

畳み込みネットワークによる仮名3文字のくずし字の文字認識

長井 歩 (群馬大学 理工学部)

仮名3文字のくずし字の文字列認識のため、近年注目を浴びている深層学習を応用した。提案法は大きく3つのネットワークから成る。1文字のくずし字の文字認識を行うネットワーク、文字と文字の境界かどうかを判断するネットワーク、さらに両ネットワークによって画像を特徴ベクトルとみなし、多数の特徴ベクトルから3文字の文字列を出力するネットワークを含む。1文字のくずし字の文字認識は92%前後と健闘したものの、3文字の文字認識では76%程度に留まった。

Recognizing Three Character String of Old Japanese Cursive by Convolutional Neural Networks

Ayumu Nagai (Department of Electronics and Informatics, Gunma University)

In order to recognize old Japanese cursive letters, we applied Deep Learning which is currently receiving great attention. Our method consists of three networks including a network to recognize a single cursive letter, to distinguish whether it is a letter or a boundary between letters, and to recognize a three character string after handling as feature vectors. The recognition rate of a single letter reached around 92%, while the rate of three character string remained in the result of 76%.

1. まえがき

「日本を今一度せんたくいたし申候」との坂本龍馬の手紙は、日本人にとってあまりにも有名である。しかし、活字ではなく龍馬の筆跡で読んだことのある人は殆どいないだろう。現代の日本人にとって、150年あまり前の生き生きとした坂本龍馬の手紙を読むよりも、英和辞典を片手に400年以上前のシェイクスピアの初版本を読む方が断然楽であろう。何故こんなことになってしまったのだろうか。その最大の理由は、現代人にはくずし字を読めないことであろう。

実は日本は古文書大国である。200年も前の古文書を誰でも神保町で手頃な価格で買えるような国は、世界広しと言えどなかなかない[1]と聞く。それなのに、くずし字が読めないから読まないのは、あまりにも悲しいことではないだろうか。

古文書を読めるようになると、様々な効用が期待できる。日本は地震、津波、火山の噴火など数々の災害を経験してきた。これらの災害の予防や予報のためには長年に渡る過去の事例の蓄積がものを言う。明治前期より前の記録はくずし字のままなので、これらの記録を掘り起こすにはくずし字を読む必要がある。

翻刻されていない古文書は9割以上[2]とも99%以上[1]とも言われる。古文書を読むには専門的な知識と訓練を要し、現状では圧倒的に人手が足りていない。この問題を解消すべく、計算機による自動的な古文書の翻刻を目標とし、まずはそ

の端緒として、本研究では仮名のくずし字の文字認識、および仮名3文字のくずし字の文字認識に取り組む。

文字認識には近年注目されている深層学習(Deep Learning)の技術を応用する。昨今は人工知能がブームになっているが、その中心的役割を担っているのが深層学習で、物体認識や音声認識を含む幅広い分野で抜群の成果を挙げている。文字認識においても、中国語の手書き文字や、日本語のひらがなの手書き文字の認識で95%を超える高い正解率を達成している。くずし字の認識においても強力な武器になることは間違いない。

本研究では、くずし字を読めない現代日本人がくずし字を読めるようになるための翻刻支援システムの実現を目指す。まずは46種の仮名文字1文字のくずし字を認識する識別器を学習する。次に仮名文字3文字のくずし字を認識する識別器を学習する。前者については92%前後の認識精度を達成し、後者については76%前後の認識精度となった。

2. 関連研究

文字認識の研究は深層学習の登場よりもずっと前から行われてきた。現代日本語の手書き文字認識に関する一連の研究はETL9[3]というデータベースの公開の影響もあり、1990年代に盛んに研究され、1997年には認識率99.31%[4]にも達した。処理の大雑把な流れは、文字部分を切り出し、あらかじめ決めた解像度に拡大縮小するなどの前処理のあと、何らかの特徴量抽出によって特徴

