Spatial Pyramid Block for Oracle Bone Inscription Detection

Guoying Liu
School of Computer and information Engineering, Anyang Normal University
455000
guoying.liu@aynu.edu.cn

Jici Xing †
Key Laboratory of Oracle Bone Inscriptions Information Processing, Ministry of Education
455000
njicixing@gmail.com

Jing Xiong
School of Computer and information Engineering,
Anyang Normal University
455000
jingxiong@aynu.edu.cn

ABSTRACT
The detection of Oracle Bone Inscription (OBI) is one of the most fundamental aspects of oracle bone morphology. However, the detection method depending on experts' experience requires long-term learning and accumulation for professional knowledge. This paper investigated the performance of the deep-learning-based object detection framework in the OBI dataset, then selected the one with the best performance as the baseline and made a series of optimization. Specifically, we first redesigned the sizes and ratios of the anchor box according to the data characteristics by using K- means clustering. Secondly, we extracted some typical noises from OBI for data augmentation. Finally, Focal Loss and Mixed-precision are used to improve the model precision and compress the memory footprint. To further improve the performance, the Spatial Pyramid Block is proposed, which can stabilize features and suppress noise interference. Experiments on our OBI benchmarks validate the superiority of the proposed method that achieves 82.1% F-measure suppressing several mainstream object detectors. Our dataset and algorithms will soon be available at http://jgw.aynu.edu.cn.

CCS Concepts
• Software and its engineering • Software creation and management • Designing Software

Keywords
Oracle; Bone; Inscription; Detection

1. INTRODUCTION
Oracle Bone Inscription (OBI) is one of the oldest characters of Chinese words, which are hieroglyphic signs inscribed onto cattle bones or turtle shells with sharp objects about 3000 years ago. OBI is an important medium for exploiting the political systems, economic status, and social lives of the Shang Dynasty (about 1600 B.C. -1046 B.C.). However, few people have the literacy of OBIs. The detection and recognition of OBIs, which combines archaeology, history, philology, and literature, requires people to have plenty of knowledge and years of experience.

As shown in Fig. 1, some OBI examples are depicted. It is easy to find that each character looks like an undirected graph composed of intersections, lines, or curves. Therefore, many researchers attempted to recognize OBIs by making use of graph theory. By cascading two-level graph coding and one-level endpoint feature coding, Feng Li and Xinlun Zhou proposed to automatically recognize off-line OBIs [1], [2]. Shaotong Gu [3] described OBIs by topological coding and recognized them based on topological registration. Qingsheng Li [4] coded each OBI by the adjacent inverse matrix and recognized OBIs based on the theory of graph isomorphism.

Figure 1: OBI Image Samples

Some other researchers tried to recognize OBIs by different kinds of character features. Xiaoqing Lv etal. [5] used the Fourier descriptor of curvature histogram to describe OBIs, Lin Meng et al. [6]-[8] employed Hough Transform to extract line features of OBIs, While Feifei Feng et al. [9] adopted Mathematical Morphology to extract character features. In [10], Jun Guo et al. proposed a hierarchical representation for OBIs, which combines a Gabor-related low-level representation and a sparse-encoder- related mid-level representation.

To accurately identify the OBIs, first of all, we need to detect their location. However, these efforts mentioned above are aiming at solving the problem of the classification for each character on OBIs. Few studies have been found to perform the detection task.

Inspired by object detection and scene text detection in the field of deep learning, we regard OBI as a special kind of data objects and their detection task can be performed by resorting to some popular methods [11]-[15], but directly migrating these methods to OBI datasets presents some problems that can lead to performance degradation. Compared with the usual data, OBI detection has the following difficulties:

ICSCA 2020, February 18–21, 2020, Langkawi, Malaysia

ACM ISBN 978-1-4503-7665-5/20/02…$15.00
https://doi.org/10.1145/3384544.3384561

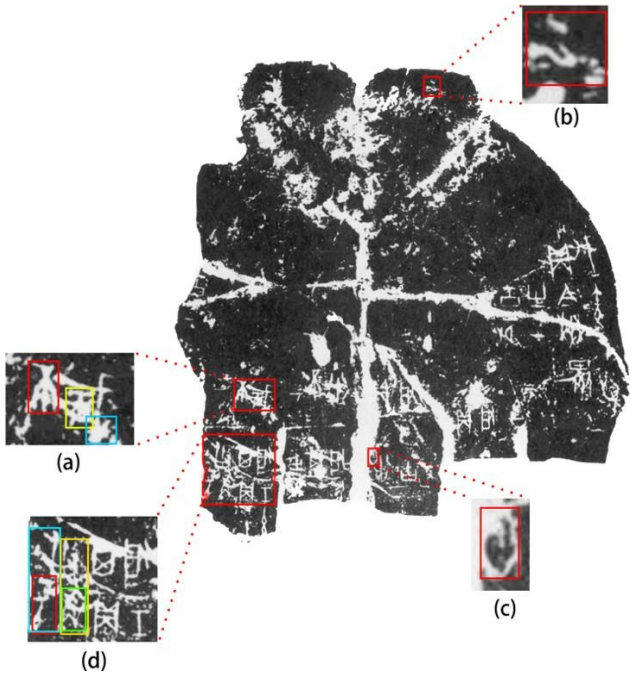

Figure 2: difficulties of OBIs detection. (a) independent characters linked together by nick and noise. (b) Noise points bringing many false positives. (c) unlabeled may be classified as true positive. (d) Densely distributed patterns causing secondary overlaps.

