Scientific Research Publishing

Applied Mathematics, 2021, 12, 224-239
https://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

An Oracle Bone Inscription Detector Based on Multi-Scale Gaussian Kernels

Guoying Liu[1], Shuanghao Chen[2*], Jing Xiong[1], Qingju Jiao[1]

[1]School of Computer and Information Engineering, Anyang Normal University, Anyang, China
[2]School of Computer and Engineering, Zhengzhou University, Zhengzhou, China
Email: guoying.liu@aynu.edu.cn, *csh_info_2020@163.com, jingxiong125@163.com, qingju588@163.com

How to cite this paper: Liu, G.Y., Chen, S.H., Xiong, J. and Jiao, Q.J. (2021) An Oracle Bone Inscription Detector Based on Multi-Scale Gaussian Kernels. *Applied Mathematics*, **12**, 224-239.
https://doi.org/10.4236/am.2021.123014

Received: February 22, 2021
Accepted: March 27, 2021
Published: March 30, 2021

Abstract

The detection of Oracle Bone Inscriptions (OBIs) is one of the most fundamental tasks in the study of Oracle Bone, which aims to locate the positions of OBIs on rubbing images. The existing methods are based on the scheme of anchor boxes, involving complex network design and a great number of anchor boxes. In order to overcome the problem, this paper proposes a simpler but more effective OBIs detector by using an anchor-free scheme, where shape-adaptive Gaussian kernels are employed to represent the spatial regions of different OBIs. More specifically, to address the problem of misdetection caused by regional overlapping between some tightly distributed OBIs, the character regions are simultaneously represented by multiscale Gaussian kernels to obtain regions with sharp edges. Besides, based on the kernel predictions of different scales, a novel post-processing pipeline is used to obtain accurate predictions of bounding boxes. Experiments show that our OBIs detector has achieved significant results on the OBIs dataset, which greatly outperforms several mainstream object detectors in both speed and efficiency. Dataset is available at http://jgw.aynu.edu.cn.

Keywords

Oracle Bone Inscriptions, Deep Learning, Object Detection, Hourglass Network

1. Introduction

Oracle Bone Inscriptions (OBIs) are of the oldest and the most mysterious ancient characters in china, which record a large number of unknown ancestors' lives, thoughts, and social states about 3600 years ago. They are very important historical materials for understanding the emergence and development of an-

cient China. The cues of OBIs' locations are valuable for the interpretation of these ancient characters. Therefore, the detection of OBIs is of the most fundamental tasks in the field of Oracle Bone study, which tries to locate the positions of OBIs on rubbing images. At present, few people pay attention to the automatic detection of OBIs, and OBI experts have to locate the OBIs only according to their knowledges and experiences, which is rather boring and time-consuming. In this paper, we mainly focus on the automatic detection of OBIs and attempt to explore a simple but efficient method to find out the precise positions of OBIs on rubbing images.

Currently, there are only a few methods for the OBIs detection task in the field of image processing. For example, Meng [1] build a single-stage OBIs detector via extending SSD300 to SSD1024. Wang [2] introduced a region-based full convolutional network and proposed a novel auxiliary detection algorithm based on character recognition, which can help the detection model reduce the false positive of cracks. In our earlier works [3] [4], we also did some simple explorations on the OBIs detection. We applied several state-of-art object detection models on OBIs dataset and compared and analyzed their detection results. Later, based on the statistical characteristics of the characters in scale size, we redesigned the size and aspect ratio of the anchor and proposed and Spatial Block to stabilize the features and alleviate noise interference during training.

Although these methods have achieved good detection results on the OBIs dataset, there are still certain limitations in accuracy and efficiency. First, due to the lack of character-level class labels in the OBIs dataset, the semantic information of the character is not easily captured through position regression. So, some special characters may be mis-detected by the detection model, for example, some compound characters composed of multiple parts are easily mis-detected as multiple characters, as shown in **Figure 1(Left)**. Similarly, multiple characters are also easy to be detected as a compound character, as shown in **Figure 1(Right)**. Second, most algorithms are based on the scheme of anchor boxes, which involve complex network design and the need for a large number of anchor boxes, such as the number of anchor boxes in DSSD [6] exceeds 40 k and the number in RetinaNet [7] exceeds 100 k. To some extent, it reduces the detection efficiency of the detection model. In this work, our main goal is to explore a simpler OBIs detector and improve the detection accuracy.

