Component detection in Chinese character using CNN

Letao Zhou[1,a)] Brian Kenji Iwana[1,b)] Kumiko Tanaka-Ishii[2,c)]
Seiichi Uchida[1,d)]

Abstract: In recent years, CNN (Convolutional Neural Network)-based models have achieved powerful results in the multi-object detection task. However, most models use region-based frameworks, which do region extraction before the CNN's training. Thus, it is still unknown if the CNN can directly understand the image's structure. This study aims to evaluate the CNN's ability of learn the structure of an image by detecting the components (sub-structure) of a character from the line-based Chinese character images.

Keywords: Convolutional Neural Network, Object detection, Image annotation

1. Introduction

Convolutional neural networks (CNNs) has been widely used in visual recognition from 2012 [1] due to its good performance in the image classification task. In [1], the authors show a significant improvement on the accuracy of image classification in ImageNet Large Scale Visual Recognition Challenge (ILSVRC). CNNs have become one of the most competitive choices for solving image classification challenges. Besides image classification, researchers also extend the application of CNNs to object detection [2].

The goal of object detection is to recognize multiple objects in a single image, not only to return the confidence of the class for each object, but also output the localization information. Among most of the works in object detection, the Region-based CNN (R-CNN) [2] has demonstrated promising results for this task. The region-based method generates the region proposals first, and then extracts a feature vector from each proposal using a single-label CNN. Finally, it classifies each region with category-specific linear classifier. Figure 1 presents an overview of this method. But, such algorithms have all focused on how to do the region extraction and output better bounding box, and it is not yet clear if CNN can directly learn the structure of an image.

In this work, we propose an experiment to evaluate CNN's ability to understand the structure of a given image. In our experiment, we obtained a Component (Sub-structure)-Chinese character database [3] as our dataset, and then employed a similar network structure to [4], which contains several convolutional and fully-connected layers as the basic architecture. The advantage of our Component-Chinese character dataset is that Chinese character's components have several typical patterns, which can be explained easier. For example, we can test the CNN's ability to correctly recognize the enclosed structure by its performance in detecting the components "冂", "口", "广" which are enclosed structure. In addition, such a detection work for line-based-structure Character images should be more representative than other detection problems. Sample Chinese character images for enclosed structure are shown in Fig. 2.

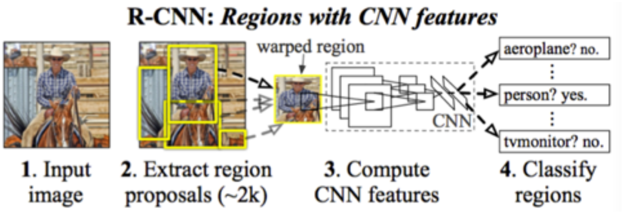

Fig. 1 Object detection system overview [2].

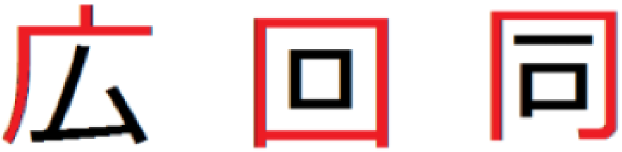

Fig. 2 Sample images of the component in enclosed structure.

Our study focuses on the questioning if the CNN can identify the given components. Since localization work is not necessary in our study, our research is essentially not a detection task but more like a multi-label annotation problem. In this paper, before the multi-label work, we first

[1] Kyushu Univerisity
[2] Tokyo University
[a)] 2IE15060K@s.kyushu-u.ac.jp
[b)] brian@human.ait.kyushu-u.ac.jp
[c)] kumiko.cl.rcast.u-tokyo.ac.jp
[d)] uchida@ait.kyushu-u.ac.jp

Tags: green, flower sun, flowers, zoo, day, sunny, sunshine
Tags: london, traffic, raw
Tags: art, girl, woman, wow, dance, jump, dancing

Fig. 3 Sample images of multi-label annotation problem [5].

propose an experiment with single-label CNN to ensure that CNN can at least recognize single structure of character image. The results show that CNNs can learn the single-component structure of images based on the raw pixels without region extraction. Through our experiment for detecting Chinese character component which have variable types of structures, we gain more information about how CNN recognize the image composition.

The remaining of this paper is organized as follows. In Section 2, we will briefly summarize the previous work in image annotation and introduce the multi-label approaches. Section 3 reviews CNNs and discusses their use. In Section 4, we will cover more details of the technical approach in our model including the dataset and the network architecture, then report a result for our experiment. Discussion and future work are drawn in Section 5.

2. Related work

In this section, We briefly discuss previous works on multi-label image annotation.

