Vol.2015-CVIM-198 No.22
2015/9/15

四直線推定に基づく情景画像中の文字認識

日並 遼太[1,a)]　劉 新豪[4]　千葉 直樹[3,b)]　佐藤 真一[2,1,c)]

概要：近年、カメラで撮影した情景画像中に含まれる文字の認識が注目を集めている。検出精度を上げるための手法として文字列の並びを表現する直線を推定し、利用する手法が提案されている。しかし、これらは各領域を認識する前の検出結果のみを使ったものであるため正確な推定は困難であり、主に検出結果の確認に用いられる。本稿では、各文字の四直線の位置をデータに基づいて統計的に推定する方法を提案する。文字種ごとに独立に四線の位置を学習させることで、少数の文字からの正確な推定を実現する。また、推定された四直線に基づいて再度、文字を検出および認識する手法を示す。実験により、提案手法の四直線推定に基づく検出・認識により認識精度が上がることを示す。

キーワード：情景画像中文字認識, 文字認識, 単語認識, 四直線推定

1. はじめに

文字認識の研究は従来、文書中の文字を対象とする光学文字認識 (OCR) が中心であったが、近年、周りの風景を撮影した画像に含まれる、看板や標識などの文字の認識に対する需要が高まっている。文字から得られる情報は画像の内容を理解する上で重要であることが多く、これを自動で認識できるようになることで、看板や標識の情報を活用した自動運転や、スマートフォンのカメラを通した外国語の文字の翻訳など、様々なサービスに応用し役立てることができる。しかし、画像中の文字には、照明条件の変化や多様な背景等の様々な問題があり、全ての文字を正確に認識するのは困難である。

画像中の文字認識は２ステップで行われ、まず大まかな文字領域を検出し、その後枠内の文字を認識する。本稿では枠内の英単語を認識するタスクにおいて、その認識率を向上させる手法を提案する。単語認識は従来、枠内の文字を一文字単位で検出し、それぞれの文字を認識を行い、その後誤検出を排除するという一方向のパイプラインで認識を行っていたため、一文字単位で検出・認識しづらい文字は最初の段階で失敗してしまうことが多かった。本手法では、まず一回目の検出・認識で、単語全体で共通するスタイル情報の一つである四直線の推定を行う。そして、この四直線を利用し、再度、より精度の高い検出・認識を行う。

[1] 東京大学, The University of Tokyo
[2] 国立情報学研究所, National Institute of Informatics
[3] 楽天株式会社, Rakuten, Inc
[4] 東京工業大学, Tokyo Institute of Technology
[a)] hinami@nii.ac.jp
[b)] naoki.a.chiba@rakuten.com
[c)] satoh@nii.ac.jp

2. 関連研究

一般的に情景画像中の単語認識は、(1) 画像から一文字単位で文字領域を検出、(2) 各検出候補の文字種の認識、(3) それまでの過程で生じた誤検出や誤認識を修正する単語全体を通しての最適化、という順に処理が行われる。まず文字領域検出は、主に連結領域に基づく手法と sliding window によるものに分類される。連結領域に基づく手法というのは、性質の似た領域を抽出するものであり、二値化した結果の連結領域を検出候補とするようなものである。近年では主に、[4] など、安定した領域を抽出する maximally stable extremal regions(MSER) が用いられている。この手法では、計算量が少なく、誤って文字領域として認識される false positive の数も少ないという利点がある一方で、図 1 の画像など、境界が曖昧な文字や、背景が複雑な文字などは一部の文字の検出に失敗する。

一方で sliding window による手法では、単語の画像全体のあらゆる位置やスケールの領域に対し検出器をかけて検出を行うため、図 1 のような画像の文字も候補として検出することができるが、誤検出が多くなってしまうという問題や、大量の領域に対して分類器をかける必要があるために計算量が大きくなってしまうという問題がある。

