

2017年度 修士論文

畳み込みニューラルネットワークによる  
Tor 上の匿名 Web 通信の識別

提出日：2018年1月30日

指導：後藤滋樹教授

早稲田大学 基幹理工学研究科 情報理工・情報通信専攻  
学籍番号：5116F004-4

阿部 航太

# 目次

|       |  |    |
|-------|--|----|
| 第1章   | 序論                                       | 5  |
| 1.1   | 研究の背景と目的                                 | 5  |
| 1.2   | 論文の構成                                    | 5  |
| 第2章   | 技術的な背景                                   | 7  |
| 2.1   | Torによる匿名通信                               | 7  |
| 2.2   | Website Fingerprinting Attack            | 8  |
| 2.2.1 | 概要                                       | 8  |
| 2.2.2 | 評価方法                                     | 8  |
| 2.2.3 | データセット                                   | 9  |
| 2.3   | 畳み込みニューラルネットワーク                          | 10 |
| 第3章   | 先行研究                                     | 13 |
| 3.1   | Optimal String Alignment Distanceを用いた識別法 | 13 |
| 3.2   | k-nearest neighbor algorithmを用いた識別法      | 14 |
| 3.3   | Autoencoderを用いた識別法                       | 14 |
| 第4章   | 提案手法                                     | 16 |
| 第5章   | 評価実験                                     | 18 |
| 5.1   | 概要                                       | 18 |
| 5.2   | Close World Test                         | 19 |
| 5.3   | Open World Test                          | 20 |
| 5.4   | Spatial Pyramid Poolingを用いた場合            | 24 |
| 5.5   | 先行研究との比較                                 | 26 |
| 第6章   | 結論                                       | 27 |

|      |    |
|------|----|
| 謝辭   | 27 |
| 參考文獻 | 29 |

## 目 一 覧

|      |   |    |
|------|---|----|
| 2.1  | Tor の通信方式 . . . . .                                     | 7  |
| 2.2  | データセットの例 . . . . .                                      | 10 |
| 2.3  | 各データに含まれる cell の個数の累積分布関数 . . . . .                     | 10 |
| 2.4  | Spatial Pyramid Pooling の構造 . . . . .                   | 12 |
| 3.1  | Autoencoder の構造 . . . . .                               | 15 |
| 4.1  | CNN への入力の例 . . . . .                                    | 17 |
| 4.2  | CNN の構造 . . . . .                                       | 17 |
| 5.1  | Close World Test における epoch 数と精度の関係 . . . . .           | 19 |
| 5.2  | Close World Test における入力データ長と精度の関係 . . . . .             | 20 |
| 5.3  | Open World Test における epoch 数と TPR の関係 . . . . .         | 21 |
| 5.4  | Open World Test における epoch 数と FPR の関係 . . . . .         | 21 |
| 5.5  | Open World Test における入力データ長と TPR の関係 . . . . .           | 22 |
| 5.6  | Open World Test における入力データ長と FPR の関係 . . . . .           | 22 |
| 5.7  | TPR・FPR の調節手法 . . . . .                                 | 23 |
| 5.8  | TPR・FPR の調節結果 . . . . .                                 | 23 |
| 5.9  | 本研究で行った Spatial Pyramid Pooling の変更 . . . . .           | 24 |
| 5.10 | Closed World Test における Pyramid Height と精度の関係 . . . . .  | 25 |
| 5.11 | Open World Test における Pyramid Height と TPR の関係 . . . . . | 25 |
| 5.12 | Open World Test における Pyramid Height と FPR の関係 . . . . . | 26 |

# 表一覽

|                                  |    |
|----------------------------------|----|
| 5.1 学習に使用したハイパーパラメータ . . . . .   | 18 |
| 5.2 提案手法と先行研究の手法の精度の比較 . . . . . | 26 |

# 第 1 章

## 序論

### 1.1 研究の背景と目的

接続経路を匿名化するプロトコルとして Tor が利用されている。Tor は違法な取引を目的とした Web サイトへのアクセスの際に利用されることがある。そのため、Tor を利用した通信であってもアクセス先を特定できる技術が必要とされる。

本研究は Website Fingerprinting Attack の新しい技術を確立して、Tor 利用者がどの Web サイトへアクセスしたかを識別することを目的とする。Website Fingerprinting Attack は、匿名通信の中でも暗号化されていない情報であるパケット長やパケット数、パケットの流れる方向の情報に基づいて、Tor の利用者がアクセスしている Web サイトを識別する。本研究では、Deep Learning の技術である畳み込みニューラルネットワークを用いて特徴量を自動的に抽出して、特徴量を手動で抽出することなく Website Fingerprinting Attack を行う手法を提案する。

### 1.2 論文の構成

本論文は以下の章により構成される。

#### 第 1 章 序論

本論文の概要について述べる。

#### 第 2 章 技術的な背景

Tor, 指紋攻撃, 畳み込みニューラルネットワークの理論を紹介する。

### 第 3 章 先行研究

Website Fingerprinting Attack についての先行研究について解説する .

### 第 4 章 提案手法

本研究の提案手法を説明する .

### 第 5 章 評価実験

提案手法の評価実験を行い , その結果を考察する .

### 第 6 章 結論

本論文の結論を述べるとともに , 残された課題を示す .

## 第 2 章

### 技術的な背景

#### 2.1 Tor による匿名通信

Tor [1,2] は、接続経路匿名化プロトコルの一つである。Tor の通信の例を図 2.1 に示す。接続経路匿名化は、ユーザーと Web サーバーの間に複数個（初期設定では 3 個）の Tor サーバーを経由させることで行われる。また、ユーザーは各 Tor サーバーと TLS (Transport Layer Security) を用いたセッションを確立する。そのため、Web サーバーに一番近いサーバー以外は、通信内容を知ることが出来ない。この構成において、各サーバーは直接接続しているコンピューターの IP アドレスのみを知ることが出来る。

さらに、Tor の通信では通信内容を cell と呼ばれる形式にカプセル化して通信を行う。cell は 512 バイトの固定長である。そのため、パケット長からの通信内容の推測が行いにくくなっている。

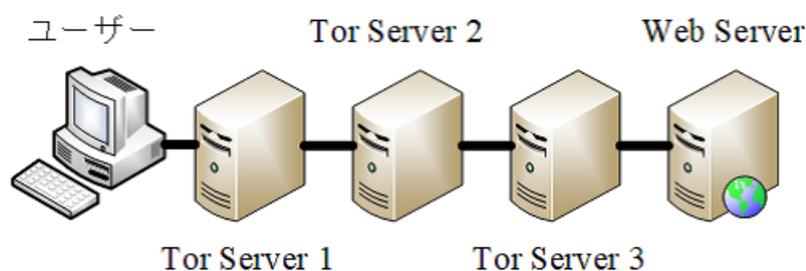


図 2.1: Tor の通信方式

## 2.2 Website Fingerprinting Attack

### 2.2.1 概要

Website Fingerprinting Attack は、VPN や Tor のようなプロキシを経由しているユーザーがアクセスした Web サイトを、ユーザーとプロキシの間を流れるパケットの特徴から特定する攻撃である。これらのようなプロキシを経由する通信では、データの暗号化が行われている。そのため、データに含まれる HTTP ヘッダの情報を見ることで、アクセスしている Web サイトを特定することはできない。しかし、暗号化されている場合でも流れているパケット数やパケット長、時間などの情報は得ることができる。Website Fingerprinting Attack では、これらのデータをキャプチャし、攻撃者が事前に収集したデータとの比較を行うことでユーザーがアクセスした Web サイトの特定を行う。

