

麻雀牌検出器構築のための視覚情報処理を考慮した学習データ生成について

鈴木 涼介^{†1,a)} 大園 忠親^{†1,b)} 新谷 虎松^{†1,c)}

概要：物体検出器の構築における課題の1つに、大規模データセットの構築が挙げられる。解決方法の1つに、CGを用いたシミュレーションに基づくデータ生成がある。しかし、現実のデータをシミュレーションすることは容易でない。検出器が十分な精度を得るために、生成データを現実のデータに適切に近似する必要がある。そこで本研究では、人間の視覚情報処理に基づき現実のデータをCGで再現可能なデータに変換する。具体的には、輪郭検出処理を適用する。物体検出器の応用領域として、麻雀牌検出タスクに取り組む。本稿では変換した生成データを学習して麻雀牌検出器を構築する手法について述べる。

1. はじめに

拡張現実感技術のような現実空間におけるインタラクティブなシステムのための情報処理において、物体検出器が有効である。物体検出器とは、画像中の物体の位置と種類を検出するシステムである。機械学習器による物体検出器の構築には、応用領域に適した大量の学習データを必要とするという問題がある。本問題を解決するために、コンピュータグラフィックス (CG) による学習データ生成手法が知られている。しかし、CGの質が物体検出器の精度に反映され、かつCGによる現実空間の完全な模倣の困難さが課題である。本研究では、本課題を克服するための手法を開発中である [1]。具体的には、物体検出対象の画像を、CGによる再現が容易な、何らかの理想的な状態に変換する処理を導入することを試みており、この理想的な状態およびそのための画像処理を探求している。そのために、人間の視覚情報処理を考慮する。本稿では、輪郭検出処理の有効性について報告する。本手法により、CGのみで構築した物体検出器の精度向上が期待できる。

物体検出器の構築のための学習データは、画像中における物体の位置と、種類 (物体名などのクラス) がラベル付けされた画像から構成される。大規模な画像データセットを保有している場合、クラウドソーシングを利用した大人数によるラベル付け作業が可能であるが、金銭的成本とラベルの信頼性の保証が課題となる。このような課題を解

決する手法の一つとして、ラベル付きの人工的な画像データ (合成データ) を生成することが挙げられるが、課題も残されている。ラベル付けされた画像データを生成する事で、画像データの収集およびラベル付け作業を自動化することが可能になり、また低頻度の事象に対する画像データも容易に取得可能になるという利点がある。しかし、画像データの生成には、様々なパラメータの設定が必要となる。正確なシミュレーションは難しく、実環境を撮影したデータ (実データ) と酷似したデータを生成することは容易でない。実環境を完全に再現するような合成データが得られないため、結果として合成データから構築した物体検出器は、実データを適切に処理できない。

本研究では、実データと合成データの差異を減少させる手法として、合成データと実データの両方に輪郭検出処理を適用する方法を検討する。機械学習器による物体検出器にはニューラルネットワークの利用を考える。機械学習の一分野であるニューラルネットワークは、人間の脳内を模倣し表現したものである。そのため、人間の視覚情報処理に基づきデータを変換することで、検出器に都合の良いデータが得られると考えられる。応用領域として、実環境の麻雀牌検出をタスクとして取り上げる。本稿では、撮影した麻雀牌の画像に対して輪郭検出処理を行うことで、CGによる再現が容易なデータに変換し、合成データを学習した検出器を適用する。

本稿の以降の構成を示す。2. では、麻雀牌検出タスクについて説明し、3. でタスクを解決するために必要な合成データの生成及びデータの抽象化手法について具体的に説明する。4. では評価実験を示し考察する。最後に 5. で本稿をまとめる。