We are motivated by the recently proposed CRAFT (Character Region Awareness for Text Detection) [8]. This work uses adaptively shaped Gaussian kernel to represent character region, where the detection of the text instances is converted to the prediction of the corresponding Gaussian map. Thus, it not only bypasses the need for anchor boxes but also enables the detection model to learn character spatial regions. In our work, we follow the formulation that represents the Oracle Bone Character region by adaptively shaped Gaussian kernel and directly outputs the Gaussian prediction of character region, as shown in **Figure 2**. However, experiments show that Gaussian kernel representation has good per-

formance only when dealing with character regions that are not rigidly bounded and it is prone to regional overlapping for some tightly distributed oracle characters, as shown in Figure 3. To overcome this problem, we represent a single character using Gaussian kernels of multiple scales simultaneously, where the smaller the scale, the larger the margin between the character regions, and then based on these kernel predictions, a progressive scale expansion strategy is used to obtain accurate character bounding boxes. Experimental results show that, compared to some state-of-art object detectors, our character detector based on multi-scale Gaussian kernels have achieved more accurate results on the OBIs dataset. The main contributions of this work are summarized as follows:

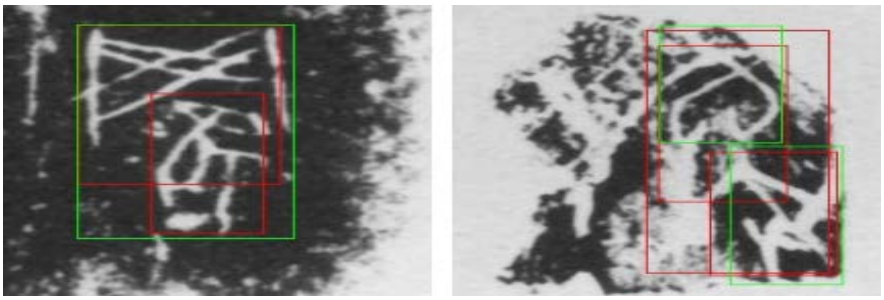


Figure 1. Examples of false detection of Faster R-CNN [5]. The red and blue boxes indicate the predicted and ground truth bounding boxes respectively.

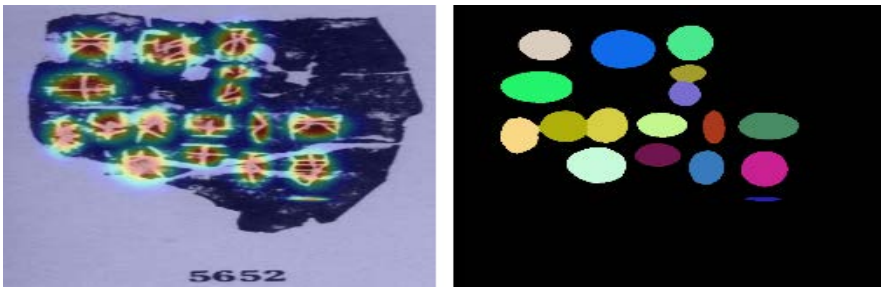
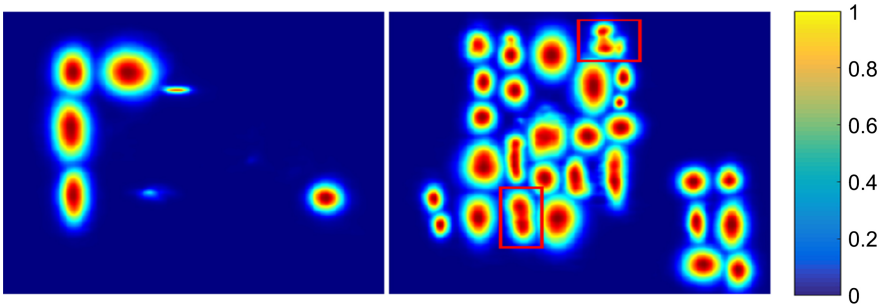


Figure 2. Visualization of the character detection based Gaussian kernel representation. Left: Heatmaps predicated by our proposed framework. Right: Segmentation result based on the heatmaps predicated.



(a) (b)

Figure 3. Outputs by our proposed framework when only using single scale Gaussian kernel. (a) and (b) indicates Heatmaps with not rigidly bounded and tightly distributed between characters.

- We firstly propose an anchor-free detector for OBIs detection. The detector uses the Gaussian kernel to represent the character spatial region, which not only bypasses the need for anchor boxes, but also enables the detection model to learn character spatial regions.
- To overcome the problem of misdetection caused by regional overlapping between some tightly distributed oracle characters, we represent character region using Gaussian kernels of multiple scales simultaneously, and then based on these kernel predictions, character regions with sharp edges are obtained in the way of progressive scale expansion.
- Experiments show that compared to some state-of-art object detectors, our character detector based on multi-scale Gaussian kernels representation has achieved excellent detection results in accuracy and efficiency on the OBIs dataset.

2. Related Work

2.1. Traditional Object Detection Methods

In the early days, most object detection methods [9] [10], adopted the detection routes of Sliding Window or Connected Components Analysis. Based on the Sliding Window method, windows of different scales are usually used to densely slide on the input image and meanwhile, the content of each window is classified by a classifier or rules made by people. The methods based on Connected Components Analysis usually first obtain the selected connected regions through a variety of ways (e.g., color clustering or extreme region extraction) and then filter out non-object regions in the candidate region based on some artificially designed rules. As one of the most successful detection methods, [11] uses Haar features and Adaboost [12] to train a series of cascaded classifiers for face detection, achieving high efficiency and satisfactory accuracy. DPM [13] is another popular method that had maintained the best results on PASCAL VOC [14] for many years. It uses a mixture of multi-scale deformable part models to represent highly variable object classes. Later, some methods further improved the accuracy of object detection based on knowledge of morphological operations [15], conditional random fields [16] and graphs [17].

