

駅構内の案内サインを対象とした 全天球画像による光学文字認識

比留川 翔哉^{1,a)} 丸山 一貴^{2,b)}

概要：文字認識は様々な場面で使用されている。我々は、駅構内にある案内サイン内の文字を認識することができれば、駅構内のナビゲーションに応用することができると考えた。本研究では、「文字認識を用いた駅構内のナビゲーション」という最終目標を達成するために、全天球カメラで駅構内の複数の案内サインを撮影し、その案内サインに含まれる文字を認識することを目的とする。本目的を達成するために、生成型学習法で生成した学習画像と畳み込みニューラルネットワークで文字認識する手法を提案する。文字認識の入力画像は、全天球カメラで撮影された全天球画像内の文字画像を用いる。本提案手法は、全天球画像から文字を抽出する3つのフェーズと文字認識、誤り訂正の合計5つのフェーズを使用する。駅で撮影した全天球画像を用いた評価実験では、看板抽出と文字認識では高い精度で抽出・識別できたが、辞書による誤り訂正は正しい認識を誤った結果に訂正してしまう結果となった。今後の課題として、認識率が低い文字種への対処と、文字セグメンテーションまでのフェーズにおける処理の改善を中心に進めていき、最終目標を達成することが挙げられる。

1. はじめに

近年の文字認識は、カメラで撮影した情景画像の文字認識を行う手法が活発に議論されている [1], [2], [3], [4], [5], [6], [7]。情景画像の文字認識の主な研究として、標識や看板内の文字認識やテキストスポッティングがある。これらの認識対象は、人間に情報提供をする目的で設置されたものである。我々は、同じ「人間に情報提供をする目的で設置された」看板である、案内サイン内の文字を認識できないかと考え、スマートフォンのカメラで撮影した案内サインを認識する手法を提案した [7]。案内サインは、図 1 のような駅利用者を目的地に誘導するために設置されている看板である。案内サイン内には、路線名や出口案内などの文字情報や、電車やトイレのマークであるピクトグラム、案内方向を示す矢印が表示されている。このような案内サインが設置されていても、スマートフォンで情報収集を行っている訪日外国人旅行客の 25%は「目的地へのルートや電車の乗り換えがわかりにくい」と回答し、日本人の 50%は駅構内で迷ったことがあるとアンケートで回答した。そのため、案内サインが設置されていても、駅利用者の一部はうまく利用できていないと考えられる。

現在の駅構内のナビゲーションを行うアプリケーションや研究は多くある。ほとんどのナビゲーションは、地磁気センサーと駅構内図の解析データなどの事前情報を利用する手法や、Wi-Fi や BLE ビーコンなどを利用して位置推定する手法などを用いている [8], [9], [10], [11], [12]。これらの手法は、正確にナビゲーションすることができるが、対応駅を増やすことが困難である。そのため、現在対応している駅数はかなり少ない。我々は案内サイン内の情報を利用することでこの問題は解決できると考える。本研究では、「案内サイン内の情報による駅構内のナビゲーション」を最終目標とする。この最終目標は、設定した目的地と駅構内の案内サインを歩行中に撮影し、取得した情報からナビゲーションを行う想定である。歩行中の案内サインの撮影にスマートフォンのカメラを使用した場合、進行方向以外の撮影は危険であるため行うことができない。そのため本研究では、進行方向以外の案内サインの情報を取得するために、全天球カメラを使用する。

本研究の目的は、最終目標の前段階である全天球カメラで撮影された案内サインの文字認識を行うことである。本提案手法は以下の5つのフェーズを用いて、全天球画像から文字認識結果の精度を算出する。

