Computers, Materials & Continua
DOI: 10.32604/cmc.2022.028187
Article

Tech Science Press

A Fast Tongue Detection and Location Algorithm in Natural Environment

Lei Zhu[1], Guojiang Xin[1,2,*], Xin Wang[1], Changsong Ding[1,2], Hao Liang[1,2] and Qilei Chen[3]

[1]School of Informatics, Hunan University of Chinese Medicine, Changsha, 410208, China
[2]TCM Big Data Analysis Laboratory of Hunan, Changsha, 410208, China
[3]Department of Computer Science, University of Massachusetts Lowell, Lowell, 01854, USA
*Corresponding Author: Guojiang Xin. Email: lovesin_guojiang@126.com
Received: 05 February 2022; Accepted: 26 April 2022

Abstract: The collection and extraction of tongue images has always been an important part of intelligent tongue diagnosis. At present, the collection of tongue images generally needs to be completed in a sealed, stable light environment, which is not conducive to the promotion of extensive tongue image and intelligent tongue diagnosis. In response to the problem, a new algorithm named GCYTD (GELU-CA-YOLO Tongue Detection) is proposed to quickly detect and locate the tongue in a natural environment, which can greatly reduce the restriction of the tongue image collection environment. The algorithm is based on the YOLO (You Only Look Once) V4-tiny network model to detect the tongue. Firstly, the GELU (Gaussian Error Liner Units) activation function is integrated into the model to improve the training speed and reduce the number of model parameters; then, the CA (Coordinate Attention) mechanism is integrated into the model to enhance the detection precision and improve the failure tolerance of the model. Compared with the other classical algorithms, Experimental results show that GCYTD algorithm has a better performance on the tongue images of all types in terms of training speed, tongue detection speed and detection precision, etc. The lighter model can contribute on deploying the tongue detection model on small mobile terminals.

Keywords: Tongue detection; YOLO V4-tiny; CA mechanism; GELU

1 Introduction

Traditional Chinese medicine thinks that some internal body diseases can be reflected by the changes on tongue characteristics. As an important part of Chinese medicine, the tongue diagnosis may result in a non-invasive diagnosis and reduce the diagnostic cost, which plays an important role in clinical diagnosis [1].

Traditional tongue diagnosis mainly relies on the physician's visual observation, analysis and judgment. The diagnostic results are often related to the physician's clinical experience and medical knowledge, which may cause the problems of subjective and inconsistent evaluation criteria. Consequently, more and more researchers focused on this problem and proposed many methods to improve

the precision, objectivity and standardization of the tongue detection and location [2,3]. For example, Tang et al. [4] proposed a two-stage method based on region detection and deep learning detection to solve the identification of the tongue edge area, tongue was detected and classified by CCNN (Cascaded Convolutional Neural Network) and a fine-grained classification network. Shamim et al. [5] proposed a tongue detection algorithm based on DCNN (Deep Convolutional Neural Networks) model with Vgg19 (Visual Geometry Group) structure, which can distinguish the benign lesions and the precancerous lesions of the tongue, and realize the early diagnosis of OCC (Oral Cavity Cancer). Lu et al. [6] proposed a joint calibration and localization algorithm based on expectation maximization, it can improve the tongue localization accuracy and tracking precision. Xin et al. [7] proposed a medical platform to automatically detect the tongue of the Internet of Things based on the popular aspect of the image sequence. Zhou et al. [8] proposed a novel end-to-end model for multi-task learning of tongue localization and segmentation, named TongueNet, in which introduced a FPN (Feature Pyramid Network) and the ROI (Region of Interests), achieves good performance for the segmentation of tongue body in terms of both robustness and accuracy. Hu et al. [9] proposed a fully-channel regional attention network for disease-location recognition with tongue images, which used a stochastic region pooling method to gain detailed regional features and used an inner-imaging channel relationship modeling method to model multi-region relations on all channels. Zheng et al. [10] proposed a tongue detection method based on image segmentation, using the texture characteristics of the tongue image to achieve tongue image detection. Tania et al. [11] studied the merits and capabilities, associated research gaps in current works on ATD (Automated Tongue Diagnosis) systems, and proposed a conceptual framework for the ATD system on mobile platform. This framework could connect tongue diagnosis with the future point-of-care health system. Thanikachalam et al. [12] proposed an intelligent Deep Learning based disease diagnosis model using the biomedical tongue image. The model incorporated Fuzzy-based Adaptive Median Filtering (FADM) technique for noise removal process and SqueezeNet model as a feature extractor.

Tongue character is the most important content of the whole tongue diagnosis system, the quality of data will directly affect the results of diagnosis. As the methods or systems mentioned above are strict for both acquisition equipment and acquisition environment, unsuitable for widespread promotion in the natural environment where there are many uncertainties such as color temperature of the light source, light intensity, firing angle, etc., which may make the acquired tongue image to have some deviation degree compared with that from the standard environment [13]. To make the tongue detection and location suitable for the image from the natural environment and widely applicated, a new algorithm named GCYTD is proposed, which combines the GELU activation function and the CA attention mechanism.

2 Algorithm Description

2.1 Yolo V4-Tiny Network Model

The YOLO network model is an object recognition and positioning algorithm based on DCNN [14,15], The most important feature is the fast detection speed, which can be used to detect targets rapidly [16]. YOLO network model [17] extracts the depth features of the input image through the backbone feature extraction network, which can improve the accuracy of target detection by using feature fusion [18] to further improve the validity of the feature [19]. YOLO V4-tiny network model is a lightweight YOLO V4 network model [20] which combines FPN and FCN (Fully Convolutional Networks) [21] to ensure the accuracy of the certain model. It is simple in the structure, light in the model, and suitable for detecting small targets.

According to the training course, however, the precision of the YOLO V4-tiny network model is slightly less than that of YOLO V4 network model, the convergence speed of the loss feature is also slower, and the training time is longer. Fig. 1 shows the loss function graph obtained from YOLO V4 network model training process. It shows that when the number of the iterations is 50 epochs, the loss function has a very large fluctuation, the magnitude reaches to about 2.5. When the number of the iterations reaches 100 epochs, the convergence speed of the loss function decreases significantly, and the volatility of the loss function is large, the loss function curve shows acute upward and downward fluctuations. The figure indicates that the original YOLO V4 network model is very unstable in the training process and vulnerable to uncertain aspects, which may lead to over-fitting and less to meet the accuracy requirements for the tongue detection.