Tor において、ユーザーは入り口ノード（図 2.1 の例では TorServer 1）としか通信を行わない。そのため、Website Fingerprinting Attack を行う場合は、ユーザーの IP アドレスを知ることが出来るユーザーと Tor 入り口ノード間の通信をキャプチャする必要がある。

そのような通信をキャプチャを行う方法は二つ存在する。一つ目は、攻撃者が入口ノードを設置してそこを通るパケットをキャプチャする方法である。また、二つ目は ISP などのネットワーク上のパケットをキャプチャ出来る攻撃者がユーザーと入り口ノードの間を流れるパケットをキャプチャする方法である。ここで、ユーザーが接続する入口ノードはランダムに決定されるため攻撃者が設置したノードに接続する確率は低い。そのため、後者の方法を用いるのがより現実的な手法である。

### 2.2.2 評価方法

Website Fingerprinting Attack の精度の評価方法には、Closed World Test と Open World Test の 2 種類がある。Closed World Test は、ユーザーが攻撃者の想定した Web サイトの集合の中の一つにだけアクセスするという前提で行われるテストである。例えば、攻撃者が想定した Web サイトが 100 個あるとすると、ユーザーはそのうちの一つの Web サイトにアクセスを行う。攻撃者はユーザーがアクセスした Web サイトについてその 100 の Web サイトのうちどのサイトにアクセスしたかを識別する。

Open World Test は、ユーザーが攻撃者が検出を行いたい Web サイト以外を含め、あらゆる Web サイトにアクセスするという条件で行うテストである。そのため、ユーザーがアクセ

スした Web サイトが攻撃者が検出すべき Web サイトかどうか，また検出すべき Web サイトのうちどの Web サイトかまで判別を行う．よって，Open World Testの方がより現実の状況に近いテストである．

### 2.2.3 データセット

Website Fingerprinting Attack の評価に使用するデータセットとして，Wang が公開しているデータセットが存在する [7]．データセットには，検出するサイト (monitored website) 用のデータとして，100 サイト分が 1 サイトにつき 90 個ずつ，検出しないサイト (non-monitored website) 用のデータとして 4000 サイト分が各 1 個ずつ用意されている．monitored website は，中国，イギリス，サウジアラビアでブロックされているアダルトコンテンツ，BitTorrent のトラッカー，宗教的または政治的な内容を含むサイトである．一方，non-monitored website は Alexa [9] の top 10000 sites から monitored website を除いたものが用いられている．

データセットの一例を図 2.2 に示す．図で示されているのは，ある Web サイトに Tor Browser [3] を用いてアクセスしたときに通信が行われたパケットから cell を抽出し，その方向と時間を示したデータである．一列目は，cell の時間時間 (秒) を最初の cell の送受信時間を 0 として示している．二列目は，cell の方向が示されている．クライアントから Tor サーバーに送られた cell は 1，Tor サーバーからクライアントに送られた cell は -1 で示されている．この並びがページ読み込み開始から終了までに送受信された cell について示されている．図 2.2 の例では先頭の 10 個の cell を示しているが，実際のデータはさらに続いており，その数は Web サイトによって大きく変化する．また同じ Web サイトに複数回アクセスを行った場合でも多少の変化がある．図 2.3 はデータセットに含まれる全 18000 個のデータにそれぞれについて 1 個のデータに含まれている cell の個数の累積分布関数を示したものである．図 2.3 では見やすさのために横軸の最大値を 10000 としているが，このときの累積分布関数の値は 0.985 であり，一番多くの cell を含んだデータに存在する cell の個数は 59416 個である．

|                |    |
|----------------|----|
| 0.0            | 1  |
| 0.0            | 1  |
| 0.116133928299 | 1  |
| 0.499715805054 | -1 |
| 0.499715805054 | -1 |
| 0.782404899597 | -1 |
| 0.969846963882 | -1 |
| 0.969846963882 | -1 |
| 0.969846963882 | -1 |
| 0.969846963882 | -1 |

図 2.2: データセットの例

パケットに含まれる cell の数は、キャプチャしたトラフィックデータのフレーム長を 600 を単位にして丸めることで数える。また、フローの制御に用いられる SENDME cell が一定間隔で送受信されるが、これを取り除かれている。

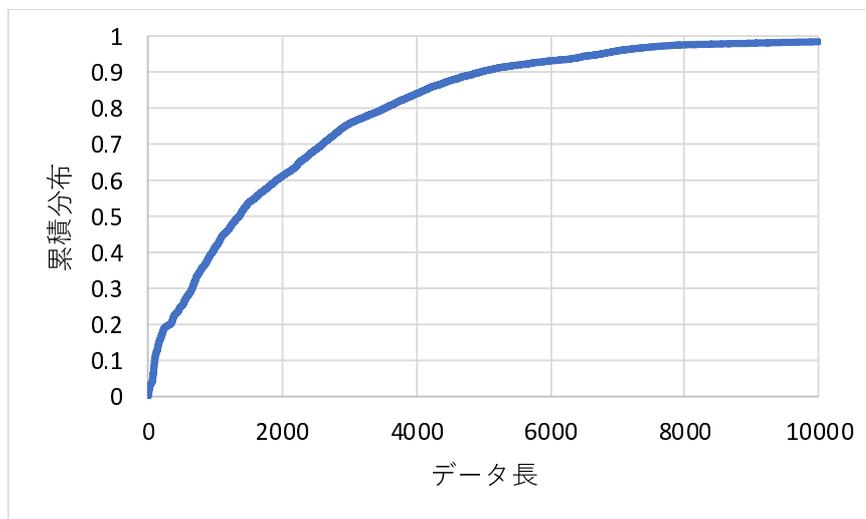


図 2.3: 各データに含まれる cell の個数の累積分布関数

## 2.3 畳み込みニューラルネットワーク

畳み込みニューラルネットワーク (CNN) は Convolution 層と Pooling 層を使用したニューラルネットワークであり、これらを用いて局所的な特徴を抽出出来ることから画像認識の分野で

良く用いられている．Convolution 層は入力に対してフィルタを用いた畳み込みを行い，特徴抽出を行う． $S \times S$  画素で  $N$  チャンネルの画像を  $L \times L$  のフィルタで畳み込みを行うとき，入力を  $x_{ijk}((i, j, k) \in [0, S-1] \times [0, S-1] \times [1, N])$ ，フィルタを  $w_{ijk}((i, j, k) \in [0, L-1] \times [0, L-1] \times [1, N])$  として出力  $u_{ij}$  は次式のように計算される．

$$u_{ij} = \sum_{k=1}^N \left[ \sum_{(p,q) \in P_{ij}} x_{pqk} w_{p-i, q-j, k} \right] + b_k \quad (2.1)$$

ただし， $P_{ij}$  は画像中の画素  $(i, j)$  を頂点とするサイズ  $L \times L$  の正方領域であり， $b_k$  はバイアス項である．

