

QR コードの認識特性とその脆弱性について

橋本 樹^{1,*} 村井 祐斗¹ 西井 大智¹ 中嶋 祥吾¹
白石 善明¹ 森井 昌克¹

概要: スマートフォン (スマホ) の流通によって、いつでもどこでも撮影でき、その効用によって QR コードの利用が加速し、単なる識別符号の枠を超えて、URL への誘導、電子決済や個人識別での認証利用等様々な用途に使われている。現在、スマホのカメラは QR コードデコーダ (認識ソフト) と連携しており、一部のスマホではカメラを向けると自動的に QR コードを認識し、記載されている URL を表示するだけでなく、その URL を有するサイトにアクセスする設定となっている。QR コードは視認によってその内容の正誤が判断できないゆえに、想定しない認証や悪意のあるサイトへの誘導が問題となっている。したがって QR コードにおいては、その内容を精査した上で、認証やサイトへの誘導を許可することが強く望まれる。本稿では QR コード自体を視認できない、いわゆる見えない QR コードの構成法を与える。実際、まったく QR コードと視認できない画像を構成し、一部のスマホでは、その画像を QR コードと誤認し、さらに任意のサイトに誘導できることを実証する。本稿で提案する「見えない QR コード」は AI でいわれるところの Adversarial Example に相当し、スマホに搭載される QR コードデコーダの脆弱性を利用して誤認識させる。

キーワード: Adversarial Example, QR コード, 位置検出パターン

Recognition Characteristics of QR Code and Their Vulnerabilities

Itsuki Hashimoto^{1,*} Yuto Murai¹ Daichi Nishii¹ Shogo Nakajima¹
Yoshiaki Shiraishi¹ Masakatu Morii¹

Abstract: With the distribution of smartphones, it is possible to take pictures anytime and anywhere, and this utility has accelerated the use of QR codes for a variety of purposes. Currently, smartphone cameras are linked to QR code decoders, and some smartphones are set to automatically recognize a QR code when the camera is pointed at it and not only display the URL, but also access the website with the URL. Since QR codes cannot determine the correctness or incorrectness of their contents, unexpected authentication and guidance to malicious sites have become problems. Therefore, it is highly desirable to allow authentication and guidance to a site after carefully examining the contents of QR Code. In this paper, we give a method of constructing a so-called invisible QR Code, in which the QR Code itself cannot be seen. We will demonstrate that some smartphones can misidentify the image as a QR code.

Keywords: Adversarial Example, QR code, Finder Pattern

1. まえがき

スマートフォンの流通によって、いつでもどこでも撮影できるようになり、その効用によって QR コード[1]の利用が加速し、単なる識別符号の枠を超えて、URL への誘導、電子決済や個人識別での認証利用等様々な用途に使われ、さらに活用が期待されている。現在、スマートフォンのカメラは QR コードデコーダ (認識ソフト) と連携しており、一部のスマートフォンではカメラを向けると自動的に QR コードを認識し、さらに URL を記載されていればその URL を表示するだけでなく、その URL を有するサイトにアクセスする設定となっている。QR コードは視認によってその内容の正誤が判断できないゆえに、想定しない認証や悪意のあるサイトへの誘導が問題となっている。したがって QR コ

ードにおいては、その内容を精査した上で、認証やサイトへの誘導を許可することが強く望まれる。

本稿では QR コード自体を視認できない、いわゆる見えない QR コードの構成法を与える。実際、まったく QR コードと視認できない画像を構成し、一部のスマートフォンでは、その画像を QR コードと誤認し、さらに任意のサイトに誘導できることを実証する。本稿で提案する「見えない QR コード」は AI でいわれるところの Adversarial Example[2] に相当し、スマートフォンに搭載される QR コードデコーダの脆弱性を利用して誤認識させる。Adversarial Example は人間の目には元の画像と同じか、せいぜいランダムなノイズがのっている程度にしか見えないが、モデルによる認識は元の入力のものとは大きく変化する。このような意図的に変化させた入力を用いてモデルへの出力を誤らせるこ

