

深層学習を用いたピクトグラム画像における 多クラス分類問題とその応用

神田剛志^{†1} 伊藤一成^{†1}

ピクトグラムとは日本語で絵記号、図記号と呼ばれるグラフィックシンボルであり、意味するものの形状を使ってその意味概念を理解させる記号である。しかし国、文化、風習の違いから、しばしばピクトグラム単体での意味概念の理解は困難であり、これに対してはピクトグラムを文字等で情報補完する手法が有効と考えられる。そこで、本稿ではその基礎研究として、深層学習を用いたピクトグラムに特化した分類器の作成と評価、またそれを用いたアプリケーションの提案を行った。評価の結果、深層学習によるピクトグラム分類器は、学習データに各分類1枚のカラー画像を元に画像処理で拡張を行ったデータセットを用いても、十分な正解率を得ることができた。また、実際の使用を想定したアプリケーションの実装についても、新たな知見を得ることができた。

Multiclass Classification in Pictogram Images Using Deep Learning and Its Application

TSUYOSHI KANDA^{†1} KAZUNARI ITO^{†1}

1. 背景

ピクトグラムとは日本語で絵記号、図記号と呼ばれるグラフィックシンボルであり、意味するものの形状を使ってその意味概念を理解させる記号である[1]。

共通化の重要性から ISO (国際標準化機構) を中心に規格の審議や策定が進められており、例えば、案内用図記号は ISO7001, 安全用の図記号は ISO7010, 装置用図記号は ISO7000 など、ピクトグラムは案内、安全、施設、機器等々、様々な用途で標準化されている。また ISO3864 では、禁止、注意、指示、安全の4項目に関するピクトグラムデザインに関し、文字や矢印等の図形を併記する際のガイドラインが策定されている。通常、世の中に広く普及しているピクトグラムは作成ガイドラインに則りデザインされており、世界共通の記号表現として、世界中で用いられている。

特に日本では、近年の外国人観光客の急激な増加や、2020年の東京オリンピック開催などの理由もあり非常に注目されてきている。ピクトグラムに関する研究も研究分野を問わず広く行われており、一例を挙げるだけでも、感性工学[2]、異文化コミュニケーション[3]、メディア情報処理[4]、記号論[5]、感性[6]、プログラミング学習[7]など様々な学問領域でピクトグラムを題材とする研究が盛んになっている。

また、近年画像認識の分野で、深層学習[8]が大きな成果を上げている。深層学習 (以下 DL) は深層ニューラルネットワーク (以下 DNN) を用いた機械学習の手法である。DL を用いることで、ピクトグラムの解析においても、他の一

般物体認識と同様に高い正解率が期待できるだろう。

また、DL では機械学習を行うため学習データに変化をもたせることで人間の手を介することなく、様々な状況での認識に対応できる。このことから、JIS の標準案内図記号 (以下 JIS Z8210[9]) により規定されている全 181 種類のピクトグラムへの対応でも、異なったデータセットを用意することで他の手法を用意せずとも、対応することができると考えられる。そこで、本稿では、DL を用いたピクトグラム画像における多クラス分類手法と、それを用いたアプリケーションを提案する。

以下、本稿の構成は、2章で概要、3章でピクトグラムの分類問題を取り扱い、4章でその評価について述べる。5章でピクトグラム抽出処理の実装と評価について言及し、6章で実利用を想定したノイズテストとその評価、7章でアプリケーションへの応用と今後の展望について述べる。

2. 提案手法の概要

本稿で提案する手法は、JIS Z8210 で規定されている全 181 種類のピクトグラムのうち、「ヘルプマーク」を除く 180 種類のピクトグラム画像に対して DNN を用いて高い精度で分類を行うというものである。ピクトグラム画像は、ピクトグラムが矩形で抽出されているものを想定した。人が撮影した画像を入力として想定しているため、多少の角度、位置ズレが想定されるものとする。

ピクトグラム画像の解析には DNN を用いる。DNN の学習には教師あり機械学習を用いる。

^{†1} 青山学院大学 社会情報学部
School of Social Informatics, Aoyama Gakuin University

また、DNN の解析は、あらかじめ指定されたクラス分類を行う分類問題として扱う。これは、入力を DNN に与えると、出力として分類結果を与えるものである。本稿では、DNN の入力にピクトグラム画像を与え、出力に分類結果を得るとした。また、学習データのピクトグラム画像に対して事前に正解ラベルデータとの対応付けを行う。