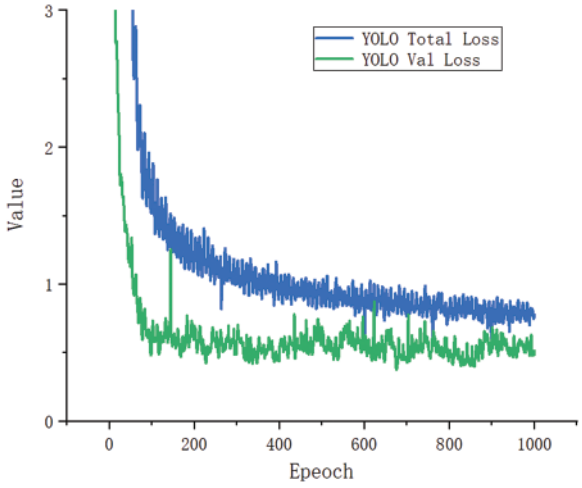

Figure 1: YOLO loss function graph

2.2 Gcytd Algorithm

2.2.1 Combined Gelu Activation Function

As shown above, the detection accuracy and training speed of YOLO V4 network model can't meet the requirements of tongue detection in a timely manner. As the collected tongue images are affected by many uncertainties, it is not appropriate to directly use YOLO V4 network model to detect the tongue.

In YOLO network model, the function of the activation function is to add nonlinear factors, so that the model can better fit the data and speed up the convergence speed of the model. However, the Mish function and the Swish function in the YOLO network model have significantly lower convergence speed than the GELU function in the training process. Focus on the shortcomings of the YOLO network model, we propose to incorporate the GELU activation function [22] into YOLO V4-tiny network model instead of the Mish and Swish activation function. GELU can make YOLO V4-tiny network model converge better and faster than Mish and Swish, and the number of model parameters is greatly reduced. The gradient comparison chart of Mish, Swish and GELU is shown in Fig. 2. From the figure, it can be seen that in the negative value range, the gradient of GELU is smaller than that of Mish and Swish, which makes the unilateral suppression effect better. In the whole value range, the gradient of GELU also changes faster, which can make the model converge faster to

effectively avoid the problem of gradient disappearance and gradient explosion, which can speed up the training speed of the model.

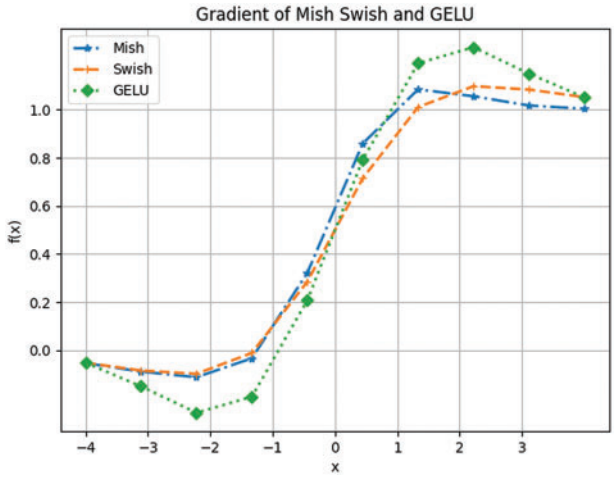

Figure 2: Gradient comparison chart of Mish, Swish and GELU

The GELU function is expressed as Eq. (1):

$$GELU\,(x) = xP\,(X \leq x) = x\varphi\,(x) = x.\frac{1}{2}\left[1 + erf\left(\frac{x}{\sqrt{2}}\right)\right] \tag{1}$$

Here, the input neuron x represents the input tongue image. The GELU function is essentially the synthesis of dropout, zone out, and ReLUs (Rectified Linear Units). It introduces the idea of random regularity, and is a kind of probability description of neuron input. Its experimental effect is better than that of Mish and Swish.

2.2.2 Feature Fusion of CA Mechanism

YOLO V4-tiny network model uses the Concat (concatenation) operation to fuse the feature maps of different scales, but simply connects feature information in the channel, which cannot reflect the importance and the relevance of different features, and meanwhile the fused feature maps cannot accurately describe the target. Therefore, we propose that adding an attention mechanism [23] after the feature fusion to obtain more accurate feature information.

The attention mechanism of target detection based on deep learning [24] essentially assigns different weights to different pixels, so that the acquired feature maps can contain more valid information. During the feature fusion process, the CA mechanism is introduced to assign the appropriate weights to the pixels in the channel dimension and the spatial location dimension. As shown in Fig. 3, as the main part of CA mechanism, CA block's role is to encode spatial location information with attention blocks. In this way, long-range dependencies can be captured along one spatial dimension and meanwhile precise positional information can be preserved in the other spatial dimension. The two attention blocks in different spatial dimensions are firstly connected by Concat operation and sent to a 1×1 convolutional transformation, and then the intermediate feature diagram of $C \times 1 \times (W+H)$ is obtained by a nonlinear activation function and normalization processing. Then the feature diagram is divided into two separate tensors along the spatial dimension, and then use the

other two 1×1 convolutional transformations to input the same number of channels into the nonlinear activation function, and get the weight of each channel feature.

