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

吉田雄大[1] 伊藤一成[1]

概要： 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，ピクトグラムのデザインは国，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る．そこで，本論文では深層学習を用いてピクトグラム画像を解析し，ピクトグラム画像から，情報表現を想起させることで，ピクトグラムの解釈を補完する手法の提案と評価を行った．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 枚の白黒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．

Proposal of Complementary Information Method for Pictogram Images using Deep Learning

YUDAI YOSHIDA[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]．世界共通の記号表現として，世界中で用いられているが，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，急速にピクトグラムを題材とする関連研究が進み始めている[2]．

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る．そこで，本論文ではピクトグラム画像を解析し，ピクトグラム画像から，情報表現を想起させることで，ピクトグラムの解釈を補完する手法の提案を行う．

また，近年画像認識の分野で，深層学習[3]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事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．この事から，文化の違いなどによるピクトグラムのデザインへの対応でも異なったデータセットを用意する事で他の手法を用意せずとも，対応する事ができると考えられる．そこで，本論文では，DL を用いたピクトグラム画像への情報補完手法を提案する．

以下本論文の構成は 2，3 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．更に，5 章で簡易的なピクトグラム抽出処理の実装について言及し，6 章で 5 章の抽出処理と 3 章の分類器を組み合わせた評価について述べ，7 章でまとめる．

1 青山学院大学社会情報学部
Faculty of Social Informatics, Aoyama Gakuin University

2. 提案手法の概要

本論文で提案する手法はピクトグラム画像に対して DNN を用いて分類し，その分類結果を文字情報として画像に付加することで情報補完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．想定するシステムの入出力例を図 1 に示す．

図 1 想定する入力と出力の例

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．DNN による解析の後に解析結果を元に文字情報の付加を画像処理で行う．

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．本論文では 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

本論文の実装ではピクトグラム画像を 10 種類に分類するとする．10 種類のサンプル画像と内容を図 2 に示す．これらは，JIS の標準案内図記号(JIS Z8210)の中でも特によく使用されているものである．

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 Chainer(バージョン 1.21.0)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 3.1.0)を用いた．

図 2　分類画像種
(上段左から，案内所，情報コーナー，お手洗い，男子，女子．下段左から，障がい者用，喫煙，エレベーター，エスカレーター，階段．)

3.2　学習データの作成方法

学習データの元画像には交通エコロジー・モビリティ財団[4]が公開している公共・一般施設のピクトグラム 10 種を利用した（図 2 参照）．この画像は白黒画像となっている．これら 10 種の画像を元に様々な画像処理を施し画像数を増やす．本論文では 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，平行移動，回転，透視変換で，この順で画像処理を行った．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，ピクトグラムの大きな移動は想定されないからである．回転は画像の中心点を中心に-25 度から 25 度まで 5 度毎に変化させる計 11 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 0,5,10,15 ピクセルのいずれか移動する．つまり 4(隅)×4(段階)の 16 通りである．

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 5 枚のノイズ画像を使用した．以上の変換，合成を行うと 1 種類の画像が 9(平行移動)×11(回転)×16(透視変換)×5(ノイズ画像数)=7920 種類の画像となる．これを用意した 10 種類の画像に適応することで 79200 枚の画像データとなる．図 3 に学習データ作成例を示す．

図 3　学習データ作成例

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 3×70×70 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する．本論文での実装は 3 層の畳み込み層と 2 層の全結合層を用いている．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，3×3 の max pooling を行った．また，すべての畳み込み層で batchnormalization[5]を利用した．Dropout[6]を確率 0.5 で最初の全結合層に利用している．損失関数は交差エントロピーを用いた．活性化関数は ReLU を，最適化アルゴリズムは Adam[7]を利用した．学習はミニバッチ学習で行い，ミニバッチのサイズは 100 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 は 10 種類それぞれの分類確率を出力し，その内の最大値を分類結果として判断した．CNN の分類結果を元に，画像処理で，入力画像とあらかじめ用意した分類結果の文字列を表記した付与画像とを合成した．今回は分類結果の文字列には日本語を用いたが，利用者の使用言語に応じて他言語に差し替えることが当然可能である．合成は元画像の下部に分類結果の画像を結合する．この際に追加する分類結果の画像は入力元画像に合わせて背景色と画像サイズを加工している．背景色は，元画像の高さ h と幅 w に対して左上を原点とした時の画像の x 座標が αw，y 座標が $(1-\alpha)h$ である点の RGB 値と x 座標が $(1-\alpha)w$，y 座標が $(1-\alpha)h$ である点の RGB 値を取りその平均とする．ここで，α は 0 から 1 のパラメータである．今回 α には 0.1 を設定した．追加する分類結果表示画像の画像サイズは元画像の高さ h と幅 w に対して，高さ $h/4$，幅 w となるようにバイリニア補間を用いてリサイズする．

