

小型ドローンを用いた 背ラベルによる書籍の存否確認

Book Availability Verification through Spine Label Detection in Libraries with Small Drones

関 啓汰^{*1}, 山田 昌尚^{*1}, 土江田 織枝^{*1}
Keita SEKI, Masanao YAMADA, Orie DOEDA
^{*1}釧路工業高等専門学校創造工学科
^{*1}National Institute of Technology, Kushiro College
Email: yoshida@kushiro-ct.ac.jp

あらまし：図書館などの書籍の存否は、貸出しと返却の情報で管理されているが、無断持ち出しなどもあるため、定期的な目視による確認が必要となる。しかし、背の高い書架の確認作業は梯子や脚立といった補助器具を使わなければならないことも多く、転倒や落下などの安全面で問題が生じる。そこで本研究では、小型ドローンを用いて取得した書籍の背表紙の画像から、背ラベルに記してある書籍の請求番号を抽出し、その情報により書籍の存否確認を行うシステムの開発を目的とする。

キーワード：書籍の存否確認、背ラベル、請求番号、小型ドローン

1. はじめに

図書館は電子書籍等が普及した現在でも幅広い年齢層で利用者が多い。近年では利便性や安全性に配慮して、段数の少ない低背の書架も用いられるようになってきているが、規模の大きい図書館では多くの書籍を扱う都合上、背の高い書架が使われている。書籍の存否は、貸出しと返却の情報で管理されているが、無断持ち出しなどもあるため、定期的な目視による管理が必要になる。その際の実地確認作業は梯子や脚立を使用した高所での作業も生じるため、作業効率の低下や怪我などの危険が伴うことが予想される。そこで小型ドローンを活用することで、これらの問題を解消することを考えた。本研究では、書籍の背表紙に貼ってある背ラベルの情報を用いて、存否を確認するシステムの開発を目的とする。

2. 先行研究

小型ドローンによる書籍の存否確認の先行研究として、書籍の背表紙の画像を特徴量マッチングにより背表紙の画像のデータベースと照合するシステムを開発した^[1]。このシステムは、破れなどの劣化のある背表紙は照合がうまくいかないなどの問題があった。また、画像データによる照合の処理は負荷が大きく、ドローンの操作を同時に実現するためには、十分な処理性能を備えたパソコンが必要になり、利用に制限が生じた。



図1 Telloの外観

3. 提案システム

本システムは、パソコンとドローンはWiFiで無線接続し、画像やデータの処理などは、パソコン上に構築したシステムで行う。ドローンの操作や背表紙の画像の取得は参考文献[1]の手法を用いる。

3.1 Telloの操作と画像の取得

本システムで使用する小型ドローンはRyze社のTello(図1)で、軽量で初心者でも扱いやすく、飛行も安定している。本システムではTelloに搭載のカメラで撮影した画像をパソコンへ転送し、それを見ながらパソコン画面に表示している操作パネルによりTelloを手動で操作する。このようなパソコンによるTelloの制御は、公式のSDKであるTello-Pythonを用いており、Telloからの画像の取得には、Tello-Pythonに含まれるTello-Video^[2]を参考に実装している。パソコンの画面にはTelloのカメラ画像の他にTelloの操作パネルを開くボタンや、スクリーンショットのボタン、画面を一時停止するボタンなどを配置している。図2はTelloの操作パネルの画面で、これによりTelloの離着陸や旋回、ホバリングなどの動作を遠隔操作する。

3.2 背ラベルの構成

本研究で対象とする図書館の書籍の背ラベルは、図3の左図のような手書きのものと、右図のようなパソコンで作成したものがあるが、本システムでは



図2 Telloの操作パネル

右図のような背ラベルを対象とする。一般的な背ラベルは、図3のように3段構成で、上段は書籍の内容を表す分類番号、中段は著者の頭文字を表す図書記号、下段はシリーズものであれば何冊目かを表す巻冊記号で、これらを合わせて請求記号といい書籍を管理する情報である。尚、下段は対象ではない書籍は空欄となる。

