

SPFFN: A Fusion-based Deep Learning Framework for Stellar Photometric Feature Recognition (Postprint)

Authors: Peng Liu, Hao Fu, Shuai Meng, Cheng Zeng, Yangyang Guo, Xue Mei and Shengqing Yao

Date: 2025-09-28T12:14:27+00:00

Abstract

Stellar classification is a fundamental task in astronomical data analysis. Photometric data offer a significant advantage over spectral data in terms of data volume. Their lower acquisition cost and broader coverage make them more suitable for stellar classification applications. This study selects photometric data from the SDSS DR18. Instead of using traditional RGB image formats, a series of preprocessing steps were applied to generate five-channel Numpy files as the data set. To enhance stellar classification performance, we propose a deep learning model based on photometric feature fusion-Stellar Photometric Features Fusion Network. Additionally, we introduce the Dynamic Enhanced Stellar Squeeze-and-Excitation module, designed to optimize the weight allocation of different photometric bands in the classification task, and investigate the impact of each band's features on classification performance. Ultimately, we found that the information from the r and z bands played a more crucial role in the stellar classification task, achieving a final classification accuracy of 87.47%, thereby demonstrating the effectiveness of photometric data in stellar classification.

Full Text

Preamble

Research in Astronomy and Astrophysics, 25:075013 (16pp), 2025 July © 2025. National Astronomical Observatories, CAS and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved. Printed in China. <https://doi.org/10.1088/1674-4527/adda7d> CSTR: 32081.14.RAA.adda7d

SPFFN: A Fusion-based Deep Learning Framework for Stellar Photometric Feature Recognition

Peng Liu¹, Hao Fu¹, Shuai Meng¹, Cheng Zeng¹, Yangyang Guo¹, Xue Mei¹, and Shengqing Yao²

¹ College of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing 211816, China; mx@njtech.edu.cn

² China Construction Industrial & Energy Engineering Group Co., Ltd. Nanjing, Nanjing 210004, China

Received 2025 March 1; revised 2025 April 28; accepted 2025 May 13; published 2025 June 18

Abstract

Stellar classification is a fundamental task in astronomical data analysis. Photometric data offer a significant advantage over spectral data in terms of data volume, lower acquisition cost, and broader coverage, making them more suitable for large-scale stellar classification applications. This study selects photometric data from SDSS DR18 and applies a series of preprocessing steps to generate five-channel Numpy files as the dataset, rather than using traditional RGB image formats. To enhance stellar classification performance, we propose a deep learning model based on photometric feature fusion—the Stellar Photometric Features Fusion Network (SPFFN). Additionally, we introduce the Dynamic Enhanced Stellar Squeeze-and-Excitation (DES-SE) module, designed to optimize weight allocation across different photometric bands in the classification task, and investigate the impact of each band's features on classification performance. Our results demonstrate that information from the r and z bands plays a more crucial role in stellar classification, achieving a final classification accuracy of 87.47% and thereby validating the effectiveness of photometric data for stellar classification.

Key words: methods: data analysis -techniques: image processing -stars: imaging

1. Introduction

With the continuous development and rapid expansion of large astronomical observatories, astronomers now have access to unprecedented volumes of astronomical data. Stars, as fundamental celestial bodies in the universe, play a crucial role in various fields including stellar evolution, galaxy structure, and cosmological studies. Accurate stellar classification is not only essential for understanding the physical properties and evolutionary processes of stars but also provides foundational data for further research on galaxies and cosmic structures.

Stellar data primarily include spectral and photometric data. Spectral data, obtained by dispersing stellar light into its constituent wavelengths, provide high-precision information about a star's chemical composition, temperature, density, velocity, and other physical parameters (Saha 1921). However, spectral observations are typically time-consuming and limited in data acquisition. In contrast, photometric data, obtained through observations across different bands, record a star's radiation intensity at various frequencies. While less information-dense than spectral data, photometric data offer higher acquisition efficiency and broader coverage, making them valuable for stellar classification research (Dolphin 2000).

Existing stellar classification studies predominantly employ the Morgan-Keenan (MK) classification system (Morgan & Keenan 1973), which categorizes stars into seven types—O, B, A, F, G, K, and M—based on their surface temperatures, with each type representing a specific temperature range. Each range is further subdivided into subclasses numbered from 0 to 9 to account for finer temperature variations.

Based on this classification system, various stellar classification methods have emerged, including template-matching methods, traditional machine learning algorithms, and deep learning approaches. Template-matching methods are typically applied to spectral data, where the observed spectrum of an astronomical object is compared with a set of standard template spectra to determine the most suitable match, thereby inferring the object's properties (Duan et al. 2009). This approach has been widely applied in astronomy. Stringer et al. (2019) used template fitting and random forest classification to identify RR Lyrae variable stars in multi-band sparse sampling data from the Dark Energy Survey. In stellar classification, Zhong et al. (2015) extracted spectral data from LAMOST (Cui et al. 2012) and applied template-matching to automatically classify over 2600 late-type K and M dwarfs according to their metallicity. Cai et al. (2024) proposed a synthetic-to-observed spectral transformation (SOST) method based on Generative Adversarial Networks (GANs) to expand the stellar spectral template library. However, for stellar classification tasks, the results are highly dependent on the quality and quantity of the template library, and creating a high-quality template library is relatively challenging. With the dramatic increase in astronomical data, template-matching methods struggle to meet the classification efficiency required for large-scale data processing, gradually revealing their shortcomings in practical applications.

In contrast to template-matching methods that rely on predefined libraries, machine learning algorithms can automatically learn features from data, allowing them to adapt more flexibly to various data types and classification tasks (Kuntzer et al. 2016). Traditional machine learning algorithms such as Support Vector Machines (SVM), decision trees, and random forests effectively extract key information from both spectral and photometric data, achieving efficient and accurate classification (Qi 2022). This ability to automatically extract features enables machine learning methods to exhibit stronger adaptability when

handling high-dimensional and complex astronomical data, eliminating dependence on template library quality and quantity. Consequently, machine learning methods have been widely applied in astronomy (e.g., Díaz-Hernández et al. 2014; Li et al. 2019; Mehta et al. 2022; Abd-elaziem et al. 2023). However, with advancing observational capabilities, the exponential growth of astronomical data, along with their high dimensionality, nonlinearity, and noise, still presents significant challenges for traditional machine learning methods.

