Publ. Astron. Soc. Japan (2023) 75 (6), 1311–1323
https://doi.org/10.1093/pasj/psad071
Advance Access Publication Date: 2023 October 27

AstroYOLO: A hybrid CNN–Transformer deep-learning object-detection model for blue horizontal-branch stars

Yuchen HE,[1,†] Jingjing WU,[1,†] Wenyu WANG,[1] Bin JIANG,[1,*] and Yanxia ZHANG[2,*]

[1]School of Mechanical, Electrical & Information Engineering, Shandong University, Weihai 264209, Shandong, China
[2]CAS Key Laboratory of Optical Astronomy, National Astronomical Observatories, Beijing 100101, China

[*]E-mail: jiangbin@sdu.edu.cn, zyx@bao.ac.cn

[†]Indicates equal contribution.

Received 2023 June 26; Accepted 2023 September 25

Abstract

Blue horizontal-branch stars (BHBs) are ideal tracers for studying the Milky Way (MW) due to their bright and nearly constant magnitude. However, an incomplete screen of BHBs from a survey would result in bias of estimation of the structure or mass of the MW. With surveys of large sky telescopes like the Sloan Digital Sky Survey (SDSS), it is possible to obtain a complete sample. Thus, detecting BHBs from massive photometric images quickly and effectually is necessary. The current acquisition methods of BHBs are mainly based on manual or semi-automatic modes. Therefore, novel approaches are required to replace manual or traditional machine-learning detection. The mainstream deep-learning-based object-detection methods are often vanilla convolutional neural networks whose ability to extract global features is limited by the receptive field of the convolution operator. Recently, a new Transformer-based method has benefited from the global receptive field advantage brought by the self-attention mechanism, exceeded the vanilla convolution model in many tasks, and achieved excellent results. Therefore, this paper proposes a hybrid convolution and Transformer model called AstroYOLO to take advantage of the convolution in local feature representation and Transformer's easier discovery of long-distance feature dependences. We conduct a comparative experiment on the 4799 SDSS DR16 photometric image dataset. The experimental results show that our model achieves 99.25% *AP*@50, 93.79% *AP*@75, and 64.45% *AP*@95 on the test dataset, outperforming the YOLOv3 and YOLOv4 object-detection models. In addition, we test on larger cutout images based on the same resolution. Our model can reach 99.02% *AP*@50, 92.00% *AP*@75, and 61.96% *AP*@95 respectively, still better than YOLOv3 and YOLOv4. These results also suggest that an appropriate size for cutout images is necessary for the performance and computation of object detection. Compared with the previous models, our model has achieved satisfactory object-detection results and can effectively improve the accuracy of BHB detection.

Key words: methods: data analysis — stars: horizontal-branch — surveys

© The Author(s) 2023. Published by Oxford University Press on behalf of the Astronomical Society of Japan.
All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

1 Introduction

Blue horizontal-branch stars (BHBs) are metal-poor Population-II objects usually located at the horizontal branch of the Hertzsprung–Russell color–magnitude diagram (Montenegro et al. 2019). The core of a BHB star burns helium, while the shell burns hydrogen, and the mass is generally 0.5–1.0 $M_{\odot}$ (Ruhland et al. 2011). Because of their bright and near-constant magnitude properties (Sirko et al. 2004), BHBs are the most prominent tracers to study the history and structure of the Milky Way (MW) (Newberg et al. 2003; Monaco et al. 2003; Niederste-Ostholt et al. 2010; Xue et al. 2011; Santucci et al. 2015; Aguado et al. 2019; Culpan et al. 2021), the age gradient of the Galactic halo (Preston et al. 1991; Whitten et al. 2019), and the closed mass of the MW (Sommer-Larsen & Christensen 1986; Sommer-Larsen et al. 1989; Xue et al. 2008; Deason et al. 2020; Bird et al. 2022). Some studies have even used BHBs to analyze the influence of the Large Magellanic Cloud on the Galactic halo (Petersen & Peñarrubia 2021). In particular, for studying the Galactic halo and Galactic mass, Bird et al. (2022) pointed out that a massive number of object samples will help reduce sampling noise and reveal the kinematic substructure. As many large-scale sky survey projects have produced massive photometric images, it is necessary to develop new object-detection methods based on deep learning to replace manual or traditional machine-learning methods to efficiently detect BHBs in sky frames and build larger tracer sample datasets for future research.

The convolutional neural network (CNN) can discover micro-features that humans cannot notice and perform different tasks accurately and efficiently. It has been gradually applied in astronomy, such as using CNN to classify the morphology of galaxies (Domínguez Sánchez et al. 2018; Cheng et al. 2021) or searching for strong gravitational lensing (Cañameras et al. 2020; Morgan et al. 2022) in photometric images. However, CNN still needs some improvements due to the limitation of the receptive field of the convolutional operator. It is challenging to achieve better results in large-scale images (Yuan et al. 2021). To solve the long-distance attention limitation problem of CNN, the Transformer model (Vaswani et al. 2017) has been gradually introduced into the field of computer vision (CV; Dosovitskiy et al. 2021). Transformer's self-attention mechanism makes it easier to extract information about each area of an image and establish relationships between those areas (Wu et al. 2021). However, the original Transformer model has many parameters and requires a huge amount of computing resources and massive datasets as data support (Yu et al. 2022). For example, the DETR object-detection model (Carion et al. 2020) adopting a vanilla Transformer requires three days of training on 16 V100 GPUs with a dataset containing 118k training images. Therefore, the combination of CNN and Transformer can give full play to their respective advantages, learn local feature information, search for long-distance dependences between these features, and significantly reduce the number of parameters and calculations of Transformer (Chen et al. 2022).

This paper presents an object-detection model called AstroYOLO that can directly locate and detect BHBs in sky images such as the Sloan Digital Sky Survey (SDSS; York et al. 2000) photometric images rather than classifying them. AstroYOLO, combining CNN and Transformer, first employs CNN to extract local features, and then uses Transformer to find the global dependences between local features. Since the traditional YOLO series models (Redmon et al. 2016; Redmon & Farhadi 2017, 2018; Bochkovskiy et al. 2020) adopted a vanilla convolution operator, it could not effectively model long-distance dependences, while the PANet (Liu et al. 2018) in its neck layers directly connected the feature maps in series, resulting in a large amount of feature redundancy. The structure of our model is similar to that of the YOLO (You Only Look Once) series model, which also consists of three parts, namely, a backbone for extracting local semantic features step by step, a neck layer for finding critical features in the multi-scale feature map, and three sets of detection heads for prediction results at different scales. Depending on the specifics of our task and data, unlike the YOLO series models, our model adopts the proposed multi-scale feature fusion module instead of PANet in the neck layers. The improved structure adaptively fuses feature maps of different scales and establishes global dependences from the fused local semantic features.

The organizational structure of this paper is as follows. Section 2 introduces the dataset and data preprocessing. Section 3 illustrates the YOLO object-detection model, Vision Transformer, and the AstroYOLO method proposed in this paper, as well as the evaluation metrics for the performance of object-detection tasks. In section 4, the comparative experimental results of AstroYOLO and previous models are analyzed qualitatively. In addition, the impact of different model structure settings on performance through multiple ablation experiments is also discussed. Finally, the main conclusion is summarized in section 5.

2 Dataset and data preprocessing

2.1 Dataset

By the coordinates in the catalog of BHBs by Xue et al. (2011), we select 4799 BHB objects from SDSS DR16 (Ahumada et al. 2020). The sky distribution is shown in figure 1.

