修士論文　　平成 30 年度 (2018)

FCN による文書画像の
平面化処理に関する研究

東京工科大学大学院

バイオ・情報メディア研究科

メディアサイエンス専攻

林　揚彬

修士論文　　平成 30 年度 (2018)

FCN による文書画像の
平面化処理に関する研究

指導教員　渡辺 大地

東京工科大学大学院

バイオ・情報メディア研究科

メディアサイエンス専攻

林　揚彬

論 文 の 要 旨

論文題目	FCN による文書画像の 平面化処理に関する研究
執筆者氏名	林　揚彬
指導教員	渡辺 大地
キーワード	FCN，平面化処理，機械学習， 領域分割，幾何変換

[要旨]

近年，デジタルコンテンツ市場は飛躍的に成長しており，古書をデジタル化し，活用するというニーズが高まっている．電子書籍を作成する場合，スキャナーを利用すれば，効率的にできる．スキャンしやすいように，紙の書籍を断裁してページごとに切り離す．断裁されたページを，スキャナーを使って画像として読み込む．しかし，断裁できない書籍には，このようデジタル化方法が利用できない．その時，情報が平面上にあることではなく，湾曲した表面に存在する場合がある．そして，曲面上の情報に対して，直線検出や OCR(光学文字認識) などの処理を行うと，精度が低下するだけでなく処理自体が困難になると考えられる．そのため，従来の認識手法を行う前に，画像の平面化処理が必要となる．

本提案手法では FCN を用いたニューラルネットワークの学習を行い，入力画像の左ページと右ページ領域を分割するセグメンテーションの手法を用いた．学習の成果として，ページ領域分割ができるが，輪郭部分の精度が低いことが分かった．より精度高いの輪郭を検出するため，FCN の出力結果に対して画像処理方法を使ってページごどを分離し，ROI 範囲内の画面に対して輪郭検出を行う．検出した輪郭に対して，輪郭にある 4 つの頂点を検出し，輪郭を上下左右 4 つの線に分ける．4 つの線を曲線か直線かを判別する．誤った頂点検出結果を修正することができる．輪郭の曲線部分を利用し，グリッド生成する．書籍ページ領域をグリッドで小さい矩形を分割し，矩形画像に対して，射影変換を行うという手法を提案する．

評価として，学習ステップに対して，ページ分割の精度を分析した．平面処理について，学習結果の精度と頂点修正後の精度を比較した．より精度の高い頂点が得られた. 分割数による平面化処理への影響について検証した．文書に対して，OCR で評価した．認識精度が上がった．

A b s t r a c t

Title	The Research of Document Image Dewarping with FCN
Author	Yangbin Lin
Advisor	Taichi Watanabe
Key Words	FCN,dewarp,Machine learning, Segmentation,Geometric transformation

[**summary**]

In recent years, the digital content market has grown dramatically, the need to digitize old books is increasing. Cutting books of paper and separate them for each page,using the scanner to scan the page as image is efficient. However, this digitization methods can not be used for books that can not be cut. In this case, the information may be present on a curved surface rather than being on a plane. If processing the imformation which on the curved surface using such as straight line detection and OCR , it is considered that not only the accuracy deteriorates but also the processing becomes difficult. So image dewarping processing is required.

In the proposed method,We used the segmentation method to divide the left page and right page region of the input image by using learn base method of FCN .

As a result of learning, page area division is possible,It was found that the accuracy of the contour is low. In order to detect contours with higher accuracy, pages are separated using the image processing method for the output result of FCN and contour detection is performed on the image within the ROI range. Four vertex in the contour are detected for the detected contour, and the contour is divided into four lines up and down, right and left. Correct incorrect vertex detection result. Using the curve portion of the contour to generate a grid.We propose a method of performing projective transformation to the rectangular image.

As an evaluation, for the learning step, the accuracy of page division is analyzed. For dewarping processing, compare the accuracy of the learning result with the accuracy after correction. Evaluate documents by OCR.Recognition accuracy went up.

目 次

図 目 次

第 1 章

はじめに

1.1 研究背景と目的

電子書籍が世界的に大人気となる始まりとなった 2007 年の Kindle の登場から既に 10 年以上，2012 年の日本上陸から 6 年であり，スマホでマンガを楽しむ人が激増している．日本の本好きの間でも，電子書籍は読書のメディアとして受け入れられつつある．紙の書籍に比べて，電子書籍はスマホやタブレット端末があればすぐに読める．印刷代や人件費が安く済むのため，紙の書籍より安い．デジタル化のデータは場所を取らないので，持ち運びやすい．紙の書籍のように，手が汚れたり，日焼けしたりしない．検索と整理はかなりしやすいなどメリットがある．

インプレス [1] の調査では，2017 年度の電子書籍市場規模は 2241 億円と推計され，2016 年度の 1976 億円から 265 億円（13.4 %）増加している．2018 年度以降の日本の電子出版市場は今後もゆるやかな拡大基調で，2022 年度には 2017 年度の 1.4 倍の 3500 億円程度になると予測される．デジタルコンテンツ市場は飛躍的に成長しており，古書をデジタル化し活用するというニーズが高まっている．国立国会図書館は，資料の利用と保存の両立を図ることを目的に，平成 28 年 3 月に「資料デジタル化基本計画 2016-2020」[2] を策定した．

電子書籍を作成する場合，スキャナーを利用すれば，効率的にできる．スキャンしやすいように，紙の書籍を断裁してページごとに切り離す．図 1.1 は裁断機とスキャナーである．紙の書籍を断裁してページごとに切り離す．断裁したページをスキャナーを使って画像として読み込む．しかし，断裁できない書籍には，このようデジタル化方法が利用できない．その代わりに，非破壊的スキャンという手法がある．その時，情報が平面上にあることではなく，湾曲した表面に存在する場合がある．しかし曲面上の情報に対して，直線検出や OCR(光学文字認識) などの処理を行うと，精度が低下するだけでなく処理自体が困難になると考えられる．そのため，従来の認識手法を行う前に，画像の平面化処理が必要となる．

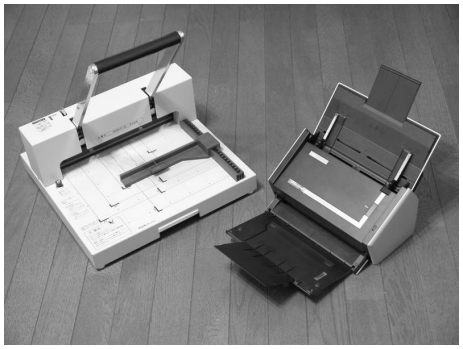

図 1.1　裁断機とスキャンナー

非破壊的スキャンに関する研究は 2 種類がある．第 1 のものは 3D 形状再構成手法である．見開き書籍の 3 次元形状を再構成するために，Brown ら [3] はプロジェクターとカメラを構成したシステムを提案した．Zhang ら [4] は非接触 3 次元デジタイザーを利用し，形状回復のための紙の物理的特性も考慮した．Meng ら [5] は 2 つのレーザービームを利用し，プラットフォームを設置して，曲面の特徴を獲得する．その他，3D 形状再構成のための多視点画像を利用した手法がある．Ulges ら [6] は画像パッチマッチングで 2 つの画像間の視差マップを計算した．山下ら [7] は NURBS で 3D 形状を構成した．Tsoi ら [8] は多視点画像からの輪郭情報を利用し，これらの画像を一緒に合成して，修正画像を生成した．Koo ら [9] は異なるカメラからの 2 つの未較正画像を使用して，SIFT マッチングによって 3D 形状を測定した．You ら [10] は，複数画像から紙上の折り目をモデル化することによって，文書の 3D 形状を再構成した．

第 2 のものは文書画像特徴を用いた形状推測手法である．文書画像特徴には照明，陰影，文字列などがある．和田ら [11] は書籍表面画像の陰影情報から, 書籍表面の 3 次元形状を復元する問題を扱う．Shape from Shading(SfS) という手法を提案した．Courteille ら [12] はスキャナの代わりにカメラを用いて SfS 手法を拡張し，遠近法による形状を推定した．Zhang ら [13] は影と背景ノイズの補正ができる，より強い SfS システムを提案した．その他，いくつかの手法は，文書内容の分析に依存している．正規化された文書のテキスト行が水平および線形でなければならないと仮定し，文字列の検出は一般的の手法である．特に Cao ら [14] は湾曲な文書を円筒上にモデ