1. It is common to connect two independent characters linked together by nick and noise (see Fig.2 (a)).

2. Noise points caused by corrosion or excavation brings many false positives (see Fig.2 (b)).

3. Incomplete characters which were not labeled may be classified as true positive (see Fig.2 (c)).

4. Densely distributed patterns may cause a secondary overlap dilemma (see Fig.2 (d)).

In this paper, we compare the performance of the popular object detection algorithms in the OBI dataset and chose the one with the best performance as our baseline. Moreover, a series of improvements are introduced according to the features of OBI data. To be specific, we use the K-means++ algorithm [16] to cluster anchor boxes, making it closer to the original size. Then, a variety of synesthetic noises are inserted to augment the training data simulating the real OBI images interference. Finally, Focal Loss [17] and mixed-precision [18] are utilized to enhance model accuracy while compressing the memory footprint. Moreover, we proposed a spatial pooling block to enrich feature maps while suppressing the noise disturbance, as illuminated in Fin.3. Compared with the original SPP layer [19], average pooling layers are utilized to form a refined creation. Experimental results have shown the superiority of our method, which raises the F-measure to 82.13%.

The contribution of this paper mainly includes two folds:

1. A strong baseline is tailored based on the features of OBI data, the performance of which achieves state-of-the-art.
2. The SPP block is designed that can further suppress the noise and improve the detection performance raising 2 points on F-measure.

2. A BRIEF REVIEW OF OBJECT DETECTION

2.1 Method Before the Deep Learning Era

Before the deep learning era, most detection methods [20] adopt Connected Components Analysis [21]- [26] or Sliding window [27]-[30] based methods. Connected Components Analysis-based methods firstly extract candidate components through a variety of ways (e.g., color clustering or extreme region extraction), and then filter out non-text components using manually designed rules or classifiers automatically trained on hand-crafted features. In sliding window classification methods, windows of varying sizes slide over the input image, where each window is classified as text regions or not. Those classified as positive are further grouped into text regions by morphological operations [28], Conditional Random Field [29], and other alternative graph-based methods [27, 30].

2.2 Object Detection in Deep Learning

Motivated by the thriving of deep learning-based object [31] or text [20] detection architectures, we thought that oracle characters as a particular object could get benefits from these fields. There are two main trends in the field of object detection: two-stage and one- stage.

The two-stage approach consists of two parts, where the former (e.g., Selective Search [32], Edge-Boxes [33], DeepMask [34] [35], RPN [11]) generates a sparse set of candidate object proposals, and the latter determines the accurate object regions. Notably, the two-stage approach (e.g. R- CNN [36], SPPnet [37], Fast R-CNN [38], Faster R-CNN [11]) achieved dominated performance on several challenging datasets (e.g. PASCAL VOC 2012 [39] and MS COCO [40]). After that, numerous effective techniques were proposed to improve the performance, such as architecture diagram [41, 42, 43], training strategy [44, 45], contextual reasoning [46, 47, 48, 49] and multiple layers exploiting [50, 51, 52, 53].

The one-stage approach considers high efficiency, which attracts much more attention recently. Sermanet et al. [54] present the OverFeat method for classification, localization, and detection based on deep convolutional networks, which is trained end-to-end from raw pixels to ultimate categories. Redmon et al. [55] use a single feed-forward convolutional network to directly predict object classes and locations, called YOLO, which is extremely fast. After that, YOLOv2 [56] is proposed to improve YOLO in several aspects, add batch normalization, high-resolution classifier, and anchor boxes. Liu et al. [12] propose the SSD method, which spreads out anchors of different scales to multiple layers within a convolution network and enforces each layer to focus on predicting objects of a specific scale. DSSD [57] introduces additional context into SSD via deconvolution operation to improve accuracy. DSOD [58] designs an efficient framework and a set of principles to learn object detectors from scratch, following the network structure of SSD. To improve accuracy, some one-stage methods [41] [42] aim to address the extreme class imbalance problem by re- designing the loss function or classification strategies. Although the one-stage detectors have made good progress, their accuracy still trails that of two-stage methods.