3.5　テストデータの作成方法

テストデータは，所属研究室の大学生 15 人に図 2 の 10 種類のピクトグラムを提示し，街中にある同様のピクトグラムを実際に撮影してもらうことで収集した．それらのピクトグラムの写真からピクトグラム矩形領域を目視で抽出した 216 枚の画像をテストデータに利用した．テストデータの内訳は案内所 11 枚，情報コーナー12 枚，お手洗い 29 枚，男子 22 枚，女子 19 枚，障がい者用 30 枚，喫煙 22 枚，エレベーター30 枚，エスカレーター24 枚，階段 17 枚である．テストデータは図 2 の画像が正方形，円形，三角形などの枠線に囲まれていても許容するものとした．喫煙に関しては，喫煙所と喫煙禁止が収集されたが，今回はこれらを区別せず，いずれも喫煙という分類にする．テストデータ例を図 4 に示す．このテストデータを用いて各々の実装の評価を行った．

図 4　テストデータ例
(左から，お手洗い，障がい者用，喫煙，喫煙)

3.6　実装詳細

ピクトグラムは比較的単純な図記号であり，学習データ

を作成する際にピクトグラム特有の性質を加味する必要がある．その点を踏まえ，以下の 3 種類の実装を行った．

実装 1：学習データに 3.2 節の方式で作成したものを利用，テストデータに 3.5 節で説明したものをそのまま利用．

実装 2：学習データに 3.2 節の方式で作成したものに加えて，それらを左右反転したものを利用，テストデータに 3.5 節で説明したものをそのまま利用．

実装 3：学習データに実装 2 のものとテストデータの一部を利用，テストデータに 3.5 節のものから学習データに利用したものを除いて利用．

以下 3.7 節から 3.9 節で各実装の詳細を述べる．

3.7 実装 1 の詳細

実装 1 の学習データには 3.2 節で作成した 79200 枚とそれらの画像をネガポジ反転したものを利用した．テストデータは 3.5 節のものをすべて利用した．したがって，実装 1 の学習データは合計 158400 枚で，テストデータは合計 216 枚である．

3.8 実装 2 の詳細

実装 2 では CNN の学習データに実装 1 の 158400 枚のデータを利用している．実装 1 では学習データをそのまま学習に利用したが，実装 2 では学習データとして CNN に各画像データを読み込ませる際に，50%の確率で左右反転処理を加えている．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，実装 1 だけではそういったピクトグラムには対応できないと想定されるからである．実装 2 のテストデータは実装 1 のものと同一である．したがって，実装 2 の学習データは合計 158400 枚で，テストデータは合計 216 枚である．

3.9 実装 3 の詳細

実装 3 の学習データは実装 2 と同様の 158400 枚に加えてテストデータの一部を利用した．3.5 節のテストデータは実際のピクトグラム画像であるため色情報が含まれている．そのため，3.5 節のテストデータの一部を利用することで，3.2 節の方式で作成した学習データには含まれていなかった色情報が活用できる．そこで，画像は 10 分類それぞれから 5 種類テストデータから学習データに転用した．その各分類 5 種類の画像を 3.2 節と同様の画像処理を行い 5×79200（5 種類×3.2 節の画像処理）の 396000 枚の画像データを追加の学習データとして用意し，合計 554400 枚とした．実装 3 では 3.5 節のテストデータの一部を学習データとして利用した．そのため，実装 3 のテストデータは 3.5 節のデータから学習に用いたものを除いた 166 枚である．男子のピクトグラムは青で，女子は赤が基本であるなど，色情報は重要な要素になると考えられるためである．

4. 評価

4.1 実装 1 の評価

3.5 節のテストデータ 216 枚を用いて実装 1 の評価を行った．各分類の再現率と出力例を表 1 と図 5 に示す．

表 1　実装 1 の正解数と再現率

分類種別	正解数/総数	再現率(%)
案内所	11/11	100.00
情報コーナー	11/12	91.67
お手洗い	26/29	89.66
男子	18/22	81.82
女子	16/19	84.21
障がい者用	29/30	96.67
喫煙	21/22	95.45
エレベーター	29/30	96.67
エスカレーター	12/24	50.00
階段	11/17	64.71
全体の正解率	184/216	85.19