Pooling 層では入力のサブサンプリングを行う．この操作によって，特徴の存在する位置が変化しても特徴抽出を行うことが出来るようになる．Pooling 層にはいくつかの種類があるが，本研究では Max Pooling および Spatial Pyramid Pooling [13] を用いる．Max Pooling はノードの入力の最大値を出力する Pooling の方式であり次式で定義される．

$$y_{ijk} = \max_{(p,q) \in P_{ij}} x_{pqk} \quad (2.2)$$

続いて，Spatial Pyramid Pooling について説明を行う．Spatial Pyramid Pooling の構造を図 2.4 に示す．Spatial Pyramid Pooling では，Pyramid Height を  $N$  としたとき，1 から  $N$  の整数  $k$  について入力を  $2^{(k-1)} \times 2^{(k-1)}$  分割にした画像をそれぞれ用意する．続いて，分割した領域においてそれぞれ Max Pooling を行い，Max Pooling の出力を連結させて，Spatial Pyramid Pooling の出力とする．図 2.4 は Pyramid Height が 3 のときの例である．

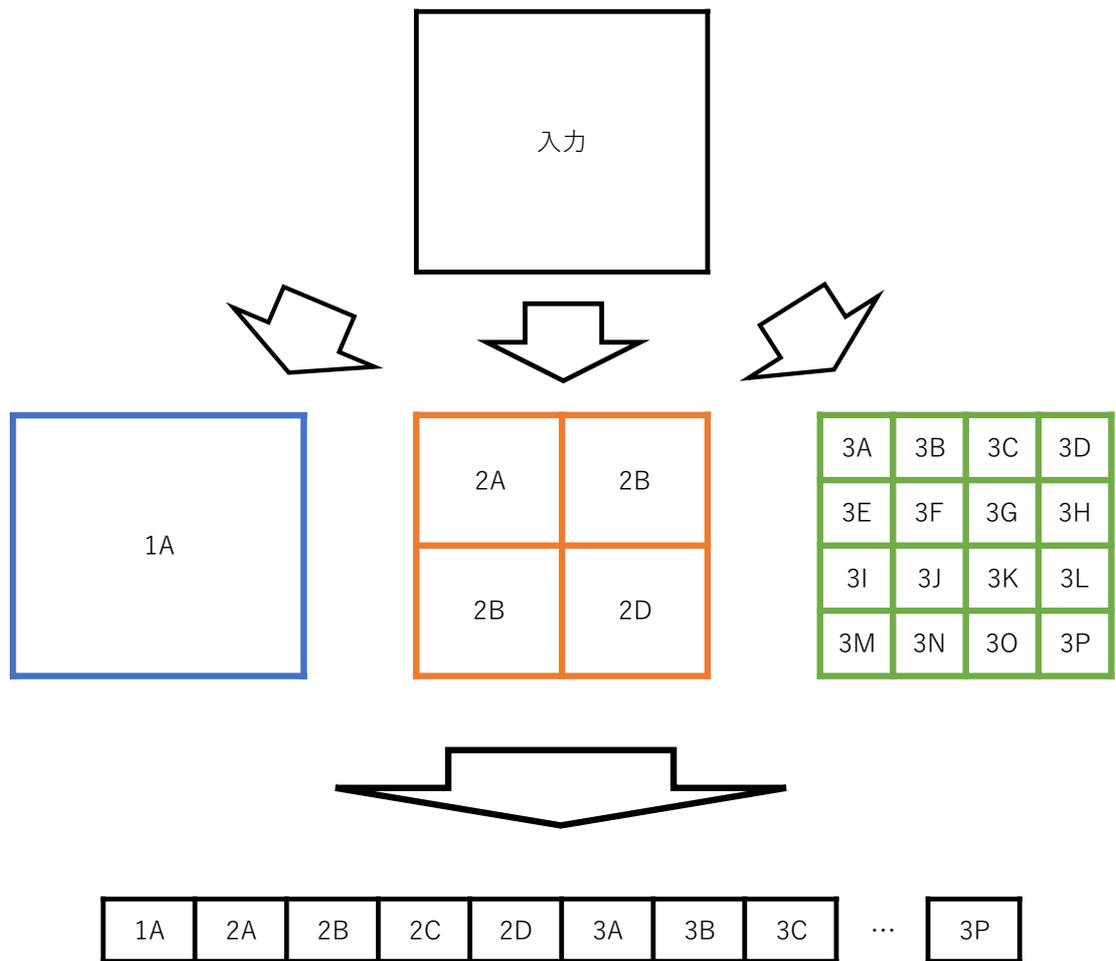


図 2.4: Spatial Pyramid Pooling の構造

## 第 3 章

### 先行研究

#### 3.1 Optimal String Alignment Distance を用いた識別法

Wang らは Optimal String Alignment Distance (OSAD) を用いた手法で Website Fingerprinting Attack を行った [4] . この手法は , Tor の cell の流れを文字列として考える . 2 つのキャプチャを比較したとき , 同じサイトへのアクセスであれば cell の流れは似た形になるため文字列の距離は小さくなり , 違うサイトであれば距離は大きくなる .

Wang らは , 文字列の距離を求めるアルゴリズムに OSAD を利用し , 次のように SVM (Support Vector Machine) を利用した . 二つの文字列  $s_1, s_2$  の距離を  $D(s_1, s_2)$  としたとき ,

$$D'(s_1, s_2) = \frac{D(s_1, s_2)}{\min(|s_1|, |s_2|)} \quad (3.1)$$

$$K(s_1, s_2) = e^{-D'(s_1, s_2)^2} \quad (3.2)$$

を求める . このとき ,  $D'(s_1, s_2) = 0$  , すなわち二つの文字列が同じ時  $K(s_1, s_2) = 1$  となる .  $D'(s_1, s_2)$  が大きくなる , すなわち文字列の距離が離れると  $K(s_1, s_2)$  は小さくなり ,  $D'(s_1, s_2)$

としたときの  $K(s_1, s_2)$  の極限は 0 となる . このことから ,  $K(s_1, s_2)$  をカーネル関数として用いることができ , これを用いて SVM を作成する . また , Wang らは 2 値分類を繰り返し多数決を取ることで多値分類を行う one-against-one 法を用いた .

## 3.2 k-nearest neighbor algorithm を用いた識別法

さらに, Wang らは k-nearest neighbor algorithm (k-NN 法) を用いた手法で Website Fingerprinting Attack を行った [5]. この手法では, まずキャプチャから特徴量を抽出する. 特徴量として, 次を使用する.

- 総転送量
- 総転送時間
- パケット長
- パケットのバーストの長さ
- 最初の 20 個のパケットの長さ

次に特徴量の重み付けを行う. これは, 複数種類の特徴量の中で重要な特徴量とあまり重要でない特徴量があるからである. 最後に, 決められた重みを用いて k-NN 法によってテストデータの分類を行う.