To overcome these limitations in handling large-scale and complex astronomical data, deep learning techniques have gradually gained widespread application in stellar classification (Róžański et al. 2022; Tan et al. 2024). Unlike traditional methods that rely on manual feature extraction, deep learning automatically learns features from raw data through multi-layer neural networks, enabling more efficient processing of high-dimensional features in complex astronomical datasets (Sharma et al. 2020). In stellar classification tasks, deep learning methods have demonstrated significant advantages when applied to both spectral (Fu et al. 2024) and photometric (Shi et al. 2023) data.

In this study, we selected photometric data for O, B, A, F, G, K, and M-type stars from the Sloan Digital Sky Survey (SDSS) Data Release 18 (DR18). After preprocessing steps including localization and mapping, we generated five-channel Numpy files to replace commonly used RGB images as the dataset. We propose the Stellar Photometric Features Fusion Network (SPFFN) model for stellar classification tasks and introduce the Dynamic Enhanced Stellar Squeeze-and-Excitation (DES-SE) module to optimize weight allocation across different photometric bands. Through a series of experiments and comparative analyses, we explored the importance of each photometric band in stellar classification and validated the effectiveness of SPFFN. The structure of this paper is as follows: Section 2 introduces the photometric data preprocessing methods and the SPFFN model architecture; Section 3 describes the dataset source, DES-SE parameter optimization, evaluation of photometric band impact, and training strategy; Section 4 presents comparative experiments with various classification algorithms; and Section 5 provides a summary and future outlook.

2.1. Mapping of Photometric Data and Numpy Construction

After obtaining image data of target stars from the SDSS database, the next step is to spatially align the star maps from different bands (u, g, r, i, z) to ensure they share the same reference frame. We employ a World Coordinate System (WCS)-based image reprojecting method (Mink 1997) for this purpose. WCS is a standard tool for converting pixel coordinates into celestial coordinates (such as R.A. and decl.) via matrix transformations to handle spatial coordinate conversion. For each image band, pixel coordinates are first converted into celestial coordinates using WCS, and then the celestial coordinates of the other bands are remapped back to the pixel coordinates of the r band using the r band's WCS header file. The specific transformation process consists of two parts.

Let the pixel position in the image be (x, y) , with the goal of converting it to celestial coordinates (α, δ) , where α is R.A. and δ is decl. This transformation can be expressed as Equation (1):

$$\begin{pmatrix} \alpha \\ \delta \end{pmatrix} = T \begin{pmatrix} x \\ y \end{pmatrix} + b$$

where T represents the transformation matrix including rotation, scaling, and translation operations, and b is the translation vector. Correspondingly, the process of converting celestial coordinates (α, δ) back to pixel coordinates (x', y') is shown in Equation (2):

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = T^{-1} \begin{pmatrix} \alpha \\ \delta \end{pmatrix} - b$$

where T^{-1} represents the inverse transformation matrix and b is the translation vector.

To improve alignment accuracy, we employed an interpolation algorithm during the image reprojection process. The nearest-neighbor interpolation method was used to compute appropriate pixel values for each new target coordinate position, ensuring smooth transitions across different resolutions and coordinate positions. In this way, all band data ultimately share the same spatial reference frame, providing a solid foundation for subsequent image synthesis and analysis.

We adopted a five-channel Numpy processing approach to fully utilize the five bands (u, g, r, i, z) in the SDSS data. Specifically, we revised the processing method to handle image data from all five bands, performing unified brightness mapping and color adjustments. Unlike traditional methods that map only three bands, our approach simultaneously processes data from all five bands, ensuring each band is fully utilized during synthesis. As a result, the generated five-channel images exhibit superior color stability and accuracy, avoiding information loss and errors associated with single-channel normalization. Additionally, the converted five-channel data can be split into three channels for display as needed or used directly for subsequent analysis.

In the image-processing workflow, we first spatially align the image data from each band to ensure they are processed in the same coordinate system. Then we apply a custom function to perform brightness mapping, normalization, and color mapping, ensuring consistent processing across all band data. The specific process is as follows: (1) **Brightness mapping**. To enhance image visibility and contrast, a brightness mapping function is first applied. The core idea is to transform pixel values of each band image using a function similar to arcsinh. For each band, the brightness representation is given by Equation (3):

$$L_{\text{band}}(x, y) = \text{arcsinh} \left(\frac{I_{\text{band}}(x, y)}{S_{\text{band}}} \right)$$

where $I_{\text{band}}(x, y)$ is the pixel value at position (x, y) in the original image and S_{band} is the scaling factor for the band. (2) **Normalization**. After brightness mapping, normalization is performed to ensure brightness values of all bands lie within the same range. Specifically, pixel values of all bands are standardized to the range $[0, 1]$ as shown in Equation (4):

$$L_{\text{band}}^{\text{norm}}(x, y) = \frac{L_{\text{band}}(x, y) - \min(L_{\text{band}})}{\max(L_{\text{band}}) - \min(L_{\text{band}})}$$

where $\min(L_{\text{band}})$ and $\max(L_{\text{band}})$ are the minimum and maximum values of the brightness image for the respective band. (3) **Color Mapping and Channel Synthesis**. After normalization, the five-band images are stacked into a unified five-channel Numpy array. Before this, all bands undergo the same brightness mapping using the inverse hyperbolic sine (arcsinh) function, which provides consistent dynamic range compression across channels. In contrast, traditional RGB synthesis methods often apply separate or fixed mappings to selected bands, potentially resulting in inconsistent brightness or color representation when extended to five-band data. By treating all bands equally and processing them synchronously, our approach minimizes channel imbalance and enhances the stability of multi-band image synthesis.

[Figure 1: see original paper] shows a complete star map after preprocessing, with the irg bands mapped to an RGB image. After alignment is completed, the next step is to perform brightness mapping and other necessary transformations on the raw measurement data to generate images better suited for visual display. Traditional methods typically use the Lupton algorithm (Lupton et al. 2004), which selects three bands from the five (u, g, r, i, z)—for example, urz or irg—and converts them into an RGB image, with each band corresponding to one color channel. [Figure 2: see original paper] displays the RGB image generated using the irg bands, while [Figure 3: see original paper] depicts the RGB image generated using the urz bands. This method enhances image contrast by normalizing the minimum and maximum values of each of the three selected bands from the ugriz filter set and using the arcsinh function to map values to an appropriate brightness range.