(1)成功例　(2)成功例　(3)成功例　(4)失敗例

図 5　実装 1 の出力例

表 1 が示すように実装 1 の全体の正解率は 85.19%であり，また多くの項目で 90%以上の再現率を出すことができている．しかし，3.8 節で想定していた通り，ピクトグラムの左右反転が入力として想定されるエスカレーターと階段の再現率は 50%，64.71%と非常に低く出ている．また，男子と女子の再現率も 81.82%と 84.21%と比較的低い．特に再現率の低いエスカレーターの出力例を図 6 に示す．

(1)成功例　(2)成功例　(3)失敗例　(4)失敗例

図 6　実装 1 のエスカレーター出力例

図 6 で見られるようにテストデータの中にも階段やエスカレーターのピクトグラムには左右反転しているものがよく見られた．しかし，実装 1 では図 6 の左側 2 つの様な向きをしているものしか学習データでは想定されていない．したがって，図 6 の右側 2 つの様な向きをしているものの認識に問題が生じ，再現率が落ちていると考えられる．実際に目視でエスカレーターと階段のテストデータの中にある学習データで想定されていない向きのものだけを抽出して改めて再現率判定してみると，エスカレーターは 7.70%(1/13)，階段は 0%(0/7)と非常に低かった．

4.2 実装 2 の評価

3.5 節のテストデータ 216 枚を用いて実装 2 の評価を行った．各分類の再現率と出力例を表 2 と図 7 に示す．

表 2　実装 2 の正解数と再現率

分類種別	正解数/総数	再現率(%)
案内所	11/11	100.00
情報コーナー	10/12	83.33
お手洗い	26/29	89.66
男子	21/22	95.45
女子	17/19	89.47
障がい者用	29/30	96.67
喫煙	19/22	86.36
エレベーター	29/30	96.67
エスカレーター	23/24	95.83
階段	16/17	94.12
全体の正解率	201/216	93.06

(1)成功例　(2)成功例　(3)成功例　(4)失敗例

図 7　実装 2 の出力例

実装 2 の全体の正解率は 93.06%となり実装 1 に比べて向上した．特にテストデータに左右反転画像の多かったエレベーターと階段において大きく再現率が向上した．出力例を見ても実装 1 で失敗していた左右反転画像も実装 2 では正しく判断できていることがわかった．その他の分類種の再現率は実装 1 から大きな変化は見られないが，一部実装 1 に比べてわずかに低下している種別も存在するため，データ数を増やして更に評価を行う必要があると考えられる．また，男子で不正解であったものの分類結果は女子と判断されており．女子で不正解であったものの分類結果は全て男子と判断されていた．

4.3　実装 3 の評価

実装 3 では 3.9 節で示した通り 3.5 節で作成したテストデータの一部からランダムで各分類 5 種合計 50 種を選択し，CNN の学習データに追加している．3.5 節のテストデータを利用することで，3.2 節の学習データには含まれていなかった色情報を利用できる．また，実装 3 では 3.5 節のテストデータの一部を学習データとして利用したため，評価は 3.5 節のテストデータから学習データとして利用したものを取り除いた 166 枚の画像を利用した．評価結果を表 3 に，出力例を図 8 に示す．実装 3 の全体の正解率は 91.57%であった．実装 3 で期待された色情報による区別は男子の再現率は 94.12%であるが，女子の再現率は 78.57%となり，期待された成果は出ていなかった．

4.4　実装毎の比較

実装 1，2，3 の性能比較を行うために，テストデータを実装 3 で用いた合計 166 枚に統一して，性能評価を行った．結果を表 4 に示す．実装 1，2，3 で一番正解率の高いものは実装 2 となった．ピクトグラムの認識においては，学習データに色情報を含めなくても十分な正解率を出すことができた．更には色情報により誤分類率が高くなる可能性も示唆された．

表 3　実装 3 の正解数と再現率

分類種別	正解数/総数	再現率(%)
案内所	6/6	100.00
情報コーナー	7/7	100.00
お手洗い	23/24	95.83
男子	16/17	94.12
女子	11/14	78.57
障がい者用	23/25	92.00
喫煙	14/17	82.35
エレベーター	25/25	100.00
エスカレーター	19/19	100.00
階段	11/12	91.17
全体の正解率	155/166	93.37

