Vol.2021-IFAT-142 No.8
Vol.2021-DC-120 No.8
2021/3/26

特許検索のためのフローチャート画像の解析

樊エイブン[1]　橋本勇太郎[1]　難波英嗣[1]

概要：特許中の図表は、特許の内容を理解する上で非常に重要な役割を果たしている。我々は、画像を解析することで、より高度な特許検索の実現を目指しており、その第一歩として、本研究では、フローチャート画像の解析に取り組む。本研究では、画像認識および文字認識技術を用いて特許中のフローチャートの構造を解析する手法を提案した。提案手法の有効性を確かめるため実験を行った結果、フローチャート中のノードを、精度 0.753、再現率 0.673 で抽出することができた。

キーワード：フローチャート，OpenCV，特許

Analysis of Flowchart Images for Patent Retrieval

REIWEN FAN[1]　YUTARO HASHIMOTO[1]
HIDETSUGU NANBA[1]

Abstract: The charts in patents play a very important role in understanding the contents of patents. We aim to realize more advanced patent retrieval by analyzing the charts, and as a first step, we work on the analysis of flowchart images. In this paper, we propose a method to analyze the structure of flowcharts in patents using image recognition and character recognition techniques. As a result of an experiment to confirm the effectiveness of our method, we obtained precision of 0.753 and recall of 0.673.

Keywords: flowchart, OpenCV, patent

1. はじめに

特許中の図表は、特許の内容を理解する上で非常に重要な役割を果たしている。例えば、非常に複雑な構造をした装置や複雑な手順は、自然言語で説明するよりも装置の構造図やフローチャートとして図示した方がはるかに分かりやすい。実際に、図表が特許の中でどの程度使われているのか調べたところ、2019 年には 306,223 件の特許が公開されており、これらの中に 6,568,824 画像が使われていることが分かった。これは 1 特許あたり平均 21 件の図表画像が使われていることになる。

従来の特許検索では、特許中のテキストの他、国際特許分類、F タームなどの各種メタデータが利用されてきた。これらに加え、我々は、画像を解析することで、より高度な特許検索の実現を目指している。その第一歩として、本研究では、フローチャート画像の解析に取り組む。フローチャート画像を解析できれば、将来的には、ある手順と類似する手順の特許を検索するといった、従来の特許検索では困難な検索を実現することが可能になると考えられる。

本論文の貢献は以下のとおりである。

- 55 言語に対応したフローチャート画像の解析を実現
- 再現率 0.673、精度 0.753 でフローチャート中のオブジェクトを抽出
- 1 画像あたり約 15 秒で解析可能

2. 関連研究

論文や特許などの図表画像を解析する研究がこれまでにいくつか行われている。Shindo らは、論文から図表画像を抽出し、折れ線グラフを解析することで数値情報を抽出している[5]。Oka らは、論文中の表画像からポリマーの物性情報を抽出している[6]。

フローチャート画像の解析における関連研究プロジェクトとして CLEF-IP がある[3]。CLEF(Conference and Labs of the Evaluation Forum)とは、ヨーロッパを中心に行われている情報検索に関するワークショップであり、CLEF-IP は特許を対象としたタスクのことを指す。このタスクは実験レベルだけではなく、現実の課題に即した検索タスクのためのデータセットを提供することで、多言語及びマルチモーダル特許検索タスクの研究の促進を図っている。CLEF-IP では図形を認識し、フローチャートの要素となるテキスト、エッジ、ノードを検出しフローチャートの認識を行っている。CLEF-IP の基本的な課題は、本研究と共通するが、CLEF-IP が実施された 2013 年当時と比べ、画像解析技術が大幅に向上している点、また、本研究では、画像データだけでなく、画像に付随するキャプションや本文のデータがセットになっているという点が異なる。

