Research in Astronomy and Astrophysics, 25:075013 (16pp), 2025 July

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 Liu[1], Hao Fu[1], Shuai Meng[1], Cheng Zeng[1], Yangyang Guo[1], Xue Mei[1], and Shengqing Yao[2]
[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. 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.

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

1. Introduction

With the continuous development and rapid expansion of large astronomical observatories, astronomers are now able to access unprecedented volumes of astronomical data. Among these, 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 data acquisition is relatively limited. In contrast, photometric data, obtained through observations across different bands, record a star's radiation intensity at various frequencies. While they are 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 are predominantly based on the Morgan-Keenan (MK) classification system (Morgan & Keenan 1973). This system 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 properties of the object (Duan et al. 2009). Template-matching has been widely applied in astronomy. Stringer et al. (2019) used template fitting and random forest classification to identify RR Lyrae variable stars in the multi-band sparse sampling data from the Dark Energy Survey. In the field of stellar classification, Zhong et al. (2015) extracted spectral data from LAMOST (Cui et al. 2012) and applied a template-matching method 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 classification 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, thus gradually revealing their shortcomings in practical applications.

In contrast to template-matching methods that rely on predefined template libraries, machine learning algorithms can automatically learn features from data, allowing them to adapt more flexibly to various types of data and classification tasks (Kuntzer et al. 2016). Traditional machine learning algorithms, such as Support Vector Machines (SVM), decision trees, and random forests, are effective in extracting key information from both spectral and photometric data, achieving efficient and accurate classification (Qi 2022). This ability to automatically extract features allows machine learning methods to exhibit stronger adaptability when handling high-dimensional and complex astronomical data, eliminating dependence on the quality and quantity of the template library. As a result, 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 the advancement of 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 the limitations of traditional machine learning methods in handling large-scale and complex astronomical data, deep learning techniques have gradually gained widespread application in stellar classification (Różański 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 data sets (Sharma et al. 2020). In stellar classification tasks, deep learning methods have demonstrated significant advantages, whether applied to spectral (Fu et al. 2024) or photometric (Shi et al. 2023) data.

In this study, we selected photometric data of O, B, A, F, G, K, and M-type stars from the Sloan Digital Sky Survey (SDSS) Data Release 18 (DR18). After preprocessing steps such as localization and mapping, we generated five-channel Numpy files to replace the commonly used RGB images as the data set. 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 the weight allocation of different photometric bands. Finally, through a series of experiments and comparative methods, we explored the importance of each photometric band in stellar classification and validated the effectiveness of SPFFN in classification tasks.

The structure of this paper is as follows: Section 2 introduces the preprocessing methods for photometric data and the SPFFN model architecture; Section 3 describes the data set source, parameter optimization of the DES-SE module, the impact of photometric bands, and the model training strategy; Section 4 presents comparative experiments with various classification algorithms; and Section 5 provides a summary of the study and future outlook.

2. Methods

2.1. Mapping of Photometric Data and Numpy Construction

After obtaining the 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 are in the same reference frame. To achieve this, we employ the World Coordinate System (WCS)-based image reprojecting method (Mink 1997). 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, the 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 WCS header file of the r band. The specific transformation process of WCS is primarily divided into two parts. Let the pixel position in the image be (x, y), and the goal is to convert it into celestial coordinates (α,δ), where α is the R.A. and δ is the decl. The transformation can be expressed as Equation (1).

$$\begin{bmatrix}\alpha\\ \delta\end{bmatrix} = T \cdot \begin{bmatrix}x\\ y\end{bmatrix} + \boldsymbol{b}, \tag{1}$$

where T represents the matrix that includes transformations such as rotation, scaling, and translation, 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).

$$\begin{bmatrix}x'\\ y'\end{bmatrix} = T^{-1} \cdot \left(\begin{bmatrix}\alpha\\ \delta\end{bmatrix} - \boldsymbol{b}'\right), \tag{2}$$

where T^{-1} represents the inverse of the transformation matrix, and b' is the translation vector.

To improve alignment accuracy, we employed an interpolation algorithm during the image reprojecting process. The nearest-neighbor interpolation method was used to compute an appropriate pixel value 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.

Figure 1. Complete star map after preprocessing.

Figure 1 shows a complete star map after alignment, with the *irg* bands mapped to an RGB image.

After the 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 in the image. Figure 2 displays the RGB image generated using the *irg* bands, and Figure 3 depicts the RGB image generated using the *urz* bands. This method enhances the 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 the values to an appropriate brightness range.

However, this traditional method has several limitations. First, the key information contained in the two discarded bands is not effectively utilized, which may result in a loss of image information. This is particularly problematic in multi-band data sets, where some astronomical features may only appear prominently in the bands that are excluded from the RGB mapping (Lin et al. 2022). Second, when normalizing the 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 the characteristic differences among all five bands (Lupton et al. 2004).

To overcome the limitations of the traditional three-channel RGB image generation method, this study proposes a five-channel Numpy processing approach to fully utilize the five bands (u, g, r, i, z) in the SDSS data. Specifically, we have revised the processing method to handle image data from all five bands, performing unified brightness mapping and color adjustments. Unlike the traditional method, which maps only three bands, our approach simultaneously processes data from all five bands, ensuring that each band is fully utilized during the synthesis process. As a result, the five-channel images generated 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 directly used 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 that all band data undergo consistent processing. The specific process is as follows:

(1) *Brightness mapping*. To enhance the visibility and contrast of the image, a brightness mapping function is first applied. The core idea of brightness mapping is to transform the pixel values of each band image using a function similar to arcsinh. For each band, the brightness representation is given by Equation (3).

$$L_{\mathrm{band}}(x, y) = \mathrm{arcsinh}\left(\frac{I_{\mathrm{band}}(x, y)}{S_{\mathrm{band}}}\right), \tag{3}$$

where $I_{\mathrm{band}}(x, y)$ 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 for each band, normalization is performed to ensure that the brightness values of all bands lie within the same range. Specifically, the pixel values of all bands are standardized in the range [0, 1], as shown in Equation (4).

$$I'_{\mathrm{band}}(x, y) = \frac{L_{\mathrm{band}}(x, y) - \min(L_{\mathrm{band}})}{\max(L_{\mathrm{band}}) - \min(L_{\mathrm{band}})}, \tag{4}$$

where $\min(L_{\mathrm{band}})$ and $\max(L_{\mathrm{band}})$ 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.

