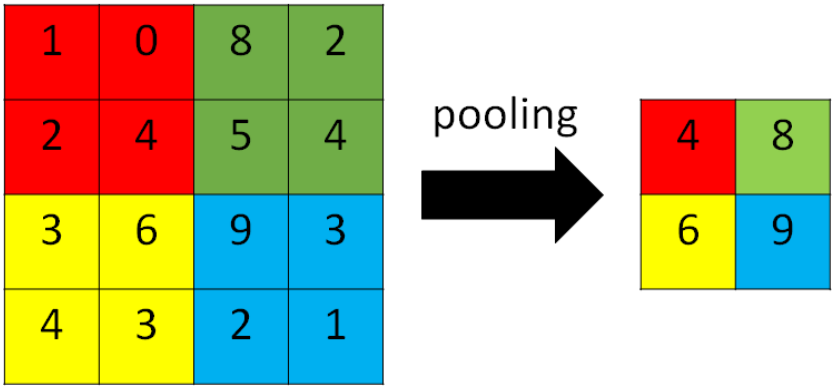


図 2.3　最大プーリング

2.1.3 正規化処理

正規化層での処理について述べる．

この層は中間層の出力に対して正規化を行う．畳込みやプーリングの処理をして，得た出力を特徴マップと呼ぶ．この特徴マップの局所領域に対して，正規化の処理を行う層が正規化層である．正規化には様々な方法が存在し，本研究の実験で用いるネットワークでは LRN(Local Response Nomalization，局所反応正規化) を用いる．LRN は，各画素に対して複数の特徴マップ間で正規化を行う．

$$z_i = \frac{x_i}{\left(\kappa + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} x_j^2\right)^{\beta}} \tag{2.6}$$

式は（2.6）の様になり，入力 x_i に対して出力 z_i を計算する．κ，α，β，n はそれぞれ定数を示す．論文 [10] では，$\kappa = 2, \alpha = 10^{-4}, \beta = 0.75$，$n = 5$ の様になっている．x_j は j 番目の特徴マップの画素値 (インデックスは省略) を示す．したがって，i 番目の特徴マップ x_i に対して，隣接する特徴マップの画素値 x_j の 2 乗和で線形変換した値を求める．LRN の効果として，入力 x_j の値が大き過ぎる場合にそれを抑制することができる．

2.1.4 出力層

CNN の出力層について述べる．

出力層はネットワークの最後に配置され，出力値は精度を計算する際に使用される．分類問題の種類 (二値分類，回帰問題，多クラス分類など) に対して関数が存在し，分類問題に沿った精度を計算することが可能である．多クラス分類では，ソフトマックス関数 (softmax function) を用いた計算が比較的に有名である．ソフトマックス関数を式 (2.7) に示す．クラス分類では，出力層のノード数を識別するクラス数分作成する必要がある．式 (2.7) では，第 L 層のネットワークで K クラ

スを識別する時，出力層の各ノード k(=1,2,3,...,K) の入力 u_k に対する出力 y_k を求めている．また，出力 y_k は確率となり $\sum_{k=1}^{K} y_k = 1$ となる．

$$y_k = \frac{\exp\left(u_k^{(L)}\right)}{\sum_{j=1}^{K} \exp\left(u_k^{(L)}\right)} \tag{2.7}$$

2.2 勾配降下法

学習では，識別データと正解データの誤差関数を最小にするように重みを調節する．勾配降下法は，この重みを調節する際に使用される手法である．重みを w とおくと，誤差関数は $E(w)$ と考えることができる．$E(w)$ は凸関数で表現され，w の値を変化させることで $E(w)$ が最小となる最適解 w を求める．勾配降下法では，∇E を用いて式 (2.8) のように反復計算をして重み w を求める．t は反復回数，ε は学習係数を表す．

$$w^{(t+1)} = w^{(t)} - \varepsilon \nabla E \tag{2.8}$$

式 (2.8) から，定数である学習係数 ε により $w^{(t+1)}$ が大きく変化することがわかる．このため ε の値が学習において重要となる．ε が極端に小さいと，$\varepsilon \nabla E$ が小さくなり w の変化量が小さくなる．このため最適解を求めるための更新回数 t が増え，計算時間が増大する．また，誤差関数 E の形状により，局所解に陥り最適解を導出できない可能性がある．

この問題の解決に，確率的勾配降下法がある．式 (2.8) では，全訓練サンプル $n = 1, 2, 3, ..., N$ に対する誤差関数 E を用いたが，確率的勾配降下法では，反復回数毎に全訓練サンプル内の 1 つの $E_n(w)$ を選び計算を行う．

$$E(w) = \sum_{n=1}^{N} E_n(w) \tag{2.9}$$

$$w^{(t+1)} = w^{(t)} - \varepsilon \nabla E_n \tag{2.10}$$

この手法により，学習時間の短縮が実現される他，反復回数毎に誤差関数 $E_n(w)$ を変えるため，局所解に陥る可能性を低減させた．

また，規模が大きいニューラルネットワークでは並列計算機を使用するため，ミニバッチを利用する学習が多い．ミニバッチとは，ある少数の訓練サンプルの誤差関数を 1 つの集団として考え，計算を行う手法である．あるミニバッチを D_t と，サンプル数を $N_t = |D_t|$ とおくと，ミニバッチを含んだ誤差関数 $E_t(w)$ は式 (2.11) のように計算できる．

$$E_t(w) = \frac{1}{N_t} \sum_{n \in D_t} E_n(w) \tag{2.11}$$

それに加えて，過学習や過適合の可能性を低減し高速に学習を行うために，確率的勾配降下法 (2.10) を応用した以下の更新式 (2.12) が用いられている．λ_w は重み減衰 (weight decay)，μ はモメンタム (momentum) を示す．

$$w^{(t+1)} = w^{(t)} - \varepsilon(\nabla E_t(w) + \lambda_w w^{(t)}) + \mu(w^{(t-1)} - w^{(t-2)}) \tag{2.12}$$

2.3 誤差逆伝播法

ニューラルネットワークの学習では，前節の勾配降下法を実行する際，重みによる誤差関数の微分を計算する必要がある．誤差逆伝播法 (back propagation) は，この微分の計算を効率的に行う手法である．誤差関数 E_n を第 l 層の重み $w_{ji}^{(l)}$ で微分した勾配を $\frac{\partial E_n}{\partial w_{ji}^{(l)}}$ とおく．

多クラス分類 (クラス数 K) の誤差関数では，正解データ d_k と識別データ y_k を用いて，式 (2.13) のような交差エントロピーで表現される．

$$E_n = -\sum_{k=1}^{K} d_k \log y_k \tag{2.13}$$

このため，通常の勾配 $\frac{\partial E_n}{\partial w_{ji}^{(l)}}$ の計算は，式 (2.14) となる．

$$\frac{\partial E_n}{\partial w_{ji}^{(l)}} = -\sum_{k} d_k \frac{1}{y_k} \frac{\partial y_k}{\partial w_{ji}^{(l)}} \tag{2.14}$$

y_k は L 層のネットワークで表現すると，出力 $z^{(L)}$ と重み $w^{(L)}$，バイアス $b^{(L)}$ を用いて式 (2.15) の様に計算される．第 L 層の入力は前層の第 $L-1$ 層の出力と同様で，前層の出力を用いて表すことができる．また，第 $L-1$ 層の入力も第 $L-2$ 層の出力を用いて表すことができる．このため，勾配の計算は第 L 層から第 1 層まで繰り返し計算する必要があるため，式 (2.15) の様な入れ子の構造になる．したがって，ネットワークがより多層になることで計算時間が増大する．

$$\begin{aligned} y_k = f(u_k) &= f(w^{(L)} z^{(L-1)} + b^{(L)}) \\ &= f(w^{(L)}(f(w^{(L-1)} z^{(L-2)} + b^{(L-1)})) + b^{(L)}) \\ &= f(w^{(L)}(f(w^{(L-1)}(...f(w^{(l-1)} z^{(l-2)} + b^{(l)}...) + b^{(L-1)})) + b^{(L)}) \end{aligned} \tag{2.15}$$