フローチャート画像を認識するこの他の研究として、Herrera-Cámara のものがある[1]。この研究では、手書きフ

1 中央大学
Chuo University

Vol.2021-IFAT-142 No.8
Vol.2021-DC-120 No.8
2021/3/26

ローチャート画像を解析し、C 言語のソースとして出力する手法を提案している。Sethi らは、深層学習関連の論文中の図表画像からフローチャートを識別し、さらにフローチャートを解析することで、Keras と Caffe でソースを出力するシステムを構築している[4]。本研究では、フローチャートを解析する際、これらの手法を参考にしている。本研究とこれらの先行研究と異なる点は、次節で述べる解析手法の中で、抽出されたオブジェクトの面積でクラスタリングすることで、再現率を低下させることなく精度を向上させた点である。

3. フローチャート画像の解析

3.1 解析手順

フローチャート画像の解析は、以下の 6 つの手順から構成される。

① 輪郭の抽出
② 輪郭の近似・座標情報の抽出
③ 形状の認識
④ 面積によるクラスタリング
⑤ 文字認識
⑥ フローチャートと本文の対応付け

以下、各手順について述べる。

① 輪郭の抽出

入力画像をグレースケール画像に変換した後、コンピュータビジョン向けライブラリ OpenCV(https://opencv.org)の輪郭検出メソッド findContours を用い、フローチャート内のオブジェクトを検出する。

② 輪郭の近似・座標情報の抽出

手順 1 で検出された輪郭は曲線などの多数の点で構成されるが、これを OpenCV の approxPolyDP メソッドを用いてより少ない点で近似し、座標情報を得る。その理由は、次のステップで、オブジェクトの形状を認識するためである。

③ 形状の認識

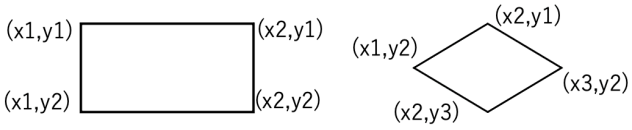
手順 2 で得られた座標情報を用い、オブジェクトの形状を認識する。形状認識には以下のルールを用いる。

```
if 頂点の数==3：三角形
else if 頂点の数==4：
    if 4 つの頂点のうち、x 座標と y 座標が 2 種類ずつある：長方形
    else if 4 つの頂点のうち、x 座標と y 座標が 3 種類ずつある：ひし形
else if 頂点の数==5：五角形
else if 頂点の数 6 以上 14 未満：楕円形
else if 頂点の数<=14：円
```

図 1：オブジェクトの形状認識手順

Figure 1 Procedure of an object-type identification

頂点の数が 4 の時に長方形とひし形を区別する方法について、図 2 を用いて説明する。図 2 から、(a)長方形の場合 x 座標は x1 と x2、y 座標は y1 と y2 の 2 種類ずつある。一方で、(b)ひし形の場合、x 座標は x1、x2、x3、y 座標は y1、y2、y3 の 3 種類ずつあることがわかる。これらの結果から、x 座標と y 座標が何種類あるかを調べれば、長方形とひし形を区別できる。



(a) 長方形　　(b) ひし形

図 2：長方形とひし形の区別

Figure 2 Distinguishing between rectangles and rhombuses

④ 面積によるクラスタリング

手順 3 で検出されたオブジェクトのうち、不要なものを除外する。ここで、不要なオブジェクトはどのようなものか、図 4 を用いて説明する。図 3 は、2 つのオブジェクトから構成されるフローチャートであるが、手順 3 の結果、(b)のように検出すべき 2 つのオブジェクトの他に、2 つのオブジェクトをまとめた(a)のようなものもオブジェクトとして検出されてしまう。

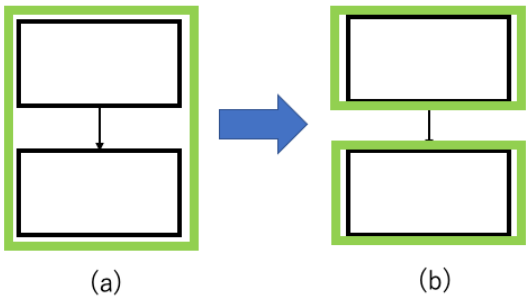


図 3：オブジェクト認識におけるクラスタリングの利用

Figure 3 Utilization of clustering in object recognition

そこで、OpenCV の contourArea メソッドを用いて各オブジェクトの面積を得た後、DBSCAN[2]を用いて面積が±90 の範囲のオブジェクトを同一クラスタに統合する。クラスタリングの結果、一番多くのオブジェクトを含むクラスタを選択する。なお、±90 という値は経験的に決めたものである。また、極端に大きい(面積が 100000 以上)あるいは極端に小さい(面積が 1000 未満)オブジェクトは事前に削除している。

