修 士 論 文

GAN を用いたデータ拡張による古文書文字認識の高精度化

令和 4 年度修了

三重大学大学院工学研究科情報工学専攻
ヒューマンコンピュータインタラクション研究室

岡野 康平

はじめに

古文書とは江戸時代以前に作成された特定の対象に意思・情報などを伝えるための文書であり，現在も民家や寺，神社などで膨大な数が発見されている．古文書の多くは歴史的事実が書かれた重要な史料となっているため，その解読は歴史の研究において欠かせないものである．しかしながら，古文書はくずし字で書かれているため，解読できる専門家が限られており，解読が十分に進んでいないのが現状である．そこで，古文書の解読を支援するための自動古文書文字認識に関する研究が行われている．

古文書文字認識の高精度化に関する先行研究として，市古の研究 [1] がある．古文書文字のデータセットは，字種により学習サンプル数が大きくばらついているため，そのまま学習を行うと，学習サンプル数が少ない字種の認識成功率が低くなる．そこで市古は，学習サンプル数が少ない字種に対してごま塩ノイズ付加によるデータ拡張を行い，先行研究である平田 [2] の手法における CNN の畳み込み層を 4 層から 8 層に増やして認識率を向上させる手法を提案している．筆者による再現実験では，畳み込み層が 4 層のときの認識率 70.3%が，8 層で 75.3%に向上している．さらに，データ拡張によって認識率が 75.3%から 76.0%に向上している．しかしながら，このデータ拡張手法では，付加したノイズの影響で誤認識が生じる場合がある．この問題を解決するためには，文字の変形によるデータ拡張が必要だと考えた．そこで筆者は，敵対的生成モデルの 1 種である DCGAN[3] を用いて古文書文字の特徴を抽出し，学習サンプル数が少ない字種の画像数を増やすことで認識率を向上させることを目指した [4]．しかしながら，学習サンプル数が少ない字種に関しては，ひどく崩れた文字画像が生成されること，また，学習サンプル数に関係なくノイズが含まれた画像が生成されることがあった．これらにより，認識率が向上する字種もあれば低下する字種も存在し，データ拡張前後での認識率は共に 75.3%と向上が見られなかった．このことより，DCGAN により生成された画像から，ノイズが含まれる画像のような文字認識に悪影響を及ぼす画像を取り除く必要があると考えた．

本研究では，敵対的生成モデルの 1 種である SAGAN[5] を用いて古文書文字の特徴を学習し，学習サンプル数が少ない字種の画像数を増やすとともに，生成文字選択用 CNN モデルを用いて，拡張データから悪影響を及ぼす画像を取り除くことで認識率を向上させ

る 2 つのデータ拡張手法を提案する．

1 つ目の手法では，元の学習サンプルで学習した選択用 CNN モデルで SAGAN による生成画像を評価し，正読した画像のみを認識用 CNN モデルの学習サンプルに加えることでデータ拡張を行う．また，元の学習サンプルが少ない字種については，SAGAN での画像生成において，似た画像が多く生成されてしまうことがある．拡張データが似た画像ばかりになることを防ぐため，Perceptual Hash[6] で類似度の高い文字画像は 1 つを残して削除する．Perceptual Hash で画像をハッシュ値に変換すると，ハッシュ値同士のハミング距離を測ることによって画像の類似度を求めることができる．このデータ拡張手法により認識率が 75.3%から 76.5%に向上した．

2 つ目の手法では, 元の学習サンプルと SAGAN による生成画像で学習した選択用 CNN モデルで SAGAN による別の生成画像群を評価し，正読した画像のみを選択してデータ拡張を行い認識用 CNN のファインチューニング [7] のための学習データとして用いる．また，選択用 CNN は，元の学習サンプルと新たに選択された生成データで学習し直す．これを認識率の向上が見られなくなるまで繰り返す．このデータ拡張手法により，認識率が 75.3%から 80.0%に向上した．

以下，第 1 章では研究背景，先行研究，研究の目的について述べる．第 2 章では本研究で用いる関連技術について説明する．第 3 章では提案手法についての説明を行う．第 4 章では提案手法による様々な実験結果について述べる．第 5 章では本研究の結論と今後の課題について述べる．

目次

第 1 章

緒言

1.1　研究背景

古文書とは図 1.1 に示すような江戸時代以前に作成された特定の対象に意思・情報などを伝えるための文書であり，現在も民家や寺，神社などで膨大な数が発見されている．古文書の多くは歴史的事実が書かれた重要な史料であるため，その解読は歴史の研究において欠かせないものである．しかしながら，古文書はくずし字で書かれているため，解読できる専門家が限られており，解読が十分に進んでいないのが現状である．そこで，古文書の解読を支援するための自動古文書文字認識に関する研究が行われている [8][9].

古文書翻刻支援システム開発プロジェクト (HCR プロジェクト) は，手書き文字認識技術を応用して古文書の翻刻を支援するシステムを開発するプロジェクトで，日本における古文書解読支援の初期研究である [10]．HCR プロジェクトにおいて作成されたデータセットが，古文書文字データベース HCD シリーズである [10]．HCD シリーズは，宗門改帳から採字した古文書文字画像で構成されたデータセットや伏見屋文書から切り出された古文書標題により構成されたデータセットを含む．古文書文字認識の基礎的題材として用いられている宗門改帳 [11][12] から採字された HCD1 は，16 字種から構成されている．しかし，このデータセットの字種数は極端に少なく，実際の応用を考えるとさらに多くの字種を含んだ大規模なデータセットが必要である．大規模なデータセットを用いた古文書文字認識技術の例として，ROIS-DS 人文学オープンデータ共同利用センター (CODH) が開発した KuroNet[13] がある．これは，AI 物体検出技術を活用し，画像中に存在する文字を直接探し出して翻刻できるくずし字認識技術である．その後開催された Kaggle くずし字認識コンペ [14] では，KuroNet を上回る AI くずし字認識モデルが現れた．さらに，くずし字認識を誰もが気軽に使えるよう，AI くずし字認識モデルを活用したスマホアプリ「みを (miwo)」[15] が開発された．「みを (miwo)」の主な機能を図 1.2 に示す．しかしながら，古文書における出現頻度の低い字種は，異体字 (形が変わった字) の場合，認識で