誤差逆伝播法では，勾配の計算を式 (2.16) と考える．

$$\frac{\partial E_n}{\partial w_{ji}^{(l)}} = \frac{\partial E_n}{\partial u_j^{(l)}} \frac{\partial u_j^{(l)}}{\partial w_{ji}^{(l)}} \tag{2.16}$$

l 層の入力 $u_j^{(l)}$ が E_n に影響を与える際，次の $l+1$ 層を経由している．そのため式 (2.16) の右辺の第 1 項は，各層に対して微分の連鎖律を考慮すると式 (2.17) のようになる．

$$\frac{\partial E_n}{\partial u_j^{(l)}} = \sum_{k} \frac{\partial E_n}{\partial u_k^{(l+1)}} \frac{\partial u_k^{(l+1)}}{\partial u_j^{(l)}} \tag{2.17}$$

$u_k^{(l+1)} = \sum_j w_{kj}^{(l+1)} f(u_j^{(l)})$ より，$\frac{\partial u_k^{(l+1)}}{\partial u_j^{(l)}} = w_{kj}^{(l+1)} f'(u_j^{(l)})$ となる．また，$\delta_j^{(l)} \equiv \frac{\partial E_n}{\partial u_k^{(l)}}$ とおき，式 (2.17) をまとめた式を式 (2.18) に示す．

$$\delta_j^{(l)} = f'(u_j^{(l)}) \sum_k \delta_j^{(l+1)} w_{kj}^{(l+1)} \tag{2.18}$$

式 (2.18) より，$\delta_j^{(l)}$ は上位の $\delta_j^{(l+1)}$ を求めることで計算ができる．このため，出力層の δ を求めることができれば，繰り返し式 (2.18) を計算することで任意の層の δ を計算することが可能である．したがって，出力層 $\rightarrow$ 入力層 の順で計算を行うため，誤差逆伝播法と呼ばれている．

式 (2.16) の右辺の第 2 項は $u_j^{(l)} = \sum_i w_{ji}^{(l)} z_i^{(l-1)}$ より，$\frac{\partial u_j^{(l)}}{\partial w_{ji}^{(l)}} = z_i^{(l-1)}$ と計算できる．よって，誤差関数の勾配の式 (2.16) は式 (2.19) のように書き直すことができる．

$$\frac{\partial E_n}{\partial w_{ji}^{(l)}} = \delta_j^{(l)} z_i^{(l-1)} \tag{2.19}$$

最後に誤差逆伝播法の一連の流れについて，説明する．

1. 訓練サンプルについて，順伝播を行いノードの入力と出力を計算する．
2. 出力層の δ を求める．
3. 出力層の δ から式 (2.18) を用いて， 逆伝播を行い各層の δ を求める．
4. 式 (2.19) より，誤差関数の勾配を求める．

以上より，誤差関数が変化した場合も，出力層の δ のみを計算すればよい．ここで，式 (2.13) の多クラス分類 (クラス数 $= K$) の際に使用される関数に誤差逆伝播法を用いて勾配を計算する．式 (2.13) の y_k はソフトマックス関数を用いて，以下のように示す．

$$E_n = -\sum_{k=1}^{K} d_k \log y_k = -\sum_{k=1}^{K} d_k \log \left(\frac{\exp(u_k^{(L)})}{\sum_{i=1}^{K} \exp(u_i^{(L)})} \right) \tag{2.20}$$

出力層 L の $\delta = \frac{\partial E_n}{\partial u_j^{(L)}}$ は，式 (2.16) を参考に式 (2.22) のように書ける．

$$\delta_j^{(L)} = -\sum_{k=1}^{K} d_k \frac{1}{y_k} \frac{\partial y_k}{\partial u_j^{(L)}} \tag{2.21}$$

$$\frac{\partial y_k}{\partial u_j^{(L)}} = \frac{\partial}{\partial u_j^{(L)}} \frac{exp(u_k^{(L)})}{\sum_{i=1}^{K} exp(u_i^{(L)})} \tag{2.22}$$

$k = j$ の時は,

$$\begin{aligned}\frac{\partial y_k}{\partial u_j^{(L)}} &= \frac{\partial}{\partial u_j^{(L)}} \frac{exp(u_j^{(L)})}{\sum_{i=1}^{K} exp(u_i^{(L)})} \\ &= \frac{exp(u_j^{(L)}) \sum_{i=1}^{K} exp(u_i^{(L)}) - exp(u_j^{(L)}) \frac{\partial}{\partial u_j^{(L)}} \left(\sum_{i=1}^{K} exp(u_i^{(L)}) \right)}{\left(\sum_{i=1}^{K} exp(u_i^{(L)}) \right)^2} \\ &= \frac{exp(u_j^{(L)}) \sum_{i=1}^{K} exp(u_i^{(L)}) - exp(u_j^{(L)}) exp(u_j^{(L)})}{\left(\sum_{i=1}^{K} exp(u_i^{(L)}) \right)^2} \\ &= \frac{exp(u_j^{(L)})}{\sum_{i=1}^{K} exp(u_i^{(L)})} - y_k^2 = y_k - y_k^2 = y_k(1 - y_k) \end{aligned} \tag{2.23}$$

$k \neq j$ の時は,

$$\begin{aligned}\frac{\partial y_k}{\partial u_j^{(L)}} &= \frac{\partial}{\partial u_j^{(L)}} \frac{exp(u_k^{(L)})}{\sum_{i=1}^{K} exp(u_i^{(L)})} = - \frac{exp(u_k^{(L)}) \frac{\partial}{\partial u_j^{(L)}} \sum_{i=1}^{K} exp(u_i^{(L)})}{\left(\sum_{i=1}^{K} exp(u_i^{(L)}) \right)^2} \\ &= - \frac{exp(u_k^{(L)}) exp(u_j^{(L)})}{\left(\sum_{i=1}^{K} exp(u_i^{(L)}) \right)^2} = y_k y_j \end{aligned} \tag{2.24}$$

したがって，勾配は式 (2.25) のようになる．ここで，$\sum_k^K d_k = 1$ を用いた．

$$\begin{aligned}\frac{\partial E_n}{\partial u_j^{(L)}} &= - \sum_{k=1}^{K} d_k \frac{1}{y_k} \frac{\partial y_k}{\partial u_j^{(L)}} = -d_j(1 - y_j) + \sum_{k \neq j}^{K} d_k y_j \\ &= \sum_{k}^{K} d_k (y_j - d_j) = y_j - d_j \end{aligned} \tag{2.25}$$

このように，前頁に示した誤差関数の勾配 δ は，識別データと正解データの差で表現できる．また，様々な問題の誤差関数 E_n に対して，この誤差逆伝播法が応用されている．