2.2. Object Detection in Deep Learning

Motivated by the thriving of deep learning-based object [18] or text [19] 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.

Two-stage approaches divide the object detection task into two stages: generates ROIs (Region of Interesting) and then classify and regress the ROIs.

Two-stage approach was introduced and popularized by R-CNN [20]. It generates ROIs using a low-level vision algorithm and then uses a DCN-based region-wise classifier to classify the ROIs independently. Later, SPP-Net [21] and

Fast-RCNN [22] improve R-CNNs by extracting ROIs from the feature maps. However, both still rely on separate proposal algorithms and cannot be trained end-to-end. Faster-RCNN [5] is allowed to be trained end-to-end by introducing RPN (region proposal network). RPN generates proposals from a set of pre-determined candidate boxes, usually known as anchor boxes, which not only makes the detectors more efficient but also allows the detectors to be trained end-to-end. Mask-RCNN [23] further improves the efficiency of Faster-RCNN by adding a mask prediction branch and can thereby detect objects and predict their masks at the same time. Other works focus on the architecture design, the contextual relationship, improving speed.

One-stage approaches remove the ROIs extraction process and directly classify and regress the candidate anchor boxes.

YoLo [24] uses a single feed-forward convolutional network to directly predict object classes and locations, which is extremely fast. After that, YoLov2 [25] further improves YoLo by using more anchor boxes and a new bounding box regression method. DSSD [6] and RON [2] adopt networks similar to the Hourglass Network [26], enabling them to combine low-level and high-level features via skip connections to predict bounding boxes more accurately. RefineDet [27] refines the locations and sizes of the anchor boxes twice, exploiting the merits of both one-stage and two-stage approaches. CornerNet [28] and CenterNet [29] are other keypoint-based approaches that directly detect an object using a pair of corners. Although these methods achieve high performance, it still has room for improvement.

2.3. Related Works of OIBs Detection

Up to now, there are only a few methods for the OBIs detection task in the field of image processing. Meng [1] build a single-stage OBIs detector via extending SSD300 to SSD1024. Wang [2] introduced a region-based full convolutional network and proposed a novel auxiliary detection algorithm based on character recognition, which can help the detection model reduce the false positive of cracks. In our earlier works [3] [4], we also did some simple explorations on OBIs detection. We applied several state-of-art object detection models on the OBIs dataset and compared and analyzed their detection results. Later, based on the statistical characteristics of the characters in scale size, we redesigned the size and aspect ratio of the anchor and proposed the Spatial Block to stabilize the features and alleviate noise interference during training.

However, most of these methods are only a few simple explorations by migrating some classic object detection models slightly modified to the OBIs dataset. Thus, there are still certain limitations in accuracy and efficiency. As mentioned above, most algorithms are based on the scheme of anchor boxes, which involve complex network design and the need for a large number of anchor boxes. Secondly, some special characters (such as compound characters) may be misdetected by the detection model. In this work, our main goal is to explore a

simpler OBIs detector and improve the detection accuracy.

3. Methodology

3.1. The Pipeline of Our Character Detection Model

Our character detection model regards oracle bone characters as special key points, which aims to predict complete and separated character regions. The overall data stream of the model is shown in **Figure 4**. Firstly, the rubbing input I_O passes through a convolutional neural network to predict a feature map $I_F \in R^{H\times W\times C}$ that incorporates multi-layer context information of feature maps. The feature map I_F is mapped to n branches by the region prediction module whose output are used to generate n scale region maps $S_1, S_2, \cdots, S_n$, where each S_i represents a character region score map of scale size. S_1 represents the character region prediction of the minimal scale, and S_n represents the character region prediction of the maximal scale. Finally, based on these obtained multi-scale Gaussian region predictions, the final accurate character bounding boxes are obtained after a series of simple post-processing operations.

3.2. Architecture of Detection Network

The OBIs detector uses the Hourglass Network [26] as its basic backbone. The Hourglass Network is a fully convolutional neural network with a cascade structure, which is composed of one or more Hourglass modules. The Hourglass module is similar to a lightweight encoding and decoding network, which down samples the input features through a series of convolution and maximum pooling, and then restores to the original resolution through a series of up sampling and convolutional layers. To reduce the loss of details caused by the max-pooling operation, skip connections are used to bring the details back to the up-sampling feature. Besides, a single hourglass module can capture global and local features in a unified structure. When multiple hourglass modules are stacked in the network, the Hourglass model can reprocess features to obtain higher-level information.