ル化し，Liang ら [15] は可展面を使用した．Das ら [16] は単一画像から 4 つ折りを検出できる CNN を利用し平面化手法を提案した．

既存の非破壊的スキャンナー用のアプリケーションも様々なものがリリースされている．フラットヘッド型はコピー機でコピーを取るときのように，ガラス面に原稿を載せてスキャンするタイプのスキャナである．しかし，1 ページをスキャンするには 7 秒がかかる．ページをめくるのも大変である．もう一つはスキャナ本体に相当する台座付きのヘッド部と，背景マットから構成されるオーバーヘッド型である．図 1.2 は代表的な製品「Scansnap SV600」[17] である．しかしページ領域分割を上手くできない場合がある．修正するため, ユーザーは手動で書籍の画像の中央 2 点と四角の頂点 4 点を指定し, 輪郭検出の修正ができる．しかし, 読み込み枚数の増加と伴って, 手動修正時間がかかる．

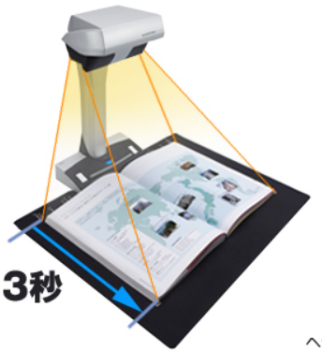

図 1.2　Scansnap SV600

参照 `http://scansnap.fujitsu.com/jp/product/sv600/vitechnology.html`

Adobe scan[18] というスマートフォンのカメラを利用したスキャンアプリケーションが開発されている．撮影された四角形文書の領域を検出することがでるが，書籍の見開き状態の画像には対応できない．まだ，右ページと左ページの分割することができない．Abbas ら [19] は畳み込みニューラルネットワークに基づく，カメラで撮影した文書の画像を透視変換で復元する手法を案した．この手法は自動的に文書の四角形領域の頂点座標を検出し，透視変換を行う end-to-end 手

法である．しかし，書籍画像などの湾曲した輪郭には対応できない．

近年ディープラーニングの手法を用いて領域分割が発表された．FCN（Fully Convolutional Networks）[20] は画像から物体をピクセル単位で予測をするネットワークである．

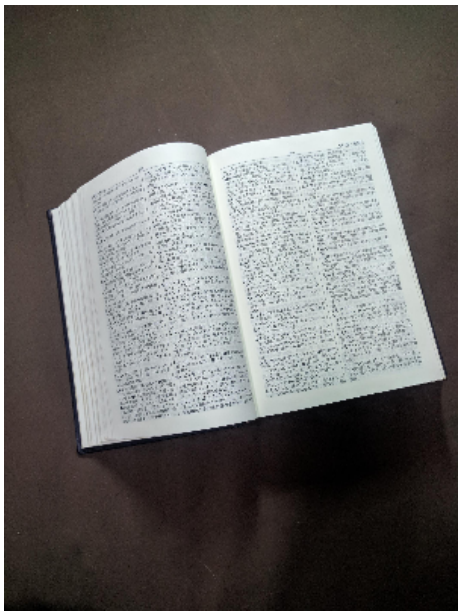

図 1.3　見開き書籍画像

カメラで撮影した 1 枚の見開き書籍画像を図 1.3 に示す．本研究では，このような画像に対し，ページ領域の分割が難しいという問題点に対して，RGB 書籍画像を入力，手動で作成したページ領域を FCN によって学習し，機械学習による解決方法に着目した．提案手法では，画像中のピクセルごとに前景（右ページと左ページ）か背景かの領域を予測するセグメンテーションを行う．学習の成果として，ページ分割ができるが，FCN による分割のみでは輪郭部分の抽出精度は低い．より精度の高い輪郭を検出するため，FCN の出力結果に対して画像処理方法を使ってページごとに分離し，ROI 範囲内の画像に対して輪郭検出を行う．検出した輪郭に対して，輪郭にある 4 つの頂点を検出し，輪郭を上下左右 4 つの線に分ける．4 つの線を曲線か直線かを判別する．誤った頂点検出結果を修正する．輪郭の曲線部分を利用し，グリッドを生成する．書籍ページ領域をグリッドで小さい矩形に分割し，矩形画像に対して，射影変換を行うという手法を提案する．評価と

して，学習ステップに対し，ページ分割の精度を分析した．平面処理については頂点修正の分析と分割数を比較した．文書に対して，OCR で文字列の認識精度に関する評価を行った．また既存手法との比較を行った．学習による分割精度は 95.6% であり，湾曲程度が大きい部分の頂点を正しく検出できた．分割数は処理時間に大きく影響があり，処理後の文字列の認識精度が上昇した．

1.2 論文構成

本論文では，2 章に FCN による領域分割について述べる．3 章で分割領域分割を利用する平面化処理説明を述べ，4 章で検証と評価を述べる．5 章で本研究における成果をまとめ，今後の展望を述べる．

第 2 章

FCN による機械学習

本研究では 1 枚の書籍画像の平面化処理をセマンティックセグメンテーション問題と幾何変換問題に分ける．本章では，FCN にようる書籍画像の領域分割について説明する．まず，2.1 節で FCN の概要とセグメンテーションの運用について述べる．2.2 節で学習するための学習データについて述べる．2.3 節で FCN の実装について述べる．

2.1　FCN による領域分割

セマンティックセグメンテーション (以下はセグメンテーションと呼ぶ) とは，画像に対してピクセルレベルでクラス分類を行う問題である．図 2.1 で入力画像とラベル画像を示す．ピクセル単位でオブジェクトごとに色付した教師データを使って学習を行う．そして，修論の際には入力画像のすべてのピクセルに対して，クラス分類を行う．

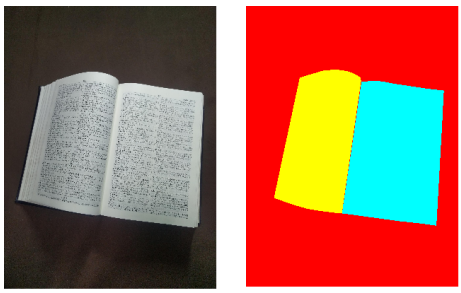

図 2.1　セグメンテーションの例：入力画像とラベリング画像

ニューラルネットワークによって，セグメンテーションを行う最も単純な方法は，すべてのピクセルを対象として，ピクセルごとに推論処理を行うことである．例えば，ある矩形領域の中心のピクセルに対してクラス分類を行うネットワークを用意して，すべてのピクセルを対象に推論処理を実行する．そのような方法ではピクセルの数だけ forward 処理を行う必要があり，多くの時間が必要になってしまう．つまり，畳み込み演算で多くの領域を再計算するという無駄な計算計算が発生してしまうことが問題になる．そのような計算の無駄を改善する方法として，FCN(Fully Convolutioanl Network)[20] という手法が提案されている．これは，1 回の forwards 処理ですべ

てのピクセルに対してクラス分類を行う．

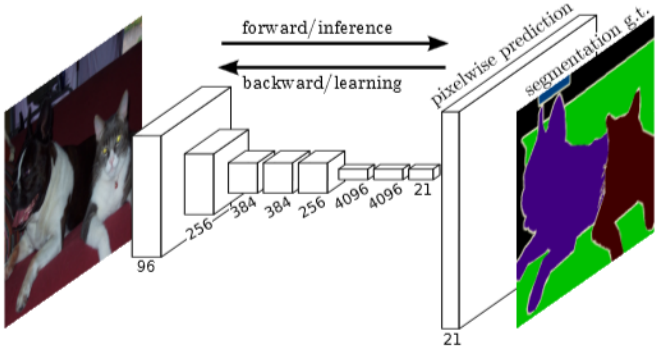

図 2.2　FCN の全体図

文献 [20] Fig.3 より引用

FCN は一般的な CNN が全結合層を含むのに対して，全結合層を同じ働きをする畳み込み層に置き換える．物体認識で用いたネットワークの全結合層では，中間データの空間ボリュームは一列に並んでノードとして処理するが，畳み込み層だけから構成したネットワークでは空間ボリュームは保たれたまま最後の出力まで処理することができる．

図 2.2 は FCN の全体図である．FCN の特徴としては，最後に空間サイズを拡大する処理を導入している点がある．この拡大処理によって，小さくなった中間データを入力画像のサイズと同じサイズまで一気に拡大することができる．FCN の最後に行う拡大処理は，バイリニア補間による拡大である．FCN ではこのバイリニア拡大を逆畳み込み演算によって実現している．