¹ 神戸大学
Kobe University
* hhashimoto@stu.kobe-u.ac.jp

とは、一種の攻撃であると見なすことができる。文献[3]では、自動運転システムに搭載された道路標識などを自動で判別する画像認識システムに対し、道路標識に特殊な光線を照射することでシステムに誤認させることを実証した。これらのことから、モデルに対する Adversarial Example を生成し、モデルの脆弱性の評価を行うことは重要である[4]。QR コードにおいても、Adversarial Example が適応可能であれば意図せずカメラが QR コードを誤認識する危険性があり、悪意のある第三者による不正な使用が考えられる。そのため本稿では、QR コードにおける Adversarial Example の実現可能性とその安全性を評価する。

2. QR コードの脆弱性と Adversarial Example

QR コードの認識率は非常に高いが、デザインとしては単に黒と白の正方形が並べられた画像データであるため、人が QR コードに格納された情報をデータとして直接解釈することはできない。つまり、QR コードの一部の内容を書き換えられていたとしても気づくことはできない。また、多くの利用者は QR コードに格納された情報は正しいと信用するため、これを利用して悪意のある第三者が偽装した QR コードを作成し、それを読み取った利用者が悪意のあるサイトに誘導されることが問題となっている。

偽装 QR コードの例として、誤り訂正符号の性質を用いて、二つの情報を出力する QR コード[5],[6]が提案されている。文献[5]では、QR コードの一部を書き換えた上でさらにノイズを一つあるいは複数のモジュールに追加することで、意図的に低い確率で本来格納されている情報とは別の情報を出力することを実証した。つまり、偽装 QR コードを読み取ることで一定の確率で悪性サイトの URL を出力し、そのサイトに誘導可能であることを示した。また、この偽装 QR コードは低い確率でしか悪意のある URL を出力されないため、この偽装 QR コードによって被害にあった被害者が、もう一度偽装 QR コードを読み取ったとしても正規の URL が表示されるため、再現性がなく偽装 QR コードが被害の原因であることを気づくことができない。そのため悪意のある第三者がこの方法で偽装 QR コードを作成すると、偽装 QR コードの発見が遅れるため被害の拡大が考えられる。これらより、偽装 QR コードの実現が QR コードの脆弱性と言える。

また、QR コードにおいて、Adversarial Example に適応することによる大きな脆弱性も考えられる。図 1 は 57.7% の確率でパンダとして認識されていた画像に小さなノイズを加えた結果、目視では違いがわからないが分類機には 99.3% の確率でテナガザルと認識された Adversarial Example の例である。

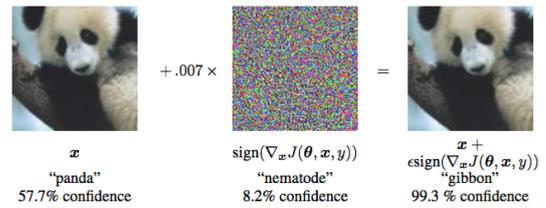


図 1 Adversarial Example の例[7]

図 1 の Adversarial Example が人の目ではパンダと認識しても分類機ではテナガザルと認識するように、今回提案する見えない QR コードは、人の目では QR コードに見えないが、カメラを向けると QR コードデコーダが QR コードと認識するような画像である。このような見えない QR コードが実現可能であれば、カメラが意図しない QR コードを誤認識する危険性がある。さらに、悪意のある第三者によって見えない QR コードが作成されれば、悪意のあるサイトの URL を意図せず出力し、その URL を有するサイトに強制的にアクセスさせられる可能性があり、QR コードの大きな脆弱性に繋がる。

