ノート

受発注伝票用 AI-OCR の開発

大野　宏*　木嶋　祐太*

Development of AI-OCR for Ordering Slips

OHNO Hiroshi* and KIJIMA Yuta*

1. 緒　　言

近年，DX（Digital Transformation）を推進して競争力を上げている企業が注目されており，新潟県でも令和 3 年 3 月に県内産業デジタル化構想 [1] をまとめ，啓蒙普及と支援を行っている。構想を作成するにあたり現状を分析したところ，紙を主体に業務を行っている企業が多く，DX を推進するためには，紙に記された文字をデジタル化する必要がある。この作業を支援するツールが人工知能技術を使った光学文字認識 [2]（Artificial Intelligence - Optical Character Recognition，AI-OCR）である。

新潟県内にも AI-OCR の開発に取り組んでいる企業があり，既存の認識モデルを使ったところ誤認識が発生するとの相談があった。今回，認識対象となる文字は英数字と句読点のみで，フォントは 1 種類であったため，ニューラルネットワークを使った認識モデルを一から作成した。これを AI-OCR に組み込んだところ，受発注伝票の英数字を正しく認識できたので紹介する。

図 1　JPG 形式の伝票文字の拡大画像

2. 開発の概要

提供された 100 枚弱の受発注伝票は JPG 形式で，この方式はデータ圧縮率が高くファイルサイズは小さいものの，一色の背景に一色の文字がある画像では文字周辺にノイズが発生する。また，図 1 に示すとおり，文字と文字の間隔が狭いため，発生するノイズにより 2 つの文字がつながる場合がある。本開発では画像処理を使った文字の分離を工夫し，1 文字ずつニューラルネットワークで認識することにした。

文字認識の手順は次のとおりである。最初に，文字を認識するニューラルネットワークを設計し，学習用の画像データを準備し学習する。次に伝票から認識したい部分を切り出し，さらに 1 個ずつ文字を切り出し，ニューラルネットワークに入力して認識する。ニュラルネットワークの設計と，学習・認識は Pytorch を使い，画像処理ライブラリは OpenCV を利用し，プログラムは Python で作成した。

3. 文字認識

3.1　認識モデルの設計

今回の認識対象は英数字と句読点のみと文字の種類が少なく，フォントが 1 種類のため，手書き数字の認識で良い結果が得られる畳み込みニューラルネットワーク Lenet[3] を参考にした。これは，1989 年にヤン・ルカンらによって提案され，単純な構造で高い認識率が得られ，ニューラルネットワーク研究の再ブームのきっかけとなった。ネットワークは画像の特徴を抽出する畳み込み層と位置ずれを吸収するプーリング

* 下越技術支援センター

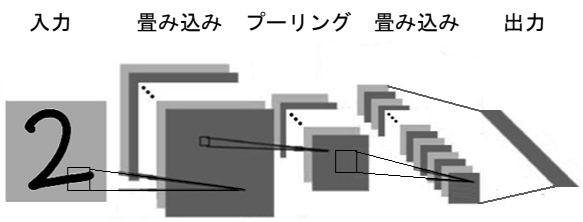


図 2　伝票の文字を認識するモデル

層が 2 段，これらの結果をまとめる全結合層が 2 段および最後の出力層から構成される。出力層は 10 個あり，0～9 のうち最も値の大きいものを認識結果として出力する。

今回提供された受発注伝票に使われている英数字は，フォントが 1 種類で大きさも一定のため，図 2 に示すとおり，Lenet より単純な構造のニューラルネットワークのモデルとする。具体的には，畳み込み層と位置ずれを吸収するプーリング層が 1 段（フィルター数は 20 枚），その次に畳み込み層をもう 1 層設け（フィルター数 50 枚），これを 37 個の出力にまとめて出力する構成になっている。この 37 個は，数字が 0～9 の 10 個，英字が A～Z の 26 個，「. 」からなる。伝票に使われている「, 」と「. 」は区別がつき難く，「, 」は出現する位置が決まっているため，これを 1 つにまとめ，出現位置によって「, 」か「. 」を判断する。

2 つ目の畳み込み層の後に過学習を抑えるドロップアウトを設ける。これは，誤差逆伝搬による学習中に一定の割合のネットワークの結合をわざと欠落させるもので，学習に使っていない文字に対しても高い認識率が期待できる。

3.2　学習用画像の作成

貸与された画像から図 3 に示すとおり，画像編集ソフト（例えば mspaint）を使い 10 個の文字を収集する。画像サイズは 48 画素×48 画素である。この画像を 2 値化して文字の重心を求め，これが画像の中心になるよう移動する。次に，①角度を変える，②明るさを変える，③少し位置を移動する，の操作をプログラムで行い，図 4 に示すとおり 810 枚に水増しする。文字は 37