## 3.3 Autoencoder を用いた識別法

我々は, Deep Learning の手法の一つである Denoising Stacked Autoencoder を用いて Website Fingerprinting Attack の先行研究を行った [14]. Autoencoder は入力層, 隠れ層, 出力層からなるニューラルネットワークである. Autoencoder の構造を図 3.1 に示す. Autoencoder は, 次式で示すように入力  $\{x_i, \dots\}$  と出力が同じになるように重み  $W, b, W', b'$  の最適化を行う. 重みを最適化した Autoencoder の中間層の値を用いることで特徴の自動抽出を行うことができ, Website Fingerprinting Attack に用いることが出来る. この手法では, Autoencoder の入力にノイズを加えることで過学習を防ぎ, さらに Autoencoder の中間層を次の Autoencoder の入力とすることで多層化した Denoising Stacked Autoencoder を使用している.

$$\min_{W, b, W', b'} \sum_i \|x_i - f'(W' f(W x_i + b) + b')\|_2^2 \quad (3.3)$$

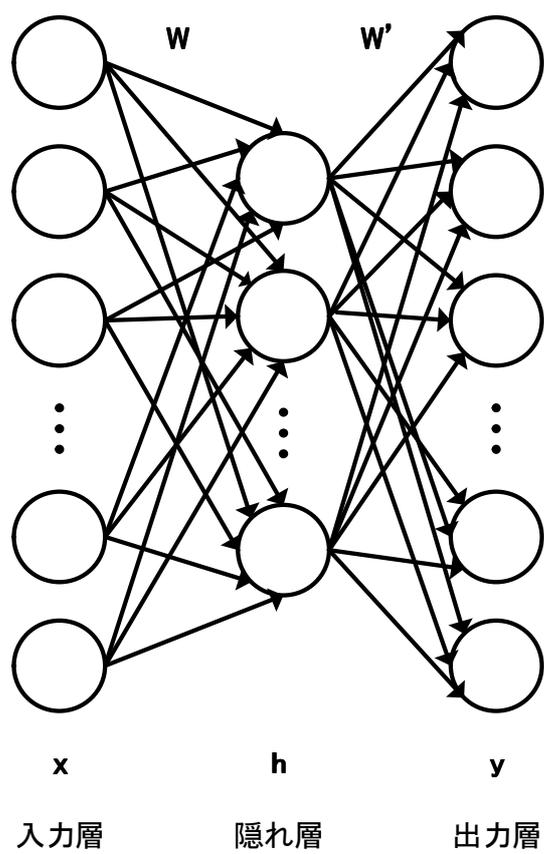


図 3.1: Autoencoder の構造

## 第 4 章

# 提案手法

本研究では, Convolution Neural Network (CNN) を利用した Website Fingerprinting Attack の技術を提案する. Website Fingerprinting Attack は次の 4 ステップで行う.

1. 教師データの収集
2. CNN への入力のためのデータの整形
3. CNN を用いた学習
4. 未知のデータの識別

教師データの収集では, 攻撃者が monitored website および non-moniteed website にアクセスしてそのトラフィックを収集し, cell の情報を抽出する必要がある. 本研究では 2.2.3 節で述べた Wang のデータセットを用いているためデータの収集は行っていない. 続いて, Tor の cell データを CNN への入力に適した形式に変換を行う. 本研究では, クライアントから Tor サーバーに送られた cell を  $(1, 0)^T$ , Tor サーバーからクライアントに送られた cell を  $(0, 1)^T$  として表し, これを時系列に沿って横方向に連結していく. 図 2.2 に示した cell の並びに対応する入力は図 4.1 のようになる. また, CNN へ入力を行う際はデータの長さを一定にする必要がある. しかし, cell の数はある web サイトにアクセスしたときに発生するトラフィック量に依存するため一定ではない. そのため, あるデータについて cell の数があらかじめ設定した値よりも小さい場合は  $(0, 0)^T$  でパディングを行い, 大きい場合は設定した値以降の cell を無視する. この操作によって CNN への入力の大きさを一定にする. 図 4.1 の例では  $(0, 0)^T$  でのパディングを行っている.

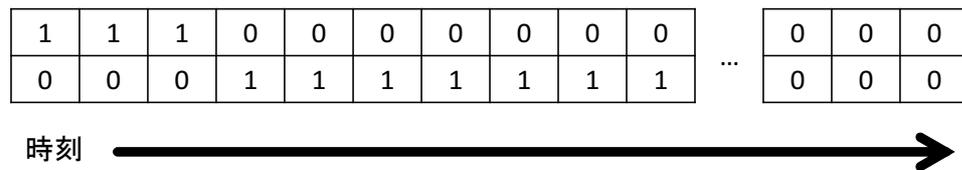


図 4.1: CNN への入力の例

続いて, CNN を用いた学習を行う. 本研究で使用するニューラルネットワークの構成図を図 4.2 に示す. 本研究では Convolution 層, Pooling 層, 全結合層をそれぞれ 2 層使用したニューラルネットワークを使用している. プーリング層, および fc2 への入力は活性化関数を経由している. 活性化関数にはランプ関数  $f(x) = \max(x, 0)$  を用いている. また, Pooling 層には最大値プーリングを用いてる. さらに, 全結合への入力の際には Dropout を用いている. 学習の際に用いる誤差関数には Softmax Cross Entropy を用いる. 学習の完了後, 学習したモデルを用いて未知のデータの識別が可能になる.

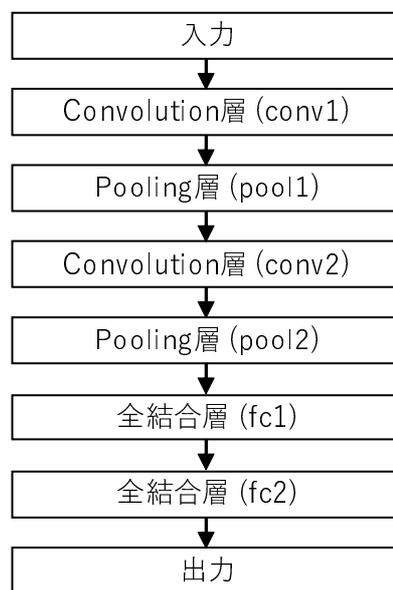


図 4.2: CNN の構造

# 第 5 章

## 評価実験

### 5.1 概要

2.2.3 節で説明を行った Wang のデータセットを用いて Closed World Test および Open World Test を行う。評価実験では、CNN の実装に Chainer 1.24.0 [8] を用いた。実験に用いたハイパーパラメータを表 5.1 に示す。これらのパラメータはグリッドサーチを用いて最適なパラメータを選択している。

表 5.1: 学習に使用したハイパーパラメータ

| パラメータ            | 値      |
|------------------|--------|
| 出力チャンネル数 (Conv1) | 8      |
| フィルタサイズ (Conv1)  | (64,2) |
| プーリングサイズ (Pool1) | 1      |
| 出力チャンネル数 (Conv2) | 8      |
| フィルタサイズ (Conv2)  | (4,1)  |
| プーリングサイズ (Pool2) | 2      |
| 全結合層 (fc1) の出力   | 200    |
| ドロップアウト率         | 0.5    |
| バッチサイズ           | 400    |

ここで、学習の際の重みの最適化には Adam (Adaptive Moment Estimation) を利用している。

## 5.2 Close World Test

Close World Test では 100 種類存在する monitored website のデータを用い、各サイトについて 90 個のデータについて 5 分割交差検定を行い分類精度の検証を行った。実験の結果を図 5.1, 5.2 に示す。