3. 分類器の実装

3.1 実装概要

本稿の実装では、ピクトグラム画像を JIS Z8210 で規定されている 180 種類に分類する。180 種類のサンプル画像の一部と内容を図 1 に示す。

DNN による解析は、畳み込みニューラルネットワーク (Convolutional Neural Network, 以下 CNN) を用いた。CNN の実装には、深層学習フレームワークの Keras (バージョン 2.2.4) を利用し、明記されていないハイパーパラメータはライブラリの初期値を用いた。また、画像処理には、コンピュータビジョンライブラリである OpenCV (バージョン 3.4.2) を用いた。



図 1 分類画像の一部

(上段左から、案内所、情報コーナー、お手洗、男子、女子。下段左から、障がい者用、喫煙、エレベータ、エスカレータ、階段。)

3.2 学習・検証データの作成方法

図記号における分類問題は、ドイツの道路交通標識分類問題である GTSRB[10]や有名だが、ピクトグラムにはこのような機械学習を目的としたデータセットが存在せず、学習に用いるデータを自ら作成する必要がある。ウェブスクレイピングで画像を収集するという手法もあるが、JIS Z8210 で規定されているピクトグラムは、その使用頻度には大きな偏りがあり、よく使用されているピクトグラムほど画像枚数が多くなる不均衡データになってしまう恐れがある。また、ピクトグラムは、バリアフリーの観点からも統一化の必要性があるため、本稿ではその標準となる JIS Z8210 で規定されているピクトグラムを用いることとした。

学習・検証データの元画像には、交通エコロジー・モビ

リティ財団[11]が公開している公共・一般施設のピクトグラム画像を利用した (図 2 参照)。また、JIS Z8210 で規定されているが交通エコロジー・モビリティ財団は公開していないピクトグラム画像に関しては、株式会社石井マーク a が公開している図記号データベースから取得した。これらの画像は全て RGB の 3 チャンネルのカラー画像となっている。

これら 180 種類の画像を元に、様々な画像処理を施し学習する画像数を増やす。本稿では、CNN の入力はピクトグラムが矩形で抽出されているものであるという前提に基づき、次の画像処理を施した。画像処理は、回転、透視変換、平行移動、左右反転で、この順で画像処理を行った。回転は画像の中心点を中心に -10 度から 10 度まで 5 度毎に変化させる計 5 通りである。これは、実際のピクトグラム画像の撮影を想定すると大幅な角度変化は想定されないためである。透視変換は、元画像の 4 隅のうち 1 隅だけを縦方向と横方向同時に中心方向へそれぞれ $0, 5, 10$ ピクセルのいずれか移動する。つまり 4 (隅) $\times 3$ (段階)の 12 通りである。平行移動は上下と左右にそれぞれ独立に $-5, 0, 5$ ピクセル移動させた計 9 通りである。これは、ピクトグラムが矩形で抽出されている前提において、ピクトグラムの大幅な移動は想定されないからである。左右反転は、反転したものと、反転していないものの計 2 通りである。これは、階段やエスカレーターのようにピクトグラムには左右反転しているものがしばしば見られることを想定しているため、吉田、伊藤の研究[12]においても、左右反転処理を加えた学習データを用いることで、分類器の精度が上がることを示されている。図 2 に回転、透視変換、平行移動、左右反転のそれぞれの例を示す。左側から -10 度の回転、左下 10 ピクセル移動による透視変換、上方向に 5 ピクセル移動の平行移動、左右反転の 4 つとなっている。



図 2 画像処理の例

また、画像処理を行うことで生じる空白領域に背景画像を合成する。背景画像は一様分布に基づき自動生成された 500 枚のノイズ画像から、学習データ 1 枚を作成する度に無作為に選択し直している。以上の変換、合成を行うと 1 種類の画像が 5 (回転) $\times 12$ (透視変換) $\times 9$ (平行移動) $\times 2$ (左右反転) $=1,080$ 種類の画像となる。このうちの 2 割のデータを無作為に抽出し検証データとして扱い、残りの

a <http://www.ishiimark.com>

8割である 864 枚を学習データとして使用する。これを、用意した 180 種類の画像に適用することで 155,520 枚の学習データ、38,880 枚の検証データとなる。図 3 に画像データ作成例を示す。