- (1) 全天球画像から看板領域の抽出
- (2) 全天球カメラによる歪みの緩和
- (3) 看板画像内の文字のセグメンテーション

¹ 明星大学大学院

² 明星大学

^{a)} 19mj003@stu.meisei-u.ac.jp

^{b)} kazutaka.maruyama@meisei-u.ac.jp



(a) 吊り下げ型案内サイン



(b) 壁設置型案内サイン



(c) 床設置型案内サイン

図 1 日本国内の駅にある案内サイン

(4) 文字認識

(5) 認識結果の誤り訂正

2. 関連研究

全天球カメラを用いた研究は、様々な分野で多数提案されている。全天球画像の補正では、屋外で撮影された情景画像を直線検出の結果から水平方向の補正を行う手法 [13] が提案されている。この手法は屋外では補正可能だが、屋内の情景画像では正しく補正できないため、本研究では使用できない。自己位置推定では、全天球画像の特徴量と GPS を用いて建造物内の位置推定を行う手法 [14] が提案されている。この手法は、通常のカメラで撮影した建造物内の画像の特徴と、全天球画像の特徴を比較し自己位置推定を行っている。駅構内のナビゲーションに応用させるには、駅構内すべての場所を撮影する必要があるため、この手法は使用できない。全天球カメラを用いたナビゲーションとして、点字ブロック上で撮影した案内サインを認識して目的地方向を通知するシステム [1] が提案されている。このシステムは、撮影画像内で注目する範囲に制限を設定した上で、取得・補正した案内サイン画像内の文字と矢印の情報や位置から算出される関連性を用いて目的地方向を特定している。本研究は、範囲の制限なく案内サインを取得する点と、矢印がない案内サインやピクトグラムを含む案内サインも対象とする点、点字ブロックの存在を前提と

しない点異なる。

看板検出には、HSV 色空間と矩形判定を用いて屋外の看板を検出する手法 [15] がある。この手法は、赤・緑・青の色相範囲と白黒の濃淡の特徴を用いて、カラー看板 98.1%、白黒看板 90.2%の精度で抽出した。本研究では、この手法に基づき案内サインを抽出する。

文字認識の研究として Android を搭載したミニカーとサーバを用いて案内サインを認識し、認識結果と RFID(Radio Frequency Identifier) を用いて目的地までの誘導を行う研究がある。この研究は、案内サインの文字を 2 値化し、サーバによる文字認識を行う。本研究と目的は同じだが、RFID が不要な点と、様々な形式がある日本語の案内サインを対象とする点異なる。情景画像の文字認識では、Attention 特徴量を用いた中国語の看板の文字認識 [6]、テキストスポッティングの研究がある [2]。これらの研究では、文字の位置と単語情報を含む膨大なラベル付きのデータセットを使用している。本研究は、生成型学習法と CNN の組み合わせで、データセットがない対象物でも文字認識することを目指している。

前章で記述した通り、既存のナビゲーションアプリでも駅構内のナビゲーションに対応したものが [8], [9]。しかしこれらのナビゲーションはごく一部の駅のみ対応しているため、ほとんどの駅で使用できない。また、視覚障害者をナビゲーションする研究 [10], [11] もあるが、事前情報や外部装置による位置推定を使用しているため、新たな駅に対応させるのは困難である。また、事前情報と GPS、物体認識を使用した東京駅のナビゲーションの研究がある [12]。これは、事前情報と GPS から屋内外の位置を特定し、建物の形状の情報を用いてナビゲーションの補助やユーザへの情報提供を行った。本研究では、カメラを使用した情報収集するという点は同じだが、案内サインの情報を使用する点と事前情報を使用しない点異なる。

3. 提案手法

本提案手法では、全天球カメラで撮影された案内サインの文字を認識するために、撮影方法と 5 つフェーズについて説明する。

3.1 全天球カメラを用いた撮影

本研究では、「案内サイン内の情報による駅構内のナビゲーション」を想定した方法で撮影された画像を用いる。撮影に使用するカメラは、2 つの魚眼レンズで前後を撮影する「RICOH THETA Z1」 [16] を使用する。撮影は、駅構内で歩行中の撮影者が全天球カメラを頭上に持ち上げる方法で行う。この撮影方法は、撮影者や駅利用者 の身体による案内サインの遮蔽を引き起こすことが少ないため、情報の欠落を減らす事ができる。

