

# Tensorflow を用いたトランプスートの識別

高橋 凜 (指導教員: 粕川正充)

## 1 はじめに

トランプゲームのコントラクトブリッジをする際、手として配られた 13 枚のカードを相手に見せないようにする時に行う図 1 のようなカードの持ち方をファンニングという。ファンニングをする時のスートと数字だけが自分に見えるようにするカードの角度は -30 度から 30 度くらいである。そこで、ファンニングしたトランプのカードがなんであるかをコンピュータに認識させるために以下の実験を行った。本研究では、cifar-10 を分類するサンプルプログラム [1] を利用して、トランプカードの角度を変えた時のスートと数字を識別した。また、ここでは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 cifar-10 という 10 のクラスで分けられている  $32 \times 32$  ピクセルのカラー画像のデータセットを識別する研究を参考にした [2]



図 1: トランプ

## 2 実験準備

### 2.1 実験環境

以下の環境で実験を行った。

- CPU : AMD ryzen7 1700 3.4GHz
- Memory DDR4 : 2666MHz, 16GB
- OS : Ubuntu16.04LTS
- GPU : NVIDIA Geforce1080ti(Memory : 11GB)
- Software : CUDA 10.0, python3.5.2, Tensorflow1.12.1

### 2.2 予備実験

cifar-10 では 10 万回の学習で結果が得られる。また、cifar-10 の学習画像は解像度  $32 \times 32$  ピクセルであるため、本研究でも最初は解像度  $32 \times 32$  ピクセルで 10 万回学習させた。解像度  $32 \times 32$  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 9.7% だった。1 万回学習させた結果、実行時間が 4 分 43 秒で正解率が 9.6% 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。

### 2.3 画像データセットの作成

データセットには昨年の既存のデータであるトランプのカードの画像を使用し、認識が楽なように最もシンプルなスートと数字が描かれている Bee というトランプカード 52 枚 1 セットを選んだ。そしてそのカード 1 枚 1 枚をファンニングする時のカードを傾ける角度の範囲である -30 度から 30 度まで適当な角度である 5 度ごとに角度を変え、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。

## 3 実験

トランプ 1 セット 52 枚を用意し、1 枚 1 枚 -30 度から 30 度まで角度を変えた画像を 676 枚作り、角度を変えた画像でもスートと数字を当てることができるのかを実験した。また、解像度によって正答率が変わることかを調べるために  $24 \times 24$  ピクセル、 $32 \times 32$  ピクセル、 $40 \times 40$  ピクセル、 $48 \times 48$  ピクセル、 $64 \times 64$  ピクセルのトランプのカードを用意した。以下は各ピクセルごとの club の A を 15 度回転させた時の画像である。図 2 参照