(1)成功例　(2)成功例　(3)成功例　(4)失敗例

図 8　実装 3 の出力例

表 4　実装 1，2，3 の性能比較

分類種別	実装 1 再現率(%)	実装 2 再現率(%)	実装 3 再現率 (%)
案内所	100.00	100.00	100.00
情報コーナー	100.00	100.00	100.00
お手洗い	91.67	91.67	95.83
男子	76.47	94.12	94.12
女子	92.86	92.86	78.57
障がい者用	96.00	96.00	92.00
喫煙	100.00	100.00	82.35
エレベーター	96.00	96.00	100.00
エスカレーター	47.37	94.74	100.00
階段	58.33	91.67	91.17
全体の正解率	85.54	95.18	93.37

5. ピクトグラム自動抽出処理

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．本論文では 3 章の実装に重きを置いているため，この抽出処理は簡易的なものである．CNN を用いた一般物体検出アルゴリズムである R-CNN(Regions with CNN)[8]を参考に実装した．

5.1　抽出処理の流れ

抽出処理は大きく下の Step 1 ～ Step 3 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:　CNN でピクトグラム分類
Step 3:　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．候補領域は以下の式(1)，式(2)の両式を満たすもののみ抽出される．

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

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

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はパラメータで本論文では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 3 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．

5.2 CNN の構造

CNN は画像データを入力するとピクトグラムである確率とピクトグラムでない確率を出力するとした．ピクトグラムである確率が、設定した閾値以上であればピクトグラムであると判定している．CNN の構造は 3.3 節と同様である．学習はミニバッチ学習で行い，ミニバッチのサイズは 200 とした．

5.3 CNN の学習データ

CNN の学習データとして正解データと不正解データを用意した．正解データはピクトグラムの画像データであり，不正解データはピクトグラムが含まれていない画像データである．正解データは 3.9 節の実装 3 で用いた 554400 枚のデータを利用している．不正解データは事前に用意された画像データではなく，学習を行う際に自動的に作成される．不正解データの作成は次の手順で行われる．

Step 1: 背景画像の選択
Step 2: Step 1 で選択された背景画像の一部を切り抜く

背景画像は 53 枚のピクトグラムの含まれていない自然背景の画像である．用意した背景画像は全て高さ，幅ともに 256 ピクセルよりも大きいものとなっている．

Step 1 の背景画像の選択は 53 枚の背景画像から 1 枚をランダムに選択するとした．Step 2 では Step 1 で選択された背景画像の一部を切り抜く処理を行っている．本手法では背景画像を元に切り抜く画像の左上を原点とした x，y 座標と高さ，幅をそれぞれランダムに設定することで背景画像の切り抜きを行っている．高さと幅は 1 ピクセルから 256 ピクセルのランダム値を設定し，設定された幅のピクセル数をw，高さのピクセル数をhとする．x 座標は 0 から $W-w$(W:背景画像の幅のピクセル数)，y 座標は 0 から $H-h$(H:背景画像の高さのピクセル数) までのランダム値とした．ここまでの処理で設定された x，y 座標，高さ，幅に基づき背景画像を切り抜いた画像を不正解データとして作成する．

CNN のミニバッチはミニバッチのサイズ n 枚の半分を正解データに，残りの半分を不正解データに割り当てる．不正解データは処理を n/2 回行うことで n/2 枚の不正解画像を生成する．この正解データと不正解データの組み合わせをミニバッチとし，CNN の学習を行っている．

6. 自動抽出処理によるテストデータの評価

6.1 評価の概要

5 章で解説したピクトグラム自動抽出処理の評価を行った．テストデータには 3.5 節で作成したテストデータのうち，学習データに含まれず，ピクトグラムが矩形や円などの図形に囲まれているものを目視で選択し作成したデータセット 77 枚を利用した．また 5.1 節の式(1)，式(2)を満たさないピクトグラムは評価対象から除外した．

本手法の評価指標として再現率と適合率を以下の様にそれぞれ 2 通り定める．

再現率 1: $\frac{\text{抽出に成功した枚数}}{\text{分類対象 10 種のピクトグラム総数}}$

再現率 1 は，評価データセット内に含まれる分類対象 10 種のピクトグラム枚数を分母，その中で抽出に成功したものを分子とした．

適合率 1: $\frac{\text{抽出に成功した枚数}}{\text{抽出された画像枚数}}$

適合率 1 は，抽出された画像枚数を分母とし，その中で分類対象 10 種のピクトグラムであるものを分子とした．

また，抽出された画像の中には分類対象 10 種以外のピクトグラムも見られた．そのため再現率 2 と抽出率 2 を 10 種のピクトグラムに限定しない指標とした．

再現率 2: $\frac{\text{抽出に成功した枚数}}{\text{ピクトグラム枚数}}$

再現率 2 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 2: $\frac{\text{抽出に成功した枚数}}{\text{抽出された画像枚数}}$