量化する。文字認識の際にはクエリ文字の特徴量に近い特徴量を持つ既知の文字を推定結果として返すというものである。一連の研究の成果である MQDF[5]という特徴量は手書き中国語の文字認識に引き継がれ、深層学習が盛んな現在でも目にする[6]。

くずし字認識の研究も少なくとも 1990 年代から進められている。基本的な処理の流れは手書き文字認識と同じである。ただし、特徴量抽出法として様々なものが利用されてきた。高次局所自己相関[7]、自己想起型ニューラルネットワーク[8]、HOG[9]などである。深層学習を用いなくくずし字認識としては[10]が最も注目すべき研究であろう。ただ、この研究は完全な自動化ではなく、計算機の支援を受けた人手による翻刻というアプローチである。

2012 年頃から注目されるようになった深層学習では、特徴量抽出は明示的には必要なくなった。その代わりに、ネットワークの構造に腐心することになった。一般物体認識の分野ではネットワークの研究が非常に盛んで、これまでに数多くのネットワークが生み出されてきた。その中でも広く用いられているものの一つに VGG[11]と呼ばれるネットワークがある。畳み込み層と MaxPool 層を多数用いた層数の比較的多いネットワークで、多数の亜種が存在する。VGG を用いた手書き日本語認識の研究として[12]がある。ひらがなに対する認識率は 96.50%、カタカナと漢字を含む全 1004 種の文字に対する正解率は 99.53%と非常に高い。この研究で用いているネットワークは本研究においても比較対象として注目する。深層学習はくずし字認識においても応用され始めており、[13]では変体仮名を対象に 75%前後の認識精度が得られている。

複数の文字の認識、すなわちテキスト認識は、1 文字の認識に比べると格段に難しくなることがある。行の検出の必要性や、前後の文字がくっつく[14]などのため、1 文字ずつに切出すのが容易ではない場合がある。くずし字の認識においては、手書き中国語よりもさらに難易度が上がると考えられる。文字の切出しと文字認識が混然となり、それぞれ独立したタスクとして処理するのが困難な場合が多いためである。実際、過去のくずし字認識の研究においても文字の切出しには多大な労力を割いている[15]。尚、アルファベットの認識においては、文字を単位に認識するのではなく、単語を単位に認識させた方が切出しの手間が省けるだけでなく、精度も良いとの研究[16]もある。定型句の多い古文書の認識においては注目すべき研究であろう。

本研究ではくずし字を認識対象とする。したがって、くずし字の学習は必須である。しかし文字の切出しを明示的に行うのではなく、文字と文字の境界を学習し、深層学習のネットワークの中に組み入れる点が特徴的である。

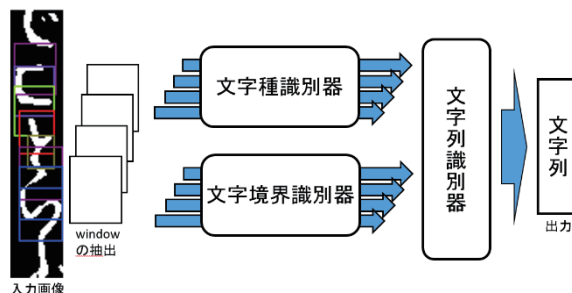


図 1：提案法のネットワーク構成

2. 提案法

2. 1. ネットワーク構成

本研究では仮名 3 文字のくずし字の文字認識を行うことを目的とする。そのためのアプローチとして近年目覚ましい成果を挙げている深層学習を用いる。深層学習は近年の計算機の進歩を背景にニューラルネットワークを様々な面で大きく進化させた手法で、物体認識を皮切りに、音声認識や画像に対する説明文の学習など、様々な分野で著しい成果を挙げている。

本研究では、3 文字の文字列画像を上から下に向かって window をスライドさせていく、sliding window 方式を採用した。その理由を 2 つ挙げる。まず、3 文字という文字数が固定されていたとしても、活字と違いくずし字の画像のアスペクト比は一定ではないこと。また 2 つ目の理由は、くずし字の文字と文字の境界が自明ではなく、文字列のどの部分が 1 文字をなすのかが一見しては分からない状況下でも、あらゆる場合に対応できるようにするためである。採用したネットワークは大きく次の 3 つの部分に分かれる (図 1 参照)。