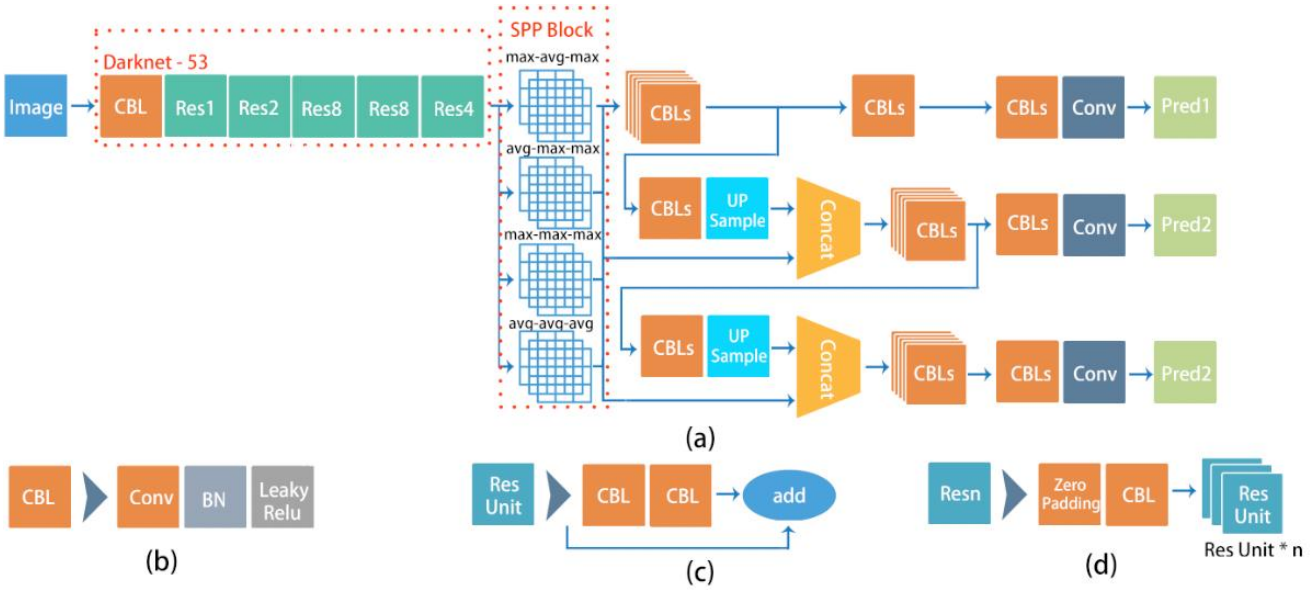

Figure 3: (a) is the mainframe, (b) is a combination of Convolution, BatchNorm and Leaky Relu in a. c is the ResUnit. D is the ResBlock.

3. CHARACTER DETECTION IN OBI

3.1 Baseline

Our baseline is based on the method with the best overall performance (Sec 4.2) in the experiment architectures. Then, a series of improvements will be performed according to the characteristics of the data OBI data.

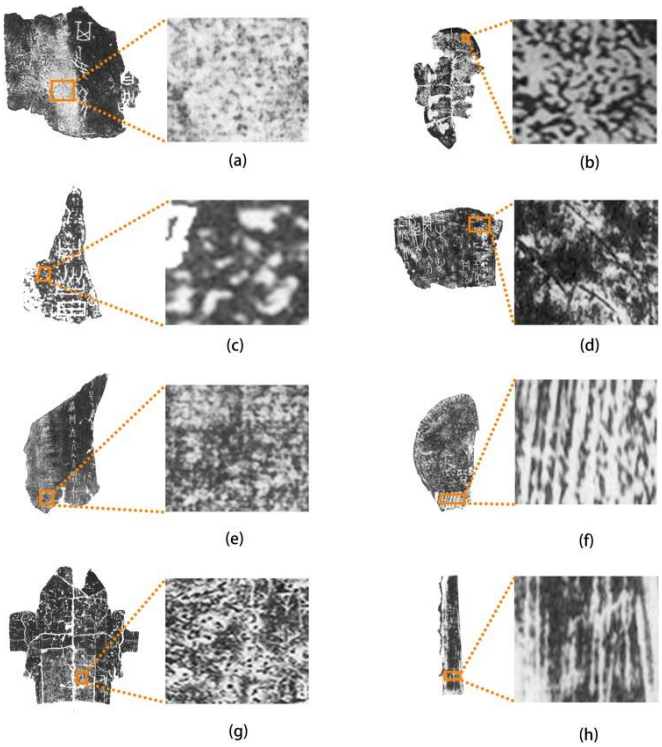

Figure 4: Noises extracted from OBIs (a) and (g) are noise points, (b), (c), and (e) are texture, (d), (f), and (h) are nick.

Data Augmentation

To enhance the generalization ability in various scales and aspect ratios, we apply data augmentation to enlarge the dataset:

1. Images are randomly rotated and cropped with areas ranging from 0.24 to 1.69 and aspect ratios ranging from 0.33 to 3. After that, Gauss and Poisson noise, blur, and lightness are randomly adjusted. We ensure that the OBI on the augmented images are still legible if they are legible before augmentation.
2. Several kinds of noises are extracted from the OBI (see Fig.4), which will be reproduced and added to the training samples.

Anchor

The anchor can be quantified by three parameters: the base scale of anchors, the feature maps where anchors searched, and the aspect ratios of anchors.

The detail of generating aspect ratios is as follows: analyze the data-augmented ground truth bounding box's size, then use the k-means++ algorithm to cluster sizes (width and height). Finally, the center point of the front K cluster serves as the sizes of anchors.

Focal Loss

We adopt Focal Loss to learn hard samples. Followed by [59] sorting the samples and then reweight them to update the network.

Mixed Precision

We use apex [60] to reduce the training requirements, which compresses memory bandwidth and storage requirements by 2X. Bandwidth-bound operations can be realized up to 2X training speedup.

Although our baseline achieves remarkable performance, it still has the same drawbacks as follows:

1. The multi-scale prediction of experiment architectures focuses on concatenating the global features of multi-scale convolutional layers while ignores the fusion of multi-scale local region features on the same convolutional layer.
2. Noise may affect the detection accuracy, which breaks down the structure of characters losing the shape, pattern, texture

information.