^{†1} 現在、名古屋工業大学

Presently with Nagoya Institute of Technology

a) suzuri@toralab.org

b) ozono@toralab.org

c) tora@toralab.org

2. 麻雀牌検出

本稿では、実環境の麻雀牌検出をタスクとして取り上げる。先行研究として、麻雀における得点計算を支援するため、麻雀牌の認識を行うシステムを開発している [2]。並んだ麻雀牌を切り抜く矩形検出器と、切り抜いた画像を分類する分類器をカスケード接続することで軽量の麻雀牌認識を可能にした。しかし、矩形検出精度及び分類精度が場合により低下することが欠点として挙げられる。特に矩形検出器が環境に頑健でない。矩形検出器は、検出時の特徴量として麻雀牌の輪郭を使用している。輪郭の中でも特に麻雀牌の境目の輪郭は撮影時の角度や環境光によって見た目が変わるため、検出が不安定であった。そこで本稿では、表面に描かれた文字や図形も特徴量として用いる。それらの特徴量は環境に左右されにくく、環境に頑健な検出器を構築することができると期待できる。加えて、後述するフィルタ処理を行うことで、強固な特徴量のみを検出に用いる。検出器として、機械学習を用いた検出器である Single Shot Multibox Detector (SSD) [3] を用いる。機械学習には、多様な状況を網羅した大量の学習データが必要とされる。既存のデータセットが存在せず、またデータの収集やラベル付けのコストがかかる。簡易にデータセットを構築するため、データの生成を行う。データ生成の手法として、CG を用いてデータセットを生成する。CG を用いて様々な環境をシミュレーションすることにより、環境に頑健な検出器を構築する。また、このシステムはオフライン環境下におけるモバイル端末での実装を目的としている。そのため、複雑性が低く、処理が軽量の検出器が必要となる。画像入力を大きくするに従い検出器は複雑性を増し計算量が増加するため、小さな画像で検出できることが望ましい。小さな画像において、麻雀牌の識別は困難なタスクである。麻雀牌の萬子は表面に描かれた 2 文字のうち、1 文字が異なるだけであり似通っている。また、字牌は画数が多く、遠くから撮影した場合には文字が潰れてしまう。他の種類の牌にも見た目が似通っているものや、潰れて見えないものがある。このような特徴を持っている麻雀牌が隣接して密集するため、麻雀牌の検出は困難なタスクであることが予想される。

3. 合成データを用いた検出器構築

3.1 CG によるデータ生成

はじめに、本研究における CG を用いた合成データの生成手法について述べる。レンダリングエンジンに Blender、レンダラーに Eevee を用いる。Blender はスクリプトによる操作が可能であるため画像データ及びラベルデータ生成を自動化できる。生成するオブジェクトは麻雀牌 ($tile_i$)136 個、雀卓 ($table$)、光源 ($light_i$)9 個、カメラ ($camera$) である。大きさのパラメータを S 、位置パラメータを P 、方向パ

ラメータを D とし、保有するパラメータを次に定義する。

- $tile_i = \langle S_{tile}, P_{tile}, D_{tile} \rangle$
- $table = \langle S_{table}, P_{table}, D_{table} \rangle$
- $light_i = \langle P_{light}, D_{light}, energy = 200W \rangle$
- $camera = \langle P_{cam}, D_{cam}, focal\ length = 30.83mm \rangle$

麻雀牌は全部で 34 種類（萬子、筒子、索子 9 種類ずつ、字牌 7 種類）あり、それぞれ 4 つずつ作成する。 $S_{tile} = H26mm \times W20mm \times D16mm$ である。直方体を面取りし、表面にそれぞれの種類に応じたテクスチャを貼り付ける。雀卓は、 $S_{table} = H0.65m \times W0.65m$ の平面で、表面にテクスチャを貼り付ける。貼り付けるテクスチャは、所有する麻雀セットをカメラで撮影したものである。準備負担が軽微であることに加え、麻雀牌の表面が学習データと認識対象で同一ソースとなるため、検出結果に有利に働くと考えられる。光源は上方 2m の位置に 1 つ、その光源を中心として 1 辺 4m の正方形を形作るように等間隔で 8 つの光源が並ぶ。カメラの焦点距離はテクスチャなどの実データを撮影したカメラと同様の焦点距離としている。

データを生成する際、オブジェクトの位置や方向が重要となる。先に定義した麻雀牌とカメラの変数パラメータに関して、経験的に値を設定する。一例として、カメラを雀卓の下に配置することができるが、そのデータには何も写っているものが無いため除外する必要がある。得点計算などのために手牌付近を撮影することが想定されるため、位置や向きを制限することができる。麻雀牌はレンダリングされる度に再配置されるため、様々な状態の画像を生成することができる。プレイヤーが綺麗に麻雀牌を並べるとは限らないため、位置や向きにノイズを加える。

