

人間が判読困難なナンバープレートの自動判別における 高解像度化画像の応用

鈴木 友哉¹ 宇田 隆哉¹

概要：我々は、人間に判読困難なナンバープレートを自動的に判別するために、CG を使って大量のトレーニングデータを作成し、実際に撮影したナンバープレート画像を読み取る研究を行ってきた。研究成果として、人間に判読できない程度まで劣化したナンバープレートの画像から、場合によっては非常に高い精度で数字を読み取ることに成功した。一方で、すべての数字が高精度で判別できるわけではなく、予測値が低いものも存在していたため、トレーニング用のデータセットを作成するにあたり、基本的な画像から拡張を行う際、拡張方法が予測値に与える影響について調査を行った。その結果、特定の数字が別の特定の数字に誤分類される確率が一定程度あり、トレーニングデータの拡張には限界を感じた。そこで本論文では、テストデータとなるナンバープレートの画像に対して、RAISR を用いて人間の視認性を向上させるための高解像度化処理を施し、機械学習の予測値に与える影響について調査した。その結果、本論文で述べられている手法を用いた限りでは、コンピュータの算出する予測値の平均が向上することはなかった。ただし、個別の画像を見ると、RAISR による効果が大きいものがあり、高解像度化が有効に働く可能性は示された。

Application of Higher Resolution Images for Automatic Classification of Unreadable License Plates

Tomoya Suzuki¹ Ryuya Uda¹

1. はじめに

我々は、人間に判読困難なナンバープレートを自動的に判別するために、基本となるナンバープレートの画像からコンピューターグラフィックス（以下 CG）を使って、機械学習のための大量のトレーニングデータを作成し、実際に撮影した様々なナンバープレートの画像を読み取る研究を行った [1]。その研究においては、実際にナンバープレートの画像を撮影し、人間に判読が困難な程度まで劣化している状態のものをテストデータとして、その数字の予測値を出力している。日本のナンバープレートはフォントや配置が一意であるため、元となる画像が 1 セットあれば、ボケやヘイズ、白飛び、傾きなどを CG で再現することによって、トレーニング用のデータセットを用意できる。結果と

して、高精度で予測が行える数字があった一方、すべての数字が高精度というわけではなかった。

そこで、トレーニング用のデータセットを作成するにあたり、基本的な画像からどのように拡張したものを追加すると、予測値がどのように変化するか調査した。その結果、どのような種類の画像をトレーニングデータに追加しても、特定の数字が別の特定の数字に誤分類される確率が一定程度あり、トレーニングデータの拡張には限界を感じた。

そこで、本論文では、テストデータとなるナンバープレートの画像に対して、人間の視認性を向上させるための高解像度化処理を施し、機械学習の予測値に与える影響について調査する。高解像度化処理では、欠損したデータを補っているが、オリジナルの画像が復元されるわけではない。しかし、人間の視認性が向上していることは事実であり、機械学習による判別にも影響があると考えた。本論文では、テストデータの高解像度化処理を行った際の判別精度の変化について述べる。

¹ 東京工科大学大学院バイオ・情報メディア研究科
Tokyo University of Technology Graduate School, Hachioji,
Tokyo 192-0982, Japan

2. 関連研究

2.1 画像の高解像度化

2.1.1 補間

画像を拡大する際に劣化を抑える方法として、周囲の4つの画素を用いて輝度値を直線的に線形補間するバイリニア補間が存在する。この方法は、実質的に周囲の4つの画素値の加重平均を求めることになる。これにより、ジャギーが目立たなくなる代わりに、エッジがぼやけてしまうという問題がある [2]。また、16の画素を用いて輝度値を三次式で補間するバイキュービック補間が存在する。この方法は、バイリニア補間と比べるとシャープな画像が得られるが、加重平均を用いる点では共通しており、エッジはぼやけてしまう [2]。CNNは、画像の特徴を、主にエッジ・テクスチャ・カラーの情報からとらえている [2]。そのため、ナンバープレートのような、テクスチャやカラーが同一の物を対象とする場合、エッジの情報が最も顕著な特徴となると考えられる。したがって、我々は、エッジの情報を失わない高解像度化が必要であると考えた。

2.1.2 機械学習を用いない高解像度化

Farsiu らは、高解像度化に関して、バイラテラルに基づく L1 ノルム最小化とロバスト正則化を使用したアプローチを提案している [3]。Yong らは、圧縮センシングの手法を用いて、低解像度入力画像から超解像度画像を生成する方法を提案している [4]。

このような機械学習を用いない手法は、極端にサイズが小さい場合に効果が小さくなる。高解像度化の際に他の画素を参照するためである。本研究で用いる画像は一樣に小さいため、これらの方法は不適である。

