情報処理学会論文誌 Vol.60 No.2 680–695 (Feb. 2019)

推薦論文

Adversarial CAPTCHA：畳込みニューラルネットワークに耐性のあるCAPTCHAの提案と評価

阿座上 知香[1,a)] 柴田 千尋[1] 宇田 隆哉[1]

受付日 2018年5月7日, 採録日 2018年11月7日

概要：Web サービスなどでよく用いられる文字列 CAPTCHA は画像認識技術の発展にともなって過度な変形やノイズが必要となり，人間にも判読が困難になっている．我々はこの問題を解決したアモーダル補完を用いた CAPTCHA を考案したが，この CAPTCHA も畳込みニューラルネットワーク（Convolutional Neural Network：以下 CNN）によって解読されることが判明した．そこで，我々は故意に CNN に誤分類を起こさせる敵対的サンプルと呼ばれる技術に着目し，CNN に対しても誤認識させることが可能な CAPTCHA（Adversarial CAPTCHA）を考案した．この論文では深い CNN においても極力少ない変分で誤分類させられるように敵対的フィルタを繰り返し適用する手法を紹介する．実験の結果，本論文で提案する CAPTCHA を用いると人間には確実かつ容易に文字を判別できる範囲で，CNN が誤分類を起こすことを確認した．また，あるモデルで作成した Adversarial CAPTCHA は，重みの異なる別の CNN に適用しても必要なフィルタの適用回数は増えるものの同様に誤分類を起こさせる．結論として，我々が提案する CAPTCHA は，アモーダル補完の演算を必要とする従来の画像認識技術および CNN のいずれを用いても一般的なコンピュータには解読できないものとなった．

キーワード：セキュリティ，CAPTCHA，人工知能，深層学習，畳込みニューラルネットワーク（CNN），Adversarial examples

Adversarial CAPTCHA: Proposal and Evaluation of CAPTCHA against Convolutional Neural Network Analysis

Tomoka Azakami[1,a)] Chihiro Shibata[1] Ryuya Uda[1]

Received: May 7, 2018, Accepted: November 7, 2018

Abstract: Ordinary text-based CAPTCHA requires noise and distortion not to be read by computers. However, too much noise and distortion for preventing automatic recognition also decrease readability of humans. Therefore, as a solution of this problem, we proposed a new CAPTCHA which is easy to be read by humans but is difficult to be read by computers. In this CAPTCHA, characters are drawn with amodal completion in a movie. When ordinary performance computers emulate amodal completion, the character recognition is not done by the time limit since it takes much time for the emulation of amodal completion. However, this CAPTCHA has been broken by Convolutional Neural Network (CNN) since CNN can recognize characters without the emulation of amodal completion. Therefore, in this paper, we proposed an improved CAPTCHA with Adversarial samples which can intentionally make misrecognition. We call it Adversarial CAPTCHA and multi-overlaid Adversarial noises are applied to this CAPTCHA in order to make misrecognition in deep CNN which is usually used for image recognition. We evaluated Adversarial CAPTCHA in experiments and the results showed that deep CNN misclassified characters with high rate although the difference of images from originals is a little. We also found that an image of Adversarial CAPTCHA which was made from a model was effective to other models by the same network and also other models by other networks when the number of filters of the noise increased. In conclusion, our Adversarial CAPTCHA has tolerance of not only ordinary image recognition with amodal completion emulation but also CNN.

Keywords: security, CAPTCHA, artificial intelligence, convolutional neural network (CNN), Adversarial examples

1. はじめに

今日，我々はさまざまなその利便性からWebサービスを利用する．しかし，誰でも簡単にアカウントを作成できることを悪用して，ボットと呼ばれる自動プログラムによって大量にアカウントを作成し，スパムメールを送りつけたり悪質なサイトへ誘導したりする行為が横行している．この問題に対して，一般的なWebサービスではCAPTCHAが利用されている．CAPTCHAとは，1950年にTuringらによって提案されたコンピュータと人間を区別するためのチューリングテストであり，ユーザに対して問題を出しそれに解答させることで人間とコンピュータを区別する仕組みである [1]．CAPTCHAは‘逆’のチューリングテストであるともいわれる [2]．なかでも最も一般的ものは，画像として表示された文字列をユーザに提示し，認識した文字列を入力させる文字列CAPTCHAであるが，画像認識技術の発展にともない，容易にコンピュータに解読されてしまうことが知られており，その有効性が疑問視されている．対抗策としてCAPTCHA作成側はCAPTCHAにノイズや歪みを文字に加えたが，人間の可読性も低下し，ユーザビリティが低下することが問題となっている．この問題を解決するための手法として，Moriらは，アモーダル補完という人間の補完能力を利用した動画CAPTCHAを提案している [3], [4]．人間は一瞬でアモーダル補完が行えるが，コンピュータがアモーダル補完をエミュレートするにはある程度の時間を要するため，画像を連続させて動画にすることで，制限時間内にコンピュータには解答できないとしている．しかし，Moriらの手法では，人間が完全な文字を認識できる時間は一瞬であり，タイミングを逃してしまうと文字を認識できなくなってしまうという問題点がある．そのため，SawadaらはそのCAPTCHAに残効を付加することで，人間が文字を認識する時間を延ばすことに成功した [5]．我々は，この手法を参考に補色と輝度の差を用いることで，人間が文字を認識する時間がさらに延びるようCAPTCHAを改善させた [6]．しかし，OCR（Optical Character Recognition：光学文字認識）以外にもニューラルネットワークによる解読も考えられており，Sivakornらは深層学習を用いてCAPTCHAを読み取ることができることを示している [7]．深層学習とは，現在最も注目されている機械学習の手法の1つで，そのなかでもCNNは特に画像認識の分野において目覚ましい結果を残している．また，今日使用されている視覚情報によるCAPTCHAは画像にできることからCNNによる解読が可能である．そこで，我々はアモーダル補完を利用した我々のCAPTCHAがCNNで分類できるのか実験を行った．結果としては，我々のCAPTCHAはCNNによってほぼ100%解読されることが判明した．一方で，この実験のなかでCNNは人間には判別できないものも判別することができ，文字が完成していなくても何かの特徴をとらえ文字として認識していることが分かった．さらに，Goodfellowらの研究によると，元の画像にごくわずかなノイズを加えることでCNNによる分類の正答率が大きく下がるという結果が発表されており，その加工された画像は人間には元の画像と区別できない程度の劣化しか感じられない [8]．そこで我々はこの結果に着目し，この技術を取り入れることで，OCRによる解読もされず，CNNの分類にも耐性のあるCAPTCHAを提案する．

2章では関連技術について説明する．深層学習の説明もこの章で行う．3章では我々が行ってきた研究やAdversarial examplesについて記述する．4章にてCNNに対抗しうるCAPTCHAの作成方法について述べる．5章は実験と評価，6章では拡充したデータセットでの評価，7章ではStirMarkによるノイズの耐性実験を述べ，8章は本論文の結論とする．

2. 要素技術

2.1 CAPTCHA

CAPTCHA（キャプチャ，“Completely Automated Public Turing test to tell Computers and Humans Apart”）はコンピュータと人間を区別するためのチューリングテストである．CAPTCHAには文字列CAPTCHA，画像CAPTCHA，動画CAPTCHA，音声CAPTCHAなどがあるが，そのどれもが今日までに発展した画像認識により解読が可能であるとされている．最も多く利用されているCAPTCHAは，文字列CAPTCHAと呼ばれる表示されたアルファベットないし数字のランダムな文字列を解答するものである．しかしこの方法は，OCR（Optical Character Recognition）を悪用することで解読することが可能となった．Moriらによるとgimpyの簡略版であるez-gimpyは92%の精度で突破できる [9]．さらに，PWNtchaなどのCAPTCHA分析プロジェクトはCAPTCHAの分析パフォーマンスを向上させた [10]．これに対抗し，作成側はテキストベースCAPTCHAの文字を歪めるなどして可読性を低下させた．しかし，この可読性の低下はボットだけではなく人間にも及ぶ．画像CAPTCHAはテキストベースのCAPTCHAの問題の解決策として提案された．この種のCAPTCHAではテーマに合った画像を選択させることで人間とコンピュータを区別する．画像CAPTCHAで

[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 192–0914, Japan
[a)] g2117001ca@edu.teu.ac.jp

本論文の内容は2017年6月のマルチメディア，分散，協調とモバイル（DICOMO2017）シンポジウムにて報告され，セキュリティ心理学とトラスト研究会主査により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である．

最も有名であるのは Google による reCAPTCHA である．reCAPTCHA ではテーマに合致した画像のみを選択しなければならない．画面上の画像のなかでイメージ内のコンテンツを認識するために複雑なアルゴリズムが必要とされていたが，Sivakorn らは 70〜80%の精度で突破できたと述べている [11]．また，深層学習の台頭により画像はコンピュータに理解できないものではなくなったといえる．reCAPTCHA のもう 1 つの問題点として，新規に次々と撮影される大量のイメージを持っていることが前提であり巨大な画像データベースを把持している組織にしか作れないということが考えられる．reCAPTCHA は巨大な組織に縛られることなく誰もが自由に使える CAPTCHA ではないため，誰もが自由に使用できる CAPTCHA が必要であるといえる．

2.2 深層学習

深層学習とは，特に，近年の応用研究の文脈においては，非常に多数の重みパラメータおよび深い階層を持つニューラルネットワークをモデルとして用いる学習を指す．一般に，ニューラルネットワークは入力層，中間層，出力層の大きく 3 つの層に分けられ，各層はユニットやノードと呼ばれる要素を持ち，隣接層間でのみ結合している．複数の入力を受取ったユニットは 1 つの出力を計算する．深層学習においては，特に中間層のネットワークが 2 層以上，多いときには数百層にいたる多層構造を持ち，それにともない非常に多数の重みパラメータを持つニューラルネットを用いることが一般的である．学習は，伝統的なニューラルネットと同様にバッチと呼ばれる数十から数百個程度の塊にデータを分け，バッチごとに逆誤差伝搬法により勾配を計算し，その勾配をもとに重みの更新が行われる．1989 年にはすでに LeCun が文字認識のために Deep Convolutional Neural Network（DCNN）を使用している [12], [13]．また，さらに最近では George らは再帰的皮質ネットワーク（Recursive Cortical Network，RCN）というモデルを使用して従来の方法より 300 倍以上効率的に CAPTCHA を解決することに成功するなど，CAPTCHA の解析にも使用される [14], [15]．この論文では，分割文字ではなく並べて配置された文字を読むことが可能であると述べられている．