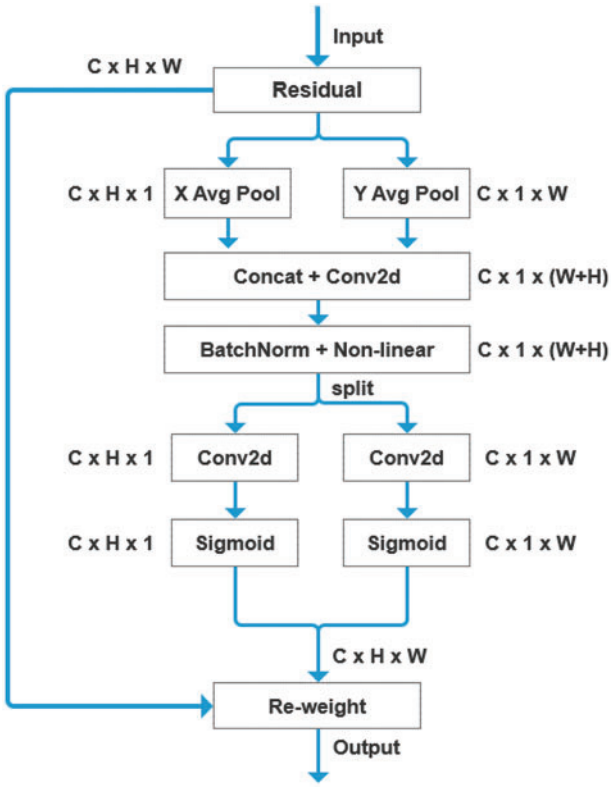

Figure 3: Coordinate Attention block. 'X Avg Pool' and 'Y Avg Pool' refer to 1D horizontal global pooling and 1D vertical global pooling, respectively

In detail, input x, each channel is encoded along the horizontal and the vertical coordinates using two spatial ranges $(H, 1)$ and $(1, W)$ of the pooled core. Therefore, the output of c channel at height h can be expressed as Eq. (2):

$$Z_c^h(h) = \frac{1}{w} \sum_{0 \le i < W} x_c(h, i) \tag{2}$$

Here, $Z_c^h(h)$ is the output associated with height h and the c channel. Similarly, the output of c channel at width w can be expressed as Eq. (3):

$$Z_c^w(w) = \frac{1}{H} \sum_{0 \le j < H} x_c(j, w) \tag{3}$$

Here, $Z_c^w(w)$ is the output associated with width w and the c channel.

The two transformations aggregate the features along two spatial dimensions to obtain a pair of direction-aware feature maps. These two transformations provide attention blocks to capture long-term dependencies along one spatial dimension and to hold accurate location information along

another spatial dimension, which helps the network model to capture the spatial location information of tongues more accurately.

Then, the global receptive field is obtained and accurate position information is encoded by Eqs. (2) and (3). The coordinate attention generation makes full use of the captured position information to effectively capture the relationship between channels. In detail, join the aggregated feature maps generated by Eqs. (2) and (3) together firstly, and then send them to a shared 1×1 convolution function F_1, yield Eq. (4):

$$f = \delta\left(F_1\left(\left[Z^h, Z^w\right]\right)\right) \tag{4}$$

Here, $F_1\left(\left[Z^h, Z^w\right]\right)$ is the Concat operation along the spatial dimension, δ represents the nonlinear activation function and $f \in R^{C/r \times (H+W)}$ is the intermediate feature map encoding spatial information in horizontal and vertical directions. Here, r represents the reduction ratio of the control block size. Firstly, f is divided into two independent tensors $f^h \in R^{C/r \times H}$ and $f^w \in R^{C/r \times W}$ along the spatial dimension. Then two 1×1 convolutional transformations are used to transform and input the two tensors f^h and f^w with the same number of channels, yield Eqs. (5) and (6):

$$g^h = \delta\left(F_h\left(f^h\right)\right) \tag{5}$$

$$g^w = \delta\left(F_w\left(f^w\right)\right) \tag{6}$$

Here, δ is the Sigmoid function. The output g^h and g^w are extended respectively as the attention weight, the final CA block output can be expressed as Eq. (7):

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{7}$$

Different from the Channel Attention only focusing on the importance of different channels, Coordinate Attention also considers the coding of the spatial information. As mentioned above, attention along both horizontal and vertical coordinates acts on tensors, each element in the two attention block maps reflects whether the interest object exists in the corresponding row and column. This coding process can make Coordinate Attention to locate the exact position of tongue more accurately, thus help the whole model to better detect the target.

2.2.3 Gcytd Algorithm Framework

To solve the problem of low detection accuracy and low speed of the YOLO V4-tiny network model, the GELU activation function and the CA mechanism are combined into the YOLO V4-tiny network model to propose a new tongue detection algorithm named GCYTD. The GELU activation function reduces the calculation amount of the model parameters, making the loss function converge faster and accelerate the detection speed of the model. The CA mechanism expands the receptive field of the network, not only taking the channel and spatial dimension into account, but also promoting the feature extraction ability of the network.

GCYTD algorithm can improve the convergence speed, the detection accuracy and speed. The training time has been sharply shortened, and the model is lighter. As shown in Fig. 4, the main functional modules of the tongue detection and location framework include four parts:

1. CBL (Convolution, Batch Normalization and Leaky-ReLU): It is a module composed of a convolution layer, a batch normalization layer and a Leaky-ReLU activation function.

2. CBG (Convolution, Batch Normalization and GELU): It is a module composed of a convolution layer, batch normalization layer and GELU activation function. CBL module and CBG module are all for feature extraction, the difference between them is that the activation function of CBG uses GELU instead of Leaky-ReLU. CBG module can reduce the number of parameters and accelerate the training speed of the model.
3. CSP (Center and Scale Prediction): The maximum pooling layer, CBG and RESn (n×Residual Unit) modules are embedded in CSP. By dividing the low-level features into two parts and combining cross-level features to enhance the learning ability of the model, more dimensional feature information can be obtained.
4. CA: Includes two attention blocks: Coordinate Information Embedding and Coordinate Attention Generation, the two attention blocks focus on the channel information and the coded spatial location information respectively [25]. In this way, more accurate tongue information can be obtained.