図 3 画像データ作成例

3.3 CNN の構造

CNN の入力 は RGB の 3 次元、画像サイズを縦横共に 70 ピクセルの $3 \times 70 \times 70$ のものとした。画像サイズが 70×70 でない場合はバイリニア補間を用いて画像のリサイズを行う。本稿での実装においては、事前に別のタスクに向けて学習した CNN のパラメータを初期値としたファインチューニングを行う。具体的には、ImageNet で学習済みの VGG16[13]を用いており、図 4 に示したモデルの conv4-3 までの重みデータを固定し、それ以降の畳み込み層と全結合層においては重みデータを再学習させる。

VGG16		Fine tuning	
input(70 × 70 RGB image)		input(70 × 70 RGB image)	
conv1-1	重み 固定	conv1-1	
conv1-2		conv1-2	
maxpool		maxpool	
conv2-1		conv2-1	
conv2-2		conv2-2	
maxpool		maxpool	
conv3-1		conv3-1	
conv3-2		conv3-2	
conv3-3		conv3-3	
maxpool		maxpool	
conv4-1		再学習	conv4-1
conv4-2			conv4-2
conv4-3	conv4-3		
maxpool	maxpool		
conv5-1	conv5-1		
conv5-2	conv5-2		
conv5-3	conv5-3		
maxpool	maxpool		
FC-4096	FC-2048		
FC-4096	FC-256		
FC-1000	FC-180		
softmax	softmax		

図 4 ファインチューニングの詳細

このモデルにおいて、畳み込み層はすべての層でフィルタサイズ 3×3 、パディングを 1 とし、 2×2 の max pooling を行っている。また、Dropout[14]を確率 0.5 で最後の全結合層に追加している。損失関数は交差エントロピーを用い、活性化関数は ReLU を、最適化アルゴリズムは学習率を 0.001

に設定した SGD を用いた。学習はミニバッチ学習で行い、ミニバッチのサイズは 128 とした。

3.4 出力

画像を入力として受け取った CNN は分類結果を出力する。CNN の最終層はクラス数と同じ 180 次元になっており、最後の全結合層において Softmax 関数を活性化関数に用いることで、180 種類それぞれの分類確率を出力する。その内の最大値を示したラベルを分類結果として判断した。

4. 分類処理の評価

3.2 節の検証データ 38,880 枚（各クラス 216 枚ずつ）を用いて CNN の評価を行ったが、結果は 180 クラス中 178 クラスで 100% の正答率を得た。

しかし、「右側にお立ちください」と「左側にお立ちください」の 2 クラスのみ、正答率がそれぞれ 77.31%、50.00% という結果になった。これは、この 2 種のピクトグラム(図 5 参照) は互いに左右反転の関係にあり、3.2 節での左右反転処理を行った際に、本来「右側にお立ちください」であるはずのピクトグラムが「左側にお立ちください」のものと全く同じになってしまうためである。



図 5 「右側にお立ちください」と「左側にお立ちください」のピクトグラム

このような左右非対称性が判別するための根拠となるピクトグラムにおいては、学習データ作成の際に反転処理を行わないように細心の注意を払う必要がある。

5. ピクトグラム自動抽出処理

5.1 抽出処理の概要

3 章で実装した分類器は、撮影画像上に含まれるピクトグラム領域が手動で抽出することを前提としている。そこで、提案手法の実用性をより高めるために、ピクトグラム自動抽出処理を 3 章の分類器の前段に組み込んだ実装を追加した。本章では、「領域抽出方式」と「Single Shot Multibox Detector[15](以下 SSD)方式」の 2 つの抽出手法について実装を行い、評価を行った。領域抽出方式は画像処理のみの抽出手法であり、SSD 方式は深層学習を用いる物体検出アルゴリズムの一種である Single Shot Multibox Detector を用いている。

