YOLO-Ti: an efficient object detection approach for tiny facial markers

Ying Li, Dongdong Weng*, Zeyu Tian, Jing Hou, Zihao Li
Beijing Engineering Research Center of Mixed Reality and Advanced Display, School of Optics and Photonics, Beijing Institute of Technology, Beijing 100081, China

ABSTRACT

In this paper, an efficient object detection method YOLO-Ti is proposed to detect tiny facial markers. Our study is driven by the practical requirements of 3D face modeling, requiring the incorporation of as many facial features as possible for reference. This research can even provide information for facial expression recognition and joint deformation. To achieve this, we first present a feature fusion module called Cross-BiFPN, which incorporates additional cross-connecting branches between different network layers to utilize low-level features more effectively. Secondly, we add a high-resolution detection head and attention module to the YOLOv8 model to improve the ability of detecting tiny objects, while at the same time ensuring the lightweight detection model by reducing redundant network layers. Thirdly, we collect a dataset of facial markers with an average size much smaller than publicly available small object datasets. Ablation studies and comparison experiments are conducted to evaluate the performance of our approach. Compared with the baseline YOLOv8 model, YOLO-Ti shows a 30.4% improvement in mAP50 while reducing model parameters by 65.1%. The automatic feature extraction provided by our model facilitates the construction of digital humans, providing significant savings in manpower and time for modelers.

Keywords: Facial markers, tiny object detection, improved YOLOv8 algorithm, 3D face reconstruction

1. INTRODUCTION

Recent years have seen a growing focus on creating naturally realistic and intelligent digital humans for AR and VR applications, particularly within the gaming and film industries. Despite the availability of convenient and automated solutions for constructing digital avatars for the general public[1-5], in pursuit of higher precision, many studios still choose high-resolution multi-view images as input to reconstruct 3D digital facial models[6-8]. The process often involves manually drawing markers on a real person's face to incorporate additional facial features, facilitating multi-view photo alignment[9-11]. Moreover, marker information can also be used to assist in subsequent steps of digital human generation, such as rigging and animation driving[12,13].

However, the challenge arises due to the large number of manually drawn facial markers, coupled with their small size. In a 1024×1024 pixel facial image obtained from shooting, for example, the size of these markers is often less than 10 pixels. Comparatively, the definition of a small target is based on the area ratio of the target bounding box to the background image, falling between 0.08% and 0.58%[14]. While the area ratio of these markers to the face is mostly below 0.01%. In order to create high-precision digital avatars, some studios currently employ a manual solution to label the location of these markers on the image one by one. It is a time-consuming and labour-intensive process. Therefore, we are exploring the use of algorithms to automatically label these markers, providing supplementary feature information for 3D reconstruction.

Most current object detection algorithms are designed for large and medium-sized objects, which are difficult to adapt to the task of detecting tiny objects such as facial markers. Directly applying existing algorithms to this task is evidently inappropriate. Therefore, this paper proposes an object detection framework YOLO-Ti to detect tiny facial markers. Our contributions are listed as follows:

- A feature fusion module Cross-BiFPN is proposed, which incorporates additional cross-connecting branches between different network layers to make better use of low-level features.

*crgj@bit.edu.cn

International Conference on Optics, Electronics, and Communication Engineering (OECE 2024), edited by Yang Yue, Proc. of SPIE Vol. 13395, 133952B · © 2024 SPIE · 0277-786X · Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.3048940

- We introduce a high-resolution detection head and explore schemes for lightweighting models, which simultaneously ensures high accuracy in detecting tiny objects while using fewer parameters.
- We build a dataset of facial markers with an average size much smaller than publicly available small object datasets. Ablation studies and comparison experiments are conducted to demonstrate the capabilities of our method.

2. RELATED WORKS

Object detection is a crucial area of computer vision research, serving as the basis for various complicated visual tasks and finding applications across industries such as agriculture and manufacturing. In recent years, although there have been great advancements in object detection due to the accelerated development of deep learning, most of the existing algorithms are primarily designed to detect large and medium-sized objects. Unfortunately, there are few models specifically tailored for tiny objects. Furthermore, tiny objects present unique challenges, being inherently diminutive with inconspicuous features. These characteristics contribute to the poor performance of existing algorithms. Consequently, improving the effectiveness of tiny object detection remains a challenging and critical study focus[15].

In order to enrich the dataset and introduce more small-scale objects, YOLOv4[16] proposed the Mosaic data enhancement method. This method reads four different images simultaneously and stochastically splices them using flipping, scaling, and cropping. However, the method has some drawbacks: the use of splicing disrupts the contextual information of the original image. For facial marker data, where markers are concentrated in the facial region at the center of the image, the Mosaic method disrupts the distribution of markers. Moreover, scaling and cropping may further reduce the size of objects in the image, resulting in a poorer ability of the detection model.