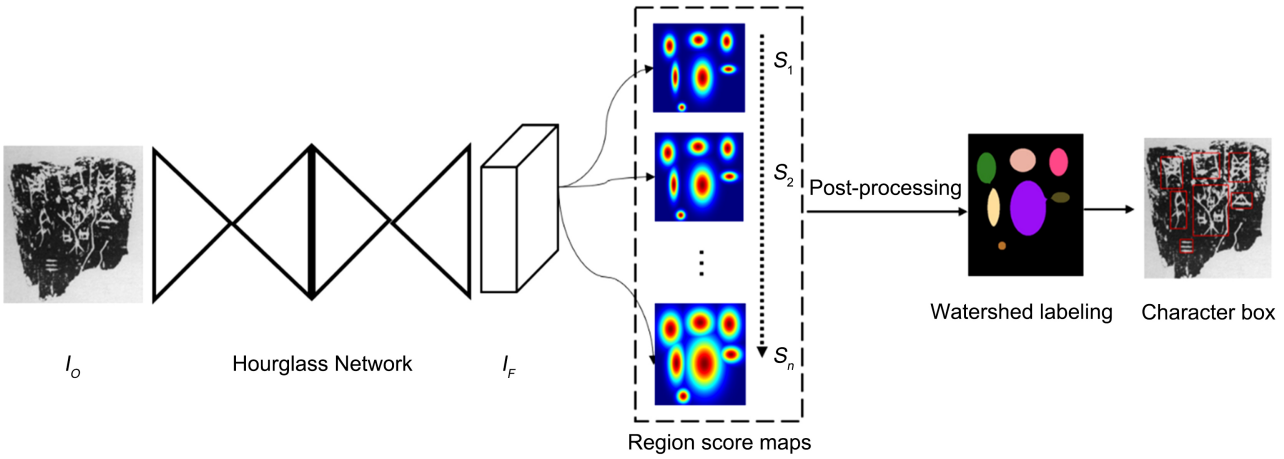


Figure 4. The overall structure of the OBIs detector based on multi-scale Gaussian kernels.

In our character detector, we stack two Hourglass modules and make a few modifications to the overall Hourglass network. Specifically, before the features are input to the Hourglass module, we use a convolutional layer with stride 2 and a 3 × 3 convolution to replace the 7 × 7 convolution in the original network, which can scale the input image to 1/2 size. Similarly, in the Hourglass module, a 3 × 3 convolution with stride 2 is used to replace the maximum pooling in the original module to down-sample the input features. At the end of the Hourglass module, we continue to add an up-sampling layer to restore the output to the original input resolution.

3.3. Loss Functions

The overall loss function of the OBIs detection model is expressed as follows:

$$L = \lambda L_{FullMap} + (1-\lambda) L_{ZoomMap} \tag{1}$$

where $L_{FullMap}$ and $L_{ZoomMap}$ represent the loss of character region instance with complete shape and multiple shrinking character region instances respectively, and λ is used to balance the weight of $L_{FullMap}$ and $L_{ZoomMap}$.

$$L_{FullMap} = L_{Pix}\left(S(p), S^*(p)\right) \tag{2}$$

where p represents the coordinate position of a pixel. $S(p)$ represents the predicted character region score with complete shape, and $S^*(p)$ represents the corresponding ground truth score.

$$L_{Pix}\left(T(p), T^*(p)\right) = \sum_p \left\| T(p) - T^*(p) \right\|_2^2 \tag{3}$$

$$L_{ZoomMap} = \sum_{i=1}^{N-1} L_{Pix}\left(Z_i(p), Z_i^*(p)\right) \tag{4}$$

where N represents the number of scales, $Z_i(p)$ represents the predicted character region score of the scale i, and $Z_i^*(p)$ represents the ground truth score of the scale i.

In addition to the character features, there is a lot of disturbance on the rubbing image that is very similar to character features, such as background noise and cracks. To enable the detection model to learn to distinguish these patterns, Online Hard Negative Mining [30] (OHEM) is applied to enforce the 1:3 ratio of positive and negative pixels in the detection loss $L_{FullMap}$.

3.4. Ground Truth Label Generation

For each training image, we generate the ground truth label of the region score with complete shape and n shrinking using character-level bounding boxes provided by the OBIs dataset, as shown in **Figure 5**. The detailed steps are as 1) According to character level bounding boxes provided by the OBIs dataset, following the shrinking principle in [8], setup n shrinking pixel spacing $D = \{d_1, d_2, \cdots, d_n\}$. 2) Based on the shrinking spacing D, shrink inward along the original bounding boxes to obtain n bounding box sets of different scales. 3) Prepare a 2D isotropic Gaussian kernel. 4) Calculate the perspective transformation matrix$M$ between the Gaussian kernel and each character box. 5)

Based on the perspective transformation matrix M, warp Gaussian map to the box area.

3.5. Inference

During inference, the detection model finally outputs n character region maps of different scales. In this section, we briefly describe how to predict the accurate character level bounding box based on the region score maps.