However, this traditional method has several limitations. First, key information contained in the two discarded bands is not effectively utilized, which may result in image information loss. This is particularly problematic in multi-band datasets, where some astronomical features may only appear prominently in excluded bands (Lin et al. 2022). Second, when normalizing maximum and minimum values, these are typically computed based on three selected channels. This approach can introduce color inconsistencies when extended to five-channel data, as different bands often have varying brightness distributions and dynamic ranges (Bertin 2011). Normalizing based on a partial subset of channels may therefore compromise the accuracy and stability of the synthesized image. These limitations become especially evident when dealing with complex astronomical

images, as traditional RGB synthesis may fail to capture characteristic differences among all five bands (Lupton et al. 2004).

To overcome these limitations, this study proposes a five-channel approach. The resulting five-channel image data are finally stored in Numpy format, facilitating efficient reading and processing in subsequent steps. By utilizing the R.A. and decl. coordinates of stars from the SDSS database, we can accurately locate target stars within the five-channel Numpy file.

After localization, edge processing is applied to the r-band image to obtain clear edge information. In subsequent analysis, we found that most stars can be fully represented within a 16×16 pixel crop box. However, for some stars, cropping errors may occur due to the influence of nearby bright stars or halos, especially when multiple stars fall within the 16×16 pixel range. To ensure data accuracy, we apply dilation and erosion operations to the image, removing noise and refining edges to clearly define the target star. Specifically, erosion and dilation are applied to each channel image to eliminate surrounding noise and enhance target star boundaries. After processing, we retain the 16×16 pixel crop box to avoid introducing irrelevant star information, ensuring the cleanliness and stability of the final data. As shown in [Figure 4: see original paper], all processed image data are consolidated into a five-channel Numpy array, enabling efficient storage and subsequent analysis.

2.2. Dynamic Enhanced Stellar Squeeze-and-Excitation Module (DES-SE)

In this study, the input data consist of photometric images from the five SDSS bands (u, g, r, i, and z), which are processed and stacked into a five-channel Numpy file. Since different bands contribute differently to the photometric features of target stars, and some bands may be affected by noise or redundant information, we propose a dynamic channel weighting mechanism to adjust each band's contribution. This mechanism enhances the model's focus on key bands while suppressing interference from irrelevant information, and is specifically designed for stellar classification tasks as DES-SE.

As illustrated in [Figure 5: see original paper], its workflow is as follows: (1) **Global Average Pooling**. The input five-channel image undergoes global average pooling (GAP) across spatial dimensions to obtain a global feature vector for each channel, as shown in Equation (5):

$$z_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x_{c,i,j}$$

where c represents the channel index and H and W represent the image height and width, respectively, and $x_{c,i,j}$ denotes the pixel value at position (i, j) in channel c . (2) **Inter-channel Feature Compression**. The channel feature

vectors are input into the first fully connected layer to compress the channel dimension, followed by nonlinear mapping through the LeakyReLU activation function, as defined in Equation (6):

$$s = \text{LeakyReLU}(W_1 z)$$

where W_1 represents the weight matrix of the first fully connected layer. (3) **Channel Weight Allocation.** The compressed features are reshaped to fit the dynamic convolutional layer, and weight distribution is generated through 1×1 dynamic convolution, as expressed in Equation (7):

$$w = \sigma(\text{DynamicConv}(s))$$

where σ represents the sigmoid activation function, ensuring generated weights fall within the range $[0, 1]$. (4) **Weight Application.** The generated dynamic weights are applied to the original input features with channel-wise weighting, as written in Equation (8):

$$\hat{x}_{c,h,w} = w_c \cdot x_{c,h,w}$$

where (h, w) represents the spatial position and $\hat{x}_{c,h,w}$ is the channel feature after weighting.

The improved DES-SE module, based on the traditional Squeeze-and-Excitation (SE) module (Hu et al. 2018), significantly enhances dynamic modeling capability between channels by incorporating the LeakyReLU activation function (Glorot & Bengio 2010) and dynamic convolution (Chen et al. 2020). This allows more efficient adaptation to the complexity and characteristics of five-band photometric data. LeakyReLU not only preserves gradient flow in the negative value region but also significantly enhances the model's sensitivity to bands with weaker photometry that may contain key features. Dynamic convolution, by integrating spatial feature interactions, generates adaptive channel weight distributions, amplifying the contribution of key bands to the classification task while attenuating interference from noisy or redundant bands.

By combining LeakyReLU and dynamic convolution, DES-SE effectively addresses limitations of traditional modules in channel weight generation. It improves feature discrimination ability for five-band data and more efficiently models complementary relationships between bands. This dynamic modeling capability plays a crucial role in enhancing overall network performance through more rational distribution of channel weights, enabling the network input to optimize feature representation and assisting subsequent neural networks in performing classification tasks.

2.3. Stellar Photometric Features Fusion Network (SPFFN)

With the rapid development of deep learning, Convolutional Neural Networks (CNNs) have become an important method for stellar classification tasks. Compared to traditional template-matching and handcrafted feature extraction methods, neural networks can automatically learn complex features embedded in data. This is particularly advantageous for multi-band photometric data, where CNNs exhibit stronger feature extraction capabilities and higher classification accuracy. Therefore, applying deep learning to stellar classification is of significant importance for more efficient and accurate analysis of astronomical data.

In this study, stellar photometric data were processed through spatial alignment and brightness mapping, resulting in a $64 \times 64 \times 5$ Numpy file representing stellar photometric information. This data format preserves the luminosity distribution features of stars across the ugriz five bands. The resolution of the photometric information is relatively low and exhibits clear locality and band-specific differences. Based on these data characteristics and the classification advantages of neural networks, this study proposes SPFFN, designed to efficiently extract multi-scale features from photometric data and enhance classification performance.

The SPFFN network introduces the DES-SE channel attention mechanism in the input stage, where global spatial information is used to weight the importance of different band channels. This mechanism dynamically enhances key photometric information while suppressing redundant features. After channel attention weighting, the network adopts a backbone structure similar to VGG16 (Simonyan & Zisserman 2014) with small kernel convolutions. The use of small kernel convolutions (3×3) offers significant advantages for low-resolution input data. On one hand, it effectively captures local features of the stellar luminosity distribution. On the other hand, compared to larger kernels, small kernels significantly reduce the model's parameter count and computational complexity, satisfying the task's requirements for lightweight design (Zeiler & Fergus 2014).

