

# 画像認識を用いたレポート自動評価システムの研究 手描き UML ダイアグラムを題材として

2016SE092 山田茉実 2016SE094 山崎紗弥

指導教員：沢田篤史

## 1 はじめに

近年，教育現場において，採点者の採点に要する時間や労力の軽減，採点の揺れや誤りを防ぐために手書き文字の自動採点の実用化が進められているが，手描き図形の自動採点は十分に実用化されていない．一方で，図形認識技術の研究が進められていることから，手描き図形処理技術の組み合わせにより手描き図形の自動採点の実現が期待できる．

採点を自動化するソフトウェアを実現するには大学によって異なる評価方法に対応することや，様々な科目を対象とすることを考慮すべきである．採点者は一般的に手描き図式の採点をする際に図式の特徴を認識し複雑さから評価をおこなうので，自動採点するには採点者の判断を補助する合理的な採点基準を提供しなくてはならない．

本研究の目的は，手描き図式を自動採点するシステムのためのソフトウェアアーキテクチャの設計である．自動採点における様々な変更に対応できるソフトウェアを効率的に開発するためには，柔軟性を担保できるアーキテクチャの構築が必要である．それに基づいてシステムにプロトタイプを作成する．本研究では採点対象の図式として UML ダイアグラムのクラス図を取り上げる．描き方の癖による採点の揺れや誤りを抑制できる判断基準と方法を提案することや，手描き図式の採点に必要な合理的な採点基準を提供することも目的とする．

アーキテクチャの設計にあたり，採点に必要な処理を分類処理，抽出処理，評価処理の三つに分割する．先に手描き図式を分類することで異なる種類の図式を誤った方法で評価する事を避け，評価方法も採点対象の図式によって変更が容易になる．分類処理には，画像や図形の認識処理で広く採用されている CNN を用いる．抽出処理では，評価するために必要な要素を抽出する方法として，塗りつぶしやマッピングといった画像処理技術を用いる．これらの技術によって隣接した矩形や線分を分離して認識することができる．評価処理には，評価基準として定量的な値を与える必要があるため，オブジェクト指向設計の構造上の複雑度を計る CK メトリクス [4] を採用する．本研究ではシステムの基本構造に，大量の画像データを効率よく扱うことができる Blackboard パターン [2] と，分割した処理を個々に実行するだけでなく処理を目的に応じて入れ替えたり，組み合わせて実行することを可能にする Command パターン [1] を用いる．

## 2 自動採点システムに関する課題

教育現場では試験やレポート課題を出題すると採点者は大量の答案データを採点しなければならない．例えば南山大学ソフトウェア工学科ではおよそ 200 名の学生が同じ講義を受ける場合がある．そこで試験を実施すると採点者は 200 もの答案を採点する必要がある．この作業を自動化すれば採点者の負担が減るが，手描き図式を自動採点するにはまず図式の特徴を認識し複雑さから評価をおこなう．様々な図式を採点対象とするには，それらの図式を認識できなければならない．例えば，オブジェクト指向モデリングで用いられるクラス図とコミュニケーション図のように似た図式を同じ評価方法で間違って評価することを避けるためにはじめに分類しておく必要がある．

図式の評価方法もいくつか存在する．例えば，オブジェクト指向複雑度メトリクスは CK メトリクス [4] や Briand メトリクス [3] がある．このように様々な評価方法に対応することを考慮して自動評価システムを設計する必要がある．

手描き図式の描き方には個人差があるので均一な方法で自動採点をおこなうのは難しい．人は一般的に手描き図式の採点をする際にその特徴を認識し複雑さから評価をおこなう．これをコンピュータで採点を自動化するには，図式の複雑さについて採点者の判断を補助する合理的な採点基準を提供しなくてはならない．

## 3 手描き図式自動評価システムの設計

### 3.1 システムの設計方針

システムを設計するにあたり，本研究では採点に必要な処理を分類処理，抽出処理，評価処理の三つに分割する．採点者による採点では，まず答案の図の種類が正しいかどうか判断し，正しかった図式の細かい部分をみて採点をおこなう．処理を分割し，採点者の採点過程に近い設計をし，それぞれの処理の精度を高め，採点処理全体の精度を高めることを狙う．また，採点者が各処理の結果を確認し，必要に応じて修正することを可能にする．