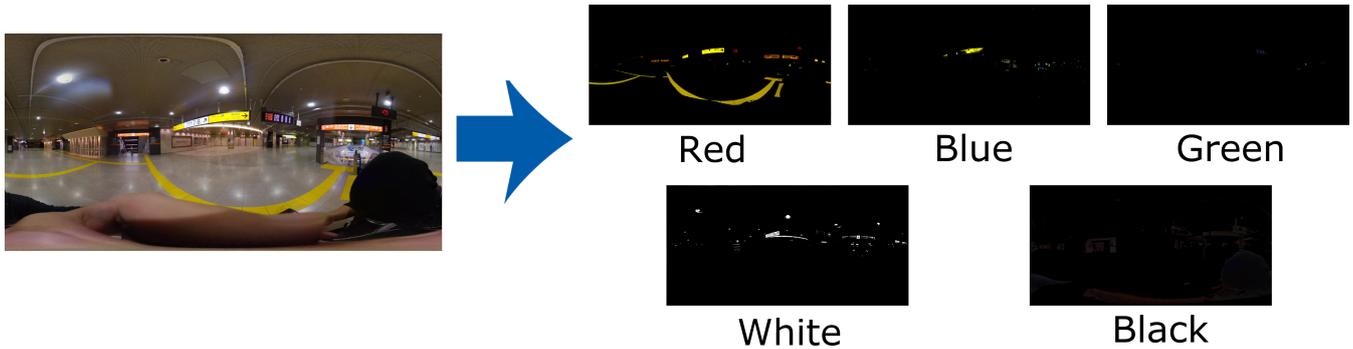


図 2 全天球画像と色相範囲を用いた各範囲の看板抽出結果

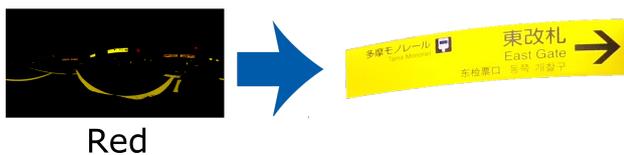


図 3 色相範囲を用いた看板抽出結果の一例

3.2 看板領域の抽出

本フェーズでは、全天球カメラで撮影された画像から既存研究 [15] の色相範囲を用いて案内サインを抽出する。この研究では、白と黒の看板は色空間を使用せずに濃淡から判別する手法を使用していたが、駅構内で撮影された全天球カメラの画像では正しく抽出できなかったため、今回は色相範囲の条件で抽出を行う。

案内サインの抽出には、選定した値を用いた色相範囲内の色のみを抽出するマスクを作成して、囲内の画像のみの画像を生成する。本提案手法で使用する値の彩度・明度の範囲は、JR 八王子駅、JR 立川駅で撮影した画像から選定した。セグメンテーションされた各画像は、床・壁・天井などの看板以外の画像も含まれる。これらは文字認識のフェーズまで看板と誤検知したまま進む可能性があるが、各フェーズの判定や文字認識の結果の信頼度から判定できるため、誤検知をある程度許容する。

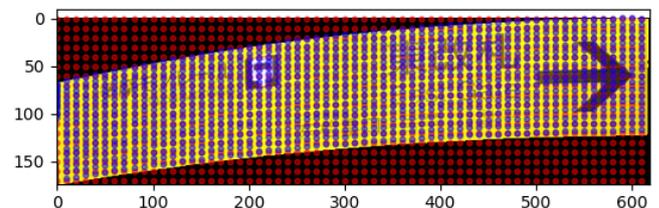
本フェーズによる処理の手順を図 2、図 3 に示す。図 2 の一番上の画像は入力画像である全天球画像、中央の 5 つの画像はそれぞれ赤 (Red)・青 (Blue)・緑 (Green)・白 (White)・黒 (Black) の色相範囲による抽出結果である。図 3 は、色相範囲の抽出結果から看板を抽出した結果の一例である。

3.3 歪み緩和

本フェーズでは、看板抽出された画像の外形情報を利用して全天球カメラの歪みを緩和する。この歪み緩和は、文字のセグメンテーションの精度向上のために行う。歪み緩和は、看板の外形の上辺と下辺から歪みを推定し、その歪みから補正後の座標群を生成して幾何学変換を行う。歪み



(a) 看板抽出で抽出された看板の外形を可視化した看板画像



(b) 外形から取得した歪みを基に算出した座標群 (赤が変換前、青が変換後)



(c) 歪み緩和後の看板画像

図 4 歪みの緩和の処理を行った例

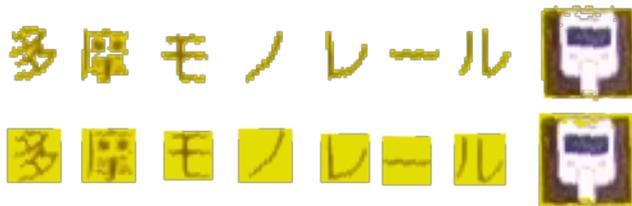
の緩和の処理のイメージを図 4 に示す。図 4(a) は抽出された看板画像の外形を赤い線で可視化した画像、図 4(c) は外形を可視化していない図 4(a) を図 4(b) の青点のような歪み緩和後の座標群で変形させた画像である。図 4(b) は、変形前の座標を赤点、歪み緩和に使用した座標群を青点でプロットした画像である。外形が正しく取得できた場合、図 4(c) のように正しく歪みを緩和することができる。

3.4 文字セグメンテーション

本フェーズは、案内サインの抽出時の色相範囲から推測できる背景色を削除し、案内サイン内にあるピクトグラムと矢印、文字の情報を抽出する。壁・天井・床などの画像はほとんどの場合、単色で抽出される。そのため、背景色の削除時に案内サイン以外の情報は破棄される。また、ノイズを多く含む画像や解像度の低い文字を含む画像は、日本語の抽出に悪影響を与えるため破棄する。背景色の削除



(a) 入力画像 (案内サインの一部)



(b) セグメンテーションの結果 (上:結合前の各文字パーツ, 下:文字の結合後の各文字)