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

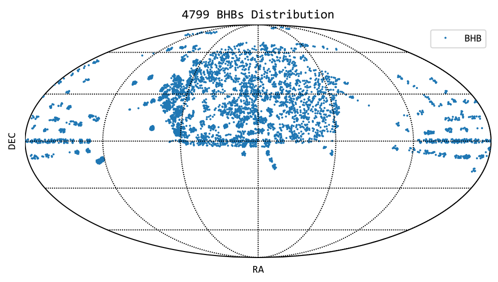

Fig. 1. BHB distribution in the dataset.

All BHB *run*, *rerun*, *camcol*, *field*, and coordinate parameters are obtained from the PhotoObj catalog through the CasJobs Server (Li & Thakar 2008). To construct the object-detection dataset, we adopt the above parameters to locate the sky frames of the three photometric bands, i.e., g, r, and i, where the BHB stars are situated in the eBOSS (Dawson et al. 2016). However, due to the influence of different exposure times of filters in CCD cameras, the images of the three bands have a slight position deviation (He et al. 2021). Therefore, we target objects in the g band and align the r- and i-band data to the g-band data through the Python-based Reproject software package (Robitaille et al. 2013) and then stack the three band images in order of i, r, and g to produce a three-channel image.

The BHB is generally located in the Galactic halo region with low extinction, so the extinction correction value is small. We used the SFD98 extinction map (Schlegel et al. 1998) to calculate the distribution of correction values in the three bands of g, r, and i and the mean values were 0.1316, 0.0910, and 0.0677, respectively. Therefore, the impact of extinction on data characteristics is negligible.

2.2 Data preprocessing

To train the object-detection model, we need to first generate the ground truth's upper-left and lower-right corner coordinates according to the BHB stars' coordinates, which are used to determine the range of the BHB stars. However, the entire sky frame is too large for neural network training, which will lead to excessive consumption of computing resources and easily cause the loss of features of small-sized objects (Zhang et al. 2020; Song et al. 2020; Chen et al. 2021). Therefore, we randomly crop the sky frame. As shown in figure 2, for a sky frame, we first determine the coordinates of the BHB star and then randomly crop a cutout image of size $H \times W$ centered with the coordinates. In this way, each BHB object is cropped N times. To achieve better network performance (Jacobs et al. 2019), we rescale the cutout image to the interval [0,1] and apply a square stretch.

Since the object-detection model is sensitive to the object's relative size, we adopt two sizes of cutout images as the datasets, namely $H = W = 352$ and $H = W = 512$, respectively. In the case of the same object size, increasing the size of the input image will reduce the object's size relative to the whole input image, so it will be more challenging to detect BHBs in a larger image of 512×512 pixels than that of 352×352 pixels. In addition, larger input images will also increase the computational load of the network. In the experiment, we randomly divide the datasets into training sets, validation sets, and test sets according to the ratio of $7:1:2$. In order to obtain enough samples to train the network to reduce the overfitting problem, we set the number of random cutouts in the datasets to 10, that is, $N = 10$. Finally, therefore, there are 33590 three-channel sky-frame images in the training sets, 480 in the validation sets, and 960 in the test sets.

3 Methodology

3.1 YOLO object-detection model

The YOLO series model (Redmon et al. 2016; Redmon & Farhadi 2017, 2018; Bochkovskiy et al. 2020) is a widely used single-stage deep-learning-based object-detection architecture. Unlike the traditional two-stage method (Girshick 2015; Ren et al. 2015) that needs to extract the region of interest (ROI) before detection, the single-stage process only needs to input the image and then performs a single inference to obtain the object position. The architecture adopted in this paper is similar to YOLOv4 (Bochkovskiy et al. 2020).

The input image is first divided into several grids before the target location is predicted. Each grid will have corresponding multiple bounding boxes to capture background objects that the foreground may cover. The model predicts the object center position coordinates and the size of the object to generate each bounding box, as well as the probability that the bounding box belongs to a specific category, i.e., confidence. Since multiple bounding boxes may be predicted as the same object, the non-maximum suppression (NMS) method (Neubeck & Van Gool 2006) is used to preserve the prediction result with the highest confidence.

YOLOv4 has three main components: a CSPDarknet53 (Wang et al. 2020) backbone for extracting multi-scale features, SPP (He et al. 2014) and modified PANet (Liu et al. 2018) for the fusion of multi-scale features, and a prediction head for generating bounding boxes of different sizes. Although YOLOv4 has been used in many computer vision tasks, it is still limited by the receptive field of convolution operators, so it cannot effectively model the long-distance dependences in images. In addition, the modified PANet

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

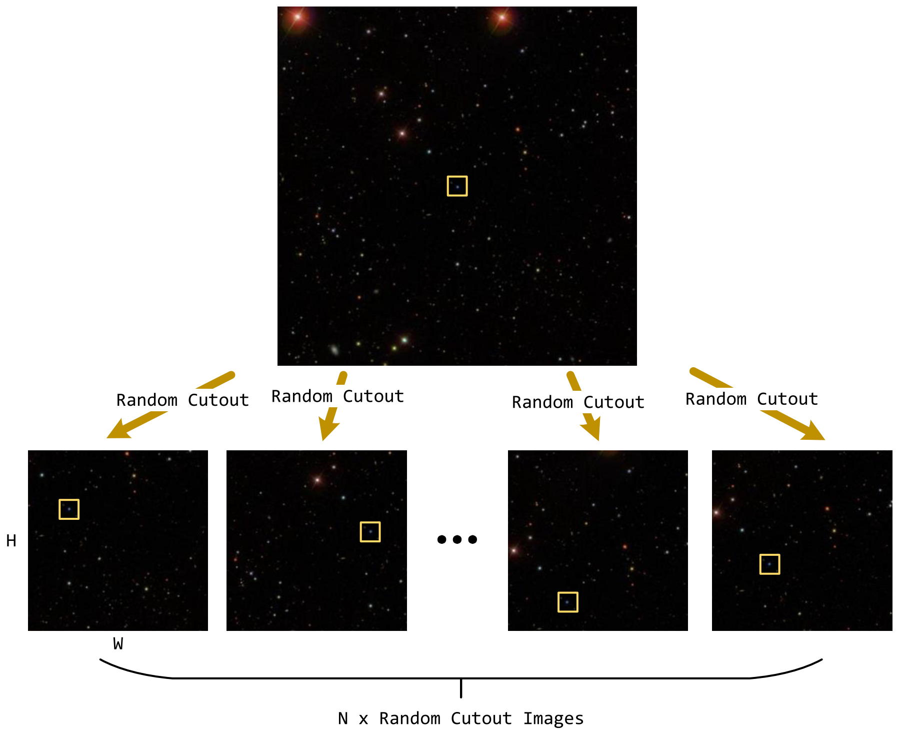

Fig. 2. Image-cropping method. The image is randomly cropped around the BHB star to reduce computation. The yellow bounding box is the BHB star. The size of the bounding box is exaggerated for easy viewing.

only fuses features from two different scales through the concatenate operation, which cannot adaptively select more significant features among these features, resulting in a large amount of feature redundancy.

3.2 Vision Transformer

The Transformer model based on self-attention (Vaswani et al. 2017) has demonstrated its powerful performance in natural language processing (NLP). Compared with the recurrent neural network (RNN) (Mikolov et al. 2010) and long short-term memory network (LSTM) (Hochreiter & Schmidhuber 1997), the long-distance dependence modeling brought by self-attention can more effectively establish the connection of different components in a sentence, so it has the advantage of learning global features. However, the convolution operator, widely used in computer vision, can only construct local feature connections due to its limitations in the receptive field. Therefore, Dosovitskiy et al. (2021) proposed Vision Transformer (ViT), which introduced the Transformer model into computer vision and took advantage of its global attention mechanism. Because ViT is a vanilla Transformer-based architecture, it is necessary to follow the principles of sentence segmentation in NLP when inputting images, dividing the image into multiple patches, and then embedding the input into the network. Our model only utilizes the Transformer block structure, so we do not provide a detailed description of the ViT architecture.