The key of the post-processing pipeline is a scale extension algorithm from [31], which adopts a novel progressive extension strategy to detect dense scene text. It uses the adjacent relationship between Gaussian heatmaps of different scales to gradually expand from the text region with the minimal kernel to the maximal kernel with complete shape. On this basis, we added some additional steps and a few modifications to suit our character detection task. We first perform a simple pre-processing on the original multi-scale gaussian map prediction and reduce the noise in the gaussian map through some morphological operations (*opening operation*, *distanceTransform*). Secondly, for the separated character regions K obtained by the scale extension algorithm, we calculated their connected components C and assigned different labels *Label*. Finally, based on these assigned *Label*, the minimum enclosing rectangle of each connected component is calculated to obtain the final accurate bounding box. The functions like *connectedComponents*, *morphologyEx*, and *minAreaRect* provided by Opencv can be applied for this purpose. The details are shown in **Algorithm 1**.

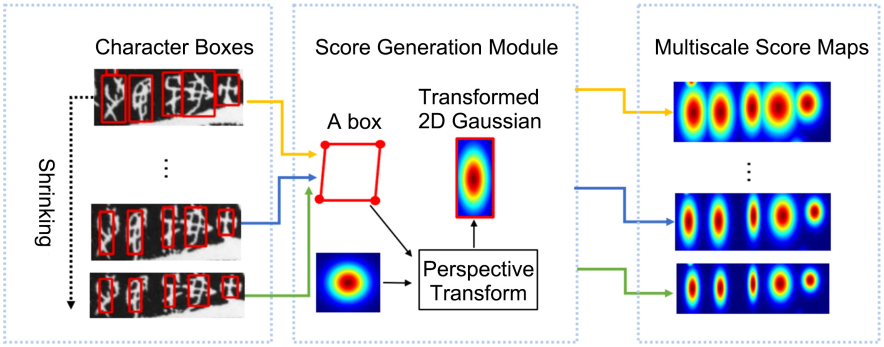


Figure 5. The generation process of ground truth label.

Algorithm 1. Post-processing pipeline of detection model.

Input: Kernel predictions $Z=\{Z_1,Z_2,\cdots,Z_n\}$
Output: Bounding box list L
Function prediction (Z)
1) Initialize a set of zero arrays $M=\{M_1,M_2,\cdots,M_n\}$
2) While $i=1$ to n do
3) If $Z_i(p)>\delta$ Then $M_i(p)=True$ // δ is a threshold with value of 0.35
4) $M \leftarrow morphologyEx(M)$
5) $K \leftarrow scaleExpanded(M)$
6) $C, Label \leftarrow connectedComponents(K)$
7) $L \leftarrow minAreaRectByLabel(C, Label)$
8) return L

4. Experiments

4.1. Oracle Bone Inscriptions Dataset

In this paper, all experiments are based on the OBIs dataset provided by the Key Laboratory of the Ministry of Education for Oracle Information Processing, Anyang Normal University. The dataset focuses on the task of OBIs detection and it mainly includes two parts: the number of oracle bone rubbing image collected from the OBIs literature collection using a high-resolution scanner, which is up to 9500 pieces, and the bounding box of characters level by hand-made. Different from the general natural scene image, the rubbing image mainly has the following characteristics:

High noises: Oracle bone rubbing, as the main carrier of OBIs, was buried in the ruins of Anyang for a long time and was not discovered until 120 years ago. Therefore, there is inevitably a certain degradation on the rubbing appearance. The most significant of these is a large amount of noise on the rubbing. These noises have different rules and are densely distributed on the rubbing image, which brings great challenges to the task of OBIs detection.

Cracks: Due to the burial environment and private excavations, many of the unearthed oracle bone rubbing have been broken, and various cracks have appeared on the surface of the rubbing. These cracks are very similar to character characteristics in texture, and it is easy to mistake for oracle bone characters.

Distribution: The characters on the same rubbing image are of different sizes, different directions, and random distribution. Besides, in the 56,743 oracle bone rubbing, there are 1425 words. Among them, there are 366 common characters, 500 not usually used, and 559 rare.

There are up to 9500 oracle rubbing records on OBIs dataset. In this experiment, the training set, validation set, and test set contain 8287, 436, and 411 data records respectively.

4.2. Experimental Environment

In this experiment, the source code of all models is based on the Pytorch deep learning framework and trained on the four Nvidia TITAN X GPUs. Especially, due to the lack of character category information in the OBI dataset, the class-agonistic strategy is adopted. By default, all characters are treated as a single category, and the same category label is assigned. During training, the rubbing image is scaled to 512 × 512 resolution, and the Adam optimizer is used to update and optimize the parameters. We start Adam at the learning rate of 0.0001, and use 0.9 momentum and 0.0001 weight decay empirically.

4.3. Evaluation Indicators

We mainly evaluate the overall performance of the character detection model from the perspective of efficiency and accuracy. The three indicators of network weight parameters, floating-point calculation, and inference speed are used to

evaluate the overall detection efficiency of the model. Precision (P), Recall (R), and F-Measure (F) has commonly used measurement indicators in mainstream object detection methods to measure the detection accuracy of the model. The calculation formulas of these indicators are as follows:

$$P = \frac{TP}{TP + FP} \tag{5}$$

$$R = \frac{TP}{TP + FN} \tag{6}$$

$$F = \frac{2 * P * R}{P + R} \tag{7}$$

where TP, FP and FN represent Ture Positive, False Positive, False Negative respectively.