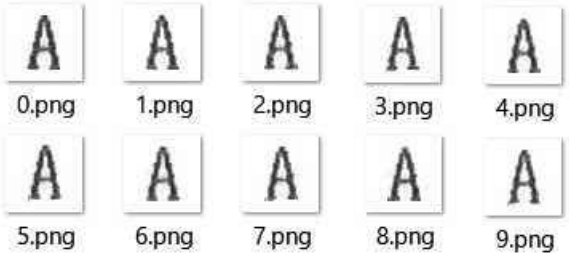


図 3　手作業で収集した画像

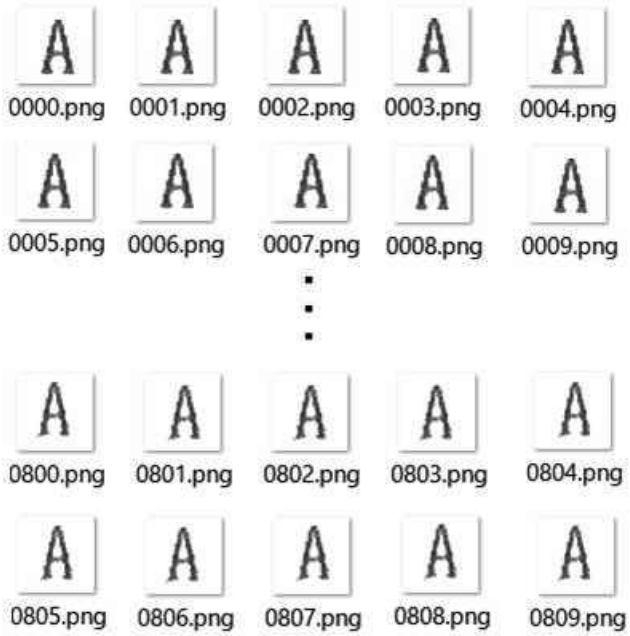


図 4　水増しした画像

種類あるため，学習用画像の枚数は全部で 29,970 枚である。

3.3　認識モデルの作成

3.2 で述べた学習データを使い，3.1 で設計したモデルに入力して得られた出力値と学習データの値との誤差を求め，最適化アルゴリズムでモデルのパラメータを更新する。最適化アルゴリズムには確率的勾配降下法（Stochastic Gradient Descent）を使う。

1 回のパラメータの更新には学習データ 29,970 個の中からランダムに取り出した 370 個の学習データで行い，これを 81 回（29,970÷370）繰り返し，すべての学習データを使う。これを 20 回（エポック）繰り返したときの誤差を図 5 に示す。学習回数が増加するたびに誤差は減った。なお認識モデルの構造が単純なため

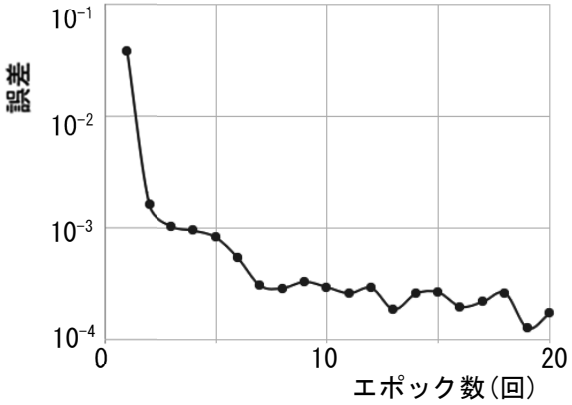
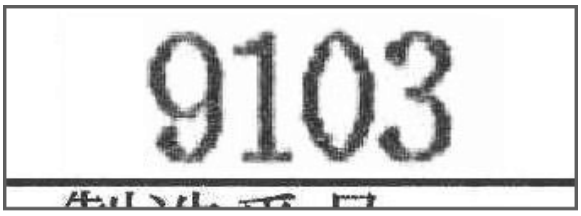


図 5　誤差の推移

発注伝票

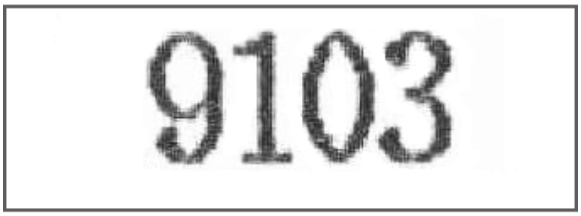
注文番号	仕入れ先	納期	発注日
V00Q00	（株）新潟商会	2021.12.12	2021.11.10
コード番号		担当部署	
19103			
製造番号			
2115107			
部品番号	製品名	仕様	
		12.0 X 120M	
発注数	数量	部事	
173	120	90275360　12.0 X 120M	