図 5 文字セグメンテーション処理を行った例

して取得した文字画像は、複数のパーツに分かれる文字の場合、各パーツを結合し1文字にする必要がある。複数のパーツに分かれる文字は、「い」・「i」・「新」など各文字のパーツが結合されていないものである。これらの文字パーツは、そのパーツ付近の文字パーツを結合することにより、1文字に成形する。文字セグメンテーションの工程を図5に示す。図5(b)は図5(a)に文字のセグメンテーションを行った図である。本フェーズは、図5(b)の下部をセグメンテーション結果として出力する。

3.5 文字認識

3.5.1 学習画像の生成

学習画像の生成は、生成型学習法を用いて学習画像を生成する。学習画像は、原画像と歪みに対応した歪んだ学習画像の2種類を生成する。

原画像は、指定した文字種とフォントデータと生成条件を用いて生成する。文字種には、JR 新宿駅に使用されている文字種と、追加した全駅名と全路線名から作成したデータに基本となるひらがな・カタカナを追加した文字種の2つを使用する。JR 新宿駅に使用されている文字種を表1、全駅名と全路線名の文字種を表2に示す。また、使用した生成条件を表3に示す。

歪んだ学習画像は、想定した歪みを再現する変換式と原画像から生成する。想定した歪みを再現する変換式[2]は、以下の通りである。

表 1 JR 新宿駅の使用文字種と文字数

文字種	文字数
ひらがな	19 文字
カタカナ	19 文字
アルファベット (大文字)	17 文字
アルファベット (小文字)	24 文字
漢字	89 文字
数字	10 文字
記号	11 文字
矢印	4 文字
ピクトグラム	7 文字
合計	200 文字

表 2 日本の駅に使用されている文字種と文字数

文字種	文字数
ひらがな	75 文字
カタカナ	79 文字
アルファベット (大文字)	26 文字
アルファベット (小文字)	26 文字
漢字	1375 文字
数字	10 文字
記号	13 文字
矢印	8 文字
ピクトグラム	19 文字
合計	1631 文字

$$map_x(x, y) = \frac{x^{1+\gamma_x}}{w^{\gamma_x}} \quad (1)$$

$$map_x(x, y) = \left| \frac{(w-x)^{\gamma_x+1}}{w^{\gamma_x}} - w + 1 \right| \quad (2)$$

$$map_x(x, y) = x \quad (3)$$

$$map_y(x, y) = y - \frac{x^2}{h \cdot (n - \gamma_y)} \quad (4)$$

$$map_y(x, y) = y - \frac{(h-x)^2}{h \cdot (n - \gamma_y)} \quad (5)$$

$$map_y(x, y) = y \quad (6)$$

x, y はそれぞれ x 座標と y 座標、 h, w は画像の高さと幅、 γ_x, γ_y は横方向と縦方向の歪み係数である。この変換式は、書籍の厚さで湾曲した歪みを想定している[17]が、全地球画像の歪みに類似しているため調整した上で応用することとした。生成した原画像と歪んだ学習画像の一例を図6、想定される歪みと生成に使用した変換式の対応表を表4に示す。これらの変換式と条件を様々な原画像に適用して生成することにより、厳密に歪み緩和することなく全地球画像の文字を認識することができる。

3.5.2 CNN を用いた学習

本研究ではCNNを用いて文字認識を行う。使用したライブラリはKerasとTensorFlowである。使用したモデル

表 3 生成する文字画像のパラメータ

条件名	使用する条件
背景色	白 (#FFFFFF), 黒 (#000000), 緑 (#009821), 黄 (#FFD900)
文字色	黒 (#000000), 白 (#FFFFFF), 白 (#FFFFFF), 白 (#FFFFFF)
日本語フォント	新ゴ Pro M
英語フォント	ミリアド R
数字フォント	ミリアド R
英語フォント	ミリアド R
画像サイズ (pixel)	128 × 128, 64 × 64, 32 × 32 16 × 16
フォントサイズ (pt)	108, 54, 27, 13
矢印	8 種類
ビクトグラム	19 種類

表 4 想定される歪みと生成に使用した変換式

レンズと看板の相対位置	想定される歪み	式番号
中央より右	右下に湾曲した歪み	(1),(4)
	右に寄せた歪み	(1),(6)
	右側のみ湾曲した歪み	(3),(4)
中央より左	左下に湾曲した歪み	(2),(5)
	左に寄せた歪み	(2),(6)
	左側のみ湾曲した歪み	(3),(5)
中央	歪みなし	なし

(a) 原画像「あ」(歪みなし) (b) 歪み画像「あ」(右下に湾曲した歪み) (c) 歪み画像「あ」(左下に湾曲した歪み)

図 6 生成した学習画像の一例

は, VGG16 [18], 各ノードの重みの初期値は ImageNet で学習されたものを使用する. VGG16 は, 畳み込み層 13 層と総結合層 3 層の計 16 層で構成されている. VGG16 は 1000 クラスを分類するモデルだが, 生成した学習画像を使用することにより今回使用する文字種すべて識別することが可能である [2], [7].