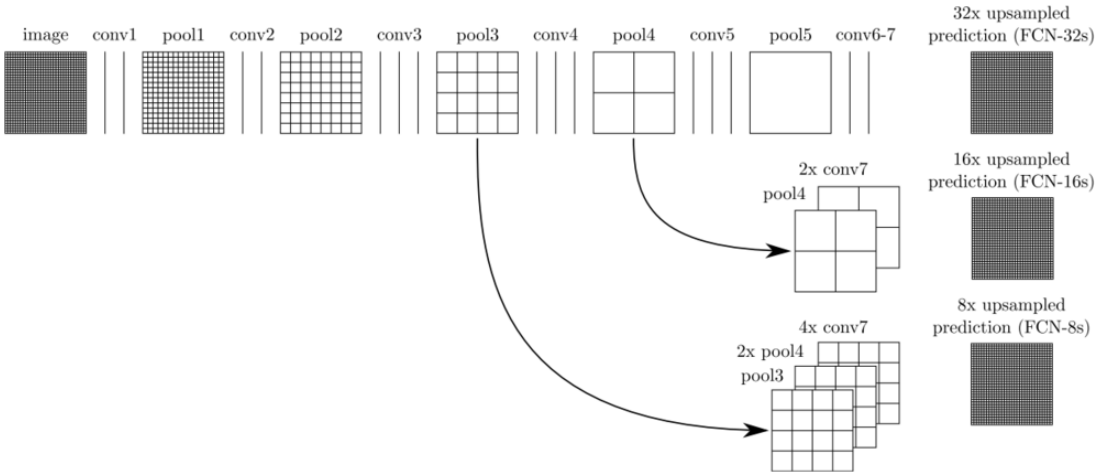

図 2.3　FCN の構造

文献 [20] Fig.3 より引用

FCN は主に VGG16[21]，全畳み込みネットワーク層と逆畳み込みネットワーク層で構成する．VGG16 の代わりに AlexNet や GoogleNet[22] などのネットワークモデルを利用することが可能である．もう一つの特徴は任意のサイズの画像の入力が可能になることである．全結合層に入力するマップのサイズが決まっているが，全結合層に変えることでサイズが変更することが可能になる．CNN は畳み込み演算を繰り返すに連れて画像の高次な特徴を抽出し，一方でローカルな特徴は Pooling を繰り返すに連れて失われている．下位のレイヤーはローカルな情報を持ち，上位のレイヤーは高次な特徴を持っている．FCN は下位のレイヤーの特徴マップをアップサンプリングした上位の特徴マップと加算する．これにより精度の向上が確認した．この手法は，FCN32s, FCN16s と FCN8s3 種類のモデルがある．図 2.3 で FCN の 3 種類モデルの構造を示す．FCN8s は一番精度が高く，複雑なモデルである．

2.2　学習データの作成

本研究において学習に用いる入力画像は，デジタルカメラやスマートフォンのカメラで撮影した見開き書籍画像である．まずは見開き書籍画像について説明する．書籍画像は 5 つの要素があ

ると考えた．

第 1 は書籍の内容である．小説や辞書など文字が多いの書籍が文字列検出方法が適用できるが，絵本や漫画など絵が多い文字が少ない場合，文字列方法が適用できない．

第 2 は撮影の背景である．書籍の色は白い場合が多くて，専用のスキャンナーは書籍の分割精度を上がるため，黒いシートを背景として使用する場合は多い，しかし，複雑な背景の場合は二値化等を行う際の閾値の設定などが難しい．図 2.4 に検出失敗例をあげる．

図 2.4　複雑な背景の領域分割

第 3 は撮影位置と角度である．専用のスキャナーでは本体を固定し，書籍の真上から撮影するが，デジタルカメラを手持ちで撮影する場合，真正面に撮影することが保証できない．同じ書籍でも角度の変化によって，映した湾曲画像が違う．

第 4 は照明である．照明の強度や角度は変化すると陰影が生じる場合がある．これは領域分割に影響がある．

第 5 は指のあるなしである．書籍が安定していない状態で，指で書籍を固定する場合がある，図 2.5 は指を含む画像の領域分割結果である，指と書籍の曲面画像と一緒に読み込むと，領域検

出が上手くできない場合がある．

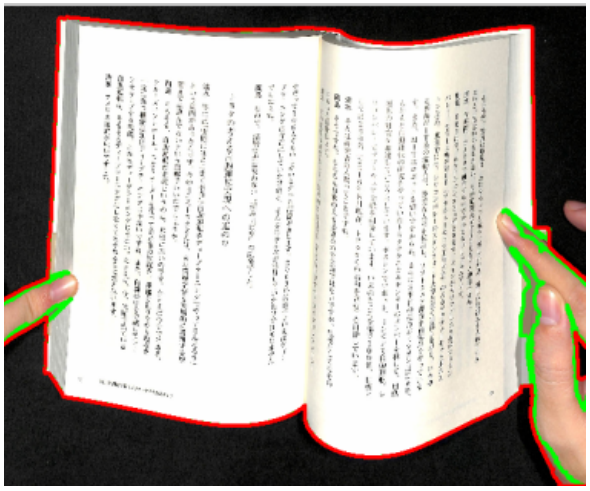

図 2.5　指を含む画像の領域分割

本研究では入力画像を 2 種類に分ける．左右ページを含めた見開き状態の書籍画像と単ページ（左ページと右ページ）の書籍画像である．図 2.6 に入力画像例を示す．その中には，見開き画像 93 枚と単ページ画像 413 枚，合わせて 506 枚を用意した．書籍の種類は辞書，絵本，雑誌，漫画，小説などありふれている紙の書籍である．FCN の汎化能力が低くならないように，多様な背景，照明，撮影角度の書籍画像を用いた．一部の画像には指がページ領域に重なっていた．

本研究では書籍画像の領域分割をセグメンテーション問題として考える．つまり，ピクセルレベルで前景としての左ページと右ページと背景 3 種類のオブジェクトに分類する．学習を行うため，教師データとしてのラベル画像を作成する必要がある．

図 2.6　入力画像例

本研究では LabelMe[23] を利用してラベル画像を作成した．LabelMe とは MIT で開発したセグメンテーションのためのアノテーションツールである．図 2.7 でラベル画像の作成を示す．Create Polygons 機能を使用して，手作業で書籍の領域にポリゴンで囲む．分割した領域は対象によって，左ページを 1，右ページを 2，その他の領域を背景として扱い 0 にするラベル画像を作成した．

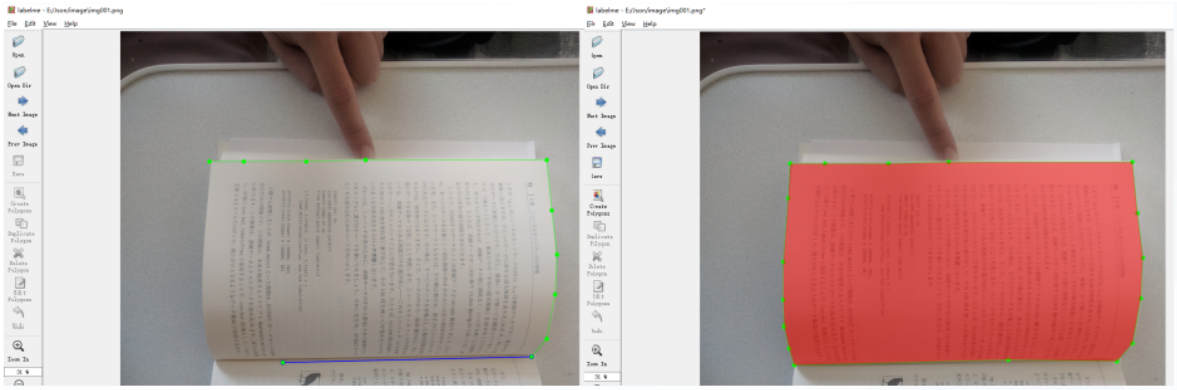

図 2.7　ラベル画像の作成

ラベル画像は入力画像と同じサイズ，画素値は 0,1,2 のみのデータである．図 2.8 は背景を赤，左ページを黄色，右ページを水色で可視化した様子である．

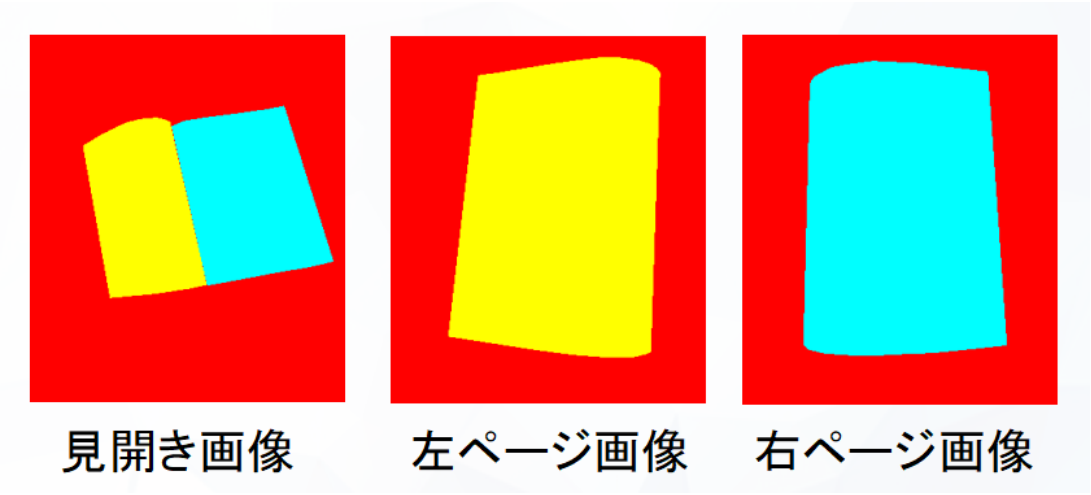

図 2.8　ラベル画像の可視化

入力画像に対する汎用能力の評価を行うため，430 対画像を訓練データとして学習に利用し，最適なパラメータを求めた．76 対訓練データ以外の画面をテストデータとして，モデルの汎用能力を評価した．