4.4. Ablation Experiments

The validity of Gaussian kernel representation: In addition to Gaussian kernels that can be used to represent character regions, binary mask is another option. To compare the difference between the two represents, we simply compare the character detection model (using only a single scale Gaussian kernel) with the state-of-art semantic segmentation model DeepLabv3 [32]. Specifically, we roughly divide the rubbing image into foreground and background regions according to the principle that whether the pixels are inside the character level box annotation provided by the OBIs dataset and then use the trained segmentation model directly to predict the foreground character regions. The visualization of these models' output results is shown in **Figure 6**. The binary mask represents the character regions using discrete values without distinction and the obtained prediction results have more regional overlapping. On the contrary, the Gaussian kernelencodes the character region based on the distance relationship with the center pixel, and the obtained character regions are clearer on the boundary.

After obtaining these binary and Gaussian region predictions, we use some simple post-processing operations (including *connectedComponents*, *minAreaRect*) to get the character bounding boxes and then calculate their P, R, F indicators respectively. The quantitative results are shown in **Table 1**. The method based on Gaussian kernel is significantly higher than the binary mask representation on all indicators. This shows once again that the Gaussian kernel representation has obvious advantages and is more conducive to expressing the tightly distributed character region.

Table 1. The quantitative results based on binary mask and Gaussian kernel represent.

Methods	Precision (P)	Recall (R)	F-Measure (F)
DeepLabv3 [32]	0.626	0.638	0.632
Gaussian(our)	0.776	0.646	0.705

Is multi-scale Gaussian kernel necessary? To answer this question, we retrain the detection model, when the number of scales is different. The assessment results are shown in Figure 7, from which we can find that with the growing of n, the F-measure keeps rising and begins to go down when $n>6$. The informative result suggests that it is not that the larger the number of scales, the better. When $n=6$, the detection model achieves the highest F-measure, thus, it is more beneficial to achieve better detection results for the task of OBIs detection when the number of scales is 6. Besides, although with the growing of n, F-measure shows a certain decline, but compared to using a single-scale Gaussian kernel, when $n>1$, the value of F-measure is significantly higher. This shows to some extent that the design of multiple kernel scales is essential and effective.

4.5. Accuracy Comparison

To better evaluate the detection effect of our character detection model, we compare our model with several mainstream object detection models, which not only include two-stage object detectors such as Faster RCNN [5], but also single-stage object detectors such as YoLov3 [35], RBFNet [34].

(a) (b) (c)

Figure 6. Comparison results based on the binary mask and Gaussian kernel represent. (a) Rubbing input; (b) Binary mask prediction by DeepLabv3 [32]; (c) Gaussian region prediction by our model.

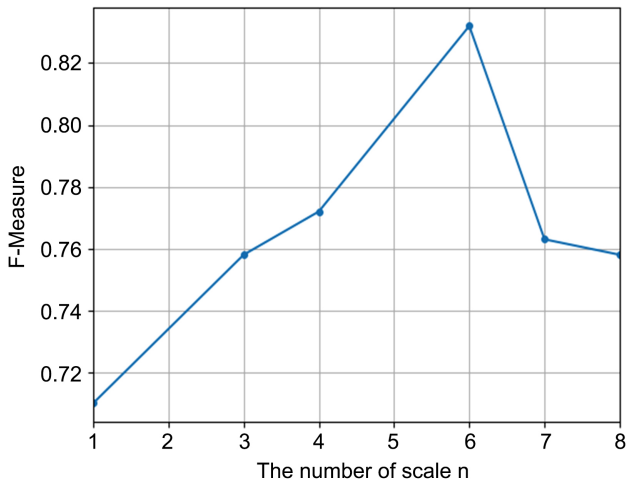


Figure 7. Ablation study on the number of scales n.

Table 2 shows the quantitative results with these state-of-art detection models. In terms of accuracy, our detector achieved the highest score of 89.7%,which is significantly better than the second place with a gap of 12%. However, in terms of recall rate, our model performed relatively weakly, almost at the bottom of all the models. For this phenomenon, we believe that the possible reason lies in the fact that for the detection methods based on anchor boxes, the non-maximum suppression (NMS) operation uses a manually set threshold to filter out some invalid candidate boxes, which may have some missed candidate boxes, resulting in a high recall rate. To more accurately evaluate the detection effect, we continue to compare the F-measure that is the balance of indicators of precision and recall. Similarly, our model still achieves the best results, far better than the second place by 5%. Therefore, this reflects the advantage of our model in accuracy to some degree. Also, it is not difficult to imagine that our model can capture more semantic information about the characters and has character area awareness by using directly Gaussian kernels to represent the character regions, so it can get more accurate detection results.

4.6. Efficiency Comparison

We evaluate the detection efficiency of our character detector by measuring its inference speed, weight parameters, floating-point operations and then compared them with several state-of-art detectors.