2.3 畳込みニューラルネットワーク

畳込みニューラルネットワーク（CNN）は主に画像認識分野で最高の能力を持つ順伝播ネットワークである．隣り合うユニットすべてが結合されたものではなく，特定のユニットのみが結合している特別なユニットを持っている．これらは畳込みとプーリングと呼ばれる画像処理の演算を行う．CNN のアーキテクチャにはいくつかあり，そのなかでも主に使用されているものを以下で説明する．また，**表 1** に各ネットワークの概要を示す．

表 1 使用された CNN アーキテクチャの概要と比較
Table 1 Overview and comparison of CNN architecture.

r name	#lay*	#conv.	#pl.	#bn.	#fc.	#params
AlexNet	14	5	3	3	3	13.14 M
VGGNet	22	13	5	1	3	20.5 M
GoogLeNet	163	70	15	73	5	14.5 M

r name: referred name (varient), lay: layers
conv: convolution layers, bn: batch-normalization layers
pl: pooling layers, fc: full-connected layers

2.3.1 AlexNet

AlexNet とは Krizhevsky らが提唱した CNN である [16]．AlexNet は重み付きの層が 8 層あり，最初の 5 層は畳込み層，残りの 3 層は全結合層である．最後に全結合層の出力は 1,000 クラスのラベル上に分布を生成する 1,000 通りのソフトマックス関数に収まる．このネットワークは多項ロジスティック回帰の目的を最大にし，これは予想分布のもとで正しいラベルの対数確率の訓練事例全体の平均を最大化することと同等である．第 2，第 4，第 5 の畳込み層のカーネルは，前の層のカーネルマップにのみ接続される．第 3 の畳込み層のカーネルは，第 2 層の全結合のカーネルマップに接続される．全結合層のニューロンは前の層のすべてのニューロンに接続されている．正規化層は，第 1 および第 2 の畳込み層に続く．最大プーリング層は正規化層と第 5 の畳込み層の両方に従う．ReLU（Rectifier Linear Unit）は，すべての畳込み層かつ全結合層に接続された層の出力に適用される．第 1 の畳込み層は，$224 \times 224 \times 3$ の入力画像を，$11 \times 11 \times 3$ サイズの 96 個のカーネルで 4 ピクセルの幅でフィルタリングする（これは，カーネルマップ内の隣接ニューロンの受容野中心間の距離である）．第 2 の畳込み層は，第 1 の畳込み層の（正規化されプーリングされた）出力を入力として取り，それを $5 \times 5 \times 48$ の 256 個のカーネルでフィルタリングする．第 3，第 4 および第 5 の畳込み層は，介在するプーリング層または正規化層なしで互いに接続される．第 3 の畳込み層は，第 2 の畳込み層の（正規化されプーリングされた）出力に接続された $3 \times 3 \times 256$ のサイズの 384 個のカーネルを有する．第 4 の畳込み層は，$3 \times 3 \times 192$ サイズの 384 個のカーネルを有し，第 5 の畳込み層は，$3 \times 3 \times 192$ サイズの 256 個のカーネルを有する．全結合層は，それぞれ 4,096 個のニューロンを有する．

2.3.2 GoogLeNet

GoogLeNet とは Google が ILSVRC2014（Imagenet Large Scale Visual Recognition Challenge 2014）のために開発した CNN である [17]．このネットワークは，パラメータを持つレイヤーだけをカウントするときの層の深さは 22 である．プーリングする場合は 27 となる．ネットワークの構築に使用される層の総数は約 100 であり，正確な数はマシンによってどのように数えられるかによって異

なる．分類の前に平均プーリングを使用し，追加の正規化層を有する．

2.3.3 VGGNet

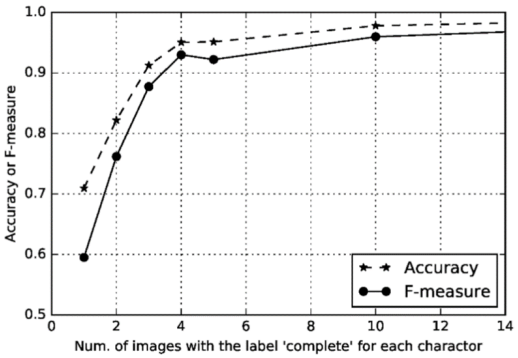
VGGNet は Oxford 大学の Simonyan らにより提案された CNN である [18]．畳込み層は 3 × 3 の非常に小さなサイズであり，各ピクセル（ストライド 1）の入力が畳み込まれる．2 つの 3 × 3 の畳込み層の積み重ね（その間にプーリング層を有さない）が 5 × 5 の受容野を持ち，その 3 つの層は 7 × 7 の受容野を有する．3 つの非線形な正規化層を組み込むことで決定関数をより区別しやすくさせる．3 × 3 の畳込み層を 3 層積み重ねた層の入力と出力の両方が積み重ねた層の重みとしてパラメータ化される．同時に単一の 7 × 7 の畳込み層が得られる．層は 81%増加する．これは，7 × 7 の畳込み層に正規化層を負わせることで 3 × 3 の層を介して分解されるようにすることができる．1 × 1 の畳込み層の組み込みは畳込み層の受容野に影響を与えることなく，決定関数の非線形性を増加させる．

3. 関連研究

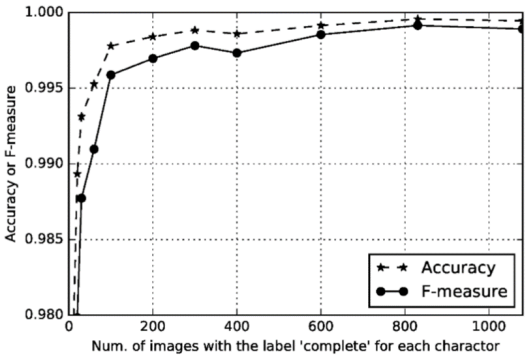
3.1 深層学習による残効と色を付加したアモーダル補完を利用した CAPTCHA の解析

我々は，残効と色を付加したアモーダル補完を利用した CAPTCHA を深層学習が理解するかどうかについて実験を行った [19]．アモーダル補完を利用した CAPTCHA は動画 CAPTCHA であるが，攻撃者が行う手法として動画を 1 枚の画像とすることを想定し，入力は画像とすることにした．入力が画像であるため，使用する深層学習のアルゴリズムは画像認識分野で最高のパフォーマンスを発揮している CNN を採用した．small dataset としてアルファベットの A，B，K，M，R，W を完全に描画した文字（correct），描きかけ（middle），ダミー（dummy）の 3 種類，各 120 枚ずつを用意し，使用する CNN のモデルを決定する実験と，描きかけとダミーのラベル分けに関する簡単な実験を行った．実験の結果，モデルは AlexNet を使用することに決定し，描きかけとダミーのラベルについては，描きかけとダミーを同一ラベルと見なし完全に描画された文字 6 種類と描きかけ+ダミーの 7 ラベルの分類問題とすることに決定した．その後，各ラベルの 90%を学習データとして使用して残りの 1 割をテストデータとして扱い，CNN が CAPTCHA を理解できるか実験を行った．結果は，描きかけの文字を完全に描画された文字と予測し分類したこと以外はほぼ完全に正しく分類することができた．

次に，large dataset として ‘0’〜‘9’ と ‘A’〜‘Z’（‘O’ と ‘C’ を除く）34 文字で構成される大規模な別のデータセットを用意した．このデータセットの場合は，画像のカテゴリを ‘完全に描画しきったもの（correct）’，‘描きかけ (1)（middle(1)）’，‘描きかけ (2)（middle(2)）’，‘ダミー（dummy）’ の 4 種類に分類させた．‘correct’ と ‘dummy’ は前回と同一であるが，‘middle’ を 2 つのタイプに分類した．‘middle(1)’ は文字として認識できない画像で構成し，‘middle(2)’ は人間によって別の文字として認識できてしまう画像で構成した．実際に，‘middle(2)’ について正しく連続したフレームを残効により文字として認識することができたことを確認した．画像は各 1,200 枚ずつ収集した．また，‘middle(1)’，‘middle(2)’，‘dummy’ は ‘correct’ ではないという観点から同一のラベルとしてとらえ，最終的には 35 のラベル，合計 82，800 の画像を持つ分類問題とした．結果は，すべての学習データを使用している場合，予測精度は 0.999 まで上昇した．両方の**図 1** の x 軸は，(*) および (*) カテゴリと文字の各ペアのための画像の数を表している．図 1 から，画像の数が 4 以上である場合に，精度および F 値の両方が 0.9 を超えていることが分かる．これは 0.9 の精度と F 値を出すために必要な ‘correct’ とラベリングされた文字は文字ごとにたった 4 つであるということである．



(a) Accuracy and F-measure with few samples.



(b) Accuracy and F-measure with sufficiently many samples.

図 1 予測精度
Fig. 1 Accuracy and F-measure.

この実験のなかで，我々は CNN が行った誤分類について興味を持った．コンピュータはときどき，文字の大部分が遮蔽物によって隠されていて，人間が正しく認識できないような文字も文字として見ていた場合があった．さらに，人間には何の文字にも見えないようなものであっても完全に描画された文字として認識する場合も存在した．この事実は，コンピュータは人間が見ることができないよう

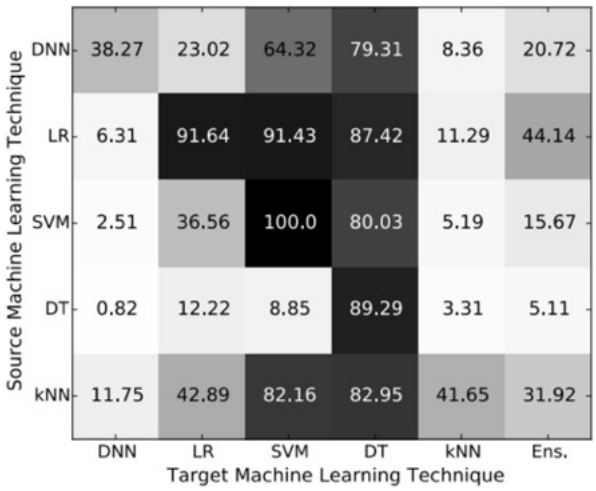


図 2 異なるモデル間での検証 [22]
Fig. 2 Validation between different models [22].