Many real-world classification problems involve multiple label classes. In multi-class classification, each sample can belong to one and only one label (one vs rest); whereas in multi-label classification, each sample can be associated with multiple labels. For example, in image annotation, a digital image often associated with multiple tags (Some sample images for multi-label annotation are shown in Figure 3).

Early work, such as [6], [7], applied machine translation methods to parse natural images and tried to establish a relationship between image regions and words. Recently, works on image tagging have mostly focused on nonparametric nearest-neighbor methods, which gain results that are more competitive. For example, [8] proposed a nearest-neighbor-based tag transfer approach, which achieved significant improvement over previous model-based methods. Recent improvements on the nonparametric approach include TagProp [9], which learns a discriminative metric for nearest neighbors to improve tagging.

3. Convolutional Neural Networks

CNNs are a powerful neural network that utilizes specific network structures, such as convolution and pooling layers, and have shown significant performance in image-related applications. Combined with recent method such as dropout layer and ReLU (rectified linear units), CNN models have outperformed existing handcrafted features. [1] reported record-breaking results on ILSVRC 2012 that contains 1000 visual-object categories. CNNs have also had success in many other fields in character [10] and digit recognition [11], [12]. However, such studies mostly focused on a single-label problem and the images in the dataset are only labeled by one class. A common approach that extends CNNs to multi-label classification is to transform it into multiple single-label classification problems, which can be trained with the ranking loss [5]. [5] comprehensively summarized the existing convolutional networks and loss function for the problem of multi-label image annotation and showed very detailed results for each method.

Since our experiment is based on CNNs, we will give a brief introduction on how CNNs work in image processing field. CNNs are very similar to multilayer perceptron (MLP) which receives an input and processes it by a series of hidden layers. Each hidden layer is made up of a set of neurons, where each neuron is fully connected to all neurons in the previous layer. CNNs contain three main types of layers: convolutional layer, pooling layer and fully-connected layer. Input will hold the raw pixel values of the image. The convolutional layer will calculate a dot product between their weights and a small region they are connected to in the input volume. Pooling layer will perform a downsampling operation along the spatial dimensions. The last fully-connected layer is called the "output layer" and in classification settings it represents the class scores. The parameters in the this layer will be trained with gradient descent so that the class scores that the neurall network computes are consistent with the labels in the training set for each image.

4. Experiment

Before the multi-label component detection work, we first propose an experiment with single-label CNN to test the ability of CNN to learn the single structure of character image. In Section 4.1, we give an introduction on our dataset. The CNN architecture we used is shown in Section 4.2. In Section 4.3, we discuss our experiment and its result in detail.

4.1 Dataset

4.1.1 The structure of Chinese characters

In this section, we give a brief introduction on the structure of Chinese characters.

The Chinese characters are written within the framework of a square and there are several basic structures. Characters are defined as being either basic characters or compound characters. Basic characters are not divisible. They contain only one component while compound characters contain two or more components. Less than 5% of all characters are basic and over 95% are compound. The structures in compound characters are usually easily identified, as most sides do not touch or intersect; they are separate. However, in some cases the sides connect, and in rare cases intersect. In some characters the sides are not evident and the character is comprised of three or more individual components. Table 1 lists the common character structures of modern Chinese characters.

Table 1 Types of character structures [13]

Description	Example Characters
Left to Right	好, 你, 吗, 他
Above to Below	主, 全, 分, 乔
Left to Middle and Right	辩, 班, 辙, 粥
Above to Middle and Below	复, 享, 兽, 养
Full Surround	囚, 回, 因, 叉
Surround from Above	冈, 闭, 咸, 凤
Surround from Below	凶, 凿, 鼎, 函
Surround from Left	匠, 区, 医, 匪
Surround from Upper Left	厘, 危, 友, 发
Surround from Upper Right	乌, 可, 包, 乃
Surround from Lower Left	勉, 处, 起, 建
Overlaid	坐, 农, 幽, 臾

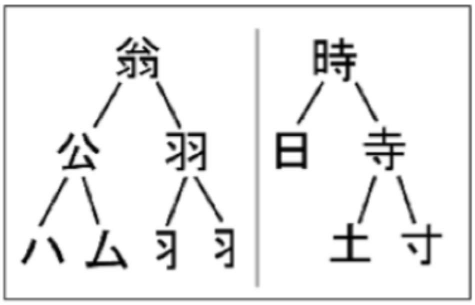

Fig. 4 Examples of component trees [3].

