

# Oracle-Bone Inscription Recognition Based on Deep Convolutional Neural Network

深い畳み込みニューラルネットワークを用いた甲骨文字の認識

著者 : Guoying Liu Anyang Normal University  
Feng Gao Anyang Normal University  
Journal of Computers

2020/06/03 河野 竣介

# 甲骨文字とは

- 甲骨文字とは約3000年前に古代中国殷の時代に使用されていた最も古い象形文字である。
- 当時の政治システムや経済状況を知ることができる
- 甲骨文字を読むには考古学、歴史、文学など様々な知識が必要である。

# 甲骨文字の特徴と研究

Fig.1 に示される通り甲骨文字は  
交差点、線、点からなっている  
無向グラフである。

研究者はグラフ理論を用いて甲骨文字の認識ができないかと考えた

これまでに文字の特徴を利用したハフ変換やヒストグラムを用いる  
ことで文字の認識が行われた。

どの研究も前処理として特徴抽出とパターンマッチングを行う  
→特徴抽出とパターンマッチングに依存しない手法がCNN

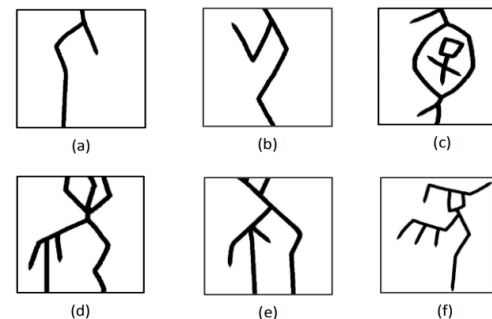


Fig. 1. Some examples of OBIs: (a) the OBI “人” (person); (b) the OBI “匕” (dagger); (c) the OBI “孕” (pregnancy); (d) the OBI “老” (old); (e) a variant of (d); (f) another variant of (d).

# 近年のCNNのディープラーニングへの使用について

- 近年のディープラーニングブームに伴いCNNは文字認識の分野に適応することができた
- CNNが回転や拡大縮小、変形不変性に強いいため

# CNN

- CNNは様々な層で構成された階層的なニューラルネットワークモデルでおもに畳み込み層とプーリング層からなり、Relu層、全結合層、損失層などで構成される。
- 畳み込み層は特徴抽出器として機能し全体的特徴と局所的特徴を、入力された画像の階層的な画像としてとらえる。
- Pooling層は畳み込み層の後に適用され、入力データをより扱いやすい形に変形するために、max-poolingという手法でデータを圧縮し、計算コストを削減する。

# 畳み込み演算

- 畳み込み層の入力 $N$ の重みフィルタのパラメータ $\{I_n | n=1,2,\dots,N, I_n \in \mathbb{R}^{M \times N}\}$  対象の局所領域を $F^t \in \mathbb{R}^{s \times s}$ および $b^t$ をバイアスパラメーターとしたとき 畳み込み演算は

$$O = f[(\sum_{n=1}^N I_n * F^t) + b^t]$$

として表される。

ここで $f()$ はReLU(活性化関数)である。

# 構造

Fig.2に示すように甲骨画像は3 \* 3の受容野が採用された畳み込み層を通っていく。

このアーキテクチャーでは畳み込み適用間隔であるストライドを2とする。

各max-pooling操作後情報を失わないために特徴マップの数を64(層1)から512(層4-5)まで増やしていく

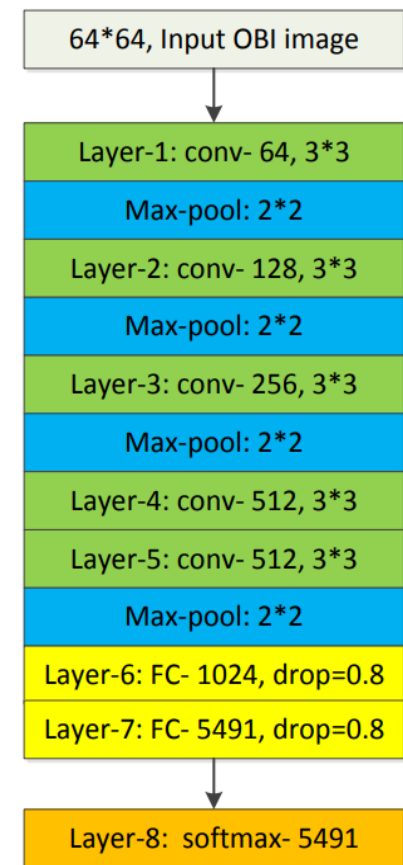


Fig. 2. The architecture of the employed neural network.