きないことが多い．これらを認識できるようにすることが今後の課題となっている．

図 1.1: 古文書画像の例 (羽柴秀吉朱印状から引用)

図 1.2: miwo の主な機能 ([15] から引用)

1.2 研究目的

研究背景で述べたように，古文書の解読は歴史の研究において欠かせないものであるが，解読できる専門家が限られており，古文書の解読を支援するための自動古文書文字認識に関する研究が行われている．しかし，出現頻度の低い字種の認識は困難であり，これらを認識できるようにすることが今後の課題となっている．

本研究では，より幅広い文字を対象に，特にサンプル数が少ない字種に対しても高い精度で古文書文字認識を行うことができるようにすることを目的とし，東京大学史料編纂所の古文書文字データセット [16] を用いて検討を行う．本研究では図 1.3 に示すように，古文書文字の候補字種と関連する情報を提示して人間による解読を支援し，翻刻作業を円滑にすることを想定している．本研究では画像生成モデルの一種である SAGAN を用いて古文書文字の特徴を学習し，学習サンプル数が少ない字種の画像数を増やすとともに，生成文字選択用 CNN モデルを用いて，拡張データから悪影響を及ぼす画像を取り除くことで認識率を向上させる 2 つのデータ拡張手法を提案する．

1 つ目の手法では，元の学習サンプルで学習した選択用 CNN モデルで SAGAN による生成画像を評価し，正読した画像のみを認識用 CNN モデルの学習サンプルに加えることでデータ拡張を行う．また，元の学習サンプルが少ない字種については，SAGAN での画像生成において，似た画像が多く生成されてしまうことがある．拡張データが似た画像ばかりになることを防ぐため，Perceptual Hash で類似度の高い文字画像は 1 つを残して削除する．Perceptual Hash で画像をハッシュ値に変換すると，ハッシュ値同士のハミング距離を測ることによって画像の類似度を求めることができる．

2 つ目の手法では, 元の学習サンプルと SAGAN による生成画像で学習した選択用 CNN モデルで SAGAN による別の生成画像群を評価し，正読した画像のみを選択してデータ拡張を行い，認識用 CNN のファインチューニングのための学習データとして用いる．また，選択用 CNN は元の学習サンプルと新たに選択された生成データで学習し直す．これを認識率の向上が見られなくなるまで繰り返す．

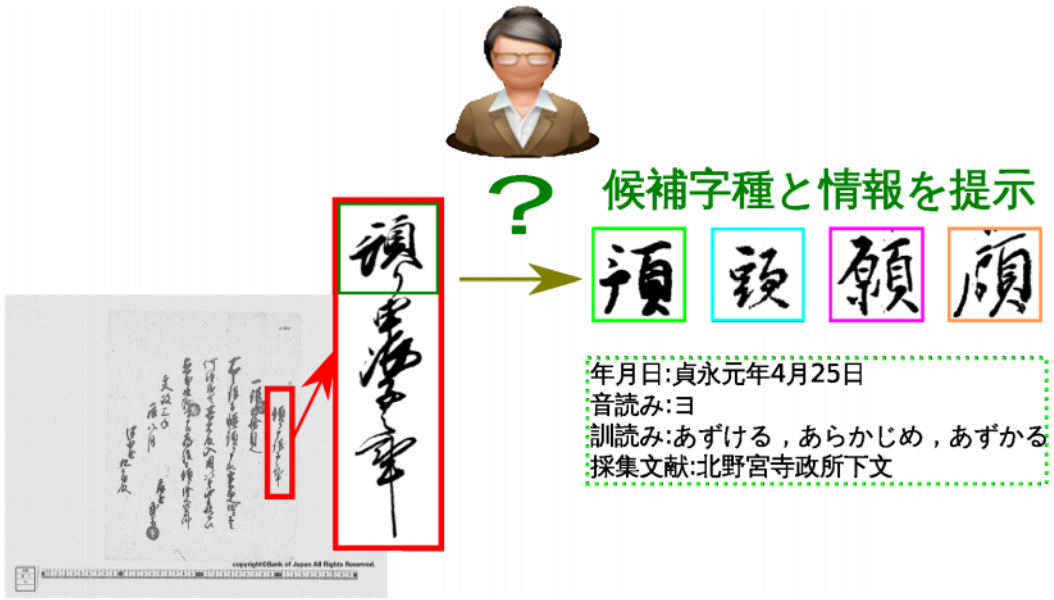

図 1.3: 本研究で想定する支援方法 ([1]p.4 から引用)

第 2 章

本研究で用いる関連技術

2.1 DCGAN

DCGAN は，畳み込みニューラルネットワークによる敵対的生成モデルの 1 種であり，学習データの特徴を学習することで，それらと類似した新たなデータを生成するものである．オリジナルの GAN[17] では生成画像がぼやけていたが，DCGAN ではより自然な画像の生成が可能となっている．DCGAN は，オリジナルの GAN の考え方に則っており，図 2.1 に示すように Generator と Discriminator の 2 つのネットワークで構成される．Generator は類似した新たなデータを生成し，Discriminator には学習データと類似データが与えられ，後者の真贋を判定する．この 2 つのネットワークを競合させて交互に学習することで，Generator が学習データに類似したデータを生成できるようになる．

Generator のネットワークは，図 2.2 に示すように 4 層の畳み込み層で構成される．入力となる 100 次元のノイズベクトル Z から転置畳み込みによって徐々に 64×64 サイズの画像へとアップサンプリングしていく．DCGAN はオリジナルの GAN と違い，ネットワークに全結合層ではなく畳み込み層を使用している．また，オリジナルの GAN の学習が安定しない問題に対応するため，Batch Normalization[18] を導入している．Discriminator は，Generator のアップサンプリング過程を逆にしたようなダウンサンプリング構造になる．