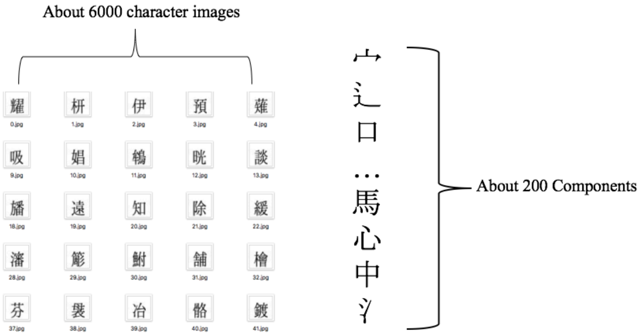

Fig. 5 Chinese characters dataset

4.1.2 Component-Chinese character database

Tanaka-Ishi and Godon [3] created a character-component database for Chinese characters. Every character in this database is represented as a component tree. Within this component tree, a node denotes a substructure of the character and its children nodes denote its components. Having the root node denote the whole character, the tree thus denotes a recursive decomposition into smaller components. Figure 4 shows an example for the characters "翁" and "時".

4.1.3 Training and test sets

Based on the Component-Chinese character database, we obtain a dataset, which contains about 6000 images of different characters that have been annotated, with several component-tags (2-10 on average) per image. We used a subset of 4000 images for training and used the rest of the images for testing. The component-tags dictionary for the images contains 200 different component-tags (Figure 5).

In this experiment, in order to test if a CNN has the ability to learn three typical types of Chinese character structures (Left to Right, Above to Below, Surround from Above), we choose three corresponding components "木", "心", and "冂", shown in Figure 6, as our test samples. Specially, since the component are sparsely distributed in all the characters, the scale of positive data and negative data in our dataset is extremely imbalanced. Table 2 shows the component scale for our test samples. Positive in Table 2 means the number of the charaters which contain the given component while the negative means the number which don't contain the component.

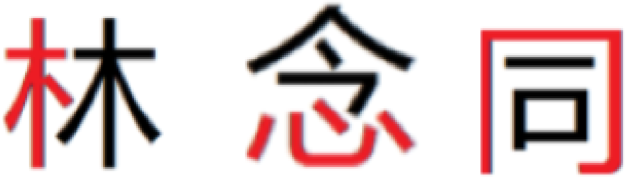

Fig. 6 Examples of testing components "木" "心" "冂"

4.2 CNN architecture

We use the open-source Caffe toolbox for our experiments. The basic architecture of the network, Fig. 7, that we used is similar to the one used in [4]. We use two convolutional layers and two fully connected layers. Before feeding the images to the convolutional layers, each images is resized to 32x32. Next, convolutional layers are set to squares of size 5 and max pooling layers are used after each convolutional layer to introduce spatial invariance. Each fully-connected layer is of size 1024. Dropout layers follow each of the fully-connected layers with a dropout ratio of 0.5. For all the layers, we used ReLU as our non-linear activation function.

4.3 Experiment methodology and results

The training process is as follows. We pose our single label experiment as a binary classification problem. We train separate CNNs for each component and if the character contains the given component, it should be labeled 1. Figure 8 shows an overview for our training process.

We computed the recall and precision for our three testing samples ("木", "心", and "冂") and the result are reported in Table 3. Our experiment achieved an extremely high precision rate for each component with our imbalance dataset (the scale of the positive data is very small, shown in Table 2). It indicates that the CNN can easily learn the structure. However, the recall for our result was relatively low. As precision = True positive matches / (True positive matches + false positive matches) and recall = True positive matches / (True positive matches + false negative matches), a high precision and a low recall show that there are many false negative matches in out result.

Table 4 lists three examples of false negative matches. It shows that our CNN is weak at discriminating similar components. The false negatives were reasonable, for example "冂" appears to be part of "喃", "内", "莴", however these characters contain different but similar components.

Table 2 Component scale for test samples

Component name	Samples of character		Training set (4000)		Valuation set (2000)		Component scale in total (6000)	
(200 in total)	positive	negative	positive	negative	positive	negative	positive	negative
心	忍意总愿	案居新内	106	3894	63	1937	169	5831
冂	冉内册冈	本政郵日	157	3843	87	1913	244	5756
木	格板枝柱	品礼状連	512	3488	301	1699	813	5187
氵	温派滩洒	必要賞品	231	3769	138	1862	369	5631
口	口中叶古	能文通急	1122	2878	663	1337	1785	4215
广	应庞店庙	便利特典	80	3920	38	1962	118	4215
火	烤炖燎烩	疑問身近	287	3713	120	1880	407	5593

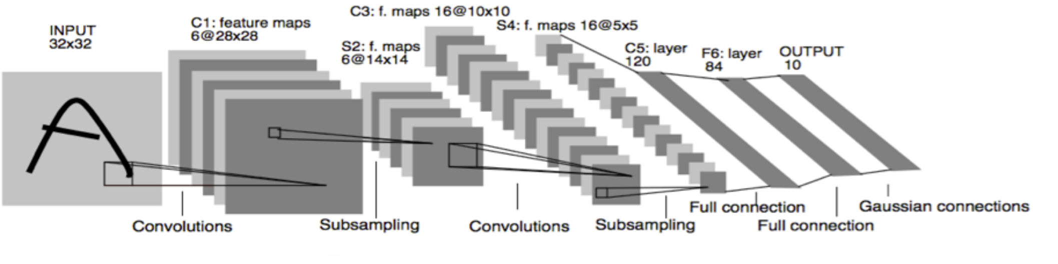

Fig. 7 CNN architecture in [4].