3. QR コードの基本事項

3.1 QR コードについて

QR コードは、1994 年に株式会社デンソーが開発した二次元コードの 1 つである。現在、特許は株式会社デンソーウェブが保有しているが、デンソーウェブは特許権を行使しないことを宣言している。そのため日本国内だけでなく国際的にも規格化されており、誰でもその仕様を入手することができ、現在では携帯端末でウェブページにアクセスする際の URL の情報を読み取るなど広く一般的に用いられている。

QR コードは従来の一次元コードに比べて、より多くの情報を格納することができ、数字だけでなく英数字や漢字など複数の言語にも対応している。また QR コードは高速に読み取ることが可能であり、一定量のノイズが生じたとしてもリード・ソロモン符号の誤り訂正能力により検出訂正できるため、格納された情報を正確に読み出すことが可能である。

3.2 QR コードの構成要素

QR コードは位置検出パターン、位置合わせパターン、タイミングパターン、データコード語、誤り訂正コード語を持ち、これらをモジュールで構成する。以上の各部分は図 2 で示したように、指定された場所に配置される。また、型番 (バージョン) や誤り訂正レベルによって QR コードに格納可能な情報量や誤りに対する訂正能力が変化する。

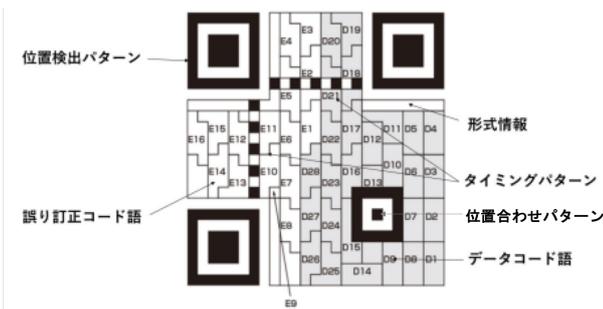


図 2 QR コードの構成

- モジュール
QR コードを構成する白または黒の正方形で、1 モジュールが 1bit を表す。
- 位置検出パターン
QR コードの 3 つの隅 (左上, 左下, 右上) に必ず配置される。位置検出パターンの中心を通る線は、どの方向からでも黒モジュールと白モジュールの比率が 1:1:3:1:1 になっており、どの方向からであっても適切に検出可能である。3 つの位置検出パターンを識別することで、画像内の QR コードを検出する。
- 位置合わせパターン
QR コードのバージョンによって決められた位置に配置される。位置合わせパターンは、歪みによって生じるモジュールの位置ずれの補正に利用される。
- タイミングパターン
黒モジュールと白モジュールを交互に配置し、QR コードのバージョンを決定するために利用される。
- データコード語と誤り訂正コード語
データコード語が実際の情報のデータであり、誤り訂正コード語がノイズに対応するための誤り訂正用のデータである。
- 型番 (バージョン)
QR コードの大きさを表す値で、型番が大きくなるほど QR コードの 1 辺のモジュール数が多くなり、格納可能な情報量も増加する。型番は 1 から 40 まで設定されている。
- 誤り訂正レベル
誤り訂正レベルは、L, M, Q, H の 4 段階に設定されており、最大でそれぞれ QR コード全体の約 7%, 15%, 25%, 30% の誤りを訂正可能で、同一の型番では誤り訂正能力が大きくなるほど格納可能な情報量は小さくなる。

4. QR コードの認識

4.1 QR コードの復号

本節では、オープンソースの QR コード読み取りライブラリである ZXing[8]における QR コードの復号アルゴリズムを示す。

Step1 : 画像の二値化

カメラで得たカラー画像のピクセルの明度を計算し、その後画像をいくつかのブロックに分け、それぞれのブロックの明度の平均を閾値として、画像内の各ピクセルを明暗の二つに分ける。この明度は、画像上のピクセルの RGB 値をそれぞれ r , g , b として明度 y を以下の式(1)で導出する。

$$y = 0.299 \times r + 0.587 \times g + 0.114 \times b \quad (1)$$

Step2 : 位置検出パターンの探索

二値化された画像内から、位置検出パターンを複数個探索し、その中から最適な三つの位置検出パターンを決める。探索した位置検出パターンが二つ以内あるいは最適な三つの位置検出パターンが見つからなければ、Step1 に戻り再度新しい画像でアルゴリズムを再開する。