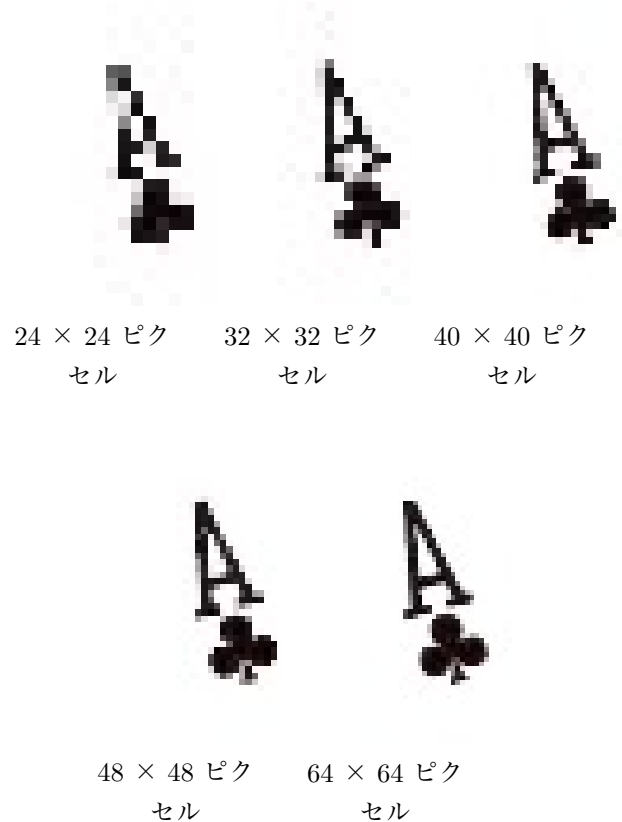


図 2: 各ピクセル数ごとのトランプの画像

### 3.1 結果

ピクセルごとによる正解率は図3, また学習時間は図4のようになった。

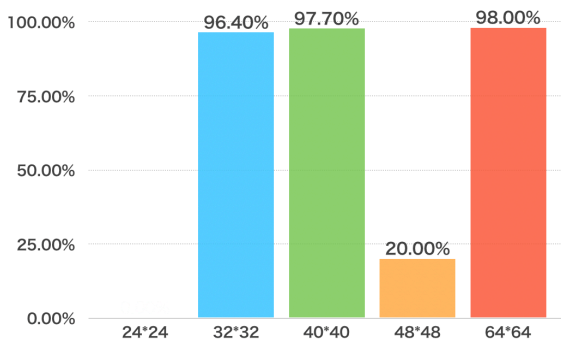


図 3: 正解率

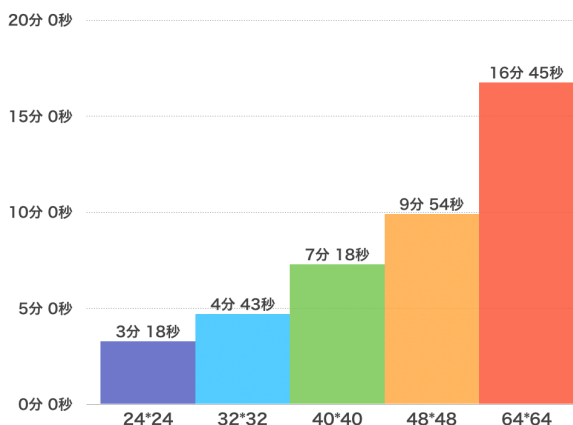


図 4: 学習時間

#### 24 × 24 ピクセル

club と diamond のように黒と赤で全く違うスートを取り出したり, 角度も 30 度と-10 度といった全く違う度数を取り出してきて, 正解率は 0% となった。

#### 32 × 32 ピクセル

club と spade のようにスートの色は一致するようになったが, 数字と角度は違うものを取り出してくることが多かった。

#### 40 × 40 ピクセル

スートと角度は一致するが異なる数字を取り出してくることがあった。

#### 48 × 48 ピクセル

40 × 40 ピクセル から 48 × 48 ピクセルに解像度を上げると 24 × 24 ピクセルの時と同じように club と diamond のように黒と赤で全く違うスートを取り出してくるようになった。

#### 64 × 64 ピクセル

48 × 48 ピクセルよりは正解率が上がったが, 異なる数字を取り出すようになった。

## 4 まとめ

結果を見てわかるように, 学習時間はピクセル数が大きくなるとともに比例して増えることがわかる。ピクセル数の小さいトランプの正答率が下がったのは角度を変えたときに解像度が小さすぎるため, スートと数字が潰れてしまったからだと言える。

## 5 今後の課題

なぜ, 48 × 48 ピクセルの時に正解率が下がり, 64 × 64 ピクセルの時に正解率が回復したのかを突き止めたいと思う。今回一種類のトランプを使用し, スートと数字の一部分を切り取って角度を変えたものをコンピュータに学習させ, 識別したが, 使用したトランプのカードは 1 番シンプルなものであったため今回のような正解率を出すことができたが, 他の種類のトランプの場合ではどうなるのか, また, 変化させる角度の範囲を広げた場合の学習時間や正解率はどうなるのかを調べてみたい。

## 参考文献

- [1] Github, <https://github.com/tensorflow/tensorflow/tree/r0.7/tensorflow/models/image/cifar10>
- [2] ayihiscope, <http://ayihis.hatenablog.com/entry/2016/08/02/162>: TensorFlow のチュートリアルにある CIFAR-10 を動かしてみる
- [3] 人工知能に関する断創録, <http://aidiary.hatenablog.com/entry/20151014/1444827123>
- [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 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, ISBN9780262035613
- [6] 新村拓也, Tensor Flow ではじめる Deep Learning 実装入門, 株式会社 インプレス, ISBN978-4-295-00318-2C3055
- [7] sonickun.log, <http://sonickun.hatenablog.com/entry/2016/07/11>: Deep Learning はじめました
- [8] 初心者のための AI 人工知能テクノロジー <https://newtechnologylifestyle.net/>: cifar-10 のデータセットを用いて cnn の画像認識を行ってみる
- [9] 渡辺裕美, Tensorflow を用いたトランプカードの識別, <http://siio.jp/pdf/grad/2017/2017grad29.pdf>