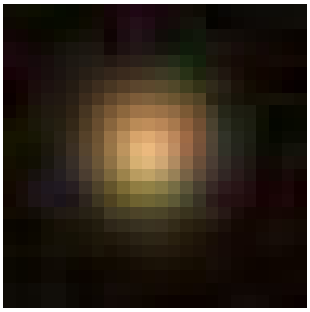

Figure 2. Star image generated in *irg* bands.

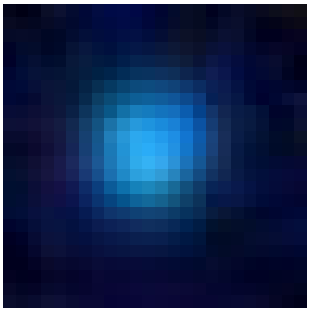

Figure 3. Star image generated in *urz* bands.

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 the stars from the SDSS database, we can accurately locate the 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 are within the 16×16 pixel range. To ensure the accuracy of the data, we apply dilation and erosion operations to the image, removing noise and refining the edges to clearly define the target star. Specifically, erosion and dilation are applied to each channel image to eliminate surrounding noise and enhance the boundaries of the target stars. After processing, we still choose to 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, 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 the target stars, and some bands may be affected by noise or redundant information, we propose a dynamic channel weighting mechanism to adjust the contributions of each band. This mechanism enhances the model's focus on key bands while suppressing the influence of interfering information on the classification results. This mechanism, specifically designed for stellar classification tasks, is termed DES-SE. As illustrated in Figure 5, its workflow is as follows:

(1) *Global Average Pooling*. The input five-channel image undergoes global average pooling (GAP) across the spatial dimensions to obtain a global feature vector for each channel. The specific process is shown in Equation (5).

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{c,i,j}, \tag{5}$$

where c represents the channel index, and H and W and represent the height and width of the image, respectively. $x_{c,i,j}$ 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 a nonlinear mapping through the LeakyReLU activation function. The specific process is defined in Equation (6).

$$s = \text{LeakyReLU}(W_1 z), \tag{6}$$

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 the weight distribution is generated through a 1×1 dynamic convolution. The specific process is expressed in Equation (7).

$$w = \sigma(\text{Conv}_{1\times1}(s)), \tag{7}$$

where σ represents the sigmoid activation function, ensuring that the weights generated are within the range [0, 1].

(4) *Weight Application*. The generated dynamic weights are applied to the original input features with channel-wise weighting. The specific process is written in Equation (8).

$$\hat{x}_c(h, w) = w_c \cdot x_c(h, w), \tag{8}$$

where (h, w) represents 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 the dynamic modeling capability between channels by incorporating the LeakyReLU activation function (Glorot & Bengio 2010) and dynamic convolution (Chen et al. 2020). This allows for more efficient adaptation to the complexity and characteristics of the five-band photometric

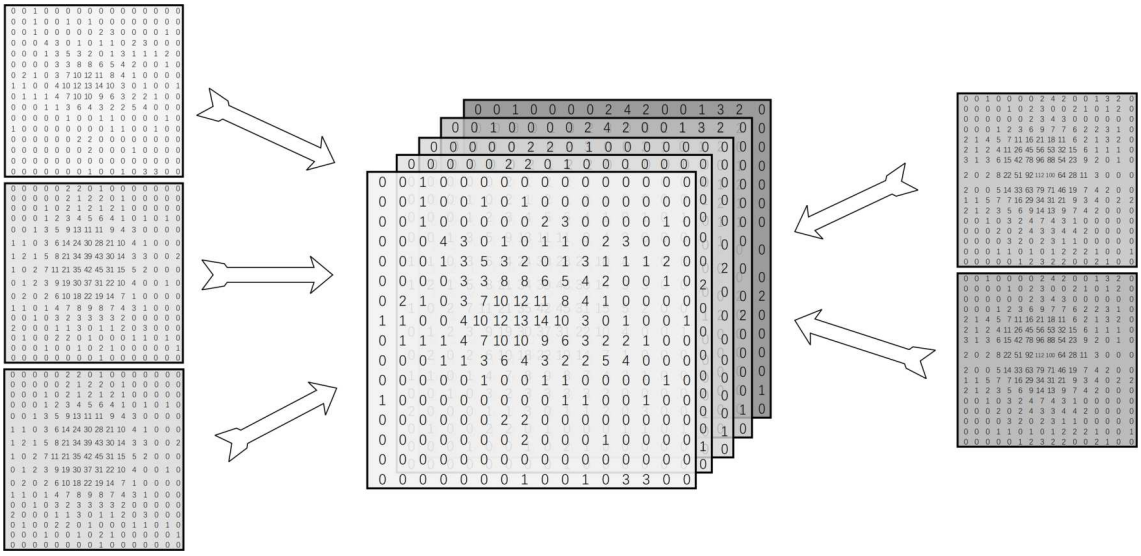

Figure 4. Numpy file generated from five bands.

