

Tensorflow を用いたトランプカードの識別

渡辺 裕美 (指導教員: 粕川 正充)

1 はじめに

機械学習用のソフトウェアライブラリである Tensorflow を使用した研究で, cifar-10 という 10 のクラスで分けられている 32×32 ピクセルのカラー画像のデータセットを識別する研究がある [1]. 本研究では, cifar-10 を分類するサンプルプログラム [2] を利用して, トランプカードの画像セットを識別した.

2 関連研究

ベテランでなければ難しいきゅうりの仕分けを, きゅうりを液晶ディスプレイを内蔵したテーブルにおき, 上部に設置してあるカメラで読み取り tensorflow の画像認識を利用し選別するという機械を開発したという報告がある [3].

3 実験準備

3.1 実験環境

以下の環境で実験を行った.

- CPU : AMD ryzen7 1700 3.4GHz 8Core 16Thread
- Memory DDR4 : 2666MHz, 8GB \times 2
- OS : Ubuntu16.04LTS
- GPU : NVIDIA Geforce1070(Memory : 8GB)
- Software : CUDA 8.0, python3.5.2, Tensorflow1.4.1

3.2 画像データセットの作成

データセットに使用するトランプカードの画像は, 実際にトランプカードを購入しスキャナーで取り込んだ. 取り込んだトランプカードは BICYCLE, Bee, TALLY-HO, Nintendo, HANAYAMA, daiso, serial-1, serial-2 の計 8 種類 416 枚. テスト用に TALLY-HO52 枚, 学習用に残り 7 種類 364 枚の画像を用意した. クラスは 52 クラスに分類した. cifar-10 のデータ構造に合うように用意した画像を元にデータセットを作成した. なお, ピクセル数が縦横で同じなのはトランプカードの縦横比が種類によって異なるために正規化する必要があったからである.

3.3 予備実験

cifar-10 では 10 万回の学習で結果が得られる. 本研究でも最初は 10 万回学習させた. 解像度 32×32 ピクセルで 10 万回学習させた結果, 学習時間が 1 時間 15 分で正解率 97.7% だった. learning_rate(図 1) を参照すると 1 万回でほぼ収束しているので 1 万回で実験した. 1 万回学習させた結果, 実行時間が 7 分弱で正解率が 96.1% だった. 正解率がほぼ変わらず学習時間が大幅に短縮されたので, 以降の学習回数はいずれも 1 万回とした. 参考までに MacBook Pro で GPU なしの tensorflow を試した場合, 学習回数が 1000 回で実行時間は 22 分かかった.

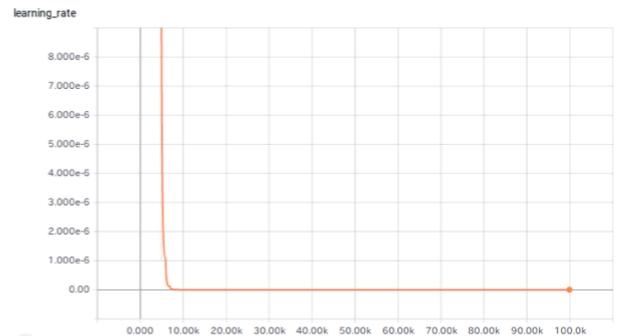


図 1: 10 万回学習させた時の learning_rate

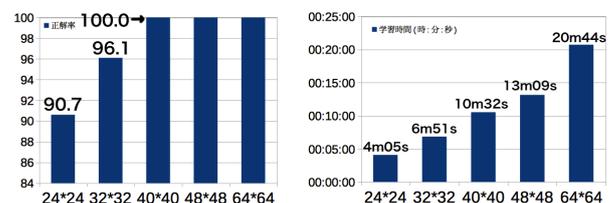


図 2: 解像度ごとの正解率 (左図), 学習時間 (右図)

4 実験 1

解像度を 24×24 ピクセル, 32×32 ピクセル, 40×40 ピクセル, 48×48 ピクセル, 64×64 ピクセルの 5 種類を用意して実験した.

4.1 結果

解像度ごとによる正解率と学習時間は図 2 のようになった.

4.2 考察

32×32 ピクセルでは 96~98% の正解率だったが, 間違えた箇所を検出するとクラブの J をスペードの J と推測していた. 32×32 ピクセルの画像 (図 3 の上段中央) を確認すると, スートのところが潰れて識別しづらくなっている. 正解率が 90% に下がった 24×24 ピクセルでは, A から 10 までは数字もスートも間違いがなかったが, 絵札ではスートの間違いの他に数字の間違いも確認できた. 24×24 ピクセルの画像 (図 3 の上段左端) をみると, 文字も潰れていることがわかった. 40×40 ピクセルの画像 (図 3 の上段右端) は正解率が 100% だった. 数字と数字の下のスートは少し潰れているが, 絵柄の中の人は視認でも見分けられる. スートも 32×32 ピクセルよりははっきりしている.