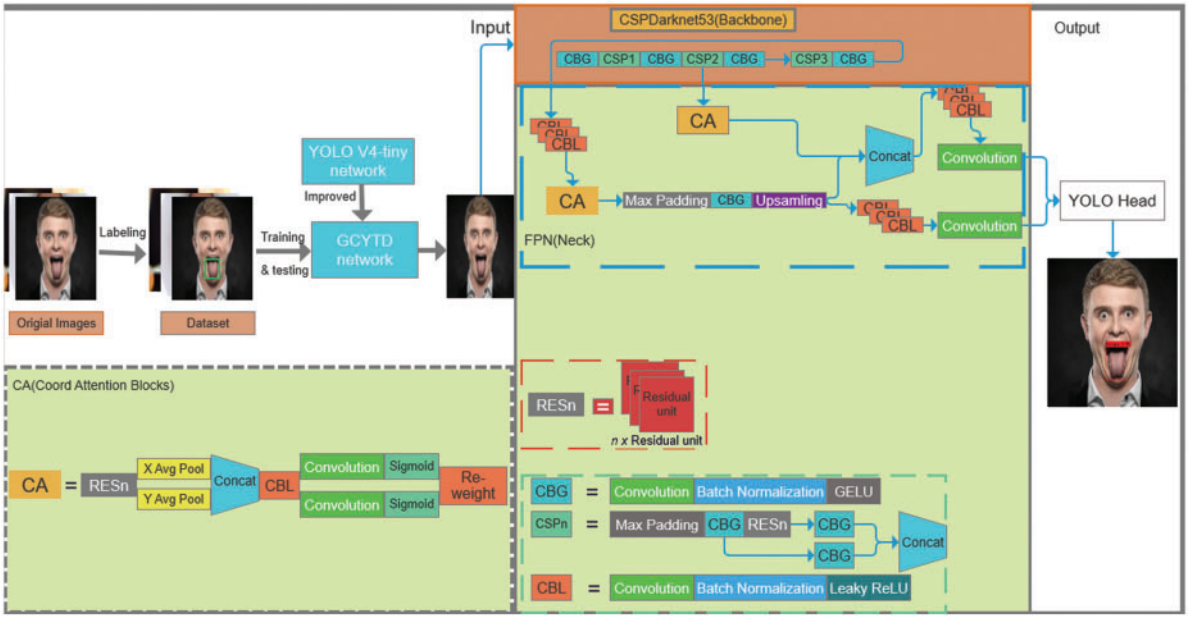

Figure 4: Tongue detection and location framework

GCYTD algorithm consists of four parts: CSPDarknet53-tiny (Cross Stage Partial Dark Network), CA blocks, FPN and YOLO Head. Fig. 5 shows the whole diagram of GCYTD algorithm: Firstly, the preliminary feature information is obtained from the input image by the CSPDarknet53-tiny module, then the information is sent to the CA blocks module to extract the information of channel dimension and spatial dimension, then the information is sent to FPN to obtain the intermediate feature maps; then use the maps to obtain a higher-dimensional feature information map by up-sampling and convolution fusion. After that, non-maximum suppression (NMS) [26] is used to remove the redundant detection frames so that each target has a unique detection frame, which makes the position information more accurate and the credibility higher. Finally, the result is output by YOLO Head.

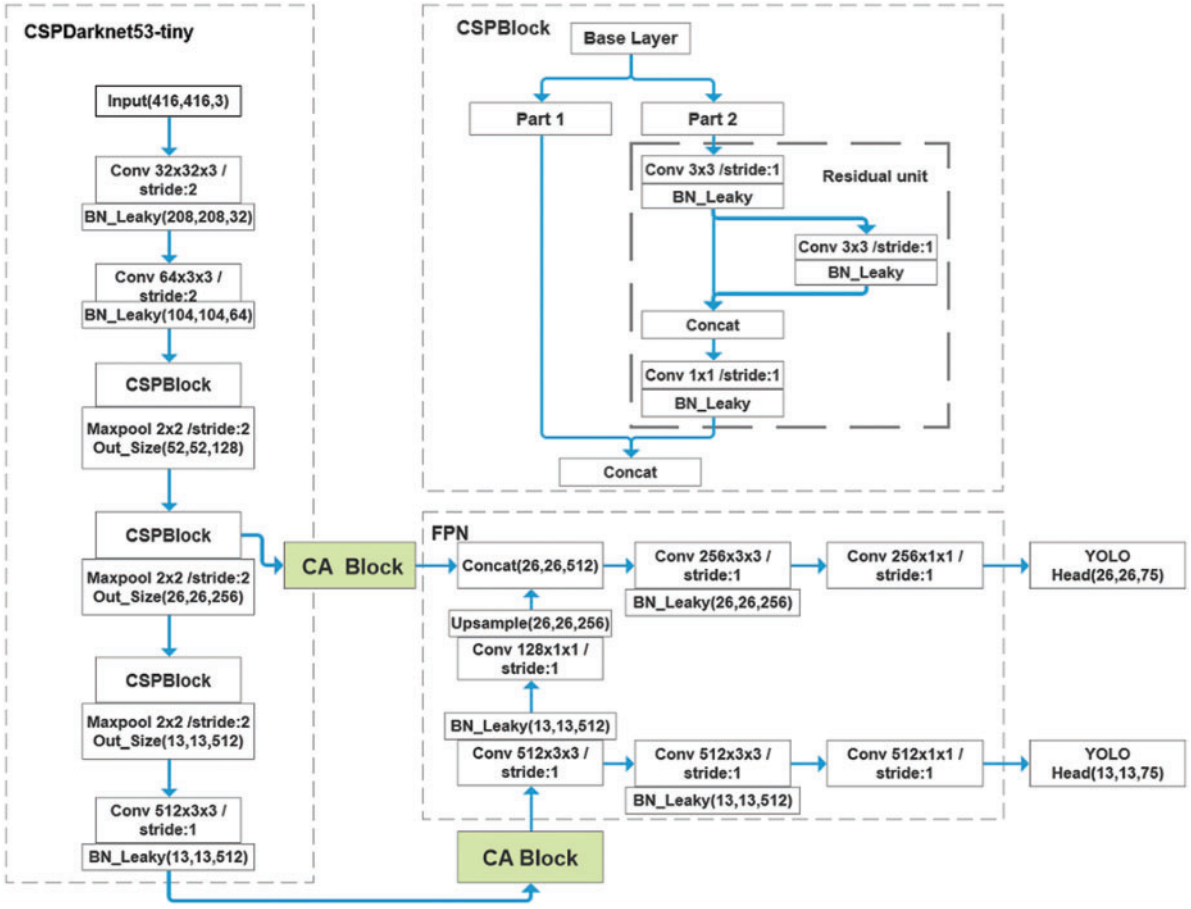

Figure 5: GCYTD algorithm framework

The loss function of GCYTD algorithm is as same as that of YOLO V4-tiny network model, including three parts of loss, and its formula is expressed as Eq. (8):

$$loss\,(o, c, O, C, l, g) = \lambda_1 loss_{conf}\,(o, c) + \lambda_2 loss_{cla}\,(O, C) + \lambda_3 loss_{loc}\,(l, g) \tag{8}$$

$loss_{conf}\,(o, c)$, $loss_{cla}\,(O, C)$ and $loss_{loc}\,(l, g)$ respectively denote the confidence loss function, classification loss function, and location loss function. λ_1, λ_2, and λ_3 are the balance coefficients.