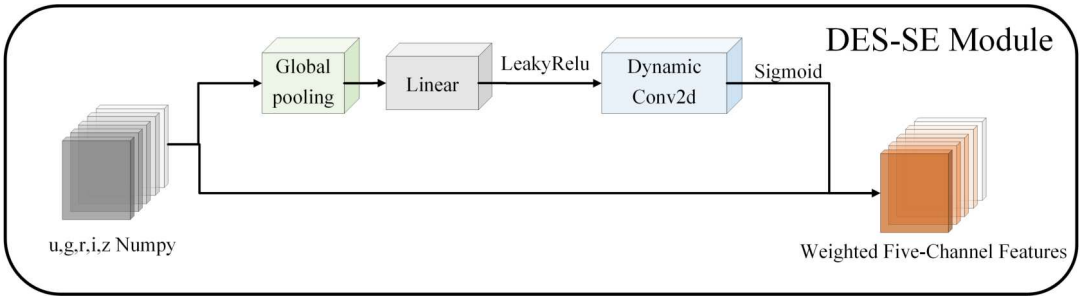

Figure 5. Structure diagram of DES-SE module.

data. The use of 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 the limitations of traditional modules in channel weight generation. It not only improves the feature discrimination ability for five-band data but also more efficiently models the complementary relationships between bands. The dynamic modeling capability plays a crucial role in enhancing the overall network performance, with a more rational distribution of channel weights. This enables the network input to optimize feature representation, thereby 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 the complex features embedded in the data. This is particularly advantageous in 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, the stellar photometric data were processed through spatial alignment and brightness mapping, resulting in a $64 \times 64 \times 5$ Numpy file representing the stellar photometric information. This data format preserves the luminosity distribution features of the star across the *ugriz* five bands.

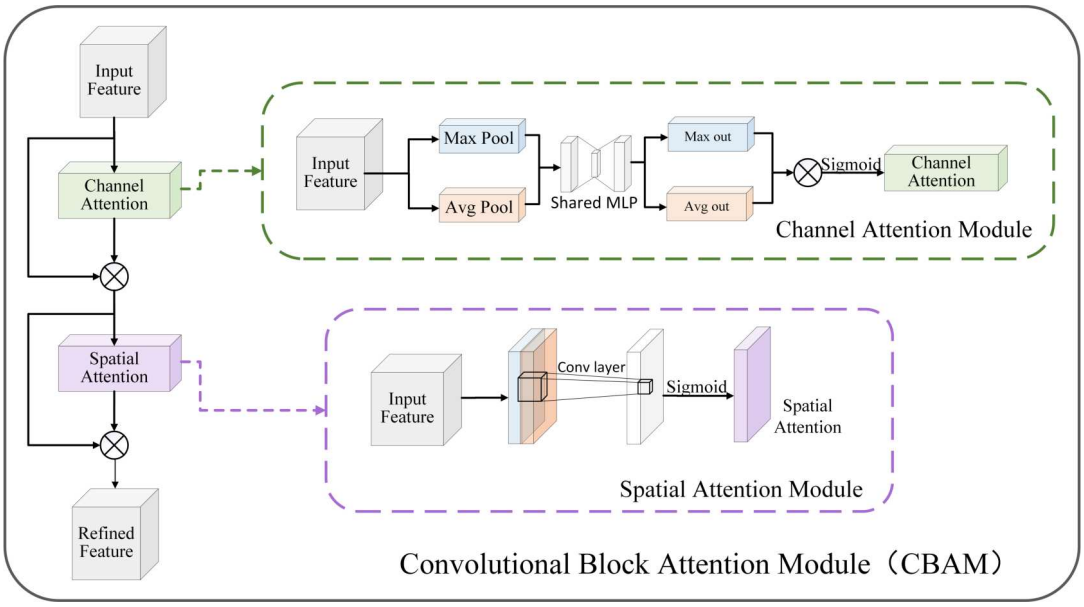

Figure 6. Structure diagram of CBAM.

The resolution of the photometric information is relatively low, and it exhibits clear locality and band-specific differences. Based on the data characteristics and the classification advantages of neural networks, this study proposes the SPFFN, designed to efficiently extract multi-scale features from the 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 the 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 the 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).

$$F_{\text{conv}} = W * X + b, \tag{9}$$

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, CBAM combines channel attention and spatial attention, enabling the model to adaptively highlight the 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).

$$M_{\text{channel}} = \sigma(W_1\delta(W_0(\text{GAP}(F)) + W_1\delta(W_0(\text{GMP}(F))))), \tag{10}$$

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 the weights based on the importance of photometric information from different bands. Spatial attention, on the other hand, focuses on the significant regions of the feature map through convolutional operations, with the weight represented by Equation (11).

$$M_{\text{spatial}} = \sigma(\text{Conv}(\text{Concat}(\text{GAP}(F), \text{GMP}(F)))), \tag{11}$$

where σ is the Sigmoid function, and Concat represents the channel concatenation operation.

Finally, CBAM enhances the attention-weighted feature map through element-wise multiplication, as shown in Equation (12).

$$F_{\text{CBAM}} = F \odot M_{\text{channel}} \odot M_{\text{spatial}}, \tag{12}$$

where $\odot$ represents the element-wise multiplication operation, and F_{CBAM} is the feature map after being weighted by CBAM.

Through the CBAM module, the SPFFN Block can adaptively capture complementary information between different bands of stellar photometric data and focus on the significant regions of the spatial distribution. This design effectively enhances the network's ability to express key photometric features, further improving the accuracy of the stellar classification task.

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: the Depthwise convolution, which extracts features within each channel, and the Pointwise convolution, which aggregates features across channels (Chollet 2017). The computational complexity of this process can be represented as Equation (13).

$$\text{FLOPs}_{\text{separable}} = H \times W \times C_{\text{in}} \times K^2 + H \times W \times C_{\text{in}} \times C_{\text{out}}, \tag{13}$$

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

In fact, this design not only reduces the number of parameters and computational complexity but, more importantly, it 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 the learning process. Traditional convolution processes both intra-channel and inter-channel features together, which may introduce unnecessary complexity, thereby reducing the model's generalization ability. In contrast, depthwise 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 its classification performance.