We focus on the Transformer block structure in ViT. As shown in figure 3, in this structure, the input is a token $X \in R^{N\times d}$, where N is $H\times W$, and d is the embedding dimension. X passes through the linear projection to generate $Q \in R^{N\times d}$, $K \in R^{N\times d}$, and $V \in R^{N\times d}$, and is then input to the multi-head self-attention (MHSA) module. For easier understanding, we assume that the number of heads in the MHSA is 1. At this point, the formula for self-attention is as follows:

$$SA(Q, K, V, d) = Softmax\left(\frac{Q\cdot K^T}{\sqrt{d}}\right)\cdot V \tag{1}$$

where d is the scaling factor, consistent with the embedding dimension of K. Equation (1) calculates the inherent attention relationships in the input token X and amplifies this relationship. Since the complete token participates in the computation, the receptive field size is the same as the token, and long-distance dependences can be effectively established.

Subsequently, the weighted token $X_w \in R^{N\times d}$ from MHSA is passed through a feed-forward network (FFN) to complete the computation of this block. The FFN calculation formula is as follows:

$$FFN(X_w) = \xi\{[LN(X_w)]W_1^T + \eta_1\}W_2^T + \eta_2 + LN(X_w) \tag{2}$$

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

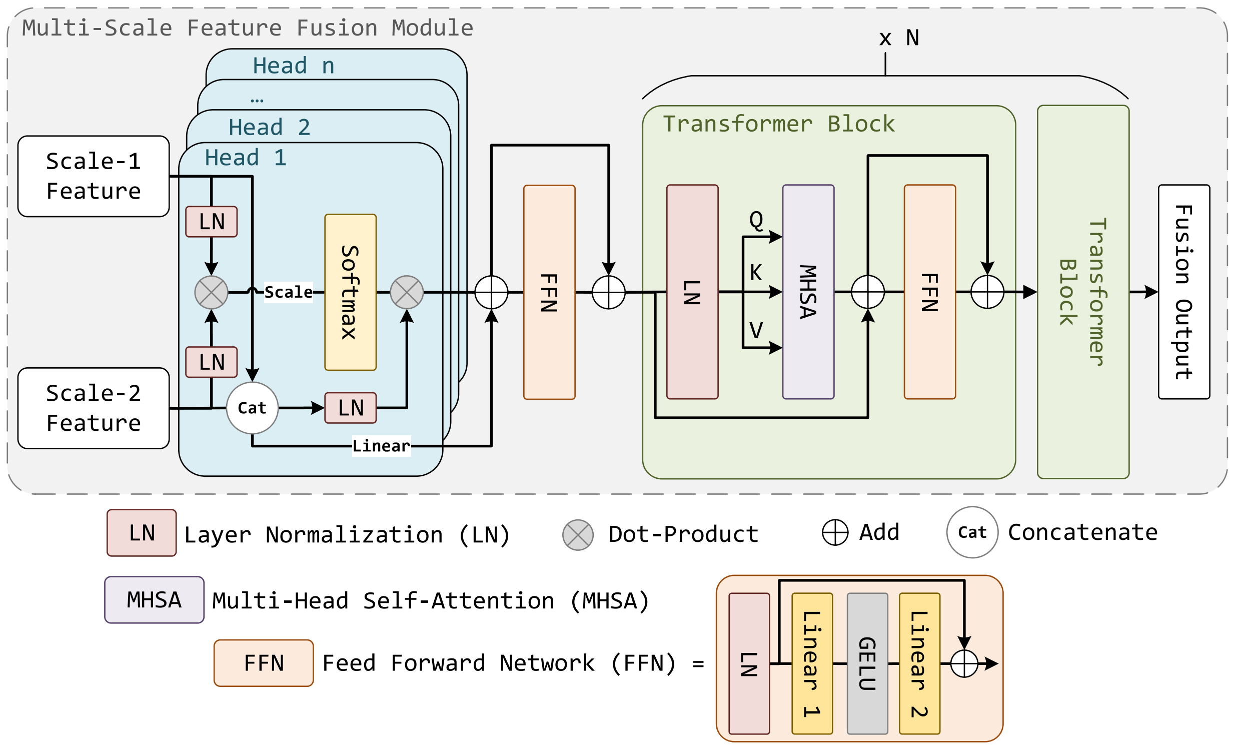

Fig. 3. Multi-scale feature fusion module. This module contains a feature aggregator for fusing two scale feature tokens and N Transformer blocks for adaptively learning the global association between different feature tokens.

where $W_1 \in R^{d\times d}$ and $W_2 \in R^{d\times d}$ are the weights of the linear layers, η_1 and η_2 are the bias of the linear layers, LN represents layer normalization, and ξ represents the GELU (Hendrycks & Gimpel 2016) activation function. The hidden channel of the last linear layer is four times that of the previous linear layer to achieve the effect of the bottleneck.

3.3 AstroYOLO

To overcome the limitation of the YOLO model in which the receptive field is solely determined by convolution and cannot screen multi-scale feature maps, we put forward a new YOLO model called AstroYOLO. As illustrated in figure 4, the AstroYOLO structure comprises a backbone, neck, and detection head. The Transformer model places greater emphasis on global features. For small-scale BHB objects situated in large-scale sky frames, we first extract local features through the convolution operator of the backbone. Its structure is CSPDarknet53, which takes in three-channel input data and has four downsampling operations to extract features of various scales. In the neck part, the main tensor flow begins with the SPP block extracting multi-scale features, passes through two upsampling multi-scale feature fusion modules (MSFFMs), and then passes through two downsampling MSFFMs in order to aggregate multi-scale features and use the self-attention mechanism in the Transformer block to adaptively learn the global associations within local features. The detection head part contains three output channels, each generating prediction results for three types of objects of different scales.

As shown in figure 3, MSFFM contains dual-scale feature fusion and N Transformer blocks for extracting global features, where $N = 3$ in the comparative experiment. For the feature fusion component, we first perform a linear projection operation on the feature tokens $X_1 \in R^{k\times d}$ and $X_2 \in R^{k\times d}$, respectively. Similar to the computation of the self-attention, we adopt the embedding dimension $d = d_{X_1} = d_{X_2}$ of X_1 and X_2 as the scaling factor and then compute the multi-scale feature similarity weight matrix $W_{\mathrm{sim}} \in R^{k\times d}$ between X_1 and X_2 by the following formula:

$$W_{\mathrm{sim}} = Softmax\left(\frac{X_1 \cdot X_2^T}{\sqrt{d}}\right) \tag{3}$$

where $k \in R^{\frac{H}{n}\times\frac{W}{n},n=\{8,16,32\}}$ is the token size; we then adopt the following method to get the multi-scale feature stacking matrix $X_S \in R^{k\times d}$:

$$X_S = (X_1 \oplus X_2)\theta_S^T + \eta_S \tag{4}$$

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

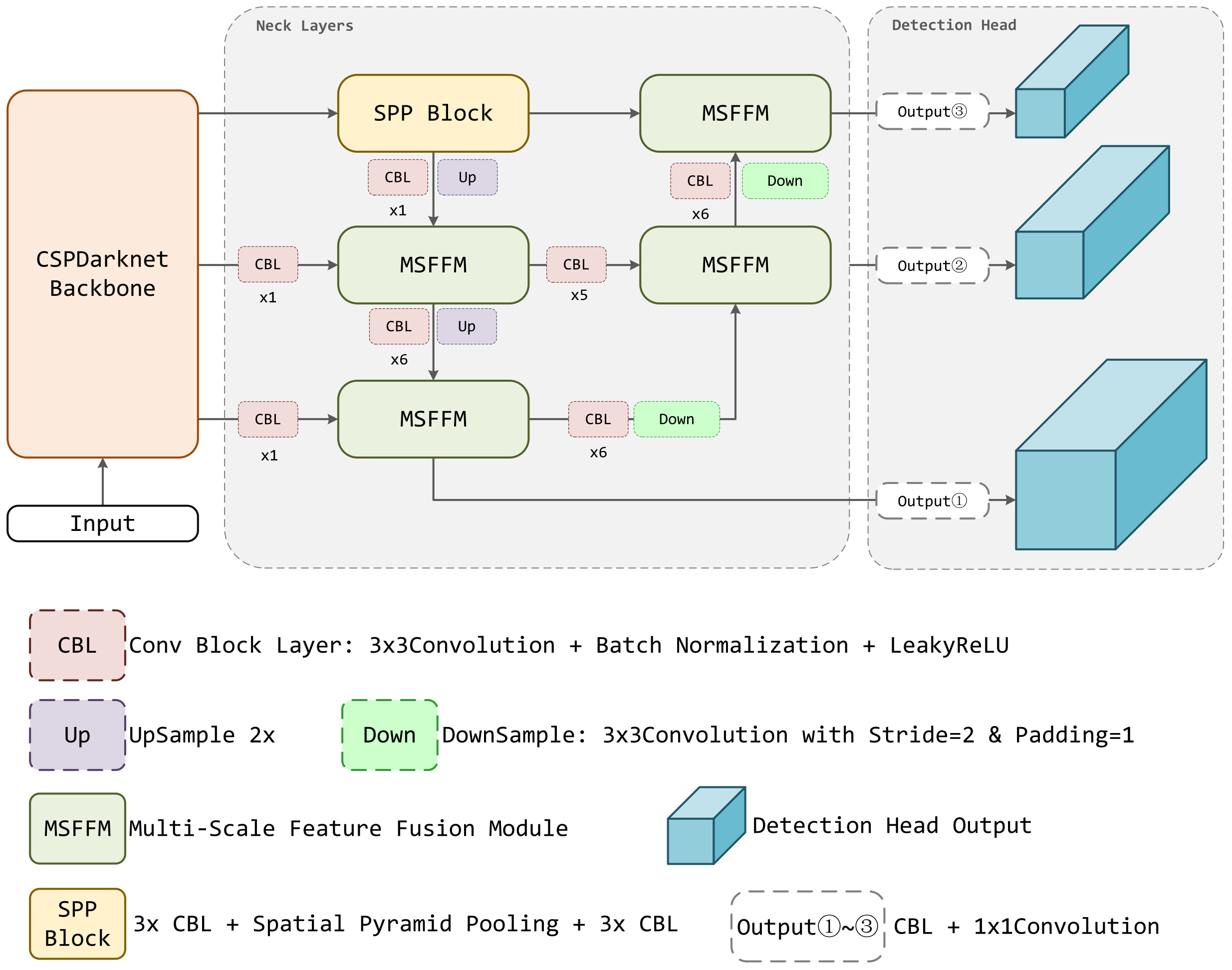
Neck Layers
Detection Head
CSPDarknet Backbone
Input
SPP Block
MSFFM
CBL
Up
Down
x1
x5
x6
Output③
Output②
Output①
CBL Conv Block Layer: 3x3Convolution + Batch Normalization + LeakyReLU
Up UpSample 2x
Down DownSample: 3x3Convolution with Stride=2 & Padding=1
MSFFM Multi-Scale Feature Fusion Module
Detection Head Output
SPP Block 3x CBL + Spatial Pyramid Pooling + 3x CBL
Output①~③ CBL + 1x1Convolution

Fig. 4. AstroYOLO model structure. The model consists of three parts: a CSPDarknet53 backbone for extracting multi-scale local features, neck layers for aggregating multi-scale local features and finding long-distance dependences between different features, and a detection head for generating multi-scale prediction results.

where $\oplus$ represents the concatenate operation, $\theta_S \in R^{d\times 2d}$ is the weight matrix of the linear operation, and $\eta_S \in R^{k\times d}$ is the corresponding bias. Finally, $W_{\rm sim}$ and X_S will be multiplied to obtain weighted multi-scale features $X_W \in R^{k\times d}$:

$$X_W = W_{\rm sim} \cdot X_S. \quad (5)$$

The feature similarity weights from the two scales are obtained through the above method, and the weighted feature token is obtained so that duplicate features can be effectively eliminated and redundancy can be reduced. Through adaptive selection, the feature tokens are passed through N Transformer blocks to find dependences between them, that is, global feature learning, and fetch the global representation of the feature token to combine local features with global long-distance connections. Subsequently, the neck section outputs three feature maps, corresponding to three prediction heads of different scales, to generate bounding boxes of objects—these bounding boxes output prediction results after passing through NMS.

3.4 Metrics for evaluation

We use the most common average precision (AP) in the object-detection task as an evaluation metric to quantitatively evaluate the bounding box size, position, and object category predicted by the model. Before calculating AP, we calculate each object's intersection-over-union (IoU) of the prediction and ground truth bounding box. The calculation is as follows:

$$IoU = \frac{area\ of\ overlap}{area\ of\ union}. \quad (6)$$

By setting the IoU threshold to filter objects with the correct position and size and then setting the category prediction probability threshold to divide the prediction results

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

of these objects into positive and negative, *Precision* and *Recall* are calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN}, \tag{8}$$

where, TP (true positive) refers to when the model correctly classifies positive samples as positive; FP (false positive) refers to when the model incorrectly classifies negative samples as positive; and FN (false negative) refers to when the model incorrectly classifies positive samples as negative.

Subsequently, we calculate the 11-point interpolation AP when the IoU threshold is k following the method of the PASCAL VOC challenge (Everingham et al. 2010):

$$AP(k) = \frac{1}{11}\sum_{r} \max_{\hat{r}:\hat{r}\geq r} Precision(\hat{r}) \tag{9}$$

where $r \in \{0, 0.1, 0.2 ... 1.0\}$ is *Recall* under the possible thresholds of different categories. Then, we calculate AP under different IoU thresholds to measure the model's accuracy following the MS COCO challenge (Lin et al. 2014) when the bounding box size and position limitations are stricter:

$$AP@[K, 0.95] = \frac{1}{20(1-K)}\sum_{k}^{0.95} AP(k) \tag{10}$$

where K is the initial IoU threshold and k is the IoU threshold starting from K and increasing every 0.05 up to 0.95. In the following experiment section, we use three starting IoU thresholds, $AP@[0.5, 0.95]$ ($AP@50$), $AP@[0.75, 0.95]$ ($AP@75$), and $AP@[0.95, 0.95]$ ($AP@95$), where $K = 0.95$ means the most rigorous evaluation of the model's prediction accuracy for the bounding box and category.

4 Experiments and results

Following the settings in subsection 2.2, we divide the cutout images into the training sets, validation sets, and test sets according to a ratio of 7 : 1 : 2, and the experiment employs five-fold cross-validation. Both YOLOv3 and YOLOv4 adopt the official code and use the best hyperparameter configuration that it provides. The AstroYOLO model is implemented using the Python-based PyTorch framework (Paszke et al. 2019).

For the $H = W = 352$ images, the batch size is set to 16, and the number of Transformer blocks is 3, that is, $N = 3$; for the $H = W = 512$ images, due to the limitation of GPU memory, the batch size is set to 8 and the number of Transformer blocks is set to 1. Under the input of two sizes, the number of heads in the MHSA (multi-head self-attention network) is 4. The optimizer uses stochastic gradient descent (SGD) with a momentum (Sutskever et al. 2013), which is 0.9, and we adopt the cosine annealing learning rate descent algorithm to reduce overfitting. The initial learning rate is set to $1e-3$, and the minimum learning rate is $1e-8$. The training stage consists of 120 epochs, of which the warmup phase is 50 epochs, and the weight configuration that performs best on the verification set is used in the final test. All experiments are run on an RTX3090. For more details about the hardware and software experiment environment, please refer to subsection 4.3.