The output feature maps from each convolutional layer in the network can be represented by Equation (9):

$$Y = \sigma(X * W + b)$$

where X represents the input feature map, W and b are the convolution kernel weights and biases, respectively, and $*$ denotes the convolution operation.

To further enhance the network's ability to represent stellar photometric data, SPFFN introduces the Convolutional Block Attention Module (CBAM; Woo et al. 2018) in each feature extraction block to strengthen feature expression. As diagrammed in [Figure 6: see original paper], CBAM combines channel attention

and spatial attention, enabling the model to adaptively highlight significant regions of the luminosity distribution across both channel and spatial dimensions, thereby allowing the network to adjust feature weights adaptively. Specifically, channel attention calculates the importance of each channel through GAP and Global Max Pooling (GMP), with the weight representation given by Equation (10):

$$M_c(F) = \sigma(W_1(\delta(W_0(F_{\text{avg}}^c)))) + \sigma(W_1(\delta(W_0(F_{\text{max}}^c))))$$

where F represents the input feature map, σ denotes the ReLU activation function, δ is the Sigmoid function, and W_0 and W_1 are the weights of the fully connected layers. Through channel attention, the network can adaptively adjust weights based on the importance of photometric information from different bands. Spatial attention, on the other hand, focuses on significant regions of the feature map through convolutional operations, with the weight represented by Equation (11):

$$M_s(F) = \sigma(\text{Conv}_{7 \times 7}(\text{Concat}(GAP(F), GMP(F))))$$

where σ is the Sigmoid function and Concat represents channel concatenation. Finally, CBAM enhances the attention-weighted feature map through element-wise multiplication, as shown in Equation (12):

$$F_{\text{CBAM}} = M_c(F) \odot M_s(F) \odot F$$

where \odot represents element-wise multiplication and F_{CBAM} is the feature map after CBAM weighting. Through the CBAM module, the SPFFN Block can adaptively capture complementary information between different bands of stellar photometric data and focus on significant regions of the spatial distribution. This design effectively enhances the network's ability to express key photometric features, further improving stellar classification accuracy.

To enhance the network's generalization ability while ensuring effective feature expression, SPFFN replaces traditional convolutions with depthwise separable convolutions. Depthwise separable convolution decomposes the standard convolution operation into two steps: Depthwise convolution, which extracts features within each channel, and Pointwise convolution, which aggregates features across channels (Chollet 2017). The computational complexity of this process can be represented as Equation (13):

$$\text{FLOPs}_{\text{separable}} = H \times W \times C_{\text{in}} \times K^2 + H \times W \times C_{\text{in}} \times C_{\text{out}}$$

where H and W represent the height and width of the feature map, C_{in} and C_{out} represent the input and output channel numbers, respectively, and K is the convolution kernel size.

This design not only reduces the number of parameters and computational complexity but, more importantly, demonstrates a significant advantage in preventing overfitting when dealing with stellar photometric data. Specifically, stellar photometric data exhibit high inter-band correlation and relatively regular spatial distribution, making the network prone to overfitting due to redundant feature expressions during learning. Traditional convolution processes both intra-channel and inter-channel features together, which may introduce unnecessary complexity and reduce the model's generalization ability. In contrast, depth-wise separable convolution splits the feature extraction process, allowing intra-channel and inter-channel information to be extracted independently, helping to reduce redundant learning and focus on the core features of each band's photometry. Additionally, this decomposition naturally introduces a sparse feature representation mechanism, effectively mitigating overfitting and improving the model's classification accuracy on the stellar photometric data test set.

To fully integrate multi-scale photometric features, SPFFN introduces a Feature Pyramid Network (FPN; Lin et al. 2017) for feature fusion in the network structure. FPN effectively combines high-level semantic information with low-level detailed features through a top-down path and lateral connections, progressively fusing feature maps at different scales, thereby enhancing the network's ability to represent stellar photometric data and improving classification performance.

In the input data, stellar photometric maps are represented as five-channel Numpy files, where the photometric distribution in different bands exhibits significant spatial features. The central region has higher photometric values, showing a "spherical" aggregation pattern, while the surrounding area approaches zero, representing the dark cosmic background. This spatial distribution makes stellar features quite distinct at different scales. For instance, the high-intensity core contains densely packed information, while the outer region, though photometry gradually decays, still holds important structural information about the star. Therefore, the network must capture multi-scale information, focusing on both high-level global photometric distribution and fine-grained features such as the star's boundary and photometric decay region. The core idea of FPN is to fuse high-level semantic features with low-level detailed features via a top-down path and progressive upsampling. For the feature at the l -th layer, FPN generates the fused feature map through lateral connections and upsampling operations, as expressed in Equation (14):

$$P_l = \text{Upsample}(P_{l+1}) + C_l$$

where C_l is the lateral feature map at the l -th layer and P_{l+1} is the fused feature map from the previous layer.

This layer-by-layer fusion mechanism enables the network to effectively combine both high-level and low-level feature information. For stellar photometric data, high-level features contain global luminosity distribution information, capturing high-luminosity concentrated features at the star's center, which aids in

distinguishing intensity differences across various bands. Low-level features preserve rich spatial details, such as luminosity decay and contours at the star's edges, especially in outer regions where luminosity gradually diminishes. FPN effectively prevents loss of these details. The progressive upsampling and lateral fusion mechanism ensures complementarity of features at different scales, allowing the network to capture both luminosity concentration features and edge attenuation information, even at limited input resolutions. For stellar images in five-channel Numpy files, FPN enables the network to more comprehensively understand multi-scale characteristics of luminosity distribution across different bands, thereby improving classification accuracy and robustness.

Finally, the fused feature maps are classified through GAP (Hsiao et al. 2019) and a fully connected layer. The output of the fully connected layer can be expressed as Equation (15):

$$y = \sigma(W_i \cdot \text{GAP}(x) + b)$$

where x is the vector after global pooling, W_i represents the weights of the fully connected layer, σ is the ReLU activation function, and y denotes the classification result.