なものも見ることができることを意味しており，文字が完全なものでなくとも何かしらの特徴をとらえて文字と答えていると考えられる．この人間とコンピュータの違いはアモーダル補完を利用した CAPTCHA を改善するための大きな発見であるといえる．

3.2 Adversarial examples

テスト画像に知覚できない非ランダムな摂動（perturbation）を適用したとき，予測誤差を最大にするために入力を最適化するとネットワークの予測が大きく乱れる場合がある．この画像を Szegedy らは Adversarial examples と呼んでいる [20]．さらに Goodfellow は，ニューラルネットワークを含むロジスティック回帰など浅いニューラルネットワークにおいて，テストデータに小さいながらも最悪の場合の摂動フィルタを意図的に適用することによって形成された Adversarial examples を誤って分類すると述べている [8]．摂動が加えられた入力は，モデルが誤った答えを高い信頼度で出力する結果となる．さらに彼らは Adversarial examples の生成方法についても述べており，これによって生成された画像は元の画像と見た目上はほとんど違いがない．

3.2.1 Adversarial examples の関連研究

ここでは，Adversarial examples に関する論文を幾つか紹介する．Papernot らは手元のモデルで作成した Adversarial examples が攻撃対象のモデルを騙すことができると述べている [21]．また，これはニューラルネットワークである必要はなく，他の機械学習手法にも適用できるとしている．異なる機械学習手法で作成した Adversarial examples をそれぞれ騙すことができるのかを検証しており，それに成功している [22]．**図 2** はその結果を示したものである．図中の Ens. は DNN（Deep Neural Network），LR (logistic regression)，SVM (support vector machines)，DT（Decision Tree），kNN（nearest neighbors）のアンサンブルである．この図の縦軸は Adversarial examples を作成したモデルである．この図の横軸はターゲットとなる（Adversarial examples を分類する）機械学習の手法である．また，数値が大きければ大きいほど，ターゲットのモデルは誤分類を起こしたことを示している．たとえば，SVM で作成した Adversarial examples は SVM での分類を 100%誤分類させている．これは，SVM で作成した Adversarial examples は SVM に対して耐性があるということになる．逆に，DT で作成した Adversarial examples は SVM に対し 8.85%しか誤分類させることができなかったということになり，DT で作成した Adversarial examples は SVM に対して耐性がないということになる．図 2 の横軸を見ると，DT（Decision Tree）が最も Adversarial examples に対して誤分類率が高い．このなかでは DNN（Deep Neural Network）が，最も誤分類率が低い．DNN で作成した Adversarial examples に対しては 38%近く DNN も誤分類を起こしているが，その他のニューラルネットワークで作成した Adversarial examples には最大で約 11%しか誤分類を起こしていない．したがって DNN は他の機械学習の手法で作成した Adversarial examples に対して比較的誤分類を起こしにくいといえる．

また，Kurakin らは Adversarial examples はカメラで読み込ませたとしても必ずではないが誤分類を起こさせるという研究を行っている [23]．これに対し Lu らはカメラの距離や角度が変わると有効ではなくなると指摘している．カメラの距離や角度が変わると Adversarial example のスケールが変化し，ほとんどの場合で正しく分類できることになると述べている [24]．しかし，Athalye らはスケールや角度が変わっても有効な Adversarial example を生成できることを論文内で示している [25]．

3.2.2 DeepCAPTCHA

Osadchy らも我々と同様に敵対的ノイズ（Adversarial Noise）を何枚も重ねる CAPTCHA を提案している [26]．Osadchy らの提案と我々の提案のアルゴリズムの差異は ϵ の決め方である．Osadchy らはメディアンフィルタへの耐性がある Adversarial examples の作成と特定のターゲットに誘導のために ϵ を少しずつ増やしている．我々は ϵ の候補のうち，視覚に与える影響が最小限かつ計算量が少ないものを計算的に求めている．ただし，メディアンフィルタによる敵対的ノイズの除去については考慮していない．

また，目的の違いもある．Osadchy らはメディアンフィルタの耐性が得られ，かつ特定のラベルに誤誘導されるような Adversarial examples を作成することを目的としているが，我々は確実に CNN が誤分類を引き起こすような Adversarial examples の作成を目的としている．我々はメディアンフィルタによる敵対的ノイズの除去については考慮していないが，電子透かし技術において一般的な除去

フィルタである StirMark を用いて，どの程度エポック数が増えるかをチェックした．また，Osadchy らの実験では文字（MNIST）で作成した Adversarial examples はメディアンフィルタによってノイズが完全に除去されてしまうことにより結果が収束していない．我々は文字による CNN に耐性のある Adversarial examples の作成に成功しているため，文字列 CAPTCHA に特化した Adversarial examples の作成という点において新規性そして有用性があるといえる．我々のアルゴリズムや ε の決め方については次章にて詳しく説明する．

4. 提案手法

3.1 節で述べたように，CNN で CAPTCHA を判別させた際，ときどき誤分類してしまう場合がある．そして，これは CNN が画像分類問題において優秀すぎるために我々人間には見ることができない情報を特徴ととらえている可能性があるということを述べた．そして，3.2 節では摂動を適用したテスト画像を入力とするとモデルは誤った答えを出力し，生成された画像は元の画像とほとんど変わらないことを紹介した．また，3.2.1 項では Adversarial examples は DNN のような層の深いネットワークは騙しにくいことを述べた．DNN や CNN のような層の深いネットワークで分類を誤らせることができれば，CAPTCHA に応用した際に大きな効力を発揮できるはずである．我々は Adversarial examples で用いる摂動（4.2 節で説明を行う）を重ねることで CNN での分類に耐性を持つ Adversarial examples を作成できるのではないかと仮説を立てた．以下にその生成方法を示す．

4.1 モデルとテスト画像の決定

3.1 節の small dataset での実験と比較を行うために，今回使用するモデルは small dataset で 7 ラベル分類問題を行った AlexNet を採用した．以前のモデルと同様エポック数は 40 とする．テスト画像は 3.1 節における small dataset の完全に描画された文字を使用した．人間が読む必要のないダミーは使用しない．使用したデータは**図 3** に示す．() の中はそれぞれの文字を表している．

4.2 Adversarial CAPTCHA の作成

次に，本論文の主要な提案の 1 つである，Adversarial CAPTCHA の作成の手法を示す．θ をモデルのパラメータとし，x をモデルへの入力，y を x に関連付けるターゲット（ターゲットを持つ学習タスク用），$J(\theta, x, y)$ は CNN を訓練するために使用するコストであるとする．Goodfellow らは，ロジスティック回帰など比較的層が浅いネットワークにおいて，θ の現在の値の近傍におけるコスト関数が，線形で近似するこができる場合，次の式のような摂動を用いることで，効率的に，誤認識へと導くことができること

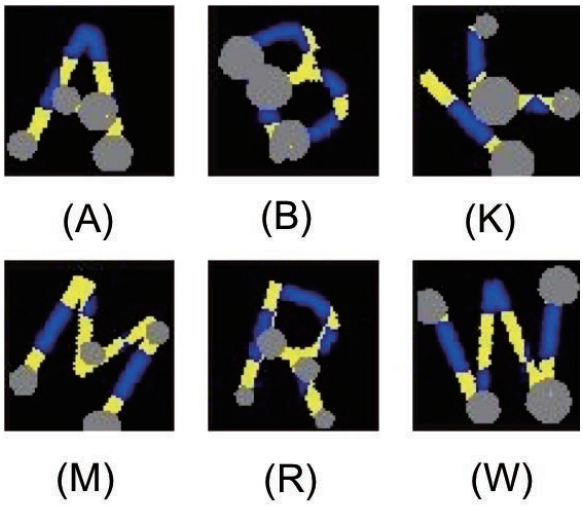


図 3 使用した 6 文字のデータ
Fig. 3 6 character data.

を示している [8]．このときの sign は符号を ±1 の 2 値に量子化するステップ関数を表す．この手法は高速勾配符号法と呼ばれる．

$$\eta = \epsilon \cdot \mathrm{sign}(\nabla_x J(\theta, x, y)) \tag{1}$$

η は RGB の各画素が $\pm\epsilon$ の 2 値となっている画像を意味する．この η を本論文では**敵対的フィルタ**または**摂動フィルタ**，ϵ を**摂動フィルタ幅**と呼ぶ．

提案手法では，Goodfellow らの手法が対象としたネットワークよりも，より深い CNN に対して有効な敵対的サンプルを作成するため，まず，高速勾配符号法を用いて，初期の摂動フィルタを作成する．すなわち，モデルに誤差逆伝播法を適用し，画像の各画素に対して，得られた勾配から符号（式では sign）を求める．符号が正だった場合は $+\epsilon$，符号が負だった場合には $-\epsilon$ とする．次に，作成した摂動フィルタをもとの画像に重ね合わせ，CNN に再度入力し，正解の文字に対する認識率がどの程度下がっているかを見る．たとえば 0.5 など，設定した認識率に至っていなければ，重ね合わせた画像に対して再度摂動フィルタを計算し，同様の重ね合わせを繰り返す．摂動フィルタを重ね合わせるときの式を式 (2) に示す．

$$x_n = x_{n-1} + \epsilon \cdot \mathrm{sign}(\nabla_{x_{n-1}} J(\theta, x_{n-1}, y))$$
$$(\text{while prediction probability} < \text{recognition ratio})$$
$$(n = 1, 2, 3 \ldots, N) \tag{2}$$

摂動フィルタを重ね合わせた回数を通常のニューラルネットの学習に習い**エポック数**と呼ぶ．設定した誤認識率に至った時点で，最終的な CAPTCHA の完成とする．本論文では，最終的に作成された摂動フィルタを複数枚重ね合わせたものを**敵対的ノイズ**，敵対的ノイズを元画像に重ね合わせた CAPTCHA を **Adversarial CAPTCHA** と

Algorithm 1 Calculate Adversarial Noise

Require: $P_{\mathrm{CNN}}(x, l)$: Prediction probability of CNN as a function of image x and label l.
Input: Image x and step width ϵ.
Output: Adversarial noise $z - x$.

```
l ← argmax_{l'} P_CNN(x, l')
z ← x
n ← 1
while n ≤ N do
  η(z) = ε · sign(∇_z J(θ, z, l))
  z ← z + η(z)
  if P_CNN(z, l) < a given recognition ratio (e.g., 0.5) then
    break
  end if
  n ← n + 1
end while
```