これらの文字検出で得られた領域に対して文字分類器を適用し認識を行う。Mishra ら [1] は HOG 特徴量と SVM を用いて文字の分類器を構成し、Wang ら [6] はニューラルネットワークを使ってこれを sliding window を組み合わせて文字の検出・認識を行っている。

最後に、得られた候補領域に対し、誤検出の排除や誤認識の修正など、単語全体を通しての最適化を行う。Mishra

図 1 連結成分に基づく手法で検出に失敗する例. (a) 連結した文字 (b) ぼやけた文字 (c) コントラストの低い文字 (d) 複雑背景

ら [1] は条件付き確率場 (CRF) を用いて、言語の知識も考慮して最適な単語を推論する方法を提案した。また、Novikova ら [5] は、言語の知識に加え、色や文字列のベースラインなどの単語内で一貫している情報を重み付き有限状態トランスデューサー (WFST) の中に組み込んで用いる方法を提案した。このように単語全体の情報を活用して文字の検出・認識を行うことで、更なる単語認識精度の向上が見込める。Jaderberg ら [7] は単語画像全体を入力として認識を行うニューラルネットワークを構築しているが、大量の訓練データが必要となるのが課題である。

3. 四直線に基づく単語認識手法

3.1 手法の概要

本手法では、文字列のスタイルが単語内で一貫していること用いて単語認識を行う。まず最初に、スコアが十分に高く、正しく認識できていると考えられる文字の抽出を行う。本稿ではこれらの文字を”信頼度の高い文字”と呼ぶこととする。その後、信頼度の高い文字を使って文字列のスタイルの推定を行い、これをトップダウンの事前知識として用い、再度文字領域の検出と文字種の認識を実行する。

本手法では、文字列のスタイルとして、タイポグラフィーに基づいたアルファベットに沿う四直線 (図 2) を用いる。これらの線を用いることで、文字のサイズや位置を制限することができ、認識する際の有用な情報として活用することができる。具体的には、四直線により検出領域を絞って初めに検出に失敗した文字を検出し直す再検出と、各文字のスコアを、四直線との位置関係に基づいて変換する再スコアリングを行うことで高精度な検出・認識を行う。

3.2 信頼度の高い文字の検出

スタイル情報としての四直線を抽出するため、信頼度の高い文字の抽出を行う。まず、文字の候補領域を検出するため、連結成分領域の検出手法を用いる。[4] の手法に従い、MSER を用いて検出された領域に加え、画像全体に大津の二値化を適用して得られる連結成分領域の検出候補とする。その後、HOG 特徴量と SVM からなる文字分類器を用いて各候補を識別し、スコアが閾値以上のものを信頼度の高い文字として選択する。

図 2 アルファベットの四直線.

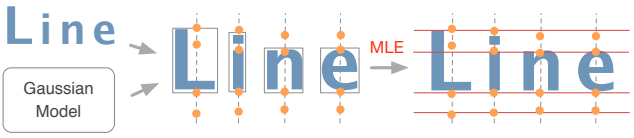

図 3 四直線の推定.

3.3 四直線推定

本節では、信頼度の高い文字から四直線を推定する方法を記述する。タイポグラフィーの用語に基づき、それぞれの線は図 2 のように ascender line、mean line、baseline、descender line と呼ぶ。まず事前に、各文字種ごとに四直線との位置関係とのガウス分布モデルを訓練しておき、このモデルを使って最尤推定を用いて文字の候補領域から四直線の式を求めるということを行う。

ガウス分布モデルの訓練. 四直線と文字との位置関係を表すガウス分布の訓練方法について説明する。まず、訓練データの単語画像に対して四直線の正解位置を求める。これは、各文字が接する線は文字の種類から決定される (A は asender line と baseline に接する等) ので、これにより各文字の矩形領域とその文字種が与えられている訓練画像からは四線を求めることが可能である。