In the input data, the stellar photometric maps are represented as five-channel Numpy files, where the photometric distribution in different bands exhibits significant spatial features. The central region of the data has higher photometric values, showing a "spherical" aggregation pattern, while the surrounding area approaches zero, representing the dark cosmic background. This spatial distribution pattern makes the stellar features quite distinct at different scales. For instance, the high-intensity core of the star contains densely packed information, while the outer region, though the 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).

$$P_l = C_l + \text{Upsample}(P_{l+1}), \tag{14}$$

where C_1 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 the high-luminosity concentrated features at the center of the star, which aids in distinguishing intensity differences across various bands. On the other hand, low-level features preserve rich spatial details, such as the luminosity decay and contours of the star's edges, especially in the outer regions where the luminosity gradually diminishes. The FPN effectively prevents the loss of these details. The progressive upsampling and lateral fusion mechanism in FPN ensure the complementarity of features at different scales, allowing the network to capture both the luminosity concentration features and the edge attenuation information of the star, even at limited input resolutions. For stellar images in the five-channel Numpy files, FPN enables the network to more comprehensively understand the 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).

$$\boldsymbol{y} = W_3\sigma(W_2\sigma(W_1\boldsymbol{x})), \tag{15}$$

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, 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,

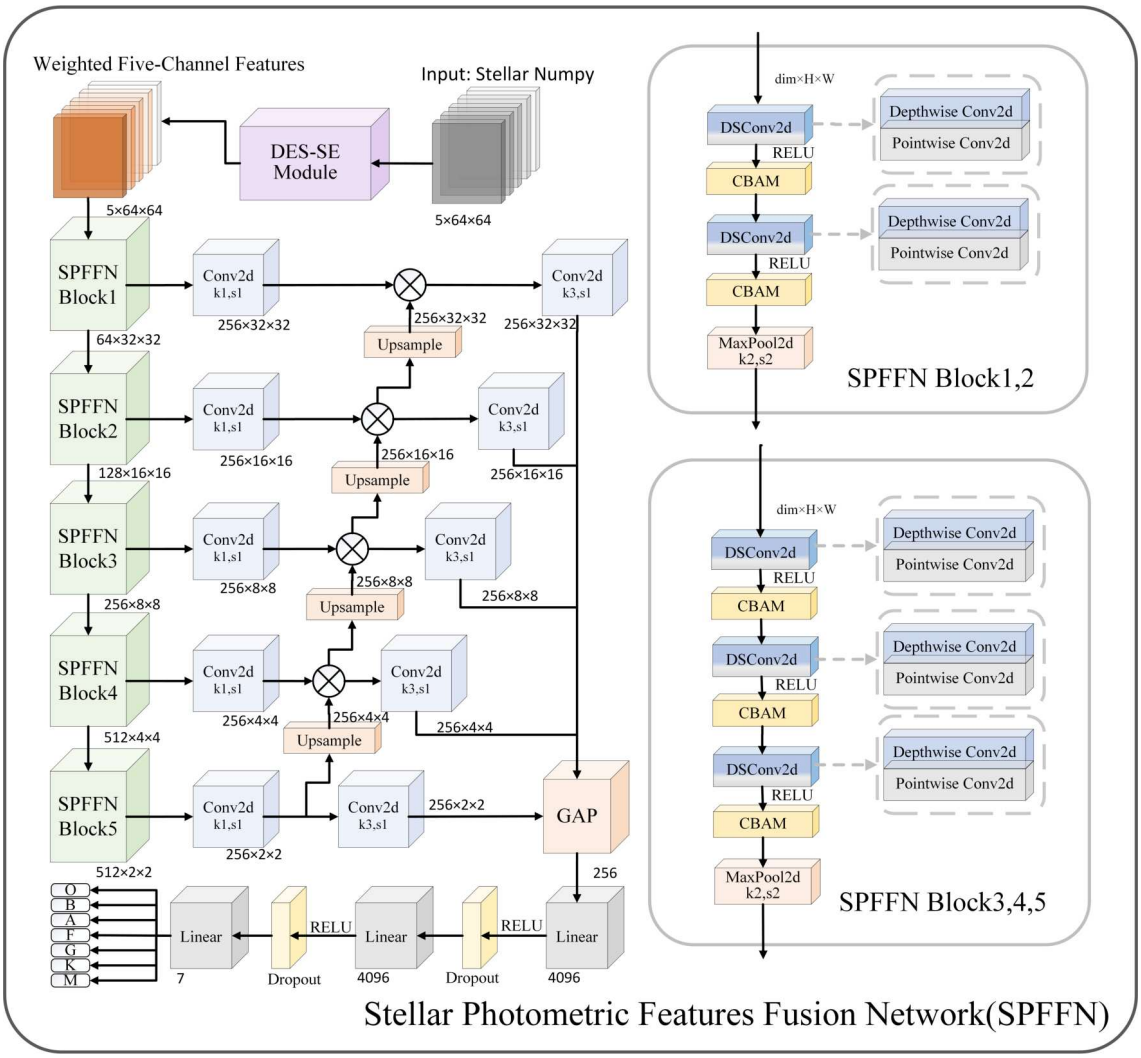

Figure 7. Network architecture diagram of SPFFN.

and FPN for multi-scale feature fusion. This architecture achieves efficient feature extraction and information integration. By enhancing the focus on photometric regions of significance, the network effectively captures both local and global features, improving the accuracy and robustness of stellar classification tasks while maintaining a low computational complexity. This provides an efficient and accurate solution for the classification of stellar photometric data. The SPFFN network architecture involved in this study has been uploaded to GitHub. The link is as follows: https://github.com/Fuhao-Edwin/SPFFN.git.