3.6 文字認識結果を訂正する辞書の利用

文字認識結果を訂正する辞書は, 「駅データ.jp」 [19] と「Wikipedia (日本語版)」 [20] から入手できる情報から作成する. 作成する辞書は, 路線や駅名を含む辞書と駅周辺のランドマークや出口情報などを含む辞書の 2 つを作成する. 路線や駅名を含む辞書は「駅データ.jp」から作成する. ランドマーク辞書は「駅データ.jp」の駅名で抽出された Wikipedia のページから作成する. Wikipedia の駅のページには「== 駅周辺 ==」という見出しが含まれているので, その見出し内の箇条書き部分を使用する.

文字認識結果の訂正には, 上記の辞書と認識結果の上位 3 位までのラベルと信頼度から評価を行う. 評価には以下の式を使用する.

$$w_{eval_i} = \frac{1}{Lv(w, dict_i) + 1} \prod_{k=0}^{len(w)} C_{acc_k} \quad (7)$$

w は認識結果後の各ラベルを結合した単語, w_{eval_i} は辞書の i 番目の単語と結合した単語の評価値, $dict$ は辞書内の単語リスト, C_{acc_k} は k ラベル目の信頼度, Lv はレーベンシュタイン距離を求める関数, len はリストの要素数を求める関数である. この式を用いて, 各ラベルを結合した単語と辞書内の単語とのレーベンシュタイン距離と, 各ラベルの信頼度の総乗を計算し, 評価値が高い単語を訂正結果とする.

4. 事前実験

事前実験では, 事前調査と調整に使用した JR 八王子駅と JR 立川駅の写真を用いて提案手法の有効さを検証した.

目視で内容を読み取れた看板領域の抽出は, 84.6%の精度で正しく抽出できた. 抽出できなかった案内サインはなかったが, 照明の影響による案内サインの分割や, 内部に照明を持たない黒の案内サインの文字部分のみの抽出などが確認できた.

文字セグメンテーションは, 抽出された案内サイン内の文字と 1 文字に分割できた文字は 22.8%となった. これは, 床設置型案内サインの背景色特定が失敗し, 文字色まで削除したためである. 吊り下げ型案内サインのみに限定した場合 71.4%の文字のセグメンテーションを行うことができた.

文字認識の精度は, 前フェーズでセグメントされた文字を用いた場合, JR 八王子駅の学習済みデータでは 83.3%, JR 立川駅の学習済みデータでは 78.6%である. 本提案手法の認識器は日本語を正しく認識できたが, 特徴量の少ない記号や, 画像サイズが極端に小さい画像の一部を正しく認識しなかった. 正しく認識した場合でも信頼度が極端に低い結果となった.

誤り訂正は, 78.6%の精度で正しく誤り訂正が行えた. 辞書内にある単語は, 文字認識の精度が高かったため高精度で訂正が行えたが, セグメンテーションに失敗して単語内の一部が欠損した単語や辞書にない単語では誤った訂正を行う傾向が確認された.

5. 実験

事前実験を踏まえて各種パラメータを調整し, 本実験を行った. 本実験では, JR 新宿駅で撮影した 5 種類の画像を使用して本提案手法による各精度を算出した.

表 5 案内サイン抽出枚数と精度 (案内サインの数:枚, 精度の単位%)

画像名	A	B	C	D	E	
全て	抽出枚数	9	13	12	12	4
	看板枚数	18	41	19	21	6
	抽出精度	50.0	31.7	63.2	57.1	66.7
可読	抽出枚数	8	7	10	9	4
	看板枚数	10	10	17	15	6
	抽出精度	80.0	70.0	58.8	60.0	66.7

表 6 文字セグメンテーションの結果の一部

看板名	a	b	c	d
看板内の対象文字数	1	68	17	2
複数文字に分割した文字数	0	62	16	1
結合に失敗した文字数	0	1	0	0
1文字に分割できた文字数	1	5	1	1

表 7 手動でセグメンテーションした画像を用いた JR 新宿駅の学習データの文字認識の結果 (精度の単位:%)

文字種	使用した文字数	認識した文字数	精度
英語	9	3	33.333
矢印	8	7	87.500
ピクトグラム	9	6	66.667
数字	3	0	0.000
ひらがな	6	5	83.333
カタカナ	3	3	100.000
漢字	29	28	96.552
総合	67	52	77.612