1. **文字種識別器**: 仮名 1 文字のくずし字を判別する識別器。文字種は 46。入力画像は 32x48 の固定サイズ。
2. **文字境界識別器**: 文字か、文字と文字の間の境界なのかを判別する二値識別器。入力画像は 32x48 の固定サイズ。
3. **文字列識別器**: 文字種識別器と文字境界識別器の出力から、3 文字の文字列の文字種を判別する識別器。入力は 46x50。

Sliding window によって得られた複数の画像に対して、2 種類の確率を計算する識別器にかける。文字種識別器はどの文字が映っているか、46 種の各仮名の確率を出力する。文字境界識別器はそもそも文字ではなく、文字と文字の境界かどうかの確率を出力する。文字境界と判断された領域は文字である確率を割り引いた、46 種の各仮名の確率を得る。これを 46 次元の特徴ベクトルとみなし、全 50 の window すべてに対応する特徴ベクトルを文字列識別器に入力として与えることによって 3 文字の文字種を得る。

2. 2. 処理の全体像

3つの識別器を用いた詳しい処理方法は次の通り。入力情報は3文字の文字列の画像データであり、その解像度やアスペクト比に特に制限は加えない。しかし、文字種識別器や文字境界識別器への入力画像の解像度は固定されている。そこで3文字の入力画像を上から下に向かって、window という窓を通して見る範囲を少しずつ下にずらしていく sliding window 方式を採用する。入力画像の上下の画像データがあれば若干(画像の幅の半分)上から下までスキャンする。Window は全部で 50 回ずらし、文字種識別器と文字境界識別器の2つの識別器への入力とする。Window のアスペクト比は固定されており、基本的には入力画像の横幅を基準に設定する。つまり window の横幅は入力画像の横幅と同じにし、縦の大きさは文字種識別機へ入力サイズのアスペクト比に合わせて設定する。これをデフォルトの window サイズとする。その上で、図1の左端の画像に示すように、window 内の文字部分の大きさに応じて window の大きさを縮小する。基本的にはデフォルトの window サイズを用いるが、window 内の縁が背景(黒い領域)ばかりの時はアスペクト比を維持しつつ window 内に納まる範囲で、なるべく window を小さくする。Window の大きさを変える理由は、学習に用いた画像データは、画像の大きさに対して文字が比較的大きく映っているため、window 内に小さく納まってしまった場合に、認識精度が低下するのを避けるためである。背景かどうかの判定は、今回の実装では単純に黒い領域かどうかで判断している。軽くノイズが乗っている程度であればガウスぼかしなどの利用によって改良することもできる。尚、文字の左右ギリギリにトリミングする必要性はない。というのは、訓練データで学習する際に、半分の画像はランダムに縮小した上で学習させているからである。それでも半分の訓練画像は縮小していないので、文字の左右ギリギリにトリミングできた方が認識精度が良くなるのが期待できる。

Sliding window によって得られた 50 の window 内の画像を 32x48 に拡大縮小し、文字種識別器と文字境界識別器の両識別器に入力する。文字種識別器は 46 種の各文字の確率を出力し、文字境界識別器は文字である確率を出力する。2つの識別器の結果を下の式によって合成する。

文字*c*の確率

$$= \frac{\text{文字境界識別機による文字である確率} + 1}{2}$$

× 文字種識別器による文字*c*の確率

上式の直感的な意味は、文字境界識別器が文字でない判断している場合は、その度合いに応じて文字の確率を割り引くということである。ただし両識別器の結果を尊重したいので、たとえ文字境界識別器が文字の確率 0%と判断していたとしても、文字種識別器が文字の確率を 100%と判断していれば、全体としてはその文字の確率を 50%として扱うということである。上式に出現する「+1」や「2」は、文字境界識別器の出力する確率の値域を 50%以上 100%未満に変換するためのもので、50%という数字はヒューリスティックに決定した値である。