Fig. 6 shows the comparison of total loss and validation loss of YOLO V4-tiny network model and GCYTD algorithm in the training set and test set. In Fig. 6, at the initial stage of tongue detection model training, the model learning efficiency of GCTYD is high, the slope of the training curve drops almost in a straight line, and the convergence speed is fast. GCYTD's total loss is 1.0 smaller than YOLO V4-tiny's, and GCYTD's validation loss is 0.5 smaller than YOLO V4-tiny's. When the number of iterations increases, the convergence speed of the loss function gradually slows down. When the number of iterations reaches to 500 epochs, the model learning efficiency gradually reaches saturation, and the loss function curve finally maintains a fluctuation at 0.6. Compared with the YOLO V4-tiny network model, GCYTD algorithm is more stable during the training process, especially in the later training stage, the minimum total loss value of GCYTD algorithm achieved by training is also about

0.4 smaller than that of the former. With better stability, GCYTD algorithm's validation loss does not change significantly, and the convergence effect and speed are better.

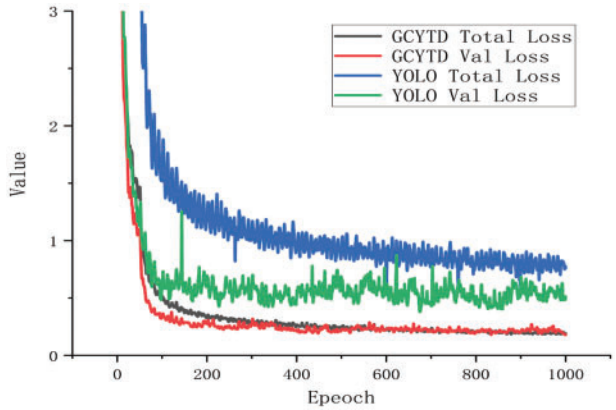

Figure 6: Loss function graph

3 Experiment and Analysis

3.1 Data Set Construction

In cooperation with the First Affiliated Hospital of Hunan University of Chinese Medicine, 1340 clinical tongue images were collected by the professional acquisition equipment. In order to enhance data integrity, another 780 tongue images were collected in natural environment, and a total of 2120 tongue images were collected, as shown in Tab. 1. The resolution of the images collected in standard environment is 5568 × 3712 and the other images do not have uniform size, which may increase the difficulty and the duration of network training. So, the resolution of all images is set as 416 × 416 firstly, which can be directly input into the neural network.

Table 1: Tongue image data set table

Acquisition amount	Acquisition environment	Initial resolution	Final resolution
1340 images	Standard environment	5568 × 3712	416 × 416
780 images	Natural environment	None	416 × 416

The tongue images are labelled by an open-source image standard software named LabeLImg, and the position information of the tongue is obtained as Ground truth [27]. 80% (1696 images) of the data set are used as training data, and the remaining 20% (424 images) are used as test data. Fig. 7 represents the image collected in standard environment. Fig. 8 represents the tongue image collected in natural environment. Fig. 9 is the labelled image of Fig. 7. Fig. 10 is the labelled image of Fig. 8.

Figure 7: Tongue image collected in standard environment

Figure 8: Tongue image collected in natural environment

Figure 9: Labelled image of Fig. 7

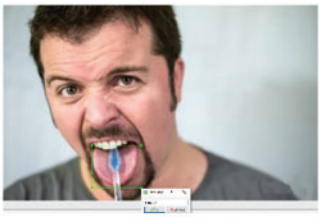

Figure 10: Labelled image of Fig. 8

The training parameter settings are shown in Tab. 2. According to the image characteristics of the tongue image and GPU (Graphics Processing Unit) performance, when Batch size is set as 32, it can get the best training and test effects.

Table 2: Training parameter setting table

Parameters	Value
Input size	416×416
Learning rate	1.1×10^{-4}
Batch size	32
Minbatch size	16
Classes	1
Epoch	100

3.2 Evaluation Index

The evaluation indexes of deep learning are used to evaluate the experimental results in this paper: *Precision, Recall,* F_1*, AP* (*Average Precision*)*, mAP* (*mean Average Precision*)*, Detection Speed, Model Size* [28]. The combination of prediction category and true category of the neural network model is classified as true positive (*TP*), false negative (*FN*), true negative (*TN*), and false positive (*FP*). For example, to discriminate tongue or non-tongue, if the true category is tongue and the prediction category is also tongue, the result is set as *TP*; if the true category is tongue and the prediction category is non-tongue, the result is set as *FN*. The specific formulae are shown in Eqs. (9)–(13). In Eq. (11), it can be seen that F_1 is related to *Precision* and *Recall*, and the higher F_1 is, the better the model effect is.

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precison + Recall} \tag{11}$$

$$AP = \int_0^1 P(r)\, dr \tag{12}$$

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{13}$$

3.3 Analysis of Experimental Results

In order to verify the effect of the proposed algorithm, six classical target detection algorithms are introduced to compare with the proposed algorithm, Faster R-CNN (Region-Convolutional Neural Network) [29], YOLO V4 [30], YOLO V4-tiny [31], EfficientDet-D0 [32], SSD 300 (Single Shot MultiBox Detector) [33] and YOLO V3 [34,35]. The backbones of these algorithms are ResNet50 (Residual Network), CSPDarknet53, CSPDarknet53-tiny, EfficientNet-B0, Vgg16, and Darknet53, respectively. The model is trained on the training set and the performance is tested on the test set. The test results are shown in Tab. 3.