図 5.1 は入力データ長を 5000 とした時の epoch 数と精度の関係を示している。epoch23 で精度が 0.90 に到達し、epoch64 で 0.91 に到達している。図 5.2 は、入力データ長を変化させたときの精度の変化を示している。データの入力長は 1000 までは 100 ごとに測定しており、それ以降は 1000 ごとに測定を行っている。図 5.2 で示している精度は epoch 数が 181 の時から 200 までの時の平均を取ったものである。結果として、入力データ長が 900 の時に精度が 0.90 に到達し、2000 の時に 91.4 となり最大となっている。2.2.3 節の図 2.3 を参照すると、データセットの全データのうちデータ長が 2000 より小さいデータは 6 割程度である。そのため、入力データ長が 2000 の時は 4 割程度のデータにおいて入力にすべての cell 情報を含むことが出来ていない。しかし、入力長が 2000 の時に一番高い精度が出ている、これは、より先頭の cell の順序は同じ Web サイトに複数回アクセスしても大きく変化しないが、より後ろの cell については画像などが読み込まれる順序や通信速度によって並びが変化しやすくなっているためであると考えられる。そのため、入力長を一定値より大きく取っても、より後ろのデータがノイズとなり精度が改善せずに悪化する結果になると考えられる。

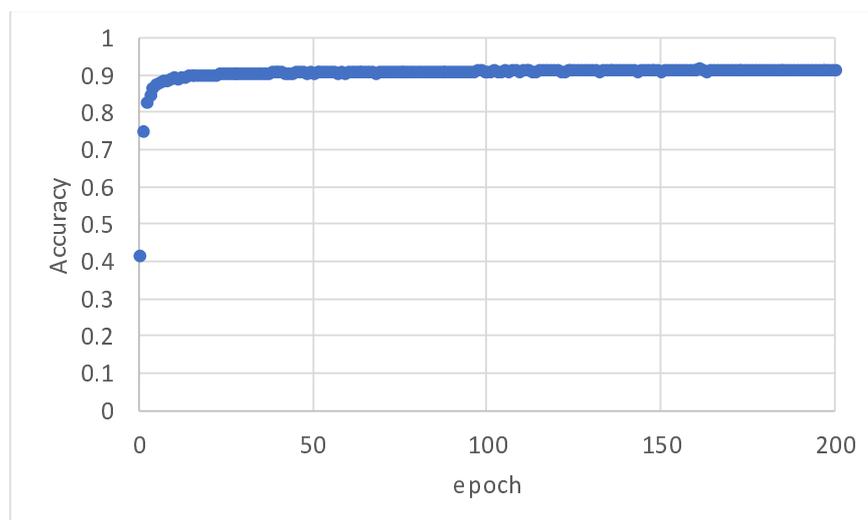


図 5.1: Close World Test における epoch 数と精度の関係

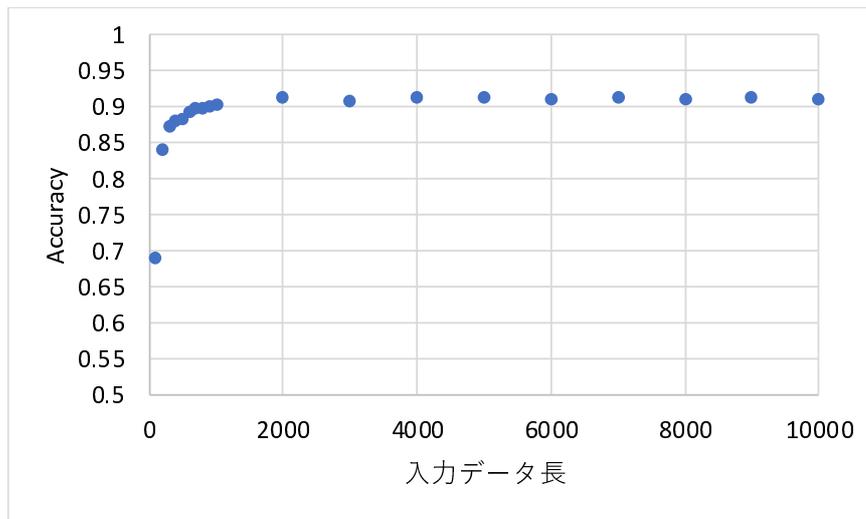


図 5.2: Close World Test における入力データ長と精度の関係

### 5.3 Open World Test

Open World Test では monitored website のデータと non-monitored website のデータを両方用いる。monitored website のデータは Close world Test と同様に各サイトについて 90 個のデータを 72 個と 18 個に分割することで教師データとテストデータとする。non-monitored website のデータについては 9000 種類のサイトのデータが各 1 個ずつあるため、これを 7200 と 1800 に分割し教師データとテストデータとする。このとき、テストデータに含まれる non-monitored website は教師データ内に存在しないため、ユーザーが攻撃者が未知の Web サイトにアクセスしたときにも正しく分類を行うことが出来るか否かを評価することが出来る。Close World Test と同様に 5 分割交差検定で評価を行う。

図 5.3 から 5.6 に Open World Test の結果を示す。図 5.3, 5.4 は入力データ長を 5000 とした時の epoch 数と True Positive Rate (TPR) および False Positive Rate (FPR) の関係を示している。TPR はテストデータに含まれる monitored website のデータを monitored website と分類し、さらに 100 種類ある monitored website のうちどの Web サイトであるかまで正しく分類できた確率である。一方、FPR はテストデータに含まれる non-monitored website のデータを誤って monitored website と分類してしまった確率である。TPR は大きい方が、FPR は小さい方が良い分類手法である。結果は、epoch 数が 73 の時に TPR が 0.89 に到達している。FPR については、0.05 ~ 0.07 で収束している。また epoch 数が小さいときに、FPR が極端に小さな

値を示している，これはほぼすべてのテストデータが non-monitoed website として判定されているためで，TPR・FPR 共に小さくなっている．図 5.5,5.6 は入力データ長を変化させたときの True Positive Rate および FPR の変化を示している．True Positive Rate は入力データ長が 2000 の時に 0.895 と最大になる．一方，FPR は入力データ長が 9000 のときに 0.005 と最小になる．