設定したパラメータを元に、ラベル情報を生成する。Blender でレンダリングする際に内部で行なっている、物体の表示・非表示の判定が取得できない。レンダリング範囲内の麻雀牌の表面が表示・非表示か判定する必要がある。麻雀牌の表面の中心座標と、外側向きの法線ベクトルを取得する。カメラから表面の中心へのベクトルと、法線ベクトルの交差する角度を θ とする。この時、面が見えるのは $\frac{\pi}{2} < \theta < \frac{3}{2}\pi$ である。厳密には θ が $\frac{\pi}{2}$ 付近では表面の情報を識別することは難しいため、 $\frac{5}{9}\pi < \theta < \frac{13}{9}\pi$ とする。2 つのベクトルの内積から計算し、それぞれの麻雀牌の表面の表示・非表示判定を得る。レンダリング範囲内かつ表示判定の麻雀牌の表面座標を、クラス名と共に xml 形式でデータ化する。データ構造は大規模画像データセット PASCAL_VOC と同様とする。ラベル情報の生成を再現可能にするため、擬似コードを Algorithm 1 に示す。initialize() は前述したオブジェクトを生成・固定パラメータをセットし、オブジェクト群を返す関数である。arrange_tiles_and_camera(random) は乱数 (random) を受け取り、その値を元に雀卓上に山・手牌・鳴き牌、捨て牌、

カメラを配置し、配置した麻雀牌のリストを返す関数である。 *obtain_render_pos(tiles)* は配置された麻雀牌のリスト *tiles* を受け取り、レンダリング範囲に入っている麻雀牌のリストを返す関数である。 *create_annotation()* はレンダリング時に表示判定がある麻雀牌のリスト *rendering_tiles* を取り、xml データを返す関数である。 *render(objects)* はオブジェクト群の情報をもとに、レンダリングする関数である。最後に、作成した xml とレンダリングした画像を保存する。この操作を決められた回数 (*number_of_render*) だけ繰り返す。今回は *number_of_render* を 1000、後述する実験では 1000 から 5000 まで変化させている。レンダリングする画像の解像度を 300px×300px とする。1 枚あたりのレンダリングに 3 秒程度を要する。

カメラのパラメータによって、レンダリング範囲に雀卓の外が映る場合がある。本研究では、デフォルト設定のまま、灰色の背景を描画するようにしている。背景の再現は様々なパターンが考えられ、シミュレーションすることが困難である。先行研究において、背景に実環境とは関係のない単色のテクスチャを学習データに使った場合、検出するバウンディングボックスが背景を含まないよう少し小さくなるが分かっている [4]。雀卓の表面や、その他の背景のシミュレーションは検出精度に悪い影響を与えないと考えられる。

3.2 視覚情報処理を考慮した画像変換

本稿では合成データと実データの差異を減少させるフィルタ処理を画像に適応する。合成データと実データの双方に前処理として多種のフィルタを適用し、変換したデータを用いて検出器を学習させ、検出する。フィルタは、視覚情報処理を考慮して選択する。視覚情報は低次である輪郭や色などの情報から、高次である意味情報を含んだ物体の情報などに分けることができる。ニューラルネットワークでも同様に、入力層に近い部分は低次、出力層に近い部分は高次の役割を担うように設計されている。低次の部分の処理である輪郭の認識、また低次と高次の間では低次で選択された視覚情報の特徴量を元に、実環境をイラストのように認識しているのではないかと仮定する。特徴量の選択を明示的に行なうことで、どのような処理が精度に影響するのかを調査する。

フィルタとしてウェーブレット分解及び eXtended Difference-of-Gaussians (xDoG) [5] の 2 種類の輪郭検出手法を用いる。ウェーブレット分解は画素を周波数ごとに分解する。高周波数を取り出すことで、有用な特徴量と考えられる輪郭のみを抽出できる。輪郭検出は人間の視覚処理における低次の役割である。xDoG はガウシアンフィルタの差分から輪郭を抽出し、更に画像をインクイラスト化する。

ハイパーパラメータは麻雀牌やその表面の文字や図形を

Algorithm 1 Generate dataset by Blender

```

1: objects = initialize()
2: number_of_render = 1000, i ← 0
3: while i < number_of_render do
4:   tiles = arrange_tiles_and_camera(random)
5:   rendered_tiles = obtain_render_pos(tiles)
6:   annotation_xml = create_annotation(rendered_tiles)
7:   image = render(objects)
8:   save(annotation_xml, image)
9:   i ← i + 1
10: end while