図式の分類処理には CNN を用いて，書く人によって個人差のある手書き図式の分類を可能にする．また，評価を自動でおこなうためには，評価基準に決定的な値を与える必要がある．その基準に CK メトリクス [4] を用いる．CK メトリクスとは，オブジェクト指向設計の構造上の複雑度を計るメトリクスである．抽出処理で塗りつぶしやマッピングなどの画像処理技術を用いて矩形と線分を抽出し，メトリクス値を算出する．評価処理でその値から図式

の複雑さを計算し、その結果を評価結果とする。

本研究のシステムの基本構造に POSA パターン [2] の Blackboard パターンと GoF デザインパターン [1] の Command パターンを用いた。採点対象の画像データを変更することなく分割した各処理をおこない、採点する大量の画像データを効率よく扱うことを目的に Blackboard パターンを用いる。また Command パターンを用いることで、分割した処理を複数組み合わせることが出来る。

### 3.2 システムのユースケース

手描き図式自動採点システムのユースケース図を図 1 に示す。

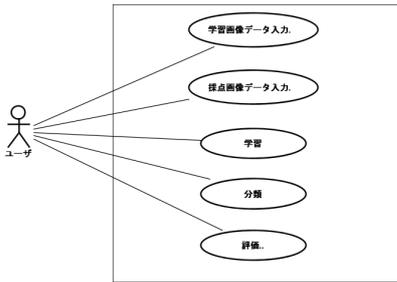


図 1 ユースケース図

システムのユースケースは、“学習画像データ入力”、“採点画像データ入力”、“学習”、“分類”、“評価”である。“学習画像データ入力”では、ユーザが学習させたい画像データとその画像の分類ラベル、評価ラベルを Excel ファイルに書き込む。“採点画像データ入力”では、ユーザが採点させたい画像データを Excel ファイルに書き込む。“学習”では、ユーザから学習指示があった際に、学習器を呼び出し入力されている画像の学習をおこなう。“分類”では、ユーザから分類指示があった際に、分類器を呼び出し採点画像を図式の種類で分類をおこなう。“評価”では、ユーザから評価指示があった際に、評価器を呼び出し採点画像の図式の評価をおこなう。

### 3.3 アーキテクチャ

Blackboard パターンと Command パターンを用いた全体アーキテクチャの静的構造を図 2 に示し、各コンポーネントの説明を述べる。

Blackboard は、Blackboard パターンの黑板コンポーネントにあたる。各処理で生成されるデータとユーザから得られるデータを各処理と独立の関係で保管させる。データの中身として Diagram をもつ。学習画像データ入力、採点画像データ入力、学習器、分類器、評価器は、Command パターンの具象コマンド、Blackboard パターンの Knowledge source にあたる。必要なデータを読み込むこと、各処理で生成されるデータを Blackboard に書き込む。command は、Command パターンの Command にあたる。ユーザから指示を得て、具象コマンドの必要な処理を呼び出す。invoker は、Blackboard パターンの

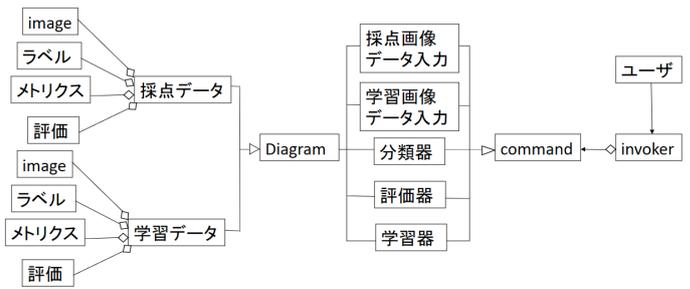


図 2 手描き図式自動評価システムの静的構造

control, Command パターンの invoker にあたる。ユーザの要求に従って、command に指示をする。ユーザは、本研究で設計するシステムの利用者である。Diagram は、Blackboard 内のデータである。採点データは採点する画像データ。画像 image, 分類ラベル, メトリクス, 評価ラベルをもち、学習データは、学習させる画像データ。画像 image, 分類ラベル, メトリクス, 評価ラベルをもち。

