

ノイズのある手書き文字に対する DnCNN を用いた認識改善の評価

2015SC017 服部匡志

指導教員：河野浩之

1 はじめに

文書などをスキャンして画像化するとき画像にノイズが乗ってしまい文字などの視認性が低下してしまうことがある。実際に画像のノイズを取り除き視認性を向上させている研究がある [2]。しかしどのようなノイズが視認性を低下させるのか従来研究されてこなかった。そこで本研究の目的は画像のノイズによる認識率の低下とその改善である。そのために MLP による文字認識と DnCNN によるノイズ除去を行う。

2 先行研究

本研究で参考にした各先行研究の概要と課題について比較したものを表 1 にまとめた。

表 1 先行研究

著者	概要	課題
高橋ら [1]	Average pooling を用いることでノイズに強くなった	大量のノイズがある文字に対して効果に差が出ない
Xu ら [4]	CNN の文字認識の認識率が良さがわかった	MaxPooling がノイズに弱いことがわかった
小松ら [2]	U-Net を用いてノイズのある文字画像を読み取りやすくした	ドキュメント全体への適用
高野ら [3]	DnCNN を MR 画像に適用して WN NM や BM3D と比較した	更に多くの画像に適用し、性能の改善を図る

高橋ら [1] や Xu ら [4] の研究ではノイズによる画像認識の精度の低下を扱っている。

小松ら [2] や高野ら [3] はニューラルネットによりノイズ除去を行っている。

3 ノイズ除去による認識改善

MNIST[5] に対して scikit-image でノイズを付与し、ニューラルネットを使用して文字認識を行う。

python(3.6.9) で MNIST[5] に対し scikit-image(0.16.2) を使用してガウスノイズ、スペckルノイズ、ごま塩ノイズを付与した。ガウスノイズとスペckルノイズでは分散を 0.05 から 0.50 まで 0.05 ずつ増加させた。ごま塩ノイズでは amount を 0.00 から 0.50 まで 0.05 ずつ増加させた。図 1 に実際に付与したノイズをまとめた。図 1 で使わ

れている p は amount を表している。

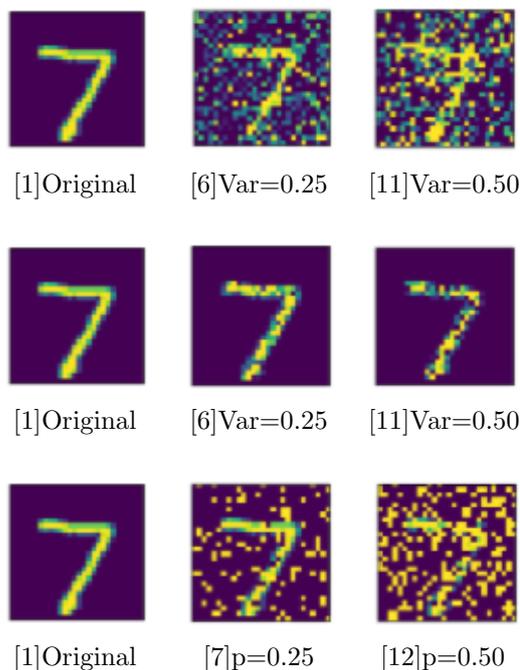


図 1 本研究で使用したノイズの一部

keras(2.4.3) で三層の MLP を作成し文字認識を行った。図 2 は作成した MLP の全体図であり、図 3 は作成した MLP のプログラムの一部である。

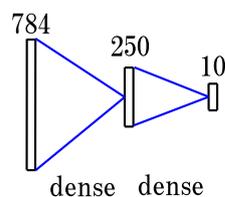


図 2 MLP の全体図

```
model = Sequential()
model.add(Dense(activation='sigmoid', input_dim=784,
                units=250))
model.add(Dense(10, activation='softmax'))
sgd = optimizers.SGD(lr=0.7)
model.compile(loss="mean_squared_error", optimizer=sgd,
              metrics=["accuracy"])
```

図 3 MLP のプログラムの一部

MLP の入力層、中間層、出力層のサイズはそれぞれ 784, 250, 10 となっている。MLP の活性化関数には sigmoid 関数、損失関数は平均二乗誤差、最適化アルゴリズムは確率的勾配降下法、学習率は 0.7、学習時のエポック数は

300, 学習時のバッチサイズは 60 となっている。学習には MNIST[5] の学習用画像 60000 枚を使用して, 学習用画像の割を検証データに使用した。