```

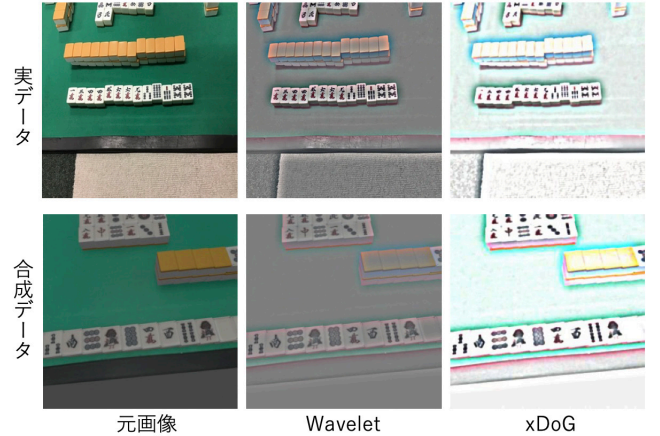


図 1 CG 画像と現実の画像の比較

視認しやすいと感じる値に設定している。合成データ、実データ、それらに各フィルタを適用した画像を図 1 に示す。それぞれの合成データを SSD に学習させ、麻雀牌検出器を構築する。

4. 評価実験

4.1 実験設定

合成データで学習した検出器の精度を測り、実用性を評価する。また、学習データを 1000 から 5000 枚まで 1000 枚ごとに評価し、検出器構築にかかるコストを評価する。学習用のデータを Blender で生成し、データに対して各種フィルタを適用した。学習データとテストデータを 8 : 2 の割合で分割した。50epoch、重み減衰は 0.9 で学習した。重みは ImageNet で事前学習されたものを用いた。学習時、20epoch 程度で損失は減少しなくなったため、過学習されていると考えられる。

ここで、予備実験としてカメラと被写体の距離に関して調査した。図 2 はデータに xDoG を適用した検出結果の一例である。検出した矩形に信頼度の値が記述されている。左の図が成功例、右の図が失敗例である。左の図と右の図では対象物とカメラの距離が異なる。被写体の中心から、左の図は 29cm、右の図は 34cm である。麻雀牌の大きさによって結果が異なり、特に 30cm 程度より離れると大きく検出率が低下することが分かる。実環境下では、手牌や鳴き牌などの得点計算などに必要である重要な牌は近くで

撮影することが容易いため、問題は発生しない。しかし、被写体の近くで撮影するようにユーザに注意を促す必要がある。実験では、左図と同様の 29cm の距離、つまりプレイヤが得点計算や捨牌を選ぶことを目的に撮影する場面を模倣してテストデータの撮影を行なった。1つの画像に14個の麻雀牌が映っている。ラベル付けに麻雀牌1つあたり5秒程度の時間を要し、麻雀牌が14個映っている場合70秒かかる。テストデータと大量に生成するのは困難であるため、今回は50枚とした。麻雀牌の個数は計700である。評価指標にF値を用いる。F値は検出器が検出する矩形領域から計算する。麻雀牌検出において、同じクラスの麻雀牌が隣接し、検出した矩形領域が両方に重なる場合がある。この時、最近傍の物体を検出したとみなす。

検出器が検出する矩形領域の信頼度に対して域値を設けた。矩形領域の域値に関して、SSDがデフォルトで採用している0.6を採用した。域値の値に対して精度と再現率はトレードオフの関係であり、域値を高くするほど精度が上昇し再現率が低下する。また、SSDでは検出する矩形の数を指定し、その数になるまで検出した矩形を統合する。矩形の数を想定する麻雀牌の数14個よりも十分に大きい50とすることで、精度を厳密に計測した。