46 種の各文字について、上式によって文字確率の推定値を計算する。これを 46 次元の特徴ベクトルとみなす。VGG などのネットワークの出力を入力画像の特徴ベクトルとして使い回すのは深層学習では今や常套手段となっている。その理由は、人間があれこれ考えを巡らせてヒューリスティックな特徴抽出器を構成するよりも、深層学習で学習したネットワークの方が手軽に、しかもヒューリスティックな特徴抽出器と遜色のない特徴ベクトルを構成できることが多いからである。こうして Sliding window の1つの window から 46 次元の特徴ベクトルが得られる。window は 50 あるので、50 本の特徴ベクトルが得られる。これらの特徴ベクトルを文字列識別器に入力し、3文字の仮名文字を識別する。

2. 3. 仮名文字1文字のくずし字の学習

ここで学習するネットワークは、図1の文字種識別器である。採用したネットワークは、畳み込み層 10 層と全結合層 3 層を含む VGG ライクなネットワークである。VGG16 を代表とする VGG の亜種は広く使われており、本研究でも亜種を用いた。(具体的なネットワークは表1参照のこと)

学習に用いたデータは人文学オープンデータ共同利用センター[17]で公開されている 46 種 228334 文字の仮名文字の画像データである。これらの文字データの出典は 15 の古文書である。そこで、この 15 のうちの 14 の古文書の文字を訓練データ、残る 1 つの古文書をテストデータとする。すなわち 14 の古文書に出現する仮名文字を訓練データとして仮名文字を学習し、残る 1 つの古文書に出現する仮名文字をテストデータとして検証に用いる。学習対象の仮名文字は全 46 種だが、文字の出現頻度は一様ではない。古文書による出現頻度の差も存在する。出現頻度が一様でないために文字によって学習の進捗に差が出てしまうのは好ましくない。そこで文字をバランス

良く学習するために、文字種ごとの出現頻度をなるべく一様にした訓練データを使って満遍なく学習する。例えば、15の古文書のうちの1つである「好色一代男」をテストデータに用いる場合、「好色一代男」を除く14の古文書のそれぞれについて、仮名文字1種につき最大10の文字データをランダムに抽出する。この処理を20回行い訓練データとした。訓練データに含まれる仮名文字は全部で最大128800(=10×46×14×20)となるが、実際には1つの古文書に10回出現しない文字データもあるため124960文字となった。

テストデータは、テストデータ用の古文書に含まれるすべての仮名文字とした。「好色一代男」をテストデータとする場合、37478文字となる。テストデータには文字種による出現頻度のばらつきは大いに存在する。

訓練データに含まれる画像も、テストデータに含まれる画像も、解像度を一定にした。仮名文字は縦長のものが圧倒的に多かったため、32x48の縦長の大きさに標準化した。まず幅が32画素になるように(アスペクト比を維持したまま)拡大縮小し、次に高さを強制的に48画素にそろえた。足りなければ補い、大きすぎる場合は切り詰めた。

(「し」の字のように極端に縦長の場合は、まず幅を強制的に65画素にすべくpaddingしたあと、アスペクト比を維持しつつ幅を32画素に縮小した。ここは改良の余地がある。)高さが48画素になるように拡大縮小しなかった理由は、文字のアスペクト比を維持するためである。3文字のくずし字認識に使う際に、文字と文字の境界が自明には分からないため、境界の情報がなくてもある程度の性能を発揮することを意図した。そのため、高さを48画素に切り詰める際にも、どの箇所を採用するかは乱数で決定した。文字データは2図に示すように、背景色を黒とするモノクロに二値化し、各種ノイズを乗せた。加えたノイズは、回転、拡大縮小、上下左右への移動、剪断、ゴマ塩ノイズ、elastic歪み[18]である。