2.1.3 機械学習を用いた高解像度化

Dong らは、2014年に Super-Resolution Convolutional Neural Network(SRCNN)を提案している [5]。これは、入力画像と出力画像の再現性を、CNNを用いて end-to-end で機械学習させるものである。これは、比較的古い手法であり、バイキュービック法で拡大された画像を処理するため計算コストが大きいという問題がある。

Romano らは、2017年に Rapid and Accurate Image Super Resolution(RAISR)を提案している [6]。これはトレーニングデータ(対応する低解像度画像と高解像度画像のペア)を使用してフィルタを作成し、トレーニングセットにない画像に適用して、高い解像度の画像を作成するものである。学習が単純で、高速なことが特徴である。ソースコードが公開されており、自由にトレーニングデータを入力できるため、本研究で使用した。

Alves は、RAISR と Variational Quantum Eigensolver(VQE)の利点を組み合わせて、量子コンピューティングで高解像度化を実行し、画質評価の観点から定量的パフォーマンスを維持するアプローチを提案している [7]。

VQE とは、量子コンピュータにより、量子状態を用いて物質の基底エネルギーの値を求める手法である。しかし、我々には、彼らの手法が要求する環境を再現できない。

2.2 画像の視認性向上

高解像度化とは別に、人間の主観的視認性を向上させる方法が多く提案されている。ナンバープレートの場合、数字の部分は同色で塗られているため、画像の高解像度化はエッジの鮮明化とほぼ同義といえる。本論文では RAISR による高解像度化手法を用いたが、類似の効果が見込める手法も存在するため、以下に紹介する。

Gudmundsson らは、遺伝的アルゴリズムを使用したエッジ構成の最適化に基づき、医療画像のエッジを検出するアルゴリズムを提案している [8]。Agaian らは、Canny 法による、脳の MRI 画像用のノイズ耐性のあるエッジ検出を提案している [9]。野澤らは、画像の輝度変化を走査線ごとに、2次 Bezier 曲線を用いて近似した情報を利用して、エッジの鮮鋭化を行う手法を提案している [10]。

3. 提案手法

本研究は、CG を利用して機械学習を行うことで、人間に判読が困難な実際のナンバープレートの画像を自動的に読み取るというものである。ナンバープレートの写真から数字を切り出して画像を作成した後、これまでの研究ではバイリニア補間を用いてその画像を拡大し、機械学習のネットワークに入力していた [1]。しかし、これらの画像は解像度が極端に低く、バイリニア補間が向いていないのではないかと考えた。そこで、本論文では、RAISR を用いて画像を高解像度化し、機械学習のネットワークに入力する方法を考えた。

3.1 RAISR の適用

本論文における提案の主要部分である、RAISR によるテストデータの高解像度化について述べる。我々はこれまで、分類の際のトレーニングデータとテストデータを 64×64 に統一し、研究を行ってきた。本論文においても、これらとの比較が可能なよう、CNN に入力する画像のサイズを 64×64 ピクセルとしている。

RAISR は2つの方法でトレーニングすることができる。1つ目は、フィルタを低解像度画像と高解像度画像のペアから直接学習して作成する方法である。2つ目は、最初にアップサンプラーを低解像度画像に適用し、アップサンプリングされた画像と高解像度画像のペアから学習する方法である。本実験では、より単純で高速な前者の方法で試行した。

はじめに、効果的に RAISR をトレーニングする方法について検討する。RAISR のトレーニング用に、高解像度画像と低解像度画像のペアを沢山用意する必要がある。今

回用いた方法では、高解像度画像を $1/2$ に縮小した低解像度画像を、RAISR が自動的に生成する仕様となっている。そこで、用意すべき高解像度画像について検討する。テストデータの詳細は 3.3 節で述べるが、実際に撮影した、人間に判読が困難なナンバープレートの写真から、数字の切り出したものである。テストデータの画像には暗いものと明るいものがあり、一部白とびや黒つぶれも存在する。また、テストデータの画像のサイズはそれぞれ異なっており、 64×64 ピクセルに拡大すると上下の余白が異なるものになる。これは、数字の周囲に残る余白は、画像の解像度により異なるためである。例えば、 8×8 ピクセルと 32×32 ピクセルの画像があり、どちらも余白が 1 ピクセルとすると、 64×64 ピクセルに拡大した場合、前者は余白が 8 ピクセル、後者は 2 ピクセルとなる。さらに、テストデータの画像のほとんどは、完全に正面から撮影されておらず、傾いている。加えて、テストデータの画像にはノイズやヘイズ、白とびの影響で、部分欠損が見られる。これらを考慮して RAISR のためのトレーニングデータを作成した。具体的な手順は 3.2 節に示す。