5.1.1 領域抽出方式

領域抽出方式は大きく次の Step 1 ~ Step 2 を順に行う。

Step 1:画像処理でピクトグラム候補領域の抽出

Step 2:結果画像の整理

画像データがピクトグラム抽出処理の入力に与えられると、まず Step 1 で画像処理が行われ、候補領域の抽出が行われる。それらの候補領域を Step 2 の入力とし、最終出力を行う。画像処理には OpenCV を用いている。

ピクトグラム画像は多くの場合、白背景に囲まれるなど自然背景には見られない構造をしている。そのため、エッジ検出を行うことで、ピクトグラム候補領域を抽出できると考えた。そこで、Step 1 の候補領域の抽出は、以下の Step 1.1~Step 1.3 を順に行う。

Step 1.1:入力画像をグレースケールに変換

Step 1.2:エッジ検出を行う

Step 1.3:エッジ情報に基づき候補領域の抽出

Step 1.2 のエッジ検出にはキャニー法を利用している。Step 1.3 では Step 1.2 で検出されたエッジ情報に基づき候補領域の抽出を行っている。候補領域は次に示す式(1)、式(2)の両式を満たすもののみ抽出される。

$$wh > \alpha WH \cdot \cdot \cdot (1)$$

$$\frac{1}{\beta} < \frac{w}{h} < \beta \cdot \cdot \cdot (2)$$

ここで、 w は各候補画像の幅のピクセル数、 h は各候補画像の高さのピクセル数、 W は入力画像の幅のピクセル数、 H は入力画像の高さのピクセル数である。また α 、 β はパラメータで本稿では α に 0.0083、 β に 1.6 を設定している。式(1)は入力画像に対して小さすぎる候補を除外するために、式(2)は縦横比が一定範囲外のものを除外するために設定されている。

また、Step 2 の画像の整理では、ある候補領域を完全に覆い尽くす他の候補領域があった場合は、覆い尽くされる候補領域を候補から外す処理を行った。

領域抽出方式の結果の例を図 6 に示す。



図 6 領域抽出方式の結果例

5.1.2 SSD 方式

次に、SSD を用いたピクトグラム候補領域の抽出手法を提案する。SSD への入力は $3 \times 300 \times 300$ となっており、出力結果にはピクトグラム候補領域の座標とその信頼度が出

力される、信頼度は 0 から 1 の値をとり、1 に近いほど検出した領域がピクトグラムである確率が高いことを示している。信頼度の閾値は 0.7 としており、信頼度が 0.7 以上であれば、抽出された領域にはピクトグラムがあると予測され、その領域がバウンディングボックスで囲まれる。

また、学習・検証用の画像として、インターネット上で無作為に収集したピクトグラムが含まれる画像 210 枚と、伊藤が実際に街中で撮影した画像 128 枚を用いる。さらに、これらの画像全てに左右反転処理を行ったものを追加することで、画像枚数を 676 枚に拡張した。このデータセットを、学習用と検証用にそれぞれ 8:2 に分割し、学習データ 541 枚、検証データ 135 枚とした。また、使用する 676 枚の画像全てに、人手でクラスと位置のラベル付けを行っている。

SSD 方式の結果例を図 7 に示す。



図 7 SSD 方式の結果例

5.1.3 評価の概要

解説した 2 つの抽出手法についての評価を行った。テストデータには、インターネット上から無作為に収集した画像 75 枚のうち、SSD 方式における学習・検証データに使用されなかった 50 枚を利用した。以下、評価に用いるこの 50 枚を評価データセットと呼称する。

また、本手法の評価指標として、再現率と適合率を以下の様に定める。

$$\text{再現率} = \frac{\text{抽出に成功した枚数}}{\text{ピクトグラム枚数}}$$

$$\text{適合率} = \frac{\text{抽出に成功した枚数}}{\text{抽出された画像枚数}}$$

再現率は、評価データセット内に含まれるピクトグラム枚数を分母、その中で抽出に成功したピクトグラムの枚数を分子とした。適合率は、抽出された画像枚数を分母とし、その中でピクトグラムである枚数を分子とした。