5 実験 2

テスト用の画像データに弱視用トランプカード (図 4) (以下 E とする), 数字とスートのあところ以外は全て絵になっているトランプカード (以下 T とする), 2~10 の絵柄が普通のトランプカードと部分的に異なり, A と絵札の絵柄の部分が普通とは大きく異なったトランプカード (以下 M とする) の計 3 種類を使用して実験を行った. なお, 画像の解像度は数字とスートが綺麗に読み取れる 64×64 ピクセルを使用した.

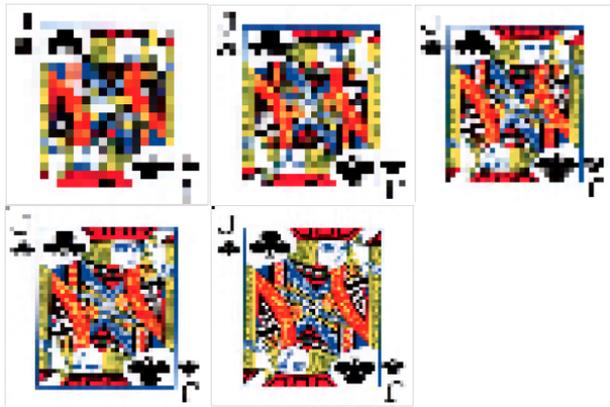


図 3: 上段が 24 × 24, 32 × 32, 40 × 40, 下段が 48 × 48, 64 × 64 ピクセルのトランプカードの画像 (すべてクラブの J)



図 4: 弱視用のトランプカード

5.1 結果

正解率は図 5 のようになった。

5.2 考察

普通のトランプカードは左上と右下に数字とスーツが来て中央に絵柄が来るが、E は数字とスーツの位置が普通のトランプカードと左右対称の位置にある。そのために、数字もスーツも認識されない結果になった。T は、いっけん数字とスーツで見分けることができそうだが、普通のトランプカードと比較すると、数字とスーツの形が異なり、位置も若干ずれている。M では、4 スートの 2 から 10 の全 36 クラスは全て正解していたが、A と絵札はほぼ不正解であった。64 × 64 ピクセルでは、解像度が小さいため数字とその下のスーツは潰れてしまい、絵柄では識別できないため不正解となった。以上の結果から、64 × 64 ピクセルでは、絵柄が大きく異なるとトランプカードはほぼ識別できないことがわかった。

6 まとめ

実験 1 より、絵札の絵柄が左右対称になっている種類もあるが、プログラムでは画像データをインプットする際に、画像を反転処理してデータを増やす工程があるので、結果的に各クラスで絵柄の特徴が似ていて、データ数が少なく解像度が小さくてもほぼ識別することができた。実験 2 より、解像度が小さく学習用データが少ないため、絵柄が大きく異なるとうまく識別できないことがわかった。また、GPU を使用しないと非常に時間がかかることがわかった。

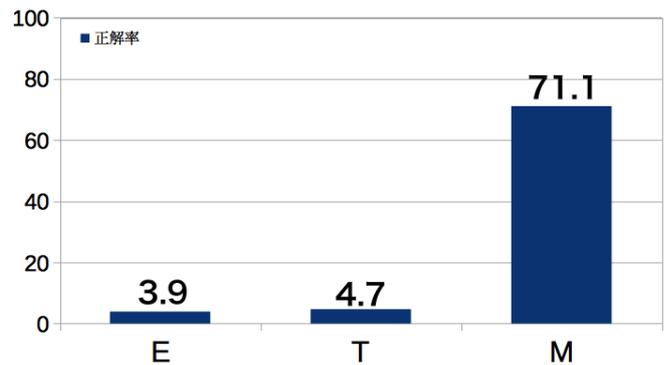


図 5: E の正解率, T の正解率, M の正解率

7 今後の課題

画像の解像度が低いため識別できなかった可能性があるため、画像の解像度をより広げて実験する。データ数が少ないことで絵柄が大きく異なった場合に識別できなかった可能性があるため、数字や数字の下のスーツの部分にバリエーションのあるトランプカードを学習用のデータに混ぜて実験する。様々な種類のトランプカードを画像データに含め識別できるか試みる。

また、数字やスーツを目印にした独自の認識アルゴリズムを開発したい。

参考文献

- [1] Ranzato, M., Krizhevsky, A. and Hinton, G. : Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images,, Proc. the 13th Inter. Conf. on Artificial Intelligence and Statistics pp. 621-628, 2010
- [2] Github, <https://github.com/tensorflow/tensorflow/tree/r0.7/tensorflow/models/image/cifar10>
- [3] 小池誠 : ディープラーニングを用いたキュウリ選果機 開発, 情報処理 Vol.58 No.11 pp.956-957, 2017
- [4] 有山圭二 : Tensorflow はじめました 実践! 最新 Google マシンラーニング, 株式会社インプレス R&D, 2016, ISBN9784802090889
- [5] 中井悦司 : Tensorflow で学ぶディープラーニング入門 ~畳み込みニューラルネットワーク徹底解説, 株式会社マイナビ出版, 2016, ISBN9784839960889
- [6] Ian, G., Yoshua, B. and Aaron, C. : Deep Learning, 2016, ISBN9780262035613