Table 1. Comparative experimental results.[*]

Model	Cutout size (pixel)	$AP@50$[†]	$AP@75$[†]	$AP@95$[†]
YOLOv3	352 × 352	95.83%	91.36%	62.59%
YOLOv4	352 × 352	98.82%	91.92%	63.13%
AstroYOLO	352 × 352	99.25%	93.79%	64.45%
YOLOv3	512 × 512	93.98%	89.20%	60.96%
YOLOv4	512 × 512	97.95%	85.52%	56.26%
AstroYOLO	512 × 512	99.02%	92.00%	61.96%

[*]For 352 × 352 and 512 × 512 images, AstroYOLO's $AP@50$, $AP@75$, and $AP@95$ are better than YOLOv3 and YOLOv4.
[†]Higher is better.

4.1 Comparative experiment

As shown in table 1, AstroYOLO has higher $AP@50$, $AP@75$, and $AP@95$ metrics than YOLOv3 and YOLOv4 at both image sizes. For the 352 × 352 pixels images, AstroYOLO's $AP@50$ improved by 0.44% (from 98.98% to 99.25%) compared to the second-highest model, $AP@75$ increased by 2.03% (from 91.92% to 93.79%), and $AP@95$ improved by 2.09% (from 63.13% to 64.45%). Therefore, it can be seen that, as the IoU threshold continues to increase, AstroYOLO's ability to predict the location of the bounding box at this kind of image size will become stronger and stronger. This demonstrates that the MSFFM successfully fuses the feature information of different scales and discovers the global dependences in many local feature tokens.

Furthermore, to verify the impact of cutout image size on the model at the same resolution, we use cutout images of 512 × 512 pixels. However, due to the memory limitation of GPU, we have to set the number of Transformer blocks to 1, affecting the model's performance. We discuss the details of this in sub-subsection 4.2.1. Although

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

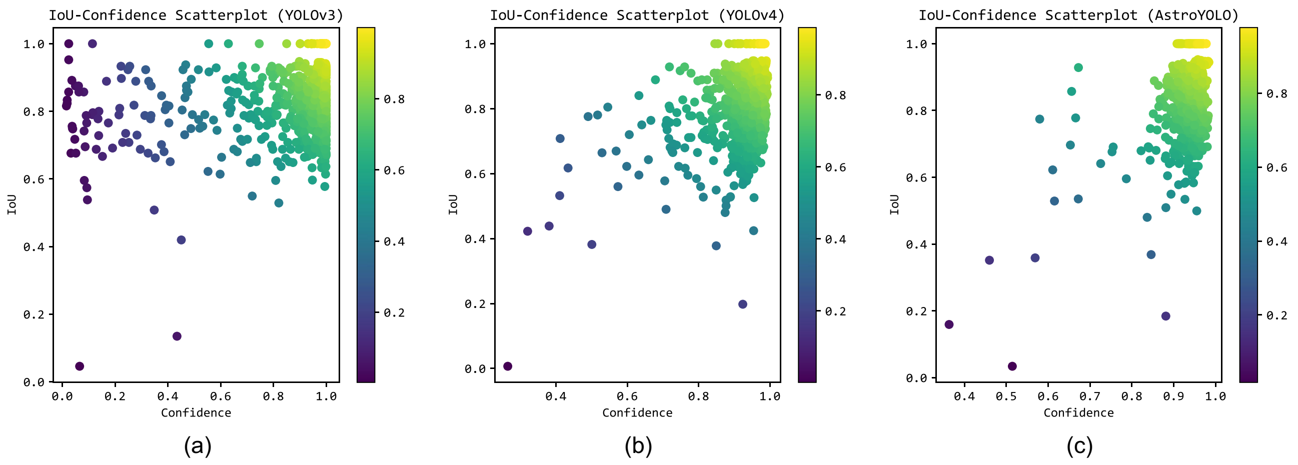

Fig. 5. IoU and confidence density plots of predictions (a) for YOLOv3, (b) for YOLOv4, and (c) for AstroYOLO. IoU shows the coincidence of the predicted bounding box and ground truth, that is, whether the predicted position is accurate; the confidence means the predicted category label probability. IoU and confidence are simultaneously maintained at a high value, located in the yellow dotted area in the upper-right area, meaning that the result is more realistic and not easily filtered out by the threshold limit. It can be seen from the figure that AstroYOLO's prediction results are the most concentrated and gathered in the upper-right area; YOLOv4 has some dispersed results, and YOLOv3 has the worst prediction results.

the optimal model structure cannot be employed, it can be found from the results in table 1 that AstroYOLO can still maintain the state-of-the-art (SOTA) effect on the three metrics, which shows that the proposed MSFFM can still play its role in fusing information of different scales in a larger cutout image size.

The horizontal axis in figure 5 represents the confidence that the object is a BHB star, that is, the prediction accuracy of the category label, and the vertical axis represents the IoU between the predicted bounding box and the ground truth, that is, the accuracy of the object position. Each point represents a prediction result, and the more the prediction results are concentrated in the upper right corner, the more accurate the object-detection effect of the model is. It can be seen from figure 5a that YOLOv3 has the worst effect, a dispersed distribution of prediction results, and a large number of low-confidence predictions. If the confidence threshold increases, many potential objects will be lost. The prediction results of YOLOv4 (figure 5b) are relatively concentrated, but there are still some predictions with low confidence and significant errors in the prediction position. AstroYOLO is the model with the best performance among the three models. As shown in figure 5c, only a relatively small number of points are dispersed, and most predictions are concentrated in the high-confidence and IoU intervals. Therefore, when the threshold increases in pursuit of a more accurate label and position prediction, AstroYOLO can retain BHB candidates to the greatest extent.

For a more intuitive analysis, we show the prediction result images of the model, as shown in figure 6. Even though the dataset has been modified to 352 × 352 pixels, since a BHB star of about 20 × 20 pixels is still tiny and inconvenient to display in the cutout image, we crop the BHB star from the cutout image according to the method in figure 6a. Figure 6b shows the prediction results of AstroYOLO. The predicted bounding box in the figure is blue, the ground truth adopts a green border, and the overlapping part of the two is cyan-blue. The first and second rows are the results of $IoU = 1.0$, from which it can be seen that the border of the box is pure cyan-blue because there is very little interference around a BHB star in data of this type so that it can produce perfect predictions. The third row is the result of $IoU > 0.9$. It can be seen from the figure that the bounding box and the ground truth do not coincide completely, but the overall prediction is still very accurate. This slight performance loss may be due to interference from other celestial bodies around a BHB star, such as the first to third columns of the third row.

In addition, we analyzed the impact of different bounding box position accuracy requirements on model performance, as shown in figure 7. We set the IoU threshold to multiple values from 0.50 to 0.95 in steps of 0.05, and a larger IoU threshold indicates a stricter requirement on the accuracy of the predicted position. It can be seen from the figure that, as the IoU threshold increases, the areas under the PR curves of the three models gradually decrease. At the same IoU threshold, the area under the PR curve of AstroYOLO is larger than that of YOLOv3 and YOLOv4. In particular, the AstroYOLO model can still maintain a high recall at a strict IoU threshold. Considering that the recall is significantly reduced after increasing the position accuracy requirements, setting the IoU threshold to 0.50 in practical applications can avoid missing the object as much as possible.