3. Experiment and Analysis

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 a 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, which has been 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 the 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 the 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., the background does not exceed 80%). This dynamic window-sizing strategy helps ensure that 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 the actual data acquisition process, several factors including catalog limitations, band incompleteness, and extraction failures led to a substantial reduction in the final data set 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 also inevitably excluded a large portion of the 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 of which had complete *ugriz* coverage, fell within the defined brightness range, and passed the cropping criteria, were used to construct the final data set. For each spectral type, several thousand five-band stellar samples were retained. To minimize selection bias, the final samples were randomly selected from the pool of valid candidates within each class.

Based on the 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 data set used in this study. To support reproducibility and facilitate future research, the full data set has been made publicly available on the Zenodo research data repository: 10.5281/zenodo.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 by optimizing the parameters. Based on the optimal parameter configuration, we further investigated the importance of each band in the classification task and validated this hypothesis through a series of experiments.

In the DES-SE module, there are differences and redundant noise in the photometric distributions of different bands, making 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.

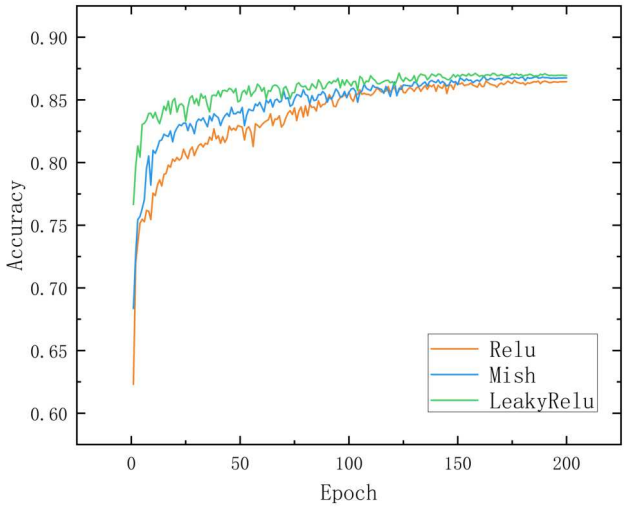

Figure 8. Accuracy curve comparison for different activation functions.

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 presents the accuracy variation curves during model training for these three activation functions. The experimental results indicate that ReLU, due to its property of completely discarding gradients in the negative region, showed significant limitations when handling complex data distributions. This was particularly evident in its inability to capture features in regions with weaker photometry, leading to overall performance inferior to the other two activation functions. Mish, as a more complex nonlinear activation function, was able to enhance feature representation in certain cases but still lagged behind LeakyReLU in terms of training convergence speed and final accuracy, while also 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. This enhanced the model's ability to capture complementary information across bands. Leveraging this advantage, LeakyReLU demonstrated faster convergence early in the training process and ultimately achieved the highest classification accuracy.

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 that 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, which is 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.

Table 1
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%

Furthermore, considering that the negative slope parameter directly affects the gradient flow in the negative region of LeakyReLU, which in turn 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 Table 1.

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 the weak light bands, while preventing excessive loss of feature information. As the negative slope increases to 0.01, the 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 drop in classification accuracy to 86.04%. This phenomenon may be due to an excessively strong negative slope, which leads the model to overemphasize low-light features while suppressing the effective learning of key light information.

In summary, the experimental results demonstrate that the choice of activation function and the reduction parameter has a significant impact on the performance of the DES-SE module. Ultimately, we selected LeakyReLU as the activation function, set the negative slope parameter to 0.001, and set the 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 the photometric data for the stellar classification task. By saving the dynamic weight distributions after multiple trials, we observed that the weight parameters varied across experiments, mainly due to the 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 the classification performance.

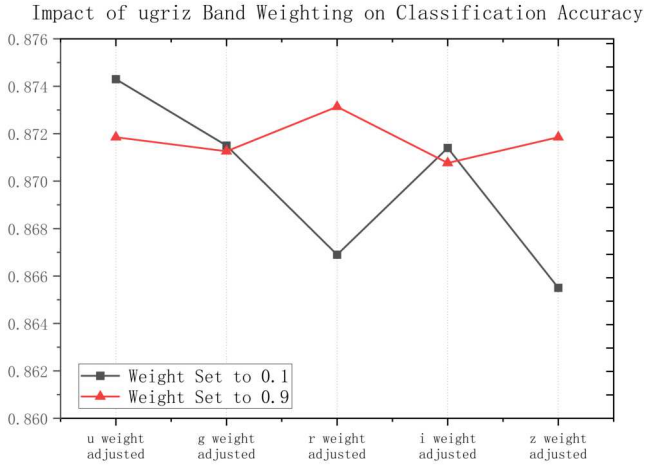

Figure 9. Impact of band weight changes on the model.

To validate this observation, we designed further experiments. For the weight parameters, values closer to 0 indicate less importance, while values closer to 1 indicate greater importance. In the previous DES-SE module, the 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 of the *ugriz* bands to either 0.1 or 0.9, while keeping the weights of the remaining bands at 0.5, and observed the change in the model's classification performance.

The experimental results are shown in Figure 9. When the weights of the *r* and *z* bands were fixed at 0.9, the classification accuracy improved, whereas when the weights were fixed at 0.1, the accuracy significantly dropped. This further confirmed the importance of these two bands in the classification task. In contrast, for the *u* band, when its weight was set to 0.9, the classification accuracy decreased, but when the weight was set to 0.1, the accuracy improved. This suggests that the *u* band may contain a significant amount of redundant or noisy information. In comparison, adjusting the weights of the *g* and *i* bands had little effect on the overall performance of the model.

These experimental results validate the crucial role of the *r* and *z* bands in stellar classification tasks, while also revealing that the *u* band has lower importance and may even negatively impact model performance. This provides valuable insights for optimizing the use of photometric data 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, with the Ubuntu 20.04 operating system. 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, and attention mechanism modules. The final model has a total of 4.32 million parameters and a total of 2.5 billion floating-point operations.