Step3 : QR コードの切り出し

探索した三つの位置検出パターンにより、画像内から QR コードを抽出する。

Step4 : 座標の計算

抽出した QR コードに歪みがある可能性があるため、射影変換を行い、QR コードの歪みを補正する。

Step5 : モジュールの明暗判定

補正した QR コードから各モジュールの明暗を判定し、検出した QR コードの情報を読み取る。

Step6 : データの復号

読み取った情報に誤りが含まれている可能性があるため、誤り訂正コードを利用し実際に格納されたデータを復号する。誤り訂正能力を超えた誤りが含まれている場合、復号することはできず、Step1 に戻り再度新しい画像でアルゴリズムを再開する。

4.2 位置検出パターンの認識

本節では、オープンソースの QR コード読み取りライブラリである ZXing における、位置検出パターンの認識手順を示す。

Step1

二値化した画像を、図 3 の赤色で可視化したように、高さ 3 ピクセル毎に水平方向の全ピクセルの探索を行う。その際、黒白黒白黒の順にそれぞれ連続で並ぶ色のピクセル数をカウントする。カウントした黒白黒白黒のパターンの中で、1:1:3:1:1 になっている箇所を検出する。

Step2

黒白黒白黒の順にピクセル数が1:1:3:1:1になっている箇所を検出した場合、例えば図4の(a)のような場合、その中心の座標を位置検出パターンを中心の座標と仮定する。ただし、カメラで撮影している場合、手ぶれなどによってQRコードが歪む可能性が想定されるので、位置検出パターンの横方向の黒白黒白黒のピクセル数が必ずしも1:1:3:1:1であるとは考えられない。そのため横方向の黒白黒白黒のピクセル数が1:1:3:1:1であることに対してある程度の検出の幅が存在する。具体的には、カウントしたピクセルは7つのモジュール分のピクセル数となっており、これから1つのモジュールの平均のピクセル数を計算し、その値とカウントした全てのモジュールのピクセル数の差がそれぞれ計算した値の50%以内であれば、カウントした黒白黒白黒のピクセル数が1:1:3:1:1になっていると見なす。

Step3

Step2で仮定した座標の縦方向に対して、黒白黒白黒の順にそれぞれ連続で並ぶ色のピクセル数をカウントする。カウントした黒白黒白黒のパターンが図4の(b)のように1:1:3:1:1になっていることを確認する。なっていないならばStep1に戻りピクセルの探索を再開する。ただし、縦方向にカウントした黒白黒白黒のピクセル数が1:1:3:1:1になっていることの必要十分条件は、Step2でカウントした横方向と、縦方向の合計ピクセル数の差が横方向の合計ピクセル数の40%以内であること、である。

Step4

Step2で仮定した座標を中心にして、左上から右下に向く斜め方向に対して、黒白黒白黒の順にそれぞれ連続で並ぶ色のピクセル数をカウントする。カウントした黒白黒白黒のパターンが図4の(c)のように1:1:3:1:1になっていることを確認する。なっていないならばStep1に戻りピクセルの探索を再開する。ただし、斜め方向にカウントした黒白黒白黒のピクセル数が1:1:3:1:1になっていることの必要十分条件は、カウントした斜め方向の合計のピクセル数から1つのモジュールの平均のピクセル数を計算し、その値とカウントした全てのモジュールのピクセル数の差がそれぞれ計算した値の75%以内であること、である。