線位置の正解値が得られたので、訓練画像から、各文字との四直線との垂直方向の位置関係を求める事ができる。まず、四直線と文字領域の中心を通る垂直方向の線とが交わる点を、各文字に対する四直線の位置とする。ここで、位置は各文字領域の高さによって正規化された値に変換する。この値を大量の訓練画像について得ることで、各文字クラスに対する線位置をモデル化するのに十分なデータが得られる。本手法では、正規化された線位置がガウス分布に従うとすると仮定し、線位置の平均と分散を 4 線 ×62 文字クラスについて求めてガウス分布のモデルを得る。このとき、 μ_{lc},σ_{lc} を線 l の文字クラス c の平均と分散とする。

四直線の推定手法. 訓練によって得られた線位置のモデルを用いて信頼度の高い文字からの線の推定を行う。図 3 で示すように各文字領域に対して、その文字の位置における線の垂直方向の位置を推定することができるので、これらを使って線式を推定することを行う。

まず最初に、baseline の決定を行う。これは、ほとんどの文字が baseline に接するため、他の線に比べてより正確に推定することができるためである。各領域の位置における baseline は、各領域の文字クラス c の線位置のモデルから得られる μ_{3c} と推定されるので、これらの位置について最小二乗法を用いることで baseline を推定を行う。

次に、その他の三線の推定を行う。l 番目の線は、傾き k、切片 b_l として、$y = kx + b_l$ という式で表される。傾き k は全ての線で、最初に推定した baseline と同じ傾きを用いた。切片は、各線について、傾き k が決定した時に尤度を最大化するような値を求めた。このとき、大文字と小文字が同型の文字については、baseline 以外の推定では除外した。与えられた N 文字の候補領域について、$c_1, c_2, \cdots, c_N$ を各文字の文字クラス、$x_1, x_2, \cdots, x_N$ を中央の水平方向の位置とする。ここで、$p(y_{il}|c_i)$ を、i 番目の文字クラスが c_i であるときの、線 l の x_i における切片が y_i である確率とすると、各線の切片 b_l は以下の式で計算される。

$$b_l = \arg\max_{b_l} \sum_{i=1}^{N} \ln p(y_{il}|c_i) = \arg\max_{b_l} \sum_{i=1}^{N} -\frac{(\tilde{\mu_{lc_i}} - y_{il})^2}{2\tilde{\sigma_{lc_i}}^2} = \frac{\sum_{i=1}^{N} \frac{1}{\tilde{\sigma_{lc_i}}^2} (\tilde{\mu_{lc_i}} - kx_i))}{\sum_{i=1}^{N} \frac{1}{\tilde{\sigma_{lc_i}}^2}}, \quad (1)$$

ここで、$\tilde{\mu_{lc_i}}, \tilde{\sigma_{lc_i}}$ は線位置モデルから得られる文字クラス c_i の領域に対する l 番目の線の位置の平均と分散である。分散 σ_{lc_i} を考慮することによって、線ごと文字種ごとに重みを変化させることができ、例えば図 3 では、ascender line を推定する際には ascender line に接している'L' が最も重視されることになる。

3.4 sliding window を用いた再検出

3.2 節の連結成分に基づく検出手法は図 1 のような画像では失敗する。sliding window に基づく手法では対照的に、連結成分に基づく手法で失敗する領域を検出することができるが、大量の候補領域が生成され、誤検出が多くなってしまう。また、あらゆる位置・スケールの膨大な数の領域に対して分類器をかける必要があるため、計算量が大きくなってしまうという問題がある。

提案手法では、sliding window を用いて一度目に検出に失敗した文字の再検出を行うが、3.3 で推定された四直線を活用することで大幅に分類器をかける窓数を削減する。まず検出領域を、図 4(a) に示すように、四直線の間の信頼度の高い文字の領域を除いた限られた領域に絞り込む。また、図 4(b) のように、窓の縦の位置と高さを四直線の情報を使って三通りに限定し、それぞれについて、窓の中心が線と交わるような状態を保ちつつ線に沿って窓をスライドさせていくことで sliding window による検出を行う。最終的には、この再検出で得られた候補に加え、MSER と大津の二値化による検出で得られた領域も組み合わせて検出候補とする。

3.5 CRF

以上の文字領域の検出と文字認識のステップにより、文字が存在する候補領域とその文字種を得ることができた。

(a) (b)

図 4 Sliding window による再検出. (a) 黄色の領域に絞って再検出を行う. (b) 検出窓の縦方向の位置は 3 種類に限定する.