Table 2
Distribution of Training and Validation Samples for Each Stellar Type

Stellar Type	Training Set	Validation Set	Total	Training/Validation Ratio
O	1044	261	1305	8:2
B	2907	727	3634	8:2
A	3940	986	4926	8:2
F	4167	1042	5209	8:2
G	3160	790	3950	8:2
K	4690	1173	5863	8:2
M	6689	1672	8361	8:2

In terms of experimental data set processing, the photometric data of the seven stellar categories were all standardized. The final data set contains a total of 33,248 five-channel stellar photometric Numpy files. Following common practice in neural network training, 80% of the samples are randomly selected as the training set, while the remaining 20% are used as the validation set for each training run. The detailed class distribution is shown in Table 2.

During the training phase, we first resized all five-channel stellar photometric Numpy files to 64×64, ensuring a uniform input size for the neural network. Then, the images were normalized using the mean values [0.485, 0.456, 0.406, 0.485, 0.456] and standard deviations [0.229, 0.224, 0.225, 0.229, 0.224], which were derived from the statistical properties of RGB images in the widely used ImageNet data set (Deng et al. 2009). Since the input data in this study consist of five-channel photometric images, which differ from the RGB images in ImageNet, the statistical properties were accordingly adjusted. This normalization configuration effectively standardized the input five-channel Numpy data, improving the stability of the model's training process and enhancing the performance of transfer learning.

In terms of network parameter settings, for the DES-SE module, the reduction parameter was set to 4 to balance computational efficiency and model performance. For the activation function in the first fully connected layer, LeakyReLU was chosen, with its negative slope set to 0.001, which helps mitigate the gradient vanishing problem that may occur with ReLU and improves training stability. Next, in the SPFFN model, the feature outputs of the five feature extraction modules (SPFFN Blocks) were set to [64,12, 256, 512, 512]. Each block's depthwise separable convolution layer used a 3×3 kernel with a padding of 1 to maintain the size of the feature maps. In the CBAM module, the reduction parameter for channel attention was set to 16, which helps effectively compress the channel dimension and enhance the expression of

Table 3
Classification Results of SPFFN

Class	True Positives	True Negatives	False Positives	False Negatives	Precision	Recall	F1-Score	Accuracy
O	180	6323	63	81	0.7407	0.6896	0.7142	⋯
B	563	5805	116	163	0.8291	0.7754	0.8014	⋯
A	795	5493	169	190	0.8246	0.8071	0.8158	⋯
F	919	5398	208	122	0.8154	0.8828	0.8477	⋯
G	604	5739	118	186	0.8365	0.7645	0.7989	⋯
K	1107	5337	138	65	0.8891	0.9445	0.9160	⋯
M	1646	4954	21	26	0.9874	0.9844	0.9859	⋯
Average	⋯	⋯	⋯	⋯	0.8461	0.8355	0.8400	0.8747

important features. Finally, during the feature output stage, the dropout rates for the two Dropout layers were set to 0.5 to reduce overfitting and improve the model's generalization ability (Srivastava et al. 1929).

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

For model evaluation, multiple evaluation metrics were used, including accuracy, precision, recall, and F1-score.

Table 3 presents the classification results of SPFFN on the seven-class star luminosity data. The performance metrics across the 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 the F1-scores reach 0.9859 and 0.9160, 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 0.87414, 0.87468, and 0.87371, respectively. These results suggest that the model performs well in the star luminosity classification task and demonstrates a strong ability to handle data from different categories.

To comprehensively demonstrate the model's performance, various visualization methods were employed in the experimental results, including accuracy-loss curves, confusion matrices, and t-SNE dimensionality reduction. The training process of the SPFFN model is shown, with Figure 10 illustrating the changes in training and validation accuracy. This indicates that the model gradually converges during training, with training accuracy steadily increasing and stabilizing. Figure 11 shows the changes in training and validation loss, where the training loss

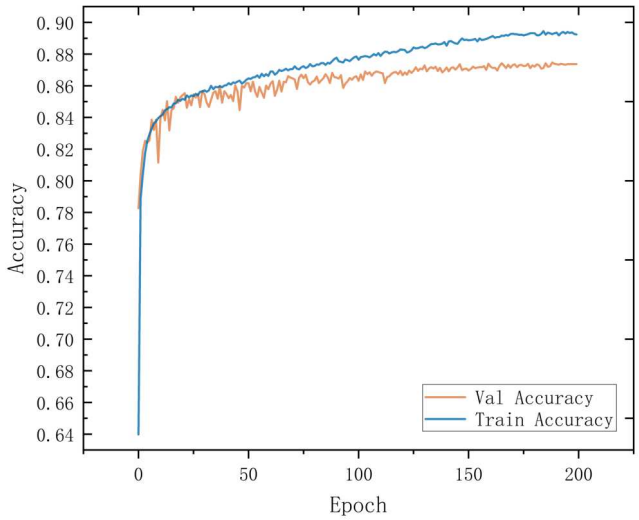

Figure 10. Training and validation accuracy curve of SPFFN for stellar classification.

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

Figure 12 presents the confusion matrix of the SPFFN model's final classification results on the test set. The confusion matrix provides an intuitive view of the model's classification performance across different categories. The values on the diagonal represent correctly classified samples, while the off-diagonal values indicate misclassified samples. Most of the predictions are concentrated along the diagonal, suggesting that the model performs well in classifying the majority of categories. Notably, the 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 of the confusion matrix reflects an ideal classification performance.