These problems motivate us to build a detector that can overcome them. The pooling layer can perform latitude reduction on the extracted results of the convolution layer, adjusting the size of the receptive field for the next convolution, controlling the resolution and channel, and preparing for the operation of the next layer. The goal of the pooling operation is to reduce the overfitting, retain the useful transmission, and remove the noise disturbance. There are two widely used pooling functions: average-pooling and max- pooling. The previous one is calculated by the following equation：

$$O = \max_{i \in I} \alpha \; for \; \alpha \; in \; \{\alpha_{x+\delta,\; y+\delta}\} \qquad (1)$$

where I and O represent the input and output feature map respectively.

α is the neighboring area of a pixel in the feature map. x, y denotes the corresponding coordinate and depicts the neighboring range. On the contrary, average-pooling calculates the average value for the feature points in the area, and retains more background information:

$$O = \frac{1}{|I|} \sum_{i \in I} \alpha \; for \; \alpha \; in \; \{\alpha_{x+\delta,\; y+\delta}\} \quad (2)$$

The symbols in Equation.2 corresponds to the meaning of the max pooling.

Inspired by [61], which utilizes the special pyramid pooling layer [37] to suppress noise and gather features, we recombined the pooling layer to form a series of SPP blocks:

$$Concat(Pooling_{max/3},\; Pooling_{avg/2},\; Pooling_{max/1}) \quad (3)$$
$$Concat(Pooling_{avg/3},\; Pooling_{max/2},\; Pooling_{avg/1}) \quad (4)$$
$$Concat(Pooling_{max/3},\; Pooling_{max/2},\; Pooling_{max/1}) \quad (5)$$
$$Concat(Pooling_{avg/3},\; Pooling_{avg/2},\; Pooling_{avg/1}) \quad (6)$$

Three different pooling layers are combined to form different SPP blocks. Where max and avg represent the type of pooling, and $/n$ represents the proportion of reduction of the feature graph after pooling.

Different from [61], we moved the feature pooling layer forward and adopted the multi-branch prediction similar to [37]. The pipeline and proposed SPP block are shown in Fig.3 and Fig.5 respectively. In the SPP stage, the feature maps are pooled in different scales then combined to supplement the features. The size of the pooled feature map follows the Equation:

$size_{pool} = size_{fmap} // size_{ni}$ (7) where $Size_{pool}$ represents the size of the sliding windows.

$size_{fmap}$ is the size of the feature map. Let $n_{i=1,2,3}$ represents the proportion of reduction of the feature graph after pooling.The stride of pooling is all 1, and the padding is utilized to keep aconstant size of the output feature maps, then we get three feature maps with the size of $Size_{fmap} \times Size_{fmap} \times channel$. These spatial pooling layers share the same input and concatenate the output according to the following equation:

$$Pool_i = Pooling_{avg/max}(F_{Input}) \qquad (8)$$
$$Pad_i = Padding(Pool_i, C) \qquad (9)$$
$$BN_i = BatchNormReLU(Pad_i) \qquad (10)$$
$$F_{Output} Concatenate_{i=1}^{n} BN_i \qquad (11)$$

Where n is the total number of padding layers. Finally, they are combined to form the SPP block.

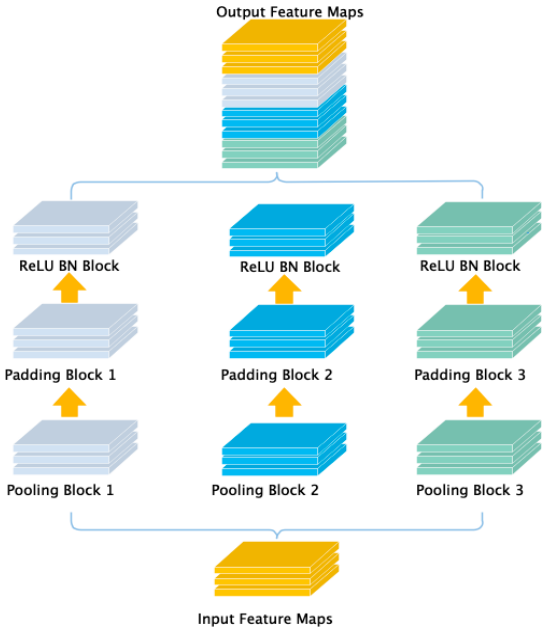

Figure 5: SPP layer

4. EXPERIMENT

4.1 Dataset

In this section, we discuss in detail the properties of OBI and how it differs from standard benchmarks.

Noise: After being buried in the ground for more than 3,000 years, the characters on the OBI became blurred after long- term corrosion. As shown in Fig. 6 (a) this image was Dirst printed by rubbing the OBI, then bound into a book, and Dinally sampled by a high-resolution scanner, which had a lot of texture noise and cracks. These naturally occurring disturbances are irregular and difDicult to reproduce through traditional data augmentation.

Fragment: The pieces of OBI are easy to be broken when unearthed, so a large number of OBI fragments were produced. As illustrated in Fig.6 (b), the broken characters often appeared on the edge of the fragments, which were very similar to the natural texture of the corresponding complete ones, making it extremely difficult to detect and identify them.