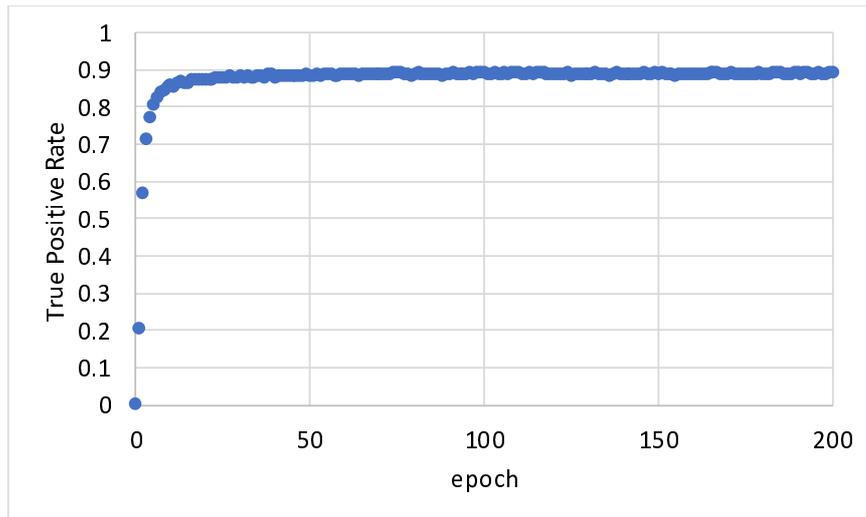


図 5.3: Open World Test における epoch 数と TPR の関係

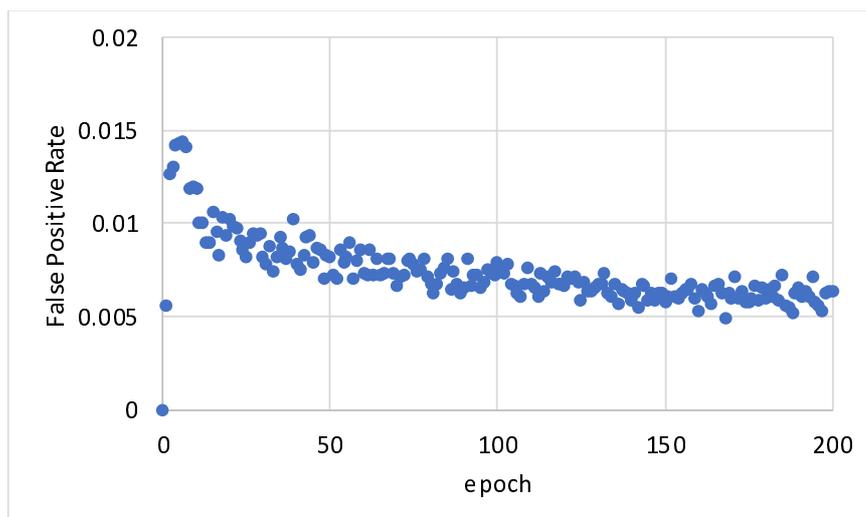


図 5.4: Open World Test における epoch 数と FPR の関係

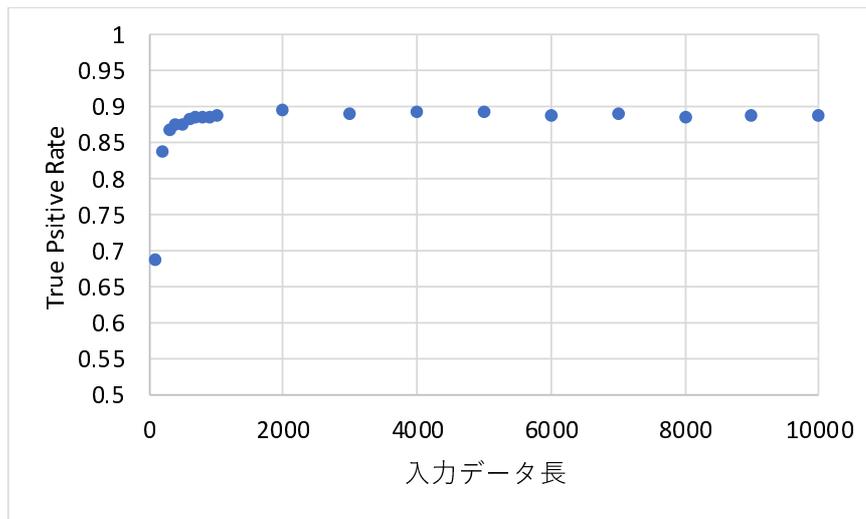


図 5.5: Open World Test における入力データ長と TPR の関係

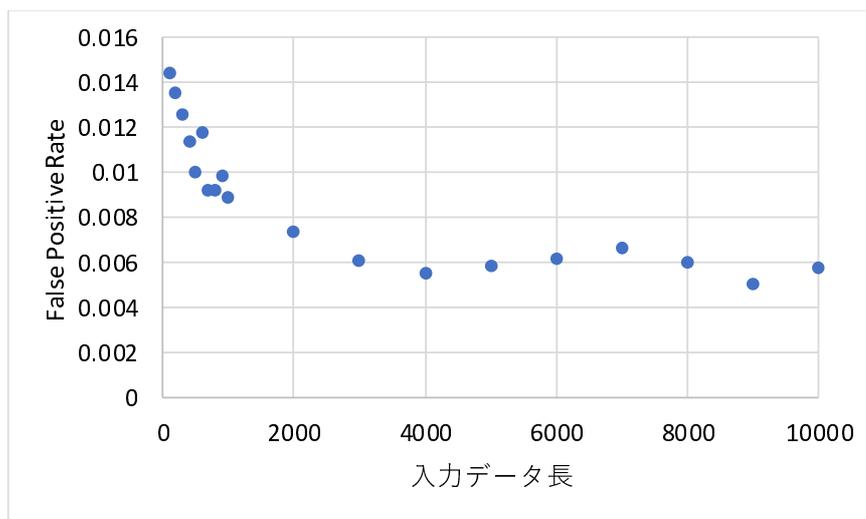


図 5.6: Open World Test における入力データ長と FPR の関係

本手法では、攻撃者の目的に応じて TPR と FPR の関係を調整することが出来る。この手法の概要を図 5.7 に示す。本手法ではテスト時に学習済みニューラルネットワークにテストデータを入力し、出力のうち一番値の大きなクラスを判定結果としている。そのため、non-monitored website のクラスの出力の値を一定値増減させてから判定を行うことで TPR と FPR の関係を調整することが出来る。図 5.7 では説明の便宜上 4 つの monited-website のクラスと 1 つの non-monitored website のクラスを用意した場合の例を示している。non-monitored website の

クラスの実出力値を一定数増加させてから判定を行った場合は, non-monitored website として判定されやすくなり結果として TPR は減少してしまうが FPR を減少させることができる. 逆に non-monitored website のクラスの実出力値を一定数減少させてから判定を行った場合は, non-monitored website として判定されづらくなり, FPR が増加してしまうが TPR を増加させることができる.

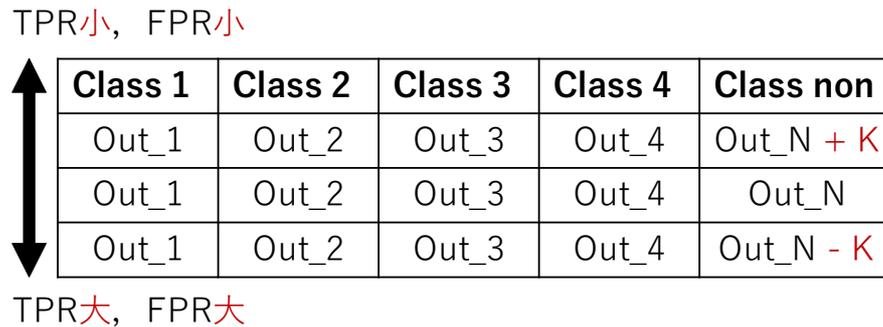


図 5.7: TPR・FPR の調節手法

この手法を用いた結果を図 5.8 に示す. 図 5.8 は, 入力データ長を 5000 とし epoch 数 200 の時のテスト結果を利用して作成している. 例として, 調節を行っていない時は TPR が 0.896 で 0.005 であるが, FPR 小さくするように設定すると TPR が 0.604 となってしまうが FPR が 0.0001 に減少, TPR を向上させた場合は FPR が 0.009 であるが TPR が 0.902 に向上している.