2.4 Fully Convolutional Network

FCN(Fully Convolutional Network)[6] は，CNN の全結合層を畳込み層に変換したネットワークである．このネットワークに逆畳込みを適用することで，画像内の物体の位置や形状を特定することができ，領域セグメンテーションの問題に CNN を応用することが可能となった．また，FCN は以下の図 2.4 の様に画像を出力するため，従来のクラス分類のネットワークと違い出力層の出力は画像サイズ × クラス数となる．学習は，CNN と同様の方法を用いて誤差が最小となる様に行う．以下で，逆畳込み処理について述べる．

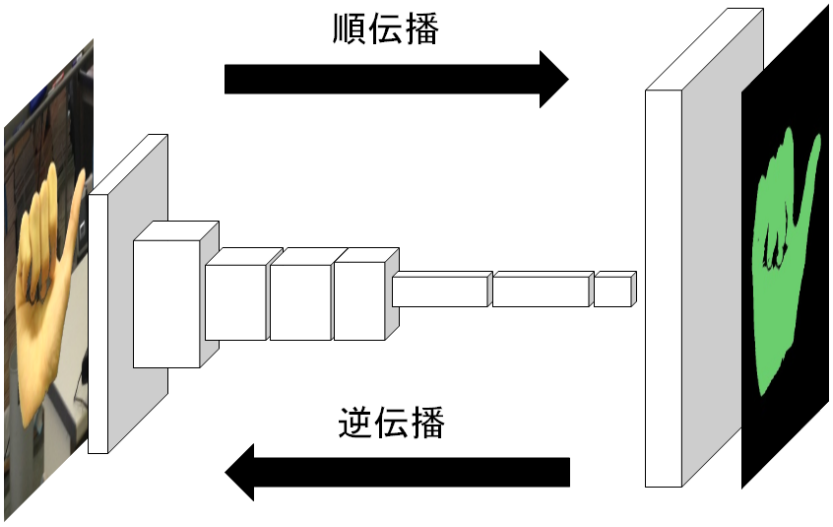


図 2.4　Fully Convolutional Network

2.4.1 逆畳込み処理

逆畳込みの処理は，ニューラルネットワークでは画像を出力する際に利用される．計算方法は，畳込みの逆伝播で用いられる計算を利用する．図 2.5 に逆畳込みの処理の例を示す．入力の各画素に対してフィルタの各画素の積を出力値に代入し，出力のインデックスが重なった部分は出力値の和を求める．これらの図では，青色で示した入力 2×2 に対して，灰色で示した 5×5 のフィルタを用いて，6×6 の出力を生成している．図 2.6 は，入力の各画素（淡青色で示した画素）に注目した際の逆畳込み処理を示し，各画素に対する出力を淡緑色で表している．

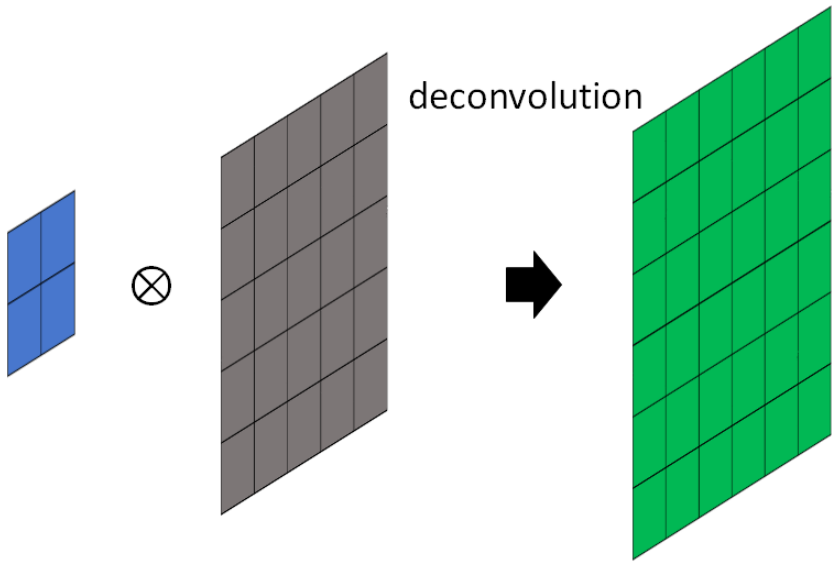


図 2.5　逆畳込み処理

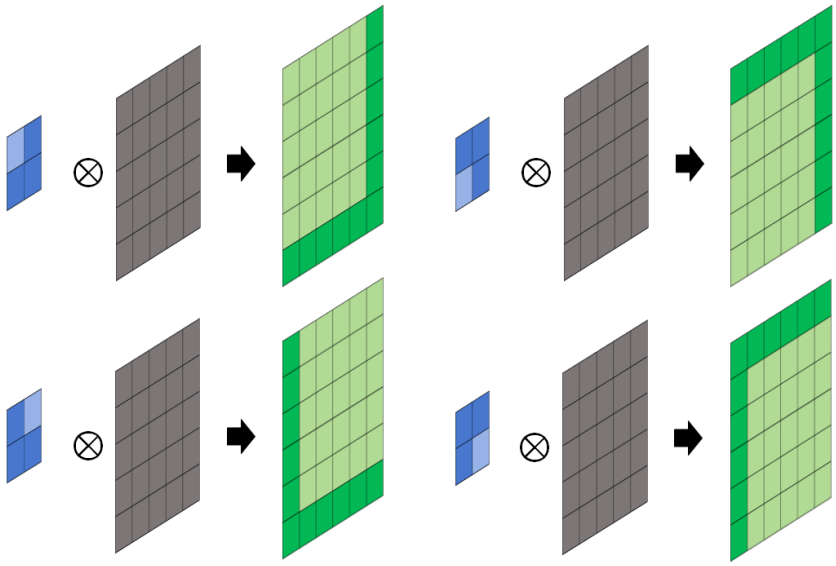


図 2.6　各画素に対する逆畳込み

出力サイズ $O \times O$ は式 (2.26) で計算できる．ここで，入力サイズを $I \times I$，ストライドを s，フィルタサイズを $W \times W$，パッチを p とする．式 (2.26) より，ストライド s の値が大きいほど出力サイズは大きくなる．図では，$O = 6$, $I = 2$, $s = 1$, $W = 5$, $p = 0$ となる．

$$O = s(I - 1) + W - 2p \tag{2.26}$$

第 3 章
提案手法

本研究では，認識精度の向上を目的として，近傍画素を考慮した誤差関数を利用した指文字認識を提案する．本章では，提案手法のアルゴリズムについて述べる．関連研究で述べた様に，領域セグメンテーションにおいて，注目画素に加えて近傍画素の情報も分割する際には重要な要素となる．しかし，従来の CNN を用いた領域セグメンテーションでは，多クラス分類と同様の誤差関数を用いてるため，注目画素に対してのみの精度を出力する．そのため，近傍画素の情報が考慮されておらず，背景と前景の境界の認識が難しいことが考えられる．そこで，指文字認識の精度を向上するために，近傍画素を考慮した誤差関数をネットワークに導入することを提案する．本研究では，近傍画素を考慮した誤差関数を利用した指文字認識を行う．以下で提案手法について詳細を述べる．CNN を用いた学習では，特徴量抽出と識別を同時に両方行うことが可能であるため，ここでは学習する画像と提案した誤差関数，学習手法について述べる．

3.1 学習画像

はじめに，学習画像を生成する．領域セグメンテーションを行うため，学習に背景を含んだ指文字画像を用いる．CNN の学習に，背景が黒色の指文字画像に擬似的な背景を合成した画像を入力とし利用する．以下に擬似的な背景を合成する手順を示す（図 3.1）．そのため，背景には物体の影がない画像を用いる．その後，画像をクリッピングし適切な大きさに切り出し，合成画像から平均画像の差分をとった正規化画像で学習を行う（図 3.2）．平均画像は全データに対する画素の平均から作成した．