In summary, as visualized in [Figure 7: see original paper], the SPFFN network, tailored to the spatial distribution and band characteristics of five-band photometric data, integrates small kernel convolutions, the CBAM attention mechanism, depthwise separable convolutions, and FPN for multi-scale feature fusion. This architecture achieves efficient feature extraction and information integration. By enhancing focus on significant photometric regions, the network effectively captures both local and global features, improving the accuracy and robustness of stellar classification tasks while maintaining low computational complexity. This provides an efficient and accurate solution for classifying stellar photometric data. The SPFFN network architecture involved in this study has been uploaded to GitHub: <https://github.com/Fuhao-Edwin/SPFFN.git>.

3.1. Data Acquisition

The data for this study come from DR18 of SDSS (Almeida et al. 2023), published in 2023. SDSS uses a specially designed imaging system for multi-band observations, consisting of 30 CCD sensors, each with a resolution of 2048×2048 pixels, arranged in a matrix of 6 rows and 5 columns. Through a combination of CCD sensors and filters, SDSS provides comprehensive coverage of five optical bands: u, g, r, i, and z, spanning wavelengths from ultraviolet to near-infrared (u band: 3551 Å, g band: 4686 Å, r band: 6166 Å, i band: 7480 Å, z band: 8932 Å). During observations, the telescope operates in drift-scan mode, slowly moving along the celestial great circle, with celestial objects passing sequentially through the CCD sensors. The exposure time for each band

is 54 s, with the band sequence following r, i, u, z, and g. The time interval between different bands is approximately 71.7 s. This imaging method not only ensures comprehensive observation across multiple spectral bands but also provides high-quality multi-color photometric data for astronomical research.

The stellar photometric data in SDSS are pre-labeled with category information determined through various methods including spectral data analysis, template-matching, and machine learning. It should be noted that MK spectral classifications are not directly available in the SDSS photometric target tables. Instead, these labels are obtained from spectroscopic catalogs, where stellar types are assigned based on spectral analysis. In this study, we identified photometric targets with available spectra by cross-matching their celestial coordinates (R.A./decl.) and used the corresponding spectroscopic classifications as ground-truth labels for the photometric data. These labels are provided as part of the SDSS catalog and are used in this study to demonstrate the accuracy and effectiveness of the proposed classification method. While the data used in this study are pre-labeled, the proposed deep learning approach is also applicable to unlabeled stellar photometric data.

Following acquisition of pre-labeled stellar photometric data from SDSS DR18, the data preparation process begins with downloading the ugriz FITS files for all target stars. Each star is expected to have corresponding FITS images across the five bands. To ensure spatial consistency, all bands are reprojected and aligned using WCS-based transformations. After alignment, we apply an arcsinh-based brightness mapping (AsinhMapping) to enhance contrast and normalize brightness levels across bands. The mapped and aligned images are then merged to form five-channel Numpy star maps, where each channel corresponds to one photometric band.

Given a target star's R.A. and decl., its position is accurately located in the aligned sky image using WCS coordinate transformation. A square cutout centered on the star is then extracted from the five-channel image. The size of the cutout window $k \times k$ is gradually increased from an initial value until the non-background region accounts for at least 20% of the area (i.e., background does not exceed 80%). This dynamic window-sizing strategy helps ensure the extracted cutout contains sufficient stellar features while minimizing contamination from background regions or adjacent sources. Once the optimal window size is determined, the region is cropped to obtain the final five-channel Numpy array representing a single stellar object.

However, during actual data acquisition, several factors including catalog limitations, band incompleteness, and extraction failures led to substantial reduction in the final dataset size. Although we initially aimed to collect up to 60,000 stars per spectral class under predefined magnitude constraints, the available number of qualifying stars in SDSS DR18 varied considerably across types and often fell short of this target. The magnitude thresholds were designed to filter out overly faint sources with high noise levels but inevitably excluded a large portion of catalog entries. In the FITS downloading stage, many targets lacked one or

more bands due to incomplete sky coverage, observational inconsistencies, or repeated field overlaps, resulting in incomplete five-band data and necessitating their removal.

During subsequent preprocessing, additional losses occurred due to misalignment in WCS reprojection or failures in R.A./decl.-based localization. Finally, in the dynamic cropping phase, some stars were discarded because their cutouts contained excessive background, bright halos from nearby sources, or failed to meet the coverage threshold for valid feature extraction. After these filtering steps, the remaining high-quality samples, each with complete ugriz coverage, falling within the defined brightness range, and passing cropping criteria, were used to construct the final dataset. For each spectral type, several thousand five-band stellar samples were retained. To minimize selection bias, final samples were randomly selected from the pool of valid candidates within each class.

Based on established magnitude constraints for each spectral type (B-type: 22, A-type: 21, F-type: 20, G-type: 22, K-type: 21, and M-type: 21), the final number of valid stellar samples obtained for each class is as follows: O-type: 1305, B-type: 3634, A-type: 4926, F-type: 5209, G-type: 3950, K-type: 5863, and M-type: 8361. These samples constitute the complete, preprocessed dataset used in this study. To support reproducibility and facilitate future research, the full dataset has been made publicly available on the Zenodo research data repository: [10.5281/zenodo.14813823](https://zenodo.org/record/14813823). All stellar coordinates and the SPFFN network architecture are available at the GitHub link in Section 2.3.

3.2. Optimizing DES-SE and Evaluating Band Impact

To investigate the importance of each band in the stellar classification task, we designed and conducted a series of experiments aimed at systematically analyzing the contribution of different band information to classification performance. First, we explored the parameter settings of the DES-SE module, striving to find the optimal configuration. Based on this optimal configuration, we further investigated the importance of each band in the classification task and validated our hypotheses through additional experiments.

In the DES-SE module, differences and redundant noise in photometric distributions across bands make the nonlinear mapping capability of the activation function in channel feature compression and dynamic modeling particularly critical. This study systematically assessed the impact of activation functions on classification performance, aiming to reveal the effects of different activation functions in handling these differences and noise, thereby optimizing feature extraction and information integration capabilities.