5.1.4 評価結果

抽出成功の判断は、目視で行なった。領域抽出方式での再現率は 54/121 の 44.63%で、適合率は 54/97 の 55.67%であった。SSD 方式における再現率は 62/121 の 51.24%で、適合率は 62/161 の 38.51%であった。再現率で SSD 方式が領域抽出方式を上回るものの、適合率では領域抽出方式の

方が約 17%高くなった。表 1 に各抽出方式の評価結果を示す。

領域抽出方式において、その抽出方式の特性から、矩形で外枠を囲われたピクトグラムは検出しやすいが、「注意」や「禁止」マーク、外枠の無いピクトグラムといった矩形以外のピクトグラムに対しては、検出できない場合が多い。そのため、矩形の外枠を持たないピクトグラムの抽出においては SSD 方式の方が適している。しかし、領域抽出方式は領域情報をもとに抽出を行うことから、ピクトグラム抽出時の上下左右のズレがほとんど存在しないことが利点として挙げられる。

また、SSD 方式の適合率の低さとして、明らかにピクトグラムでない背景部分での誤検知が多いことが原因に挙げられる。SSD 方式の誤検知においては、信頼度の閾値設定と学習データを増やすことによる精度の上昇によって減少させることができるが、背景部分での誤検知が多いことから、学習データが不十分であることが原因と考えられる。

表 1 評価データセットを用いた各抽出方式の評価

抽出方式	再現率(%)	適合率(%)
領域抽出方式	44.63	55.67
SSD 方式	51.24	38.51

6. ノイズに対する分類処理の評価実験

6.1 評価の概要

実際の街中で使われているピクトグラムを撮影する場合、ピクトグラムを覆う障害物や汚れなどによるノイズを含む場面が想定される。そのため、本章では 3.2 節のテストデータに機械的にノイズを含ませることでノイズ耐性の評価実験を行った。

6.2 評価実験の詳細

本稿ではノイズの大きさを区別される 3 種類のテストで評価を行った。以下、この評価方法を「ランダムテスト」と呼称する。

ランダムテストは、3.2 節のテスト画像に対して正方形のノイズの付加を、ランダムに大きさと位置を設定することで行った。ノイズの幅は、それぞれ $W \times 1/4$, $W \times 1/3$, $W \times 1/2$ に設定し、高さも $H \times 1/4$, $H \times 1/3$, $H \times 1/2$ に設定した。ここで、 W と H はテスト画像元々の幅と高さのピクセル数を表し、ノイズの大きさによるテスト種別を区別するために、以下、それぞれのテスト種別を「 $W \times 1/4$ 」、「 $W \times 1/3$ 」、「 $W \times 1/2$ 」と呼称する。

ノイズの位置は、ノイズが画像外にはみ出ないようにランダムで設定する。ノイズ付加の例を図 8 に示す。この作業を全 180 種類のピクトグラム 1 クラスごとに 100 回行ったものを本章のテストデータとし、その正答率を評価した。

よって、このランダムテストにおける全テスト画像枚数は 180 (クラス数) $\times 100$ (1 クラスごとの画像枚数) $\times 3$ ($1/4$, $1/3$, $1/2$ の 3 種類) $= 54,000$ 枚となる。



図 8 ランダムテストのノイズ例
(左から $W \times 1/4$, $W \times 1/3$, $W \times 1/2$)

6.3 ランダムテストの評価結果

6.2 節で示したテストデータを用いて CNN の評価を行った。CNN は 3 章の実装を使用した。ランダムテストの結果を表 2 に示す。

表 2 6.2 節のテストデータを用いた CNN の評価

ランダムテスト種別	正解数/総数	再現率(%)
$W \times 1/4$	17,336/18,000	96.31
$W \times 1/3$	16,345/18,000	90.81
$W \times 1/2$	9,125/18,000	50.69
全体の正解率	42,806/54,000	79.27

「 $W \times 1/4$ 」、「 $W \times 1/3$ 」のテストにおいて、再現率はそれぞれ 96.31%、90.81%となった。しかし、「 $W \times 1/2$ 」の場合、再現率は 50.69%であり、本稿において実装した CNN の実用的な限界はおおよそ「 $W \times 1/3$ 」、つまりノイズが画像全体の $1/9$ 程度であると考えられる。