次に、RAISR で高解像度化されたテスト画像を、どのようなトレーニングデータでトレーニングされたモデルで分類するべきかについて議論する。考えられるのは、RAISR のトレーニングに用いたの高解像度画像を、分類のためのモデルのトレーニング用画像とする方法である。テスト画像が、RAISR により効果的に高解像度化されているならば、その結果は RAISR のトレーニング用の高解像度画像に近いものになっていると推察される。これが事実であれば、分類のためのモデルのトレーニング用画像も、RAISR のトレーニング用の高解像度画像と同一であるのが、最も適している。従って、分類のためのモデルのトレーニングデータも 3.2 節に示すものを用いる。

3.2 トレーニングデータ

3.2.1 基本画像

トレーニングデータは CG によって作成される。この CG は、ナンバープレートの形状の平面を 3D で作成し、そこにナンバープレートの画像をテクスチャとして貼り付けたものである。なお、我々が研究で扱うのは、一連指定番号と呼ばれる 4 桁の数字だけである。数字が 4 桁に満たない場合には、中黒と呼ばれる点も使用される。我々は、テクスチャに使用する画像を、十分な解像度で撮影された写真から、各数字および中黒につき 1 枚ずつ（合計 11 枚）作成した。本論文では、これらの画像を基本画像と呼称する。また、以降はナンバープレートの「数字」という表現には中黒も含むものとする。

基本画像の作成のため、我々はナンバープレートの撮影を行った。カメラは、iPhone 7 に備え付けられているものを用いた。iPhone で撮影された写真は HEIC 形式で保

存されるため、これを iMazing HEIC Converter を利用して PNG 形式に変換している。以後、本論文で示されている iPhone で撮影された写真は、すべて同様の変換が行われたものである。目視により汚れや塗装ハゲのないナンバープレートを選択し、ナンバープレート全体が収まるように撮影した。その後、写真からナンバープレートのみを切り出し、手動で傾きを修正した後、一連指定番号のみを手作業で切り出した。切り出された各数字は、 30×60 ピクセルの画像として保存した。背景は白で塗りつぶした。数字によって形状が異なるため厳密ではないが、上下の空白が 1 から 2 ピクセルの範囲に収まるように配置した。ただし、中黒は中央に配置した。画像サイズを 30×60 ピクセルとした理由は、この時点で数字が読めなくなるほどに劣化させないため、 64×64 ピクセルより小さい画像とすることを前提として、3 や 5 などの奇数で割り切れる値としたかった（フィルタによる処理が行われるため）ためや、評価を行う上で 10 の倍数が都合がよいと考えたためである。

3.2.2 トレーニングデータの作成

まず、完全な正面から見た基本画像が 3 次元回転によって新たな画像となる。実験においては、正面を π とし、X、Y、Z 方向それぞれ、 $-\pi/4$ 、 $-\pi/8$ 、 π （正面）、 $\pi/8$ 、 $\pi/4$ の 5 段階の回転が行われるものとした。したがって、基本画像は各数字につき 125 枚、合計 1375 枚に拡張されることとなる。以上の手順で作成された 1375 枚の画像は、 64×64 ピクセルになるように余白が白で埋められる。ここまでで作成した画像を、基本トレーニングデータと呼称する。

基本トレーニングデータに対し、さらに 4 つの方法で、各 4 枚ずつ拡張した画像を加えた。

1 つ目は、明度を上げた画像である。すべての基本トレーニングデータに対して、全画素につき元の明度と 51, 102, 153, 204 との和を新たな明度とした 4 つの画像を新たに作成した。ただし、明度が 255 を上回った画素は、255 とした。

2 つ目は、明度を下げた画像である。すべての基本トレーニングデータに対して、全画素につき元の明度から 51, 102, 153, 204 を引いたものを新たな明度とした 4 つの画像を新たに作成した。ただし、明度が 0 を下回った画素は、0 とした。

3 つ目は、数字を縮小した画像である。数字の縮小とは画像全体に占める数字部分の割合を小さくすることである。具体的な縮小の割合は、 $28/30$ 、 $26/30$ 、 $24/30$ 、 $22/30$ の 4 種とした。これは、数字が画像の $2/3$ 以下しか占めないという状況は起こりえないと想定し、その間の大きさを均等に区切ったものである。また、画像サイズの変更の際に端数が生じない値となるようにも考慮している。