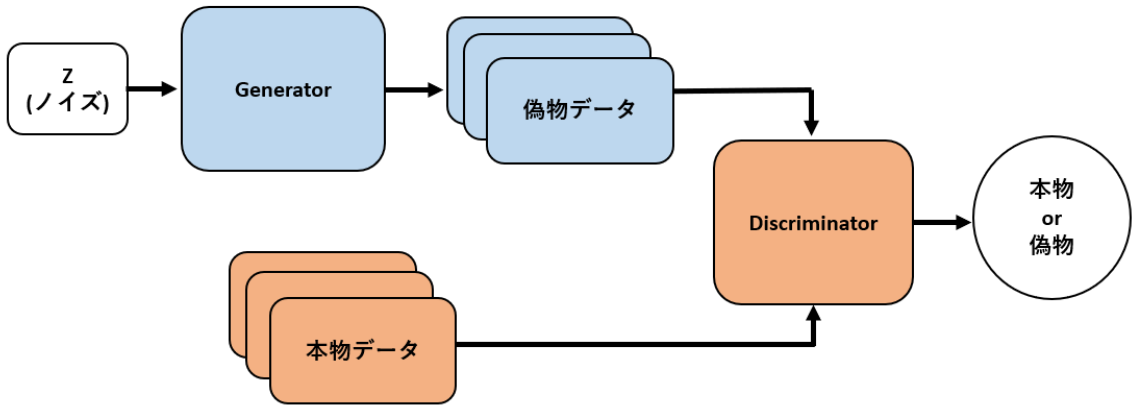

図 2.1: DCGAN の概要図

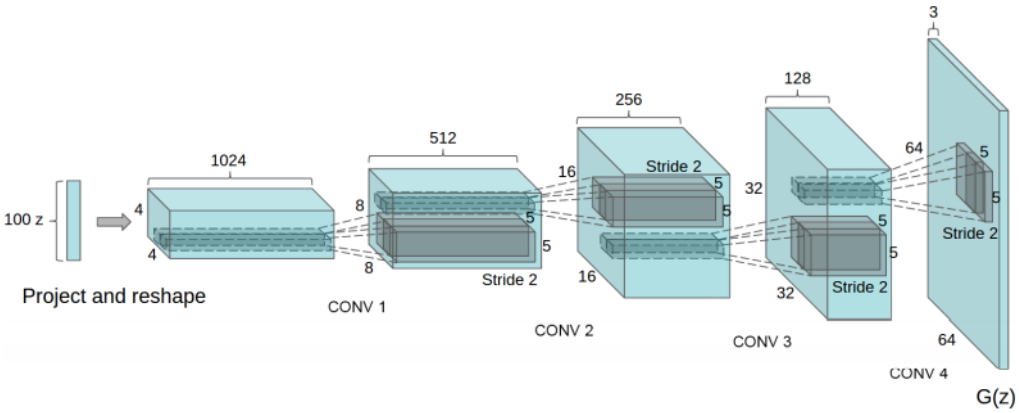

図 2.2: Generator の構成図 ([3]p.4 から引用)

2.2 SAGAN

SAGAN は畳み込みニューラルネットワークによる敵対的生成モデルの 1 種であり，こちらもオリジナルの GAN の考え方に則っている．GAN と同様に，Generator と Discriminator の 2 つのネットワークを競合させて交互に学習することで, Generator が学習データに類似した新たなデータを生成できるようになる．また SAGAN は，Discriminator の Batch Normalization を Spectral Normalization に置き換えることで，学習の安定性を向上させた SNGAN[19] の発展である．SNGAN からの変更点としては，Discriminator だけでなく Generator にも Spectral Normalization を使用している点である．また，Generator と Discriminator ともに図 2.3 のような Self-attention 機構を導入することで，画像の大域的な情報を学習し画質の向上につなげている．

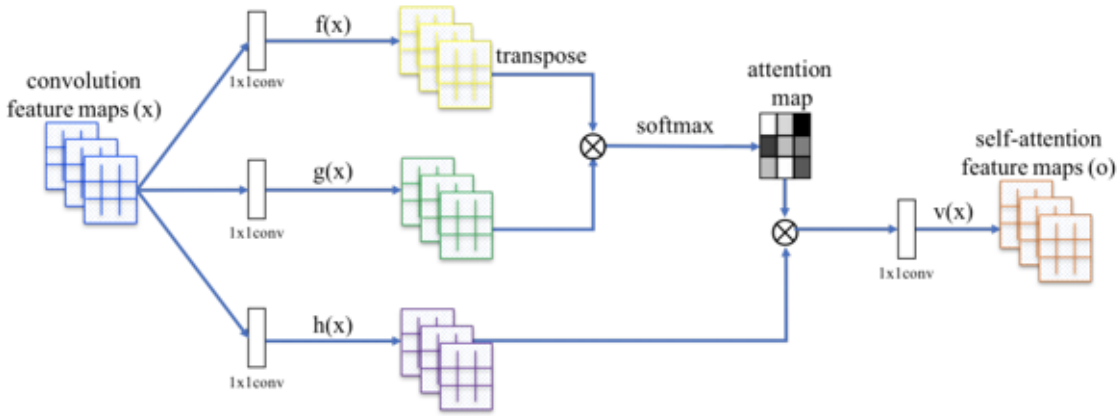

図 2.3: Self-attention の構成図 ([5]p.3 から引用)

2.3 CNN

CNN(畳み込みニューラルネットワーク) は深層学習モデルの 1 種であり，畳み込み層とプーリング層が積み重なったニューラルネットワークである．本研究では文字認識用に市古の提案した CNN(図 2.4) を用いる．この CNN は，平田が用いた 4 層の畳み込み層を持つ CNN に対し，畳み込み層を 8 層にしたものである．平田は，活字漢字認識用ネットワークからの転移学習を用いて CNN の認識率を向上させた．その際に用いた CNN のネットワークは，変動が少ない活字の学習に適したものであり，字形の変動が少ないため，4 つの畳み込み層で十分に学習することができていた．市古は，字形の変動が大きな古文書文字の特徴を学習できるように学習パラメータを増加させようと，畳み込み層を 8 層に増やした．