図 3.1　擬似背景の合成手順

図 3.2　平均画像による正規化処理

3.2　近傍画素を考慮した誤差関数の導入

近傍画素を考慮した誤差関数を提案する．近傍画素を考慮した関数にグラフカットで用いられているエネルギー関数がある．エネルギー関数の主な構造を式 (3.1) に示す．

$$E = \sum_{p \in P} g_p(L_p) + \sum_{(p,q) \in N} h(p,q)\delta(L_p, L_q) \tag{3.1}$$

入力画像 P に対する画素を p とおき，画素 p のラベル値を L_p とおく．q は p に隣接している画素を示す．ここで，第 1 項をデータ項，第 2 項を平滑化項と呼ぶ．第 2 項の平滑化項を加えることで，注目画素の近傍領域を考慮することが可能となる．したがって，式 (3.1) を参考に近傍画素を考慮した誤差関数を提案する．

以下に，誤差逆伝播法で述べた従来の誤差関数と本研究で提案した誤差関数を示す．

1. 従来の誤差関数

$$E_n = -\sum_{k=1}^{K} d_k \log y_k \tag{3.2}$$

2. 提案した誤差関数

$$E_n = -\sum_{k=1}^{K} d_k \left(\log y_k - \lambda \sum_{q=1}^{Q} \exp\left(-\frac{(u_k - u_q)^2}{2\sigma^2} \right) \delta(l_k, l_q) \right) \tag{3.3}$$

従来の注目画素のみの誤差関数 (3.2) と比較して，提案した誤差関数 (3.3) では，式 (3.1) で用いられている様な平滑化項を導入した．データ項では，従来と同様にソフトマックス関数 $y_k \left(= \frac{\exp(u_k^{(L)})}{\sum_{j=1}^{K} \exp(u_k^{(L)})} \right)$ を用いた関数を利用する．平滑化項では，Q の近傍数に対して注目画素の尤度 u_k と近傍画素の尤度 u_q の差を利用した関数を用いる．λ は平滑化項がデータ項に与える影響を調節する定数，σ は $(u_k - u_q)$ の差に平滑度を加える定数を示す．そして，δ はクロネッカーのデルタを表し，注目画素のラベル値 l_k と近傍画素のラベル値 l_q に対して以下のように出力する

関数を示す．

$$\delta(l_k, l_q) = \begin{cases} 0 & (l_k = l_q) \\ 1 & (l_k \neq l_q) \end{cases} \tag{3.4}$$

また，CNN の学習では誤差逆伝播法を利用するため，誤差関数 E が入力 u_j に対して微分可能である必要がある．そこで以下では，提案した誤差関数の微分の計算を示す．

$$\frac{\partial E_n}{\partial u_j^{(L)}} = -\frac{\partial}{\partial u_j^{(L)}} \sum_{k=1}^{K} d_k \log y_k + \frac{\partial}{\partial u_j^{(L)}} \lambda \sum_{k=1}^{K} \sum_{q=1}^{Q} d_k \exp\left(-\frac{(u_k - u_q)^2}{2\sigma^2}\right) \delta(l_k, l_q) \tag{3.5}$$

第 1 項の微分は，誤差逆伝播法の式 (3.1) より，

$$-\frac{\partial}{\partial u_j^{(L)}} \sum_{k=1}^{K} d_k \log y_k = y_j - d_j \tag{3.6}$$

となる．また，第 2 項の微分は，$\sum_{k=1}^{K} d_k = 1$ を用いて，

$$\frac{\partial}{\partial u_j^{(L)}} \lambda \sum_{k=1}^{K} \sum_{q=1}^{Q} d_k \exp\left(-\frac{(u_k - u_q)^2}{2\sigma^2}\right) \delta(l_k, l_q) \simeq \frac{\lambda}{\sigma^2} \sum_{k=1}^{K} \sum_{q=1}^{Q} (u_k - u_q) \exp\left(-\frac{(u_k - u_q)^2}{2\sigma^2}\right) \delta(l_k, l_q) \tag{3.7}$$

となる．よって，誤差関数の微分は式 (3.8) の様に計算できる．

$$\frac{\partial E_n}{\partial u_j^{(L)}} = y_j - d_j - \frac{\lambda}{\sigma^2} \sum_{k=1}^{K} \sum_{q=1}^{Q} (u_k - u_q) \exp\left(-\frac{(u_k - u_q)^2}{2\sigma^2}\right) \delta(l_k, l_q) \tag{3.8}$$

また，式 (3.5) の第 2 項の微分は，以下の 3 つの式から成る．ここで，k は注目画素，q は近傍画素を示すため，$k \neq q$ が成り立つ．

$$\begin{cases} \lambda \sum_{q \neq j}^{Q} d_j \left(-\frac{1}{\sigma^2}(u_j - u_q)\right) \exp\left(-\frac{(u_j - u_q)^2}{2\sigma^2}\right) \delta(l_j, l_q) & (k = j,\ q \neq j) \\ \lambda \sum_{k \neq j}^{K} d_k \left(\frac{1}{\sigma^2}(u_k - u_j)\right) \exp\left(-\frac{(u_k - u_j)^2}{2\sigma^2}\right) \delta(l_k, l_j) & (k \neq j,\ q = j) \\ 0 & (k \neq j,\ q \neq j) \end{cases} \tag{3.9}$$

したがって，式 (3.8) に提案した誤差関数の勾配を示す．また，この式が出力層の逆伝播する際の式となる．以上より，近傍画素を考慮した平滑化項を誤差関数に導入し，学習・評価を行う．近傍領域 Q と，λ，σ は学習する画像により調節が可能である．これらのパラメータについては，以下の節で述べる．平滑化項を導入したことにより，誤差関数 E_n 自身の大きさも平滑化項の分だけ大きくなることが考えられる．

3.3 学習手法について

学習では，事前に学習した学習済みモデルを使用する．学習済みモデルを使用する利点として，多くの画像を事前に学習しているため，似たような特徴量を含む画像を学習する際には学習が速く進む可能性がある．また，パラメータがある程度更新されているため，始めから学習する時と比べ学習が収束しやすい事が挙げられる．このような理由から，本研究では事前に学習したモデルのパラメータを使用した実験を行うことを提案する．しかしながら，欠点としてネットワークの構成がある程度等しくないと，学習済みモデルのパラメータ全てを利用することができない．また，事前に学習した画像に依存するため，最適解が得られない可能性が考えられる．従来のクラス分類のネットワークでは，出力層のノード数を分類したいクラス数に変更するだけで，学習が行うことが可能であったが，領域セグメンテーションの問題では，出力層の前層に位置する逆畳込み層にもクラス数の影響を考慮する必要がある．そのため，学習済みモデルよりクラス数が多い学習では，学習済みモデルを利用できない．

したがって，ネットワークの構成は FCN-AlexNet を参考にした以下のネットワークで学習を行う．conv は畳込み層，pool はプーリング層，norm は正規化層，deconv は逆畳込み層を示す．また，提案した誤差関数の平滑化項のパラメータを $\lambda = 1, 10, 100$，$\sigma = 1$ とし，4 近傍 $(Q = 4)$ の場合で学習を行い，提案した誤差関数と従来の誤差関数を用いた場合の精度を比較する．