Variant: The OBI characters appeared in the Yin and Shang dynasties when there was no uniform writing standard (see Fig. 6(c)). Besides, the Shang and Zhou dynasties (about 1600 B.C. – 256 B.C.) spanned more than 1300 years. The evolution of the characters is noticeable, leading to a large number of variant characters in OBI. Statistically, there are 1,032 sets of variants in a total of 3,085 characters, accounting for 49.5 percent. The appearance of variant characters in OBI is very frequent, which brings great difficulties to the detection and recognition.

Distribution: The characters on the same piece of OBI vary in size, direction, and distribution (see Fig. 6(d)), significantly increasing the difficulty of detection and recognition. In the 56,743 oracle bone inscriptions, there are 1,425 words. Among them, there are 366 common characters, 500 not usually used, and 559 rare.

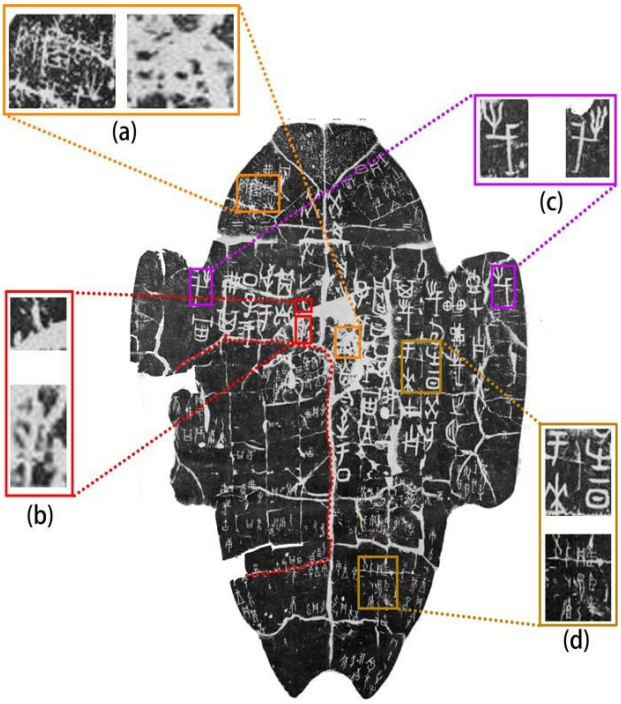

Figure 6: OBI images. (a)Noise. (b)Fragment. (c)Variant. (d)Distribution.

Dataset Generation: We have collected 9500 OBI rubbings (up to now) from by a high-resolution scanner then labeled every character with the upper left and lower right coordinates. Notably, this work only focuses on the detection task, so the number of classes in this dataset are all regarded as one. Our data will be updated and published at http://jgw.aynu.edu.cn.

4.2 Experimental architecture

We selected five mature object detection frameworks as experimental objects: Faster R-CNN, SSD, YOLOv3, RFBnet, and RefineDet. Their performance is summarized in Table.1. We find that the YOLOv3 outperforms all the other networks. Therefore, it will be modified following Sec 3.1 as our baseline. Interestingly, we also found that YOLOv3 is also widely used in other fields, including video game detection [62], license plate detection [63], and 3d detection [64], all of which have excellent performance. That may be due to the sophisticated design of DarkNet and the generalization capabilities of the entire processing mechanism. YOLOv3 runs faster than the others on images because it is simpler in structure and enjoy on-stage fashion (without RPN head). Notably, we only give a fixed number of training epochs and use the default hyperparameters except the data related configuration, which may not reach the limits of the algorithm on the OBI dataset. The performance of our baseline and the proposed method is listed in Table 3.

Table 1. Evaluations on Experiment Architecture

Method	F	P	R	Speed	Memory
FRCNN	0.766	0.754	0.778	3FPS	1714MB
SSD	0.753	0.748	0.758	9FPS	**668MB**
YOLOv3	**0.78**	**0.776**	0.784	**17FPS**	828MB
RefDet	0.778	0.752	**0.805**	14FPS	1451MB
RFBnet	0.775	0.761	0.789	15FPS	991MB

Where the baseline is as described in Sec3.1. MMM indicates that 3 max pools are used in the space pooling layer, while AAA means average pools; MAM means the layer is composed of Max pool, average pools, max pools, AMA means average pools, Max pools, and average pools. ALL is that all the above four combinations are employed, as shown in Fig. 2. Some typical results are shown in Fig. 7.

4.3 Ablation Study

To verify the effectiveness of our baseline and method, we do a series of comparative experiments on the collected OBI dataset as described in Tabel.2.

Table 2. Ablation of Our Baseline

Clustered Box	Data Augmentation	Focal loss	Mixed Precision	F
√				0.785
√	√			0.794
√	√	√		0.803
√	√	√	√	0.803

The Influence of Clustered Anchor: We study the effect of the use of clustered anchor by training a K-means++ cluster and set K (number of clusters) gradually from 6 to 11 with Euclidean distance. However, some outliners cannot be well clustered, and large K will slow down the model convergence. To keep a good balance of performance and speed, we keep k to 9 with corresponding centers ([48,50], [91,105], [159,238], [192,112], [288,409], [339,203], [464,698], [577,395], [973,911]) by default in the following experiments.