図 2.4 において，Layer1，Layer2 の Conv-32 は 32 枚のフィルタ，Layer3，Layer4 の Conv-64 は 64 枚のフィルタ，Layer5，Layer6 の Conv-128 は 128 枚のフィルタ，Layer7，Layer8 の Conv-256 は 256 枚のフィルタを持つ畳み込み層を表す．フィルタサイズはすべて 3 × 3 である．Max-Pooling は 2 × 2 で行う．Flatten は 2 次元の特徴マップを 1 次元のベクトルに変換することを表し，変換後のノード数は 10,816 である．Layer9，Layer10，Layer11 の FC-512 は 512 のノードを持つ全結合層を表す．また，Layer1 から 8 の活性化関数には式 (2.1) で示す ReLU 関数を使用する．

$$f(x) = \max(0, x) \tag{2.1}$$

出力層である Layer12 では，式 (2.2) で示す Softmax 関数を使用して，各クラスの事後確率を出力する．

$$f(x) = \frac{e^{x_i}}{\sum_{k=1}^{n} e^{x_k}} \qquad i = 1, ..., n \tag{2.2}$$

ここで，x, n はそれぞれ出力層への入力ベクトルとクラス数である．
損失関数には式 (2.3) で定義される交差エントロピー誤差を使用する．

$$E_{cross} = -\sum_{k=1}^{n} t_k \log y_k \tag{2.3}$$

$\boldsymbol{t} = (t_1, t_2, ..., t_k, ..., t_n)$ は正解ベクトルであり，正解クラスの成分値を 1，それ以外を 0 とする．$\boldsymbol{y} = (y_1, y_2, ..., y_k, ..., y_n)$ はネットワークの出力である各クラスの事後確率を成分とするベクトルである．

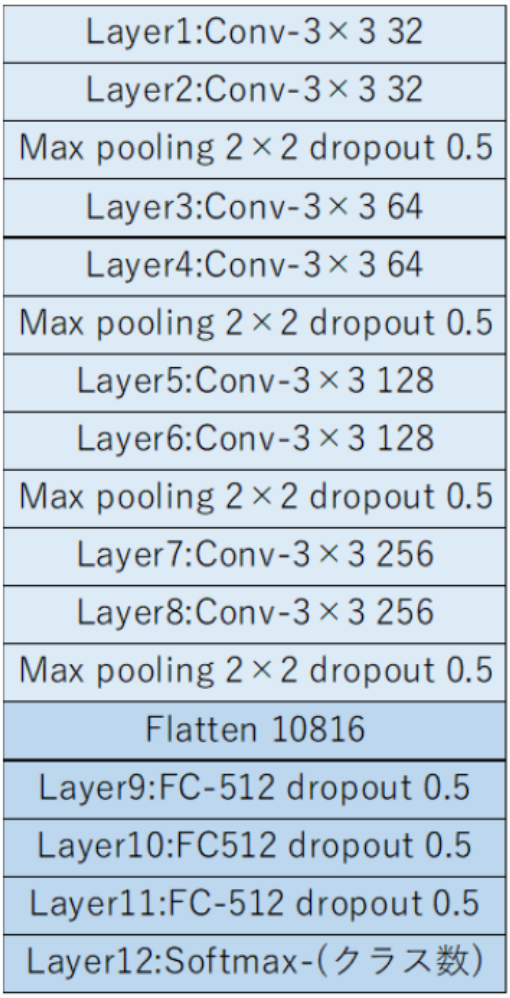
Layer1:Conv-3×3 32
Layer2:Conv-3×3 32
Max pooling 2×2 dropout 0.5
Layer3:Conv-3×3 64
Layer4:Conv-3×3 64
Max pooling 2×2 dropout 0.5
Layer5:Conv-3×3 128
Layer6:Conv-3×3 128
Max pooling 2×2 dropout 0.5
Layer7:Conv-3×3 256
Layer8:Conv-3×3 256
Max pooling 2×2 dropout 0.5
Flatten 10816
Layer9:FC-512 dropout 0.5
Layer10:FC512 dropout 0.5
Layer11:FC-512 dropout 0.5
Layer12:Softmax-(クラス数)

図 2.4: CNN の構成

第 3 章

提案手法

3.1　概要

提案手法の概要を図 3.1 に示す．学習用データセットの各画像に対して前処理を行い，SAGAN によるデータ拡張を行う．また，この際にデータ選択を行う．SAGAN による画像生成において，学習サンプル数が少ない字種に関してはひどく崩れた文字画像が生成されること，また，学習サンプル数に関係なくノイズが含まれた画像が生成されることがある．これらが認識用 CNN の学習に悪影響を及ぼすことを防ぐため，このような画像を取り除く必要があると考えた．その後，データ拡張を行ったデータセットを用いて認識用 CNN の学習を行い，評価用データセットの認識を行う．

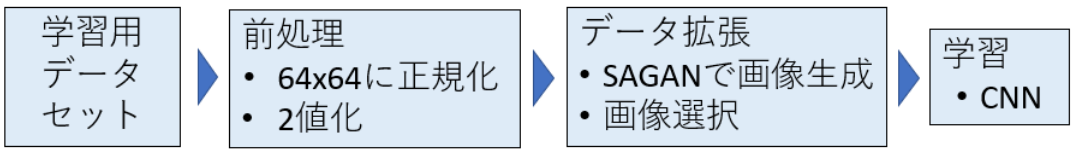

図 3.1: 提案手法の概要

3.2　前処理

本研究に使用する学習用データセットには，図 3.2 のようにカラー画像，白黒画像が含まれ，画像ごとにサイズが異なる．そのため，前処理として画像サイズを 64 × 64 ピクセルに正規化し，2 値化する．図 3.3 に前処理後の画像例を示す．

図 3.2: 原画像の例

図 3.3: 前処理後の画像例

3.3　データ拡張

3.3.1　SAGAN によるデータ拡張

元の学習サンプルが多い字種についてはデータ拡張の効果があまり期待できないため，本研究では，学習サンプル数が 100 未満の字種に対して字種ごとに学習した SAGAN を用いて図 3.4 のような画像を生成し，データ拡張を行う．

また予備的な画像生成実験から，SAGAN の学習において字種ごとに最適な学習回数にばらつきがあることが分かった．そこで，学習サンプル数が 11 から 59 の字種については 700epoch まで学習を行い，300epoch から 700epoch までの間，1epoch おきに 1 枚の生成画像を保存した．学習サンプル数が 60 から 99 の字種については 1000epoch まで学習を行い，600epoch から 1000epoch までの間，1epoch おきに 1 枚の生成画像を保存した．これにより 1 字種につき 200 枚の画像が得られ，そこからデータ選択を行った．