In the experimental setup, we evaluated the effectiveness of three activation functions—ReLU (Nair & Hinton 2010), LeakyReLU (Qi et al. 2023), and Mish (Misra 2019)—within the DES-SE module. [Figure 8: see original paper] presents accuracy variation curves during model training for these three activation functions. The results indicate that ReLU, due to its property of completely dis-

carding gradients in the negative region, showed significant limitations when handling complex data distributions, particularly in its inability to capture features in regions with weaker photometry, leading to inferior overall performance. Mish, as a more complex nonlinear activation function, was able to enhance feature representation in certain cases but still lagged behind LeakyReLU in training convergence speed and final accuracy while introducing additional computational overhead. In contrast, LeakyReLU, with its ability to maintain gradient flow in the negative region, proved effective in handling bands with weaker photometry that may still contain critical features, enhancing the model's ability to capture complementary information across bands. Leveraging this advantage, LeakyReLU demonstrated faster convergence early in training and ultimately achieved the highest classification accuracy.

Furthermore, considering that the negative slope parameter directly affects gradient flow in the negative region of LeakyReLU, which impacts the model's feature extraction ability, we conducted tests with five different negative slope values: 0.0001, 0.0005, 0.001, 0.01, and 0.1. The experimental results are summarized in .

** Variation in Classification Accuracy for Different Negative Slope Values**

Negative Slope (LeakyReLU)	Accuracy
0.0001	87.23%
0.0005	87.36%
0.001	87.47%
0.01	87.02%
0.1	86.04%

When the negative slope is set to smaller values (0.0001, 0.0005, and 0.001), the model's classification accuracy is 87.23%, 87.36%, and 87.47%, respectively, with minimal variation. This indicates that maintaining a certain level of gradient flow in the negative region helps the model capture key features in weak light bands while preventing excessive loss of feature information. As the negative slope increases to 0.01, classification accuracy decreases to 87.02%. Although the drop is small, this trend suggests that a larger negative gradient may interfere with the model's ability to capture important features in the positive region. Further increasing the negative slope to 0.1 results in a significant accuracy drop to 86.04%, likely due to an excessively strong negative slope causing the model to overemphasize low-light features while suppressing effective learning of key light information.

In addition, the reduction parameter in the channel attention mechanism determines the feature compression ratio, which significantly affects the model's feature representation ability and complexity. Since the input Numpy file has five channels, the reduction value can only be set to 2, 3, or 4 to ensure the compressed channel count is at least 1. The experimental results show that when

reduction is set to 4, the model achieves the highest classification accuracy, approximately 1% higher than other settings. This suggests that an appropriate compression ratio strikes the optimal balance between retaining key features and reducing redundant information.

In summary, the experimental results demonstrate that the choice of activation function and reduction parameter significantly impacts DES-SE module performance. Ultimately, we selected LeakyReLU as the activation function, set the negative slope parameter to 0.001, and set reduction to 4. With this configuration, the model achieved the highest classification accuracy, validating the superiority and rationality of these settings.

With the optimal parameter configuration, we further explored the importance of different bands in photometric data for stellar classification. By saving dynamic weight distributions after multiple trials, we observed that weight parameters varied across experiments, mainly due to random initialization of the neural network and the non-deterministic nature of the training process. Nevertheless, the overall trend indicates that the r and z bands are relatively more important, while the u band holds the least significance. In comparison, the g and i bands have less impact on classification performance.

To validate this observation, we designed further experiments. For weight parameters, values closer to 0 indicate less importance, while values closer to 1 indicate greater importance. In the previous DES-SE module, initial weights for the five bands were all set to 0.5 and dynamically adjusted during training. In this experiment, we fixed the weight of one ugriz band to either 0.1 or 0.9 while keeping the remaining bands at 0.5, and observed changes in classification performance. The results are shown in [Figure 9: see original paper]. When the weights of the r and z bands were fixed at 0.9, classification accuracy improved, whereas fixing them at 0.1 caused accuracy to drop significantly. This further confirmed the importance of these two bands. In contrast, for the u band, setting its weight to 0.9 decreased accuracy, but setting it to 0.1 improved accuracy, suggesting that the u band may contain significant redundant or noisy information. Adjusting weights of the g and i bands had little effect on overall model performance. These results validate the crucial role of the r and z bands in stellar classification tasks while revealing that the u band has lower importance and may even negatively impact performance, providing valuable insights for optimizing photometric data usage and feature selection in future work.

3.3. Training Strategy and Experimental Results

All experiments in this study were conducted on a computing platform equipped with an Intel(R) Xeon(R) Platinum 8352V CPU and an NVIDIA RTX 4090 (24GB) GPU, running Ubuntu 20.04. The experimental environment consisted of Python 3.8 and PyTorch 1.11.0. The network model used for the stellar classification task was SPFFN, which incorporates multiple convolutional layers, depthwise separable convolutions, the DES-SE module, and CBAM. To prevent

overfitting and improve generalization, dropout rates for the two Dropout layers were set to 0.5 (Srivastava et al. 2014).

In terms of training strategy, no pre-trained weights were used, and the number of output classes was set to 7 for the seven-class classification task. The total training epochs were set to 200, and batch size was set to 8 to ensure model convergence within an appropriate timeframe while maintaining good computational efficiency. The optimizer chosen was AdamW (Loshchilov 2017), with a learning rate of $5e-5$ and weight decay of $5e-2$, which help enhance generalization and prevent overfitting.

For model evaluation, multiple metrics were used including accuracy, precision, recall, and F1-score. presents SPFFN' s classification results on the seven-class star luminosity data.

** Classification Results of SPFFN**

Class	True Positives	True Negatives	False Positives	False Negatives	Precision	Recall	F1-Score	Accuracy
O	95	32869	282	1210	0.252	0.073	0.113	0.964
B	2084	30789	1362	221	0.605	0.904	0.725	0.958
A	3832	28672	1479	473	0.721	0.890	0.797	0.967
F	4302	28315	836	1003	0.837	0.811	0.824	0.967
G	2729	29872	1283	1472	0.680	0.650	0.665	0.933
K	5271	27542	609	934	0.896	0.849	0.872	0.974
M	8238	24768	93	245	0.989	0.971	0.980	0.996
Average	-	-	-	-	0.854	0.835	0.840	0.965

The performance metrics across categories indicate that the model achieves high precision, recall, and F1-scores in most categories, with particularly strong performance in the M and K classes, where F1-scores reach 0.980 and 0.872, respectively. However, the O and G classes exhibit relatively weaker performance with declines in both precision and recall. Overall, the model achieves an accuracy of 0.87468, with weighted average precision, recall, and F1-scores of 0.854, 0.835, and 0.840, respectively. These results suggest that the model performs well in star luminosity classification tasks and demonstrates strong ability to handle data from different categories.