Return: $z - x$

呼ぶ．敵対的ノイズの作成の擬似コードをアルゴリズム 1 に示す．

最適なステップ幅 ϵ^* は，次のようにして決定する．まず，候補となるステップ幅を複数用意しておく．その後，ステップ幅を大きいものから順に用いて，アルゴリズム 1 に従い，敵対的ノイズを作成していく．その際，平均二乗誤差（MSE）や平均絶対値誤差（MAE）などを用いて，作成された敵対的ノイズの量を計測し，それがあらかじめ定めておいたしきい値を初めて下回った時点のステップ幅 ϵ を最適なステップ幅 ϵ^* として出力する．その擬似コードをアルゴリズム 2 に示す．

こうすることで，ステップ幅のすべての候補のうちで，敵対的ノイズの量がしきい値を下回るもののうち，最大の ϵ を求めることができる．一般に，ステップ幅が大きいほど，敵対的ノイズを作成するためのエポック数は少なくてすむため，計算量が少なくなる．したがって，最適なステップ幅 ϵ^* として，作成される敵対的ノイズの量がしきい値を下回るもののうち，計算量が最も少ないようなものが求まることになる．

Algorithm 2 Optimize Step Width

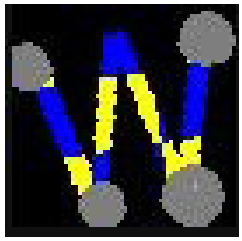
Require: `AdvNois`(x, ϵ) : Adversarial noise calculated for image x and step width ϵ (Alg. 1).
E: list of candidates for step widths.
Input: Image x and threshold t.
Output: Optimized step width ϵ^*.

```
sort E in descending order
for i ← 0 to the length of E −1 do
  if MSE(AdvNois(x, E[i])) < t then
    ε* ← E[i]
    break
  end if
end for
```

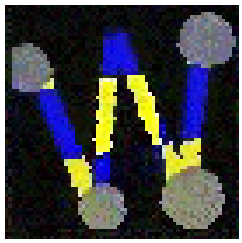
Return: ϵ^*

作成した敵対的ノイズを**図 4**(a)，元の画像を (b)，(a) を (b) に重ねてできた Adversarial examples を (c) に示す．

(a) 敵対的ノイズ：η
(a) Adversarial noise: η



(b) 元の画像：x
(b) original sample: x



(c) 敵対的ノイズを重ね合わせてでき上がった画像
(c) Aversarial CAPTCHA generated with adv.noise

図 4 Adversarial CAPTCHA
Fig. 4 Adversarial CAPTCHA.

Goodfellow らが作成した Adversarial examples は敵対的ノイズを 1 枚合成したものであるが，我々はこれを最大で 40 エポック行う．つまり，1 枚の画像に対し，最大で N= 40 枚の敵対的ノイズを足して Adversarial examples を作成する．

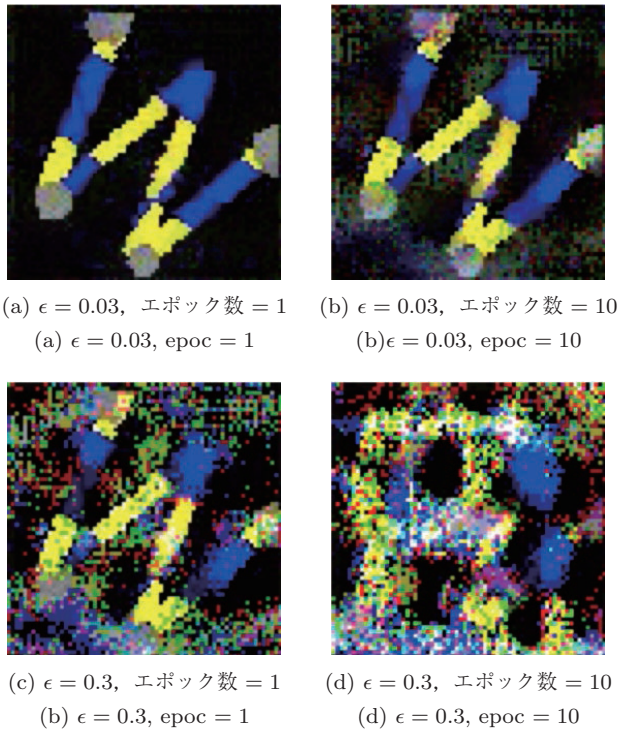
5. 実験と評価

5.1 摂動フィルタ幅の決定

はじめに small dataset のうち 1 文字を使用し，異なる摂動フィルタ幅で Adversarial CAPTCHA を作成した．使用した文字は W である．使用した摂動フィルタ幅は，ϵ =0.001，0.003，0.007，0.01，0.03，0.07，0.1，0.3 の 8 通りである．この Adversarial examples を作成した同じモデルを用いて，予測精度がどのように推移するかについて実験を行った．さらに，平均 2 乗誤差（mean squared error：MSE）と平均絶対誤差（Mean Absolute Error：MAE）についての値も同時に計算を行った．平均 2 乗誤差は今回の場合 0 に近いほど敵対的ノイズの影響が小さいということになる．平均絶対誤差は予測値が正解から平均的にどの程度離れているかを示す値であり，モデルの予測精度の「悪さ」

表 2 摂動フィルタ幅ごとの MSE と MAE
Table 2 MSE and MAE per perturbation filter width.

#step_width	#under50%epoc	#under50%MSE	#under50%MAE	#under10%epoc	#under10%MSE	#under10%MAE
0.001	unknown	unknown	unknown	unknown	unknown	unknown
0.003	36	0.006	0.067	40	0.007	0.073
0.007	17	0.007	0.071	19	0.008	0.078
0.01	12	0.007	0.071	14	0.009	0.080
0.03	6	0.013	0.096	6	0.013	0.096
0.07	4	0.031	0.139	4	0.031	0.139
0.1	3	0.038	0.171	4	0.052	0.177
0.3	2	0.202	0.336	2	0.202	0.336



(a) $\epsilon = 0.03$，エポック数 $= 1$ (b) $\epsilon = 0.03$，エポック数 $= 10$
(a) $\epsilon = 0.03$, epoc $= 1$ (b)$\epsilon = 0.03$, epoc $= 10$

(c) $\epsilon = 0.3$，エポック数 $= 1$ (d) $\epsilon = 0.3$，エポック数 $= 10$
(b) $\epsilon = 0.3$, epoc $= 1$ (d) $\epsilon = 0.3$, epoc $= 10$

図 5 摂動フィルタ幅 0.03 と 0.3 の劣化度の違い
Fig. 5 Difference in degradation degree between perturbation filter width 0.03 and 0.3.

を表す．これは 0 に近い値であるほど優れているモデルであるといえる．結果を**表 2** に示す．

図 5 の (a) は摂動フィルタ幅 0.03，エポック数 1，(b) は摂動フィルタ幅 0.03，エポック数 10，(c) は摂動フィルタ幅 0.3，エポック数 1，(d) は摂動フィルタ幅 0.3，エポック数 10 である．図 5 から摂動フィルタ幅が大きいほど劣化が早いことが見てとれる．

表 2 にあるように，摂動フィルタ幅が小さいほど MSE と MAE は数値が小さいことが分かった．この結果は，作成される敵対的ノイズは摂動フィルタ幅が小さいほど見た目が劣化せず優秀であることを示している．しかし，摂動フィルタ幅が 0.001 のものはエポック回数が上限の 40 になっても予測精度が 50%を下回ることはなかった．さらに，摂動フィルタ幅が大きいほど誤分類させるまでのエポック数は少なくて済むことが分かった．また，図 5 から

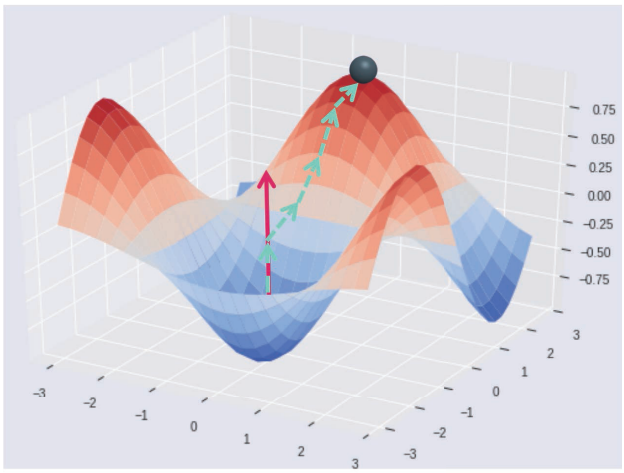


図 6 摂動フィルタ幅の更新方法の違いによる勾配の向き
Fig. 6 Direction of gradient due to difference in updating method of ε.

エポック数が少ないほど見た目が劣化しないことが分かるが，図 5 (b)，(c) を見比べるとエポック数がたとえ 1 だとしても摂動フィルタ幅が大きい方が画像の劣化が激しい．これは，エポック数=1 として摂動フィルタ幅を変えるだけでは必要以上に大きくノイズが乗り過ぎてしまい，見た目が大きく損なわれてしまうということである．合わせて表 2 の予測精度が 10%を下回る際の摂動フィルタ幅 0.03 と 0.3 を見ると，摂動フィルタ幅 0.03 はエポック数 6 で予測精度が 10%を下回っているが，摂動フィルタ幅 0.3 はエポック数 2 である．これは摂動フィルタ幅 0.03 は図 5 の (b) のときも劣化が少ない状態で予測精度が 10%を下回ることになり，摂動フィルタ幅 0.3 は図 5 の (c) よりも劣化した状態でないと予測精度が 10%を下回らないということになる．なぜこのようなことが起こるのかについては**図 6** を参考されたい．図 6 の赤くなっている部分は損失が大きいことを示しており，青くなっている部分は損失が小さいことを示している．また，矢印はエポック数を示している．Adversarial examples を作成するときは．入力画像の損失が大きくなる方向に勾配を進めようとする．そのときどれくらい損失が大きい方向に進むのかを決める値が摂動フィルタ幅である．これを一度に大きくしてしまうと損失は大