Table 3: Detection results of different target detection algorithms

Algorithms	Precision (%)	Recall (%)	mAP (%)	F_1 (%)	AP (%)	Detection speed (f/s)	Model size (MB)
Faster R-CNN	81.09	88.25	85.10	84.51	85.10	13.74	108.16
YOLO V3	95.91	93.11	97.81	94.00	97.81	6.84	236.32
YOLO V4	99.93	91.23	**99.85**	**99.90**	**99.85**	7.56	244.39
SSD 300	67.78	87.12	76.24	84.32	84.12	12.04	90.47
EfficientDet-D0	**99.93**	**99.78**	99.80	**99.90**	99.80	1.08	**14.94**
YOLO V4-tiny	95.71	74.53	83.43	84.00	83.43	7.03	70.82
Ours (100 epoch)	98.09	96.24	97.43	97.00	97.43	25.97	**22.63**
Ours(1000epoch)	98.31	96.87	98.89	98.00	98.89	**27.09**	**22.63**

In Tab. 3, In term of *Precision*, *Recall* and *Model Size*, EfficientDet-D0 achieves the maximum value, but its *Detection Speed* is too slow, only about 1 f/s, which may cause data distortion; In term of *mAP*, F_1, and *AP*, YOLO V4 achieves the maximum value, but its *Model Size* is too big which is not suitable for deploying on mobile terminals or smart tongue diagnostic devices. The proposed algorithm achieves the maximum value in term of *Detection Speed.* When the number of the training iterations increases to 1000 epochs, the *mAP* value of the proposed algorithm can achieve 98.89%, which is 1.46% higher than the number of 100 epochs, and the *Detection Speed* can also reach to 27.09 f/s, improving more than 1 f/s.

Through the comprehensive analysis of Tab. 3, compared with the other six algorithms, the proposed algorithm can detect the tongue quickly and the model is more lightweight while ensuring the precision. The *Precision* of the proposed algorithm could meet the precision requirement of tongue detection in complex natural environment, and the *Detection Speed* could also meet the requirement of real-time detection, better than all of the others.

Fig. 11 shows the detection results on different images including four types of tongue images: Standard environment, with occlusion, multiple-target, and dark light environment. The red rectangle shows the probability of the tongue position. a1–a4, b1–b4, c1–c4, d1–d4, e1–e4, f1–f4, g1–g4 are the detection results of Faster R-CNN, YOLO V3, YOLO V4, SSD 300, EfficientDet-D0, YOLO V4-tiny, and ours, respectively.

In Fig. 11, Figs. 11a1–11a4 show that Faster R-CNN can recognize the single-target tongue in standard environment and the tongue with occlusion, but fail to detect the multi-target tongue and the tongue in dark light environment. Figs. 11b1–11b4 show that YOLO V3 has good performance for tongue detection in standard environment, with occlusion, and with multiple-target, but it can't detect the tongue in dark light environment; Figs. 11c1–11c4 show that YOLO V4 can detect the tongue in standard environment and that with occlusion, partially detect the multi-target tongue, and fail to detect the tongue in dark light environment. Figs. 11d1–11d4 show that SSD-300 only has good performance for tongue detection in standard environment; Figs. 11e1–11e4 show that EfficientDet-D0 has good performance for tongue detection in standard environment and that with occlusion, partially detect the multi-target tongue, and fail to detect tongue in dark light environment. Figs. 11f1–11f4 show that YOLO V4-tiny achieves good performance for tongue detection in standard

environment, with occlusion, and with multiple-target, but fails in dark light environment. Figs. 11g1–11g4 show that our method has the best detection performance for all types, especially for the image in dark light environment, which has obvious advantages over the other six methods.

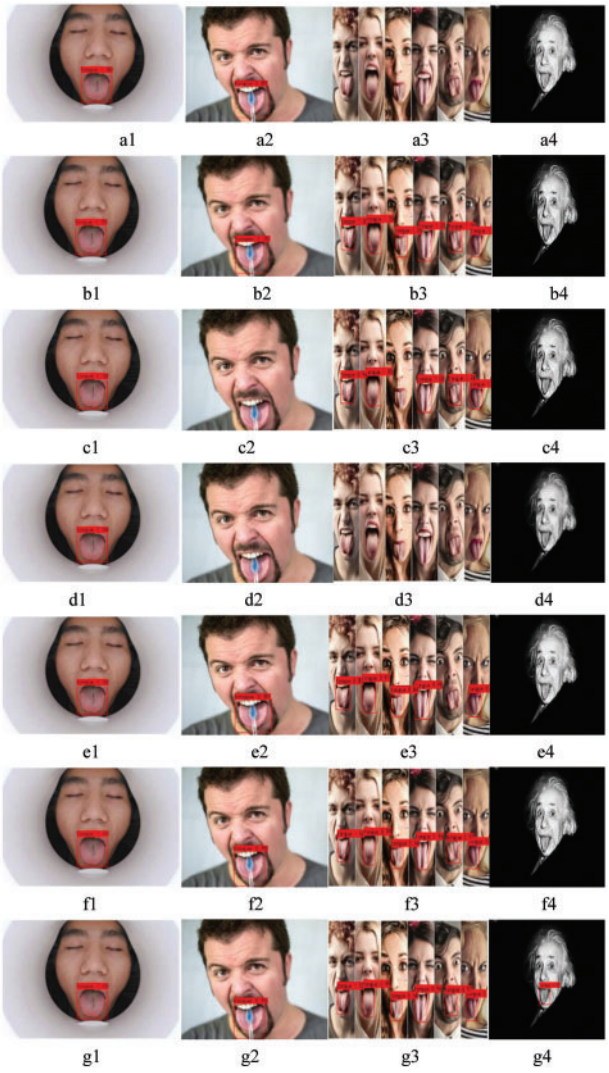

Figure 11: Detection results of tongue

Note: a: Faster R-CNN, b: YOLO V3, c: YOLO V4, d: SSD 300, e: EfficientDet-D0, f: YOLO V4-tiny, g: Ours Col 1: Standard environment, Col 2: With occlusion, Col 3: Multiple-target, Col 4: Dark light environment.