図 3.4: SAGAN により生成した画像の例

3.3.2　データ選択手法 1

SAGAN では，ひどく崩れた文字画像が生成されることがある．拡張データにそのような画像が含まれることを除くため，元の学習サンプルで学習した選択用 CNN モデルで生成画像を評価し，正読した画像のみを選択して認識用 CNN モデルの学習サンプルに加えることでデータ拡張を行う．また，元の学習サンプルが少ない字種については，SAGAN での画像生成において，似た画像が多く生成されてしまうことがある．拡張データが似た画像ばかりになることを防ぐため，Perceptual Hash で類似度の高い文字画像は生成順での

最初の 1 つを残して削除する．Perceptual Hash で画像をハッシュ値に変換すると，ハッシュ値同士のハミング距離を測ることによって画像の類似度を求めることができる．ハミング距離が 0 であれば同じ画像である．本研究では，ハミング距離が 1 から 10 であれば類似画像であるとする．図 3.5 の 2 枚の文字画像では，ハミング距離は 26 であった．また，図 3.6 の 2 枚の文字画像では，ハミング距離は 3 であった．

図 3.5: 生成画像が類似画像ではないと判定した例

図 3.6: 生成画像が類似画像であると判定した例

3.3.3　データ選択手法 2

データ選択手法 2 の概要を図 3.7 に示す．まず，SAGAN により 2 つの画像群を生成する．そのうちの 1 つの画像群と元の学習サンプルで選択用 CNN の学習を行う．そして，学習済選択用 CNN モデルによりもう一方の生成画像群を認識し，正読画像のみを選択してデータ拡張を行い，認識用 CNN のファインチューニングのための学習データとして用いる．また選択用 CNN は，元の学習サンプルと新たに選択された生成データで学習し直す．これを認識率の向上が見られなくなるまで繰り返す．

選択用 CNN

・元の学習サンプル，SAGAN による生成画像，前のループの正読画像で学習.

・SAGAN による別の生成画像群を認識.

認識用 CNN

・元の学習サンプル，正読画像で学習.

・評価用画像を認識.

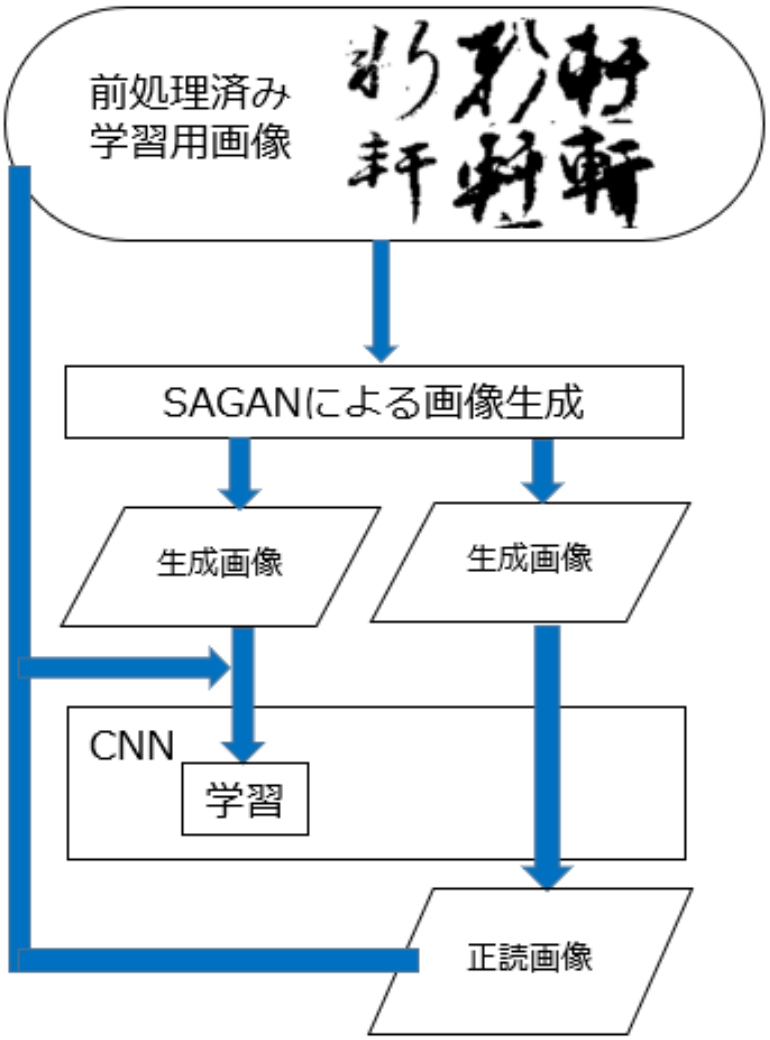

図 3.7: データ選択手法 2 の概要

3.4　CNN による学習

CNN には市古と同じものを使用する．最適化アルゴリズムは Adam，学習率は 0.001 である．学習は，データ拡張前のデータセットとデータ拡張後のデータセットに対して，それぞれ 300epoch を上限として学習を行う．ただし，30epoch の間 loss が低下しなければ途中で終了する．

3.4.1　ファインチューニング

データ選択手法 2 において，認識用 CNN のファインチューニングを行う．2 回目以降の認識用 CNN の学習時には，図 3.8 に赤枠で示すように，Layer5 から Layer12 の学習を行う．