図2：(左) オリジナル画像を二値化した画像
(右) 各種ノイズを加えた画像

2. 4. 文字境界識別器の学習

ここで学習するネットワークは、図1の文字境

界識別器である。入力画像が文字なのか、それとも文字と文字の間の境界なのかを判断する。学習に利用したネットワークは、出力層のユニットの個数が2個であることを除いて2章の文字種の識別器と同じである。2章のネットワークはクラス数が46だったのに対し、文字境界識別器は文字か、文字と文字の間の境界なのかを判別するため、クラス数は2つになる。そこで出力層のユニットの個数だけを変更し、それ以外のネットワークは文字種識別機と同じものを用いた。

学習に用いたデータは、人文学オープンデータ共同利用センター[17]のデータである。[17]には仮名文字の画像データだけでなく、採取元の古文書の画像データもある。そこで、図2に示すように、文字と文字の間の境界(赤枠)と文字そのもののデータ(緑枠)の2種類の画像データを抽出した。学習方法は2章と同じである。

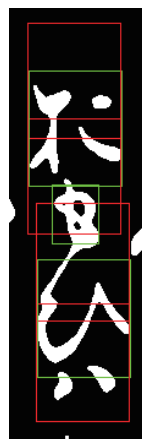


図3：赤枠で囲った部分が文字と文字の間の境界として学習に用いる画像データ。緑枠で囲った部分は境界データ。これら2種のデータを判別する識別器を学習する。

2. 5. 仮名文字3文字のくずし字の学習

ここで学習するネットワークは、図1の文字列識別器である。このネットワークへの入力は、46次元の特徴ベクトルが50本である。特徴ベクトルは1枚のsliding windowに対応し、その要素はwindow内に各文字が映り込んでいる確率の推測値である。これは文字種識別器や文字境界識別器から計算した推測値である。Sliding windowは今回の実験では50回ずらすので、このような特徴ベクトルが50本入力される。これらの情報から3文字の文字種を出力できるように学習するのである。

このような場合、リカレント・ニューラルネットが有力な候補になる。リカレント・ニュー

ラルネットにもさまざまな種類があるが、本研究では双方向 LSTM を用いる。

3. 実験結果

本研究では Keras で実装し、実験に用いた GPU は GTX Titan X である。

3. 1. 文字種識別器の学習

学習に用いたデータは人文学オープンデータ共同利用センター[17]で公開されている仮名データのうち、テストデータには「好色一代男」に由来する文字データを用い、訓練データにはそれを除く 14 の古文書に由来する文字データを用いた。訓練データの学習方法としては、勾配降下法的一种である AdaDelta を用いて交差エントロピーを最小化させるべく学習した。反復 (エポック) 回数は 50 回で、テストデータに対する損失が最小の重みパラメータを採用した。日本語の手書き文字認識の研究[12]で成績の良かった M7_2 と M11 の 2 つのネットワークをベースに表 1 のようなネットワークを構成した。ただし、本研究で用いた M7_2 と M11 は[12]とは少し違うネットワークである。[12]では最終 FC 層を除いて、畳み込み層と FC 層の直後に dropout を行っているとのことだが、本研究で使用した M7_2 と M11 は maxpool 層の直後と、最終 FC 層の直前で行う。その方が成績が良かったためである。dropout のパラメータも、0.5 ではなく、maxpool 層の直後のものは 0.25 にした。提案ネットについても同様である。表 2 に示すように、テストデータに対する損失が最小になったときの正解率は、どのネットワークも 92% 程度となった。提案ネットはパラメータ数が少ない点が優れている。

表 1 : 比較したネットワーク

M7-2	M11	提案ネット
input		
conv3-64	conv3-64	conv3-64
	conv3-64	conv3-64
maxpool		
conv3-128	conv3-128	conv3-128
	conv3-128	conv3-128
		conv3-128
		conv3-128
maxpool		
conv3-192	conv3-256	conv3-256
	conv3-256	conv3-256
maxpool		
conv3-256	conv3-512	
	conv3-512	
maxpool		
FC-1024	FC-1024	FC-512
FC-1024	FC-1024	FC-256
FC-46		