3.4 Ablation Experiment

To further evaluate the performance of the proposed algorithm. Four algorithms based on the YOLO V4-tiny network model are compared including the original model, model with GELU activation function, model with CA mechanism, and model with GELU activation function and CA mechanism (ours). The detection results are shown in Tab. 4. It shows that the model with GELU activation function and the model with CA mechanism improve the performance comparing with the original one. And our algorithm is the best.

Table 4: Evaluation index of detection results of different algorithms

Algorithm	Precision (%)	Recall (%)	mAP (%)	F1(%)	AP (%)	Detection speed (f/s)	Model size (MB)
YOLO V4-tiny	95.71	74.53	83.43	84.00	83.43	7.03	70.82
YOLO V4-tiny+GELU	96.14	93.53	95.97	95.00	95.97	6.72	**22.42**
YOLO V4-tiny+CA	96.59	94.36	96.59	96.00	96.00	26.02	22.5
Ours	**98.31**	**96.87**	**98.89**	**98.00**	**98.89**	**27.09**	22.63

Fig. 12 shows the detection results of four algorithms based on the YOLO V4-tiny network model. a1-a6, b1-b6, c1-c6, d1-d6 respectively represent YOLO V4-tiny (named A1), YOLO V4-tiny+GELU (named A2), YOLO V4-tiny+CA (named A3), and ours. The red rectangle shows the probability of the tongue position. Fig. 12 shows that the four algorithms can accurately detect the position of the tongue with single target in standard environment and that with occlusion. A2 algorithm and A3 algorithm fail to detect the position of the tongue with multiple-target in b3 and c3. For the detection of multiple-tongue images, the A2 algorithm can detect only 4 tongues in b4, and the other algorithms can detect 5 tongues; For the tongue image in dark light environment, all the algorithms except A1 can accurately locate the position of the tongue. For all types of tongue images, ours has the best performance.

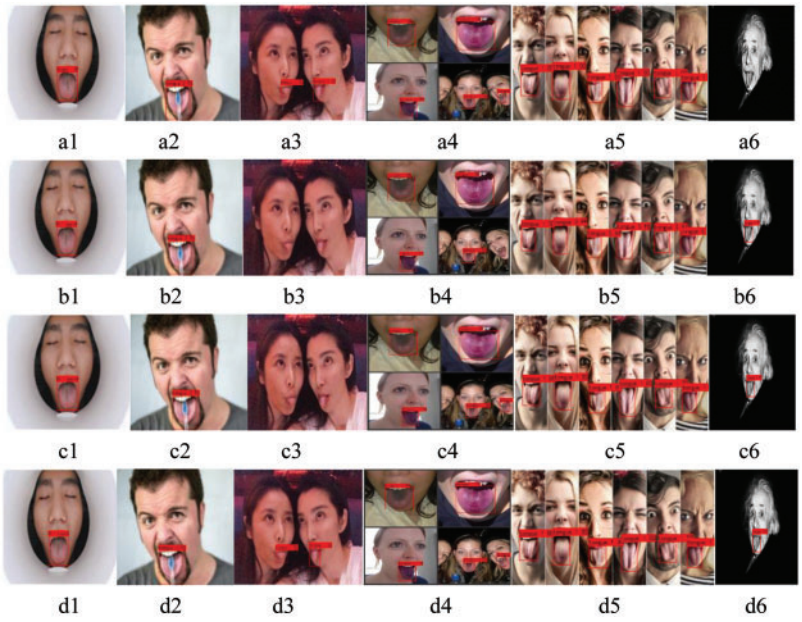

Figure 12: Detection results of tongue

Note: a: YOLO V4-tiny, b: YOLO V4-tiny+GELU, c: YOLO V4-tiny+CA, d: Ours Col 1: Standard environment, Col 2: With occlusion, Col 3-5: Multiple-target, Col 6: Dark light environment

4 Conclusion

By incorporating the GELU activation function and the CA mechanism into YOLO V4-tiny network model, a new tongue detection algorithm named GCYTD is proposed in this paper. It

expands the receptive field of the network model, not only considering the information of the channel and spatial dimension, but also improving the feature extraction ability of the network model. It greatly reduces the amount of model parameters and solves the problem of gradient explosion and gradient disappearance, which makes the detection faster and the model lighter. Experimental results show that in a complex natural environment, GCYTD algorithm has better performance than the other classical detection algorithms. It is not susceptible to uncertainties such as light source color temperature, light intensity, shooting angle, equipment differences, etc. As the model is lighter, it's easier to deploy the tongue detection models on small mobile terminals and to develop the intelligent tongue diagnosis instruments.

Funding Statement: This work was supported by the Key Research and Development Plan of China (No. 2017YFC1703306), Key Project of Education Department in Hunan Province (No. 18A227), Key Project of Traditional Chinese Medicine Scientific Research Plan in Hunan Province (2020002).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Liu, X. Y. Wang and L. Zhou, "Study on extraction and recognition of traditional Chinese medicine tongue manifestation: Based on deep learning and migration learning," *Journal of Traditional Chinese Medicine*, vol. 60, no. 10, pp. 835–840, 2019.

[2] L. Liu, O. L. Wan, X. G. Wang, P. Fei, J. Chen *et al.,* "Deep learning for generic object detection: A survey," *International Journal of Computer Vision (IJCV)*, vol. 128, no. 2, pp. 261–318, 2020.

[3] J. Wang, T. Zhang, Y. Cheng and N. Al-Nabhan, "Deep learning for object detection: A survey," *Computer Systems Science and Engineering*, vol. 38, no. 2, pp. 165–182, 2021.

[4] W. J. Tang, Y. Gao, L. Liu, T. W. Xia and Q. Xu, "An automatic recognition of tooth- marked tongue based on tongue region detection and tongue landmark detection via deep learning," *IEEE Access*, vol. 8, pp. 153470–153478, 2020.

[5] M. Z. M. Shamim and S. Syed, "Automated detection of oral pre-cancerous tongue lesions using deep learning for early diagnosis of oral cavity cancer," arXiv:1909.08987v1, 2019.