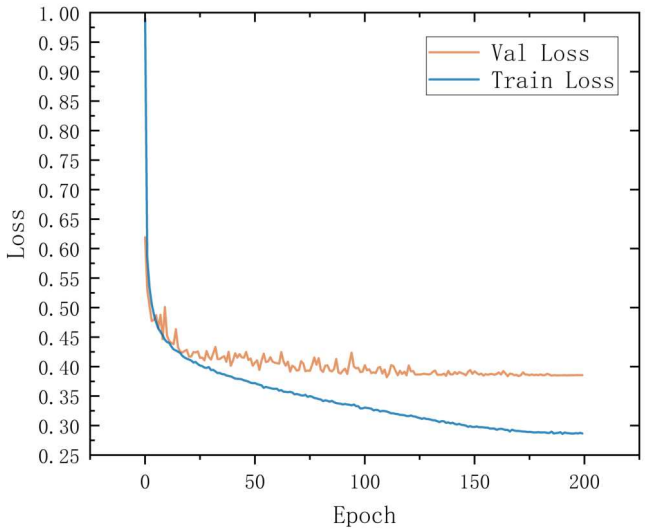

Figure 11. Training and validation loss curve of SPFFN for stellar classification.

As shown in Figure 13, we applied the t-SNE algorithm to perform dimensionality reduction and visualize the feature distribution. t-SNE is capable of mapping high-dimensional features into a two-dimensional space, allowing for easier observation of the distribution among different categories (Van der Maaten & Hinton 2008). The points in the figure, represented by different colors, correspond to the seven star types (categories 0–6). From the visualization, most categories exhibit clear clustering in the feature space; however, some overlap between categories is observed. Overall, the model performs well in classifying the majority of categories, but for certain categories, such as O-type stars, there remains some classification bias, indicating room for improvement in distinguishing these more complex categories.

4. Discussions

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 algorithms 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 the field of 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 the stellar luminosity data classification task, thereby visually verifying its superiority and application potential.

The Random Forest algorithm is an ensemble learning-based classification method, which constructs multiple decision trees and takes a vote on their predicted results to determine the final classification. In this experiment, the input data were uniformly scaled to a 64×64 size, maintaining the five feature channels. The default hyperparameter configuration was used, including 100 trees and an unrestricted tree depth, to ensure the stability of model training and to avoid overfitting.

ResNet50 and DenseNet121 are classic deep learning architectures widely used in image classification tasks. ResNet50 alleviates the vanishing gradient problem by introducing 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 the network's representational capacity. In this experiment, the hyperparameters for ResNet50 and DenseNet121 were set to be consistent with those used in the SPFFN model, including the optimizer, learning rate schedule, and batch size, ensuring a fair comparison between the two 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 the *gri* and *urz* three-band combinations to extract and integrate features across all luminosity bands, achieving excellent performance in the seven-class classification task. In the study by Shi et al. (2023), SCNet achieved an accuracy of 86.1% on this task.

Table 4 presents the comparative experimental results of different models in the stellar photometric image classification task. The SPFFN model performs excellently across all evaluation metrics, achieving an accuracy of 87.47%, average precision of 84.62%, average recall of 83.55%, and average F1-score of 84.01%. In contrast, the machine learning algorithm Random Forest shows a 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 the stellar photometric data classification task and do not exhibit superior accuracy. SFNet and SCNet, which have been recently developed

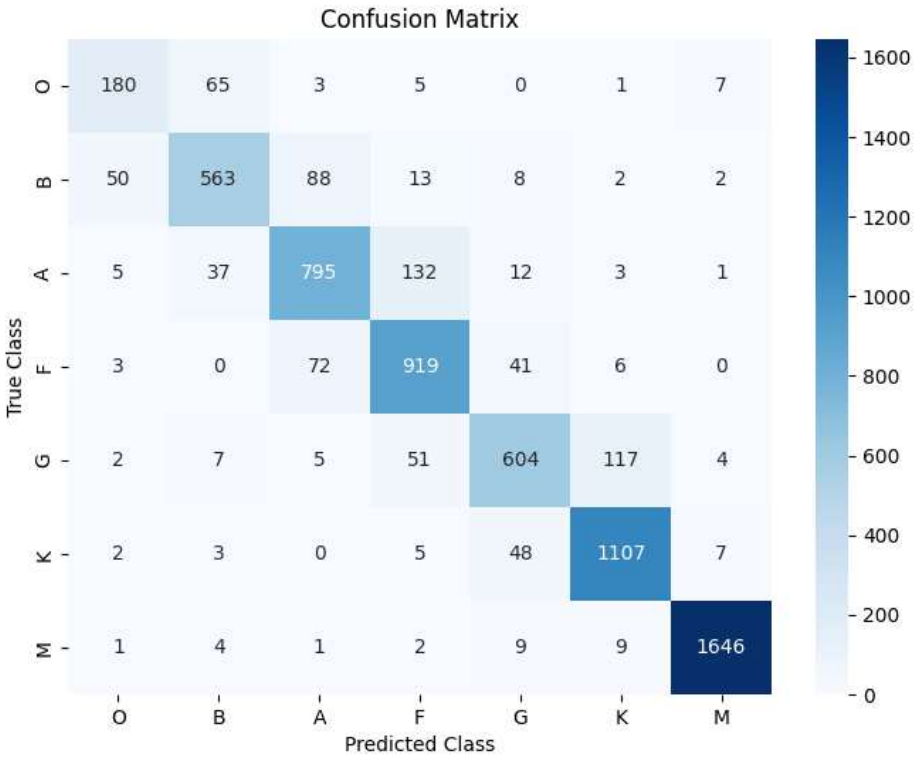

Figure 12. Confusion matrix of SPFFN on the validation set for stellar classification.