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

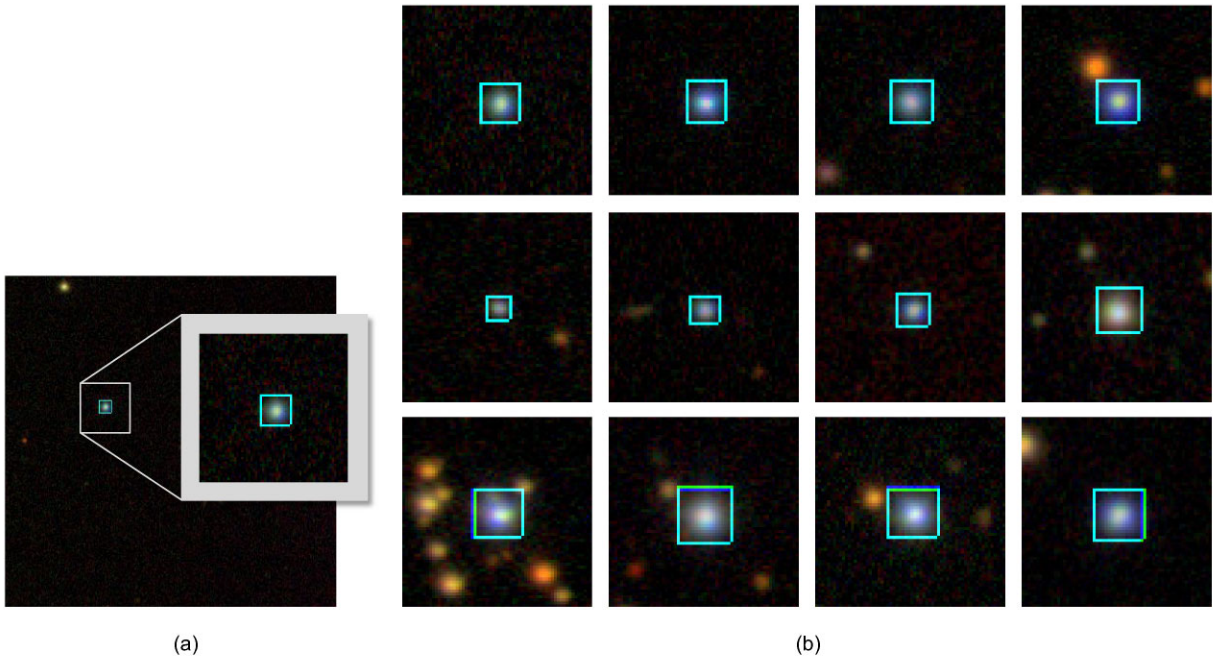

Fig. 6. AstroYOLO's detection result display. (a) In order to more clearly show the overlap of the predicted bounding box and the ground truth, we crop the area around the BHB star. (b) 12 prediction results from AstroYOLO are displayed. The predicted bounding box uses a blue border, and the ground truth uses a green border, so the overlapping part is cyan-blue. The first and second lines are both $IoU = 1.0$ predictions, so the two borders completely overlap; the third line shows $IoU > 0.9$ predictions, so it is seen that most of the borders overlap and become cyan; only a small part does not overlap.

4.2 Ablation experiment

To further explore the influence of model structure on performance, we implement three sets of ablation experiments. The first set of experiments is used to verify the impact of the number of Transformer blocks in MSFFM on global feature extraction; the second set of experiments compares the extraction performance of local features of the CSP-Darknet53 with traditional ResNet-152 (He et al. 2016) and a new SOTA model called ConvNeXt (Liu et al. 2022) as a backbone; the last set of experiments is used to test the performance impact of the backbone parameter amount.

4.2.1 Transformer block number

Since the Transformer block extracts the global features in the fused tokens, its number N will affect the model's performance. In order to verify the above impact, we compare the model performance of $N = 1, 2, 3, 4$ in the 352 $\times$ 352 cutout image dataset. The experimental results are shown in table 2. Since the *AP*@50 index is relatively simple, there is no significant difference between the four N settings. However, for *AP*@75 and *AP*@95, as $N = 3$, the increased parameter amount can better help the model learn global features, and the results are significantly improved compared to $N = 1, 2$.

Nevertheless, when $N = 4$, the model cannot learn more information by increasing the number of parameters due to the limitation of feature number, and it is easy to overfit, so the results of the three metrics are all lower than those with $N = 3$. Therefore, we choose $N = 3$ as the final model structure under our computing resources. For the 512 $\times$ 512 cutout images, the larger image size increases the computational load of the model, so we adopt the structure of $N = 1$; it can be seen from the results in table 1 that AstroYOLO can still maintain the optimal performance.

4.2.2 Backbone selection

To verify the rationality of the backbone using CSP-Darknet53, we select ResNet-152, a representative vanilla convolutional network, and ConvNeXt, which combines large kernel convolution and achieves SOTA on the ImageNet dataset, as a comparison. Table 3 shows that CSP-Darknet53 achieves the best performance in the three metrics, followed by ConvNeXt, and ResNet-152 performs the worst. The reason for the poor performance of ResNet-152 may be that a BHB star's image size is relatively small, the features are limited, and the deep network cannot extract more useful feature information from it. CSPDarknet53 has fewer layers and a parallel feature extraction structure, which can extract more local features within a limited

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

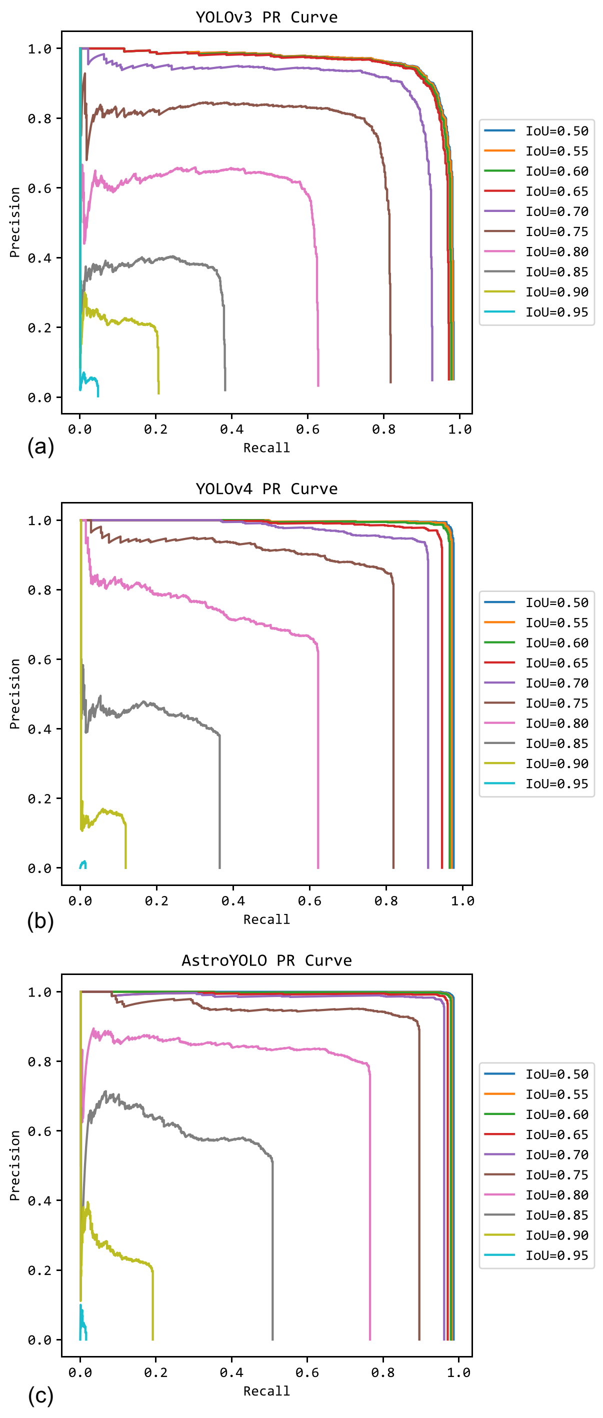

Fig. 7. PR curves of the three models at different IoU thresholds (a) for YOLOv3, (b) for YOLOv4, (c) for AstroYOLO. The IoU threshold denotes the requirement for the accuracy of the bounding box prediction position, and the higher the value, the stricter the accuracy requirement. As shown, the areas under the PR curves of the three models decrease as the IoU threshold increases. At the same IoU threshold, the area under the PR curve of AstroYOLO is larger than that of YOLOv3 and YOLOv4. In particular, at a strict IoU threshold, AstroYOLO can still maintain a high recall.