4 つ目は、一部分を切り出した画像である。画像の一部分を切り出すことで、ノイズやヘイズ、白とびによって数

字の一部が欠損した状態が疑似的に再現され、精度の向上が期待されるためである。すべての基本トレーニングデータに対して、左上が 40×40 ピクセルのサイズで切り出された画像を新たに作成した。右上、左下、右下に対しても同様に画像を作成した。画像サイズは変更せず、切り出された部分以外は白で塗りつぶした。

以上の拡張により、各数字につき 2125 枚の画像が作成される。すべてのトレーニングデータは前述の処理後、最後にグレースケールに変換される。これらを RAISR 及び分類のためのモデルのトレーニングデータとして用いる。

3.3 テストデータ

本節では、テストデータの作成方法と、高解像度化について述べる。

3.3.1 テストデータの作成

実際に 3.1 項で述べた高解像度化を施すテストデータについて述べる。

評価のためのテストデータとして、ナンバープレートの数字部分の画像を、各数字につき 100 個、合計 1100 個を用意した。このテストデータは、我々がナンバープレートを撮影し、その中で人間による判読が困難なものについて、手動で切り出して用意したものである。当然であるが、すべてのテストデータは異なるものであり、同じ写真の同じ部分から複数個のテストデータを作成してはいない。

テストデータ作成のための撮影は、日本のナンバープレートが装着された、本学のスクールバスに対して行った。連写機能などは用いず、一枚一枚の写真は、被写体との距離や角度を少しずつ変更しながら撮影している。撮影許可の都合により、被写体との距離や角度は厳密に設定されていないが、車両の正面に対して概ね左右 45 度ずつの範囲に収まるように撮影しており、撮影されたナンバープレートの数字が人間によって画像から読み取れない距離を取っている。同様に撮影許可の都合により、すべての写真はスクールバスのナンバープレートから撮影されたものであるが、これらのバスのナンバープレートの仕様は、一般の乗用車のものと同様である。本実験においては、遮蔽物による影響を排除するため、撮影日として降水のない日を選んでいるが、それ以外の制限は設けていない。すなわち、各数字について、撮影時の光量や光源の位置を揃えるための工夫はしていない。スクールバスは使用される車両が日によって変わるため、およそ 20 台のバスを数日かけて撮影した。多くの場合、監視カメラにフラッシュやその他の光源が装備されていないことを考慮し、撮影時にフラッシュは使用していない。

撮影された写真から、ナンバープレートを目視で確認し、一連指定番号の各数字を手動でトリミングした。トリミングの範囲は目視であり、基本画像と極力同じになるよう、範囲を調整してトリミングしている。具体的には、余白が



図 1 テスト画像の例

Fig. 1 examples of test images

確保できる解像度の場合には、最低 1 ピクセル確保する。ボケや解像度の低下などにより、隣の文字と接している場合には、接している箇所の中間を境目として切り出す。同様の原因でナンバープレートの縁と接している場合は、ナンバープレートの縁が入らないように（テストデータの画像に数字以外の線が含まれないように）切り出す。なお、遮蔽物によって数字の半分以上が隠蔽されている写真や、過剰な改変が行われた写真は本実験において使用していない。トリミングされた画像の最小サイズは 2×5 ピクセル、最大サイズは 21×32 ピクセル、平均サイズは 4.91×8.89 ピクセルであった。RAISR の高解像度化による倍率は 2 倍で固定であるため、拡大前の画像サイズは最大でも 32×32 ピクセルである必要がある。トリミングされた時点の各数字の画像サイズが様々であるため、人間が数字を推測可能な画像も含まれてはいるが、基本的には人間が確信を持って判読できない状態である。テストデータの画像の例を図 1 に示す。なお、画像サイズが小さく、そのままでは読者に視認が困難であるため、ピクセルを整数倍して拡大したものを示している。

3.3.2 RAISR によるテストデータの高解像度化

RAISR の高解像度化による倍率は 2 倍で固定であるため、テストデータの画像を、RAISR で高解像度化した際に 64×64 ピクセルになるように、 32×32 ピクセルにサイズを統一することを考える。今回の手法では、高解像度化が主眼に据えられている。従って、歪みや縦横比の変更を伴う変形を行ってしまうと、情報が欠損することが問題となり得る。例えば、最近傍補間があるが、仮に 31×31

ピクセルの画像が 32×32 ピクセルに補間されるのであれば、その歪みは元画像の $1/31$ であるが、 3×3 ピクセルの画像が 4×4 ピクセルに補間されるのであれば、その歪みは元画像の $1/3$ となる。