The Influence of Data Augmentation: Data augmentation has proven to play an essential role in various tasks in the field of computer vision, especially when data is insufficient. Mirroring, rotation, and scaling have been widely used in object detection, so this part mainly displays the synthesis of noise for augmentation. There was a 30 percent chance that any one of these noise images (see Fig.4) would be superimposed on the original data during training.

We make a comparison between the model with data augmentation and without that. Compared to the model without data augmentation, the model with data augmentation can make about 0.9% improvement on F-measure.

Table 3. our baseline and proposed method

Method	F	P	R	Speed	Memory
BaseLine	0.803	0.793	0.813	17FPS	**890MB**
BaseLine AAA	0.808	0.800	0.816	14FPS	951MB
BaseLine +MMM	0.811	0.795	0.827	17FPS	943MB
BaseLine +MAM	0.813	0.798	0.828	16FPS	930MB
BaseLine +AMA	0.810	0.793	0.827	15FPS	940MB
BaseLIne +ALL	0.821	0.812	0.830	12FPS	1342MB

The Influence of Focal Loss: To investigate the effectiveness of focal loss, we firstly replace the loss function with the traditional cross-entropy loss to make the final prediction. The F-measure drops 0.6% when the Focal loss is removed, which indicates that the imbalance of positive and negative sample proportion also has adverse effects on OBIs single category data.

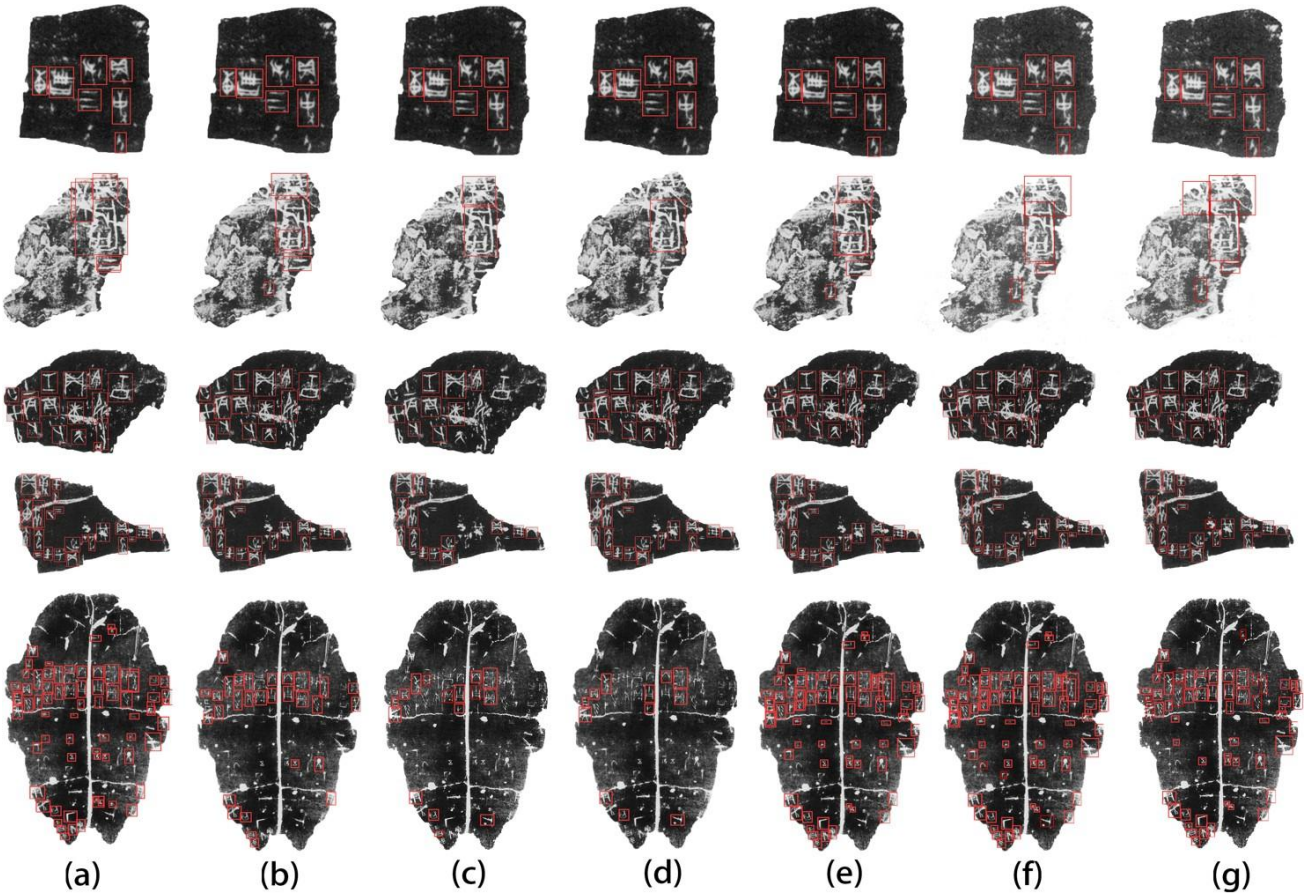

Figure 7: Detection Result. (a) Faster R-CNN, (b) SSD, (c)RFBnet, (d)RefineDet, (e)Yolov3, (f)Baseline, (g)Baseline+ALL

The Influence of Mixed-Precision: We study the validity of Mixed-precision by using Nvidia apex tools. Specifically, we set up the network and optimizer to prioritize Float16 as the default data type of tensors. Compared with the method using mixed- precision (see Table.4), F-measure was not affected at all, but the memory occupancy and training time were reduced by nearly 50%, which indicates the effectiveness of mixed-precision.