Step9

以上より、仮定した座標を位置検出パターンを中心座標であると認識する。

以上の位置検出パターン認識手順において、位置検出パターンの横縦斜め方向に対する黒白黒白黒の比率のみを識別に用いるため図5の(a)から(b)のように位置検出パターンのデザインを変更した場合も位置検出パターンとして認識する。つまり、図6のような三つの位置検出パターンのデザインを変更したQRコードを、ZXingを用いて検出可能である。

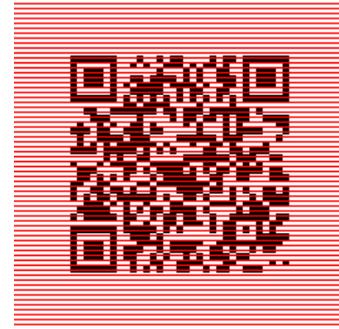


図3 水平方向のピクセル探索の可視化

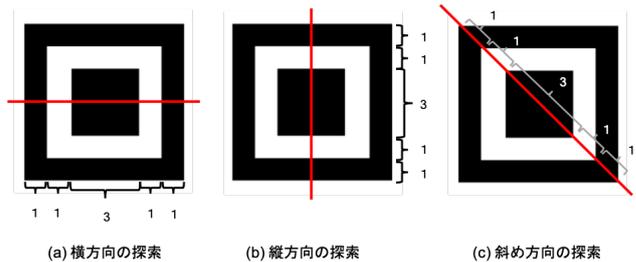


図4 位置検出パターン探索の可視化

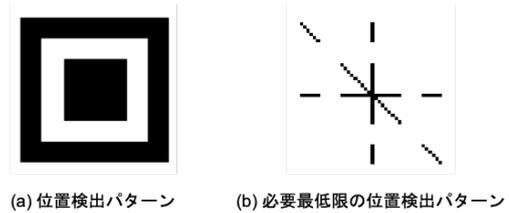


図5 位置検出パターンのデザイン



図6 位置検出パターンを変更したQRコード

図6に関して、理論上は位置検出パターンの横縦斜め方向のピクセル幅が1ピクセルだとしても位置検出パターンとして認識するが、カメラを用いる場合は手ぶれなどが想定されるため、読み取りやすさを考慮して図6の位置検出パターンの横縦斜め方向のピクセル幅は2ピクセルで構成している。

また、位置検出パターンを構成するピクセルの色において、4.1 節にある通りカラー画像を二値化したものに対して位置検出パターンの探索を行うことから、必ずしも黒もしくは白である必要はなく、二値化された時に黒もしくは白になるような色であれば良い。

4.3 モジュールの認識

4.1 節の QR コード復号アルゴリズムにおけるモジュールの明暗判定では、モジュール全体の白黒ではなく、モジュールの中央の 1 ピクセルのみをそのモジュールの明暗判定に用いているため、図 7 の(a)の位置検出パターンと位置合わせパターンを除くモジュールを図 7 の(b)のようにドット化したとしても正常に読み取ることが可能である。

また、モジュールを構成するピクセルの色において、4.1 節にある通り、カラー画像を二値化したものに対して QR コード復号アルゴリズムを進めていくため、位置検出パターンと同様に、必ずしも黒もしくは白である必要はなく、二値化された時に黒もしくは白になるような色であれば良い。

また位置合わせパターンにおいては、4.1 節の QR コード復号アルゴリズムにおける座標計算において、QR コードの歪みを補正するために行う射影変換の一つの角として位置合わせパターンの座標が用いられる。しかし、位置合わせパターンが探索できなかった場合、三つの位置検出パターンの座標から位置合わせパターンの座標を推測するため、必ずしも位置合わせパターンは必要ではなく、ドット化した場合も読み取ることが可能である。

さらに、年々スマートフォンのカメラは高性能になっており、ほとんどのスマートフォンのメインカメラの画素数が約 1200 万画素以上と非常に高精細である。そのため、QR コードを読み取る際においても、各モジュールに割り当てられる画素数が増加しており、モジュールのドット化を行う際も、非常に少ないピクセル数で構成したとしてもモジュールの中央に配置しておけば、精確にモジュールの明暗を判定することが可能である。