5.1 看板領域の抽出

看板領域の抽出では、目視で内容を読み取れる案内サインは正しく抽出できた。全地球画像内に確認できる案内サインの数と抽出できた枚数と抽出精度を表 5 に示す。目視で内容を認識できる案内サインの約 66%の案内サインを抽出することができた。

5.2 文字セグメンテーション

本フェーズは、前フェーズで抽出された全ての画像から文字のセグメンテーションを行った。セグメンテーションできた文字数を表 6 に示す。ほとんどの画像において正しく背景色が削除できなかったため、文字セグメンテーションできた画像は少ない結果になった。

5.3 CNN を用いた文字認識の精度

文字認識は、前フェーズの文字セグメンテーション精度が低いため、文字セグメンテーションが正しく処理できた場合の精度を算出するために手動でセグメンテーションを行った画像を用いて精度算出を行った。手動でセグメンテーションした画像と各学習済みデータを用いた認識結果を表 7、表 8 に示す。漢字と矢印・ピクトグラムは高い精度で認識できたが、英語と数字の認識精度は他の文字種の精度と比べ低い結果となった。

表 8 手動でセグメンテーションした画像を用いた全駅の学習データの文字認識の結果 (精度の単位:%)

文字種	使用した文字数	認識した文字数	精度
英語	9	0	0.000
矢印	8	7	87.500
ピクトグラム	9	6	66.667
数字	3	0	0.000
ひらがな	6	1	16.667
カタカナ	3	0	0.000
漢字	29	6	20.690
総合	67	20	29.851

表 9 誤り訂正の精度

単語の内訳		単語数
辞書内にある	訂正できた単語	2
	訂正できなかった単語	2
辞書内に無い	訂正できた単語	0
	訂正できなかった単語	6

5.4 文字認識結果を訂正する辞書の利用した実験結果

本フェーズは、前フェーズで算出された、上位 3 位までの認識結果と信頼度を用いて誤り訂正を行った。文字認識結果を訂正する辞書を利用した誤り訂正の精度を表 9 に示す。

6. 考察

6.1 全地球画像を用いた看板抽出

抽出できた案内サインの内訳と精度は表 5 のような結果になった。抽出された案内サインの一部を図 7 に示す。目視で読み取れる案内サインは全体で 65.5%の精度で抽出できた。しかし抽出されたほとんどの看板は、分割や欠損した状態で抽出された。この原因は、案内サイン内部や外部の照明により、1枚の案内サイン内で複数の色相範囲に分割されたためだと考えられる。案内サインが分割された例を図 8 に示す。図 8 は、内部の照明を受けて色相範囲の赤と白に分割された画像である。図 8(a) の取得できている部分は想定通り抽出できているが、図 8(b) は想定以上の照明を受けて白飛びしてしまった。分割される問題の解決方法は、撮影時のカメラの設定の改良や、HSV 色空間の範囲の追加調査と調整が挙げられる。

抽出できなかった画像として、内部に照明を搭載していない黒色の案内サインがある。この案内サインは、電車の出発時刻を表示する発車標や通路内の壁にある乗り場番号が該当する。これらは色相範囲・黒 (Black) で抽出することを想定していたが、正しく抽出できないものが多い。この問題は、白と黒の色相値は一意に定まらないため、この手法では全て正確に抽出することができないと考える。ただし、他の色相範囲内で該当の案内サインが抽出できている場合や、看板抽出時に文字のみを抽出できる場合が確認できた。そのため、複数の色相範囲を考慮した抽出方法や、



(a) 成功例: 下部に背景が入っているが歪み緩和可能

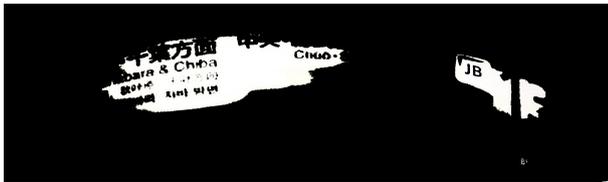


(b) 失敗例: 左側の上下の外形部分が内部照明で照らされていない

図 7 外形通り抽出できている案内サインとできていない案内サイン



(a) 色相範囲・赤の画像



(b) 色相範囲・白の画像

図 8 照明の影響を受けた案内サインの画像

看板と文字のみの 2 種類の画像が抽出される前提の看板・文字抽出の手法に変更することで、精度を向上できると考える。

6.2 全天球カメラによる歪みの緩和

外形情報を利用した歪み緩和は、看板抽出時に 1 枚の案内サインとして抽出できていないため、正しく行えなかった。図 7 を歪み緩和を行った画像を図 9 に示す。本提案手法は、外形が正しく取得できていない場合、正しく補正ができない。しかし、文字のセグメンテーションを考慮した場合、図 9(b) の一部文字は目視で読み取ることができるため、次の処理に及ぶ影響は少ないと考える。