Table 4. our baseline and proposed method

Batch:32 Scale:416	Without Mixed-Precision	With Mixed-Precision
Training Memory	8.23GB	3.26GB
Training Time	14H	6H

The Influence of SPP Block: As shown in Table 2, when the SPP layers are inserted, the detection performance is improved to different degrees. We tried different forms of combination where Max pool, average poo, Max pool combination makes smaller computation and memory overhead and can get a 1.2% F-measure increase. When the SPP block is used (ALL), the F-measure increases by 2 points., but reduce the speed of 5 FPS and have about 400 MB of memory overhead.

5. CONCLUSION

In this paper, we investigate the performance of popular object detection architecture on OBIs dataset. From the experiment, YOLOv3 performed well in all tests, with higher precision, faster speed, less overlap, and sensitivity to small targets and tolerance noise.

Moreover, we tailor-make a YOLOv3 baseline which has reached 80.1 F-measure. Finally, SPP block is proposed to further improve detection performance. Although our method has made some progress in the field of OBI detection, there are still some issues needed to be further studied. For example, some OBIs have too few samples to train an accurate detector which can avoid overfitting, resulting in serious degradation with these OBIs. The study of all of this issue is left as our future work.

6. ACKOWLEDGMENTS

This work is supported by the joint fund of National Natural Science Foundation of China (NSFC) and Henan Province of China under Grant U1804153, partly supported by the Program for Changjiang Scholars and Innovative Research Team in University (PCSIRT) under Grant 2017PT35, and partly supported by the Program of Innovative Research Team (in Science and Technology) in University of Henan Province of China under Grant 17IRTSTHN012.

7. REFERENCES

[1] Li, F., & Zhou, X. (1996). Recognition of Jia Gu Wen based on Graph theory. Journal of Electronics, 18(suppl.), 41-[2] Zhou, X., Li, F., Hua, X., & Wei, J. (1996). A method of Jia

[2] Gu Wen recognition based on a two-level classification. Journal of Fudan University (Natural Science), 35(5), 481-486.

[3] Meng, L., & Izumi, T. (2017). A combined recognition system for oracle bone inscriptions. Int. J. Mechatron. Syst., 7(4), 235-244.

[4] Li, Q., Yang, Y., & Wang, A. (2011). Recognition of inscriptions on bones or tortoise shells based on graph isomorphism. Computer Engineering and Applications, 47(8), 112-114.

[5] Lv, X., Li, M., Cai, K., Wang, X., & Tang, Y. (2010). A graphic-based method for Chinese oracle-bone classification. Journal of Beijing Information Science and Technology University, 25(Z2), 92-96.

[6] Meng, L. (2017). Two-stage recognition for oracle bone inscriptions. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10485, 672-682.

[7] Meng, L., & Izumi, T. (2017). A combined recognition system for oracle bone inscriptions. Int. J. Mechatron. Syst., 7(4), 235-244.

[8] Meng, L. (2017). Recognition of oracle bone inscriptions by extracting line features on image processing. Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods (pp. 606-611).

[9] Feng, G., Gu, S., & Yang, Y. (2013). Feature extraction method of Oracle-bone inscriptions based on mathematical morphology. Journal of Chinese Information Processing, 27(2), 79-85.

[10] Guo, C., Wang, E., Roman, R., Chao, H., & Rui, Y. (2016). Building hierarchical representations for oracle character and sketch recognition. IEEE Trans. Image Process., 25(1), 104-118.

[11] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. (2015)

[12] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision, Springer (2016) 21 – 37

[13] Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)

[14] Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 4203 – 4212

[15] Liu, S., Huang, D., et al.: Receptive field block net for accurate and fast object de-tection. In: Proceedings of the European Conference on Computer Vision (ECCV). (2018) 385 – 400

[16] Arthur, D., and Vassilvitskii, S. 2007. k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM sympo- sium on Discrete algorithms, 1027–1035. Society for Indus- trial and Applied Mathematics.58. Guo, C., Wang, E., Roman, R., Chao, H., & Rui, Y. (2016). Building hierarchical representations for oracle character and sketch recognition. IEEE Trans. Image Process., 25(1), 104-118.

[17] Lin T Y, Goyal P, Girshick R, et al. Focal Loss for Dense Object Detection[J]. IEEE Transactions on Pattern Analysis

[18] Bordes, J.; Maher, M.; and Sechrest, M. 2009. Nvidia apex: High definition physics with clothing and vegetation. In Game Developers Confer- ence.