表 3.1　FCN 層構造

名称	パッチ	ストライド	出力マップサイズ	関数
input	-	-	$256 \times 256 \times 3$	-
conv1	11×11	4	$112 \times 112 \times 96$	-
pool1	3×3	2	$56 \times 56 \times 96$	MAX
norm1	5×5	1	$56 \times 56 \times 96$	LRN
conv2	5×5	1	$56 \times 56 \times 256$	-
pool2	3×3	2	$28 \times 28 \times 256$	MAX
norm2	5×5	1	$28 \times 28 \times 256$	LRN
conv3	3×3	1	$28 \times 28 \times 384$	-
conv4	3×3	1	$28 \times 28 \times 384$	-
conv5	3×3	1	$28 \times 28 \times 256$	-
pool5	3×3	2	$14 \times 14 \times 256$	MAX
conv6	6×6	1	$9 \times 9 \times 4096$	-
conv7	1×1	1	$9 \times 9 \times 4096$	-
conv8	1×1	1	$9 \times 9 \times 21$	-
deconv8	63×63	32	$319 \times 319 \times 21$	-
output	-	-	$256 \times 256 \times 21$	softmax

第 4 章
実験

前章では，提案手法について説明した．本章では，提案手法の有効性を確かめるために行った認識実験について述べる．まず，実験データについて説明し，後に認識実験について述べる．

4.1 実験データ

認識対象として，静止指文字の 20 種類 (あ〜と) を扱う（図 4.1）．本研究では，１枚の画像から認識することが可能であるものを対象にしたため，動きを含む「の」,「も」,「り」,「を」,「ん」，濁音，半濁音の文字は対象にしない．実験データは被験者 8 人から得たものを利用する．１つの指文字に対して 3 パターンの画像がある．さらに角度の違いを考慮するために $\pm 5°$，$\pm 10°$ の回転を加えたものを作成した．したがって，1 つの指文字に対して 120 枚存在し，全体で $2400(= 120 \times 20)$ 枚の指文字画像を利用する．

さらに，2 つの擬似的な背景（図 4.2，図 4.3）を利用して，1 つの指文字を 240 枚にし，全データを $4800(= 240 \times 20)$ 枚にした．従来手法である CNN による領域セグメンテーションの研究を参考に画像サイズは 256×256 に設定した．

図 4.1　実験画像

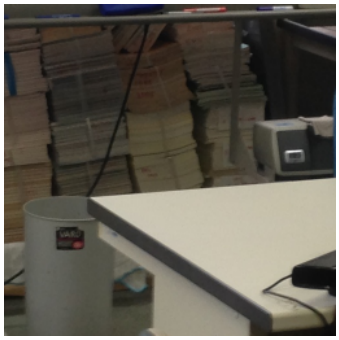


図 4.2　背景画像 1

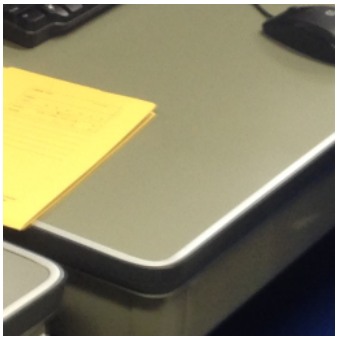


図 4.3　背景画像 2

また，評価に用いるための指文字のクラス名とラベル値を以下に示す．教師 (ラベル) 画像はこの表のクラス値を代入し，作成する．教師画像に配色を施した画像を図 4.4 に示す．

表 4.1　クラス名とラベル値

ラベル値	0	1	2	3	4	5	6	7	8	9	10
クラス名	背景	あ	い	う	え	お	か	き	く	け	こ
ラベル値	11	12	13	14	15	16	17	18	19	20	
クラス名	さ	し	す	せ	そ	た	ち	つ	て	と	

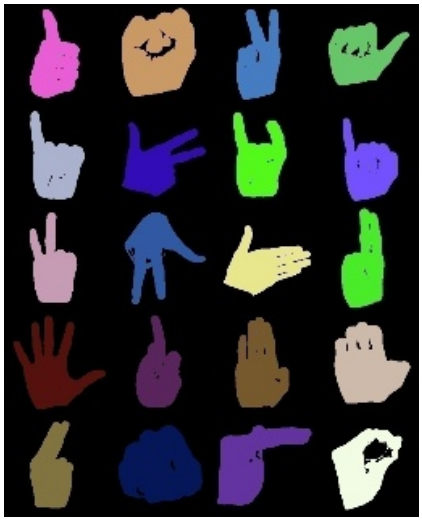


図 4.4　教師 (ラベル) 画像

4.2 実験概要

実験に用いたパラメータを以下に示す．また交差検証法により，データを図 4.5 の様に訓練用と検証用の画像に分割した．この実験では，全データを 5 個のデータに分割した．したがって，1 個のデータの訓練用画像枚数は 3840 枚，検証用画像枚数は 960 枚になる．

表 4.2　実験パラメータ

学習回数	10000
weight decay	0.0005
momentum	0.9
learning rate	0.0001
gamma	0.1
batch_size	20
λ	1，10，100
σ	1

図 4.5　交差検証法

4.3 評価方法

実験で求められた CNN の出力に対して，以下の 3 つの関数を用いて評価を行う．

1. *Accuracy*

   1 枚の画像に対する平均正解率を示す．背景クラスを考慮した全クラスで以下の式を用いて計算する．

$$Accuracy = \frac{\text{正解数}}{\text{画像サイズ}} \tag{4.1}$$

2. *IU*(Interaction Over Union)

   k クラスに対しての重なり度を示す．L_k は正解データを示し，P_k は識別データを示す．*IU* は図 4.6 の黄色の部分を示す．背景クラスを考慮せず，指文字のクラスのみで計算する．

$$IU = \frac{L_k \cap P_k}{L_k \cup P_k} \tag{4.2}$$

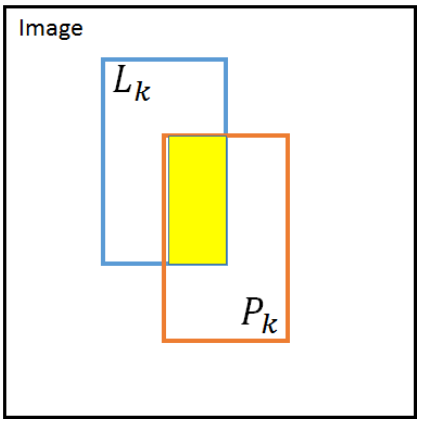


図 4.6　重なり度合い

3. 適合率 ($precision$) と再現率 ($recall$)，F 値 ($F\text{-}measure$)

以下の表から適合率，再現率の計算を行う．適合率は，システムが出した結果の中から，正しい割合を表すので，正確性に関する指標を示す．また再現率は，正解として出力されるべきものの中から，実際に正解として出力される割合を表すので，網羅性に関する指標を示す．この場合も IU と同様に背景クラスを考慮せず，指文字クラスのみで計算する．F 値は適合率と再現率の調和平均を表す．

表 4.3　tp，fp，fn，tn の定義

		正解結果	
		正	負
予測結果	正	tp	fp
	負	fn	tn

$$precision = \frac{tp}{tp + fp} \tag{4.3}$$

$$recall = \frac{tp}{tp + fn} \tag{4.4}$$

$$F\text{-}measure = \frac{2recall \times precision}{recall + precision} \tag{4.5}$$