3.3 背ラベルの情報の取得

本システムの開発環境は、全体の処理は Python3 で行い、画像処理は OpenCV4 を用いている。そして Google 社のオープンソースの光学式文字認識ソフトウェア Tesseract-OCR と pyocr を使用して、背ラベルの画像から請求記号を読み取り、それをデジタルデータに変換する。背表紙の画像から背ラベル領域を取得し、請求記号のデータ変換までの図4の STEP1 から STEP3 の処理を以下に示す。

STEP1. 背表紙の画像(図4左図)から OpenCV の `approxPolyDP()`関数を用いて、輪郭をより少ない点で近似する。その輪郭について、4つの角をもつ、それぞれの線のなす角の余弦の絶対値が最大のもの、面積の値などのいくつかの条件を満たしたものを矩形とする。取得した矩形領域を背ラベルの領域とし、抽出し、ファイル名を付けて指定のディレクトリに保存する。

STEP2. 背ラベルの画像から文字の部分を読み取りやすくするために、画像のグレースケール化と2値化の処理をする。

STEP3. Tesseract-OCR と pyocr により、背ラベルの画像から請求記号を読み取り、それをデジタルデータへ変換する。そして事前に作成した請求記号と本のタイトル名を紐づけしたデータベースと照合し、書籍の存否を確認する。

4. 評価実験

図5に示す書籍30冊について、本システムによる背ラベルの領域の取得と、背ラベルの画像から読み取った請求記号が正しかったかということについて評価実験を行った。



図3 背ラベル



図4 背ラベルの情報を抽出する過程

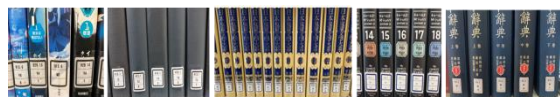


図5 評価実験の対象の書籍

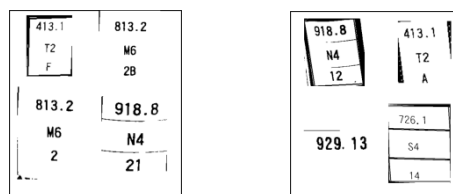
実験の結果、背ラベルの部分の画像の取得は30冊全てが行えた。そして取得した画像から請求記号を正確に読み取れたものは10冊で、その他の20冊は、読み取りがうまくいかなかった。図6(a)に請求記号の読み取りがうまくいった背ラベルの画像の一例と、図6(b)にはうまくいかなかった画像の一例を示す。また、図7は背ラベルの画像から Tesseract-OCR と pyocr を用いた請求記号の読み取りを行った結果、失敗した一例を示す。図7の請求記号を矩形で囲った部分が読み取った部分となっている。図7の左図は、請求記号が 918.8 N4 12 に対し、背ラベルの縦の線を文字として読み取ったため、正しく読み取れたのは、918.8 までだった。また、図7の右図は請求記号が 413.1 T2 B に対し、T を 1 と読み取ってしまい 413.1 までは正しく読み取れたが、そこからは正しく読み取ることができなかった。

5. まとめ

図書館の書籍の背表紙から背ラベルの部分を取得し、請求記号の情報を使って存否を確認するシステムの開発を行った。評価実験の結果から、背ラベルの枠の部分の内側と外側について、それぞれを矩形領域と判定してしまい、一つの背ラベルに対して2つの画像を抽出することがあった。また、背ラベルの画像から請求記号を読み取る際には、背ラベルの項目を区切る横の線や枠の線を請求記号の文字として誤認識することで正しく読み取りが行えないことを確認した。今後は、背ラベルの取得方法を見直すことや、背ラベルの枠や横の線を削除する方法を検討することで、正しく書籍の存否を確認できるシステムへと改良を進める予定である。

参考文献

- (1) 齋藤大夢, 土江田織枝, “小型ドローンを用いた書籍判別システム開発”, 釧路高専卒業論文, 2022.
- (2) “Python3 用の Tello Video ”,
<https://github.com/f41ardu/Tello-Python> (2024年1月31日参照).



(a) 成功した画像 (b) 失敗した画像

図6 取得した背ラベルの画像

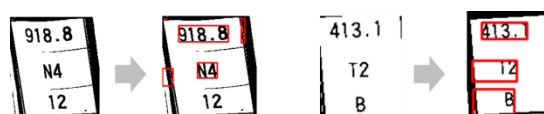


図7 請求記号の読み取りに失敗した例