⑤ 文字認識

Tesseract(https://github.com/tesseract-ocr)を用いて、各オブジェクトの内部にある文字の認識を行う。Tesseract はオープンソースの OCR エンジンである。画像全体だけでなく、座標を指定すると、画像の一部だけ文字認識することができるため、フローチャートのノードを認識した後に、ノード内の文字列だけ文字認識することができる。さらに、認識時に Tesseract で使用する言語モデルを指定すれば、55 言語で文字認識をすることができるため、本研究で開発する

Vol.2021-IFAT-142 No.8
Vol.2021-DC-120 No.8
2021/3/26

システムは、日本国特許だけでなく、米国特許や中国特許にも容易に適用することができる。なお、Tesseract で画像の一部を文字認識する場合、矩形領域の座標が必要となるため、各オブジェクトの矩形領域の座標は、手順 2 とは別に boundingRect メソッドを用いて得る。

⑥ フローチャートと本文の対応付け

画像と本文との対応付けを行う。日本国特許では、本文は XML 形式であり、テキストには段落タグが付与されている。各画像には ID 番号が付与されており、本文中の言及個所との対応付けも容易に行うことができる。

以上の 6 手順を実行した場合、1 画像あたり約 15 秒で解析できることを確認した。なお、本来ならばフローチャートのノードだけでなくノードとノードをつなぐエッジについても解析すべきであるが、エッジ解析については今後の課題としたい。

3.2 システムの動作例

本研究のシステムの動作例を図 4 と 5 に示す。図 4 では、形状認識の際に形状ごとに色分けをしており、長方形は緑、楕円形は青、ひし形は赤で表している。解析結果からわかるとおり、矩形領域、条件分岐を示すひし形、フローの最初と最後を示す楕円が正しく解析されている。一方で、条件分岐が連続する個所の右側の領域など、抽出すべきでない個所を誤って抽出してしまっている。

図 5 は、図 4 の解析結果として得られたノードについて、ノードの ID(ID)、クラスター番号(CLUSTER)、面積(AREA)、座標情報(COORDINATE)、形状の種類(POLY)、ノード内の文字(TXT)の出力結果を表している。また、この図に対して図 6 のテキストが付随している。画像を解釈するために、本文中の言及個所が利用できる点が CLEF-IP と本研究との違いである。

図19

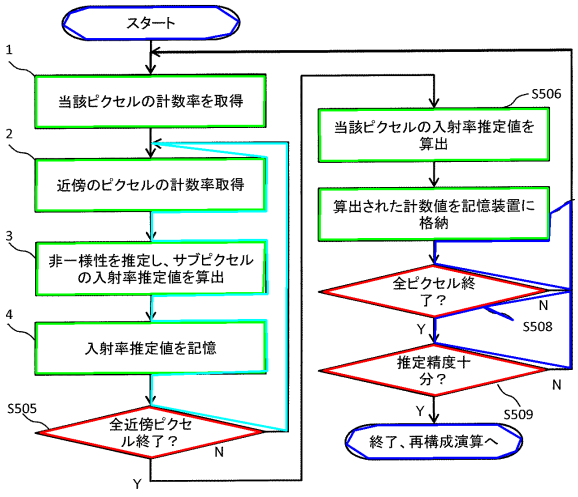


図 4：ノード抽出の結果

Figure 4 Result of node extraction

```
<ID>2</ID>
<CLUSTER>0</CLUSTER>
<AREA>27474.0</AREA>
<COORDINATE>[708, 1017, 493, 113]</CORDINATE>
<POLY>diamond</POLY>
<TXT>推定 精度 十<br><br>一一 分? 二一</TXT>
</NODE>
<NODE>
<ID>3</ID>
<CLUSTER>0</CLUSTER>
<AREA>56831.0</AREA>
<COORDINATE>[67, 966, 510, 116]</CORDINATE>
<POLY>rectangle</POLY>
<TXT>入射 率 推定 値 を 記憶</TXT>
</NODE>
<NODE>
```