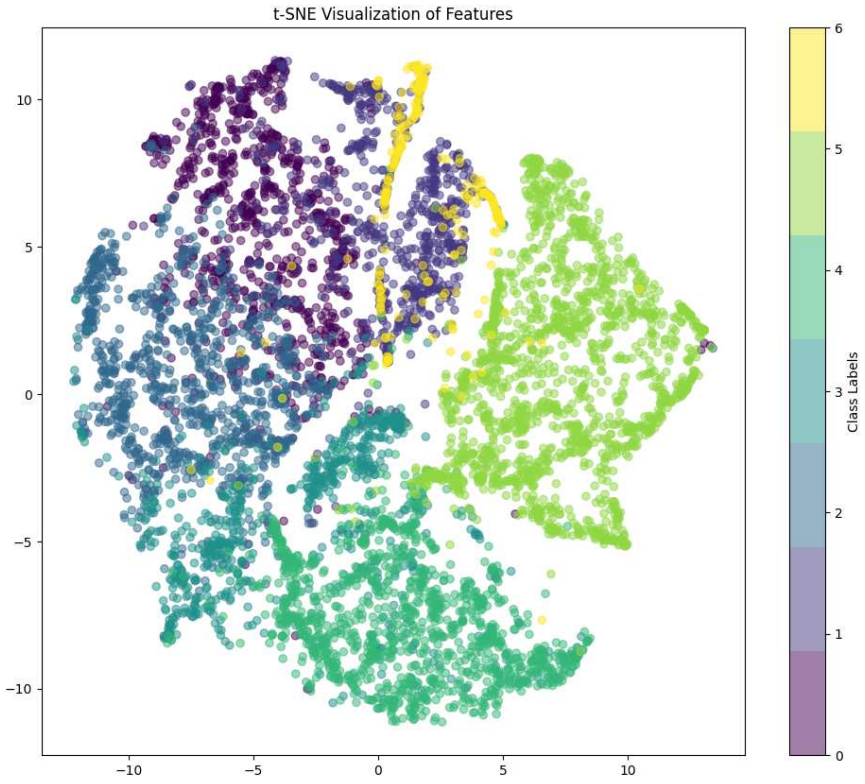

Figure 13. T-SNE visualization of SPFFN classification results on the validation set.

Table 4
Comparison of Classification Metrics for Different Network Models

Model	Accuracy	Average Precision	Average Recall	Average F1-Score
Random Forest	83.62%	80.29%	77.86%	78.71%
ResNet50	82.93%	79.46%	77.68%	78.34%
DenseNet121	83.84%	80.56%	78.97%	79.54%
SFNet	84.62%	81.52%	79.65%	80.35%
SCNet	86.10%	85.21%	83.19%	···
SPFFN	87.47%	84.62%	83.55%	84.01%

specifically for stellar classification in the astronomical field, 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 the compared models, confirming its effectiveness and potential for stellar classification tasks.

Apart from the comparative performance of SPFFN and other classifiers, it is also important to discuss how photometric classification relates to conventional spectral classification methods.

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 the 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 terms of 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 stellar 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 data sets.

5. Conclusion

In this study, our goal is to classify stellar photometric data using deep learning techniques. First, 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 as the data set. Additionally, the DES-SE module was utilized to optimize the weight distribution of different bands in the classification task, and the impact of features from various bands on classification performance was analyzed. Finally, we proposed the SPFFN model, a feature fusion network specifically designed for large-scale photometric data sets. Through comparative experiments with five classifiers (Random Forest, ResNet50, DenseNet121, SFNet, and SCNet), we validated the significant advantages of SPFFN in stellar classification tasks.

In summary, this study demonstrates the feasibility and effectiveness of photometric data in stellar classification and presents a high-accuracy classification scheme suitable for large-scale photometric data sets, offering an effective solution for future stellar classification research. Moving forward, we plan to further enhance the model's classification accuracy and explore the integration of spectral and photometric data, employing multimodal learning approaches to provide new research directions in the field of stellar classification.

ORCID iDs

Hao Fu https://orcid.org/0009-0000-2184-5035

References

Abd-elaziem, Ayman H., & Soliman, Tamer H. M. 2023, IJAACI, 3, 29
Almeida, A., Anderson, S. F., Argudo-Fernández, M., et al. 2023, ApJS, 267, 44
Bertin, E. 2011, in ASP Conference Proceedings (San Francisco, CA: ASP), 435
Breiman, L. 2001, Machine Learning, 45, 5
Cai, J., Yan, Z., Yang, H., et al. 2024, A&A, 687, A15
Chen, Y., Dai, X., Liu, M., et al. 2020, in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (Piscataway, NJ: IEEE), 11030
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
Deng, J., Dong, W., Socher, R., et al. 2009, in IEEE Conf. on Computer Vision and Pattern Recognition, IEEE (Piscataway, NJ: IEEE), 248
Díaz-Hernández, R., Peregrina-Barreto, H., Altamirano-Robles, L., et al. 2014, ExA, 38, 193
Dolphin, A. E. 2000, PASP, 112, 1383
Duan, F.-Q., Liu, R., Guo, P., et al. 2009, RAA, 9, 341
Fu, H., Liu, P., Qi, X., et al. 2024, RAA, 24, 095023
Glorot, X., & Bengio, Y. 2010, in Proc. Thirteenth Int. Conf. on Artificial Intelligence and Statistics, JMLR Workshop and Conf. Proc., 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. Sys. Arc., 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), 1 (Piscataway, NJ: IEEE), 1
Mink, D. J. 1997, in ASPC Conf. Series, 125 (San Francisco, CA: ASP), 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, 1822 (Berlin: Springer), 528
Qi, Z. 2022, in SHS Web of Conf. 144, EDP Sciences, 03006
Różański, T., Niemczura, E., Lemiesz, J., Posiłek, N., & Różański, 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. 1929, JMLR, 15, 2014
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 *(Zurich, Switzerland)* (Berlin: Springer), 818
Zhong, J., Lépine, S., Hou, J., et al. 2015, AJ, 150, 42