### 3.4 自動評価機能の設計

自動採点システムの採点に必要な処理として分割した分類処理、抽出処理、評価処理について詳細な設計を示す。

#### 3.4.1 分類処理

分類処理では、手描き図式の画像から図の種類を識別し画像データを種類ごとに分類する。手描きの図式は、同じ種類の図式であっても、多種多様なバリエーションを含んでいる。そのバリエーションに対応して分類をおこなうために、畳み込みニューラルネットワーク (Convolutional Neural Network) を用いる。CNN は、畳み込み層、プーリング層、全結合層から構成されるニューラルネットワークである。画像データは各ピクセル値同士の位置情報が重要になるが、ほかのニューラルネットワークでは入力データを 1 次元化し学習をおこなうので、その場合数値同士の位置情報が失われてしまう。それを解決するのが CNN であり、画像内の手描き図式の位置情報を失うことなく学習させることができる。

本研究で設計した CNN モデルを図 3 に示す。

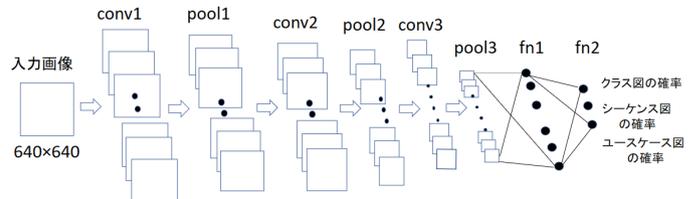


図 3 cnn モデル図

層の数は、中間層 3 層 (図 3 中の conv1, conv2, conv3), 全結合層 2 層 (図 3 中の fn1, fn2) とした。conv1 で線要素を抽出、conv2 で矩形要素を抽出、conv3 で要素同士

の繋がりを抽出する。また、畳み込み (conv) のあとにそれぞれ Maxpooling(図 3 中の pool1, pool2, pool3) をおこない、データの扱いをしやすくし特徴を際立たせる。fn1 では 1 次元配列に変換し、fn2 ではクラス数である 3 クラスに分類する。fn2 の出力に softmax 関数を用い、そのクラスに分類される確率を出力する。

入力画像サイズ 640 × 640 の白黒二値画像とする。一般的に答案用紙をスキャンして得られるサイズのまま CNN への入力とする学習に必要な時間やメモリ量が大きくなりすぎるのが懸念されるので、640 × 640 に縮小する。

畳み込み層のフィルタサイズは、4 × 4, 8 × 8, 10 × 10 の 3 パターンとする。畳み込みを繰り返した際の画像サイズに対応したフィルタサイズにするために、この 3 パターンのフィルタサイズで試行し、精度の良いものを採用する。また、スライドを 1 × 1, 2 × 2 の 2 パターンとする。入力画像のサイズが大きいため、1 × 1 と 2 × 2 の両方で試行し、精度と実行効率を考慮して決定する。

pooling 層のフィルタを 2 × 2 の Maxpooling とする。pooling のフィルタサイズを大きくしすぎると、粗い特徴になってしまうことと、画像サイズが小さくなりすぎてしまうので 2 × 2 を設定する。

最適化手法は Adam を用いる。Adam は、小さな極小値に陥ることなく最小値を目指す計算手法である。さらにパラメータ毎に最近の勾配の値に比べて、大きかったのか、小さかったのかを考慮したパラメータ更新をおこなうことで、パラメータ毎に適切な更新ができる。

バッチサイズは一般的に 2 の n 乗とされ、データ数の大きさに合わせ 64 とする。エポック数は学習するにあたって、損失関数が収束される 100 にする。

### 3.4.2 抽出処理

画像処理技術は画像を利用したり、画像から必要なデータを得たりするために、コンピュータを用いて画像を処理する技術である。本研究では、塗りつぶしやマッピングの処理をおこなう手法を用いて隣接した矩形線分を分離して認識した。