表 2 : 1 文字の仮名文字の学習結果

	M7_2	M11	提案ネット
正解率 [%]	92.09	92.75	92.08
パラメータ数	3409134	8927726	2816174

3. 2. 文字境界識別器の学習

文字境界識別器も文字種識別器と同様に学習させた。ただし、訓練データの作り方が違い、elastic 歪みは入っていないなどの細かい違いが存在する。学習に使用したネットワークも最終層のユニット数が 2 つである点を除いて文字種識別器と同じである。提案ネットによる正解率は 94.74% となった。

3. 3. 文字列識別器の学習

実験に用いたネットワークは、内部ユニット数 128 の LSTM を 2 層経たあと、内部ユニット数 128x2 の双方向 LSTM 1 層という構成のリカレント・ネットワークである。出力層に文字数が 3 文字との情報を用いている。学習アルゴリズムは RMSProp を用いた。テストデータの損失が最小になった時の正解率は 76.38% となった。この時の訓練データに対する正解率は 89.07% であり、必ずしも学習がうまくいってはいないことが分かる。尚、このネットワークのパラメータ数は 496174 である。

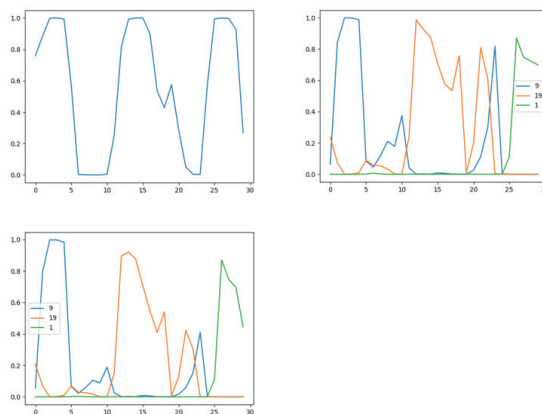


図 3 : 図 1 左端の画像を sliding window で上から下まで window を少しずつずらし、2 つ識別器に入力する。(右上) 文字境界識別機の出力結果。Window 内が文字である確率を表す。横軸は window 番号。(左下) 文字種識別機の出力結果。ここでは「こ」、「と」、「い」の 3 文字の確率だけを表示。それぞれ凡例の 9, 19, 1 に対応する。Window 内がこれら 3 文字である確率を表す。(右下) 2 つの識別器の結果を合成した結果。文字境界の乱れを抑制できている。

4. Discussion

本研究では学習に使用する画像の解像度を 32x48 に揃えた。この解像度で十分読めると判断したからである。文字認識においては、人間の性能を超える結果を出す事例があるので、人間が読める程度の解像度があれば十分と考えている。ただし、いろいろ問題はあつた。人間が読めると述べたが、非常に縦長の画像を 48 画素で cropping してしまうと、読めなくなってしまうこともある。また、本研究は仮名を対象としているが、将来漢字も認識対象に含める場合にも同じことが言えるかどうかは何とも言えない。一回り大きな画像で学習すべきかもしれない。

文字種識別器の誤りで最も多かったのは「う」を「か」と誤認してしまった事例で 13 例ある。率にして 3.51% である。ワースト 10 を右の表に掲載する。「か」が関係するのは「可」を母字とする変体仮名の影響が大きそうである。

順位	正誤	誤り率	事例数
1	う か	3.51	13
2	も き	2.86	20
3	も て	2.58	18
4	ろ か	2.38	3
5	り か	2.19	17
6	も す	2.00	14
7	く て	1.93	10
8	う ら	1.89	7
9	か ら	1.86	16
10	け を	1.85	5

提案法の実行速度について、現在の実装は 3 種類の識別器が統合できておらず、通しての実行ができない状態である。参考までに、個別に実行した場合、文字種識別器の識別のみに要す時間は 1 サンプル辺り 0.20 ミリ秒、文字列識別器の識別時間は 1 サンプル辺り 0.21 ミリ秒だが、3 文字の文字画像から文字列識別器への入力データを作るまでの時間は 1 サンプル辺り 31 ミリ秒となっている。(すべてバッチサイズは 128)