Haris, Shakhnarovich, and Ukita[17] introduced an end-to-end trained super-resolution approach based on the Faster R-CNN[18] network to address low-resolution regions, enhancing the detection performance of small targets. Nevertheless, the deep network structure of Faster R-CNN poses challenges in extracting features for dense objects after aggregation. YOLO-Z[19] improves on the YOLOv5[20] model by replacing the Path Aggregation Network (PANet) structure[21] of the neck network with Bidirectional Feature Pyramid Network (BiFPN)[22], enhancing feature fusion capability. ACAM-YOLO[23] addresses the challenge posed by a large proportion of objects and partial scene occlusion in aerial object detection. The Adaptive Co-Attention Module (ACAM) is integrated into both YOLOv5's backbone network and feature fusion network to facilitate efficient feature extraction. However, ACAM-YOLO adds the ACAM module after different feature maps to detect various sizes of targets, including large, medium and small. It increases computational effort and does not specifically give more attention to small objects.

Akyon, Altinuc, and Temizel[24] proposed Slicing-Aided Hyper Inference (SAHI), a general framework applicable to any object detector. In this approach, a high-resolution image is sliced into small localized images during the inference stage, and the results are then combined after detecting the localized images separately. Although it shows high accuracy in detecting small objects, it is essentially a data preprocessing method and does not alter the detection capability of the original model.

Based on the above discussion, it is found that most existing detection algorithms are designed to enhance the detection of medium and small objects, which are usually evaluated for algorithmic performance on aerial datasets such as VisDrone[25-28]. There are fewer models specifically improved for detecting tiny targets, such as facial markers.

3. METHODOLOGY

3.1 The additional high-resolution head

YOLOv8[29] architecture consists of three main components: the backbone network, neck network, and head network. In the backbone network, the input image undergoes five CBS blocks to obtain a 32-fold downsampled feature map. As shown in Figure 1, assuming an input size of 640×640, the three feature maps passed to the neck network have resolutions of 80×80, 40×40 and 20×20. While multi-scale feature fusion is implemented in the neck network through the PANet module, it does not alter the scale of feature maps. However, in tasks involving the detection of tiny objects, the targets are often very small. For instance, our self-constructed face dataset used in this paper includes many tiny markers with an average size of less than 7×7 pixels. Due to numerous down-sampling and pooling operations, most features of such objects are lost, making them challenging to detect[30].

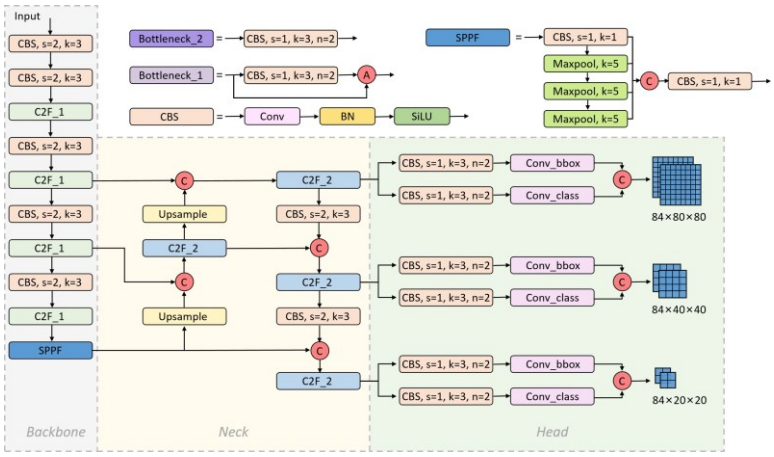


Figure 1. Overall structure of YOLOv8.

To ensure effective detection of the aforementioned tiny objects, we introduce a new detection head onto YOLOv8, utilizing the underlying features of layer P_2. The structure is illustrated in Figure 2. This high-resolution feature map has a scale of 160×160 pixels and preserves more low-level feature information, including details about tiny objects, as only two down-sampling operations are performed in the backbone network.

In the PANet module of the neck network, specifically, the top-down paths also incorporate a feature map H_2, matching the scale of the layer P_2 in the backbone network. These maps are used for feature fusion through concatenation and are subsequently output. Alongside the original three detection heads, this addition effectively mitigates the negative impact caused by scale differences. The resolutions of the four-level feature maps output via H_2, H_3, H_4, and H_5 are 160×160, 80×80, 40×40, and 20×20. Despite the additional computational and memory overhead introduced by the extra detection head, it plays a crucial role in enhancing the model's detection capability for tiny objects. We conduct experiments in Section 4 to demonstrate its effectiveness.

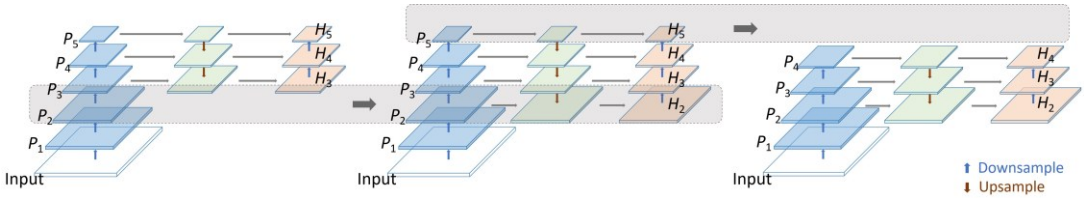


Figure 2. Adjustment of the model structure, involving the removal of the detection head H_5 designed for large objects and the addition of H_2, specifically for tiny objects.

