Application of Deep Learning in Analysis of Stellar Spectra

A dissertation submitted in
partial fulfilment for the degree of

Master of Technology

in

Computer Science

by

Piyush Yayati
Roll no. - **CS2312**

under the supervision of

Dr. Sarbani Palit
Computer Vision and Pattern Recognition Unit (CVPRU)

Indian Statistical Institute, Kolkata
June, 2025

CERTIFICATE

This is to certify that the dissertation entitled "**Application of Deep Learning in Analysis of Stellar Spectra**" submitted by **Piyush Yayati** to the Indian Statistical Institute, Kolkata, in partial fulfillment of the requirements for the degree of Master of Technology in Computer Science, is an authentic and genuine record of the research work carried out by the candidate under my supervision and guidance. I affirm that the dissertation has met all the necessary requirements in accordance with the regulations of this institute.

Sarbani Palit 11-6-2025

Dr. Sarbani Palit
CVPR Unit
Indian Statistical Institute
Kolkata - 700108
India

Acknowledgement

I would like to express my heartfelt thanks to Dr. Sarbani Palit, my advisor at the Computer Vision and Pattern Recognition Unit of the Indian Statistical Institute, Kolkata, for her expert guidance, steadfast encouragement, and inspiration throughout my research journey. Her deep expertise and innovative input have greatly enriched my understanding and shaped my approach to rigorous research.

I am also sincerely grateful to Ankita Sarkar, Senior Research Fellow at the Indian Statistical Institute, for her essential help in collecting and preparing the datasets required for this work. Her ongoing suggestions and encouragement have played a vital role in the project's success.

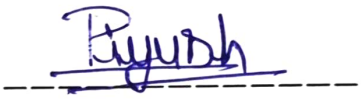
My gratitude also goes out to all the faculty members at the Indian Statistical Institute for their wise counsel, insightful feedback, and valuable teaching, which have offered critical perspectives to my research.

Lastly, I wish to thank my parents and family for their endless support. I am also thankful to all my friends for their continuous assistance and motivation. My appreciation extends to everyone who has contributed to my development and achievements, even if I have not mentioned them by name here.

Declaration

I, **Piyush Yayati**, with Roll No. **CS2312**, hereby declare that the material pre- sented in the dissertation titled **Application of Deep Learning in Analysis of Stellar Spectra** represents original work carried out by me for the degree of **Master of Technology in Computer Science** at the **Indian Statistical Institute, Kolkata**.

Furthermore, I affirm that no sections of this report have been sourced or copied from external references without proper attribution. I am aware that any instances of plagiarism or the use of unacknowledged materials from third parties will be treated with the utmost seriousness and consequences.



Piyush Yayati
M.Tech (CS), CS2312
Indian Statistical Institute

Abstract

In recent years, the analysis of high-resolution stellar spectra has become increasingly important for estimating key stellar parameters such as effective temperature ($T_{\rm eff}$), surface gravity ($\log g$), metallicity ([M/H]), and rotational velocity ($v \sin i$). Traditional methods often rely on manual calibration or spectrum synthesis, which can be time-consuming and error-prone, especially for M dwarfs whose spectra are dense with molecular features. In this study, we investigate the use of convolutional neural networks (CNNs) to automate the estimation of stellar parameters using synthetic and observed data.We adopt a StarNet-like CNN architecture trained on synthetic spectra generated from the PHOENIX-ACES model grid, and evaluate its performance on real observations from the CARMENES survey. Separate models are developed for each parameter, allowing for dedicated tuning and improved prediction accuracy. The training process includes flux normalization and data augmentation across multiple spectral windows. To assess performance, we compare predicted parameters with literature values and find strong agreement, especially for $T_{\rm eff}$ and $\log g$, with significantly reduced mean squared error.This approach demonstrates the effectiveness of deep learning in spectroscopic analysis, offering a scalable solution for stellar parameter estimation. The results reinforce the potential of CNNs to support large-scale stellar surveys and contribute to more accurate stellar characterization.

Keywords: Stellar spectra, CARMENES, PHOENIX-ACES, convolutional neural network (CNN), deep learning, stellar parameter estimation, StarNet, $T_{\rm eff}$, $\log g$, [M/H], $v \sin i$

Motivation

Accurately determining stellar atmospheric parameters—such as effective temperature ($T_{\rm eff}$), surface gravity ($\log g$), and metallicity ([Fe/H])—is fundamental to many areas of astrophysics. These parameters play a critical role in characterizing stellar populations, tracing galactic evolution, and studying potential exoplanet hosts. Traditionally, these values have been derived through physical model fitting and synthetic spectral analysis. While reliable, such techniques are often computationally expensive and may struggle with noisy observations or spectra that diverge from theoretical expectations.