別のアプローチとして、全天球画像を方向ごとに平面展開する方法 [1] も考えられる。これは、撮影者の正面や側面などに領域を限定し展開することで、歪みがない画像に展開することができる。しかしこの手法では、領域を限定する必要があるため、吊り下げ型案内サインが領域内に入らない可能性や、案内サインが複数の領域に含まれた場合に分割されてしまう可能性がある。色相範囲から案内サインの可能性のある領域を特定し、その領域中の各座標で平面展開して結合するといった手法が考えられる。



(a) 成功例: ほとんど補正できている



(b) 失敗例: 左側が破綻している

図 9 図 7 の歪みを緩和した画像

6.3 文字セグメンテーション

文字のセグメンテーションは、ほとんどの画像に対して正しく分割できなかった。主な原因は背景色の範囲の特定の失敗である。背景色の範囲の特定には、色相範囲の情報と看板画像の色相範囲内の色を使用している。看板画像の色相範囲内の色情報を使用している理由は、色相範囲のみ使用して特定した背景色より、各看板画像の背景に近い色を抽出するためである。しかし、案内サインが想定以上に照明の影響を受けた環境は、複数の背景色に分離する可能性がある。本提案手法では、色相範囲の 1 つの範囲から背景色を特定するため、複数の背景色に分離した案内サインの背景色の特定は難しい。そのため、背景色を様々な色相範囲から取得し、複数色の背景を削除できるようにする必要がある。

6.4 CNN を用いた文字認識

手でセグメンテーションした画像は、JR 新宿駅の学習済みデータでは高精度で認識できたが、全駅・全路線の学習済みデータは、一部の文字種を除き低い精度だった。学習済みデータによる精度差は、ラベル数の差によるものだと考えられる。JR 新宿駅はラベル数が 200 に対し、全駅・全路線はラベル数が 1631 と大きくかけ離れている。そのため、CNN モデルを適切なものに変更することや、GPS やナビゲーション時の設定などで駅を特定し、学習済みデータを切り替えて適切なデータで文字認識することで高精度で認識できると考える。

表 7、表 8 の認識結果から、文字種によって大きな精度差があることを確認した。案内サインにおける英語は、日本語や矢印、ピクトグラムより小さく表示されている。そのため接写しない限り 1 文字のピクセル数は日本語の半分未満になることが多い。数字は、比較的大きく記述されていることが多いが、どの数字も特徴量が少ないため、VGG16 のような比較的大きいモデルでは、畳込みの過程で特徴量がなくなってしまう可能性がある。英語と数字の精度を向上させるためには、学習画像の生成方法を全角文字と同様に横方向に伸ばす処理を行う手法や、抽出された案内サインのレイアウト情報を解析して各情報の言語を特定し学習済みデータを切り替える手法が挙げられる。

6.5 辞書を用いた認識結果の訂正

本研究では、駅名辞書と路線名辞書とランドマーク辞書を作成し、その辞書と認識結果の訂正ができるかを検証した。その結果、辞書内にはない単語が訂正できない問題や、正しい結果を誤った結果に訂正する問題が見られた。これは生成した辞書が不完全だった点と、訂正に使用した式が辞書に非常に強く依存している点が挙げられる。生成した辞書は、東西南北の出口や改札、「きつぷうりば」や「みどりの窓口」といった一般的に使用される駅構内の施設名や単語が含まれていないためである。そのため、一般的に使用される単語を絞り込み、辞書に追加することで解消できる。訂正に使用した式は、辞書に含まれている単語に訂正しようとするため、「新宿御苑」は駅名辞書内にある「新宿御苑前」駅に訂正した。辞書内の単語を重視しすぎないよう式(7)を調整する必要がある。

7. まとめ

本研究では、全天球カメラで撮影した駅構内にある案内サインの文字認識を行う手法を提案した。全天球カメラで撮影した全天球画像から看板を抽出する手法は、一部の黒色の看板以外ほとんどの案内サインを含む看板を抽出することができた。全天球カメラによる歪みを緩和する手法は、看板の外形を利用するため、外形が正しく取得できている看板の画像には有効だった。歪みを緩和した看板画像から文字をセグメンテーションする手法は、背景色を正しく削除できない場合があるため改善する余地がある。看板内の文字の認識は、正しくセグメンテーションされている文字に対しては有効であった。辞書を用いた誤り訂正は、辞書に記載されている単語を重視しすぎる傾向があり、改善が必要である。