Table 3 shows the efficiency comparison with these models. In inference speed, our model achieved the fastest inference speed of 23FPS, which 5FPS higher than the second place YoLov3 [35]. In weight parameters, our model requires fewer parameters, occupying only 12.73M, which is much lower than the 26.29M of the suboptimal model SSD [19]. In terms of floating-point operations, our model is only weaker than YoLov3 [35] and won the second position. Nevertheless, the number of floating-point operations is only 57.34 GMac, which is far lower than other state-of-art detection models. It is comprehensively known that our model can achieve faster inference speed while Has a lighter computing burden.

Table 2. Accuracy quantitative results with state-of-art detection models.

Methods	Precision (P)	Recall (R)	F-Measure (F)
FasterRCNN [5]	0.754	0.778	0.766
SSD [19]	0.748	0.758	0.753
RefineDet [33]	0.752	0.805	0.778
RBFNet [34]	0.761	0.789	0.775
YoLov3 [35]	0.776	0.784	0.78
Ours	**0.897**	0.775	**0.832**

Table 3. Comparison results of detection efficiency with state-of-art detection model.

Methods	Speed(FPS)	Parameters(M)	Flops(GMac)
Faster RCNN [5]	3	41.37	129.27
SSD [19]	9	26.29	90.4
RefineDet [33]	14	34.44	97.94
RBFNet [34]	15	36.64	103.65
YoLov3 [35]	17	61.92	50.06
Ours	23	12.73	57.34

5. Conclusion

In this paper, we first propose an anchor-free OBIs detector for OBIs detection. The detector uses adaptively shaped Gaussian kernel to represent the spatial region of the characters, which not only bypasses the need for anchor boxes but also enables the detection model to learn character spatial regions. Furthermore, to address the problem of misdetection caused by regional overlapping between some tightly distributed characters, the character region is simultaneously represented by multiscale Gaussian kernels to obtain character regions with sharp edges. Finally, based on these kernel predictions of different scales, a novel post-processing pipeline is used to obtain accurate bounding box predictions. The experimental results show that our OBIs detector has achieved good detection results on the OBIs dataset.

Fund

This work is supported by the joint fund of National Natural Science Foundation of China (NSFC) and Henan Province of China under Grant U1804153, and partly supported by the Scientific and Technological Research Projects in Henan province under Grant 212102310545 and 212102210502 and the Anyang Science and Technology Plan Project under Grant 2021C01GX020.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Meng, L. (2017) Two-Stage Recognition for Oracle Bone Inscriptions. *Lecture Notes in Computer Science*, **10485**, 672-682. https://doi.org/10.1007/978-3-319-68548-9_61

[2] Hao, W. (2019) Research on Oracle Detection and Recognition Based on Deep Learning. South China University of Technology, Guangzhou.

[3] Xing, J., Liu, G. and Xiong, J. (2019) Oracle Bone Inscription Detection: A Survey of Oracle Bone Inscription Detection Based on Deep Learning Algorithm. *Proceedings*

of the International Conference on Artificial Intelligence, *Information Processing and Cloud Computing*, Sanya, December 2019, Article No. 39. https://doi.org/10.1145/3371425.3371434

[4] Liu, G., Xing, J. and Xiong, J. (2020) Spatial Pyramid Block for Oracle Bone Inscription Detection. *ICSCA* 2020: *Proceedings of the* 2020 9*th International Conference on Software and Computer Applications*, February 2020, 133-140. https://doi.org/10.1145/3384544.3384561

[5] Ren, S., He, K. and Girshick, R. (2016) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031

[6] Fu, C.Y., Liu, W., Ranga, A., Tyagi, A. and Berg, A.C. (2017) DSSD: Deconvolutional Single Shot Detector.

[7] Lin, T.Y., Goyal, P., Girshick, R., He, K. and Dollar, P. (2020) Focal Loss for Dense Object Detection. *EEE Transactions on Pattern Analysis and Machine Intelligence*, **42**, 318-327. https://doi.org/10.1109/TPAMI.2018.2858826

[8] Baek, Y., Lee, B., Han, D., Yun, S. and Lee, H. (2019) Character Region Awareness for Text Detection. *IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition*, Long Beach, 15-20 June 2019, 9357-9366. https://doi.org/10.1109/CVPR.2019.00959

[9] Epshtein, B., Ofek, E. and Wexler, Y. (2010) Detecting Text in Natural Scenes with Stroke Width Transform. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, 13-18 June 2010, 2963-2970. https://doi.org/10.1109/CVPR.2010.5540041

[10] Huang, W., Lin, Z., Yang, J. and Wang, J. (2013) Text Localization in Natural Images Using Stroke Feature Transform and Text Covariance Descriptors. *IEEE International Conference on Computer Vision*, Sydney, 1-8 December 2013, 1241-1248. https://doi.org/10.1109/ICCV.2013.157

[11] Papageorgiou, C.P., Oren, M. and Poggio, T. (1998) A General Framework for Object Detection. *Sixth International Conference on Computer Vision*, Bombay, 7 January 1998, 555-562. https://doi.org/10.1109/ICCV.1998.710772

[12] Schapire, R.E. (2013) Explaining AdaBoost: Empirical Inference. Springer, Berlin, Heidelberg.