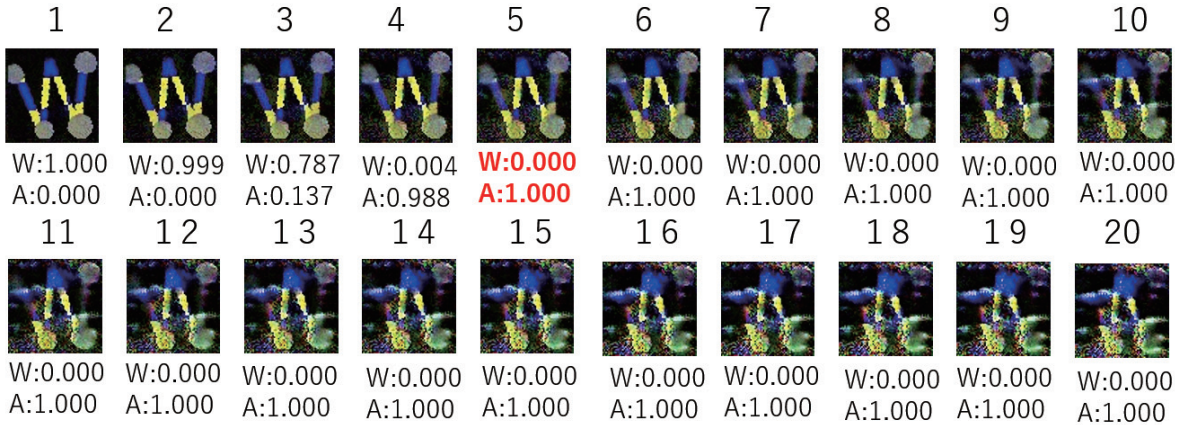


図 7 摂動フィルタ幅 0.07 のときの Adversarial example の変化
Fig. 7 Adversalial example change when the perturbation filter width is 0.07.

きくなるものの損失の頂上には到達しない．図 6 の赤い矢印を見ると損失が増えているものの頂上には達していないことが見てとれる．また，摂動フィルタ幅が大きいと 1 回に載るノイズが多いために画像の劣化が激しくなる．青い矢印のように小さな摂動フィルタ幅でエポックを繰り返しながら損失が大きくなる方向へ導いていくと，画像の劣化は小さいまま損失の頂上まで登ることが可能となる．

以上のことから摂動フィルタ幅は小さければ良いということではなく，さらに摂動フィルタ幅が大きすぎても良くないことが分かる．これにより，MSE，MAE がともに小さい値でさらに予測精度が早い段階で落ちる摂動フィルタ幅が適切であるといえる．

表 2 を見ると，0.007 と 0.01 は 50%を下回ったときの MSE，MAE は同じ値であるにもかかわらず，必要なエポック数は 0.01 の方が少ない．さらに，予測精度が 10%を下回るときの MSE，MAE を比べても摂動フィルタ幅 0.007 と 0.01 の差はほとんどないが，エポック数は摂動フィルタ幅 0.01 の方が 5 回早く 0.007 の MSE と MAE の値に到達している．また，予測精度が 50%を下回るときの摂動フィルタ幅 0.07 と 0.03 はエポック数が 2 回しか変わらないにもかかわらず，摂動フィルタ幅 0.03 の方が MSE は．018 小さく，MAE も.043 小さい．予測精度が 10%を下回るときも同様である．このことから，用いる摂動フィルタ幅は 0.01 と 0.03 が適切であると結論付けることができる．

図 7 は摂動フィルタ幅が 0.07 でエポック数を重ねていったときの AdversarialCAPTCHA がどう変化するかについてまとめたものである．エポック数が 5 回になったとき完全に W から A へ評価が逆転したことが分かる．さらに，エポック数が 5 回目の画像を人間が見た場合，若干画像の劣化が見てとれるものの W と認識するのにまったく問題がないことが分かる．

5.2 Small dataset での Adversarial CAPTCHA の作成

次に，4.1 節で述べた式 (1) の摂動フィルタ η から敵対的ノイズを作成し，3.1 節で述べたアモーダル補完を利用した CAPTCHA の small dataset に対して評価を行った．式 (1) の y には small data set の文字分である 6 ラベルを指定した．この y には特定のラベルのみを指定することも可能であり，ある特定のラベルに誤認識させることも可能であるが，本研究の目的はどのラベルにかは問わず誤認識させること，特定のラベルを指定したフィルタを作成すると，そのラベルの文字が人間にも視認可能となってしまう可能性があることを考慮して，y の対象を全ラベルとした．摂動フィルタ幅は先の実験により 0.01 と 0.03 が適切であることが分かった．今回は誤分類までのエポック数が少ない 0.03 を用いて実験を行った．

結果を**表 3**，**表 4**，**表 5**，**表 6** に示す．表 3 は 1 エポック学習させたものであり，同様に表 4 は 5 エポック，表 5 は 10 エポック，表 6 は 20 エポック学習させたものである．各表の 1 行目の#W，#B，#K，#A，#M，#R は，敵対的ノイズの影響により，それぞれ ‘W’，‘B’，‘K’，‘A’，‘M’，‘R’ の文字と分類された確率であり，#DMY はダミー画像と分類された確率である．#MSE とは平均 2 乗誤差，#MAE とは平均絶対誤差である．平均 2 乗誤差は測定値のばらつき具合を数量的に表すものであり，今回の場合 0 に近いほど敵対的ノイズの影響が小さいということになる．平均絶対誤差は予測値が正解から平均的にどの程度離れているかを示す値であり，モデルの予測精度の「悪さ」を表す．これは 0 に近い値であるほど優れているモデルであるといえる．各表の 2 行目から 7 行目の ‘W’，‘B’，‘K’，‘A’，‘M’，‘R’ は，それぞれ画像の文字を表している．たとえば，2 行目の ‘W’ に対して#A の値が 1.000 であれば，‘W’ の文字が ‘A’ であると分類された確率が 1.000 ということである．各表の 8 行目から 13 行目の ‘W’，‘B’，‘K’，‘A’，‘M’，‘R’ も同様であるが，2〜7 行目の評価は学習に使用したモ

表 3 敵対的ノイズあり 1 エポックでの予測精度
Table 3 Accuracy in 1 epoch (with Adversarial noise).

#Char	#DMY	#W	#B	#K	#A	#M	#R	#MSE	#MAE
W	0.000	1.000	0.000	0.000	0.000	0.000	0.000	0.001	0.030
B	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.001	0.030
K	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.001	0.030
A	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.001	0.030
M	0.000	0.000	0.000	0.000	0.000	1.000	0.000	0.001	0.030
R	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.001	0.030
W	0.000	1.000	0.000	0.000	0.000	0.000	0.000	—	—
B	0.000	0.000	1.000	0.000	0.000	0.000	0.000	—	—
K	0.000	0.000	0.000	1.000	0.000	0.000	0.000	—	—
A	0.000	0.000	0.000	0.000	1.000	0.000	0.000	—	—
M	0.000	0.000	0.000	0.000	0.000	1.000	0.000	—	—
R	0.000	0.000	0.000	0.000	0.000	0.000	1.000	—	—

表 4 敵対的ノイズあり 5 エポックでの予測精度
Table 4 Accuracy in 5 epoch (with Adversarial noise).

#Char	#DMY	#W	#B	#K	#A	#M	#R	#MSE	#MAE
W	0.033	0.889	0.000	0.000	0.077	0.000	0.001	0.010	0.090
B	0.000	0.000	1.000	0.000	0.000	0.000	0.000	0.010	0.085
K	0.004	0.000	0.000	0.996	0.000	0.000	0.000	0.009	0.081
A	0.052	0.000	0.000	0.000	0.946	0.000	0.001	0.010	0.086
M	0.002	0.000	0.000	0.000	0.000	0.998	0.000	0.010	0.085
R	0.222	0.000	0.001	0.000	0.001	0.001	0.775	0.010	0.087
W	0.000	1.000	0.000	0.000	0.000	0.000	0.000	—	—
B	0.000	0.000	1.000	0.000	0.000	0.000	0.000	—	—
K	0.000	0.000	0.000	1.000	0.000	0.000	0.000	—	—
A	0.000	0.000	0.000	0.000	1.000	0.000	0.000	—	—
M	0.000	0.000	0.000	0.000	0.000	1.000	0.000	—	—
R	0.000	0.000	0.000	0.000	0.000	0.000	1.000	—	—

表 5 敵対的ノイズあり 10 エポックでの予測精度
Table 5 Accuracy in 10 epoch (with Adversarial noise).

#Char	#DMY	#W	#B	#K	#A	#M	#R	#MSE	#MAE
W	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.028	0.137
B	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.030	0.143
K	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.030	0.143
A	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.030	0.144
M	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.033	0.151
R	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.028	0.136
W	0.000	0.999	0.000	0.000	0.000	0.000	0.000	—	—
B	0.000	0.000	1.000	0.000	0.000	0.000	0.000	—	—
K	0.922	0.011	0.001	0.014	0.049	0.003	0.000	—	—
A	0.990	0.000	0.000	0.000	0.010	0.000	0.000	—	—
M	1.000	0.000	0.000	0.000	0.000	0.000	0.000	—	—
R	1.000	0.000	0.000	0.000	0.000	0.000	0.000	—	—

表 6 敵対的ノイズあり 20 エポックでの予測精度
Table 6 Accuracy in 20 epoch (with Adversarial noise).