そこで、画像を正方形にするために、画像サイズが 4×4 ピクセル以下であれば、 4×4 ピクセルになるまで余白を追加し、隣接ピクセルと同一色で埋めることとした。具体的な処理としては、まず、上方向への余白の追加、次に、下方向への余白の追加を、1 ピクセルずつ、縦が 4 ピクセルになるまで繰り返す。その後、左方向への余白の追加、続いて右方向への余白の追加を、1 ピクセルずつ、横が 4 ピクセルになるまで繰り返す。

同様の手順で、 4×4 ピクセルより大きく 8×8 ピクセル以下であれば 8×8 ピクセルに、 8×8 ピクセルより大きく 16×16 ピクセル以下であれば 16×16 ピクセルに、 16×16 ピクセルより大きく 32×32 ピクセル以下であれば 32×32 ピクセルに統一した。最後に、最近傍補間によってすべての画像を 32×32 ピクセルに統一した。この条件で最近傍補間を用いるならば、画像サイズを整数倍していることになり、少なくとも情報の欠損は起こらない。

すべてのテストデータは前述の処理後、最後にグレースケールに変換される。以上の手順で作成された 32×32 ピクセルの画像は 3.1 で示した方法で高解像度化される。

3.3.3 バイリニア補間によるテストデータの高解像度化

RAISR を使用したテストデータと比較するために、バイリニア補間を使用したテストデータも作成する。具体的な手順は、3.3.1 項および 3.3.2 項で示したものと同様であるが、で示したものと同様であるが、最後に行う 2 倍の高解像度化のみを RAISR からバイリニア補間に変更している。

3.4 CNN

我々は、機械学習のネットワークとしては、画像分類によく用いられている畳み込みニューラルネットワーク (CNN) を選択し、評価のための CNN に ResNet を使用した。ResNet は、2015 年に He らによって提案された [11]。これは、誤差が上層から下層へ伝播するにつれて誤差の値が 0 に近づいてしまい、下層のパラメータのトレーニングが進まなくなる問題を改善したものである。ResNet は画像分類の分野において広く使用されているネットワークであり、これをベースとして改良したネットワークに関する論文も発表されている [12][13][14]。本研究においては、最も一般的な ResNet50 を用いることとした。

実際には、トレーニングデータのうち 9 割がトレーニングに、1 割が検証に使用される。また、バッチサイズは我々の動作環境で設定できる最大値であった 256 とした。ソフトマックス層によって出力されるユニット数は、0 から 9 の数字および中黒を対象とするため 11 である。数字に対応したラベルが与える際には、中黒のラベルは「10」

とした。エポック数は上限を設けず、アーリーストッピングを導入した。これは、過学習が進む前にトレーニングを中断させることを目的とするものである。本評価においては、検証時の損失関数の値 (Loss) を監視し、下降が認められなくなった場合に学習が停止されることとした。具体的には、Patience を 3 エポックとし、3 エポック以上 Loss の下降が認められない場合に、その時点で Loss が最小だったエポックのモデルが採用される。

3.5 予測値の確率の算出

テストデータのセットについてテストが行われ、ラベルごとの予測値が確率として出力される。つまり、どのラベルの数字にどれくらいの確率で分類されるかが各数字について出力される。トレーニングおよびテストはそれぞれ 5 回ずつ行い、各回の平均値と標準偏差を求めている。

4. 実装

4.1 RAISR

3.1 項で述べられている、RAISR の実装には、公開されている Python3 のコードを利用した [15]。RAISR のトレーニングデータを train フォルダに、高解像度化する 64×64 ピクセルの画像を test フォルダに配した。データ作成の際の回転には Java 3D を使用した。画像の読み込みや保存、変換、補間など処理には、OpenCV を利用している。画像の切り出しなどの行列計算には、Numpy を用いた。また、このプログラムは結果が BMP 形式で出力されるため、OpenCV で PNG 形式に変換している。

4.2 CNN

CNN の実装には、機械学習のためのソフトウェアライブラリである TensorFlow と、TensorFlow 上で実行可能なニューラルネットワークライブラリのひとつである Keras を用いた。TensorFlow のバージョンは 1.12.0、Keras のバージョンは 2.2.4 である。Keras の Model クラス API の中に定義されている predict() メソッドによって、テストデータに対するラベルごとの予測値が確率として出力される。

5. 評価

紙面の都合上、以下の表では、中黒を 3.4 節で述べたラベルに合わせ「10」と表記している。

3.5 節に述べられている、予測値の確率について評価する。

テスト画像に RAISR を使用した場合の予測の平均値と標準偏差を、それぞれ表 1 と表 2 に示す。また、テスト画像にバイリニア補間を使用した場合の予測の平均値と標準偏差を、それぞれ表 3 と表 4 に示す。

表 1 と表 3 を見比べると、傾向に違いが認められる。しかし、標準偏差を考慮すると、「0」から「9」に関しては誤

表 1 予測の平均値 (RAISR)