[13] Felzenszwalb, P.F., Girshick, R.B., McAllester, D. and Ramanan, D. (2010) Object Detection with Discriminatively Trained Part Based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**, 1627-1645. https://doi.org/10.1109/TPAMI.2009.167

[14] Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J. and Zisserman, A. (2015) The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, **111**, 98-136. https://doi.org/10.1007/s11263-014-0733-5

[15] Lee, J.J., Lee, P.H., Lee, S.W., Yuille, A. and Koch, C. (2011) AdaBoost for Text Detection in Natural Scene. *IEEE International Conference on Document Analysis and Recognition*, Beijing, 18-21 September 2011, 429-434. https://doi.org/10.1109/ICDAR.2011.93

[16] Wang, K., Babenko, B. and Belongie, S. (2011) End-to-End Scene Text Recognition. *IEEE International Conference on Computer Vision*, Barcelona, 1457-1464.

[17] Wang, T., Wu, D.J., Coates, A. and Ng, A.Y. (2012) End-to-End Text Recognition with Convolutional Neural Networks. Proceedings of the 21st International Confe-

rence on Pattern Recognition, Tsukuba, Japan, 11-15 November 2012, 3304-3308.

[18] Li, Y., He, K., Sun, J., *et al.* (2016) R-fcn: Object Detection via Region-Based Fully Convolutional Networks. *Proceedings of the* 30*th International Conference on Neural Information Processing*, Morehouse Lane, Red Hook, December 2016, 379-387.

[19] Wang, Y., Xie, H., Zha, Z.-J., Xing, M., Fu, Z. and Zhang, Y. (2020) ContourNet: Taking a Further Step toward Accurate Arbitrary-Shaped Scene Text Detection. *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Seattle, 13-19 June 2020, 11753-11762. https://doi.org/10.1109/CVPR42600.2020.01177

[20] Liu, W., Fu, C.H., Reed, S., *et al.* (2016) SSD: Single Shot Multi-Box Detector. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., Eds., *Computer Vision—ECCV* 2016, Springer, Cham, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2

[21] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

[22] Girshick, R. (2015) Fast R-CNN. 2015 *IEEE International Conference on Computer Vision*, Santiago, 7-13 December 2015, 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[23] He, K., Gkioxari, G., Dollár, P. and Girshick, R. (2020) Mask R-CNN. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42**, 386-397. https://doi.org/10.1109/TPAMI.2018.2844175

[24] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

[25] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. 30*th IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 6517-6525. https://doi.org/10.1109/CVPR.2017.690

[26] Newell, A., Yang, K. and Deng, J. (2016) Stacked Hourglass Networks for Human Pose Estimation. In: Leibe, B., Matas, J., Sebe, N. and Welling, M., Eds., *Computer Vision—ECCV* 2016, Springer, Cham, 483-499. https://doi.org/10.1007/978-3-319-46484-8_29

[27] Lin, G., Milan, A., Shen, C. and Reid, I. (2017) RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **42**, 1228-1242. https://doi.org/10.1109/CVPR.2017.549

[28] Law, H. and Deng, J. (2020) CornerNet: Detecting Objects as Paired Keypoints. *International Journal of Computer Vision*, **128**, 642-656. https://doi.org/10.1007/s11263-019-01204-1

[29] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q. and Tian, Q. (2019) CenterNet: Keypoint Triplets for Object Detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision* (*ICCV*), 27 October-2 November 2019, 6569-6578. https://doi.org/10.1109/ICCV.2019.00667

[30] Shrivastava, A., Gupta, A. and Girshick, R.B. (2026) Training Region-Based Object Detectors with Online Hard Example Mining. 2016 *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 761-769. https://doi.org/10.1109/CVPR.2016.89

[31] Wang, W., Li, X. and Liu, T. (2019) Shape Robust Text Detection with Progressive Scale Expansion Network. *IEEE/CVF Computer Society Conference on Computer*

Vision and Pattern Recognition, Long Beach, 15-20 June 2019, 9336-9345. https://doi.org/10.1109/CVPR.2019.00956

[32] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A.L. (2018) DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**, 6834-6848. https://doi.org/10.1109/TPAMI.2017.2699184

[33] Zhang, S., Wen, L., Bian, X., Lei, Z. and Li, S.Z. (2018) Single-Shot Refinement Neural Network for Object Detection. *IEEE Transactions on Circuits and Systems for Video Technology*, **31**, 674-687. https://doi.org/10.1109/TCSVT.2020.2986402

[34] Liu, S., Huang, D. and Wang, Y. (2018) Receptive Field Block Net for Accurate and Fast Object Detection. Computer Vision. In: Ferrari, V., Hebert, M., Sminchisescu, C. and Weiss, Y., Eds., *Computer Vision—ECCV* 2018, Springer, Cham, 404-419. https://doi.org/10.1007/978-3-030-01252-6_24

[35] Redmon, J. and Farhadi, A. (2018) YOLOv3: An Incremental Improvement.