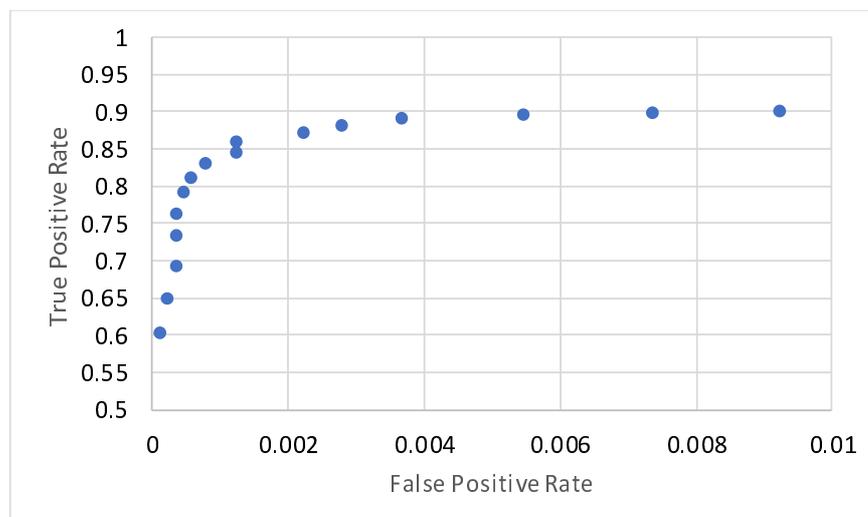


図 5.8: TPR・FPR の調節結果

## 5.4 Spatial Pyramid Pooling を用いた場合

図 4.2 における最後の Pooling 層 (pool2) に Max Pooling ではなく Spatial Pyramid Pooling を用いて評価実験を行った。使用したパラメータは表 5.1 と同様であり、また、入力データ長は 5000 に固定した。Spatial Pyramid Pooling を用いた pool2 における Pyramid Height のみを变化させた。

本研究では入力の次元が  $2 \times 5000$  と一方向 (時間方向) にのみ大きいため、画像認識によく用いられるような 2 次元の Spatial Pyramid Pooling は適応することが出来ない。そのため、本研究では図 5.9 のように時間方向にのみ分割を行う。

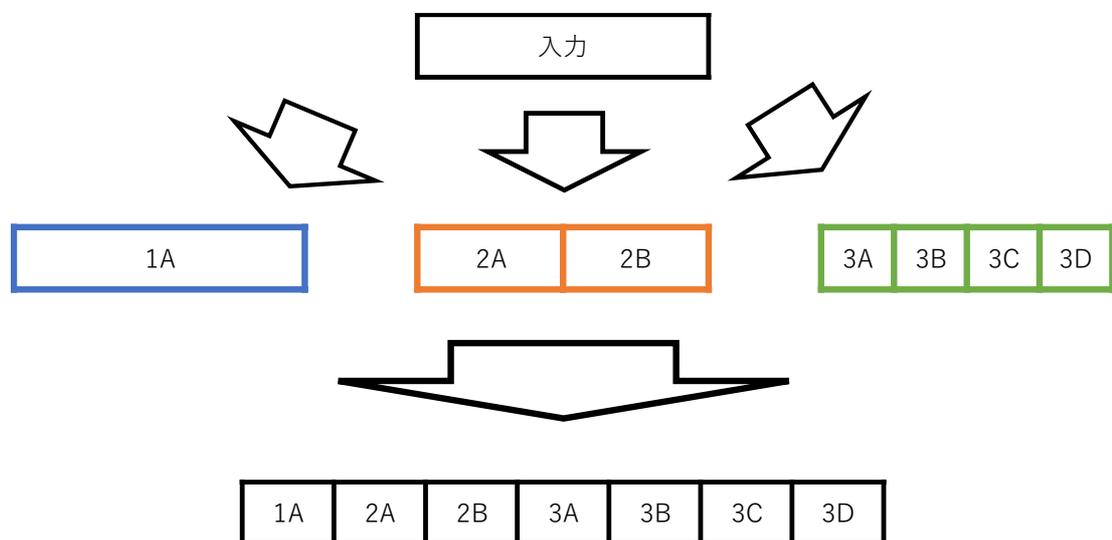


図 5.9: 本研究で行った Spatial Pyramid Pooling の変更

Closed World Test における精度を図 5.10 に示す。また、Open World Test における TPR を図 5.11 に、FPR を図 5.12 に示す。図に示した結果は、epoch 数が 181 の時から 200 までの時の平均を取ったものである。Pyramid Height を変化させる事による精度の大きな変化は見られなかったが、Closed World Test、Open World Test とともに精度が Max Pooling を用いた場合よりも悪くなった。そのため、本研究においては Pooling 層に Max Pooling を用いた方がよいと言える。