Table 1 Mean values by prediction (RAISR)

	1 回目	2 回目	3 回目	4 回目	5 回目
0	0.09334953	0.044576733	0.041092712	0.059361862	0.390019188
1	0.279294158	0.261872314	0.362293305	0.379495648	0.390621731
2	0.058737888	0.067664178	0.044176263	0.129818671	0.118579555
3	0.199866788	0.143488232	0.108307166	0.3326466	0.423254089
4	0.179411467	0.185355518	0.202601813	0.184459714	0.155931706
5	0.236095842	0.202308147	0.033451719	0.309534851	0.42839626
6	0.107505549	0.171057447	0.039787422	0.340308961	0.306521643
7	0.14662847	0.098324641	0.159293538	0.200028152	0.234801079
8	0.029632299	0.019436201	0.0068915	0.032374157	0.078523471
9	0.051848277	0.007415475	0.02566524	0.025073094	0.048343174
10	0.726009459	0.876983691	0.525218854	0.639043499	0.273737982

表 2 予測の標準偏差 (RAISR)

Table 2 Standard deviation values by prediction (RAISR)

	1 回目	2 回目	3 回目	4 回目	5 回目
0	0.23342189	0.14621863	0.13566701	0.15364224	0.4345623
1	0.38897324	0.38400105	0.37898773	0.39361674	0.42845592
2	0.12981373	0.17536274	0.07422978	0.22056778	0.26652765
3	0.30486608	0.2582692	0.20283632	0.37380844	0.45430937
4	0.2757046	0.27417526	0.2130097	0.24529493	0.29349652
5	0.34703457	0.33121252	0.094296314	0.36367086	0.44421324
6	0.18530336	0.2944808	0.07961428	0.31530723	0.41994584
7	0.2595168	0.19652665	0.21538067	0.2612401	0.36448833
8	0.082278065	0.06069608	0.051428154	0.09505375	0.2040015
9	0.13495253	0.030807812	0.059531797	0.069719374	0.19624373
10	0.30355155	0.21944568	0.2811192	0.323904	0.3590668

表 3 予測の平均値 (BILINEAR)

Table 3 Mean values by prediction (BILINEAR)

	1 回目	2 回目	3 回目	4 回目	5 回目
0	0.069038466	0.040090651	0.027094114	0.044227415	0.368468142
1	0.256309492	0.216124374	0.325512867	0.355254939	0.402199492
2	0.045821525	0.088909677	0.046699994	0.168343942	0.13903417
3	0.211090485	0.169456259	0.096895399	0.320871947	0.500970932
4	0.134437754	0.166176969	0.194208401	0.164867999	0.141378255
5	0.187054141	0.140850453	0.032606263	0.266567582	0.439862486
6	0.066544516	0.095228253	0.025364789	0.266989937	0.233836169
7	0.140245223	0.121776837	0.142981708	0.208825812	0.242708134
8	0.010229321	0.005947954	0.002685535	0.020125418	0.074688503
9	0.037246863	0.00263962	0.014324608	0.015695653	0.028038743
10	0.99783216	0.999518002	0.962393904	0.960974709	0.64439584

表 4 予測の標準偏差 (BILINEAR)

Table 4 Standard deviation values by prediction (BILINEAR)

	1 回目	2 回目	3 回目	4 回目	5 回目
0	0.19760048	0.15078078	0.0952676	0.11749066	0.4315115
1	0.37985364	0.370255	0.3750166	0.38916847	0.4200669
2	0.09369775	0.20146258	0.07809542	0.2628537	0.28565398
3	0.31962454	0.30475283	0.18329637	0.36569646	0.4643688
4	0.2236793	0.2646267	0.20425719	0.21511956	0.29578325
5	0.2940031	0.27897874	0.10703407	0.33143643	0.44035706
6	0.12080122	0.20569667	0.036493707	0.26164123	0.352332
7	0.26645014	0.22683036	0.21890588	0.29156992	0.35298392
8	0.026046447	0.02081631	0.012736611	0.046216443	0.1959552
9	0.095308565	0.007455376	0.0219756	0.037751015	0.12298879
10	0.002261373	0.000411814	0.03480661	0.07206901	0.22235832

差の範囲内であり、有意な差はないといえる。「10」の値は RAISR を使用した場合の方が小さくなり、逆に偏差は大きくなっている。

各回を比較すると、5 回目の学習が最も予測の平均値が大きくなっている。

表 5 正しく分類されたサイズの大きい画像の割合

Table 5 Percentages of large images predicted correctly

	1 回目	2 回目	3 回目	4 回目	5 回目
RAISR	40.51%	40.51%	30.38%	43.04%	37.97%
BILINEAR	40.51%	36.71%	29.11%	45.57%	56.96%

