インテリジェントシステム 期末レポート

B151235 山下直哉

2018/11/16

1 Abstract

授業において様々な手法での機械学習の理論を学習した。このレポートは、理論学習の演習として将棋の駒の識別モデルを作成したことについて、結果と展望を記すものである。識別器作成には Microsoft Azure が提供している、Custom Vision を使用した。Custom Vision は千代田による「松屋警察」(2007)[1](98% の確率で松屋の牛丼を判定可能) など有効なパフォーマンスを示した先行事例があり、手軽さも他のツールを圧倒している。将棋の駒の判別として本来ならば、相手駒である反転駒を考慮して 30 クラス分類となるが、実際の対局のキャプチャ画像を使用したため成銀の画像が入手できず、28 クラス分類で識別器を作成した。それぞれ 100 枚の画像を使用し、Precision:92.7%, Recall:90.0% の精度で識別ができる識別器を作成できた。また、使用した全ツールは github[2]にて公開した。

2 先行事例研究

インターネット上において機械学習分野ではすぎゃーんが Tensor Flow によるアイドル顔識別器 (2016)[3]、その後改良を加え 2000 人のアイドル顔を自力でタグ付けし 1000 人程度の識別が可能な識別器 (2017)[4]などを作成し、機械学習の普及に寄与していた。

すぎゃーんのブログにおいて、将棋駒の分類を機械学習によって画像認識する話題[5]があり、Tensor Flow の学習済みモデルを適用し識別器を作成していた。そこでは識別の精度は悪く、研究は失敗したかのように書かれていた。しかし、学習画像と検定画像は大きく異なっており (図 1)、これをもって機械学習による画像識別機が将棋駒の判別に向いていないと帰結させるのは早計に思えた。

そこで私は NHK 杯の録画から盤面が映されているコマを画像として保存し、その画像に対して学習し、NHK 杯の画像に対して識別することでどれだけの精度を持つ識別機が作成できるか研究した。

図 1: すぎゃーんが使用した学習画像と検定画像

[1] ちょまど Madoka @chomado https://twitter.com/chomado/status/898812060624068609 - 2018/11/16 閲覧
[2] Detect Shogi-peacies by deep-leaning AI - conao3/shogi-detect https://github.com/conao3/shogi-detect - 2018/11/16 閲覧
[3] TensorFlow によるアイドル顔識別器の話 - 2016.12.13 TensorFlow User Group #2 https://qiita.com/sugyan/items/f89cba95d67ab297d306 - 2018/11/16 閲覧
[4] TensorFlow と出会った「ドルヲタ」エンジニアが 1 年かけてたどり着いた境地 ー LINE すぎゃーん（sugyan）氏 https://press.forkwell.com/entry/2017/03/22/085525 - 2018/11/16 閲覧
[5] 将棋駒画像の分類器をラクして作る https://memo.sugyan.com/entry/2018/05/02/182830 - 2018/11/16 閲覧

3　データ作成

学習・検定データは以下の通り収集した。

1. NHK 杯の映像を入手し、将棋盤の全面が写ったコマを手動で保存した。（盤面画像: 約 200 枚）
2. Python による古典的な特徴解析により将棋盤領域を検出し、将棋盤領域のみの画像を作成した (図 2)。
3. Python により将棋盤領域のみの画像を 9 × 9 のマス目画像にスライスした。（マス目画像: 約 16,000 枚）
4. PHP による Web アプリケーションを作成し、手動で駒画像と空き画像に分類した（駒画像: 約 5,000 枚, 空白画像: 10,000 枚: 33px × 33px）
5. 前段のアプリケーションにより、手動で 28 クラス分類を行った (図 3)。
6. サンプル数の足りない駒画像については Python によりランダムノイズを与え、学習データを増やした。

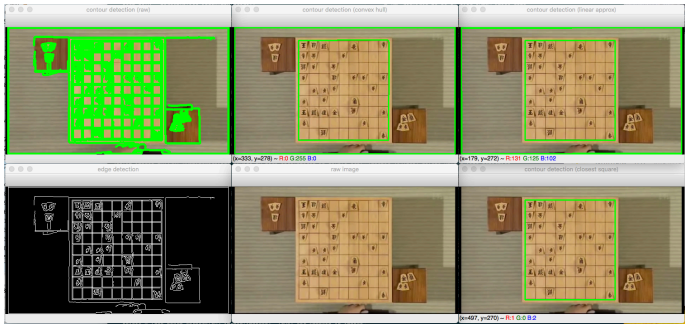

図 2: Python による古典的な特徴解析

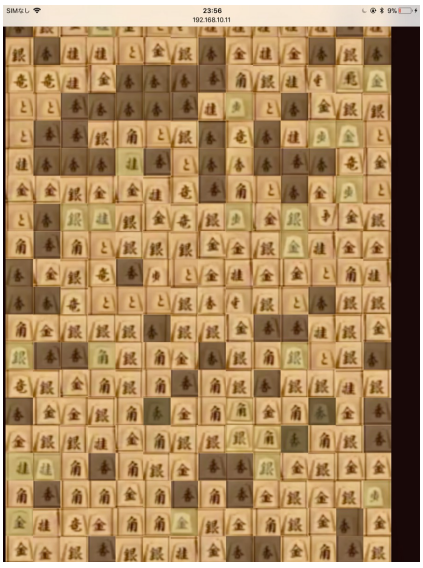

図 3: PHP による Web アプリケーションでの画像分類

4　学習・性能分析

本研究では識別器作成機として Microsoft Azure の Custom Vision[6]を選定した。Custom Vision は 1 クラスにつき最低 5 枚、推奨 50 枚の画像を必要とし、画像をアップロード・タグ付けし、「Train」ボタンを押すだけで識別器を作成できる。

Python によるランダムノイズを加え、各クラス 3000 枚の画像を用意したが、そのうち 100 枚の画像をアップロードし学習を行った。

結果として Precision: 92.7 %, Recall: 90.0% の識別器を作成できた (図 4)。クラス別に正答率を見ていくと、「桂馬」や「歩」の識別率が高く、「香車」や「成桂」の識別率が低かった。

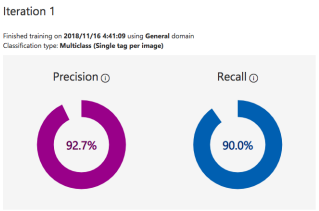

図 4: Custom Vision による識別器作成, 評価

5　考察

実際の対局映像のキャプチャを利用しているため、クラスにおいては極端に元データの少ないクラスが現れていた。例えば識別率の低い結果となった「香車 (Recall: 77.2%)」については「歩」との混同が多い結果となった。これは手動で分類する際にも目を細めて分類した難しい問題であり、もっと解像度の高い画像で行った場合、識別率が上がる可能性がある。本研究においては 1 マスは 33px 正方形として正規化し扱ったため、100px 正方形などでもう一度試してみたい。

また「成桂 (Recall: 77.8%)」については、本研究において、NHK 杯の実際の対局から画像を入手したため「成桂」の元データを収集することが困難で、200 枚の盤面データから収集できた「成桂」元データはわずか 10 枚だった。そのため学習データの大部分を Python のランダムノイズにより生成された画像であり、それに由来し正答率が低くなってしまったと考えられる。

今後は Tensor Flow などを利用して Deep learning を利用した識別器を作成したい。

[6] Microsoft Azure - Custom Vision `https://www.customvision.ai` - 2018/11/16 閲覧