In recent years, the availability of large and diverse spectroscopic datasets from surveys like APOGEE and CARMENES has opened the door to data-driven approaches. Among these, deep learning—particularly convolutional neural networks (CNNs)—has shown considerable promise. CNNs are well-suited for learning complex, hierarchical patterns in spectral data and can generalize well even when faced with subtle variations or noise in the input.

The StarNet model serves as a notable example of this potential. Trained on near-infrared spectra from APOGEE, it demonstrated the capability of CNNs to estimate stellar parameters directly from normalized flux vectors, bypassing the need for explicit physical modeling. StarNet delivered competitive performance across a broad parameter space while offering significantly faster inference compared to traditional pipelines. However, it primarily operated on well-structured, standardized data with uniform resolution and wavelength coverage.

In contrast, the CARMENES survey provides high-resolution spectra in both the visible (VIS) and near-infrared (NIR) ranges, but the data it produces is more complex and less uniform. The spectra are recorded in multiple echelle orders, often with overlapping and unevenly sampled wavelength regions. This presents both a challenge and an opportunity: while the data is rich in detail, it requires substantial preprocessing—such as smoothing, normalization, order stitching, and resolution matching—before it can be effectively used in a deep learning model.

This project is motivated by the goal of extending the deep learning methodology pioneered by StarNet to a more challenging, real-world dataset like CARMENES. By developing a comprehensive preprocessing framework and adapting a CNN-based

model to handle high-resolution, variable-length spectra, this work aims to enable accurate and scalable estimation of stellar parameters from observed data. In doing so, it contributes to the broader effort of integrating machine learning into astrophysical workflows, particularly in preparation for the data volumes expected from future spectroscopic missions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The precise estimation of stellar atmospheric parameters—such as effective temperature ($T_{\rm eff}$), surface gravity ($\log g$), metallicity ([M/H]), and projected rotational velocity ($v \sin i$)—is fundamental to modern astrophysics. These parameters are essential for understanding stellar evolution, characterizing stellar populations, and identifying suitable host stars for exoplanet studies. In particular, M dwarfs are among the most promising targets in exoplanet research due to their abundance and favorable planet-to-star contrast ratios. However, determining their physical properties remains challenging due to their intrinsic faintness, complex molecular line-dominated spectra, and the absence of a well-defined continuum [1].

Traditional approaches for stellar parameter estimation include spectroscopic fitting techniques and empirical calibrations based on photometric indices or pseudo-equivalent widths. While widely used, these methods are often limited by their reliance on simplified models, manual feature selection, and assumptions that may not hold for all spectral types—especially in cooler stars like M dwarfs [1].

With the increasing availability of high-resolution spectral data from large surveys such as APOGEE and CARMENES, there is a growing demand for fast, accurate, and scalable parameter estimation methods. In recent years, deep learning techniques have emerged as powerful tools for analyzing large-scale astrophysical datasets. Convolutional neural networks (CNNs), in particular, offer a promising approach for learning complex patterns in spectra and predicting physical parameters directly from flux values [2].

In this work, we build upon the StarNet architecture introduced by Fabbro et al. [2], which employs a deep CNN model to infer stellar parameters from infrared spectra. StarNet demonstrated that it is possible to achieve high accuracy when trained on large synthetic datasets and tested on real observations from the APOGEE survey. The model is composed of one-dimensional convolutional and fully connected layers and is optimized using supervised regression with mean squared error as the loss function.

In parallel, the CARMENES collaboration developed a deep learning framework focused specifically on M dwarfs [1]. Their models were trained on synthetic spectra from the PHOENIX-ACES library and later applied to observed CARMENES spectra covering both the optical and near-infrared ranges. Separate CNNs were constructed for each stellar parameter, and the results were benchmarked against literature values. The study highlighted both the potential and the challenges of using synthetic data to train models that generalize well to real observations, emphasizing the impact of the so-called "synthetic gap".

This thesis aims to combine and adapt these ideas to develop a CNN-based pipeline for estimating T_{eff}, $\log g$, [M/H], and $v \sin i$ from high-resolution spectra. Using a StarNet-inspired architecture trained on PHOENIX-ACES synthetic spectra, we evaluate the model's ability to generalize to real CARMENES observations of M dwarfs. The methodology, results, and comparative analysis are presented in the subsequent chapters.

Chapter 2

Related Work

Machine learning has become a transformative approach in the analysis of stellar spectra, offering efficient and scalable solutions for estimating stellar parameters from large spectroscopic surveys. Over the last decade, a variety of models—from empirical fitting pipelines to sophisticated neural network architectures—have been developed to tackle this challenge.