[19] He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence 37 (2015) 1904 – 1916

[20] Long, S., He, X., Ya, C.: Scene text detection and recognition: The deep learning era. arXiv preprint arXiv:1811.04256 (2018)

[21] Epshtein, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE (2010) 2963 – 2970

[22] Huang, W., Lin, Z., Yang, J., Wang, J.: Text localization in natural images using stroke feature transform and text covariance descriptors. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 1241 – 1248

[23] Jain, A.K., Yu, B.: Automatic text location in images and video frames. Pattern recognition 31 (1998) 2055 – 2076

[24] Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In: 2012 IEEEConference on Computer Vision and Pattern Recognition, IEEE (2012) 1083 – 1090

[25] Yi, C., Tian, Y.: Text string detection from natural scenes by structure-based partition and grouping. IEEE Transactions on Image Processing 20 (2011) 2594 – 2605

[26] Yin, X.C., Yin, X., Huang, K., Hao, H.W.: Robust text detection in natural scene images. IEEE transactions on pattern analysis and machine intelligence 36 (2014) 970 – 983

[27] Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., Wu, D.J., Ng, A.Y.: Text detection and character recognition in scene images with unsuper-vised feature learning. In: ICDAR. Volume 11. (2011) 440 – 445

[28] Lee, J.J., Lee, P.H., Lee, S.W., Yuille, A., Koch, C.: Adaboost for text detection in natural scene. In: 2011 International Conference on Document Analysis and Recognition, IEEE (2011) 429 – 434

[29] Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International Conference on Computer Vision, IEEE (2011) 1457 – 1464

[30] Wang, T., Wu, D.J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural networks. In: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), IEEE (2012) 3304 – 3308

[31] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2017) 7310 – 7311

[32] Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision 104 (2013) 154 – 171

[33] Zitnick, C.L., Doll´ar, P.: Edge boxes: Locating object proposals from edges. In: European conference on computer vision, Springer (2014) 391 – 405

[34] Pinheiro, P.O., Collobert, R., Doll´ar, P.: Learning to segment object candidates. In: Advances in Neural Information Processing Systems. (2015) 1990 – 1998

[35] Pinheiro, P.O., Lin, T.Y., Collobert, R., Doll´ar, P.: Learning to refine object segments. In: European Conference on Computer Vision, Springer (2016) 75 – 91

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

[37] He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence 37 (2015) 1904 – 1916

[38] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. (2015) 1440 – 1448

[39] Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision 111 (2015) 98 – 136

[40] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll´ar, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision, Springer (2014) 740 – 755

[41] Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems. (2016) 379 – 387

[42] Lee, H., Eum, S., Kwon, H.: Me r-cnn: Multi-expert r- cnn for object detection. arXiv preprint arXiv:1704.01069 (2017)

[43] Zhu, Y., Zhao, C., Wang, J., Zhao, X., Wu, Y., Lu, H.: Couplenet: Coupling global structure with local parts for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. (2017) 4126 – 4134

[44] Shrivastava, A., Gupta, A., Girshick, R.: Training region-based object detectors with online hard example mining. In: IEEE Conference on Computer Vision Pat-tern Recognition. (2016)

[47] Wang, X., Shrivastava, A., Gupta, A.: A-fast-rcnn: Hard positive generation via adversary for object detection. (2017)

[48] Bell, S., Lawrence Zitnick, C., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 2874 – 2883

[49] Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware cnn model. In: Proceedings of the IEEE International Confer-ence on Computer Vision. (2015) 1134 – 1142

[50] Shrivastava, A., Gupta, A.: Contextual priming and feedback for faster r-cnn. In: European Conference on Computer Vision, Springer (2016) 330 – 348

[51] Zeng, X., Ouyang, W., Yang, B., Yan, J., Wang, X.: Gated bi-directional cnn for object detection. In: European Conference on Computer Vision, Springer (2016) 354 – 369

[52] Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep con-volutional neural network for fast object detection. In: European conference on computer vision, Springer (2016) 354 – 370

[53] Kong, T., Sun, F., Yao, A., Liu, H., Lu, M., Chen, Y.: Ron: Reverse connection with objectness prior networks for object detection. (2017)

[54] Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: IEEE Conference on Computer Vision Pattern Recognition. (2017)

[55] Shrivastava, A., Sukthankar, R., Malik, J., Gupta, A.: Beyond skip connections: Top-down modulation for object detection. (2016)

[56] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., Lecun, Y.: Overfeat: Integrated recognition, localization and detection using convolutional networks. Eprint Arxiv (2013)

[57] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. (2015)

[58] Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. In:IEEE Conference on Computer Vision Pattern Recognition. (2017)

[59] Fu, C.Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd : Deconvolutional single shot detector. (2017)

[60] Shen, Z., Zhuang, L., Li, J., Jiang, Y.G., Chen, Y., Xue, X.: Dsod: Learning deeply supervised object detectors from scratch. (2017)

[61] Paszke A, Gross S, Chintala S, et al. Automatic differentiation in pytorch[J]. 2017.

[62] Yao, W.; Sun, Z.; and Chen, X. 2019. Understanding video content: Efficient hero detec- tion and recognition for the game" honor of kings". arXiv preprint arXiv:1907.07854.

[63] Laroca, R.; Zanlorensi, L. A.; Gonc¸alves, G. R.; Todt, E.; Schwartz, W. R.; and Menotti, D. 2019. An efficient and layout-independent automatic license plate recognition system based on the yolo detector. arXiv preprint arXiv:1909.01754.

[64] Simon M, Amende K, Kraus A, et al. Complexer-YOLO: Real-Time 3D Object Detection and Tracking on Semantic Point Clouds[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2019: 0-0.