# 構造

- 入力甲骨画像は64\*64なので5層畳み込み層と4層プーリング層のあと、特徴マップは平坦化されて8192の次元を持つベクトルとつながる。
- 2つの全結合層(層6および層7)に、それぞれ1024および5491の隠れ層が伴う。
- 最後にソフトマックス関数を使って5491個に分類を行う
- これにより5491もの異なるOBI文字を認識する。

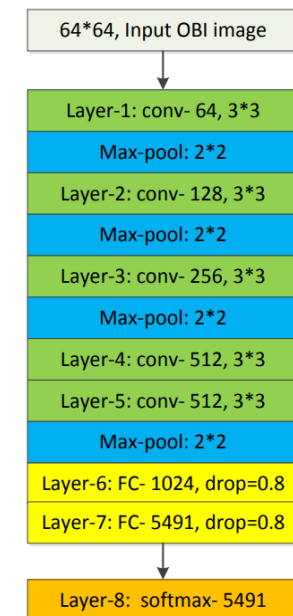


Fig. 2. The architecture of the employed neural network.



# 活性化

- 活性化関数 (ReLU) はネットワークに非線形性を追加することができる
- 活性化関数を使うことによりトレーニングプロセスを最適な状態にすばやく収束でき、ネットワークもまた、従来のものと比較して明らかに改善することができパフォーマンスが向上します。
- Leaky ReLUはよく利用される形式です

$$f(x) = \begin{cases} \lambda x, & x < 0 \\ x, & x \geq 0 \end{cases}$$

本研究では $\lambda=0$ を入力するスタンダードなReLUを使う。

# 初期化とトレーニング

- 最適な重みを見つけるためにネットワーク内すべての層についてすべてのバイアスを0にし、ディープニューラルネットワークで使われるゼロ平均ガウス分布 $N(0, \sigma^2)$ を初期化する

その後、ネットワークを最初の推定値と勾配の二次モーメントを計算する適応モーメント推定(Adam)(アルゴリズム勾配の平均・分散(一次・二次モーメント)から、各パラメータ個別の学習率を計算する最適化手法。)によりトレーニングします

これにより最適な重みを求めることができる

# 実験

- データセット

すり写しによって作られた画像全5491種44868枚

最高277枚最低10未満

10枚未満の画像についてはFig4 のように回転などを加えて増強を行った

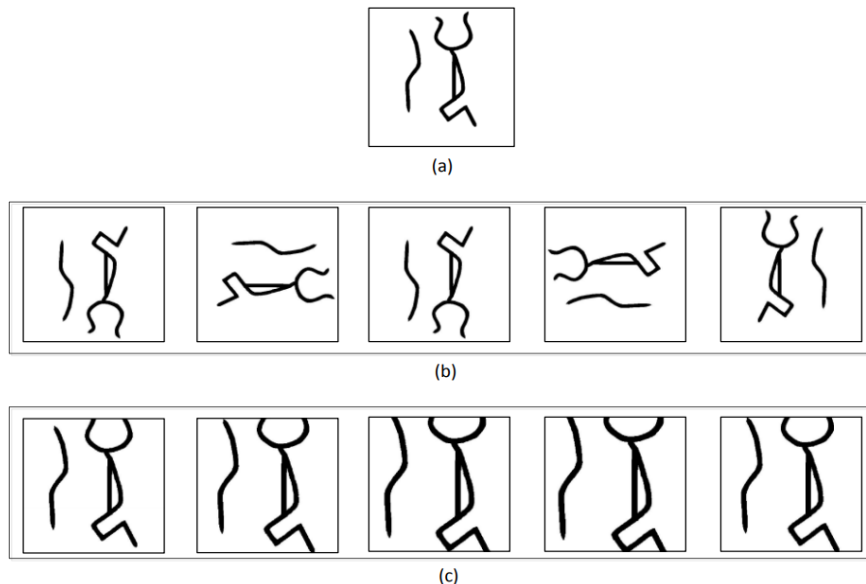


Fig. 4. An example of data augmentation: (a) OBI sample with index 02085; (b) results of randomly flipping (a) 5 times; (c) results of randomly cropping (a) 5 times.