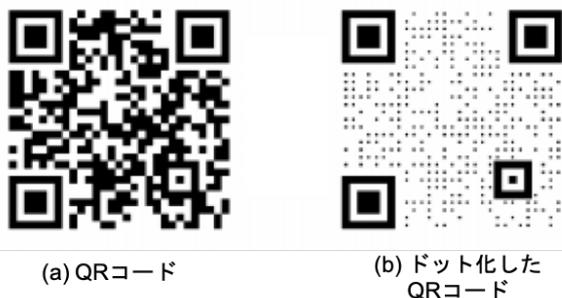


図 7 QR コードにおけるモジュールのドット化

5. 見えない QR コードの作成 —Hidden QR code—

4 節の位置検出パターンの認識条件を満たした上で、人間の目では QR コードに見えない Hidden QR code の一つである斑点模様の QR コードを提案する。

5.1 Hidden QR code の作成

まず、図 5 の(b)のような認識可能な位置検出パターンの中で、縦横斜めのピクセル幅をモジュールサイズにした位置検出パターンにデザインを変更する(図 8). 次に、位置検出パターンの認識に影響を与えない部分に対して、三つの位置検出パターンが同じデザインにならないようにランダムな黒モジュールを配置する(図 9). さらに、QR コードは黒と白で構成されるため、人の目で見えないようにするために、モジュールに色を付与する。カラー画像を二値化する際に、4.1 節の式(1)を利用して、黒と白になるような色のモジュールを複数作成する(図 10). そして、図 9 の(b)の



図 8 位置検出パターンをモジュールサイズで変更した QR コード

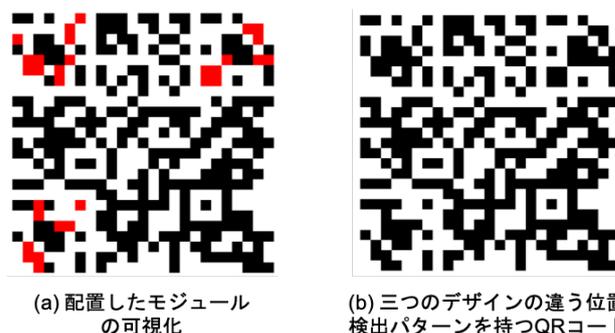


図 9 三つの位置検出パターンにランダムなモジュールを追加した QR コード

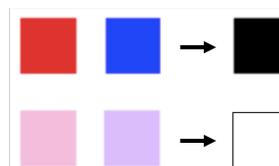


図 10 二値化した際に、黒と白になる色のモジュール

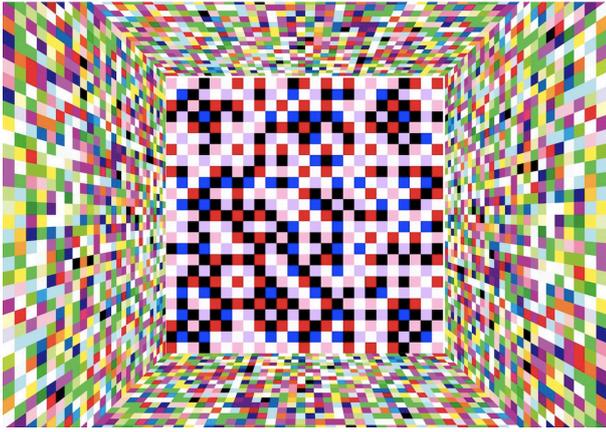


図 11 画像内に埋め込んだ Hidden QR code

QR コードの白と黒のモジュールに対して、図 10 の 6 色のモジュールを用いて斑点模様になるように、QR コードを再構成する。最後に、斑点模様の背景画像に、色を付与した QR コードを埋め込むことによって人の目では QR コードが含まれた画像と認識できないようにする(図 11)。