小松ら [2] が使用している U-Net は画像セグメンテーションの手法である。小松ら [2] の研究では, ノイズのかかった手書き文字画像を判読できる文字画像に出力するのに, U-Net を利用することが有効手段の一つであることを確認していて, U-Net を利用するのが妥当であるといえる。しかし MNIST に対して画像サイズの問題で実装できないので, 本研究では高野ら [3] の研究で利用している DnCNN を利用してノイズ除去を行う。

keras(2.4.3) を使用して DnCNN を作成しノイズ除去を行った。DnCNN は Zhang[6] らが発表したデノイズニューラルネットワークである。図 4 は作成した DnCNN のプログラムの一部である。

```
input_img = Input(shape=(28, 28, 1))
x = Conv2D(64, (3, 3), padding='same')(input_img)
x = Activation('relu')(x)

for i in range(18):
    x = Conv2D(64, (3, 3), padding='same')(x)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)

output_img = Conv2D(1, (3, 3), padding='same')(x)

model = Model(input_img, output_img)
model.compile(optimizer='adam', loss='mean_squared_error')
```

図 4 DnCNN のプログラムの一部

DnCNN の入力層と出力層のサイズは $28 * 28 * 1$ となっている。中間層は 20 層あり, 第 1 層は畳み込みと活性化関数の Rectified Linear Unit(ReLU), 第 2 層から第 19 層は畳み込みと Batch Normalization(BN) と活性化関数 ReLU, 第 20 層は畳み込みのみで構成されている。畳み込みは $3 * 3$ のサイズで畳み込みを行い, 第 1 層から第 19 層では 64 個, 第 20 層では 1 個の特徴マップを作成する。畳み込みではゼロパディングを行うことにより画像サイズは変化しない。出力には入力層の画像から中間層で畳み込んだ画像を引いて出力する。損失関数は平均二乗誤差, 最適化アルゴリズムは Adam, 学習時のエポック数は 10, 学習時のバッチサイズは 32 となっている。学習にはノイズを付与した MNIST[5] の学習用画像 60000 枚を使用した。

4 実験結果

図 5 は認識率の向上について各ノイズごとにまとめている。図 5 の縦軸はノイズの除去前の認識率とノイズ除去後の認識率の差であり, 横軸は分散 (ごま塩ノイズでは amount) である。

ノイズがガウスノイズの場合, 分散が 0.5 の時 54.57% の改善ができた。スペックルノイズの場合, 分散が 0.5 の時 1.61% の改善ができた。ごま塩ノイズの場合, amount が

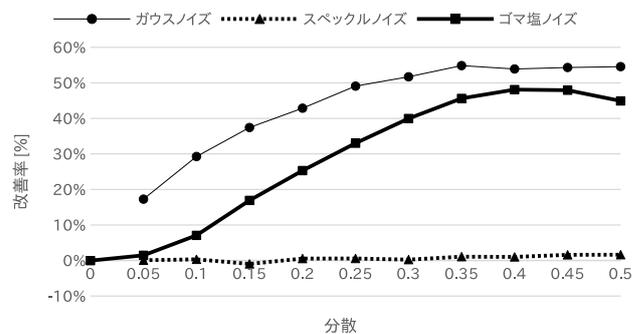


図 5 ノイズごとの改善率の比較

0.4 の時 48.10% の改善ができた。

5 まとめ

画像のノイズを用いて認識率の改善を評価した。その結果スペックルノイズでは認識率があまり改善できず, ガウスノイズとごま塩ノイズでは認識率が改善できることがわかった。

参考文献

- [1] 高橋飛翔, “Deep Learning を用いた画質劣化にロバストな手書き文字認識,” 南山大学, Jan. 2020. 卒業論文.
- [2] 小松里奈, ゴンサルベスタッド, “U-Net による手書き文字画像内のノイズ除去,” 人工知能学会第 32 回全国大会, vol. 2018, no. 1, 4M1-01, pp. 1-4, June 2018. doi: https://doi.org/10.11517/pjsai.JSAI2018.0_4M101
- [3] 高野航平, 伊藤 聡志, “ディープラーニングを利用した MR 画像の雑音除去に関する検討,” 電子情報通信学会東京支部学生会研究発表会, vol. 2018, no. 1, pp. 126, March 2018.
- [4] Z. Xu, Y. Terada, D. Jia, Z. Cai, and S. Gao, “Recognition effects of deep convolutional neural network on smudged handwritten digits,” Proc. 2018 5th International Conference on ICISCE, pp. 412-416, 2018.
- [5] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition,” in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998. doi: 10.1109/5.726791.
- [6] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, “Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising,” IEEE Transactions on Image Processing, vol. 26, issue 7, pp. 3142-3155, July 2017. DOI: 10.1109/TIP.2017.2662206