4.2 実験結果及び考察

実験結果を図3に示す。横軸が検出器が学習したデータの個数、縦軸がテストデータに対するF値である。ウェーブレット分解及びxDoGにより、元画像よりもF値が向上した。また、ウェーブレット分解とxDoGの間に差は見られなかった。xDoGのインクイラスト化は結果に寄与せず、またどちらのフィルタも有する輪郭検出の処理がF値を向上させたと考えられる。データ数に関して、数と精度は比例しなかった。1000枚で過学習もしくは精度が飽和している状態であるため、検出器構築には1000枚で充分である。一方、精度が向上しないのは、個々の麻雀牌の解像度が低いことが原因であると考えられる。そこで、実データを検出器の入力サイズにリサイズするのではなく、入力サイズに分割すればよい。モデルの複雑性は変わらないが、元の画像に対する計算量は増加するため、タスクに応じて実用的な分割サイズを調査する必要がある。また、本稿では実データで学習した検出器を評価していない。どれほどの実データを用意すれば今回の実験と同程度の精度を得られるのかを調査することで、本手法の有用性を正確に示すことができる。これらを今後の課題とする。

5. おわりに

本稿では、CGによって多様な合成データを生成し、麻雀牌検出のための検出器の学習データとして利用した。合成データと実データの間に差異があるため、実データをCGで再現可能なデータに変換する手法について述べた。

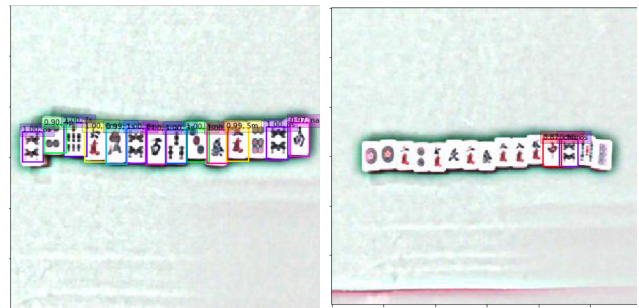


図2 検出の成功例と失敗例

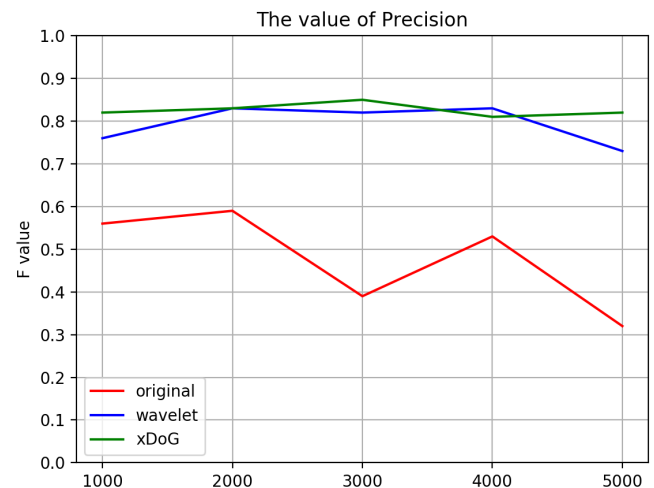


図3 データ数によるF値の変化

データの変換手法については、人間の視覚情報処理に基づき、輪郭検出フィルタを適用した。具体的には、ウェーブレット分解及びxDoGを適用した。検出器にはSSDを利用し、変換した合成データを学習させた。実験結果から、これらの輪郭検出が精度の向上に有効であることが分かった。計算量は検出にかかる演算に比べれば非常に少ないため、輪郭検出フィルタを積極的に用いるべきである。

謝辞 本研究の一部はJSPS科研費JP16K00420, 19K12097, 19K12266の助成を受けたものです。

参考文献

- [1] 鈴木涼介, 大園忠親, 新谷虎松. “抽象化された学習データを利用した麻雀牌検出器構築手法について”. 信学技報, vol.119, No.317, pp.7-12, 2019.
- [2] R. Suzuki, T. Ozono, T. Shintani, “An Offline Mahjong Support System Based on Augmented Reality with Context-aware Image Recognition”. ESKM2019, IEEE, pp.127-132, 2019.
- [3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg. “SSD: Single Shot MultiBox Detector”. ECCV, pp.21-37, 2016.
- [4] 鈴木涼介, 大園忠親, 新谷虎松, “AR環境における記号理解のための学習データ生成手法の検討”. 信学技報, Vol.119, No.139, AI2019-6, pp.83-88, 2019.
- [5] H. Winnemoller, J. E. Kyprianidis, and S. C. Olsen. “Xdog: an extended difference-of-gaussians compendium including advanced image stylization”. Comput. Graphics, vol.36, no.6, pp.740-753, 2012.