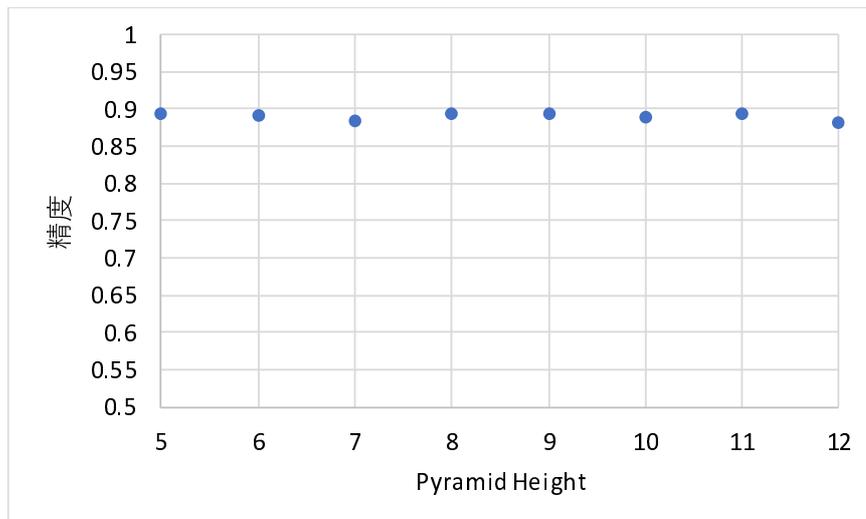


図 5.10: Closed World Test における Pyramid Height と精度の関係

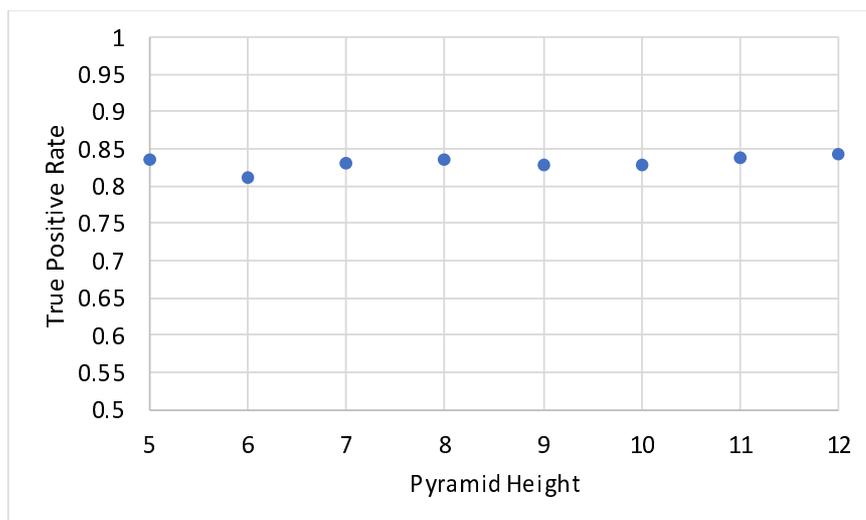


図 5.11: Open World Test における Pyramid Height と TPR の関係

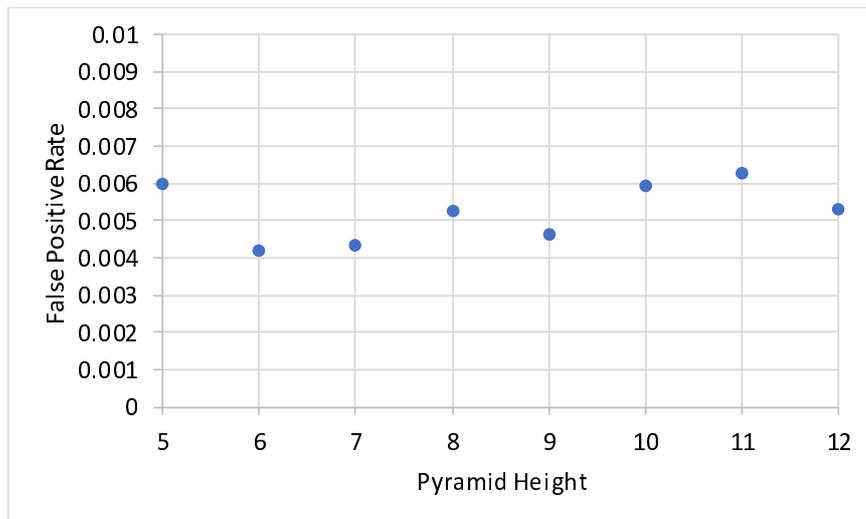


図 5.12: Open World Test における Pyramid Height と FPR の関係

## 5.5 先行研究との比較

3章の手法の結果と CNN を用いた提案手法について精度の比較を行う．表 5.2 に提案手法および先行研究の手法の Close World Test における精度，Open World Test における TPR と FPR を示す．先行研究の精度は論文中に記載されているものを示している．

CNN を用いた提案手法では Close World Test において先行研究で一番精度の高い k-NN 法を用いた手法と同等の精度を実現している．また，Open World Test においては，k-NN 法を用いた手法と同じ FPR においてより高い TPR を示している．

表 5.2: 提案手法と先行研究の手法の精度の比較

| 手法               | 精度   | TPR  | FPR   |
|------------------|------|------|-------|
| CNN (提案手法)       | 0.91 | 0.89 | 0.006 |
| Autoencoder [14] | 0.88 | 0.86 | 0.02  |
| OSAD [4]         | 0.90 | 0.83 | 0.06  |
| k-NN [5]         | 0.91 | 0.85 | 0.006 |

## 第 6 章

### 結論

本研究では，畳み込みニューラルネットワークを利用した匿名通信 Tor への Website Fingerprinting Attack の手法の提案を行なった．評価実験の結果，Close World Test において 0.91 の精度が得られ，Open World Test では 0.89 の TPR および 0.006 の FPR が得られた．これらの結果は既存手法よりも高精度なものであり，本手法が Tor における Website Fingerprinting Attack において有効なことが示された．

今後の課題として，Tor の Hidden Service を用いた Web サイトにおける Website Fingerprinting Attack の精度の検証が挙げられる．Hidden Service ではクライアント側が Tor を用いる時と同様の手順を Web サーバー側が行うことにより，Web サーバーが匿名性を得ることが出来る．身元を隠したまま Web サーバーを構築することが出来るため，Hidden Service を用いた違法な Web サイトが多数存在している．Hidden Service を用いても HTTP の通信を行うことによるファイルサイズ等の特徴は現れるため Website Fingerprinting Attack を適用可能であると考えられるが，より詳しい検証が必要である．

また，Spatial Pyramid Pooling の特徴として可変長の入力であっても出力を固定長にすることが出来る．そのため，可変長の入力を用いた識別によって Website Fingerprinting Attack の精度を向上させることが出来る可能性がある．このような技法を検証する必要がある．

# 謝辞

本修士論文の作成に当たり，日頃よりご指導いただいた後藤滋樹教授に深く感謝致します．  
また、本研究についてアドバイスをくださった内田真人教授、堀江輝樹氏に深く感謝致します．  
最後に，本研究を進めるにあたり，多大な御協力をいただきました後藤滋樹研究室の皆様に  
感謝致します．

## 参考文献

- [1] The Tor Project, "Tor Project — Privacy Online," <https://www.torproject.org/>.
- [2] Roger Dingledine, Nick Mathewson and Paul Syverson, "Tor: the second-generation onion router," In Proceedings of the 13th USENIX Security Symposium, pp. 303 – 320, Aug 2004.
- [3] The Tor Project, "Tor Browser - Tor Project," <https://www.torproject.org/projects/torbrowser.html>.
- [4] Tao Wang and Ian Goldber, "Improved Website Fingerprinting on Tor," WPES '13 Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society, pp. 201 – 212, Nov 2013.
- [5] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson and Ian Goldberg, "Effective attacks and provable defenses for website fingerprinting," 23th USENIX Security Symposium, pp. 143 – 157, Aug 2014.
- [6] Xiang Cai, Rishab Nithyanand, and Rob Johnson, "CS-BuFLO: A Congestion Sensitive Website Fingerprinting Defense," WPES '14 Proceedings of the 13th Workshop on Privacy in the Electronic Society, pp. 121 – 130, Nov 2014.
- [7] Tao Wang, "Website Fingerprinting," <https://www.cse.ust.hk/~taow/wf/>.
- [8] Preferred Networks, "Chainer: A flexible framework for neural networks," <https://chainer.org/>.
- [9] Alexa, "Alexa: Keyword Research, Competitive Analysis, & Website Ranking," <https://www.alexa.com/>.
- [10] Andriy Panchenko, Lukas Niessen, Andreas Zinnen and Thomas Engel, "Website Fingerprinting in Onion Routing Based Anonymization Networks," WPES '11 Proceedings of

- the 10th annual ACM workshop on Privacy in the electronic society, pp. 103 – 114, Oct 2011.
- [11] Andriy Panchenko, Fabian Lanze, Andreas Zinnen, Martin Henze, “Website Fingerprinting at Internet Scale,” Proceedings of the 23rd Internet Society (ISOC) Network and Distributed System Security Symposium (NDSS 2016), San Diego, USA, February 2016.
- [12] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson, “Touching from a distance: website fingerprinting attacks and defenses,” CCS ’12 Proceedings of the 2012 ACM conference on Computer and communications security, pp. 605 – 616, Oct 2012.
- [13] Kaiming He, Xiangyu Zhang, and Shaoqing Ren, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence Volume 37 Issue: 9, pp. 1904 – 1916, Jan 2015.
- [14] Kota Abe and Shigeki Goto, “Fingerprinting Attack on Tor Anonymity using Deep Learning,” Proceedings of the APAN42 Research Workshop, pp. 15 – 20, Aug 2016.
- [15] 阿部航太, 後藤滋樹 “畳み込みニューラルネットワークによる匿名通信 Tor の Website Fingerprinting,” SCIS2018 暗号と情報セキュリティシンポジウム, Jan 2018.
- [16] 麻生 英樹, 安田 宗樹, 前田 新一, 岡野原 大輔, 岡谷 貴之, 久保 陽太郎, ボレガラ ダヌシカ “深層学習,” 近代科学社, 東京, 2015.
- [17] 岡谷貴之, 齋藤真樹 “ディープラーニング,” 研究報告コンピュータビジョンとイメージメディア (CVIM), pp. 1 – 17, Jan 2013.