以下に示す (a) から (h) の処理を順におこなう。

はじめに矩形の抽出をおこなう。(a) 入力データを読み込み、次の処理で背景を塗りつぶすために二値化して白黒を反転する。(b) 背景の塗りつぶしをおこない、閉鎖された図形のみ画像内に残す。(c) (b) で行われた閉鎖された図形のみ残した画像の白黒を反転する。(d) (a) で得られた画像と (c) で得られた画像を結合することで、矩形内の文字を削除する。(e) 膨張処理を繰り返し、線分を削除する。(f) 指定した範囲の大きさのオブジェクトの輪郭を検出する。検出されたオブジェクトの個数と外接矩形の x 座標, y 座標を取得して、Excel ファイルに出力する。

次に線分の抽出をおこなう。(g) 検出した外接矩形の範囲を入力データから削除して、線分の抽出をおこなう。(h) 矩形抽出と同様に二値化する。(i) 指定した範囲の大きさ

のオブジェクトの輪郭を検出する。検出された線分をセグメントし、個数と位置要素を取得して Excel ファイルに出力する。

### 3.4.3 評価処理

採点者はそれぞれ判断基準に基づいて採点をおこなっているが、評価システムでは合理的な数値基準を用いて判断する必要がある。本研究では、図式の複雑度を定量的な基準値で計るために CK メトリクス [4] を用いる。CK メトリクスとは、オブジェクト指向設計の構造上の複雑度を計るメトリクスである。CK メトリクスには、次の六つの測定量がある。その項目のなかで、NOC の計測対象クラスのサブクラス数と CBO の計測対象クラスに関係しているクラス数の二つの項目からクラス図の複雑度はクラス図内のクラスの数と線分の数が大きく関係している。それらの特徴を抽出することでクラス図の評価に関する合理的な基準を提供できる。

NOC と CBO の計算方法は次のとおりである。NOC は 3.4.2 節の図から抽出した矩形の数である。CBO は最大値とする。最大値の算出方法は (1) セグメントした矩形ごとにその外接矩形の中に線分の四隅の x 座標, y 座標が重なっているか調べる。(2) 重なっている場合、その線分の外接矩形の対角に矩形が重なっているならばサブクラスとして数える。全ての矩形のサブクラスを計測し、その最大値を算出する。

## 4 実装と実データによる評価

### 4.1 実装

3 章で述べた設計に従って、実装をおこなう。python を実装プログラムの言語とし、python の実行に適している開発環境である Anaconda を用いる。python には多くのライブラリが用意されており、高度な数値計算やデータ分析、機械学習などが用意にできるようになっている。各処理に必要なライブラリを用いて、実装をおこなった。

### 4.2 各処理の実データによる評価

#### 4.2.1 分類処理の評価

手描きのクラス図、ユースケース図、シーケンス図をそれぞれ 90 枚ずつを学習用データセットとし、さらにそれぞれ 30 枚ずつを評価用データセットとする。

CNN のパラメータと実データを用いて分類させたときの精度の関係を表 1 に示す。

表 1 設定パラメータとその際の結果 1

フィルタサイズ	スライド	分類精度
4 × 4	1 × 1	72.0 %
8 × 8	1 × 1	33.3 %
10 × 10	1 × 1	33.3 %
4 × 4	2 × 2	63.0 %
8 × 8	2 × 2	71.0 %
10 × 10	2 × 2	70.0 %

スライドを1層目に2×2にし2,3層目を1×1に変えた結果を表2に示す。

表2 設定パラメータとその際の結果2

フィルタサイズ	分類精度
4×4	88.8%
8×8	80.0%
10×10	33.3%

#### 4.2.2 抽出処理の評価

手描きのクラス図データ90枚を入力して実験をおこなった。サンプルに対する矩形認識率は50.0%である。なお、矩形が隙間なくしっかり描かれていて、線分と矩形によって閉鎖的な図ができていない、という制約を与えた63枚のサンプルに対する矩形認識率は71.4%である。

#### 4.2.3 評価処理の評価

4.2.2節と同様に手描きのクラス図データ90枚を入力して実験をおこなった。サンプルに対するNOCの最大値を用いた評価の精度は32.2%である。矩形と線分の個数が正しく抽出できているクラス図データ25枚を入力して実験した結果、NOCを用いた評価の精度は96.0%であった。