#Char	#DMY	#W	#B	#K	#A	#M	#R	#MSE	#MAE
W	0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.077	0.223
B	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.085	0.230
K	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.080	0.222
A	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.093	0.244
M	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.094	0.243
R	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.083	0.227
W	0.000	0.000	0.000	0.000	1.000	0.000	0.000	—	—
B	1.000	0.000	0.000	0.000	0.000	0.000	0.000	—	—
K	1.000	0.000	0.000	0.000	0.000	0.000	0.000	—	—
A	1.000	0.000	0.000	0.000	0.000	0.000	0.000	—	—
M	1.000	0.000	0.000	0.000	0.000	0.000	0.000	—	—
R	1.000	0.000	0.000	0.000	0.000	0.000	0.000	—	—

デルと敵対的ノイズを作成するために使用したモデルが同一の場合のものであり，8〜13 行目の評価は学習に使用したモデルと敵対的ノイズを作成するために使用したモデルが異なる場合の評価である．これは，まず，Goodfellow らの実験において，学習に使用したモデルと敵対的ノイズを作成するために使用したモデルが同一の場合であったため [8]，アモーダル補完を利用した CAPTCHA でも同様の結果が得られるか試している．次に，実際に CAPTCHA が攻撃を受ける場合，作成者と攻撃者が同一のモデルを使用することは考えられないので，異なるモデルの場合にはどのような結果となるか確認している．

表 3〜表 6 の結果について説明する．たとえば，表 3 は 1 エポックのものであり，‘W’ の文字は同一モデルの場合でも異なるモデルの場合でも，1.000 の確率で ‘W’ と分類されているが，表 4 では ‘W’ の文字が同一モデルの場合に ‘W’ と分類させる確率が 0.889 となっている．そして，20 エポック学習させた場合には，表 6 に示すように，‘W’ の文字は同一モデルの場合でも異なるモデルの場合でも，1.000 の確率で ‘A’ と分類されている．基本的に，深層学習においてはエポック数を増加させることで分類精度を高めるので，敵対的ノイズを加えることでエポック数の増加とともに誤分類を起こさせている．今回の結果は，この手法がアモーダル補完を利用した CAPTCHA への攻撃に対して有効な対抗策であることを示している．なお，表 6 より，20 エポック目の場合には，同一モデルでも異なるモデルでも，‘W’ の文字は 1.000 の確率で ‘A’ と誤分類され，それ以外の文字は 1.000 の確率でダミーと誤分類されている．つまり，この手法により完全に誤分類させることに成功している．

次に，エポック数の増加とともに，その文字がその文字として分類される確率をグラフにまとめた．これはたとえば，W という文字が ‘W’ であると分類される確率である．このグラフを**図 8** に示す．図 8 の同 ‘W’〜‘R’ が，同じモデルの場合の ‘W’〜‘R’ のそれぞれの文字，異 ‘W’〜‘R’ が，異なるモデルの場合の ‘W’〜‘R’ のそれぞれの文字を示す．横軸はエポック数，縦軸は確率である．いずれの文字もエポック数の増加とともに正しく分類される確率が低下している．40 エポック目の場合まで確認したが，20 エポック目以降には過学習による影響などで確率が変化することはなかった．なお，それぞれの文字が 0.9 以上の確率で誤分類されるエポック数は，同一モデルの場合，‘W’：7，‘B’：9，‘K’：8，‘A’：7，‘M’：6，‘R’：6 であり，異なるモデルの場合，‘W’：16，‘B’：12，‘K’：10，‘A’：10，‘M’：9，‘R’：9 であった．

異なるモデルの場合，エポック数の増加とともに誤分類されるようになる確率の増加が，同一モデルの場合よりも少し遅く，少し緩やかであった．いい換えると，異なるモデルの場合には敵対的ノイズによる影響を与えにくいが，一般的に精度を高めるために行われる 20 エポック程度の学習の範囲内では，誤分類に十分な影響を与えられる結果となった．

なお，‘W’ の文字のみ他の文字よりもグラフの傾きが緩やかであり，表 3〜表 6 の分類でも，‘W’ のみダミー文字

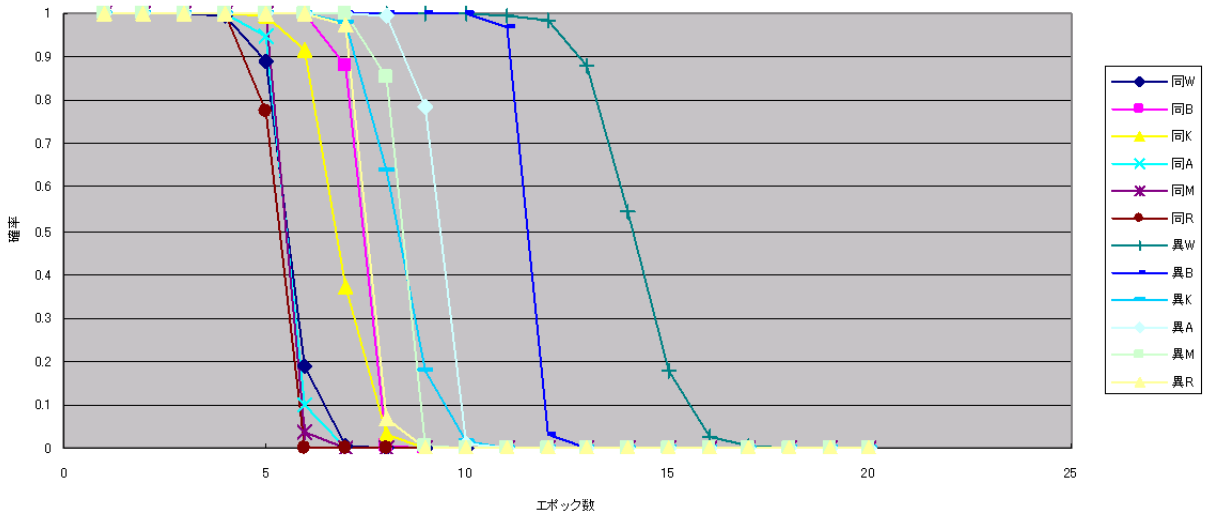
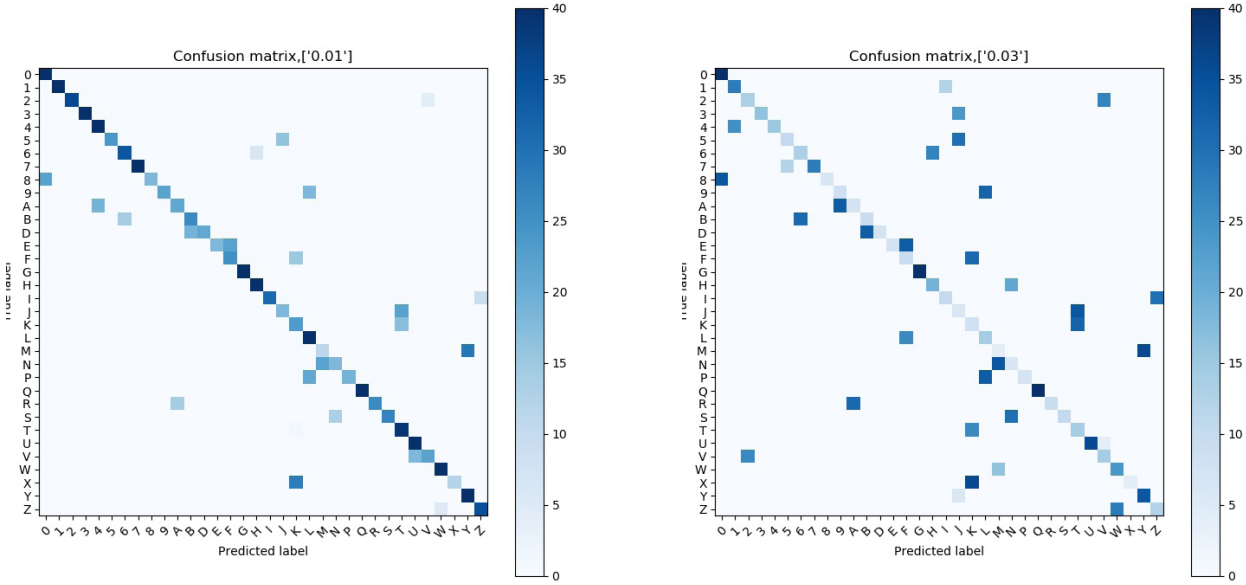


図 8 正しい分類とエポック数の関係
Fig. 8 Relationship between correct classification and epoch.



[1] 摂動フィルタ幅 0.01 の混同行列
[1] Confusion matrix with perturbation filter width of 0.01

[2] 摂動フィルタ幅 0.03 の混同行列
[2] Confusion matrix with perturbation filter width of 0.03

図 9 各摂動フィルタ幅の混同行列
Fig. 9 Confusion matrix of each perturbation filter width.

ではなく ‘A’ に誤分類されている．これはおそらく，‘W’ の文字の一部が ‘A’ の文字の特徴に類似しているためと推測される．敵対的ノイズはすべてのラベルから作成しているが，‘W’ と ‘A’ は文字の特徴が類似しているため，‘A’ の文字の特徴が敵対的ノイズに強く出たのではないかと思われる．

サンプルを ‘A’，‘B’，‘C’，‘D’，‘E’ の 5 つに分類することを考え，‘A’ が正解，それ以外を不正解とする．図 6 の最適解が ‘A’ をターゲットとしている場合，各エポックで求まる接線のベクトルを敵対的ノイズとして減算することにより，分類結果を最適解 ‘A’ から遠ざけることが可能となる（図 6 の青い矢印）．この場合，入力サンプルが ‘A’ と分類される確率が下がり，‘B’，‘C’，‘D’，‘E’ のうちもともと分類される確率が高かったものが正解として選ばれることになる．当然，それは ‘A’ と類似の形状や色などの特徴を多く含むものとなる．このように，正解の ‘A’ に分類される確率を下げる敵対的ノイズのみを乗せた場合，攻撃者は ‘B’，‘C’，‘D’，‘E’ のうち正解と判定された文字と類似の特徴を多く含む文字を答えることで，正解する確率を上げることが可能である．

なお，本論文では実験していないが，図 6 の最適解が ‘B’，‘C’，‘D’，‘E’ をターゲットとするような敵対的ノイズ

表 7 摂動フィルタ幅 0.01 の評価
Table 7 Evaluation of perturbation filter width 0.01.

char	Precision	recall	f1-score	diagonal
0	0.65	1.00	0.78	40
1	1.00	1.00	1.00	40
2	1.00	0.90	0.95	36
3	1.00	1.00	1.00	40
4	0.68	1.00	0.81	40
5	1.00	0.60	0.75	24
6	0.71	0.85	0.77	34
7	1.00	1.00	1.00	40
8	1.00	0.45	0.62	18
9	1.00	0.55	0.71	22
A	0.60	0.53	0.56	21
B	0.58	0.65	0.61	26
D	1.00	0.53	0.69	21
E	1.00	0.45	0.62	18
F	0.53	0.62	0.57	25
G	1.00	1.00	1.00	40
H	0.87	1.00	0.93	40
I	1.00	0.78	0.87	31
J	0.53	0.45	0.49	18
K	0.34	0.57	0.43	23
L	0.51	1.00	0.67	40
M	0.33	0.28	0.30	11
N	0.58	0.45	0.51	18
P	1.00	0.47	0.64	19
Q	1.00	1.00	1.00	40
R	1.00	0.65	0.79	26
S	1.00	0.68	0.81	27
T	0.50	0.97	0.66	39
U	0.69	1.00	0.82	40
V	0.85	0.55	0.67	22
W	0.89	1.00	0.94	40
X	1.00	0.30	0.46	12
Y	0.58	1.00	0.73	40
Z	0.80	0.88	0.83	35