2.3　FCN の実装

本研究では VGG16 の代わりに VGG19 というモデルの畳み込み層を採用した．逆畳み込みネットワーク層は FCN8s の基本構造を採用した．画像サイズを 300 x 400 にリサイズして扱った．バッチサイズを 2 に設定した．GPU は GTX1070 8G を使用し，8 時間をかけてネットワークモデルのパラメーター更新を 5000 回行った．分類問題でよく使われる交差エントロピーを誤差関数に定義した．

学習段階の誤差関数の推移を図 2.9 に示す．テスト段階の誤差関数の推移を図 2.10 に示す．学習が進むにつ入れて，訓練データとテストデータを使って評価した精度は両方とも向上していることが分かった．

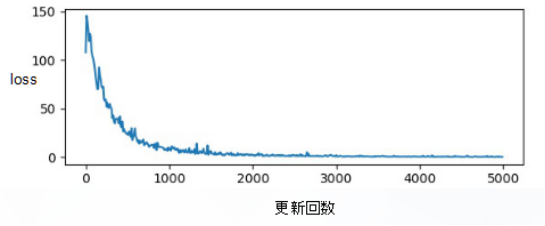

図 2.9　学習誤差関数の推移

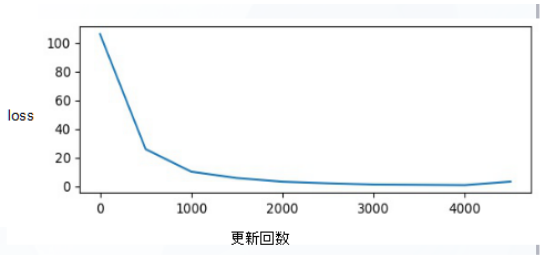

図 2.10　テスト誤差関数の推移

学習結果について，2.2 節で説明したように 506 対の学習データの中，430 対のデータを学習集合とし，残り 76 対のテストデータに対して実験を行った．一部のテスト結果を図 2.11 に示す．

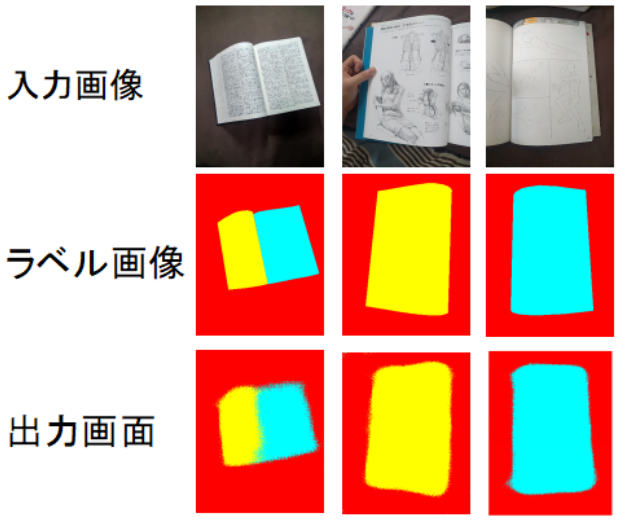

図 2.11　学習の実験結果例

第 3 章

射影変換による平面化処理

本章では FCN による領域分割の結果を利用する幾何変化手法を組み合わせる平面化処理を提案する．3.1 節で提案手法の概要を述べる．3.2 節で分割領域を利用する平面化処理について述べる．3.3 節で輪郭の検出について述べた後，3.4 節で頂点の検出と修正について述べ，曲線と直線の判別方法についてのべる．3.5 節でグリッド生成について述べる．3.6 節で輪郭を利用する射影変換について述べる．

3.1　提案手法の概要

図 3.1 に提案手法の 2 つの主要な処理ステップを示す．まず，RGB 書籍画像を入力，学習済みの FCN を利用し，画像中のピクセルごとに前景（右ページと左ページ）か背景かの領域を予測するセグメンテーションを行う．次に，FCN の出力結果に対して画像処理方法を使ってページごとに分離し，ROI 範囲内の画像に対して輪郭検出を行う．検出した輪郭に対して，輪郭にある 4 つの頂点を検出し，輪郭を上下左右 4 つの線に分ける．4 つの線を曲線か直線かを判別する．輪郭の曲線部分を利用し，グリッドを生成する．書籍ページ領域をグリッドで小さい四角形領域を分割し，分割画像に対して，射影変換を行うという手法を提案する．

図 3.1　提案手法の流れ

3.2　分割領域を利用する平面化処理

2.3 節で学習による書籍画像の領域分割ができることが分かった．この節は学習の出力結果を利用する平面化処理について説明する．輪郭を利用する平面化処理は先行手法がいくつか存在している．

N Stamatopoulos ら [24] はカメラで撮影した書籍の画像を 2 ステップ平面化処理という手法を提案した．まずは曲面上の文字列を検出し，検出した区域を矩形に変換して粗い平面化を行う．図 3.2 にその様子を示す．そして，単語の検出に基づいて細かい歪み修正処理を行う．本研究では同じように，ページ輪郭の曲線と直線部分を利用し，グリッドを作成し射影変換を行う平面処理手法を提案する．

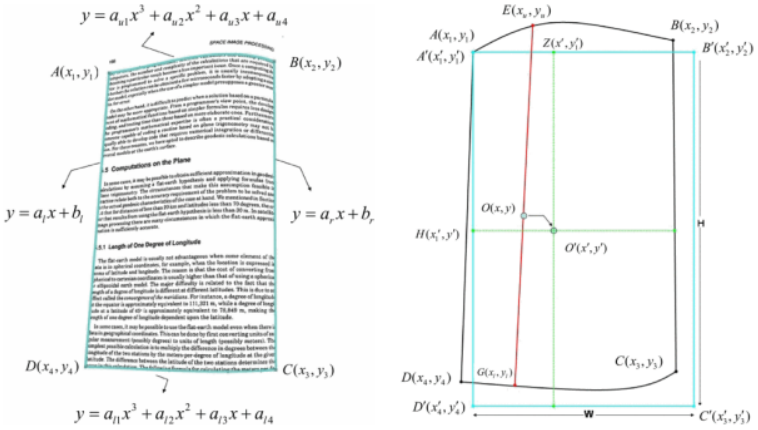

図 3.2　N Stamatopoulos らの提案手法

文献 [24] Fig.1 より引用

3.3　輪郭検出

学習による書籍画像の領域分割結果はそのまま利用することができない．実験結果の入力画像（上段左），正解ラベル（上段右），出力ラベル（下段左）と誤差画像（下段右）を図 3.3 に示す．誤差画像は正解ラベルと出力ラベルの差を求めた結果である．黒い部分は正しく予測した領域であ

る．ページ両方を分割することができるが，輪郭部分の精度が低いことが分かった．輪郭検出する前処理が必要である．

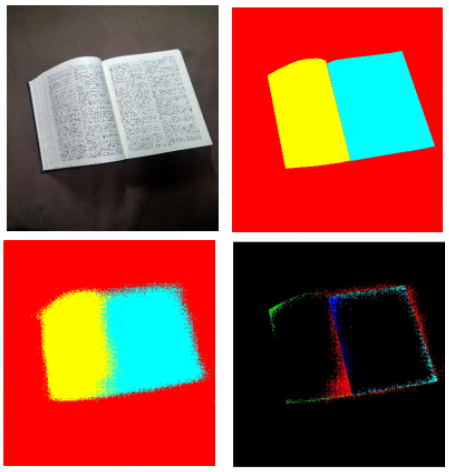

図 3.3　領域分割の精度

学習の出力結果はラベル画像と同じ 0,1,2 のみの画面データである．出力結果画像を P0 と呼ぶ．まず，P0 に対して画素処理を行う．左ページと右ページを分割するため，画素値は 2 を 0，1 を 255 に置き換える P1 と画素値は 1 を 0，2 を 255 にに置き換える P2 を生成する．図 3.4 にその様子を示す．