5.2 Hidden QR code の読み取りの評価実験

5.1 節で作成した図 11 の Hidden QR code を埋め込んだ画像に対して、iPhone 12(メインカメラ画素数:1200 万画素)、iPhone 6S(メインカメラ画素数:1200 万画素)、AQUOS sense3 plus メインカメラ画素数:1220 万画素)、Google pixel 4a(メインカメラ画素数:1220 万画素)を使用し、それぞれ標準にインストールされている QR コードデコーダを用いて読み取り可否を調査した。また、iPhone 12 にインストールした、LINE、ICONIT、クルクル、QR コードリーダー、Qrafter のそれぞれの QR コードデコーダアプリを用いて読み取り可否を調査した。

5.3 評価結果

図 11 において、上記の端末の標準にインストールされている QR コードデコーダを用いた場合、上記の全ての端末において読み取ることがわかった。また、iPhone 12 にインストールした、複数の QR コードデコーダアプリに関しては、ICONIT とクルクルを用いた時は読み取らず、LINE、QR コードリーダー、Qrafter を用いた時は読み取ることがわかった。

5.4 考察

結果から、iOS と Android の端末に標準にインストールされている QR コードデコーダや、LINE、QR コードリーダー、Qrafter では、図 11 を読み取ることから、ZXing を用いているかもしくは同様の位置検出パターン認識アルゴリズムであることが考えられ、ICONIT とクルクルでは、図 11 を読み取らないことから、4.2 節とは異なる位置検出パターン探索アルゴリズムであることが考えられる。

6. むすび

本稿では、QR コード読み取りライブラリである ZXing における QR コード復号アルゴリズム、特に位置検出パターンの認識条件を明記し、その条件を満たした上で、さらに人の目では認識できない Hidden QR code の構成法を与え、実際に作成した。iOS や Android の端末に標準にインストールされている QR コードデコーダや、いくつかの QR コードデコードアプリにおいて、提案した Hidden QR code を読み取ることを確認した。これは、人の目では QR コードには見えないが、カメラを向けると QR コードデコーダが自動で QR コードを認識する可能性があることを意味しており、意図しない QR コードを誤認識する危険性がある。さらに、この Hidden QR code を悪意のある第三者が作成すれば、悪意のあるサイトに強制的にアクセスさせられる可能性があるため、QR コードの大きな脆弱性と言える。QR コードが普及している現代社会においては、影響は大きなものと考えられるため対策されることが強く望まれる。

謝辞

本研究の一部は JSPS 科研費 20K11810 の助成を受けたものである。

参考文献

- [1] 日本工業規格, JIS, X0510, 二次元コードシンボル-QR コード-基本仕様, 2004.
- [2] Ian J, Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples", arXiv preprint arXiv:1412.6572, 2014.
- [3] A. Gnanasambandam, A. M. Sherman, and S. H.Chan, "Optical Adversarial Attack", arXiv preprint arXiv:2108.06247v2, 2021.
- [4] M. Cisse, Y. Adi, N. Neverova, and J. Keshet, "Houdini : Fooling Deep Structured Prediction Models", arXiv preprint arXiv:1707.05373, 2017.
- [5] 瀧田慎, 大熊浩也, 森井昌克, "二つの情報を出力する QR コードの構成—悪性サイトに誘導する QR コードの存在とその脅威—", 電子情報通信学会論文誌 D, Vol.J103-D, No.4, pp.291-300, 2020
- [6] M. Takita, H. Okuma, and M. Morii, "A Construction of Fake QR Codes Based on Error-Correcting Codes," 2018 Sixth International Symposium on Computing and Networking (CANDAR), Takayama, 2018, pp.188-193.
- [7] Ian J, Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples", arXiv preprint arXiv:1707.05373, p3,2017.
- [8] <https://zxing.github.io/zxing/apidocs/>
- [9] 今井秀樹, "符号理論", 電子情報通信学会, 1990.