To comprehensively demonstrate model performance, various visualization methods were employed in the experimental results, including accuracy-loss curves, confusion matrices, and t-SNE dimensionality reduction. [Figure 10: see original paper] illustrates the training and validation accuracy changes during SPFFN model training, showing that the model gradually converges with training accuracy steadily increasing and stabilizing. [Figure 11: see original paper] shows training and validation loss changes, where training loss

decreases rapidly and validation loss declines progressively, indicating good performance on both sets. Training accuracy ultimately converges at 90%, while validation accuracy reaches 87.47%. Training loss converges at 0.27, and validation loss at 0.41.

[Figure 12: see original paper] presents the confusion matrix of SPFFN' s final classification results on the test set, providing an intuitive view of classification performance across different categories. Values on the diagonal represent correctly classified samples, while off-diagonal values indicate misclassifications. Most predictions are concentrated along the diagonal, suggesting good performance in classifying the majority of categories. Notably, predictions for the M class are almost perfect, with few misclassifications. Although there are slightly more misclassifications for the O and G classes, the overall distribution reflects ideal classification performance.

As shown in [Figure 13: see original paper], we applied the t-SNE algorithm to perform dimensionality reduction and visualize feature distribution. t-SNE can map high-dimensional features into two-dimensional space, allowing easier observation of distributions among different categories (Van der Maaten & Hinton 2008). Points in the figure, represented by different colors, correspond to the seven star types (categories 0-6). The visualization shows that most categories exhibit clear clustering in feature space, though some overlap between categories is observed. Overall, the model performs well in classifying most categories, but for certain categories such as O-type stars, some classification bias remains, indicating room for improvement in distinguishing these more complex categories.

4.1. Model Performance and Comparative Analysis

To validate the effectiveness of the proposed model, various comparative experiments were designed, selecting representative classification algorithms for comparison. These include classic machine learning methods (such as Random Forest; Breiman 2001), well-known deep learning architectures (such as ResNet50; He et al. 2016 and DenseNet121; Huang et al. 2017), and recently proposed deep learning models in astronomy (such as SFNet; Fu et al. 2024 and SCNet; Shi et al. 2023). Through these comparative experiments, this study aims to comprehensively evaluate and compare the performance of the proposed model in stellar luminosity data classification tasks, thereby visually verifying its superiority and application potential.

The Random Forest algorithm is an ensemble learning-based classification method that constructs multiple decision trees and determines final classification through voting on their predicted results. In this experiment, input data were uniformly scaled to 64×64 size, maintaining five feature channels. Default hyperparameter configuration was used, including 100 trees and unrestricted tree depth, to ensure stable training and avoid overfitting.

ResNet50 and DenseNet121 are classic deep learning architectures widely used

in image classification tasks. ResNet50 alleviates the vanishing gradient problem through residual connections, allowing the network to effectively train deeper layers. In contrast, DenseNet121 employs dense connections where each layer is connected to all previous layers, enhancing feature reuse and gradient flow, thus improving representational capacity. In this experiment, hyperparameters for ResNet50 and DenseNet121 were set consistent with those used in the SPFFN model, including optimizer, learning rate schedule, and batch size, ensuring fair comparison between networks in the stellar luminosity image classification task.

SFNet and SCNet are recent deep learning models specifically proposed for stellar classification in astronomy. SFNet is primarily used for classifying stellar spectral data and is particularly effective in processing stellar wavelet images generated by continuous wavelet transforms (Fu et al. 2024). The network employs an adaptive feature extraction module to effectively capture both local and global features in stellar spectral data, demonstrating superior classification performance. In this comparative experiment, SFNet's hyperparameters were set to match those of the SPFFN model. SCNet, on the other hand, is specifically designed for stellar luminosity data and processes RGB images generated from different band combinations. SCNet uses RGB images created from gri and urz three-band combinations to extract and integrate features across all luminosity bands, achieving excellent performance in seven-class classification tasks. In the study by Shi et al. (2023), SCNet achieved an accuracy of 86.1% on this task.

presents the comparative experimental results of different models in the stellar photometric image classification task.

**** Comparative Experimental Results of Different Models****

Model	Accuracy	Average Precision	Average Recall	Average F1-Score
Random Forest	83.62%	82.15%	81.23%	81.68%
ResNet50	85.34%	83.21%	82.45%	82.83%
DenseNet121	85.67%	83.89%	82.91%	83.40%
SFNet	86.23%	84.12%	83.34%	83.73%
SCNet	86.10%	84.05%	83.21%	83.63%
SPFFN	87.47%	85.42%	84.36%	84.89%

The SPFFN model performs excellently across all evaluation metrics, achieving an accuracy of 87.47%, average precision of 85.42%, average recall of 84.36%, and average F1-score of 84.89%. In contrast, the machine learning algorithm Random Forest shows relatively weaker performance with an accuracy of 83.62%. Although ResNet50 and DenseNet121, as classic deep learning architectures, possess advantages in feature extraction, they are prone to overfitting in stellar photometric data classification tasks and do not exhibit superior accuracy. SFNet

and SCNet, which have been recently developed specifically for stellar classification in astronomy, outperform other classification algorithms but still fall short of the classification performance of the proposed SPFFN model. Overall, SPFFN demonstrates superior performance compared to all compared models, confirming its effectiveness and potential for stellar classification tasks.

4.2. Comparison with Spectral Classification

Although spectral data offer higher precision for stellar classification due to their rich physical content—such as absorption line features and elemental abundances—photometric data present significant advantages in terms of scalability and accessibility. Spectral classification is often constrained by high observational costs and time-intensive procedures, limiting its practicality for large-scale sky surveys. In contrast, photometric data can be efficiently obtained in bulk across wide sky regions using multi-band imaging systems, such as those deployed by SDSS, thereby enabling rapid and extensive sky coverage.

While spectral classification excels at fine-grained distinctions including luminosity classes and chemical composition, photometric classification remains broadly consistent with spectral methods in categorizing stars by effective temperature. Photometric color indices (e.g., $g - r$, $r - i$) exhibit strong correlations with stellar surface temperatures, which are also fundamental parameters in spectral classification. This alignment ensures that photometric classification can effectively reflect the overall framework of traditional spectral schemes, particularly the MK classification system.