4.4 実験結果

実験による結果を述べる．交差検証法によるそれぞれの結果と平均の精度を計算した．従来手法による結果を表 4.4 に示す．事前学習のモデルを使用したことにより，高い精度を出すことができている．精度に対しては 5 つの識別器は同じ程度の精度を出力した．その中でも識別器 5 が一番高い精度を示し，識別器 3 が一番精度が低い結果となった．

表 4.4　従来手法の実験結果 (20 文字)

評価関数	従来 1	従来 2	従来 3	従来 4	従来 5	平均
Accuracy	0.6429	0.6432	0.6427	0.6453	0.6457	0.6440
IU	0.8711	0.8738	0.8733	0.8758	0.8767	0.8741
precision	0.8868	0.8889	0.8878	0.8943	0.8952	0.8906
recall	0.9797	0.9805	0.9813	0.9766	0.9767	0.9789
F-measure	0.9304	0.9319	0.9317	0.9331	0.9336	0.9321

以下で提案手法についての結果を述べる．表は順に $\lambda = 1, 10, 100$ の時を示す．$\lambda = 1$ の時以外では，従来手法と比較して正解率と適合率が向上した．また，表から λ の値が大きい程精度が向上していることがわかる．これは，λ が増加するほど平滑化項の影響が大きくなり，誤差関数に良い影響を与えていると考えられる．

表 4.5　提案手法の実験結果 ($\lambda = 1$)

評価関数	提案 1	提案 2	提案 3	提案 4	提案 5	平均
Accuracy	0.6421	0.6431	0.6423	0.6451	0.6456	0.6436
IU	0.8701	0.8736	0.8727	0.8758	0.8768	0.8738
precision	0.8849	0.8885	0.8868	0.8940	0.8951	0.8899
recall	0.9808	0.9808	0.9817	0.9769	0.9769	0.9794
F-measure	0.9298	0.9319	0.9313	0.9331	0.9336	0.9319

表 4.6　提案手法の実験結果 ($\lambda = 10$)

評価関数	提案 1	提案 2	提案 3	提案 4	提案 5	平均
Accuracy	0.6430	0.6431	0.6434	0.6455	0.6454	0.6441
IU	0.8714	0.8738	0.8744	0.8762	0.8764	0.8744
precision	0.8872	0.8889	0.8897	0.8952	0.8947	0.8911
recall	0.9797	0.9805	0.9803	0.9760	0.9768	0.9787
F-measure	0.9306	0.9319	0.9323	0.9333	0.9334	0.9323

表 4.7　提案手法の実験結果 ($\lambda = 100$)

評価関数	提案 1	提案 2	提案 3	提案 4	提案 5	平均
Accuray	0.6582	0.6587	0.6589	0.6604	0.6603	0.6593
IU	0.8717	0.8720	0.8728	0.8681	0.8711	0.8711
precision	0.9260	0.9287	0.9291	0.9331	0.9331	0.9300
recall	0.9367	0.9344	0.9349	0.9254	0.9288	0.9320
F-measure	0.9308	0.9310	0.9314	0.9287	0.9304	0.9305

また以下の図 4.7〜4.10 で，各カテゴリの精度を示す．ここでは，従来手法と提案した手法の中で精度が大きく向上した $\lambda = 100$ の時を比較する．