表 8 摂動フィルタ幅 0.03 の評価
Table 8 Evaluation of perturbation filter width 0.03.

char	Precision	recall	f1-score	diagonal
0	0.54	1.00	0.70	40
1	0.53	0.70	0.60	28
2	0.33	0.33	0.33	13
3	1.00	0.40	0.57	16
4	1.00	0.38	0.55	15
5	0.45	0.25	0.32	10
6	0.30	0.33	0.31	13
7	1.00	0.70	0.82	28
8	1.00	0.15	0.26	6
9	0.20	0.20	0.20	8
A	0.18	0.17	0.18	7
B	0.21	0.23	0.22	9
D	1.00	0.17	0.30	7
E	1.00	0.17	0.30	7
F	0.13	0.23	0.17	9
G	1.00	1.00	1.00	40
H	0.41	0.47	0.44	19
I	0.45	0.25	0.32	10
J	0.09	0.15	0.11	6
K	0.08	0.20	0.11	8
L	0.18	0.35	0.24	14
M	0.07	0.10	0.09	4
N	0.11	0.15	0.12	6
P	1.00	0.17	0.30	7
Q	1.00	1.00	1.00	40
R	1.00	0.23	0.37	9
S	1.00	0.25	0.40	10
T	0.17	0.35	0.23	14
U	1.00	0.90	0.95	36
V	0.31	0.35	0.33	14
W	0.46	0.60	0.52	24
X	1.00	0.10	0.18	4
Y	0.49	0.85	0.62	34
Z	0.29	0.30	0.29	12

を作成することも可能である．たとえば，最適解が ‘B’ をターゲットとしている場合，各エポックで求まる接線のベクトルを敵対的ノイズとして加算することにより，分類結果を最適解 ‘B’ に近づけることが可能となる．この場合，入力サンプルが ‘A’ と分類される確率が相対的に下がり，‘C’, ‘D’, ‘E’ はそのままで ‘B’ と分類される確率が上がる．これに先ほど述べた正解の ‘A’ に分類される確率を下げる敵対的ノイズも重畳すれば，高確率で ‘B’ に誤分類されるように誘導できる．このように，任意の文字に誤分類させるような敵対的ノイズを作成することが可能である．このため，ターゲットをランダムに決定することによって攻撃者（= 機械）が Adversarial examples の変化パターンを予測して回答することを防ぐことは容易であるといえる．

6. large dataset での Adversarial examples の検証結果

本章では，3.1 節で述べた文字の数を ‘0-9’, ‘A-Z’（‘O’ と ‘C’ を除く）の 34 文字に拡張した large dataset に Adversarial examples を適用し評価を行った結果について述べる．使用したデータは large dataset のうち ‘correct’ のみとし，‘dummy’ は使用しなかった．そのため学習に使用した総データ数は 40,800 枚である．使用した評価用モデルは Adversarial examples を作成したモデルと同一のものとした．また，摂動フィルタ幅は 0.01 と 0.03 とし，摂動を重ねる回数は 40 とした．作成された CAPTCHA40 枚すべてでテストを行った．**図 9** は評価を混同行列にし可視化したものである．[1] を見ると，対角線上のセルが濃くなっている．これは，Adversarial examples を適用した

CAPTCHA を分類したとき，ほとんどが上手く分類できていることを表す．図 9 の [2] では混同行列の対角線上のセルは，学習済みネットワークによって観測値のクラスが正しく推定されている数を示す．つまり，真のクラスと予測されたクラスが正解と一致する比率を示している．そして，対角線から外れたセルは分類器に誤りが発生している箇所を示す．図 9 の対角線が薄くなっており，対角線から外れたセルが目立つ．これは誤分類が多く発生し上手く分類できていないことを表している．small dataset の場合，摂動フィルタ幅が 0.01 の場合でも早い段階で誤分類が起きていたが 34 文字にした際には早い段階での誤分類は発生しなかった．また，**表 7**，**表 8** に摂動フィルタ幅 0.01，0.03 それぞれの評価を記した．表中の diagonal は 40 エポック中いつ誤分類が始まるかを示している．diagonal に注目すると，同じ摂動を重ねていても誤分類が始まるタイミングが文字によって大きく異なることが分かる．早く誤分類する文字と誤分類を起こさなかった文字には 10 倍の差を見ることができる．特に ‘0’ や ‘G’，‘Q’ はいずれも摂動を 40 枚重ねたとしても誤分類を起こすことなく分類できている．早く誤分類を起こす文字は [8，9，A，B，D，E，F，J，K，M，N，P，R，X] であり，‘B’，‘P’，‘R’ など似た形状を持っている文字が比較的早く誤分類を起こしやすいことが分かった．このような文字に関しては提案方式が有効である．また，5.2 節で述べたターゲットをいずれかの文字にする方法を用いることで，今回誤分類を起こさなかった文字に関しても提案方式を有効にすることが可能である．

7. StirMark による Adversarial examples への攻撃

前章では適切な文字や摂動フィルタ幅を使用すれば，画像を劣化させることなく CNN を騙すことができることが分かった．しかし，摂動を除去することができれば CNN は分類を誤ることなく CAPTCHA を解読することができる．本章では，Adversarial CAPTCHA のノイズ耐性を検証するため，電子透かしの強度判定用ソフトである StirMark を使用して StirMark 適用後の Adversarial CAPTCHA がどの程度 CNN の分類に耐えられるのかについて実験を行った結果について述べる [27]．

7.1 StirMark

StirMark とは Petitcolas らが開発した画像透かしアルゴリズムやその他のステガノグラフィ技術の堅牢性テストのためのソフトウェアである．これはデジタル画像に適用することができ，埋め込まれた透かしまたはステガノグラフィーメッセージを検出して画像から復号することができないように透かしを歪ませる．このアルゴリズムは軽微な幾何学的歪みを適用し，イメージはわずかな伸び，剪断，

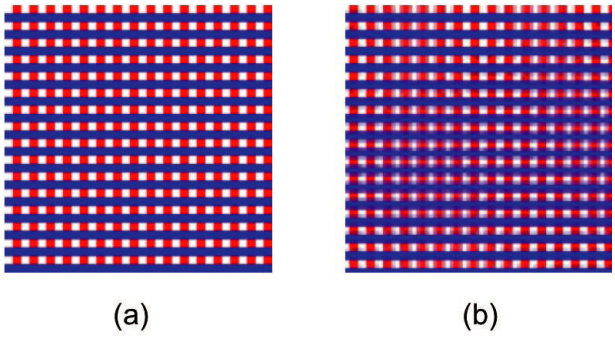


図 10 StirMark 適用後と適用前の検証画像
Fig. 10 Verification image after StirMark application and before application.

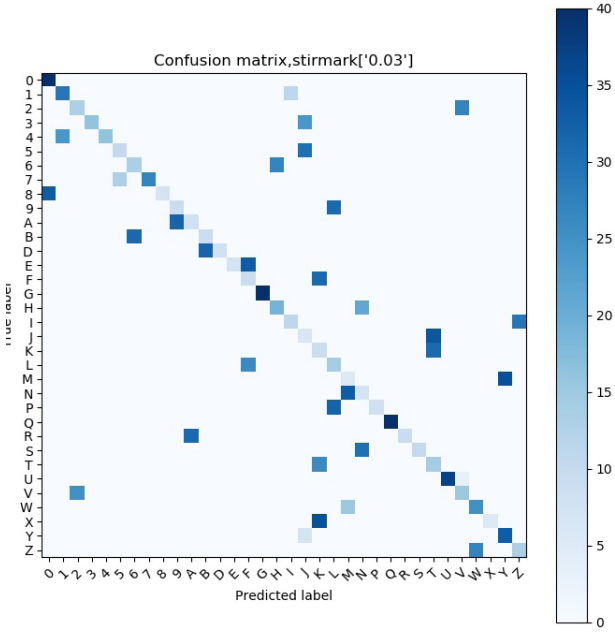


図 11 StirMark を適用後の混同行列
Fig. 11 Confusion matrix after application of StirMark.

シフト，または目立たないランダムな量だけ回転し，ナイキスト補間を使用して再サンプリングする．さらに，すべてのサンプル値に小さく滑らかに分布した誤差を導入する伝達関数が適用される．

7.2 StirMark の検証実験

StirMark はオプションを付けずにデフォルトで実行した．デフォルトで実行する場合，オプションの設定は -‘i’2.0% -‘o’0.7 -‘d’1.5 と同様になる．‘i’ オプションは内側の画像が出力画像の端に移動するものであり，‘o’ オプションは外側の画像（外挿される）が出力画像の端に移動するものである．オプションがデフォルトで実行される場合，これらが相殺されてしまう可能性があるためオリジナル画像と比較して StirMark を通した画像が歪んでいることを実験にて確認した．なお，‘d’ オプションは画素の RGB 値を変更するものであり，画像の歪みには関係しない．

表 9 StirMark 適用後の摂動フィルタ幅 0.03 の評価
Table 9 Evaluation of perturbation filter width 0.03 (with StirMark).

char	Precision	recall	f1-score	diagonal
0	0.55	1.00	0.71	40
1	0.55	0.72	0.62	29
2	0.34	0.33	0.33	13
3	1.00	0.40	0.57	16
4	1.00	0.40	0.57	16
5	0.43	0.25	0.32	10
6	0.30	0.33	0.31	13
7	1.00	0.68	0.81	27
8	1.00	0.17	0.30	7
9	0.22	0.23	0.22	9
A	0.21	0.20	0.20	8
B	0.22	0.23	0.22	9
D	1.00	0.20	0.33	8
E	1.00	0.17	0.30	7
F	0.13	0.23	0.17	9
G	1.00	1.00	1.00	40
H	0.41	0.47	0.44	19
I	0.50	0.28	0.35	11
J	0.09	0.15	0.11	6
K	0.09	0.23	0.13	9
L	0.18	0.35	0.24	14
M	0.09	0.12	0.11	5
N	0.12	0.17	0.14	7
P	1.00	0.20	0.33	8
Q	1.00	1.00	1.00	40
R	1.00	0.23	0.37	9
S	1.00	0.25	0.40	10
T	0.18	0.35	0.24	14
U	1.00	0.93	0.96	37
V	0.33	0.38	0.35	15
W	0.48	0.62	0.54	25
X	1.00	0.12	0.22	5
Y	0.49	0.82	0.61	33
Z	0.31	0.33	0.32	13