6. 考察

バイリニア補間を使用した場合と、RAISR を使用した場合で、差が見られなかったことについて考察する。「10」の精度に差があることを考慮すると、方法を変えることで精度に変化が出る可能性がある。バイリニア補間を使用した場合、全体的に 0 に分類される傾向が強かった状態が、RAISR を使用した場合にわずかに解消されたことが考えられるからである。つまり、トレーニングデータを増やす、トレーニングデータに別の処理を加える、モデルのパラメータを変更するなどの方法により、予測結果が変わり得る。

RAISR を使用したほうが有効に働く場面について考察する。まず、我々はサイズの大きいテスト画像に対して、有効に働くのではないかと考えた。そこで我々は切り出された状態のテスト画像のサイズが 5×8 ピクセル以上のもの（上位 79 個）について調査した。確率が最も高かった予測値が真値と一致していた場合を「正しく分類された」ものとし、その割合を百分率で算出した。表 5 にその結果を示す。

これを見ると、RAISR を使用した場合に大きな優位性は認められない。特に、予測値が比較的高かった 5 回目において、RAISR を使用した場合は低い値を示している。理由として、大きな画像はバイリニア補間を使用した場合でも、エッジの情報が変わりづらいことが考えられる。

我々はコントラストの大きいテスト画像に対して、有効に働くのではないかと考えた。そこで我々は切り出された状態のテスト画像の輝度の差が 140 以上のもの（上位 93 個）について調査し、サイズの大きい画像と同様の基準によって割合を百分率で算出した。ただし、高いコントラストが、インパルスノイズなどの得意ながその存在によってもたらされたものでないことを目視で確認した。表 6 にその結果を示す。

ここからも、RAISR を使用した場合に特別な優位性は認められない。予測値が比較的高かった 5 回目において、RAISR を使用した場合は値が上昇しているものの、27.96% と高いとは言えない。理由として、コントラストが高い画像は、エッジの形状が強く出るため、トレーニングデータとうまくマッチしないことが考えられる。

次に、RAISR が有用な点についてさらに探るため、次の作業を行った。まず、RAISR を使用した場合に 5 回の学習とも正しく分類されたテスト画像（サイズの大きい画

表 6 正しく分類されたコントラストの大きい画像の割合

Table 6 Percentages of high contrast images predicted correctly

	1 回目	2 回目	3 回目	4 回目	5 回目
RAISR	15.05%	10.75%	6.45%	22.58%	27.96%
BILINEAR	15.05%	11.83%	10.75%	20.43%	23.66%

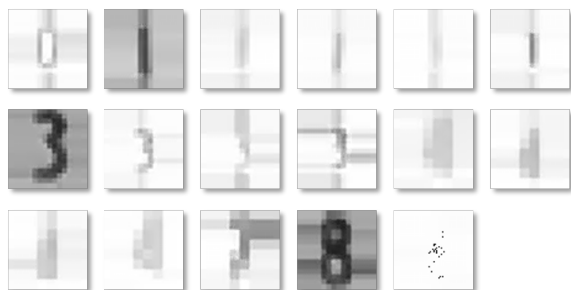


図 2 RAISR を使用したテスト画像

Fig. 2 Test images using RAISR

像と同様)を抽出する。続いて、まず、バイリニア補間を使用した場合に5回の学習とも正しく分類されたテスト画像を抽出する。前者に含まれており、後者に含まれない画像を抽出する。最終的に抽出された画像(RAISR使用後のもの)を図2に示す。比較のために、同じ画像で、バイリニア補間が適用されたものを図3に示す。いずれも左上から順に「0」「1」「1」「1」「1」「1」「3」「3」「3」「3」「4」「4」「4」「4」「7」「8」「10(中黒)」の画像である。これらを見比べても、我々の視覚ではほとんど違いが認められない。しかし、4つの「3」の画像に注目すると、両者のエッジに若干の形状の違いがみられる。また、比較的暗い画像である「1」と「3」と「8」は、エッジがうまく強調されているように見える。この点から、RAISRは高コントラスト比の画像はうまく高解像度化できる可能性が高いといえる。テスト画像を、RAISRに入力する前に高コントラスト処理することで、より効果の高い高解像度化が行えることが示唆される。また、トレーニングデータを改善する、何度も高解像度化する、GANなどの画像生成を用いるなどの方法により、高解像度化が有効に働く可能性がある。「10」の画像を見ると、RAISRを適用した場合に黒いインパルスノイズのような斑点が発生している。しかし、ほかの「10」の画像がこの中に入らないことを考慮すると、有効に働いた可能性は低い。