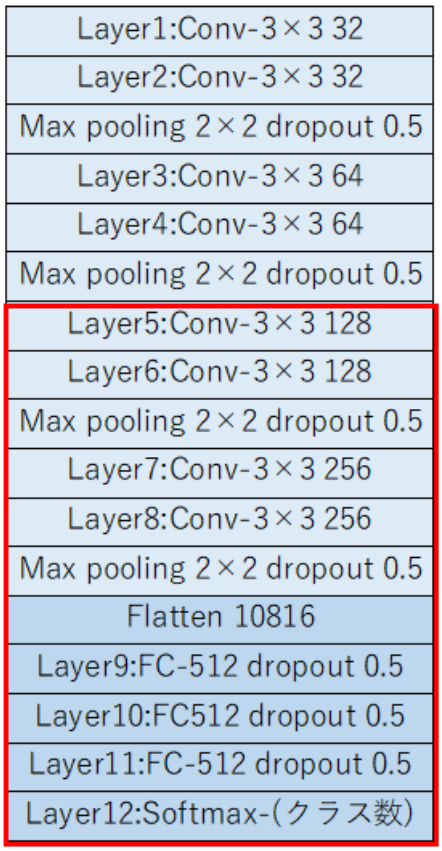

図 3.8: ファインチューニング時の認識用 CNN 学習レイヤー

3.5 評価手法

3.5.1 概要

データセットを学習サンプルと評価サンプルが 3:1 になるように分割して CNN による認識を行い認識率を計算する．また，最も良い結果が得られた手法に関して認識率に加え，precision，recall，F1 score，また統計的有意性の検証を行う．

3.5.2 統計的有意性の検証

提案手法に関して，統計的有意性を検証するためにマクネマー検定を行う．まず検証のため，「データ拡張前では読めたが，提案手法では読めなかった文字画像数」を b,「データ拡張前では読めなかったが，提案手法では読めた文字画像数」を c として計算する．帰無仮説は「2 標本の比率に差がない」，対立仮説は「2 標本の比率に差がある」とし，有意水準 $a = 0.01$ とする．続いて，式 (3.1) に示すように検定統計量を求める．そして，検定統計量から p 値を算出し，p 値＜有意水準であれば有意性があるとする．

$$X^2 = \frac{(b-c)^2}{b+c} \tag{3.1}$$

第 4 章

実験

4.1　実験データ

本研究では東京大学資料編算所の古文書文字データセットを用いる．図 4.1 にデータセットの画像例を示す．データセットには漢字 4,644 字種が含まれている．本研究では，学習サンプル数が 100 未満の字種をデータ拡張で 100 枚に増加させることにする．また，原画像が少なすぎる字種は学習もデータ拡張も困難であることが予備実験により分かったため，本研究ではこれらを使用せず，各字種 11 画像以上ある 1300 字種，計 182,421 枚を用いる．

図 4.1: データセットの画像例

4.2　予備実験 1

4.2.1　実験

まず SAGAN により生成された画像によるデータ拡張の効果を調べるために，先行研究で用いた DCGAN との比較実験を行った．元の学習サンプルが 100 未満の字種に対して，100 サンプルへのデータ拡張を行い，データ拡張前後の認識率を比較した．データ選択は行っていない．

4.2.2　実験結果

表 4.1 にデータ拡張前と DCGAN，SAGAN により生成された画像によるデータ拡張後の認識率を比較した結果を示す．この結果より，元の学習サンプル数が 11-49，70-79 の字種に対しては DCGAN，50-69，80-89，100 以上の字種に対しては SAGAN によるデータ拡張が最も高い認識率となった．90-99 の字種に対しては，データ拡張により認識率が低下した．全字種による認識率では，SAGAN による認識率がデータ拡張前に対して，75.3%から 76.2%に向上し最も高い値となった．

表 4.1: データ拡張前後での認識率の比較

元の学習サンプル数	字種数	データ拡張なし (%)	DCGAN(%)	SAGAN(%)
11-19	353	39.4	50.0	42.9
20-29	201	45.4	53.9	48.6
30-39	138	55.9	61.3	54.9
40-49	92	58.1	61.1	58.9
50-59	79	62.4	62.8	64.5
60-69	45	60.3	58.4	62.5
70-79	37	69.4	69.7	68.3
80-89	35	68.7	66.3	69.2
90-99	33	70.6	64.9	68.5
100 以上	287	82.2	80.9	82.8
全字種	1300	75.3	75.3	76.2

4.2.3　考察

予備実験 1 では，元の学習サンプル数が 11-49，70-79 の字種に対しては DCGAN，50-69，80-89，100 以上の字種に対しては SAGAN によるデータ拡張が最も高い認識率となった．元の学習サンプルが少ない字種については，CNN の学習が不十分でありデータ拡張による認識率向上の余地があったと考えられる．しかしながら，90-99 の字種は元々ある程度学習が進み認識率が高かったため，ひどく崩れた文字画像が学習に加わったことによる悪影響で認識率が低下したと考えられる．

4.3　予備実験 2

4.3.1　実験

認識用 CNN のファインチューニングにデータ選択なしの生成画像を用いる場合とデータ選択した生成画像を用いる場合を比較し，データ選択の効果を確かめる．元の学習サンプルが 100 未満の字種に対して，100 サンプルへのデータ拡張を行い，図 3.6 の学習毎の認識率の変化を比較する．

4.3.2　実験結果

表 4.2 にデータ選択手法 2 における選択の有無による認識率の結果を示す．データ選択を行わず，認識用 CNN のファインチューニングに全生成画像を使用した場合，6 回目の 79.6%が最も高い値となった．データ選択を行った場合，6 回目の 80.0%が最も高い値となった．データ選択を行わない場合に比べ，データ選択を行った方が認識率が高い結果となった．

表 4.2: データ選択手法 2 における選択の有無による認識率の比較

学習回数	データ選択なし (%)	データ選択あり (%)
1	75.9	76.2
2	78.2	78.7
3	78.9	79.5
4	79.4	79.8
5	79.5	79.9
6	**79.6**	**80.0**

4.3.3 考察

データ選択を行わない場合に比べ，データ選択を行った方が認識率が高い結果となった．これは，正読画像を次の学習時のデータ選択に使用することで，悪影響を与える生成画像を除外し，認識により有効なデータを使用することができたためだと考えられる．

4.4 認識実験

4.4.1 実験

予備実験の結果を受けて，本実験では SAGAN を使用し，データ拡張前，予備実験 1 のように単純にデータ拡張した場合，データ選択手法 1 と 2 によるデータ拡張後の認識率を比較する．

4.4.2 実験結果

表 4.3 にデータ拡張前，単純にデータ拡張した場合，データ選択手法 1 と 2 によるデータ拡張後の認識率の結果を示す．手法 1 により，11-79，100 以上の字種については認識率が向上した．しかしながら，80-99 の字種については認識率が低下した．全字種については 75.3%から 76.5%に向上した．手法 2 により，全カテゴリについて認識率が向上した．全字種については 75.3%から 80.0%に向上した．