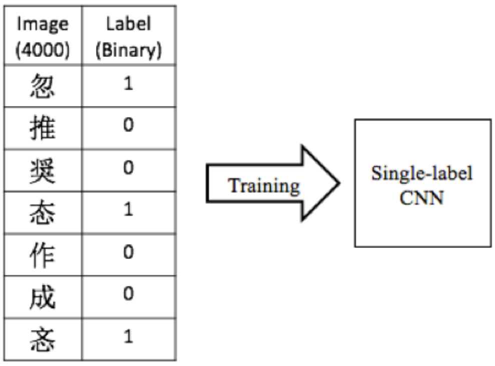

Image (4000)	Label (Binary)
忽	1
推	0
奨	0
态	1
作	0
成	0
忞	1

Fig. 8 Training process for the component "心"

Table 3 Result on single-component experiment.

Component name	Precision	Recall
木	95%	80%
心	99%	91%
冂	99%	65%
氵	99%	95%
口	92%	86%
广	97%	92%
火	98%	90%

Another component “虍” is often recognized to “心” and ” 冫” is often recognized to ” 氵”. Since the component ” 木” plays a sub-structure role in many other components, such as ” 東” and ” 束”, our CNNs model output many false negative matches. It seems that CNNs haven’t learnt that both of the component ” 木” and ” 口” is an individual structure but not the one connected with other parts.

5. Conclusion and future work

In this paper, we propose a single-label convolutional neural network based experiment to test the CNN’s ability to recognize the component of Characters directly. Our approach is inherently “structure detection from raw pixels”, where we utilize the character’s component as our dataset to describe this problem in an easier way. Our experiment indicates that simple CNNs (only containing two convolutional layers) can also learn the single-structure of character image, but it also shows a poor performance in discriminating between the similar components. As further work, we plan to adopt the loss function proposed in [5] to change our single-label CNNs into a multi-label one and do more evaluation work on multi-label Component-Chinese character dataset.

Table 4 Examples of false negative

Component	Samples
木	襴 陳 籟
心	瀘 戯
冂	喃 内 蒿

References

[1] Krizhevsky, A., Sutskever, I. and Hinton, G. E.: Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, pp. 1097–1105 (2012).

[2] Girshick, R., Donahue, J., Darrell, T. and Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587 (2014).

[3] Tanaka-Ishii, K. and Godon, J.: Kansuke: A logograph lookup interface based on a few modified stroke prototypes, *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol. 16, No. 2, p. 11 (2009).

[4] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324 (1998).

[5] Gong, Y., Jia, Y., Leung, T., Toshev, A. and Ioffe, S.: Deep convolutional ranking for multilabel image annotation, *arXiv preprint arXiv:1312.4894* (2013).

[6] Duygulu, P., Barnard, K., de Freitas, J. F. and Forsyth, D. A.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, *European conference on computer vision*, Springer, pp. 97–112 (2002).

[7] Barnard, K. and Forsyth, D.: Learning the semantics of words and pictures, *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Vol. 2, IEEE, pp. 408–415 (2001).

[8] Makadia, A., Pavlovic, V. and Kumar, S.: A new baseline for

image annotation, *European conference on computer vision*, Springer, pp. 316–329 (2008).

[9] Guillaumin, M., Mensink, T., Verbeek, J. and Schmid, C.: Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation, *2009 IEEE 12th international conference on computer vision*, IEEE, pp. 309–316 (2009).

[10] Uchida, S., Ide, S., Iwana, B. and Zhu, A.: A Further Step to Perfect Accuracy by Training CNN with Larger Data, *International Conference on Frontiers of Handwriting Recognition* (2016).

[11] Ciregan, D., Meier, U. and Schmidhuber, J.: Multi-column deep neural networks for image classification, *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, IEEE, pp. 3642–3649 (2012).

[12] Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S. and Shet, V.: Multi-digit number recognition from street view imagery using deep convolutional neural networks, *arXiv preprint arXiv:1312.6082* (2013).

[13] ArchChinese.com: Arch Chinese - Chinese Character Structure, `http://www.archchinese.com/arch_character_structure.html`.