Considering both the model's detection capability and computational efficiency, we opt to remove the detection head H_5 designed for large objects, while incorporating H_2 specifically for tiny ones. The output feature map of H_5 has a resolution of 20×20 pixels, initially intended for detecting large-sized objects in the YOLOv8. Increasing the depth of the network generally improves the capability to detect objects, but for the specific task of detecting small objects, there is little benefit in simply increasing the depth of the network. Although features from the lower layers are fused in the bottom-up paths of the neck network, after multiple down-sampling and pooling operations, the top output layer essentially excludes information about tiny object features. Additionally, through experimentation, we find that retaining or discarding the top layer has little impact on detection effectiveness for tiny objects such as facial markers. Therefore, to reduce the parameters, we remove layers H_5 and P_5, essentially shifting the entire model structure down by one layer, as depicted in Figure 2. The final output of the three-level feature map resolutions is 160×160, 80×80, and 40×40.

3.2 Cross-BiFPN module

Feature fusion is a widely used technique in object detection, entailing the integration of extracted low-level features--such as stripe shape, object contour, and pixel distribution—with high-level abstract semantic information. To alleviate the impact of tiny object characteristics on algorithms, many researchers have refined the architecture of the feature fusion module, aiming at enhanced results. One of the most notable structures is the Feature Pyramid Network (FPN)[31]. After layer-by-layer feature extraction from the input image, an additional top-down path with multiple up-sampling

operations efficiently merges information from different layers. The combined information is then fed into the prediction head to produce the final results. However, a limitation of FPN is its single top-down path, posing challenges in efficiently transmitting bottom-layer information to the last layer.

The YOLOv8 model adopts the PANet structure, which includes an extra bottom-up path, facilitating the transmission of information from the bottom layer upwards. However, it lacks a direct combination of features extracted from the backbone with the additional bottom-up path. BiFPN is a simple and efficient multi-scale feature fusion method. It removes PANet nodes that contribute less to feature extraction and introduces skipping connections between output and input nodes at the same scale. This ensures a comprehensive fusion of multi-scale features in each layer while significantly preserving the original features.

For the tiny object detection task, to fully utilize the low-layer information from the backbone network, we propose the Cross-BiFPN module. In comparison to the BiFPN module, this approach does not use direct skips between output and input nodes of the same scale. Instead, it establishes a cross-skipping connection between the input nodes of the lower layer of the backbone network (I_1, I_2 and I_3) and the output nodes of the higher layer of the head network (H_2, H_3 and H_4). These cross-skipping connections ensure that feature information related to tiny objects is better preserved. Since feature maps of different layers have distinct sizes, the outputs of I_1, I_2 and I_3 also need to undergo a down-sampling operation before connecting to H_2, H_3 and H_4, as shown in Figure 3.

In the BiFPN module, the fused feature of two feature maps is computed by adding their weights, which may overcompress channels when dealing with a large number of model branches. Therefore, in order to maximize the richness of detail and semantic information in the output features and preserve features for optimal detection performance, we discard the idea of adding and instead adopt concatenation for feature fusion. Taking the output of the H_3 as an example, the calculation process is as follows:

$$\begin{cases} H_2^{\text{out}'} = \text{Upsample}\left(H_2^{\text{out}}\right) \\ D_2^{\text{out}} = \text{Downsample}\left(I_2^{\text{out}}\right) \\ H_3^{\text{out}} = \text{Conv}\left(\text{Concat}\left(D_2^{\text{out}} + N_3^{\text{out}} + H_2^{\text{out}'}\right)\right) \end{cases} \tag{1}$$

where Downsample( ) represents the max-pooling operation, Concat( ) denotes concatenation of all input features along channel dimensions.

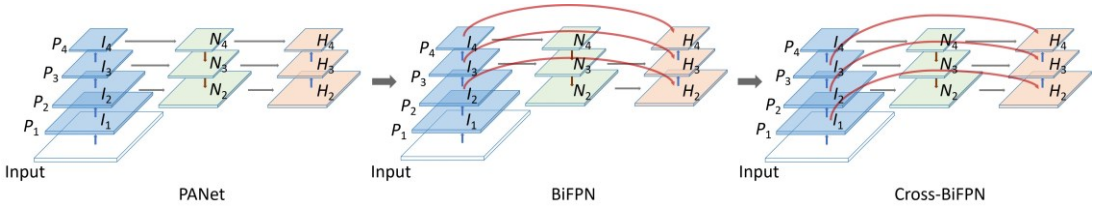


Figure 3. Structure of the proposed Cross-BiFPN module.

3.3 Self attention mechanism

Squeeze and Excitation (SE)[32] is a lightweight attention module designed primarily to address the issue of the varying importance of different channels in the feature map. In the conventional process, researchers typically assign the same weight to all channels of the feature map. However, the importance of different channels varies. In the case of a given feature map, the SE attention module derives the attention mapping for channel dimensions and then multiplies the input feature with the attention mapping. The detailed process is illustrated in Figure 4.