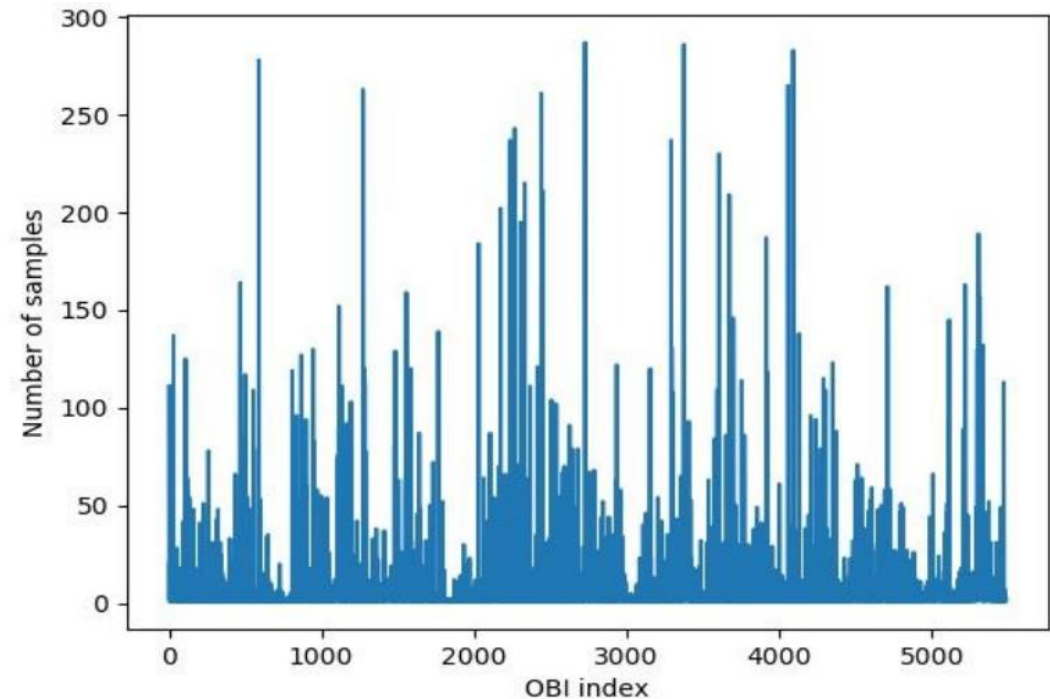


Fig. 3. Sample number of OBIs.

# ネットワークのトレーニング結果

- ネットワークをトレーニングするためにデータセットをトレーニングデータとテストデータに分ける。

最大反復数を20000として、  
トレーニング中にトップと損失の値を  
記録した

最上位の正確さは0.9375

損失値は0.7846であった。

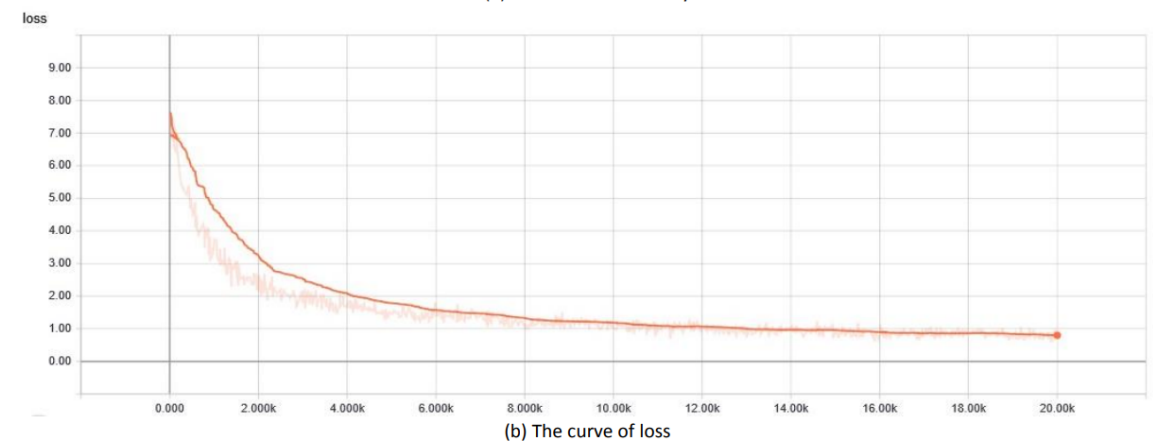
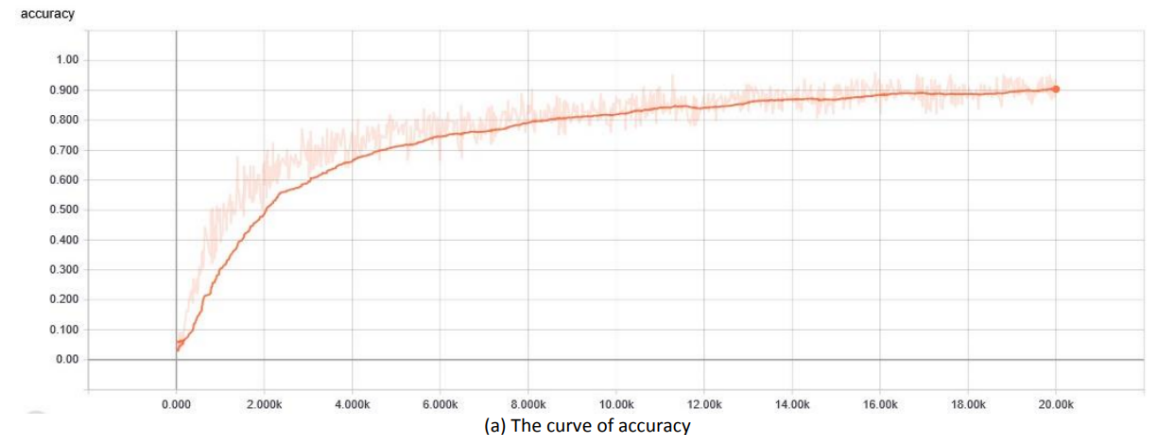


Fig. 5. The training process of our network: (a) the curve of accuracy; (b) the curve of loss function.