The aim of this study is not to replace spectral classification but to harness the efficiency and availability of photometric data for scalable stellar analysis and to explore the relative importance of different photometric bands in classification tasks. With the vast volume of stellar sources captured in modern sky surveys, models such as SPFFN offer a practical and accurate solution for preliminary classification, source filtering, and candidate pre-selection for follow-up spectroscopic studies. As such, photometric classification serves as a complementary approach to spectral analysis, extending the reach of stellar research across broader and deeper datasets.

5. Conclusion

In this study, our goal is to classify stellar photometric data using deep learning techniques. We obtained seven categories of stellar photometric data from SDSS DR18, moving away from traditional RGB image formats and instead using Numpy files generated through a series of preprocessing steps specifically designed for stellar classification. We propose the Stellar Photometric Features Fusion Network (SPFFN), which introduces the Dynamic Enhanced Stellar Squeeze-and-Excitation (DES-SE) module to optimize weight allocation across different photometric bands. Through extensive experiments, we systematically investigated the impact of each band on classification performance and

validated the effectiveness of the SPFFN model.

Our results demonstrate that the r and z bands play a more crucial role in stellar classification tasks, while the u band has relatively low importance and may even negatively impact performance. The SPFFN model achieved a final classification accuracy of 87.47%, outperforming traditional machine learning algorithms and existing deep learning models. The integration of small kernel convolutions, CBAM attention mechanism, depthwise separable convolutions, and FPN for multi-scale feature fusion enables SPFFN to efficiently extract and integrate photometric features while maintaining low computational complexity.

This study not only provides an efficient and accurate solution for stellar classification but also offers valuable insights into the relative importance of different photometric bands. Future work will focus on expanding the dataset to include more stellar types and exploring the application of the proposed method to other astronomical classification tasks. The combination of photometric data efficiency and deep learning capability opens new avenues for large-scale stellar analysis in the era of big data astronomy.

References

- Abd-elaziem, M. A., El-Baz, A. S., & Hassan, M. A. 2023, *J. Astrophys. Astron.*, 44, 1
- Almeida, A., Prieto, A., & Ahumada, R. 2023, *ApJS*, 265, 44
- Bertin, E. 2011, in *ASPC Conf. Series*, 442, 435
- Breiman, L. 2001, *Mach. Learn.*, 45, 5
- Cai, Y., Zhang, J., & Li, X. 2024, *MNRAS*, 527, 11045
- Chen, Y., Dai, X., Liu, M., et al. 2020, in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE)*, 520
- Chollet, F. 2017, in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE)*, 1800
- Cui, X.-Q., Zhao, Y.-H., Chu, Y.-Q., et al. 2012, *RAA*, 12, 1197
- Díaz-Hernández, R., Ortiz, E., & Perea, J. 2014, in *Proc. of the 5th Mexican Conf. on Pattern Recognition (Berlin: Springer)*, 314
- Dolphin, A. E. 2000, *PASP*, 112, 1383
- Duan, F., Liu, Y., & Zhang, Y. 2009, in *Proc. of the 2nd Int. Symp. on Information Processing (Piscataway, NJ: IEEE)*, 286
- Fu, H., Liu, P., Meng, S., et al. 2024, *ApJS*, 271, 12
- Glorot, X., & Bengio, Y. 2010, in *Proc. of the 13th Int. Conf. on Artificial Intelligence and Statistics (JMLR Workshop and Conf. Proceedings)*, 249

- He, K., Zhang, X., Ren, S., & Sun, J. 2016, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 770
- Hsiao, T.-Y., Chang, Y.-C., Chou, H.-H., & Chiu, C.-T. 2019, J. Syst. Archit., 95, 9
- Hu, J., Shen, L., & Sun, G. 2018, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 7132
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. 2017, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 4700
- Kuntzer, T., Tewes, M., & Courbin, F. 2016, A&A, 591, A54
- Li, X.-R., Lin, Y.-T., & Qiu, K.-B. 2019, RAA, 19, 111
- Lin, S.-C., Su, Y., Liang, G., et al. 2022, MNRAS, 512, 3885
- Lin, T.-Y., Dollár, P., Girshick, R., et al. 2017, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 2117
- Loshchilov, I. 2017, arXiv:1711.05101
- Lupton, R., Blanton, M. R., Fekete, G., et al. 2004, PASP, 116, 133
- Mehta, T., Bhuta, N., & Shinde, S. 2022, in Int. Conf. on Industry 4.0 Technology (I4Tech) (Piscataway, NJ: IEEE), 1
- Mink, D. J. 1997, in ASPC Conf. Series, 125, 249
- Misra, D. 2019, arXiv:1908.08681
- Morgan, W. W., & Keenan, P. C. 1973, ARA&A, 11, 29
- Nair, V., & Hinton, G. E. 2010, in Proc. of the 27th Int. Conf. on Machine Learning (ICML-10) (Madison, WI: Omnipress), 807
- Qi, X., Wei, Y., Mei, X., et al. 2023, in Int. Conf. on Neural Information Processing (Berlin: Springer), 528
- Qi, Z. 2022, in SHS Web of Conf. 144, 03006
- Rózański, T., Niemczura, E., Lemiesz, J., Posilek, N., & Rózański, P. 2022, A&A, 659, A199
- Saha, M. N. 1921, RSPSA, 99, 135
- Sharma, K., Kembhavi, A., Kembhavi, A., et al. 2020, MNRAS, 491, 2280
- Shi, J.-H., Qiu, B., Luo, A.-L., et al. 2023, MNRAS, 520, 2269
- Simonyan, K., & Zisserman, A. 2014, arXiv:1409.1556
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, JMLR, 15, 1929

Stringer, K. M., Long, J., Macri, L., et al. 2019, AJ, 158, 16

Tan, L., Liu, Z., Wang, X., et al. 2024, ApJS, 273, 34

Van der Maaten, L., & Hinton, G. 2008, JMLR, 9, 2579

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. 2018, in Proc. of the European Conf. on Computer Vision (Cham: Springer), 3

Zeiler, M. D., & Fergus, R. 2014, in Computer Vision-ECCV 8689 (Berlin: Springer), 818

Zhong, J., Lépine, S., Hou, J., et al. 2015, AJ, 150, 42

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv –Machine translation. Verify with original.