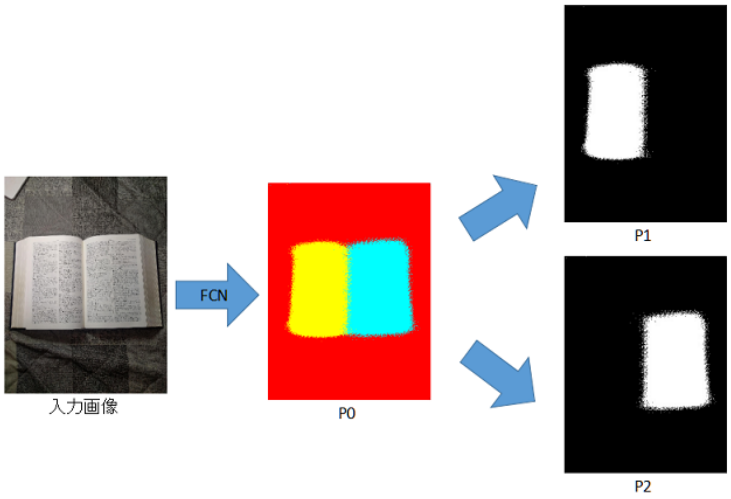

図 3.4　出力結果画像の処理

そして，P1 と P2 二値画像対して輪郭検出を行う．輪郭検出には鈴木ら [25] の輪郭検出アルゴリズムを利用した．ここで P2 の輪郭検出の結果を例とする．しかし，予測精度が低いの原因で領域の輪郭が連続していない．予測失敗の原因で書籍領域に離れる画素が存在する．検出した輪郭の数は一つではない．

この問題に対して，P2 にモルフォロジー変換 [26] を行う．モルフォロジー変換は二値画像を対象とし，収縮と膨張という処理である．まず，ノイズ（書籍領域に離れる画素）を除去するため，全要素の値が 1 の 5x5 サイズのカーネルを使用し，オープニング処理 (収縮の後に膨張）を行う．そして，全要素の値が 1 の 8x8 サイズのカーネルを使用し，膨張処理を行う．この処理は領域を増やし，輪郭が連続している．モルフォロジー処理した結果に対して輪郭検出を行う．検出した輪郭に対して外接矩形 ROI （Region of Interest-関心領域）[26] を求める．図 3.5 に結果を示す．ページ領域が ROI に収まることができる．

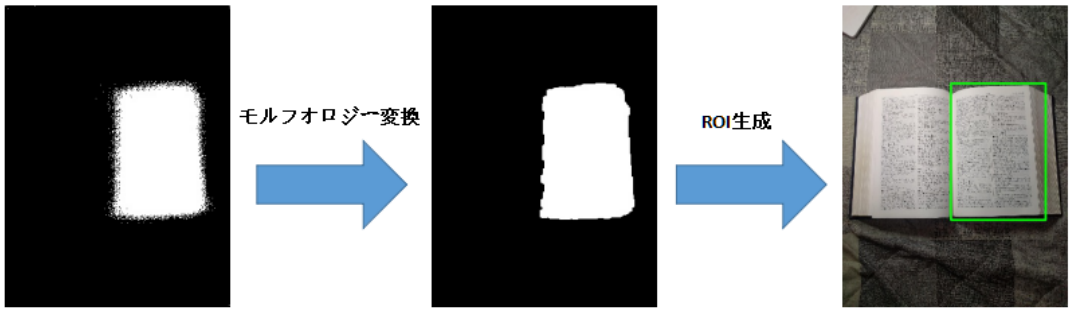

図 3.5　モルフォロジー変換と ROI

入力画像の ROI の範囲内に輪郭検出を行う．入力画像は P0 と違い，RGB 画像である．輪郭検出する前に二値化処理が必要である．そこで大津 [27] の適用型二値化閾値を採用した．結果として ROI に検出した輪郭を描画する．図 3.6 に結果を示す．精度より高い輪郭が検出することができる．

図 3.6　ページ輪郭検出

3.4　輪郭上の直線と曲線の検出

本研究では輪郭を利用し，曲面の立体形状を推定する．輪郭から得る曲面情報は 2 つがある．上と下の曲線と縦方向の直線である．そこで，検出した輪郭に対して，直線と曲線の検出が必要である．提案手法として，まず輪郭にある 4 つの頂点を検出し，輪郭を上下左右 4 つの線に分ける．4 つの線について曲線か直線かを判別する．

頂点を検出するため，輪郭の矩形近似によって，4 つの頂点を推定する．輪郭の近似方法は Douglas–Peucker アルゴリズム [28] を採用した．Douglas–Peucker アルゴリズムとは，線分で構成された曲線をより少ない点で同様の曲線に間引くアルゴリズムである．その手順を以下に説明する．図 3.7 にアルゴリズムの流れを示す．

1. 始点 S と終点 E による近似曲線を引き, 各頂点と近似曲線との距離を計算する.
2. 精度 ε よりも大きく, 近似曲線から最も距離がある点 N に次の近似曲線を引く.
3. N と S と E で近似曲線を引く.
4. すべての点が ε に収まるまで再帰的に繰り返す.

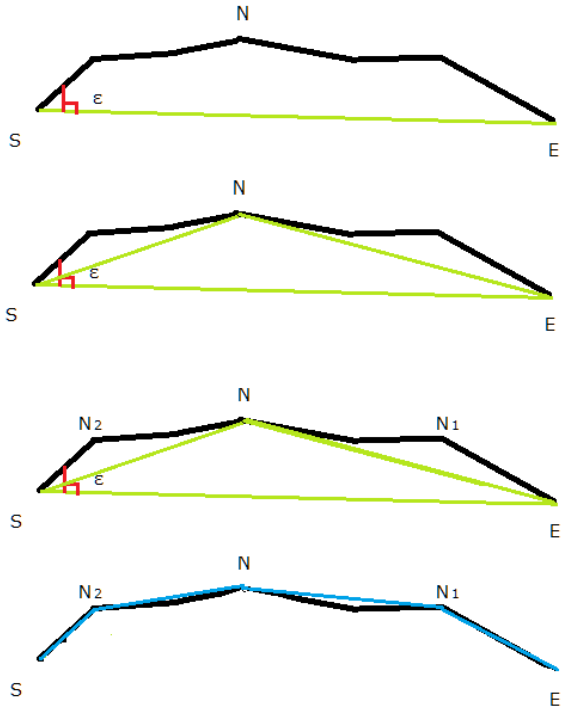

図 3.7　Douglas–Peucker アルゴリズム

輪郭を矩形で近似するの閾値ϵ を以下の式 (3.1) で定義する．

$$\epsilon = \alpha L_c \tag{3.1}$$

ここで L_c は輪郭の周長である．α は精度係数である．本研究では α を 0.06 に設定した．

図 3.8 で頂点検出の結果を示す．しかし，すべての理想的な頂点を検出できるとは限らない．この結果に対して，修正の必要がある．

図 3.8　頂点検出の結果

提案アルゴリズムは頂点修正だけでなく，輪郭の直線と曲線の判定することもできる．その手順を以下に説明する．図 3.9 で修正アルゴリズムを示す．

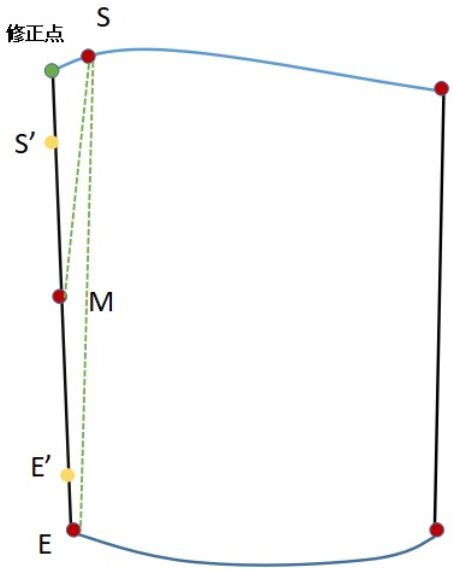

図 3.9　頂点修正アルゴリズム

1. 隣接している頂点を始点 S と終点 E にする．
2. $\widehat{SE}$ の中点 M を求める．
3. S,E,M の位置関係を判定する．3 点が共線をもつ場合，$\widehat{SE}$ を直線と判定する．3 点が共線をもたない場合，$\widehat{SE}$ 内部 S' と E' を求める．
4. S'，E'，M の位置関係を判定する．3 点が共線をもたない場合，$\widehat{SE}$ を曲線と判定する．3 点が共線をもつ場合，以下の頂点修正を行う．
5. $\widehat{SM}$ と $\widehat{EM}$ の直線曲線判定を行う．共線条件を満たさない弧の頂点 S もしくは E を修正対象とする．
6. 修正対象から弧内部に順次検索し，共線条件を満たす点を求める．この点を修正頂点とする．