In the initial step, the input feature map of size $H\times W\times C$ undergoes compression, utilizing global average pooling in the spatial dimension to obtain a feature map of $1\times 1\times C$. Excitation stage involves two Fully-Connected (FC) layers. The first FC layer reduces channels as C/r, followed by the application of the ReLU activation function. Subsequently. The second FC layer elevates the feature map back to C channels. Afterwards, it introduces the Sigmoid activation function to generate attention weights for channels, maintaining a size of $1\times 1\times C$. This approach benefits from additional nonlinear processes, accommodating complex correlations between channels. The hyperparameter r represents the ratio of channel compression, with a default value of 16. Finally, the original input feature map is multiplied channel-by-channel with the channel attention weights, resulting in the final output map of size $H\times W\times C$.

According to Reference[32], the SE module was integrated into distinct models for different datasets and tasks, and the model performances were improved, proving the validity of the SE attention module. In Subsection 3.2, we introduce the Cross-BiFPN module for feature fusion in the neck network. The concatenation operation involves the channel stacking of input features from different branches. In scenarios with a large number of branches, assigning different weights to distinct channels becomes useful. Additionally, given the characteristics of tiny-sized objects with subtle features, we incorporate the SE attention module after the H_2 feature map. This module performs adaptive weight assignment for channel dimensions, activating features relevant to classification and localization tasks while suppressing irrelevant information in the input features. Figure 5 shows an overview of the proposed model.

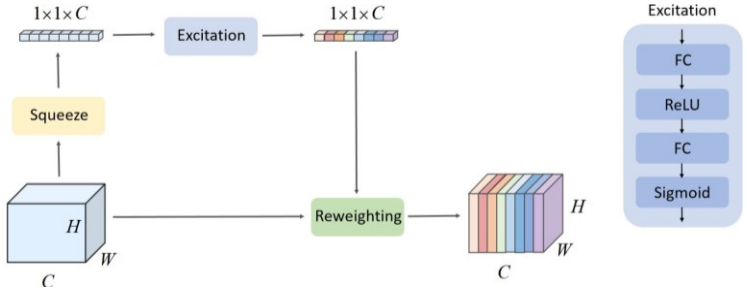


Figure 4. The process of Squeeze and Excitation.

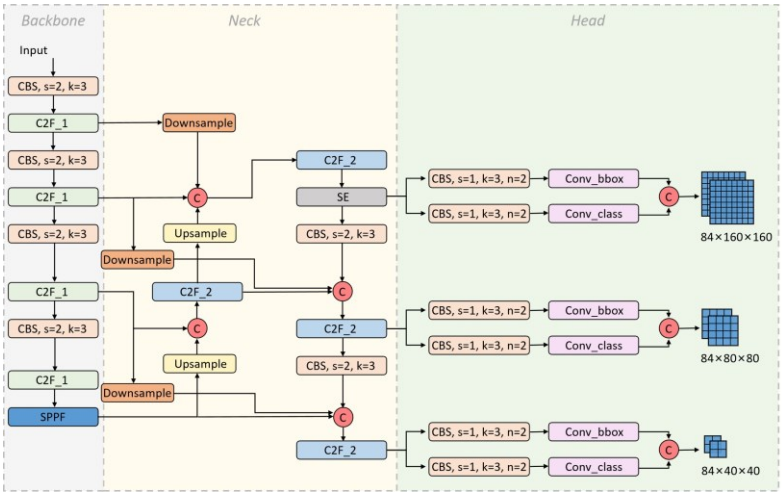


Figure 5. Overview of the proposed YOLO-Ti structure.

3.4 Loss function

YOLO-Ti has the same loss function settings as in YOLOv8. The classification loss L_{cls} assesses the model's performance by computing the cross-entropy of the probability distribution between prediction and ground truth[33]:

$$L_{\text{cls}} = -\frac{1}{M}\sum_{i}\sum_{c=1}^{M} y_{ic}\log\left(p_{ic}\right) \tag{2}$$

where y_{ic} is a sign function, y_{ic}=1 if in fact sample i belongs to category c, otherwise y_{ic}=0. p_{ic} denotes the probability of predicting that sample i belongs to category c, and M represents the number of categories.

The bounding box loss consists of L_{dfl} and L_{ciou}. L_{dfl} is calculated using the difference between the predicted distance from the center point of the bounding box to each edge and the true value:

$$L_{\text{dfl}}\left(S_i, S_{i+1}\right) = -\left(\left(d_{i+1} - d\right)\log\left(S_i\right) + \left(d - d_i\right)\log\left(S_{i+1}\right)\right) \tag{3}$$

where d is the true distance, d_i and d_{i+1} represent two values closest to d, one left and one right, S_i and S_{i+1} denote the predicted probability respectively. This equation helps the network focus more quickly on the distribution of neighborhood of the target in the form of cross-entropy.