One of the pioneering data-driven methods, *The Cannon* [3], introduced a generative framework for inferring stellar parameters. This approach modeled spectral flux as a function of stellar parameters using polynomial regression and then inverted this relationship for prediction. While *The Cannon* achieved notable accuracy, its performance depended heavily on the quality and comprehensiveness of the training set, often struggling with low signal-to-noise data or rare parameter regimes.

The advent of deep learning has led to significant advancements in the field. The *StarNet* model [2] adopted a discriminative strategy, employing a convolutional neural network (CNN) trained on infrared spectra from the APOGEE survey. *StarNet* demonstrated robust performance in predicting key stellar parameters—such as effective temperature ($T_{\rm eff}$), surface gravity ($\log g$), and metallicity ([Fe/H])—directly from normalized spectra. Unlike *The Cannon*, which relied on interpolation in spectral space, *StarNet* learned a non-linear mapping from spectral flux to physical parameters, showing improved resilience to noise and scalability, albeit requiring large datasets to mitigate overfitting.

The *CARMENES* project [1] focused on M dwarf stars, leveraging high-resolution spectra in the optical and near-infrared. Their deep learning models were trained exclusively on synthetic spectra generated from the PHOENIX-ACES grid, enabling broad coverage of the parameter space. Unlike *StarNet*, which used a single multi-output architecture, *CARMENES* developed separate CNNs for each stellar parameter ($T_{\rm eff}$, $\log g$, [M/H], $v \sin i$), resulting in enhanced prediction accuracy. However, the project also highlighted the persistent challenge of the "synthetic gap"—a mismatch between synthetic and real spectral distributions that can impair model

generalization [1].

To address this issue, the authors explored strategies such as spectral window selection, signal-to-noise augmentation, and the application of parameter constraints based on theoretical stellar evolution models. Notably, they integrated the PARSEC stellar evolutionary tracks [4] to exclude physically implausible parameter combinations.

Early foundational work in this area includes the application of artificial neural networks (ANNs) for stellar parameter estimation from synthetic spectra [5, 6]. More recent studies have advanced the field by introducing CNN-based approaches for automated classification and regression from stellar spectra [7, 8]. These works have established best practices for data normalization, network architecture design, and uncertainty quantification, laying the groundwork for the application of deep learning to large-scale spectroscopic datasets.

Despite the success of modern machine learning models, several open challenges remain. Issues such as model interpretability, generalizability across diverse spectral types, and sensitivity to noise, continuum fitting, and instrumental effects continue to pose obstacles. Addressing these challenges is essential for the broader integration of deep learning techniques into mainstream astrophysical analysis pipelines.

Chapter 3

Dataset

The *StarNet* model was trained on spectra from the APOGEE survey, part of SDSS-III and SDSS-IV, which provides high-quality near-infrared (H-band, 1.5–1.7 μm) spectra at a resolution of $R \sim 22{,}500$. These spectra were continuum-normalized and paired with stellar parameters—effective temperature ($T_{\rm eff}$), surface gravity ($\log g$), and metallicity ([Fe/H])—extracted from the ASPCAP pipeline. The dataset consisted of over 100,000 stars, enabling the training of a robust deep learning model for stellar parameter estimation.

In contrast, the present work utilizes data from the CARMENES survey, which delivers high-resolution spectra in both the visible (0.52–0.96 μm) and near-infrared (0.96–1.71 μm) bands, with resolutions of $R \sim 80{,}000$ and $R \sim 94{,}600$ respectively. The spectra used here are telluric-corrected templates, interpolated at B-spline knot points, and normalized to suppress the continuum shape. While the CARMENES dataset is smaller in size—comprising 382 high-quality stellar spectra in this work—it benefits from higher spectral resolution and detailed preprocessing. Stellar parameter labels are obtained from literature and external catalogs. Despite its smaller scale, the dataset's quality and coverage make it well-suited for training deep learning models aimed at precise stellar characterization.