共線条件は三角不等式を利用し，以下の式 (3.2) で定義する．ここで δ を 0.0015 に指定した．

$$|SM| + |ME| - |SE| < \delta \tag{3.2}$$

図 3.8 に対して，頂点修正を行った結果を図 3.10 に示す．青い点は輪郭近似法検出した結果．赤い点は修正後の頂点である．

図 3.10　頂点修正の結果

3.5　グリッドの生成

3.4 節で述べた方法で 4 つの頂点を得ることができた，この 4 つの頂点を利用し，輪郭曲線を上と下の曲線と縦方向の直線に分ける．ここで入力画像の前景境界は上下が曲線であり，左右は直線であることを前提として述べる．

グリッド生成には上下の曲線を利用する．上の連続画素を点集合を A$\{a_1, a_2, \cdots, a_n\}$ と表す，下の連続画素を点集合を B$\{b_1, b_2, \cdots, b_m\}$ と表す．点集合 A,B に対して，画素の座標によって左から右への順番で並べる．4 つの頂点を a_1, a_n, b_1, b_m で表す．図 3.11 で 4 つの頂点と輪郭曲線を示す．ここで上の曲線を構成する点集合 A を例として説明する．

図 3.11　4 頂点と輪郭曲線

まずは点集合 A の中すべて隣接している画素のユークリッド距離 L_k を求める．d は 2 点間の距離である．そして，曲線の周長は L_a とする．

$$L_k = d(a_k, a_{k+1}) \quad k = \{1, 2, \cdots, n-1\} \tag{3.3}$$

$$L_a = \sum_{k=1}^{n-1} L_k \tag{3.4}$$

曲線上の j 等分点を求めるため，始点 a_1 から a_k までの曲線の長さと曲線の周長の比を計算しておく．計算の結果を集合 P で表す．

$$P_c = \frac{\sum_{k=1}^{c} L_k}{L_a} \qquad c = \{1, 2, \cdots, n-1\} \tag{3.5}$$

集合 P の元に対して，j 等分点を検索する．以下の (3.6) 式を満足する元に対応する点を等分点として扱う．ここで δ を 0.001 に指定した．j 等分点を点集合 U で表す．

$$\frac{jp_c}{k} - 1 \geqq \delta \qquad k = \{1, 2, \cdots, j\} \tag{3.6}$$

点集合 B に対しても以上の方法で j 等分点を求める．j 等分点を点集合 D で表す．点集合 U と点集合 D 中同じ U_i と D_i 等分点で線を引き，縦方向のグリッド線を生成する．

そして，線分 U_iD_i に対して，線分公式を利用し，m 等分点を求める．同じよう等分点で線を引き，横方向のグリッド線を生成する．

図 3.12 で作成するグリッドを示す．赤線は輪郭の曲線である．緑線は縦方向のグリッド線である．青線は横方向のグリッド線である．ここで j を 10 に指定し，m を 15 に指定した．

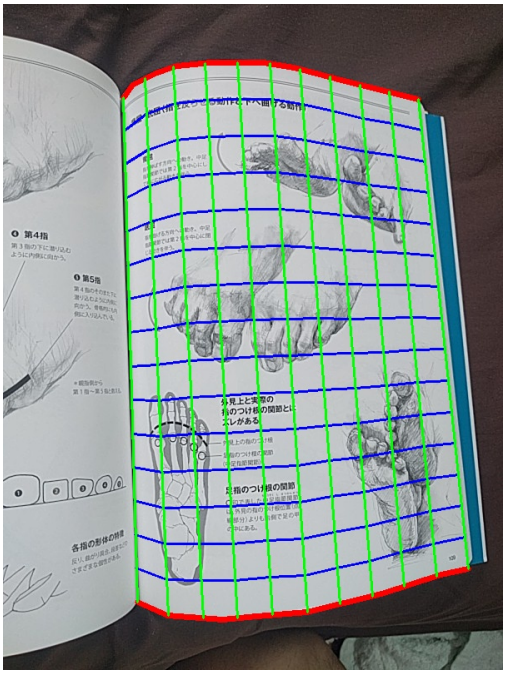

図 3.12　グリッドの生成

3.6　射影変換による平面処理

この節は射影変換による平面処理処理手法について説明する．まずは射影変換について説明する．

射影変換とは，3 次元座標系にある平面の座標を，投影面の 2 次元座標系に変換して対応付けることをいう．曲面画像による平面化では，歪みを持つ 2 次元のグリッド点を矩形となる 2 次元

の点になるよう平面を変形すればよい．この変形には射影変換を用いる．

グリッド点のうち 1 つの四角形を構成する領域をそれぞれ局所平面とする．局所平面の頂点となる 4 点を選択し，対応する 4 点の変換後座標を指定することで，射影変換に必要な変換行列を決定できる．

変換前の座標を (x, y)，変換後の座標を (u, v) とする．このとき射影変換は，式 (3.7),(3.8) と表すことができ，行列式としては式 (3.9) のように表現できる．

$$u = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1} \tag{3.7}$$

$$u = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1} \tag{3.8}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = H_m \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.9}$$

H_m は 3 × 3 の行列で表す射影変換のパラメータである．H_m によって画像の移動や回転, 拡大, 縮小を表現する．射影変換行列 H_m を式 (3.10) に示す．

$$H_m = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \tag{3.10}$$

H_m は 8 つのパラメータを持つ．変換前後の 4 点の座標を式 (3.9) に当てはめ，8 元 1 次連立方程式を解くことで射影変換行列 H_m を決定する．

本研究の平面化処理では，前節生成したグリッドで分割するすべての四角形領域を射影変換を行う．

まずは平面処理後のページの幅 W さと長さ H を確定する．式 (3.11) ,(3.12) は W と H の計算公式である．L_aと L_b は前節求めた上下の曲線の周長である．変換後の幅 W は上下の曲線の中

に，周長は一番長い方に設定する．通常の印刷物使用する A 判と B 判の長さと幅の比は $\sqrt{2}$:1 である．ここで係数 β を $\sqrt{2}$ に設定する．

$$W = \max\{L_a, L_b\} \tag{3.11}$$

$$H = \beta W \tag{3.12}$$

そして，変換前グリッドの四角形領域の 4 頂点座標に対し，同じ等分の変換後の矩形 4 頂点の座標を計算し，変換後座標とする．変換前後の座標により，射影変換行列 H_m を決定し，射影変換を行う．最後にすべてのグリッド領域を処理した結果を HxW 領域に合成し，ページ領域の平面化処理を完成する．図 3.13 は入力画像と平面処理の結果画像である．j と m は 10 と 15 に設定した．

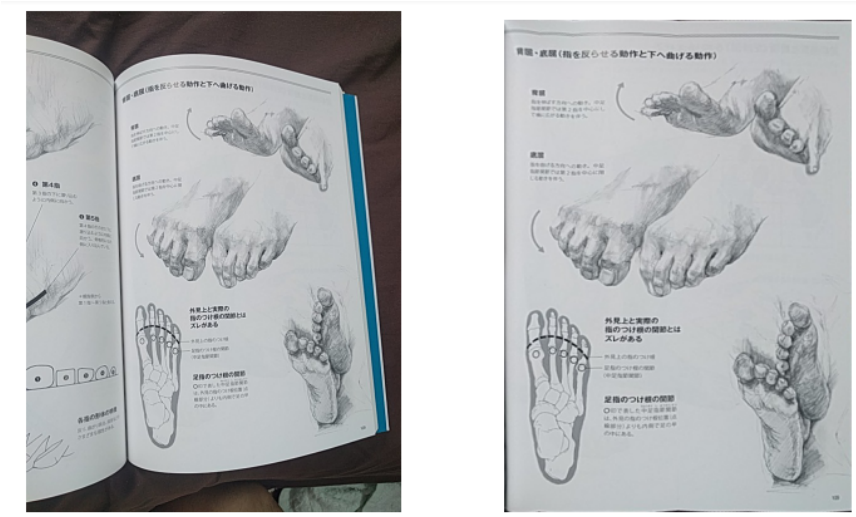

図 3.13　入力画像と平面処理結果

第 4 章

評価分析

本章では，提案手法の各段階の結果に対し，評価分析を行う．4.1 節では FCN の出力結果と精度について評価分析を行う．4.2 節では頂点検出と修正後を比較し，評価分析を行う 4.3 節ではグリッドの分割による平面処理への影響について検証する．4.4 節では文書に対して，OCR 結果と比較評価を行う．本研究では OpenCV と python[29] を用いて実装を行った．