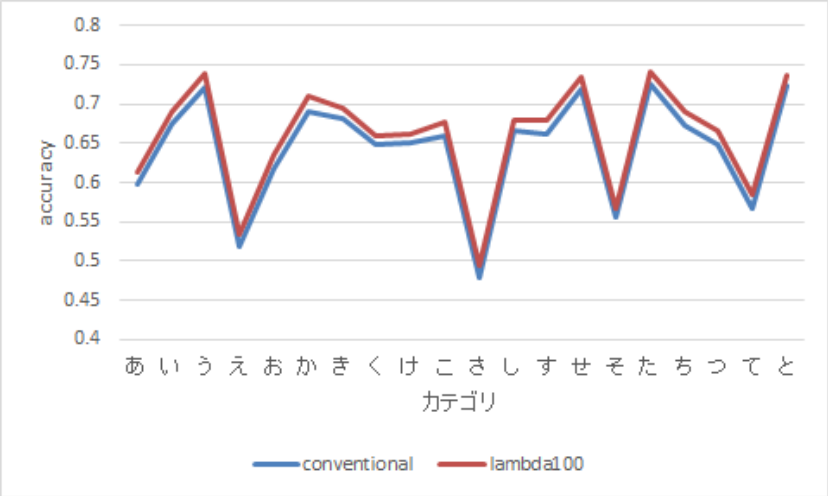


図 4.7　各カテゴリの正解率

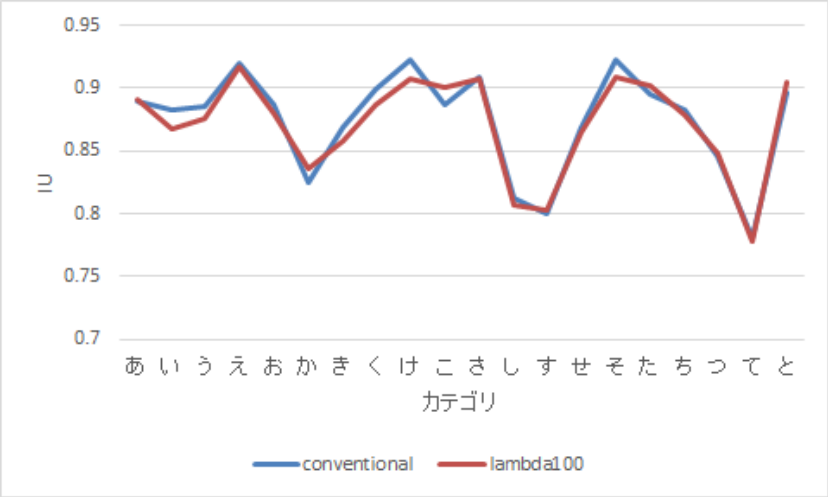


図 4.8　各カテゴリの重なり度

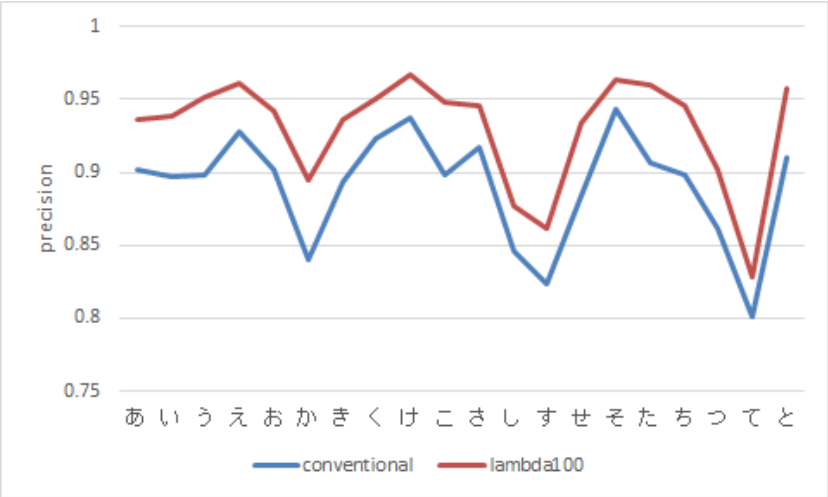


図 4.9　各カテゴリの適合率

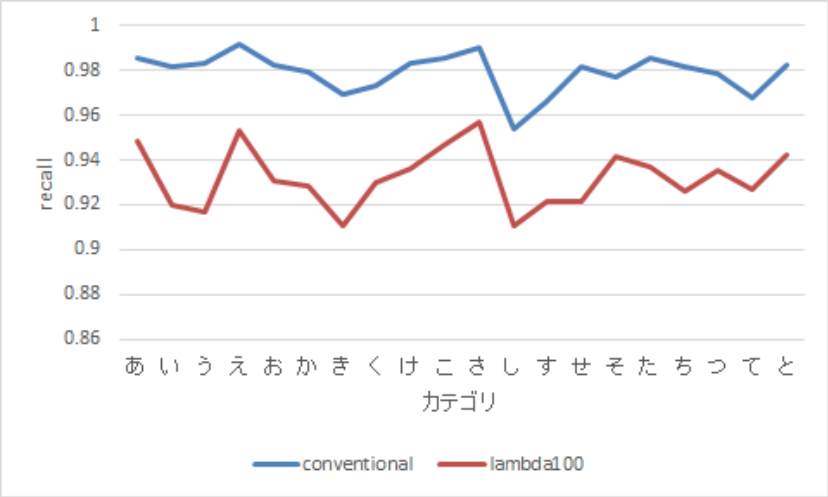


図 4.10　各カテゴリの再現率

以上の図より，正解率と適合率は全てのカテゴリで従来手法より提案手法の方が精度が高いことがわかる．また，重なり度に関しては精度向上が少なく，再現率では全てのカテゴリで精度が低下したことがわかる．

従来手法と提案手法の各々の誤差関数の推移を図 4.11 に示す．この図から平滑化項を導入したことにより，誤差が増加していることがわかる．ただし，この図では平滑化項のスケールがそれぞれ違うため，誤差が大きいから精度が低いということにはならない．

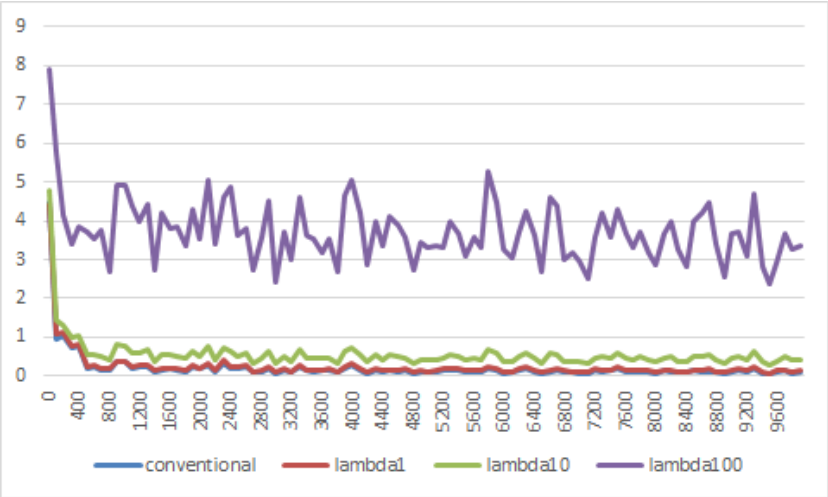


図 4.11　平均誤差の推移

4.5 考察

実験結果に対する考察を述べる．平滑化項を導入した誤差関数を用いた実験では，従来の誤差関数を用いた結果と比較して $\lambda = 1$ 以外の値で精度を向上させることができた．特に，$Accuracy$ と $precision$ では λ の値を大きくするほど精度が大きく向上した．しかしながら，$recall$，IU または $F\text{-}measure$ では，λ の値を大きくすることが必ずしも精度向上に繋がるとは言えないことがわかる．また，従来の誤差関数では粗い画像を出力していたため $recall$ が高い値を示していたが，提案手法では $recall$ の値を抑え，$precision$ の値を向上させることができた．これより，従来と比較して出力画像の粗さを抑えることが可能になったと言える．したがって，学習画像と評価方法により精度には多少の差はあるが，誤差関数に平滑化項を導入することは有効な手段であると考えられる．

また，λ を大きくすることで精度が向上した理由に，平滑化項の値がデータ項と比較して微小であったため，λ の値を大きくすることで平滑化項が及ぼす誤差関数への影響が大きく変化したためであると考えられる．実験結果より，平滑化項のスケールが大きいほど精度に影響があることから，誤差関数には近傍画素の情報が精度を向上させる上では必要であることがわかる．しかし，$\lambda = 100$ では誤差関数のスケールが大きく変化するのに対して $Accuracy$，IU では極端な精度の変化が見られなかった．これは，学習係数 ε が 0.0001 と小さいため，∇E の変化に対しても w に大きな影響がなかったと考えられる．

従来手法と提案手法を用いた識別画像を以下の表 4.8，表 4.9 に示す．表から手の平や甲，又は指同士が接するような指文字は抽出できているが，指同士が互いに離れて表現するような指文字は抽出が難しいことがわかる．これはネットワーク内のプーリング層で行っているダウンサンプリングが重要な情報を削除している可能性がある．また逆畳込み 1 層により急な拡張を行っているため，細かい指の表現を上手く抽出できていないと考えられる．解決策としてネットワーク構造の最適化が考えられる．本研究では，計算機の性能的な都合により実験できなかったが，例えば，段階的に拡張を行うために複数の逆畳込みを用いたり，より階層的な特徴量を得るために畳込みの層数を増やすことが挙げられる．

表 4.8　識別画像例 (あ行，か行)