In addition, L_{ciou} measures the differences between the predicted bounding box and the ground truth bounding box:

$$L_{\text{ciou}} = 1 - IoU + \frac{\rho^2\left(b, b_{\text{gt}}\right)}{s^2} + \alpha v \tag{4}$$

where IoU quantifies overlap by calculating the ratio of the intersection area to the union area between two bounding boxes. b and b_{gt} denote the center points of the two boxes, ρ is the Euclidean distance between them, s refers to the diagonal distance between the minimum bounding rectangle of the two boxes. α represents a weight function, and v is used to measure the difference between boxes in terms of aspect ratio. The total loss is calculated as follows:

$$L_{\text{total}} = L_{\text{cls}} + L_{\text{dfl}} + L_{\text{ciou}} \tag{5}$$

4. EXPERIMENTS

4.1 Experimental conditions

(1) Datasets

We collected facial images from 5 volunteers using a professional camera for our dataset. On each volunteer's face, we drew tiny, evenly distributed markers, as illustrated in Figure 6. We made sure to capture multiple sets of full-face photos from a unified viewpoint. Each volunteer's face had a varying number of markers with different colors and shapes, which could differ in each shooting session. All data collected in this paper is for non-profit academic use only.

The facial dataset contains 1320 high-resolution face images with a size of 1024×1024 pixels. The labeling of facial data was done by a pre-trained program combined with manual labeling. The annotations include four classes: red marker, black marker, blue marker, and green marker. There are a total of 103,681 instances in the dataset, with an average of about 80 bounding boxes per image. Figure 7 shows the distribution of class labels and the scale of bounding boxes.

In this dataset, 90% of objects have an absolute size of smaller than 10 pixels, with the largest object being less than 16 pixels. Table 1 compares the size distribution of objects in different datasets. The mean and standard deviation of absolute sizes in our dataset of facial markers are 6.7 pixels and 2.1 pixels, significantly smaller than other aerial image and natural image datasets. Compared to the public AI-TOD dataset[34] of tiny objects, the sizes are nearly half.

Table 1. Mean and standard deviation of object size across different datasets.

Dataset[34]	Absolute size (pixel)	Relative size (pixel)
PASCAL VOC 07++12	156.6±111.2	0.372±0.265
MS COCO	99.5±107.5	0.190±0.203
DIOR	65.7±91.8	0.082±0.115
Airbus-Ship	44.9±44.1	0.058±0.057
VisDrone	35.8±32.8	0.030±0.026
DOTA-v1.0	55.3±63.1	0.028±0.034
DOTA-v1.5	34.0±47.8	0.016±0.026
xView	34.9±39.9	0.011±0.013
AI-TOD	12.8±5.9	0.016±0.007
Ours	6.7±2.1	0.007±0.002

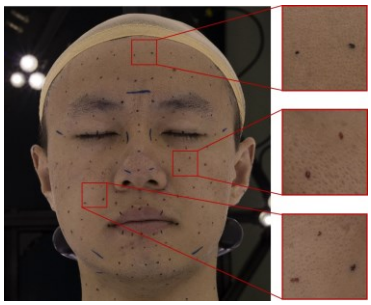
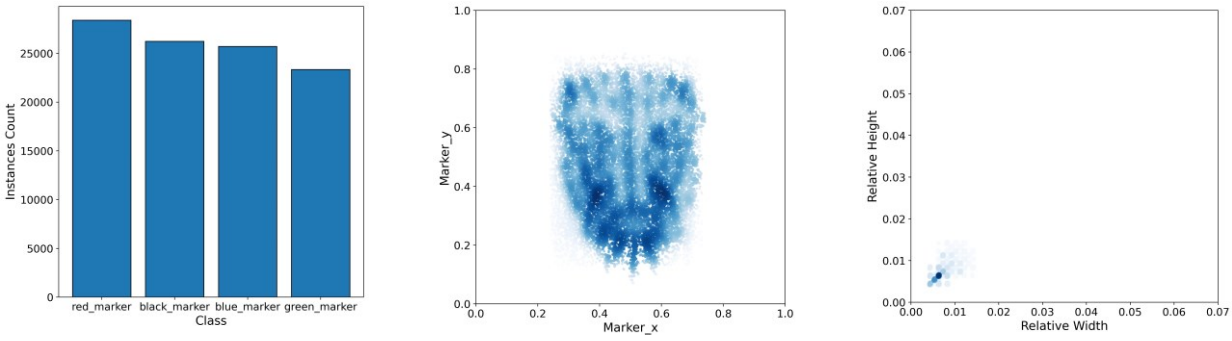


Figure 6. Visual examples of drawn facial markers.



(a) (b) (c)

Figure 7. Analysis of the facial marker dataset. (a): The distribution of class labels; (b): The visualization of object positions; (c): The visualization of bounding box scales.

(2) Implementation details

All experiments were performed with NVIDIA GeForce RTX 3080 graphics card, Intel(R) Core (TM) i7-12700K processor, Windows operating system (version 10), PyTorch (version 1.12.1) and Python (version 3.8.18). Program acceleration was achieved through CUDA 11.3.1 and CUDNN 8.2.1.