また，図 4.2 に表 4.3 の結果をグラフ化したものを示す．横軸を元の学習サンプル数，右の縦軸を字種数，左の縦軸を認識率としている．データ拡張前，単純にデータ拡張した場合，データ選択手法 1 と 2 によるデータ拡張後の認識率を棒グラフ，字種数を折れ線グラフで表している．元の学習サンプルが少ない字種が多く，元の学習サンプルが多い字種と比較し認識率も低いことがわかる．

表 4.4 に最も高い認識率であった手法 2 とデータ拡張前について，Precision，Recall，F1 score を比較した結果を示す．手法 2 により，すべての項目について向上した．また，マクネマー検定により，統計的有意性の検証を行った．データ拡張前では読めたが，提案手法では読めなかった数 $b = 1848$，データ拡張前では読めなかったが，提案手法では読めた数 $c = 3943$ であったため，式 (3.1) より，検定統計量 $X^2 = 757.9$ となった．p 値を算出すると $7.685 \times 10^{-167} < 0.01$ となり，有意水準 1%で有意差があることが示された．

4.4.3 考察

手法 2 により，全カテゴリについて認識率が向上した．全字種に対しては 75.3%から 80.0%に向上した．これは，図 4.3 のような正読画像を次の学習時のデータ選択に使用し，図 4.4 のような誤読画像を棄却することで，認識に有効なデータを選んで使用することができたためだと考えられる．

表 4.3: データ拡張前後での認識率の結果

元の学習サンプル数	字種数	データ拡張なし (%)	SAGAN(%)	手法 1(%)	手法 2(%)
11-19	353	39.4	42.9	42.9	48.3
20-29	201	45.4	48.6	50.1	52.3
30-39	138	55.9	54.9	58.0	62.1
40-49	92	58.1	58.9	61.4	66.8
50-59	79	62.4	64.5	66.1	70.5
60-69	45	60.3	62.5	62.4	71.5
70-79	37	69.4	68.3	73.1	77.3
80-89	35	68.7	69.2	66.8	75.2
90-99	33	70.6	68.5	66.8	77.7
100 以上	287	82.2	82.8	82.9	85.6
全字種	1300	75.3	76.2	76.5	80.0

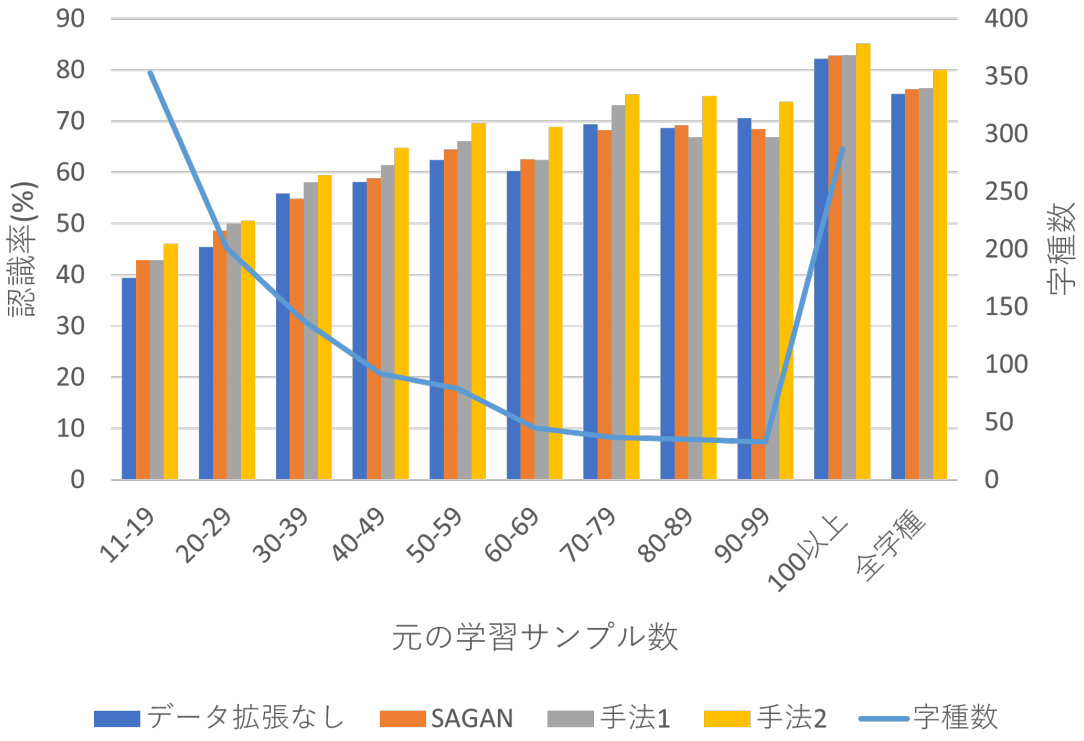

図 4.2: データ拡張前後での認識率の比較

表 4.4: データ拡張前とデータ選択手法 2 の比較

手法	データ選択なし	データ選択手法 2
Accuracy	0.753	0.800
Precision	0.669	0.735
Recall	0.565	0.634
F1 score	0.585	0.660