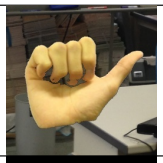
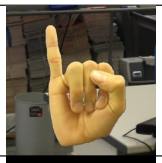
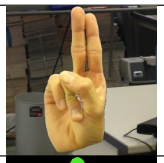
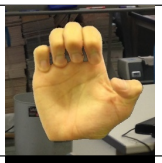
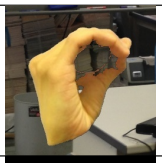
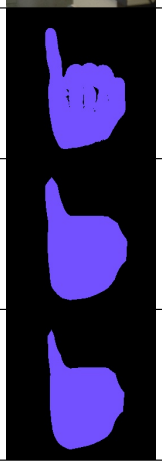
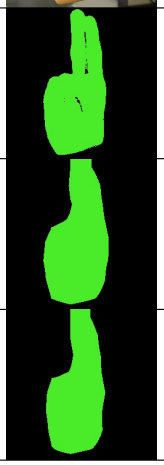
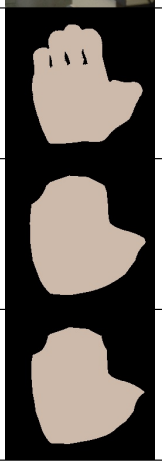
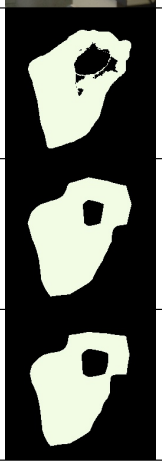
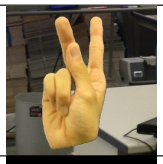
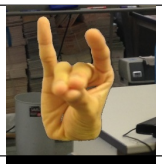
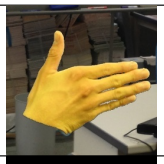
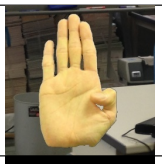
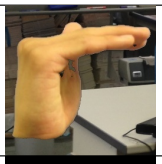
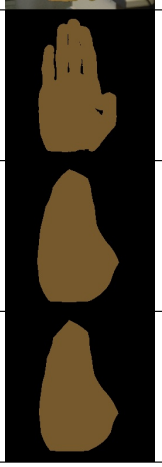
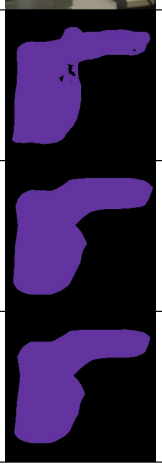
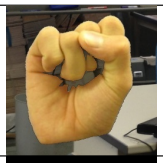
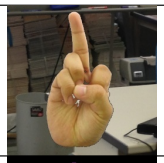
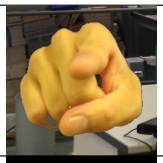
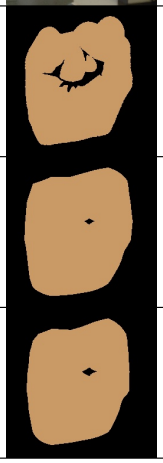
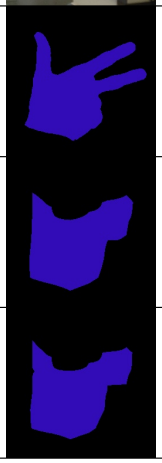
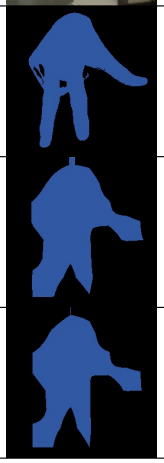
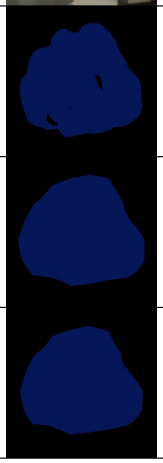
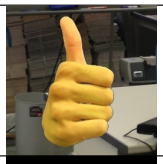
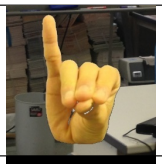
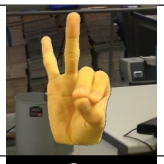
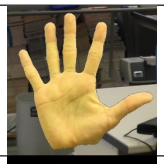
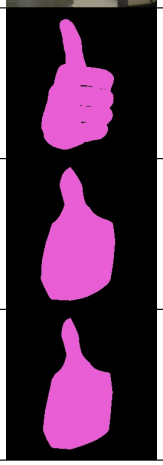
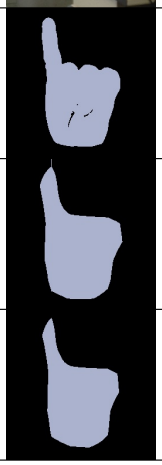
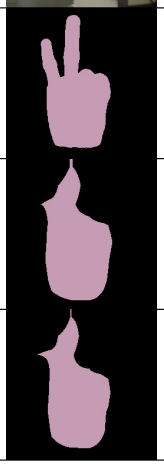
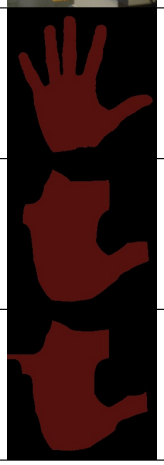
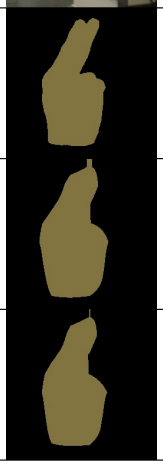
カテゴリ	あ	い	う	え	お
入力画像					
教師画像					
従来手法					
提案手法					
カテゴリ	か	き	く	け	こ
入力画像					
教師画像					
従来手法					
提案手法					

表 4.9　識別画像例 (さ行，た行)

カテゴリ	さ	し	す	せ	そ
入力画像					
教師画像					
従来手法					
提案手法					
カテゴリ	た	ち	つ	て	と
入力画像					
教師画像					
従来手法					
提案手法					

おわりに

本研究では，日本語指文字 20 文字に対して領域セグメンテーションの精度向上のために，誤差関数に平滑化項を導入した手法を提案した．また，実験的な評価を行い，従来手法と比べ精度を向上させることができた．今後の課題として，より精度を向上させるために，CNN のネットワークや学習パラメータの最適化が考えられる．ネットワークに関しては，畳込み層や逆畳込み層をより多層にすることで，段階的な拡張を行うことができ，滑らかな画像が生成できると考えられる．重要な情報の欠落を抑えるためには，プーリング層のカーネルサイズを最適化することが挙げられる．学習パラメータについては，入力画像に対する学習係数，平滑化項の定数の決め方などが挙げられる．また，本研究では指文字が画像内の中央に位置し各指文字の大きさは同程度の画像を用いていたことから，今後は位置や大きさを変化させた画像を用いることで，より実用的な画像認識システムに近づくと考えられる．

謝辞

日ごろから多くの御指導を頂きました太田義勝教授，鈴木秀智准教授に深く感謝いたします．そして，日頃何かとお世話になりました落合美子事務員に感謝いたします．また，本論文作成にあたって特にお世話になりました鈴木秀智准教授に深く感謝いたします．最後に，日頃から熱心に討論して頂いた研究室の諸氏に感謝いたします．

参考文献

[1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, ”Gradient-based learning applied to document recognition”, Proc. of the IEEE, pages 2278-2324, 1998.

[2] Paul L. Rosin, ”Thresholding for Change Detection”, Com-puter Vision and Image Understanding, vol. 86, pp. 79-95, 2002.

[3] Yuri Boykov, and Marie-Pierre Jolly, ”Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Object in N-D Images”, in Proc, International Conf. an Computer Vision, 2001.

[4] 大野雄士朗: ”HMM による日本語指文字の動きを考慮した単語認識”, 修士論文, 平成 27 年度

[5] Andrew V. Goldberg and Robert E. Tarjan, ”A New Approach to the Maximum-Flow Problem”, Jouranal of the ACM No.35 pp.921-940, 1988.

[6] Rumelhart et al. ”Learning representations by back-propagating erroes”, Nature, VOL 323, 1986.

[7] J. Long et al. ”Fully convolutional networks for semantic segmentation.” In CVPR, 2015.

[8] Y.LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. ”Backpropagation applied to handwritten zip code recognition”, 1998.

[9] Y.LeCun, L. Bottou, G. Orr, and K. Muller. ”Efficient backprop”, 1998.

[10] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, ”ImageNet Classification with Deep Convolutional Neural Networks”, 2012.

[11] Karen Simonyan and Andrew Zisserman, ”Very deep convolutional networks for large-scale image recognition”, arXiv preprint arXiv:1409. 1556, 2014.

[12] Christian Szegedy et al. ”Going Deeper with Convolutions”, 2015.

[13] 菊田智也：”類似指文字を考慮した段階的指文字認識”, 修士論文, 平成 23 年度

[14] 岡谷貴之, ”機械学習プロフェッショナルシリーズ　深層学習”, 講談社, 2015

[15] Ross Girshick, et al. ”Rich feature hierarchies for accurate object detection and semantic segmentation”, Proceedings of the IEEE conference on computer vision and pattern recognition, 2014

[16] C. Farabet, C. Couprie, L. Najman, Y. LeCun, ”Learning hierarchical features for scene labeling”, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013

付録

1　作成したプログラムおよび実験データについて

home/kojima/caffe_workspace/result

学習したモデルが置かれているディレクトリ．

home/kojima/caffe_workspace/data

学習に使用するデータが置かれているディレクトリ．

home/kojima/workspace

検証するプログラムが置かれているディレクトリ．各プログラム．