図 6　発注伝票の例



(a) 切り出された画像

(b) 横方向に膨張した画像



(c) 抽出した文字列

図 7　文字の抽出

1 エポックに要する時間は，計算を高速化する Graphics Processing Unit のないパーソナルコンピュータ（CPU Intel Core i7-8750H 2.20GHz メモリ 8GB）で約 75 秒であった。

3.4　文字の切り出し

伝票の中から認識したい文字列の画像を切り出す。今回認識したい文字は図 6 に示す伝票のうち，注文番号や納期などである。これらの位置はほぼ一定になるため，伝票の罫線の左上の角を基準とし，ここから一定の大きさの画像を切り出す。図 7(a)にコード番号の画像を示す。少し大きめに画像を切り出すため認識したい英数字以外の線や漢字の一部が含まれてしまう。そのため，横方向に何回か膨張処理を行い，図 7(b)のように文字を一体化させると，文字部分が一番大きなラベル（塊り）になる。この画像にラベリング処理を行い最も大きな面積のラベルだけを取り出すと，図 7(c)に示す英数字の文字列が得られる。

切り出した画像には複数の文字があり，文字と文字の間隔はほぼ一定（20 画素）である。そのため，先端の位置と終端の位置を求め，文字数で等分して 1 文字ずつ切り出す。この文字の重心が，48×48 画素の中央になるよう画像を調整し，モデルに入力して認識する。

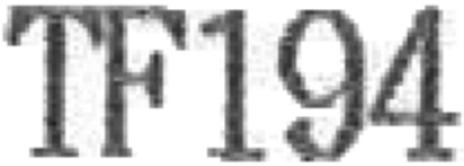
4.　認識実験

提供された画像に対して認識実験を行った。切り出して認識した文字数は全部で 2,772 個あり，認識結果が正しかった文字は 2,767 個，間違った文字は 5 個で，認識率は 99.82％であった。認識率 100％を目指し，次章に示す改良に取り組んだ。

5.　文字の切り出しの改良

認識結果を間違ったときの切り出した文字の例を図 8 に示す。これを見ると，文字の左端に

図 8　誤認識したときの切り出した文字



(a) 認識する文字列と周辺ノイズ

(b) 縦方向に画像を膨張

(c) 文字の先端終端と隙間の検出

図 9　文字切り出しの改良

隣の文字の一部が含まれ，文字の重心も少しずれていることが分かる。その対策として，畳み込み層とプーリング層を増やして位置ずれを減らす方法と，隣の文字がなるべく入らないよう文字の切り出しを改良する方法がある。本開発では，処理が単純で速い後者の方法を採用した。

今回提供された画像は図 9(a) に示すとおり JPG であり文字周辺にノイズが発生するため，単純な画像処理では文字の先端と終端の位置が

図 10　改良後の切り出した文字

若干ずれてしまう。そのため，1 文字ずつ切り出したとき，先の図 8 に示すように隣の文字の一部が含まれ誤認識してしまう。そこで，図 9(b) に示すとおり縦方向に文字を膨張させ，文字の先端と終端を再計算する。この処理は文字と文字の隙間の一部も検出可能となり，図 9(c) のように縦線を入れ，この部分には文字がないと判断する。この処理の後に画像を当分割し文字を切り出すと，図 10 に示すように左上先端のノイズ部分が小さくなり，「D」と正しく認識するようになった。提供された画像に対して再度認識実験を行ったところ，誤認識は発生せず，認識率は 100%になった。

6.　結　　言

(1) 認識する文字の大きさはほぼ一定でフォントも 1 種類のため，構造が単純なニューラルネットワークモデルで高い認識率が得られた。

(2) 画像形式が JPG のためノイズが発生するが文字の切り出しを改良したところ，100%の認識率が得られた。

参考文献

1) https://www.pref.niigata.lg.jp/sec/sogyosuishin/1356891598360.html、県内産業デジタル化構想，2022 年 3 月 15 日

2) 坂本俊之，“PyTorch ではじめる AI 開発”，2021, pp. 256-324.

3) Y. LeCun et.al, “Gradient-based Learning Ap plied to Document Recongnition”, Proceedings of the IEEE, Vol.86, No.11, 1998, pp. 2278- 2324.