図 4.3: 正読画像例

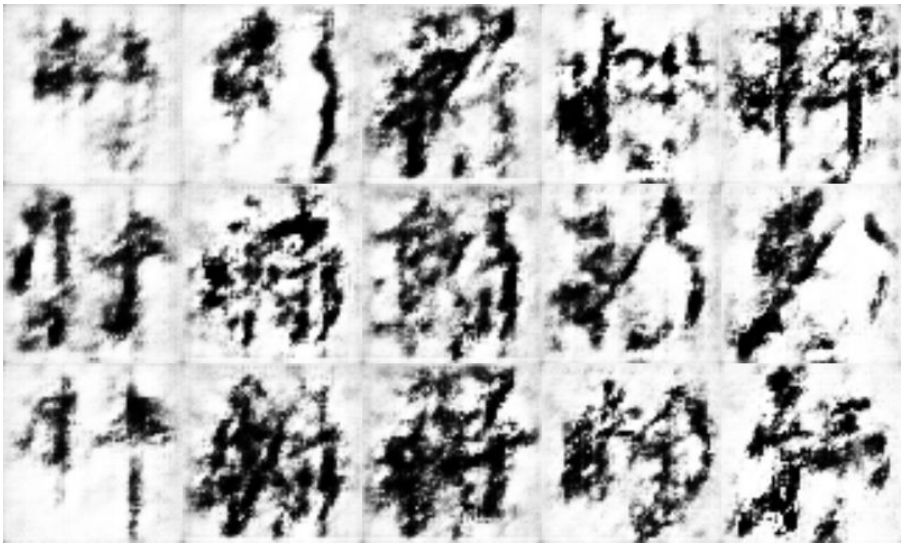

図 4.4: 誤読画像例

第 5 章

結言

5.1　まとめ

本研究では，SAGAN を用いて古文書の学習サンプルが少ない字種の画像データ数を増加させるデータ拡張手法を提案した．元の学習サンプルと SAGAN による生成画像で学習した選択用 CNN モデルで SAGAN による別の生成画像群を評価し，正読した画像のみを選択してデータ拡張を行い，認識用 CNN のファインチューニングのための学習データとして用いた．これを認識率の向上が見られなくなるまで繰り返した．本手法により，CNN を用いた文字認識率が全カテゴリについて向上した．全字種については 75.3%から 80.0%に向上した．

5.2　今後の展望

今後の展望として，大きく 3 つのことを考えていく必要がある．1 つ目は画像生成手法についてである．本研究で使用した GAN のような生成モデルの類に VAE[20] がある．VAE は，GAN と比べると画質の面で劣るが多様性には長けているという特徴がある．近年では，VAE の画質の課題を解決した VQVAE[21] や VQVAE2[22] といったモデルの研究が進んでいる．今後，このようなモデルの使用を検討したい．2 つ目はデータ選択手法についてである．本手法を改良したデータ選択手法，または全く別のアプローチによるデータ選択手法について検討したい．3 つ目は画像認識モデルについてである．本研究では，データ拡張手法に着目したため，先行研究と同じ画像認識モデルを使用した．モデル改良による認識率向上も検討する必要があると考えている．

付録 A

付録

本研究に用いたデータセットは以下のディレクトリに格納する．

- /home/okano/Experiment/WordImg_Dataset2

本研究に関するプログラムはすべて以下のディレクトリに格納する．
プログラムの使用方法，データの詳細については各ディレクトリの README に記述する．

- /home/okano/Experiment/Script/cnn

謝辞

本研究を進めるにあたって，ディスカッションにおいて多くの適切な助言や御指導いただいた若林哲史教授，盛田健人助教，白井伸宙助教に深く感謝いたします．林田祐樹教授には，本論文を作成するにあたり副査として適切なご助言をいただきました．ありがとうございました．また，古文書のデータセット提供に携わっていただいた東京電機大学の大山航教授，東京大学史料編纂所様，誠にありがとうございました。最後になりましたが，日頃お世話になった吉永みゆき事務員，ここまでの大学生活を支えてくれた家族，ヒューマンコンピュータインタラクション研究室の皆様，友人たちに今一度感謝の意を表して，本論文の結びといたします．

参考文献

[1] 市古慎之介,“古文書文字認識の高精度化に関する研究”，三重大学卒業論文，2020

[2] 平田貴嗣,“古文書文字認識の高精度化に関する研究”，三重大学卒業論文，2017

[3] Alec Radford, et al. “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, 2016

[4] 岡野康平,“DCGAN による古文書文字認識の高精度化に関する研究”，三重大学卒業論文，2021

[5] Han Zhang, Ian Goodfellow, et al. “Self-Attention Generative Adversarial Networks”, 2018

[6] JohannesBuchner(2021). https://github.com/JohannesBuchner/imagehash

[7] Zhizhong Li, Derek Hoiem. ”Learning without Forgetting”, 2017

[8] 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, No.14, pp.21-30, 1995.

[9] “特集 挑戦 古文書 OCR”, 人文学と情報処理, no.8, 1998.

[10] 山田奨治，加藤寧，川口洋，原正一郎，石原康人，柴山守，笠谷和比古，小島正美，梅田三千雄，山本和彦,“古文書翻刻支援システム開発プロジェクト報告 (1)-プロジェクト概要-,”情報処理学会研究報告, Vol.2000, No.8, pp.1-8, 2000

[11] 和泉勇治，加藤寧，根元義章，山田奨治，柴山守，川口洋,“ニューラルネットワークを用いた古文書個別文字認識に関する一検討,”情報処理学会研究報告, Vol.2000, No.8, pp.9-15, 2000.

[12] T. Horiuchi, S. Kato, “A Study on Japanese Historical Character Recognition Using Modular Neural Networks,”International Conference on Innovative Computing, Information and Control, pp. 1507-1510, 2009.

[13] Alex Lamb, Tarin Clanuwat, Asanobu Kitamoto, ”KuroNet: Regularized Residual U-Nets for End-to-End Kuzushiji Character Recognition”, 2020

[14] 人文学オープンデータ共同利用センター,”Kaggle コンペティション: くずし字認識,” 2019, http://codh.rois.ac.jp/competition/kaggle/

[15] 人文学オープンデータ共同利用センター,”みを (miwo):AI くずし字認識アプリ,” 2021, http://codh.rois.ac.jp/miwo/

[16] 東京大学,“東京大学史料編纂所-Historiographical Institute The University of Tokyo,” 1984-, http://wwwap.hi.u-tokyo.ac.jp/ships

[17] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, ”Generative Adversarial Networks”, 2014

[18] Sergey Ioffe, Christian Szegedy, ”Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, 2015

[19] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, Yuichi Yoshida, ”Spectral Normalization for Generative Adversarial Networks”, 2018

[20] Diederik P Kingma, Max Welling, ”Auto-Encoding Variational Bayes”, 2013

[21] Aaron van den Oord, Oriol Vinyals, Koray Kavukcuoglu, ”Neural Discrete Representation Learning”, 2017

[22] Ali Razavi, Aaron van den Oord, Oriol Vinyals, ”Generating Diverse High-Fidelity Images with VQ-VAE-2”, 2019