The dataset images and labels were split into training, validation, and test sets in a ratio of 8:1:1. During the training of the original YOLOv8 model, we utilized the official pre-trained model to speed up the network training process. The training employed the Adam optimizer with an initial learning rate of e^{-3}, and the learning rate was reduced to e^{-5} for the last epoch. The weights for classification loss L_{cls}, bounding box loss L_{ciou} and L_{dfl} were used with default values of 0.5, 7.5 and 1.5. The maximum number of training epochs on the dataset was set to 100, and the training batch size was 16. The parameters used for training YOLO-Ti were consistent with those used for YOLOv8.

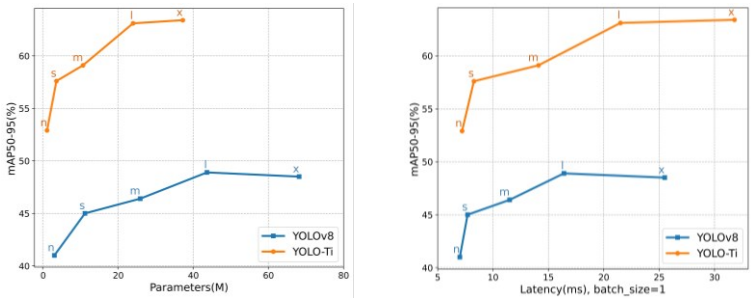
4.2 Experimental results

(1) Improvements based on models of different sizes

This subsection involves the improvement of YOLOv8 models of different sizes, namely YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x, following the methods described in Section 3. The original models and their improved YOLO-Ti models were individually trained on the facial marker dataset.

Figure 8a depicts the performance of models of different sizes (n, s, m, l, x). It can be seen that from size n to size l, the mAP50-95 of the model improves as the number of parameters increases. However, from size l to size x, the mAP50-95 remains relatively constant, even showing a slight decrease. Overall, the YOLO-Ti model, compared to the YOLOv8 model, shows substantial advancements in detection performance, coupled with a reduction in parameters.

Figure 8b shows the relationship between the inference time required for different-sized models and their mAP50-95. The overall trend is similar to Figure 8a. Taking example of the YOLO-Tin model, we can calculate a detection speed of approximately 139 frames per second, using the reciprocal of latency.



(a) (b)

Figure 8. Performance comparison of YOLOv8 and YOLO-Ti models of different sizes on facial marker dataset. (a): Relationship between mAP and parameters; (b): Relationship between mAP and latency.

(2) Ablation study

In this subsection, we conducted ablation experiments to explore the impact of each improved or added module on overall performance, using the original YOLOv8n as the baseline. Experiments were carried out under the same configuration parameters, with input sizes set to 640×640 pixels. After 100 iterations, we recorded several key indicators for each training task. The experimental results are presented in Table 2.

Table 2. Ablation experiment results for YOLOv8n on the facial marker dataset.

Model	Precision (%)	Recall (%)	mAP50 (%)	mAP50-95 (%)	Params (M)	FLOPs (G)
YOLOv8n	93.2	38.8	59.9	41.0	3.01	8.2
YOLOv8n+H_2-H_5	89.4	77.6	87.4	49.8	0.99	10.6
YOLOv8n+H_2+SE	88.2	77.9	88.2	50.2	0.99	10.6
YOLOv8n+H_2+Cross-BiFPN	89.5	79.2	89.4	53.0	1.05	11.2
YOLOv8n+H_2+Cross-BiFPN+SE	90.4	80.4	90.3	52.9	1.05	11.2

According to Table 2, removing the original large object detection head H_5 and adding the H_2 layer not only reduces the model parameters by 66%, but also significantly improves the model's mAP in detection tasks for tiny facial makers, especially in terms of recall rate. In addition, the integration of SE and Cross-BiFPN modules enhances the model's performance, with mAP increasing by 0.8% and 2.0%, respectively. Overall, despite a slight decrease in Precision, there is a substantial 41.6% improvement in Recall, indicating an enhanced ability to detect more tiny objects. This highlights the availability of adopting a high-resolution detection head, incorporating attention and feature fusion modules into the neck of the network. Figure 9 displays some visualization results of YOLOv8n and our YOLO-Tin for facial markers.

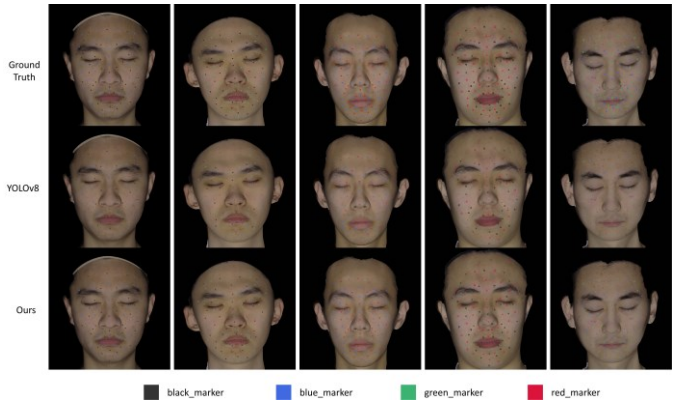


Figure 9. Visual results: YOLOv8n vs. YOLO-Tin on facial marker dataset.