[6] J. Lu, Z. T. Yang, K. Z. Okkelberg and M. Ghovanloo, "Joint magnetic calibration and localization based on expectation maximization for tongue tracking," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 1, pp. 52–63, 2018.

[7] Y. Xin, Y. Cao and Z. Liu, "Automatic tongue verification based on appearance manifold learning in image sequences for the internet of medical things platform," *IEEE Access*, vol. 6, pp. 43885–43891, 2018.

[8] C. G. Zhou, H. Y. Fan and Z. Y. Li, "Tonguenet: Accurate localization and segmentation for tongue images using deep neural networks," *IEEE Access*, vol. 7, pp. 148779–148789, 2019.

[9] Y. Hu, G. H. Wen, M. N. Luo, P. Yang, D. Dai *et al.,* "Fully-channel regional network for disease-location recognition with tongue images," *Artificial Intelligence in Medicine*, vol. 118, no. 102110, pp. 1–13, 2021.

[10] F. Zheng, X. Y. Huang, B. L. Wang and Y. H. Wang, "A method for tongue detection based on image segmentation," *Journal of Xiamen University (Natural Science)*, vol. 55, no. 6, pp. 895–900, 2016.

[11] M. H. Tania, K. Lwin and M. A. Hossain, "Advances in automated tongue diagnosis techniques," *Integrative Medicine Research*, vol. 8, no. 1, pp. 42–56, 2019.

[12] V. Thanikachalam, S. Shanthi, K. Kalirajan, S. Abdel-Khalek and M. Omri, "Intelligent deep learning based disease diagnosis using biomedical tongue images," *Computers Materials & Continua*, vol. 70, no. 3, pp. 5667–5681, 2022.

[13] Q. Liu, X. Y. Huang, B. L. Wang and L. H. Wang, "A method for color cast detection and color correction of tongue inspection images under natural environment," *Journal of Xiamen University (Natural Science)*, vol. 55, no. 2, pp. 278–284, 2016.

[14] Y. M. Wang, K. B. Jia and P. Y. Liu, "Impolite pedestrian detection by using enhanced YOLOV3-tiny," *Journal on Artificial Intelligence*, vol. 2, no. 3, pp. 113–124, 2020.
[15] Y. Ding and Z. Fu, "Multi-UAV cooperative GPS spoofing based on YOLO nano," *Journal of Cyber Security*, vol. 3, no. 2, pp. 69–78, 2021.
[16] S. Albahli, N. Nida, A. Irtaza, M. Haroon and M. T. Mahmood, "Melanoma lesion detection and segmentation using YOLOV4-darkNet and active contour," *IEEE Access*, vol. 8, pp. 198403–198414, 2020.
[17] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *IEEE Proc. CVPR*, Honolulu, Hawaii, USA, pp. 7263–7271, 2017.
[18] Q. Xu, Y. Zeng, W. J. Tang and W. Peng, "Multi-task joint learning model for segmenting and classifying tongue images using a deep neural network," *Journal of Biomedical and Health Informatics*, vol. 24, no. 9, pp. 2481–2489, 2020.
[19] Y. Li, H. Wang, L. M. Dang and H. Moon, "A deep learning-based hybrid framework for object detection and recognition in autonomous driving," *IEEE Access*, vol. 8, pp. 194228–194239, 2020.
[20] H. X. Fu, G. Q. Song and Y. C. Wang, "Improved YOLOV4 marine target detection combined with CBAM," *Symmetry*, vol. 13, no. 4, pp. 623–623, 2021.
[21] S. Evan, J. Long and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 6, pp. 640–651, 2017.
[22] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUS)," arXiv: 1606.08415v4, 2016.
[23] W. Sun, G. Z. Dai, X. R. Zhang, X. Z. He and X. Chen, "TBE-net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021. https://doi.org/10.1109/TITS.2021.3130403.
[24] H. P. Wu, Y. L. Liu and J. W. Wang, "Review of text classification methods on deep learning," *Computers, Materials & Continua*, vol. 63, no. 3, pp. 1309–1321, 2020.
[25] Q. Hou, D. Zhou and J. Feng, "Coordinate attention for efficient mobile network design," in *IEEE Proc. CVPR*, online, pp. 13713–13722, 2021.
[26] T. Q. Nguyue, S. H. Kim and I. S. Na, "An efficient non-maximum suppression for pedestrian detection using mean-shift algorithm and linear SVM classifier," *Journal of KIISE: Computing Practices and Letters*, vol. 20, no. 2, pp. 111–115, 2014.
[27] J. G. Yang, J. Fan, W. W. Zhe, G. L. Li, T. Y. Liu *et al.,* "Cost-effective data annotation using game-based crowdsourcing," in *Proc. VLDB*, Munich, Germany, vol. 12, no. 1, pp. 57–70, 2018.
[28] J. F. Qiu, Q. H. Wu, G. R. Ding, Y. H. Xu and S. Feng, "A survey of machine learning for big data processing," *ERUASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, pp. 1–16, 2016.
[29] S. Q. Ren, K. M. He, G. H. Ross and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 39, no. 6, pp. 1137–1142, 2017.
[30] A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao, "YOLOV4: Optimal speed and accuracy of object detection," arXiv: 2004.10934, 2020.
[31] Z. Jiang, L. Zhao, S. Li and Y. Jia, "Real-time object detection method based on improved YOLOV4-tiny," arXiv: 2011.04244v2, 2020.
[32] M. X. Tan, R. M. Pang and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *IEEE Proc. CVPR*, online, pp. 10778–10787, 2020.
[33] G. H. Yu, H. H. Fan, H. Y. Zhou, T. Wu and H. J. Zhu, "Vehicle target detection method based on improved SSD model," *Journal on Artificial Intelligence*, vol. 2, no. 3, pp. 125–135, 2020.
[34] W. Sun, L. Dai, X. R. Zhang, P. S. Chang and X. Z. He, "RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring," *Applied Intelligence*, pp. 1–16, 2021.
[35] Q. Liu, S. Lu and L. Lan, "YOLOV3 attention face detector with high accuracy and efficiency," *Computer Systems Science and Engineering*, vol. 37, no. 2, pp. 283–295, 2021.