図 5：フローチャート中のオブジェクトの形状認識および Tesseract を用いた文字認識結果

Figure 5 Result of object-type identification and character recognition by Tesseract

発明の名称:「放射線撮像装置、放射線撮像方法及び放射線撮像プログラム」
キャプション:本発明の第４の実施形態に係るＸ線ＣＴ装置における非一様性の推定～入射率推定値算出の流れを示すフローチャートである。
本文:この場合の処理は、例えば、図１９に示すフローチャートに従って、ピクセル毎の非一様性推定から入射率推定値算出までの処理を行う。

図 6：図 4 のフローチャート画像に関するテキスト

Figure 6 Texts about the flowchart image in Figure 4

4. 実験

3 節で述べた手法の有効性を確認するため実験を行った。

実験データ

日本国特許公開公報 2018 年から抽出した 37 画像(フローチャート)を用いる。これらの画像に対し、人手で 502 ノードの座標を判定した。なお、各座標はオブジェクトの形状に関係なく、すべて矩形領域の座標として近似的に判定している。

評価方法

出力結果と比較し性能を評価する。ノードの座標の一致度の評価尺度として精度と再現率を用いる。なお、システムの出力と人手のノードを比較する際、領域の重なる面積が 90%を超えた時、2 つのノードが一致したと判定する。また、比較手法としてクラスタリングをしない場合の精度、再現率を求める。

比較手法

3.1 節で述べた 6 つの手順のうち、手順 4 においてクラスタリングを行った場合(提案手法)とクラスタリングを行わず、すべてのノードを抽出した場合(ベースライン手法)とで比較した。

実験結果

実験結果を表 1 に示す。表 1 より、提案手法と比較手法

Vol.2021-IFAT-142 No.8
Vol.2021-DC-120 No.8
2021/3/26

を比べると再現率を低下させることなく高い精度を得ることができたため、クラスタリングは有効であり、提案手法のほうが優れているという結果が得られた。

表 1　オブジェクトの抽出結果

Table 1　Experimental result of object extraction.

	精度	再現率
(1) 提案手法	0.753	0.673
(2) ベースライン手法	0.693	0.673

考察

解析に失敗した例をいくつか示す。図 7 は、オブジェクトが全く抽出できなかった画像の一部を拡大したものである。おそらく画像をスキャンした際に混入したノイズではないかと思われるが、オブジェクトの輪郭線が不鮮明であることがわかる。このような画像に対しては、前処理としてノイズ除去あるいはエッジを強調するようなフィルタを適用する必要があると考えられる。

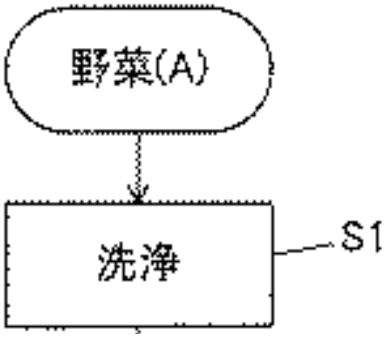


図 7：オブジェクトが抽出できなかった画像の例

Figure 7　Example of an image from which no objects could be extracted

図 8 は、フローチャートの中に、条件分岐などで閉ループが存在する場合に発生する誤り例である。同様のものは図 5 にも数カ所見られる。このような誤りについては、オブジェクトの輪郭座標である程度検出できるのではないかと思われる。3.1 節の手順 2 において、輪郭座標を近似した場合、座標は、順にオブジェクトの周りに沿ってリストとして得られる。この時、x 座標または y 座標の増加あるいは減少が一定回数以上ある複雑な形状のものはエラーとして抽出しないという方法が考えられる。