Table 2. Impact of different Transformer block numbers on model performance.*

Transformer block (N)	$AP@50$†	$AP@75$†	$AP@95$†
1	99.36%	91.86%	61.53%
2	98.75%	91.73%	61.30%
3	99.24%	93.79%	64.45%
4	98.97%	92.51%	63.05%

*For $AP@50$, there is little difference under the four N settings. However, for $AP@75$ and $AP@95$, the model's performance is significantly higher than that of $N = 1, 2$ when $N = 3$. Furthermore, when N increases to 4, all three metrics decline.
†Higher is better.

Table 3. Experimental results of different backbones.

Backbone	$AP@50$†	$AP@75$†	$AP@95$†
ResNet-152	89.06%	64.56%	40.66%
ConvNeXt	99.06%	84.43%	61.53%
CSPDarknet53‡	99.25%	87.99%	64.42%

*ResNet-152 and ConvNeXt are serial vanilla convolutional networks, where ConvNeXt adopts a large kernel convolution with *kernel size* = 7. Furthermore, CSPDarknet53 is a parallel feature encoder.
†Higher is better.
‡Due to a large amount of the parameters in the ConvNeXt, our hardware cannot satisfy the training requirement when its *batch size* = 16, so all models in this experiment are trained with *batch size* = 8, which makes the results not consistent with the previous results.

number of parameters. In addition, CSPDarknet53 adopts the Mish activation function (Misra 2020), which can make the gradient smoother and improve the accuracy of feature extraction compared with the ReLU activation function in ResNet-152.

ConvNeXt adopts large kernel convolutions, a convolution operator with *kernel size* = 7, to expand the convolution receptive field to learn long-distance dependences better. However, because the image size of a BHB star is limited, the backbone needs to extract its local features instead of global features, so the large kernel convolution operator of ConvNeXt cannot effectively play a role, and a large convolution kernel is not conducive to the learning of small-scale local features. Hence, its performance is slightly inferior to CSPDarknet53.

4.2.3 Backbone feature channel

CSPDarknet53's backbone has five feature extraction stages, each of which adopts a 3×3 convolution with *stride* = 2 to halve the resolution of the feature map and double the number of feature channels. To reduce the amount of computation, we compare the impact of [32, 64, 128, 256, 512] channels and [64, 128, 256, 512, 1024] channels on model performance. It can be seen from table 4 that the model's performance drops significantly with [32, 64, 128, 256, 512] channels. This result shows that the backbone,

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

Table 4. Experimental results of the number of two backbone feature channels.*

Backbone feature channel	*AP@50*†	*AP@75*†	*AP@95*†
[32,64,128,256,512]	98.91%	90.51%	59.74%
[64,128,256,512,1024]	99.24%	93.79%	64.45%

*The five channels correspond to the backbone's five feature extraction stages, each containing a downsampling convolution to halve the feature map resolution and double the number of channels.
†Higher is better.

Table 5. Details of hardware and software experiment environment.

Experimental environment	
CPU	Intel i9-12900k
RAM	128GB
GPU	NVIDIA RTX3090
System	Ubuntu 22.04
Python ver.	Python 3.8
CUDA	CUDA 11.7
Framework	PyTorch 1.13
Automatic mixed precision	Enable

as the essential local feature encoder of the model, needs sufficient parameters to meet its feature learning needs. Therefore, at least a five-stage backbone with channels of 64, 128, 256, 512, and 1024 is required to extract local features.

4.3 Hardware and software experiment environment

The detailed hardware and software experimental environment is shown in table 5, and all experiments are completed using the configuration in the table. The table shows that our model can achieve the training and inference process in a consumer-grade hardware environment without needing multi-GPU enterprise-grade server hardware. On our dataset and a single RTX3090, our model only takes approximately two days to complete the training, and the hardware requirements are much less than the vanilla Transformer model. Regarding software, we have enabled PyTorch's automatic mixed precision feature to reduce video memory usage and speed up training.

5 Conclusion

In this paper, we propose a new object-detection model called AstroYOLO. In the integrated model, we introduce MSFFM, which reduces the redundancy of feature tokens by calculating similarity weights for local feature tokens of different scales from the backbone and then filters out key tokens. Moreover, the Transformer block can utilize its global attention to discover the feature correlation between these tokens and maximize the combination of local and global features. Furthermore, compared with the vanilla Transformer model, our model has fewer parameters and requires fewer computing resources, which is more suitable for training and deployment on a small dataset.

To evaluate the performance of our proposed model, we test our model with two datasets with different cutout image sizes and conduct comparative experiments. The 352 × 352 dataset is the benchmark to evaluate our model's capability. Its experimental result shows that AstroYOLO is better than the YOLOv3 and YOLOv4 object-detection models. The performance of these models on the 512 × 512 dataset is acceptable but inferior to that on the 352 × 352 dataset. Appropriate cutout image size contributes to the performance of a model. In addition, we perform three sets of ablation experiments to verify the impact of different architecture designs on the performance of AstroYOLO. It can be concluded from the results that CSPDarknet53 with a feature channel configuration of [64, 128, 256, 512, 1024] is used as the backbone, and the model structure with the number of Transformer blocks as three situations can achieve the best detection performance. With enough representative BHB star training samples, the created AstroYOLO model helps to target BHB star candidates from the imaging surveys (e.g., SDSS, DESI, LSST), which can reduce the uncertainty in the measurement of the MW. Moreover, AstroYOLO may also be used to detect other kinds of objects of interest through images.

Data availability

The code has been open-sourced at ⟨https://github.com/dzxrly/AstroYOLO⟩.

Acknowledgments

The study was funded by the National Natural Science Foundation of China under grant Nos. 12273076, 12133001, and 11873066, the China Manned Space Project with science research grant Nos. CMS-CSST-2021-A04 and CMS-CSST-2021-A06, and the Natural Science Foundation of Hebei Province under grant No. A2018106014. We acknowledge SDSS databases. Funding for the Sloan Digital Sky Survey IV has been provided by the Alfred P. Sloan Foundation, the US Department of Energy Office of Science, and the Participating Institutions. SDSS-IV acknowledges support and resources from the Center for High-Performance Computing at the University of Utah. The SDSS web site is www.sdss.org. SDSS-IV is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS Collaboration including the Brazilian Participation Group, the Carnegie Institution for Science, Carnegie Mellon University, the Chilean Participation Group, the French Participation Group, Harvard-Smithsonian Center for Astrophysics,

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

Instituto de Astrofísica de Canarias, The Johns Hopkins University, Kavli Institute for the Physics and Mathematics of the Universe (IPMU)/University of Tokyo, Lawrence Berkeley National Laboratory, Leibniz Institut für Astrophysik Potsdam (AIP), Max-Planck-Institut für Astronomie (MPIA Heidelberg), Max-Planck-Institut für Astrophysik (MPA Garching), Max-Planck-Institut für Extraterrestrische Physik (MPE), National Astronomical Observatories of China, New Mexico State University, New York University, University of Notre Dame, Observatário Nacional/MCTI, The Ohio State University, Pennsylvania State University, Shanghai Astronomical Observatory, United Kingdom Participation Group, Universidad Nacional Autónoma de México, University of Arizona, University of Colorado Boulder, University of Oxford, University of Portsmouth, University of Utah, University of Virginia, University of Washington, University of Wisconsin, Vanderbilt University, and Yale University.