適合率 2 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．

抽出結果例を図 9 に示す．

図 9 抽出処理の結果例

6.2　評価結果

抽出成功の判断は，目視で確認した．評価結果は 10 種のピクトグラムに限定した再現率 1 は 80/89 の 89.89%で，適合率 1 は 80/109 の 73.39%であった．10 種のピクトグラムに限定しない再現率 2 は 98/111 の 88.29%で，適合率 2 は 98/109 の 89.91%であった．再現率 1，2 では大きな差がないが，適合率では 1 と 2 で約 16%適合率 2 の方が高くなった．3 章の実装において今回は 10 種に限定したが，将来的には 10 種以上の分類を想定しており，汎用的に利用可能なことが示唆される．

6.3　抽出画像の分類

6.2 節で自動抽出された画像を用いて，3 章の実装の評価を行った．評価には 4.4 節で一番正解率の高かった実装 2 を利用した．その結果を表 5 に，出力例を図 10 に示す．

表 5　6.2 節の抽出結果を利用した 3 章実装の評価

分類種別	正解数/総数	再現率(%)
案内所	0/1	0.00
情報コーナー	6/8	75.00
お手洗い	7/7	100.00
男子	4/8	50.00
女子	3/5	60.00
障がい者用	14/15	93.33
喫煙	9/13	69.23
エレベーター	9/10	90.00
エスカレーター	7/7	100.00
階段	5/6	83.33
全体の正解率	64/80	80.00

(1)成功例　(2)成功例　(3)失敗例　(4)失敗例

図 10　6.2 節の抽出結果を利用した 3 章実装の出力例

全体の正解率は 80%となった．4.3 節からデータ数が減っているため単純な比較は難しいが，入力画像に乱れが生じる結果，正解率は約 13%低下している．その原因として 2 点考えられる．図 10 の(3)の失敗例にあるように 5.1 節の抽出処理の Step 3 を行った結果ピクトグラム単体ではなく，文字表記を含めたプレート全てが抽出される場合が見られ，この場合に誤分類が見られる．また，図 10 の(4)のように抽出の際にかなり位置ズレが生じてしまっているものにも誤分類が見られた．どちらも 3.2 節の学習データ作成の際に想定してなかったものであり，抽出処理と組み合わせる際には学習データの作成方法を再考する必要があるだろう．または，5 章の画像処理の正解率を高めることも有効であるだろう．

6.4　自動抽出と分類を利用した情報補完手法

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ピクトグラム画像への情報補完を行う．その合成出力例を図 11 に示す．

図 11　自動抽出と分類を利用した情報補完例

7. 結論と今後の展望

本論文では，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と評価を行った．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 枚ずつの白黒画像を元に，ネガポジ反転，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出ることがわかった．また，簡易的なピクトグラム抽出処理を実装し，3 章の分類器と組み合わせた情報補完手法の実装を行い評価した．

今後は，3，4 章については学習データを更に増やしてのさらなる検証を行い，比較実験を行う予定である．また，深層学習以外の手法を用いた手法との性能比較を行う．正解率向上の観点からは CNN の構造やパラメータの最適化や学習データの量，質の改善などを行っていきたい．5 章以下で解説したピクトグラム自動抽出は簡易的な手法であったため，より正解率の高く様々な状況に対応できる抽出手法の提案を今後行う．また，動画への適用や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] 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)
[3] LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. Nature. vol. 521, p. 436-444 (2015)
[4] 交通エコロジー・モビリティ財団 http://www.ecomo.or.jp/barrierfree/pictogram/picto_top.html
[5] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML (2015)
[6] Srivastava, N.,Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout, *J.Mach. Learn. Res.*, Vol. 15, No. 1,pp.1929-1958 (2014)
[7] Kingma, P. and Ba, Jimmy. Adam: A method for stochastic optimization. CoRR, abs/1412.6980 (2014)
[8] Girshick, R., Donahue, J., Darrell, T. & Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proc. Conference on Computer Vision and Pattern Recognition 580–587 (2014)