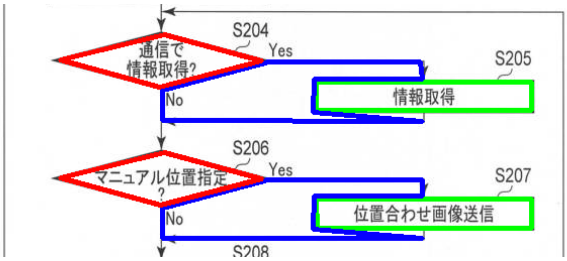


図 8：オブジェクトが誤って抽出された画像の例

Figure 8　Example of an image with incorrectly extracted objects

この他、抽出されたオブジェクト内の文字列を Tesseract で文字認識し、出力された文字列が不自然な日本語である場合、そのオブジェクトを抽出しないという方法も考えられる。3.1 節手順 5 で述べたとおり、Tesseract で画像の一部を文字認識する場合、矩形領域の座標が必要となるが、エラーオブジェクトの場合、他のオブジェクト内の文字列を断片的に切り取ってしまうことがあるため、この場合文字認識すると、オブジェクト内の文字列は不自然な日本語となってしまう。そこで、言語モデル等を用いて認識された文字列の自然さを測ることにより、エラーオブジェクトかそうでないかの判定がある程度可能になると考えられる。

5. おわりに

本研究では、特許中のフローチャートの構造を解析するシステムを構築した。本システムは、まず、物体認識技術を用いてフローチャートの輪郭を抽出、輪郭の近似、座標情報の抽出を行う。次に、抽出されたノードの座標情報に条件を定め、形状を分類し、ノードの形状認識を行う。そして、Tesseract を用いて抽出されたノード内の文字の認識を行う。日本国特許公開公報 2018 年から選択した 37 件のフローチャート画像を用いて評価を行った。実験の結果、フローチャート中のノードを、精度は約 0.753、再現率は約 0.673 で抽出することができた。

6. 今後の課題

オブジェクト抽出の再現率と精度を向上させるための手法については、すでに 4 節の考察で述べたが、この他にもいくつか取り組むべき課題が残されている。例えば、3.1 節で述べたノードとノードをつなぐエッジの検出はそのひとつである。

本研究では、フローチャート画像を入力の前提としているが、本来ならば、特許中の各画像がフローチャートかどうかを事前に判定する必要がある。現在、画像の種別を判定するためのデータセットを構築中であり、構築が終わり次第、画像種別の判定にも取り組む予定である。

フローチャートの解析結果と本文中の言及個所との詳細な対応関係についても検討する必要があると考えている。実際にいくつかの事例を見たところ、例えば、フローチャート中の条件分岐の個所に対応する本文中のテキストに「ならば」という表現が見られた。おそらくフローチャートの形状と本文には何らかの関係があると考えられるが、フローチャートの解析精度をさらに向上できれば、フローチャートの形状とテキストの記述に関する詳細な分析に取り組みたい。

謝辞　本研究は JSPS 科研費 20H042101 の助成を受けたものである。

参考文献

[1] Herrera-Cámara J.I.. FLOW2CODE - From Hand-drawn Flowchart to Code Execution, Master Thesis, Texas A&M University, 2017.
[2] Ester, M., Kriegel, H.P., Sander, J., and Xu, X.. A density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, KDD-96 Proceedings, 1996, p.226-231.
[3] Piroi, F., Lupu, M. and Hanbury, A.. Overview of CLEF-IP 2013 Lab Information Retrieval in the Patent Domain, Information Access Evaluation. Multilinguality, Multimodality, and Visualization. CLEF 2013. Lecture Notes in Computer Science, vol. 8138. Springer, Berlin, Heidelberg, 2013.
[4] Sethi, A., Sankaran, A., Panwar, N., Khare, S., and Mani, S.. DLPaper2Code: Auto-generation of Code from Deep Learning Research Papers, Proceedings of the 32th AAAI Conference on Artificial Intelligence, 2018.
[5] Shindo, H. and Matsumoto, Y.. Automatic Reading of Tables and Figures in Scientific Papers, CBI Annual Meeting, 2019.
[6] Oka, H., Shindo, H., Goto, K., Matsumoto, Y., Yoshizawa, A., Kuwajima, I., and Ishii, M.. Automatic Extraction of Polymer Data from Tables in Xml, Proceedings of the 3rd International Workshop on Scientific Document Analysis (SCIDOCA 2018), 2018.