4.1 FCN の出力結果

第 3 章では一部の出力結果を示した．ページ両方を分割することができるが，輪郭部分の精度が低いことが分かった．正解ラベル画像と比較してページ領域・背景領域の抽出精度を確かめた．教師データと出力結果とで前景背景のクラス分けが一致した画素を，正しく分割・抽出を行った領域とした． 教師データと実験結果の差分画像を求める．差分画像の中に，画素値は 0 ではない画素は抽出失敗の画素であり，これらの数を C_e とする．画像の全部画素の数を T とする．検出率を P と表す．式 (4.1) により求めた出力結果全体の平均の検出率は 95.6% である．

$$P = 1 - \frac{C_e}{T} \tag{4.1}$$

その他，いくつかのケースにおいて，領域分割が失敗した．両ページ画像に対し，領域が不完全であり，ページ領域内部を背景として認識したことがある．背景領域の色が白い，ページの色に近い部分が分割できなった場合や，撮影角度が水平ではない場合や，ページの表面が光沢があるなどがあった．図 4.1 でそれぞれの場合の失敗例を示す．76 対のテストデータの検証結果の中で，62 枚が適切に領域を抽出できた．主な原因は学習データの数が足りない，またはモデル構造と損失関数について改良する必要がある．

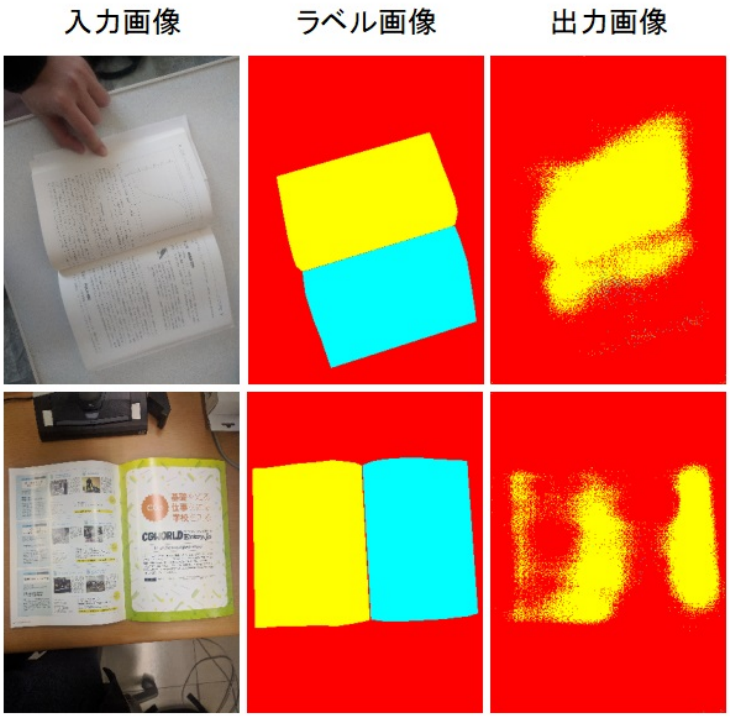

図 4.1　領域分割失敗例

4.2　頂点検出と修正

この節は Douglas–Peucker アルゴリズムを用いる頂点検出手法と提案アルゴリズムの検証を行う．Douglas–Peucker アルゴリズムによる輪郭近似法の結果を青い点で示す. 提案アルゴリズムによる修正手法の結果を赤い点で示す．図 4.2 は輪郭近似法で検出成功の結果である．図 4.3 は輪郭近似法で検出失敗の結果とその結果に基づいての修正結果である．

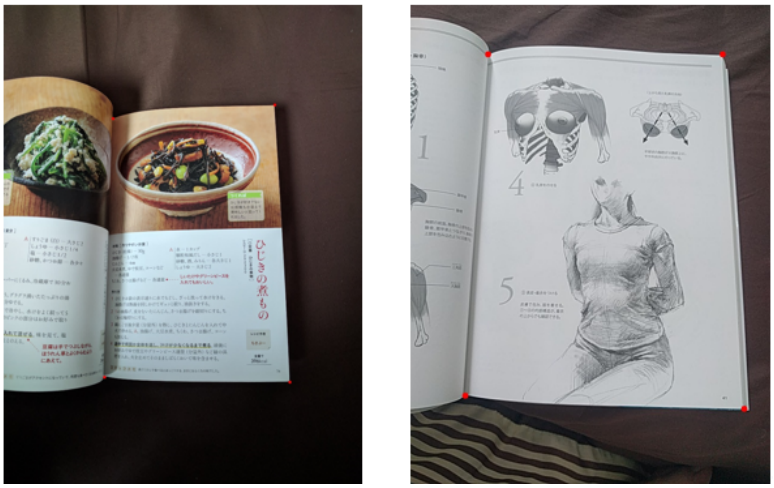

図 4.2　頂点検出成功例

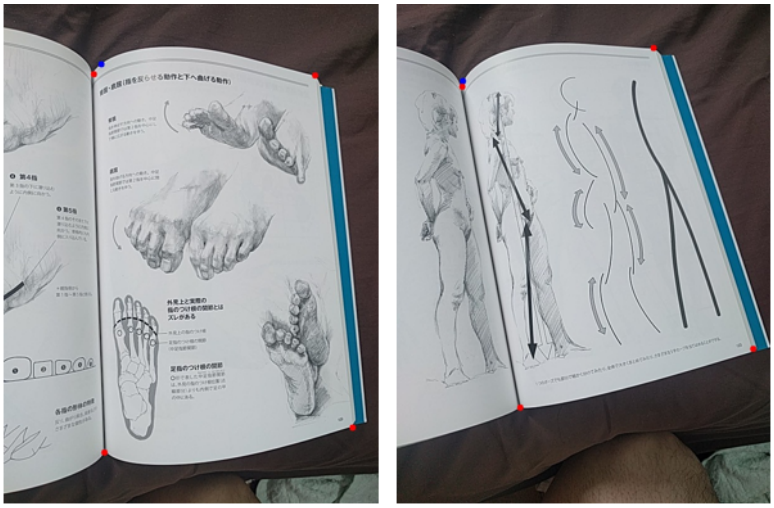

図 4.3　頂点検出失敗と修正結果

図 4.2 では提案修正手法と輪郭近似法同じ結果をできた．このような湾曲程度は大きくない入力画像に対して，輪郭近似法は正しく検出できることが分かった．これに対し，図 4.3 のような湾曲程度が大きい，輪郭が斜めになっている場合，ズレが生じることがあった．検出失敗頂点は主に曲線の曲率が大きいところに現れることが分かった．提案修正手法ではより良い結果ができて，この場合に有効性を示した．

4.3　グリッドの分割

この節ではグリッドの分割による平面処理への影響について検証する．異なる分割の処理時間と平面化処理結果を比較する．図 4.4 は入力画像のサイズ 2400X3200 に対して，150 分割と 600 分割の結果である．図 4.5 は 300X400 入力画像対して，150 分割と 15000 分割の結果である．

入力画像　　15×10　　20×30

図 4.4　分割比較結果 a

入力画像　　15 × 10　　150 × 100

図 4.5　分割比較結果 b

15×10 150×100

図 4.6　直線の結果比較

図 4.4 では 150 分割と 600 分割の結果は大きな差がないが平面化処理の時間は 8.0 秒と 18.2 秒であった．これに対し，湾曲程度が大きい 300X400 入力画像に対して，平面化処理の時間は 1.8 秒と 44.9 秒であった．100 倍の分割数に対して，処理時間が大きく伸びた．図 4.6 は元々は直線の部分を平面化処理後の結果比較である．15000 分割の結果は直線に似ているが，150 分割の方がより滑らかな線形となっていた．

4.4　文字認識

この節では画像内の文書に対して，文字認識 OCR と比較評価を行う．本研究では OCR として Google で公開されている Cloud Vision[30] を使用した．評価の対象について，撮影した入力画像と提案手法での平面化処理後の結果であり，それらに対する性能比較を行う対象とした．

図 4.7 は Cloud Vision を利用し，処理前の入力画像と処理後の画像を OCR によって認識した結果である．水色の枠は認識した文書ブロックであり，緑色の枠は段落に認識した内容である．黄色の線は認識したの文字列である．

Cloud Vision では少し湾曲した文字列が検出できるが，処理前ではタイトルの部分は曲がっているので，認識しづらいところがあった．処理前の文字認識の正確率 73.9 %に対して，処理後の正確率は 82.6 %であり，認識精度が上がることが分かった．