しかし、この中にはまだ誤検出や誤認識が含まれている事が多い。これらから適切な文字領域を選択し、正しい文字種を認識するために、[1], [2] と同様に CRF を適用した。この CRF のコストを最小化することで、最適な候補の選択を行う。基本的には全て Mishra ら [1] と同様のモデルを用いているが、CRF の unary cost に使う分類器のスコアには、四直線に基づき以下のようにスコアを変換したものを用いている。

四直線に基づく再スコアリング. CRF を適用する前に、推定された四直線に基づいて分類器の出力するスコアの変換を行う。推定された線位置を、各領域がある文字クラスをとるとしたときの線位置のガウス分布モデルに当てはめることで以下のようにスコアの変換を行った。

$$S(x_i = c_j) = p(c_j|x_i) \prod_{l=1}^{4} \exp\left(-\frac{(\tilde{\mu_{lc_j}} - y_{il})^2}{\tilde{\sigma_{lc_j}}^2}\right), \quad (2)$$

ここで、$p(c_j|x_i)$ は、ノード x_i がクラス c_j に分類されるときのスコアである。

このようなスコアの変換処理を行うことで四直線との位置関係を考慮した認識が実現される。例えば、ascender line と baseline の間に挟まれている領域の場合、大文字'C' として分類されるスコアは、'C' のモデルと実際の線位置が適合しているため下がらないのに対し、小文字'c' へのスコアが下がることになり、大文字'C' と認識されやすくなる。これに加え、四直線から離れたところに位置する領域には全ての文字クラスに対して低いスコアが割り当てられることになる。そのため、誤検出された候補を非文字として正しく判定することができるようになる。

4. 実験

4.1 文字分類器と線位置モデルの学習

文字分類器には、特徴量として HOG 特徴量、分類器としては RBF カーネルを使用した多クラスの SVM を用いた。SVM のパラメータは交差検定により求めており、学習における訓練データには、ICDAR 2003、Chars74k ?、IIIT 5k-Word を合わせた、合計で 27715 文字の画像を用いた。また、四直線の位置のモデルの学習には、IIIT 5K-Word の訓練データである 3000 枚の単語画像を用いて学習を行った。

表 1 検出手法を変えたときの文字検出の再現率の比較

Detection method	Recall(%)
Proposed (MSER+Otsu+sliding window)	**81.25**
MSER + Otsu	72.96
MSER	68.24
Otsu	55.29

表 2 検出手法を変えたときの単語認識精度の比較

Method	ICDAR03(FULL)	ICDAR03(50)
Proposed	**84.42**	**87.79**
MSER + Otsu	82.09	86.28
MSER	77.21	83.72
Otsu	74.30	77.91

4.2 文字検出の評価

単語画像からの文字検出の性能評価を行うため、IIIT 5K-Word データセットでの文字検出の recall を測定した。このとき、検出された領域と正解の領域との intersection over union (IoU) が 60%以上のものを検出されたものとして recall の計算を行った。検出手法を変えたときの結果の詳細を表 1 に示す。MSER と大津の二値化による検出では 72.96%であり、それぞれ単体の検出と比べると高く、提案手法による sliding window 再検出を用いることで、8.29%上昇することが確認された。また、画像全体に sliding window をかけた場合とも実験して比較を行ったところ、recall80%に達するのに、従来の sliding window では 1 画像あたり 250 回分類器をかける必要があるのに対し、提案手法では MSER と大津の二値化による候補領域を加えても 40 回程であり、提案手法が高効率で高い recall を達成していることが確認できた。