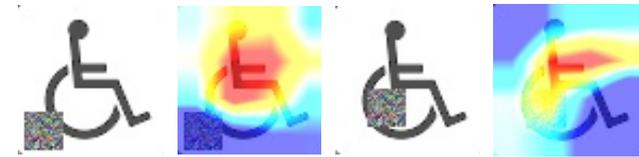
また、最も再現率が低い「 $W \times 1/2$ 」のテストにおいては、図 9 で示すように、ピクトグラムの中心がノイズで完全に隠れてしまい「注意」のマークだけしか手がかりがなく、目視であっても判別できないケースも多く存在する。このようなケースは CNN 側の問題ではなく、抽象度が高く、特徴量が少ないピクトグラム特有の問題である。



(1)感電注意 (2)障害物注意 (3)転落注意
図 9 ノイズにより目視でも判別できない画像例

さらに、Grad-CAM[16]を用いることで、ノイズの大きさが $W \times 1/4$ の一番小さいものであっても、その位置によってピクトグラムの識別がより困難になることが分かる。図 10 は、6.2 節のノイズ画像とそのヒートマップ (Grad-CAM

により作成)をペアとした例を2組示したものである。このヒートマップでは、色が赤に近づく箇所ほど画像の予測に対して貢献度が高いことを示している。また、(1)のノイズ画像は正しい識別に成功しているが、(2)のノイズ画像では正しい識別に成功していない。



(1)ノイズ画像1 Grad-CAM1 (2)ノイズ画像2 Grad-CAM2
図10 ノイズ画像とそれに対応するヒートマップの例

図10(1)のノイズ画像1では、ノイズの位置が左下にありピクトグラムにほとんど被さっていないため、しっかりと識別できていると推測できる。また、(1)のヒートマップ(Grad-CAM1)を見ると、赤と黄色の領域が画像の中心から広がっており、しっかりとピクトグラム全体を予測の根拠としていることが分かる。

しかし、(2)のノイズ画像では、ノイズが画像の中心近くにあり、これが正しい予測を妨げていると推測される。対応する(2)のヒートマップ(Grad-CAM2)を見ると、赤の領域がノイズ画像を避けるように(1)の半分程度に減少している。これはノイズ画像により予測の根拠が減少していることを示しており、実際に(2)のノイズ画像2では正しい識別には成功していない。

このテストにより、ピクトグラムはその性質上、高度に抽象化されており、判断のための特徴量が少ないという問題を抱えていることが判明した。また、「禁止」や「注意」のマークといった、共通するパーツにより構成されるピクトグラムが多いことも、誤認識が起こる原因と考えられる。図11にパーツとして頻繁に用いられる「禁止」「注意」「指示」のピクトグラムを示す。



(1)一般禁止 (2)一般注意 (3)一般指示

図11 多用されるピクトグラムのパーツ例

7. 結論と今後の展望

本稿では、深層学習を用いたピクトグラム画像における多クラス分類手法の評価と自動抽出処理の実装を行った。その結果、深層学習の学習データを構築する際に元画像が各分類1枚ずつのカラー画像を元に、左右反転、平行移動、

回転、透視変換の画像処理により作成された学習データセットであっても十分な正解率がとることが分かった。

さらに、本稿での手法を用いることで、ピクトグラムの情報補完を行うことが可能となる。例えば、3章と5章の実装を2段階で利用し、抽出と分類を行い、その結果を元画像に合成してピクトグラム画像への情報補完を行うといったものである。具体的には、ピクトグラム画像の下にCNN分類処理の出力結果(ラベル)を合成、表示するといった出力例が考えられる。その合成出力例を図12に示す。