(3) Comparison with other methods

In order to validate the effectiveness of the proposed method, we compared our model on the self-constructed facial marker dataset with previous methods, such as YOLOv4[16], YOLOv5s[20], YOLOv7[35], Faster-RCNN[18], and Centernet[36]. Table 3 reports the experimental results. It can be seen that the YOLO-Tin model performs best in detecting tiny markers, with its highest mAP50 and AP50 values in each category. Compared to YOLOv8n, YOLO-Tin's mAP is improved by 30.4%, demonstrating that it has stronger detection capabilities for tiny objects while using fewer parameters and computational complexity.

It's noteworthy that the Faster-RCNN model shows relatively poor performance while detecting tiny facial markers. This is attributed to the deep network of Faster-RCNN, leading to the loss of fine details in tiny object features and resulting in numerous false negatives. This observation supports the idea that simply increasing network depth and parameters may not be effective for tiny object detection tasks. In terms of the characteristics of tiny objects, our approach appropriately reduces the size of the network, pays more attention to low-level features, and effectively utilizes the features of tiny objects in the images.

Table 3. Comparative performance of YOLO-Ti against six other detection models on the facial marker dataset.

Model	**AP50 (%)**				**mAP50 (%)**	**Params (M)**
	Black marker	**Blue marker**	**Green marker**	**Red marker**		
Anchor-based one-stage:						
YOLOv4	57.4	39.6	42.0	45.0	46.0	64.4
YOLOv5s	63.7	47.7	57.1	69.6	59.5	7.2
YOLOv7	68.0	37.4	48.5	68.5	55.6	36.9
Anchor-based two-stage:						
Faster-RCNN	10.2	10.6	26.6	29.6	19.3	28.3
Anchor-free:						
Centernet	49.9	42.1	54.4	72.6	54.8	32.7
YOLOv8n	63.9	47.1	60.6	68.1	59.9	3.0
YOLO-Tin	95.1	81.8	90.6	93.6	90.3	1.1

5. CONCLUSIONS

In this paper, we propose a detection model YOLO-Ti, based on the YOLOv8 model for the detection task of facial markers, which is suitable for tiny targets. First, a high-resolution detection head is added for detecting tiny-sized targets. Second, an improved feature fusion module, Cross-BiFPN, is presented to incorporate additional cross-skipping connections between different network layers, enhancing the utilization of low-level features. Third, a SE attention module is introduced into the head network to further strengthen the model's ability to detect tiny objects.

In addition, we make modifications to the base model of different scales and evaluate their enhancements. Ablation studies and comparison experiments results show that our model has the highest mAP value and is suitable for the tiny target detection task. However, the detection speed decreases as the model complexity increases. Our future work is to further improve the detection capability of the model without reducing the detection speed, and combine it with face reconstruction, target tracking and other applications.

ACKNOWLEDGEMENTS

This work was supported by the National Key Research and Development Program of China (No.2022YFF0902303) and the Beijing Municipal Science & Technology Commission and Administrative Commission of Zhongguancun Science Park under Grant Z221100007722002.

REFERENCES

[1] Deng, Y., Yang, J., Xu, S., Chen, D., Jia, Y. and Tong, X., "Accurate 3D face reconstruction with weakly-supervised learning: from single image to image set," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 285-295 (2019).
[2] Li, J., Kuang, Z., Zhao, Y., He, M., Bladin, K. and Li, H., "Dynamic facial asset and rig generation from a single scan," ACM Transactions on Graphics 39(6), 1-18 (2020).
[3] Riviere, J., Gotardo, P., Bradley, D., Ghosh, A. and Beeler, T., "Single-shot high-quality facial geometry and skin appearance capture," ACM Transactions on Graphics 39(4), 1-12 (2020).
[4] Feng, Y., Feng, H., Black, M. and Bolkart, T., "Learning an animatable detailed 3D face model from in-the-wild images," ACM Transactions on Graphics 40(4), 1-13 (2021).
[5] Wu, S., Rupprecht, C. and Vedaldi, A., "Unsupervised learning of probably symmetric deformable 3D objects from images in the wild," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 1-10 (2020).
[6] Wu, F., Bao, L., Chen, Y., Ling, Y., Song, Y., Li, S., Ngan, K. and Liu, W., "MVF-Net: multi-view 3D face morphable model regression," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 959-968 (2019).
[7] Shang, J., Shen, T., Li, S., Zhou, L., Zhen, M., Fang, T. and Quan, L., "Self-supervised monocular 3D face reconstruction by occlusion-aware multi-view geometry consistency," 2020 European Conference on Computer Vision (ECCV), 53-70 (2020).
[8] Li, T., Liu, S., Bolkart, T., Liu, J., Li, H. and Zhao, Y., "Topologically consistent multi-view face inference using volumetric sampling," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 3804-3814 (2021).
[9] Wood, E., Baltrušaitis, T., Hewitt, C., Johnson, M., Shen, J., Milosavljevic, N., Wilde, D., Garbin, S., Sharp, T., Stojiljkovic, I., Cashman, T. and Valentin, J., "3D face reconstruction with dense landmarks," 2022 European Conference on Computer Vision (ECCV), 160-177 (2022).
[10] Guo, J., Zhu, X., Yang, Y., Yang, F., Lei, Z. and Li, S., "Towards fast, accurate and stable 3D dense face alignment," 2020 European Conference on Computer Vision (ECCV), 152-168 (2020).
[11] Liu, F., Zhu, R., Zeng, D., Zhao, Q. and Liu, X., "Disentangling features in 3D face shapes for joint face reconstruction and recognition," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5216-5225 (2018).
[12] Egger, B., Smith, W. A. P., Tewari, A., Wuhrer, S., Zollhoefer, M., Beeler, T., Bernard, F. and Bolkart, T., "3D morphable face models-past, present, and future," ACM Transactions on Graphics 39(5), 1-13 (2020).
[13] Ravikumar, S., Davidson, C., Kit, D., Campbell, N., Benedetti, L. and Cosker, D., "Reading between the dots: combining 3D markers and FACS classification for high-quality blendshape facial animation," Proceedings of the 42nd Graphics Interface Conference, 143-151 (2016).
[14] Zhou, W., Cai, C., Zheng, L., Li, C. and Zeng, D., "ASSD-YOLO: a small object detection method based on improved YOLOv7 for airport surface surveillance," Multimedia Tools and Applications 83, 55527-55548 (2024).
[15] Wang, J., "Feature enhancement algorithm for recognition of small targets in low resolution images," 2023 IEEE 6th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), 352-355 (2023).
[16] Bochkovskiy, A., Wang, C. and Liao, H., "YOLOv4: optimal speed and accuracy of object detection," arXiv 2004.10934 (2020).
[17] Haris, M., Shakhnarovich, G. and Ukita, N., "Task-driven super resolution: object detection in low-resolution images," Proceedings of the 28th International Conference on Neural Information Processing, 387-395 (2021).
[18] Ren, S., He, K., Girshick, R. and Sun, J., "Faster R-CNN: towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6), 1137-1149 (2017).