まず，**図 10** (a) に示す青と赤の格子模様を持つ 64 × 64 ピクセルの画像を作成し PPM 形式に変換した後 StirMark を-‘i’2.0% -‘o’0.7 -‘d’0 のオプションで実行した．これは，StirMark が PNM（PBM，PGM，PPM の総称）形式しか扱えないためと ‘d’ オプションの影響をなくすためである．変換後の画像を図 10 に示す．

図 10 (b) では格子の間隔はオリジナルから変化していないが，細くなっている箇所と太くなっている箇所があり画像に変化があることが確認できた．

7.3 実験結果

StirMark を適用するために，Adversarial CAPTCHA を ppm に変換し StirMark 適用後に再度 jpg 形式に変換した．jpg に再変換する際は圧縮はしていない．また，CAPTCHA の評価は 6 章と同条件で行った．**図 11** は StirMark を適用した後の評価を混同行列で表したものである．また，**表 9** は StirMark 適用後の評価を示している．図 11 を見ても見た目上はあまり変わらないように見えるが，表 9 の ‘diagonal’ を見ると文字によっては誤分類を起こすタイミングに変化が見られた．しかし，表 9 の ‘diagonal’ を見て分かるとおり 1 回程しか遅延していないことが分かる．この結果から，Adversarial CAPTCHA はノイズ除去に対する耐性があると述べることができる．

8. 結論

今回の敵対的ノイズの重ね合わせにより CNN に確実な誤分類を起こさせることに成功した．よって，アモーダル補完を利用した CAPTCHA は，従来の画像処理技術および CNN のどちらを用いても，他人の一般的なコンピュータを乗っ取った攻撃者には破ることは困難であり，一方で，アモーダル補完により人間には認識しやすいものとなっているといえる．また，文字を増やした場合には誤分類を起こすタイミングが遅くなった．これは，‘dummy’ を用いなかったことによって，必ずどの文字かに分類されるような問題にしたことが原因ではないかと考えられる．今回誤分類を早い段階で起こした文字は ‘2’ や ‘Z’, ‘M’ や ‘N’ などの「似た特徴を持つ文字」が存在する文字であった．誤分類を起こさなかった ‘0’ や ‘G’ などは「似た特徴を持つ文字」が存在しない文字であり，必ずいずれかの文字に分類するような問題だった場合には誤分類を起こさないと考えられる．しかし，誤分類を起こしやすい文字を CAPTCHA として用いることで，Adversarial examples の効果を大いに発揮できることは期待できるといえる．もちろん，高性能なコンピュータを使用すれば，アモーダル補完の演算によりアモーダル補完を利用した CAPTCHA を解読することは可能であるが，ボットが Web サービスのアカウントを次々と取得して，ステルスマーケティングや詐欺などを行うことを阻止できるという意義は大きい．また，StirMark を利用した Adversarial CAPTCHA への攻撃も，誤分類するタイミングが 1 回遅延する程度に収まったことから，ノイズ除去に関しても耐性があるということができる．機械学習には SVM（Support Vector Machine）や SOM（Self Organizing Map），また畳み込み層を使わない深層学習での解析は理論的に敵対的ノイズの影響はないため識別は可能である．しかし，識別精度の点から画像認識には深層学習，なかでも CNN が現在は最も一般的に使われていることは明らかであり，自動運転技術の研究やスマートフォンのアプリケーションなどはすでに存在する．そのため，我々は本研究において SVM や SOM ではなく CNN を騙せるかどうかに焦点を当てた．

今後の課題としては，今回の実験では CNN における同一のネットワークを使用して，誤分類が起きるかどうか確

認したが，ネットワークを変更された場合に同様の誤分類が起きるかどうかを調べる必要がある．画像認識に用いられるネットワークの構造は類似しているため，おそらく，いくつかのネットワークで作成した敵対的ノイズを重複して用いることにより対応は可能と思われる．

謝辞 本研究は JSPS 科研費 JP18K11248 の助成を受けたものです．

参考文献

[1] Turing, M.A.: Computing Machinery and Intelligence, *Journal of the Mind Association* (*MIND*), Vol.LIX, No.236, pp.433–60 (1950).

[2] Coates, A.L., Baird, H.S. and Faternan, R.J.: Pessimal print: A reverse Turing test, *Proc. 6th International Conference on Document Analysis and Recognition*, pp.1154–1158 (2001).

[3] Kanizsa, G.: Original title: Grammatica del vedere, *La Grammaire du Voir. Diderot* (1996).

[4] Mori, T., Uda, R. and Kikuchi, M.: *Proposal of Movie CAPTCHA Method Using Amodal Completion, Proc. SAINT 2012*, pp.11–18 (2012).

[5] Sawada, K. and Uda, R.: *Effective CAPTCHA with Amodal Completionand Aftereffects, Proc. IMCOM '16*, Article No.53 (2016).

[6] Azakami, T. and Uda, R.: *Effective CAPTCHA with Amodal Completion and Aftereffects by Complementary Colors and Difference of Luminance, Proc. AINA-2016*, pp.232–237 (2016).

[7] Sivakorn, S., Polakis, I. and Keromytis, A.D.: *I Am Robot: (Deep) Learning to Break Semantic Image CAPTCHAs, Proc. EuroS&P* (2016).

[8] Goodfellow, I.J., Shlens, J., Szegedy, C., *Explaning and Harnessing Adversarial Examples, Proc. ICLR* (2015).

[9] Mori, G. and Malik, J.: Recognizing objects in Adversarial clutter: Breaking a visual CAPTCHA, *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003*, Vol.1, pp.I-134–I-141 (2003).

[10] Chellapilla, K. and Simard, P.Y.: Using Machine Learning to Break Visual Human Interaction Proofs (HIPs), *Advances in Neural Information Processing Systems*, Vol.17, pp.265–272 (2004).

[11] Sivakorn, S., Polakis, J. and Keromytis, A.D.: I'm not a human: Breaking the Google reCAPTCHA, *Black Hat ASIA 2016* (2016).

[12] LeCun, Y.: Generalization and Network Design Strategies, Technical Report.CRG-TR-89-4 (1989).

[13] LeCun, Y., Boser, B., Denker, S.J., et al.: Backpropagation applied to handwritten zip code recognition, *Neural Computation 1*, pp.541–551 (1989).

[14] KarSimonyan, S. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition, *Intl. Conf. Learning Representations* (*ICLR*) (2015).

[15] George, D., Lehrach, W., Kansky, K., et al.: A generative vision model that trains with high data efficiency and breaks text-based CAPTCHAs, *Science*, Vol.358, No.6363 (2017).

[16] Krizhevsky, A., Sutskever, I. and Hinton, G.E.: *ImageNet Classification with Deep Convolutional Neural Networks* (2012).

[17] Szegedy, C., Liu, W., Jua, Y., Sermanet, P., Reed, S., Angguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A.: *Going Deeper with Convolutions* (*CVPR2015*) (2015).

[18] Simonyan, K. and Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recoginition, *Intl. Conf. Learning Representations* (*ICLR*) (2015).

[19] Azakami, T., Shibata, C. and Uda, R.: Challenge of Deep Learning against CAPTCHA with Amodal Completion and Aftereffects by Colors, *The 19th International Conference on Network-Based Information Systems* (*NBiS 2016*), pp.127–134 (2016).

[20] Szegedy, C., Zaremba, W., Sutskever, B.A., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, B.: Intriguing properties of neural networks, *International Conference on Learning Representations* (*ICLR 2014*), abs/1312.6199 (2014).

[21] Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z.B. and Swami, A.: Practical Black-Box Attacks against Machine Learning, arXiv preprint arXiv:1602.02697 (2016).

[22] Papernot, N., McDaniel, P. and Goodfellow, I.: Transferability in Machine Learning: From Phenomena to Black-Box Attacks using Adversarial Samples, arXiv preprint arXiv:1605.07277 (2016).

[23] Kurakin, A., Goodfellow, I. and Bengio, S.: Adversarial examples in the physical world, *International Conference on Learning Representations* (*ICLR 2017*), arXiv:1607.02533 (2017).

[24] Lu, J., Sibai, H., Fabry, E. and Forsyth, D.: NO Need to Worry about Adversarial examples in Object Detection in Autonomous Vehicles, arXiv:1707.03501v1 (2017).

[25] Athalye, A., Engstrom, L., Ilyas, A. and Kwok, K.: Synthesizing Robust Adversarial examples, *Under Review as a Conference Paper at ICLR 2018*, arXiv:1707.07397v2 (2018).

[26] Osadchy, M., Hernandez-Castro, J., Gibson, S., Dunkelman, O. and Pérez-Cabo, D.: No Bot Expects the DeepCAPTCHA! Introducing Immutable Adversarial Examples, With Applications to CAPTCHA Generation, *IEEE Trans. Information Forensics and Security*, Vol.12, No.11, pp.2640–2653 (2017).

[27] Petitcolas, A.P., Anderson, R.J. and Kuhn, M.: Attacks on copyright marking system, *2nd International Information Hiding Workshop*, Portland, USA (1998).

推薦文

DICOMO2017 の発表論文の中でも評価が高く，優秀プレゼンテーション賞を受賞している．特に，ニューラルネットを利用したキャプチャ解読を防ぐ手法の有効性を十分な精度で示していることから，推薦する．

（セキュリティ心理学とトラスト研究会主査　寺田真敏）

阿座上 知香

2017 年東京工科大学卒業．東京工科大学大学院修士課程在学，専門は機械学習の応用とネットワークセキュリティ．

柴田 千尋

東京大学工学部卒業．東京大学大学院工学系研究科博士課程修了後，東京大学大学院情報学環特任研究員等を経て，2016 年より東京工科大学バイオ情報メディア研究科専任講師．専門は機械学習の理論と応用．博士（工学）．

宇田 隆哉 （正会員）

1998 年慶應義塾大学理工学部計測工学科卒業．2000 年同大学大学院理工学研究科計測工学専攻前期博士課程修了．2002 年同大学院理工学研究科開放環境科学専攻後期博士課程修了．博士（工学）．現在，東京工科大学コンピュータサイエンス学部講師．ネットワークセキュリティの研究に従事．2002 年 IFIP/SEC 2002 Best Student Paper Award 受賞．電子情報通信学会会員．