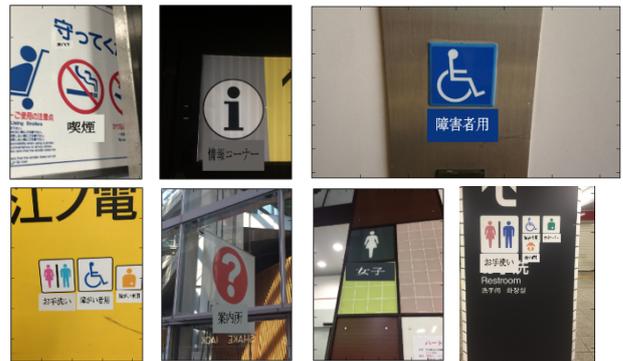


図12 自動抽出とCNN分類器を利用したアプリケーションの出力例

今後は、分類処理については学習データを増やしてさらなる検証を行い、比較実験を行う予定である。正解率向上の観点からは学習データの量、質の改善などを行っていききたい。ただし、ピクトグラムは街中でよく見かけるものとそうでないものの差が非常に大きい不均衡データであり、実データを用いた学習やテストにおいては、困難な部分も多く存在する。また、3章でも述べたように、「ピクトグラムには機械学習を目的としたデータセットが存在しない」という点や、「交通標識と比べて自由度が高い」といった点がGTSRBなどの交通標識分類問題との相違点として挙げられる。特に、後者の「交通標識と比べて自由度が高い」という点は、データセットが存在しないことも相まって、CNNの汎化性能を高める上での非常に大きな問題となっている。

さらに、5章で解説したピクトグラム自動抽出については、学習データを十分に用意することでSSD方式の再現率、適合率ともに、領域抽出方式を上回るものとなると予想される。今後は、精度の向上を目的としたデータセットの追加拡充を行っていく。また、アプリケーションとしての利用を見据え、より情報補完として効果的な出力の方法なども検討を進める予定である。

参考文献

- 1) 太田幸夫: 国際安全標識ピクトグラムデザインの研究 <http://www.tamabi.ac.jp/soumu/gai/hojo/seika/2003/kyoudou-ota1.pdf>
- 2) 大野森太郎, 西綾香, 原田利宣. 色彩表現を用いたピクトグラムにおける視覚言語の抽出とその検証. 日本感性工学会論文誌,

- 2015, vol.14, no.3, p. 391-400.
- 3) Mori, Y. Takasaki, T. and Ishida, T. Patterns in pictogram communication. In *Proceedings of the 2009 international workshop on Intercultural collaboration (IWIC '09)*. ACM, 2009, p. 277-280.
 - 4) 上西くるみ, 青木輝勝: ピクトグラムマッチングのための輪郭情報を取り入れた局所形状記述子, 情報処理学会研究報告コンピュータビジョンとイメージメディア研究会 Vol.2017-CVIM-205 No.5 (2017)
 - 5) Enass, M, M, H. The semiotics of pictogram in the Signage Systems. *International Design Journal*, 2015, vol. 5, no. 2, pp. 301-315.
 - 6) 藤森誠, 伊藤一成, Martin J. Dürst, 橋田浩一: ピクトグラムの群配置における感性的認識に関する検証, 日本感性工学会論文誌, Vol.8, No.1, pp.113-118(2008)
 - 7) 伊藤一成:ピクトグラミング - 人型ピクトグラムを用いたプログラミング学習環境 -, 情報処理学会論文誌 TCE, Vol.4, No.2, pp. 47-61, (2018年6月)
 - 8) LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature*. vol. 521, p. 436-444 (2015)
 - 9) 国土交通省 案内用図記号 (平成 31 年 2 月 20 日改正)
http://www.mlit.go.jp/sogoseisaku/barrierfree/sosei_barrierfree_tk_000145.html
 - 10) The German Traffic Sign Recognition Benchmark
<http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset>
 - 11) 交通エコロジー・モビリティ財団
http://www.ecomo.or.jp/barrierfree/pictogram/picto_top.html
 - 12) 吉田雄大, 伊藤一成: 深層学習を用いたピクトグラム画像への情報補完手法の提案, マルチメディア, 分散, 協調とモバイル DICOMO2017 シンポジウム, pp. 1075-1080 (2017)
 - 13) K. Simoyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, (2015)
 - 14) Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout, *J. Mach. Learn. Res.*, Vol. 15, No. 1, pp.1929-1958 (2014)
 - 15) Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot Multibox Detector", *Computer Vision-ECCV2016* pp. 21-37(2016)
 - 16) Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra, Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, *The IEEE International Conference on Computer Vision (ICCV)*, pp. 618-626 (2017)