7. おわりに

本論文では、人間が判読困難なナンバープレートの数字のCNNによる分類に関して、テストデータとなるナンバープレートの画像に対して、人間の視認性を向上させるための高解像度化処理を施し、機械学習による判別精度に与えられる影響について調査した。

その結果、本実験の手法では大きな改善は見られなかったが、高解像度化が有効に働く可能性について示唆が得ら

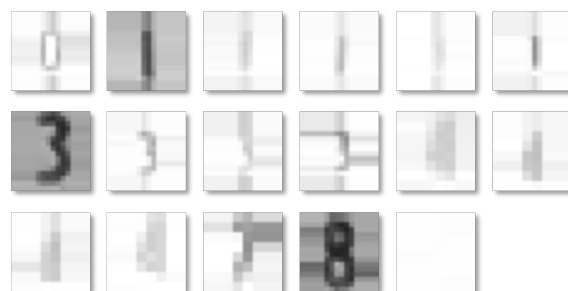


図 3 バイリニア補間を使用したテスト画像

Fig. 3 Test images using bilinear interpolation

れた。

RAISRによる手法は、トレーニングデータの選択や画像サイズなど、検討すべき点が残っている。また、ほかの手法による高解像度化についても検討の必要がある。さらに、テストデータに合わせたトレーニングデータについても、最適なものを探っていくことが必要であると考えている。

参考文献

- [1] Suzuki, T. and Uda, R.: Classifying License Plate Numerals Using CNN, *Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication (IMCOM) 2019* (Lee, S., Ismail, R. and Choo, H., eds.), Cham, Springer International Publishing, pp. 1063–1075 (2019).
- [2] デジタル画像処理編集委員会: デジタル画像処理 [改訂第二版], 公益財団法人画像情報教育振興協会 (CG-ARTS 協会) (2020).
- [3] Farsiu, S., Robinson, M. D., Elad, M. and Milanfar, P.: Fast and robust multiframe super resolution, *IEEE Transactions on Image Processing*, Vol. 13, No. 10, pp. 1327–1344 (2004).
- [4] Yang, J., Wright, J., Huang, T. S. and Ma, Y.: Image super-resolution as sparse representation of raw image patches, *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8 (2008).
- [5] Dong, C., Loy, C. C., He, K. and Tang, X.: Image Super-Resolution Using Deep Convolutional Networks, *CoRR*, Vol. abs/1501.00092 (online), available from <http://arxiv.org/abs/1501.00092> (2015).
- [6] Romano, Y., Isidoro, J. and Milanfar, P.: RAISR: Rapid and Accurate Image Super Resolution, *IEEE Transactions on Computational Imaging*, Vol. 3, No. 1, pp. 110–125 (2017).
- [7] Alves, Y.: Super-Resolution for Imagery Enhancement Using Variational Quantum Eigensolver, *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1–5 (2019).
- [8] Gudmundsson, M., El-Kwae, E. A. and Kabuka, M. R.: Edge detection in medical images using a genetic algorithm, *IEEE Transactions on Medical Imaging*, Vol. 17, No. 3, pp. 469–474 (1998).
- [9] Agaian, S. and Almuntashri, A.: Noise-resilient edge detection algorithm for brain MRI images, *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3689–3692 (2009).
- [10] 麻衣野澤, 昌志亀田: 画像の曲線近似表現を用いたエッジ

の鮮鋭化, 第 78 回全国大会講演論文集, Vol. 2016, No. 1, pp. 65–66 (2016).

- [11] He, K., Zhang, X., Ren, S. and Sun, J.: Deep Residual Learning for Image Recognition, *CoRR*, Vol. abs/1512.03385 (online), available from <http://arxiv.org/abs/1512.03385> (2015).
- [12] Li, B. and He, Y.: An Improved ResNet Based on the Adjustable Shortcut Connections, *IEEE Access*, Vol. 6, pp. 18967–18974 (2018).
- [13] Hsiao, S. and Jang, J. R.: Improving ResNet-based Feature Extractor for Face Recognition via Re-ranking and Approximate Nearest Neighbor, *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 1–8 (2019).
- [14] Wen, L., Zhao, Y., Li, S., Cheng, H. and Zhang, C.: MST-ResNet: A Multiscale Spatial Temporal ResNet for Steering Prediction, *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, pp. 246–251 (2019).
- [15] Chen, J.: A Python implementation of RAISR (MIT. Copyright (c) 2017 James Chen), GitHub (online), available from <https://github.com/movehand/raisr> (accessed 2020-05-18).