5. まとめ

深層学習によって、仮名 3 文字のくずし字の文字認識を行った。大きく 3 つのネットワークから成る。それらは、1 文字のくずし字の文字認識を行うネットワーク、文字と文字の境界かどうかを判断するネットワーク、さらに両ネットワークによって画像を特徴ベクトルとみなし、多数の特徴ベクトルから 3 文字の文字列を出力できるネットワークの 3 つである。いずれも、深層学習では比較的よく用いられるネットワークであるが、それらの組み合わせで実現した。1 文字の文字認識は 92% 前後と健闘したものの、3 文字の文字認識では 76% 程度に留まった。

参考文献

- 1) 中野三敏：和本のすすめ、岩波新書、2011.
- 2) 金田一秀穂：ロバート キャンベル 日本人はくずし字を学ぼう、文藝春秋 8 月号、2015.
- 3) ETL 文字データベース：入手先 <<http://etlcdb.db.aist.go.jp>>. (参照 2017-11-14)
- 4) 鈴木雅人, 大町真一郎, 加藤寧, 阿曾弘具, 根元義章：混合マハラノビス関数による高精度な類似文字識別手法, 情報通信学会論文誌 (D-II), Vol.J80-D-II, No.10, pp.2752-2760.
- 5) F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake: Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition, IEEE Trans. On PAMI, Vol.9, No.1, pp.149-153, 1987.
- 6) Qiu-Feng Wang, Fei Yin, Cheng-Lin Liu: Unsupervised Language Model Adaption for Handwritten Chinese Text Recognition, Pattern Recognition, Vol.47, No.3, pp.1202-1216, 2014.
- 7) 山田奨治：高次局所自己相関特徴による古文書かな文字認識, 情報処理学会研究報告, Vol.95-CH-25, pp.21-30, 1995.
- 8) 梅田三千雄, 橋本智広：認識処理を援用した文字切り出しによる古文書のキャラクタスポッティング, 電気学会論文誌, Vol.122-C, No.11, pp.1876-1884, 2002.
- 9) 寺沢憲吾, 川嶋稔夫：文書画像からの全文検索のオンラインサービス, 人文科学とコンピュータシンポジウム, pp.329-334, 2011.
- 10) 山本純子, 大澤留次郎：古典籍翻刻の省力化くずし字を含む新方式 OCR 技術の開発, 情報管理, Vol.58, No.11, pp.819-827, 2016.
- 11) K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR, 2015.
- 12) Charlie Tsai, Recognizing Handwritten Japanese Characters Using Deep Convolutional Neural Networks, Technical Report, Stanford Univ., 2016.
- 13) 早坂太一, 大野瓦, 加藤弓枝, 山本和明：ディープラーニングによる変体仮名の翻刻および WWW アプリケーション開発の試み, 人文科学とコンピュータシンポジウム, pp.7-12, 2016.
- 14) Liang Xu, Fei Yin, Qiu-Feng Wang, Cheng-Lin Liu: An Over-segmentation method for Single-touching Chinese Handwriting with Learning-based Filtering, IJDAR, Vol.17, No.1, pp.91-104, 2014.
- 15) 手操俊文, 坪井昭憲, 吉村ミツ, 八村広三郎：江戸期版本画像におけるキャラクタスポッティング, 人文科学とコンピュータシンポジウム, pp.117-124, 2006.
- 16) Max Jaderberg, Karen Simonyan, Andrea Vedaldi, Andrew Zisserman: Reading Text in the Wild with Convolutional Neural Networks, Int. J. of Computer Vision, Vol.116, No.1, pp.1-20, 2016.
- 17) 日本古典籍字形データセット. 人文学オープンデータ共同利用センター, 入手先 <<https://sites.google.com/view/alcon2017prmu>> (参照 2017-11-14)
- 18) Patrice Y. Simard, Dave Steinkraus, John C. Platt: Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis, ICDAR, pp.958-962, 2003.