Table 3.1: Comparison of Datasets Used in StarNet and Carmenes

Attribute	**StarNet (APOGEE)**	**CARMENES**
Survey	APOGEE (SDSS-III/IV)	CARMENES (Calar Alto Observatory)
Spectral Range	Near-Infrared (H-band: 1.5–1.7 μm)	VIS: 0.52–0.96 μm; NIR: 0.96–1.71 μm
Resolution	$R \sim 22{,}500$	VIS: $R \sim 80{,}000$; NIR: $R \sim 94{,}600$
Flux Preprocessing	Continuum-normalized	Telluric-corrected, normalized, B-spline interpolated
Labels (Ground Truth)	T_{eff}, $\log g$, [Fe/H] from ASPCAP pipeline	T_{eff}, $\log g$, [Fe/H] from literature/catalogs
Dataset Size	$\sim$100,000 spectra	382 high-quality spectra (or more, depending on subset)
Data Format	1D flux vector per star (7,215 points)	1D flux vector over fixed wavelength grid

Chapter 4

Methodology

4.1 Data Preprocessing

StarNet

The StarNet model was trained and evaluated using both synthetic and observed stellar spectra from the APOGEE survey. The observed dataset, publicly available through the official GitHub repository, includes a total of 559,484 spectral visits corresponding to 143,467 unique stars. Each star was typically observed multiple times (on average, four visits per star) to improve the robustness of spectral measurements across different observing nights.

Data Selection: To ensure high-quality and reliable training data, a series of filtering criteria were applied to the original set of visit spectra:

- Combined spectral signal-to-noise ratio (S/N) ≥ 200
- STARFLAG and ASPCAPFLAG both set to 0 (i.e., no known quality issues)
- Effective temperature $4000\,\mathrm{K} < T_{\mathrm{eff}} < 5500\,\mathrm{K}$
- Metallicity [Fe/H] > -3.0 dex
- Surface gravity $\log g \neq -9999$ (the ASPCAP default for missing values)
- Radial velocity scatter $\mathrm{V_{scatter}} < 1.0$ km/s

After applying these selection criteria, the dataset was reduced from 559,484 to 53,135 high-quality visit spectra, corresponding to 17,149 distinct stars. From this subset, 85% (i.e., 44,784 visits across 14,498 stars) was used for training and validation. Specifically, 41,000 visits were allocated for training and 3,784 for validation.

Target Parameters: The model was trained to predict the following stellar parameters:

- Effective temperature (T_{eff})
- Surface gravity ($\log g$)
- Metallicity ([Fe/H])

Data Preprocessing: Each stellar spectrum consists of 7214 flux values. The APOGEE spectrum is divided into three wavelength regions, or "chips":

- **Blue chip:** shortest wavelength range
- **Green chip:** mid-range wavelengths
- **Red chip:** longest wavelength range

To normalize the flux values, the median flux is computed for each chip and used to divide the corresponding segment of the spectrum. This chip-wise normalization helps minimize the influence of large-scale flux variations and emphasizes line features critical for parameter prediction.

4.2 CARMENES

4.2.1 Dataset Overview

The CARMENES 2023 dataset consists of high-resolution stellar spectra obtained using a dual-channel spectrograph that observes in the visible (VIS) and near-infrared (NIR) bands. Each spectrum records flux values across thousands of wavelength bins. Initially, the dataset comprised over 380 stars. However, to ensure consistency in data shape, only spectra with fixed dimensions—$(61, 3699)$ for VIS and $(56, 1999)$ for NIR—were retained. This filtering step resulted in the removal of 51 stars, leading to a final dataset of 331 well-structured, high-quality spectra.

4.2.2 Spectral Preprocessing Pipeline

To prepare the spectra for modeling, a systematic preprocessing strategy was adopted. This procedure was applied independently to both the VIS and NIR spectra of each star and consisted of three primary operations, repeated three times in succession to refine the result:

1. **Smoothing and Normalization:** Each spectrum was smoothed using the Savitzky-Golay filter to reduce small-scale noise while preserving the underlying shape of spectral features. This method was preferred over traditional boxcar filters, which often introduce edge effects and alter signal length. The Savitzky-Golay filter [9] uses two key parameters—window size and polynomial order—both empirically chosen using Bayesian Information Criterion [10](BIC) to optimize smoothness and signal integrity.

2. **Continuum Subtraction:** After smoothing, the continuum was subtracted to isolate absorption features and remove global trends in the flux values.

3. **Polynomial Approximation:** A low-degree polynomial was fitted to the continuum-subtracted spectrum to estimate and remove any residual curvature in the baseline. This step enhances the visibility of fine absorption structures.

This three-step cycle was iterated three times to ensure robust normalization and accurate continuum removal, yielding clean, continuum-normalized spectra.

4.2.3 Order Stitching and Variable-Length Handling

CARMENES spectra are recorded in multiple spectral orders. For example, VIS spectra have shape $(61, 3699)$, where the first dimension represents the number of spectral orders. These orders contain overlapping wavelength regions that must be combined to produce a continuous 1D spectrum.