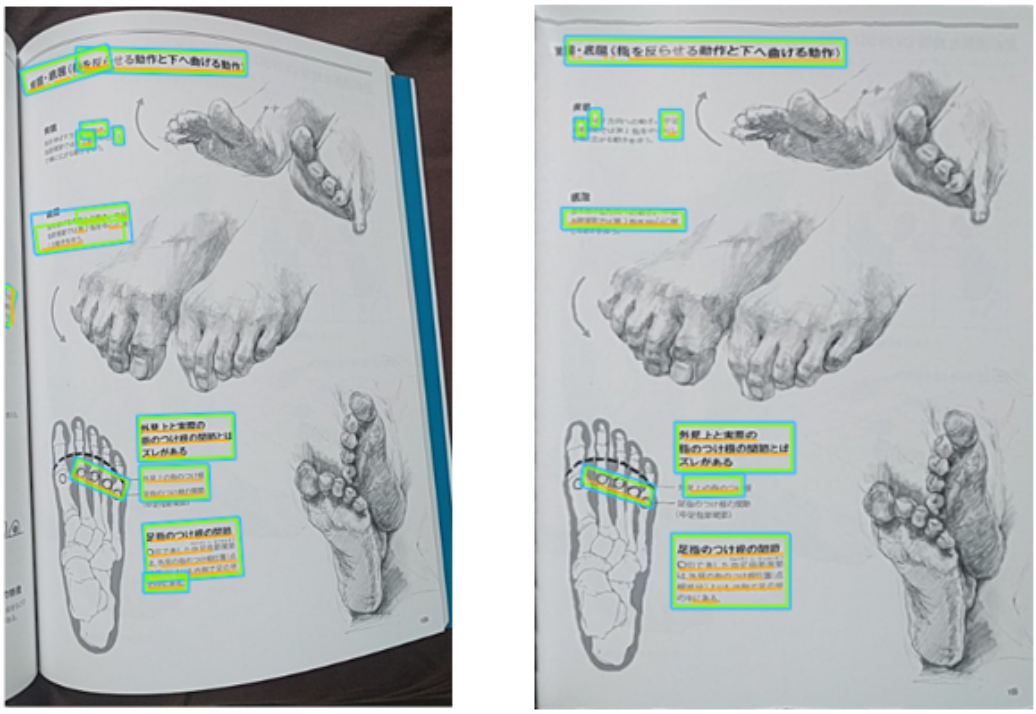

図 4.7　Cloud Vision による OCR 結果

第 5 章

まとめ

本研究では，カメラで撮影した 1 枚の見開き書籍画像に対し，ページ領域の分割が難しいという問題点に対して，機械学習による解決方法に着目した．提案手法では，RGB 書籍画像を入力，手動で作成したページ領域を FCN に学習し，画像中のピクセルごどに前景（右ページと左ページ）か背景かの領域を予測するセグメンテーションを行う．学習の成果として，ページ分割ができるが，輪郭部分の精度が低いことが分かった．より精度高いの輪郭を検出するため，FCN の出力結果に対して画像処理方法を使ってページごどを分離し，ROI 範囲内の画面に対して輪郭検出を行う．検出した輪郭に対して，輪郭にある 4 つの頂点を検出し，輪郭を上下左右 4 つの線に分ける．4 つの線を曲線か直線かを判別する．誤った頂点検出結果を修正することができる．輪郭の曲線部分を利用し，グリッド生成する．書籍ページ領域をグリッドで小さい矩形を分割し，矩形画像に対して，射影変換を行うという手法を提案した．

今後の課題として，FCN で画像を出力結果はぼやけた画像という問題があり，現状の機械学習の手法ではこの問題を根本的に解決していない．解決案として，本研究ではモルフォロジー処理という後処理を利用した．その他，領域分割でよく利用する手法である CRF(Conditional random field) [31] を後処理として，FCN と組み合わせば，より平滑な輪郭を検出することが可能であると考える．ROI 内の輪郭検出では，指と背景の影響で，正しく検出できないという問題がまだ解決していない．指の問題に対して，指を分類クラスに追加すると考える．また，学習データ拡張が必要である．現在のグリッドは，上下の曲線を指定した均等分割することで生成するが，今後は曲線の湾曲程度によって，不均等分割する手法が望まれる．

今後の展望として，深層学習にようる平面処理は本研究のように，2 段階の処理手法ではなく，end-to-end での手法実現を期待する．

謝辞

本研究を締めくくるにあたり，ご指導ならびに適切なご助言を下さいました先生方に感謝の意を表します．また，様々な相談に応じて下さった，研究室のメンバーに深く感謝致します. 特に柿本研の王さん，FCN の実装や貴重な意見を頂きありがとうございます。

参考文献

[1] インプレス. 2017 年度の市場規模は電子書籍、電子雑誌合わせて 2500 億円を突破！ 2022 年には 3500 億円規模に 『電子書籍ビジネス調査報告書 2018』7 月 30 日発売. https://www.impress.co.jp/newsrelease/2018/07/. 参照:2019.2.22.

[2] 国立国会図書館. 資料デジタル化基本計画 2016-2020. https://www.ndl.go.jp/jp/preservation/digitization/. 参照:2019.02.22.

[3] Michael S Brown and W Brent Seales. Document restoration using 3d shape: a general deskewing algorithm for arbitrarily warped documents. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Vol. 2, pp. 367–374. IEEE, 2001.

[4] Li Zhang, Yu Zhang, and Chew Tan. An improved physically-based method for geometric restoration of distorted document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 4, pp. 728–734, 2008.

[5] Gaofeng Meng, Ying Wang, Shenquan Qu, Shiming Xiang, and Chunhong Pan. Active flattening of curved document images via two structured beams. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3890–3897, 2014.

[6] Adrian Ulges, Christoph H. Lampert, and Thomas M. Breuel. Document capture using stereo vision. In *ACM Symposium on Document Engineering*, 2004.

[7] Atsushi Yamashita, Atsushi Kawarago, Toru Kaneko, and Kenjiro T. Miura. Shape reconstruction and image restoration for non-flat surfaces of documents with a stereo vision system. *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, Vol. 1, pp. 482–485 Vol.1, 2004.

[8] Yau-Chat Tsoi and Michael S. Brown. Multi-view document rectification using boundary. *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[9] Hyung Il Koo, Jinho Kim, and Nam Ik Cho. Composition of a dewarped and enhanced document image from two view images. *IEEE Transactions on Image Processing*, Vol. 18, No. 7, pp. 1551–1562, 2009.

[10] Shaodi You, Yasuyuki Matsushita, Sudipta Sinha, Yusuke Bou, and Katsushi Ikeuchi. Multiview rectification of folded documents. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 40, No. 2, pp. 505–511, 2018.

[11] Toshikazu Wada, Hiroyuki Ukida, and Takashi Matsuyama. Shape from shading with interreflections under a proximal light source: Distortion-free copying of an unfolded book. *International Journal of Computer Vision*, Vol. 24, No. 2, pp. 125–135, 1997.

[12] Frédéric Courteille, Alain Crouzil, Jean-Denis Durou, and Pierre Gurdjos. Shape from shading for the digitization of curved documents. *Machine Vision and Applications*, Vol. 18, No. 5, pp. 301–316, 2007.

[13] Li Zhang, Andy M Yip, Michael S Brown, and Chew Lim Tan. A unified framework for document restoration using inpainting and shape-from-shading. *Pattern Recognition*, Vol. 42, No. 11, pp. 2961–2978, 2009.

[14] Huaigu Cao, Xiaoqing Ding, and Changsong Liu. A cylindrical surface model to rectify the bound document image. In *null*, p. 228. IEEE, 2003.

[15] Jian Liang, Daniel DeMenthon, and David Doermann. Geometric rectification of camera-captured document images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 4, pp. 591–605, 2008.

[16] Sagnik Das, Gaurav Mishra, Akshay Sudharshana, and Roy Shilkrot. The common fold: Utilizing the four-fold to dewarp printed documents from a single image. In *Proceedings of the 2017 ACM Symposium on Document Engineering*, pp. 125–128. ACM, 2017.

[17] Fujitsu. スキャナー ScanSnapSV600. http://scansnap.fujitsu.com/jp/product/sv600/. 参照:2019.02.22.

[18] Adobe. Adobe Scan アプリで、文書をスキャンして PDF に変換. https://acrobat.adobe.com/jp/ja/mobile/. 参照:2019.2.22.

[19] Syed Ammar Abbas and Sibt ul Hussain. Recovering homography from camera captured documents using convolutional neural networks. *CoRR*, Vol. abs/1709.03524, , 2017.

[20] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[22] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[23] wkentaro. Image Polygonal Annotation with Python. https://github.com/wkentaro/labelme/. 参照:2019.2.22.

[24] Nikolaos Stamatopoulos, Basilios Gatos, Ioannis Pratikakis, and Stavros J. Perantonis. A two-step dewarping of camera document images. *2008 The Eighth IAPR International Workshop on Document Analysis Systems*, pp. 209–216, 2008.

[25] Satoshi Suzuki, et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, Vol. 30, No. 1, pp. 32–46, 1985.

[26] ディジタル画像処理［改訂新版］編集委員会. ディジタル画像処理［改訂新版］. 画像情報教育振興協会, 2015.

[27] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, Vol. 9, No. 1, pp. 62–66, 1979.

[28] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, Vol. 10, No. 2, pp. 112–122, 1973.

[29] OpenCV team. OpenCV library. `https://opencv.org/`. 参照:2019.02.22.

[30] Google. Google Cloud Vision — Vision API を利用. `https://cloud.google.com/vision/`. 参照:2019.01.24.

[31] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pp. 109–117, 2011.