4.3 単語認識

単語認識実験を行い、提案手法の有効性を検証した。まずは、再検出の単語認識性能への効果を検証するため、検出手法を変えて単語認識の性能評価を行った。評価用のデータセットには ICDAR 2003 [3] を用い、評価方式は [1], [2] と同様に、語彙リストの中から最も編集距離の短い単語を選択するという方式で行った。ICDAR(50) では 50 語、ICDAR(FULL) では全テストデータ 860 語の語彙リストから選択を行う。単語認識精度の比較結果を表 2 に示す。提案手法により、MSER と大津の二値化のみの場合と比べて ICDAR03(FULL) と ICDAR03(50) でそれぞれ 2.21%, 1.28%精度が上昇し、再検出の有効性が検証できた。

次に再スコアリングの単語認識精度への貢献の検証を行った。ここでは、ICDAR03(FULL) での評価に加え、編集距離による修正を行わず大文字と小文字も含め完全に一致する ICDAR(Perfect) についても性能の比較を行った。再スコアリングをした場合としなかった場合での、ICDAR2003 の単語認識精度の比較を表 3 に示す。編集距離による修正がない ICDAR03(FULL) のタスクについては 0.82%のみの増加であったが、ICDAR03(Perfect) では

表 3 再スコアリングの有無による単語認識精度の比較

Method	ICDAR03(FULL)	ICDAR03(Perfect)
With re-scoring	**84.42**	**55.23**
Without re-scoring	83.60	47.09

表 4 関連手法との単語認識精度の比較.

Method	ICDAR03(FULL)	ICDAR03(50)	SVT
Proposed	**84.42**	**87.79**	**76.51**
Shi et al. [2]	79.30	87.44	73.51
Mishra et al. [1]	-	81.78	73.26

8.14%増加しており、大文字と小文字を区別しない文字認識において、本手法の四直線推定に基づく再スコアリングが非常に有効であることがわかる。

最後に、関連手法と単語認識精度の比較を行った。同じ CRF のモデルを使っている [1], [2] との比較を行い、テストデータには ICDAR2003 と SVT を用いて精度の評価を行った。表 4 が認識精度の結果であるが、提案手法は全ての評価方法において他の手法を上回った。これは、(1) 2 つの段階からなる検出手法が少ない誤検出での高い recall を達成していること、また、(2) 四直線を用いた再スコアリングにより、誤検出を減らしつつ文字分類精度を高めたこと、によるものであると考えられる。

5. おわりに

本稿では、単語のスタイル情報の一つである四直線を推定し、文字検出・認識に利用することで、単語認識の性能を向上させる手法を提案した。本手法により、最初の文字認識結果から推定された四直線を利用し、検出に失敗した文字領域の再検出、および誤認識された文字の修正が可能となった。標準的な複数のデータセットを用いた実験により、本手法が他の手法と比べて優れていることを示した。

参考文献

[1] Mishra, A., Alahari, K. and Jawahar, C. V.: Top-down and bottom-up cues for scene text recognition, CVPR (2012)
[2] Shi, C., Wang, C., Xiao, B., Zhang, Y., Gao, S. and Zhang, Z.: Scene text recognition using part-based tree-structured character detection, CVPR (2013).
[3] Lucas, S., Panaretos, A., Sosa, L., Tang, A., Wong, S. and Young, R.: ICDAR 2003 robust reading competitions, ICDAR (2003).
[4] Neumann, L. and Matas, J.: Text localization in real-world images us- ing efficiently pruned exhaustive search, ICDAR (2011).
[5] Novikova, T., Barinova, O., Kohli, P. and Lempitsky, V.: Large-lexicon attribute-consistent text recognition in natural images, ECCV (2012).
[6] Wang, T., Wu, D. J. and Ng, A. Y.: End-to-end text recognition with convolutional neural networks, ICPR (2012).
[7] Jaderberg, M., Simonyan, K., Vedaldi, A. and Zisserman, A.: Reading Text in the Wild with Convolutional Neural Networks, NIPS Deep Learning Workshop (2014).