[19] Benjumea, A., Teeti, I., Cuzzolin, F. and Bradley, A., "YOLO-Z: Improving small object detection in YOLOv5 for autonomous vehicles," arXiv 2112.11798 (2021).
[20] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., "You only look once: unified, real-time object detection," 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 779-788 (2016).
[21] Liu, S., Qi, L., Qin, H., Shi, J. and Jia, J., "Path aggregation network for instance segmentation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8759-8768 (2018).
[22] Tan, M., Pang, R. and Le, Q., "EfficientDet: scalable and efficient object detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10781-10790 (2020).
[23] Li, Z., Wang, Z. and He, Y., "Aerial-photography dense small target detection algorithm based on adaptive cooperative attention mechanism," Acta Aeronautica et Astronautica Sinica 44(8), 327944 (2023).
[24] Akyon, F., Altinuc, S. and Temizel, A., "Slicing aided hyper inference and fine-tuning for small object detection," 2022 IEEE International Conference on Image Processing (ICIP), 966-970 (2022).
[25] Zhang, S., Xin, M., Wang, X. and Zhang, M., "Anchor-free network with guided attention for ship detection in aerial imagery," Journal of Applied Remote Sensing 15(2), 024511 (2021).
[26] Li, Z., Liu, X., Zhao, Y., Liu, B., Huang, Z. and Hong, R., "A lightweight multi-scale aggregated model for detecting aerial images captured by UAVs," Journal of Visual Communication and Image Representation 77, 103058 (2021).
[27] Liu, Y., Ma, D. and Wang, Y., "Lightweight multi-target detection algorithm for unmanned aerial vehicle aerial imagery," Journal of Applied Remote Sensing 17(4), 046505 (2023).
[28] Wang, X., He, N., Hong, C., Wang, Q. and Chen, M., "Improved YOLOX-X based UAV aerial photography object detection algorithm," Image and Vision Computing 135, 104697 (2023).
[29] Jocher, G., Chaurasia, A. and Qiu, J., YOLO by ultralytics, <https://github.com/ultralytics/ultralytics>.
[30] Zhou, L., Zhang, S., Qiu, T., Xu, W., Feng, Z. and Song, M., "PatchDetector: Pluggable and non-intrusive patch for small object detection," Neurocomputing 589, 127715 (2024).
[31] Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B. and Belongie, S., "Feature pyramid networks for object detection," 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 936-944 (2017).
[32] Hu, J., Shen, L., Albanie, S., Sun, G. and Wu, E., "Squeeze-and-excitation networks," IEEE Transactions on Pattern Analysis and Machine Intelligence 42(8), 2011-2023 (2020).
[33] Gao, X., Nguyen, M. and Yan, W., "A high-accuracy deformable model for human face mask detection," Image and Video Technology 14403, 96-109 (2024).
[34] Wang, J., Yang, W., Guo, H., Zhang, R. and Xia, G., "Tiny object detection in aerial images," 2020 25th International Conference on Pattern Recognition (ICPR), 3791-3798 (2021).
[35] Wang, C., Bochkovskiy, A. and Liao, H., "YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7464-7475 (2023).
[36] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q. and Tian, Q., "CenterNet: keypoint triplets for object detection," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 6568-6577 (2019).