References

Aguado, D. S., et al. 2019, MNRAS, 490, 2241
Ahumada, R., et al. 2020, ApJS, 249, 3
Bird, S. A., et al. 2022, MNRAS, 516, 731
Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. 2020, arXiv:2004.10934
Cañameras, R., et al. 2020, A&A, 644, A163
Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. 2020, in Computer Vision – ECCV 2020 (Part I), ed. A. Vedaldi, et al. (Cham: Springer), 213
Chen, Y., Dai, X., Chen, D., Liu, M., Dong, X., Yuan, L., & Liu, Z. 2022, in Proc. 2022 IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR) (Los Alamitos: IEEE Computer Society), 5260
Chen, Y., Li, G., Jin, C., Liu, S., & Li, T. 2021, Proc. 35th AAAI Conf. Artificial Intelligence (Palo Alto: AAAI Press), 1105
Cheng, T.-Y., et al. 2021, MNRAS, 507, 4425
Culpan, R., Pelisoli, I., & Geier, S. 2021, A&A, 654, A107
Dawson, K. S., et al. 2016, AJ, 151, 44
Deason, A. J., et al. 2021, MNRAS, 501, 5964
Domínguez Sánchez, H., Huertas-Company, M., Bernardi, M., Tuccillo, D., & Fischer, J. L. 2018, MNRAS, 476, 3661
Dosovitskiy, A., et al. 2021, in Proc. 9th Int. Conf. Learning Representations, ICLR 2021, OpenrRview.net (Vienna: ICLR), id. YicbFdNTTy
Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. 2010, Int. J. Comput. Vision, 88, 303
Girshick, R. 2015, in Proc. IEEE Int. Conf. Computer Vision (ICCV), ed. E. Mortensen & S. Fidler (Los Alamitos: IEEE Computer Society), 1440
He, K., Zhang, X., Ren, S., & Sun, J. 2014, in Computer Vision – ECCV 2014 (Part III), ed. D. Fleet, et al. (Cham: Springer), 346
He, K., Zhang, X., Ren, S., & Sun, J. 2016, in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), ed. E. Mortensen & K. Saenko (Los Alamitos: IEEE Computer Society), 770
He, Z., Qiu, B., Luo, A.-L., Shi, J., Kong, X., & Jiang, X. 2021, MNRAS, 508, 2039
Hendrycks, D., & Gimpel, K. 2017, Proc. 5th Int. Conf. Learning Representations, ICLR 2017, OpenrRview.net (Vienna: ICLR), id. Bk0MRI5lg
Hochreiter, S., & Schmidhuber, J. 1997, Neural Comput., 9, 1735
Jacobs, C., et al. 2019, ApJS, 243, 17
Li, N., & Thakar, A. R. 2008, Comput. Sci. Eng., 10, 18
Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. 2014, in Computer Vision – ECCV 2014 (Part V), ed. D. Fleet, et al. (Cham: Springer), 740
Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. 2018, in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), ed. E. Mortensen & W. Brendel (Los Alamitos: IEEE Computer Society), 8759
Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. 2022, in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR) (Los Alamitos: IEEE Computer Society), 11966
Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., & Khudanpur, S. 2010, in Proc. Interspeech 2010, 11th Annual Conf. ISCA (Baixas: International Speech Communication Association), 1045
Misra, D. 2020, Paper presented at 31st British Machine Vision Virtual Conf. (British Machine Vision Association), id. 928
Monaco, L., Bellazzini, M., Ferraro, F. R., & Pancino, E. 2003, ApJ, 597, L25
Montenegro, K., Minniti, D., Alonso-García, J., Hempel, M., Saito, R. K., Beers, T. C., & Brown, D. 2019, ApJ, 872, 206
Morgan, R., et al. 2022, ApJ, 927, 109
Neubeck, A., & Van Gool, L. 2006, in Proc. 18th Int. Conf. Pattern Recognition (ICPR'06), Vol. 3 (Los Alamitos: IEEE Computer Society), 850
Newberg, H. J., et al. 2003, ApJ, 596, L191
Niederste-Ostholt, M., Belokurov, V., Evans, N. W., & Peñarrubia, J. 2010, ApJ, 712, 516
Paszke, A., et al. 2019, in Advances in Neural Information Processing Systems 32 (Red Hook: Curran Associates), 7994
Petersen, M. S., & Peñarrubia, J. 2021, Nat. Astron., 5, 251
Preston, G. W., Shectman, S. A., & Beers, T. C. 1991, ApJ, 375, 121
Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016, in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (Los Alamitos: IEEE Computer Society), 779
Redmon, J., & Farhadi, A. 2017, in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (Los Alamitos: IEEE Computer Society), 6517
Redmon, J., & Farhadi, A. 2018, arXiv:1804.02767
Ren, S., He, K., Girshick, R., & Sun, J. 2015, in Advances in Neural Information Processing Systems 28 (NIPS 2015), ed. C. Cortes, et al. (Red Hook: Curran Associates), 91
Robitaille, T., Deil, C., & Ginsburg, A. 2013, Astrophysics Source Code Library, ascl:2011.023
Ruhland, C., Bell, E. F., Rix, H.-W., & Xue, X.-X. 2011, ApJ, 731, 119
Santucci, R. M., et al. 2015, ApJ, 813, L16
Schlegel, D. J., Finkbeiner, D. P., & Davis, M. 1998, ApJ, 500, 525
Sirko, E., et al. 2004, AJ, 127, 899
Sommer-Larsen, J., & Christensen, P. R. 1986, MNRAS, 219, 537
Sommer-Larsen, J., Christensen, P. R., & Carter, D. 1989, MNRAS, 238, 225
Song, X., Dai, Y., Zhou, D., Liu, L., Li, W., Li, H., & Yang, R. 2020, in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), ed. E. Mortensen & W. Brendel (Los Alamitos: IEEE Computer Society), 5630

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. 2013, Proc. Machine Learning Res., 28, 1139

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. 2017, in Advances in Neural Information Processing Systems 30 (NIPS 2017), ed. U. V. Luxburg, et al. (Red Hook: Curran Associates), 5999

Wang, C.-Y., Mark Liao, H.-Y., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. 2020, in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), ed. E. Mortensen & W. Brendel (Los Alamitos: IEEE Computer Society), 1571

Whitten, D. D., et al. 2019, ApJ, 884, 67

Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., & Zhang, L. 2021, in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), ed. E. Mortensen (Los Alamitos: IEEE Computer Society), 22

Xue, X. X., et al. 2008, ApJ, 684, 1143

Xue, X.-X., et al. 2011, ApJ, 738, 79

York, D. G., et al. 2000, AJ, 120, 1579

Yu, W., Luo, M., Zhou, P., Si, C., Zhou, Y., Wang, X., Feng, J., & Yan, S. 2022, in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), (Los Alamitos: IEEE Computer Society), 10809

Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., & Wu, W. 2021, in Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), ed. E. Mortensen (Los Alamitos: IEEE Computer Society), 559

Zhang, Y., et al. 2020, in Proc. 2020 IEEE 17th Int. Symp. Biomedical Imaging (ISBI) (Piscataway: IEEE), 217

Downloaded from https://academic.oup.com/pasj/article/75/6/1311/7330817 by Shandong University Library, on 19 April 2025

© The Author(s) 2023. Published by Oxford University Press on behalf of the Astronomical Society of Japan.
All rights reserved. For permissions, please e-mail: journals.permissions@oup.com