# ネットワークのテスト結果

- テストデータでのネットワークのテスト結果

accuracy:top1,top3はそれぞれ91.56 %、97.16 %

OBI-1、OBI-2 はそれぞれ違う文字であり

CNはOBI-1をOBI-2と誤認識した数

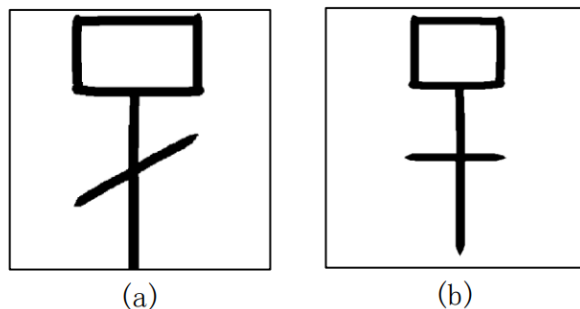


Fig. 6. Two OBIs with serious misclassification: (a) The OBI “子”(child); (b) The OBI “巳” ( the sixth of the twelve Earthly Branches, which is an ordering system used throughout East Asia).

OBI-1	OBI-2	CN
004000	004002	193
001004	001000	183
038000	039000	107
073014	073000	102
004002	004000	91
011001	011000	64
001004	001001	61
150015	150100	53
<b>005000</b>	<b>150075</b>	<b>45</b>
006059	058000	39
010007	010000	39
039005	039000	38
093001	093000	37
001001	001000	36
094002	094023	31
085048	085049	30
150010	020000	29
150075	005000	28
016012	016013	26
001013	001008	25
010000	010007	25
030000	034000	23
096012	096017	23
004002	058000	22
016000	015000	21

# 全体の結果

OBI-1	OBI-2	CN	OBI-1	OBI-2	CN	OBI-1	OBI-2	CN	OBI-1	OBI-2	CN
004000	004002	193	021002	021000	21	134014	093000	14	134014	134006	11
001004	001000	183	063022	063021	21	150005	150006	14	150010	014000	11
038000	039000	107	018035	018033	20	151000	070000	14	003002	003004	10
073014	073000	102	001000	001001	18	001125	001124	13	003013	003126	10
004002	004000	91	010150	010091	18	004022	004021	13	003013	003128	10
011001	011000	64	033007	033002	18	039000	039005	13	018031	018039	10
001004	001001	61	021000	021002	17	076000	020000	13	026001	026000	10
150015	150100	53	030000	033005	17	151121	001170	13	032003	032000	10
<b>005000</b>	<b>150075</b>	<b>45</b>	150010	069000	17	001014	001000	12	039000	038000	10
006059	058000	39	011000	150059	16	033003	033000	12	039011	039000	10
010007	010000	39	020000	014000	16	047000	046001	12	051025	051023	10
039005	039000	38	044001	045000	16	073000	073014	12	051025	051024	10
093001	093000	37	062003	062002	16	089008	089010	12	066001	150004	10
001001	001000	36	064000	141014	16	115008	115029	12	069000	150010	10
094002	094023	31	070011	150026	16	125003	018000	12	091099	048045	10
085048	085049	30	999056	999110	16	150079	003077	12	100000	073000	10
150010	020000	29	001141	083000	15	151097	151098	12	102010	102004	10
150075	005000	28	069000	020000	15	999083	038032	12	107023	107022	10
016012	016013	26	079006	079003	15	001120	001124	11	122066	122064	10
001013	001008	25	141014	064000	15	003080	003077	11	141002	141003	10
010000	010007	25	006098	006000	14	077033	077032	11	009000	009002	9
030000	034000	23	017000	017001	14	077115	077116	11	027004	027008	9
096012	096017	23	032002	031045	14	087024	087025	11	029004	083000	9
004002	058000	22	048075	048073	14	091099	020005	11	044000	046000	9
016000	015000	21	063062	063056	14	114016	114017	11	047000	048021	9

# 今後の課題

- 一部の甲骨画像に対してはサンプルの画像が少なすぎて正確なトレーニングができず学習不足が起きている。
- そのためほかの文字との誤認識が起きている。