### 5 考察

本研究では、システムの基本構造にBlackboardパターンを用いたが、Pipes and Filtersパターンを用いた設計も提案できる。本研究では大量のデータを扱う必要があるため、Pipes and Filtersパターンのように処理を次々に加えていく方法は非効率である。

また、本研究では採点に必要な処理を分離しておこなった。処理を分割せずCNNを用いて複雑さを学習させて採点する方法も考えられるが、本研究に比べて似た図式の分類が難しくなる。処理を分割することで、採点者の思考過程に近い設計をすることができ、採点者がそれぞれの処理結果を確認して必要に応じて修正することができる。

本研究で実験的にCNNのフィルタサイズとスライドを変えた結果、分類処理の最も高い精度は88.8%となった。学習できていない部分を省いても、スライドを同じ値にせず、各層での画像サイズに合わせた大きさにすることで精度を高めることができた。また、分類に要した時間は20秒程度で、人手でおこなうより早く効率的である。テストデータで分類をした結果、誤分類された画像に共通した特徴が見つからず誤分類された理由を示すことができなかった。テストデータを増やし誤分類の特徴を見つけることができれば、それに対応した前処理やハイパーパラメータの設定をし、分類精度を高めることができる。本研究では層の数を三つにして線分要素、矩形要素、その繋がりを順に抽出したが、ユースケース図やシーケンス図には別の要素がある。なので、ユースケース図ならば円形要素を抽出する層、シーケンス図ならば文字要素を抽出する層などを加えることでより正確に分類することが出来ると思う。

抽出処理では塗りつぶしによって矩形と線分を分離して抽出したが、抽出処理と評価処理の精度を上げるためには二つの課題がある。一つ目は矩形が完全に閉じられていないときに矩形として抽出できないことである。本研究では縮小処理をおこなったが十分な課題解決に至らなかった。二つ目は矩形と線分の間に閉鎖的な図ができてしまうとその部分をすべて矩形として抽出してしまうことである。その解決方法として指定したオブジェクトの範囲以上の矩形があった場合、そのオブジェクトの中心を始点として塗りつぶしをおこなうことで解決すると考えられる。

評価処理では、抽出処理で依存や実現のような点線の線分を認識できなかったため評価精度は低かった。それらを認識する方法としてHOG特徴量とSVMを用いることで線分の種類と矢印の方向が認識できると考えられる。

### 6 おわりに

本研究では、手描き図式を自動採点するシステムのためのソフトウェアアーキテクチャの設計を目的としてシステムのプロトタイプを作成した。

採点に必要な処理を分類処理、抽出処理、評価処理の三つに分割することで、異なる種類の図式を誤った方法で評価することを避け、採点対象の図式に合わせて容易に評価方法を変更が可能になった。システムの基本構造にBlackboardパターンとCommandパターンを用いた。

分類処理の精度が最も高かったのは88.8%であった。本研究では、パラメータのフィルタサイズとスライドを変化させ精度の高いものを採用したが、より精度が高めるために、ほかのパラメータも適当に設定する必要がある。

抽出処理での矩形認識率は50.0%で、制約を与えたデータの矩形認識率は71.4%である。精度を上げるためにはSVMといった機械学習を用いることが考えられる。

BlackboardパターンとCommandパターンの設計をおこなったが、実装とその妥当性の検証までには至らなかったため、今後の課題である。

### 参考文献

- [1] E. Gamma, R. Helm, R. Johnson, and J. M. Vissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995.
- [2] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal: Pattern-Oriented Software Architecture, A System of Patterns Volume 1, WILEY, 1996.
- [3] L. C. Briand, P. Devanbu, and W. Melo: "An Investigation into Coupling Measures for C++," Proc. of The IEEE 19th Int'l Conference on Software Eng., pp. 412-421, 1997.
- [4] S. R. Chidamber and C. F. Kemerer: "A metrics suite for object-oriented design," IEEE Software, vol. 20, no. 6, pp. 476-493, 1994.