今後の課題は、認識率が低い文字種への対処と、文字セグメンテーションまでのフェーズにおける処理の改善を中心に進めていき、最終目標である「文字認識を用いた駅構内のナビゲーション」を達成することである。

参考文献

- [1] 山中優太郎, 粥川青汰, 高木啓伸, 長岡雄一, 平塚義宗, 栗原聡, "One-Shot Wayfinding System: 360度スマートフォンカメラと矢印分析を用いた視覚障害者のための公共施設における方向決定支援システム", 第28回インタラクティブシステムとソフトウェアに関するワークショップ(WISS2020), pp.7-12, 2020.
- [2] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, *et al.*, "FOTS: Fast oriented text spotting with a unified network", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.5676-5685, 2018.
- [3] Qin Siyang, Bissacco Alessandro, Raptis Michalis, Fujii Yasuhisa and Xiao, Ying, "Towards unconstrained end-to-end text spotting", Proceedings of the IEEE International Conference on Computer Vision, pp.4704-4714, 2019.

- [4] Sayyan N. Shaikh, Neelakantha V. Londhe, "OCR Based Mapless Navigation Method of Robot", International Journal of Inventive Engineering and Sciences, Vol. 1, No. 8, pp. 6-12, 2013.
- [5] Dongjin Lee, Hosub Yoon, Myung-Ae Chung, Jaehong Kim, "Robust Sign Recognition System at Subway Stations Using Verification Knowledge", ETRI Journal Volume 36 Number 5, pp.696-703, 2014.
- [6] Kai Zhou, Yongsheng Zhou, Rui Zhang, Xiaolin Wei, "An Improved Convolutional Block Attention Module for Chinese Character Recognition", document Analysis Systems, pp.18-29, 2020.
- [7] Shoya Hirukawa, Kazutaka Maruyama, "Optical Character Recognition for Navigation Signs in Japanese Stations", 14th IAPR International Workshop on Document Analysis Systems (DAS), Short Paper, 2020.
- [8] Yahoo!JAPAN, "Yahoo!MAP", <<https://map.yahoo.co.jp/promo/>>, (参照 2021/01/15).
- [9] NAVITIME, サービス紹介ページ 駅構内ルート, <<https://products.navitime.co.jp/function/2535.html>> (参照 2021/01/15).
- [10] João Guerreiro, Dragan Ahmetovic, Daisuke Sato, Kris Kitani, *et al.*, "Airport Accessibility and Navigation Assistance for People with Visual Impairments", Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, pp.1-14, 2019.
- [11] João Guerreiro, Daisuke Sato, Saki Asakawa, Huixu Dong, *et al.*, "CaBot: Designing and Evaluating an Autonomous Navigation Robot for Blind People", The 21st International ACM SIGACCESS Conference on Computers and Accessibility, pp.68-82, 2019.
- [12] 市川裕介, 中村幸博, 中村泰治, 手塚博久, 瀬下仁志, 深田聡, 三井英毅, "2020 Airport/Station ~訪日外国人向け空港~駅でのおもてなし~", 映像情報メディア学会誌, Vol. 71 No. 3, pp.185-191, 2017.
- [13] 石井雄飛, 栗原徹, "全天球画像の円筒平面における直線検出を用いた水平補正", 情報処理学会第79回全国大会講演論文集, Vol. 2017, No. 1, pp.259-260, 2017.
- [14] Takeshi Yashiro, Haruhisa Hirayama, Ken Sakamura, "An Indoor Localization Service using 360 Degree Spherical Camera", 2020 IEEE 2nd Global Conference on Life Sciences and Technologies, pp.17-18, 2020.
- [15] 野村松信, "情景画像における看板抽出アルゴリズムの開発に関する研究", 秋田大学, 博士論文, 2015.
- [16] RICOH, "製品紹介 RICOH THETA Z1", <<https://theta360.com/ja/about/theta/z1.html>>, (参照 2021/01/15).
- [17] 比留川翔哉, 丸山一貴, "紙面の色と見開きの歪みを考慮した光学文字認識の実装と評価", 第18回情報科学技術フォーラム, No. 3, pp. 25-30, 2019.
- [18] Simonyan, Karen, and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition", arXiv preprint arXiv:1409.1556, 2014.
- [19] 駅データ.jp, 駅データ 無料ダウンロード サービス『駅データ.jp』, <<https://www.ekidata.jp/>> (参照 2021/01/15).
- [20] Wikipedia, メインページ, <<https://ja.wikipedia.org/wiki/%E3%83%A1%E3%82%A4%E3%83%B3%E3%83%9A%E3%83%BC%E3%82%B8>> (参照 2021/01/15).