During the stitching process, preference is given to order i over order $i+1$ in overlapping regions. Specifically, the flux from order i is retained as-is, and the overlapping portion of order $i+1$ is interpolated to align with order i. This strategy ensures continuity and avoids redundancy across orders.

However, due to variation in the size and position of overlapping regions, the final stitched spectra differ in total length from star to star. This variable-length output is addressed during model training by resampling or interpolation techniques to produce uniform-length inputs.

4.2.4 Spectral Resolution Considerations

The resolving power of the CARMENES instrument in the NIR arm is approximately $R \approx 80{,}400$. This defines the smallest resolvable wavelength difference $\Delta\lambda$ at a given wavelength λ as:

$$\Delta\lambda = \frac{\lambda}{R}$$

For example, at $\lambda = 10{,}000\,\text{Å}$, the minimum distinguishable wavelength interval is:

$$\Delta\lambda \approx \frac{10{,}000}{80{,}400} \approx 0.1244\,\text{Å}$$

This resolution plays a key role in preparing synthetic spectra for comparison with observational data. While numerous stellar atmosphere models exist—such as PHOENIX, MARCS, and ATLAS—we selected the BT-Settl models due to their wide wavelength coverage and high physical fidelity. However, because BT-Settl spectra are generated at much higher resolutions than CARMENES observations, a series of preprocessing steps were necessary to ensure comparability.

4.2.5 Preprocessing of Synthetic Spectra

To align the synthetic BT-Settl spectra with the observed CARMENES data, the following steps were applied:

1. **Wavelength Truncation:** The synthetic spectra were first trimmed to match the CARMENES VIS and NIR wavelength coverage. This ensured that only the relevant spectral regions were retained for analysis.
2. **Resolution Degradation:** A Gaussian low-pass filter was applied to the high-resolution BT-Settl spectra to degrade their resolution. The filter width was selected based on the CARMENES resolving power, effectively simulating how the instrument would observe the same theoretical spectrum.
3. **Wavelength Interpolation:** After resolution adjustment, the BT-Settl spectra were interpolated onto the wavelength grid of the observed CARMENES spectra. This step aligned the two datasets, allowing for direct comparison at each wavelength point.
4. **Flux Normalization:** Finally, the synthetic flux values were normalized using the same procedure applied to the observed spectra. This removed continuum variations and emphasized relative absorption features.

Following these steps, the BT-Settl synthetic spectra were in a comparable format to the observed CARMENES data—matched in resolution, wavelength, and normalization—allowing them to be used effectively for model training and benchmarking.

4.2.6 Wavelength Grid Alignment and Gap Handling

To ensure consistency across all samples, observed spectra were trimmed to a standardized wavelength range within the VIS and NIR bands. Some spectra exhibited small gaps due to instrument stitching or detector artifacts. These gaps were identified, catalogued, and treated appropriately. Spectra with major discontinuities were excluded. Additionally, all wavelength grids were validated for uniqueness and precision to ensure proper alignment between samples.

4.2.7 Ground Truth Validation

Each stellar spectrum was associated with physical parameters: effective temperature (T_{eff}), surface gravity ($\log g$), and metallicity ([Fe/H]). All labels were carefully reviewed, and entries containing missing or placeholder values (e.g., -9999) were removed to preserve label integrity and training quality.

4.2.8 Final Dataset Summary

The final dataset used in this work consists of 331 carefully preprocessed stellar spectra from CARMENES, supplemented by resolution-adjusted BT-Settl models for validation and comparison. Each spectrum underwent smoothing, continuum normalization, wavelength alignment, and order stitching. The resulting data is well-suited for training convolutional neural networks to predict stellar parameters across diverse observational conditions.

Chapter 5

Architecture

In this chapter, we describe the deep learning models developed for predicting stellar atmospheric parameters from spectral data. Two distinct convolutional neural network (CNN) architectures were explored: one inspired by the StarNet model, and the other designed in the style of a simplified VGG-based 1D CNN.

The StarNet-like architecture closely follows the design principles laid out in the original StarNet work, adapting its convolutional and dense layers to suit the structure and resolution of the current dataset. It is tailored to handle high-resolution spectra and capture complex patterns present in the flux values across wavelengths.

In parallel, a lightweight VGG-style 1D CNN model was implemented, consisting of stacked convolutional blocks followed by pooling and a fully connected regression layer. This design offers a more compact alternative that remains effective for learning from one-dimensional spectral inputs.

Both models were evaluated under different regression settings. Specifically, they were trained to predict either one, two, or all three stellar parameters—effective temperature ($T_{\rm eff}$), surface gravity ($\log g$), and metallicity ([Fe/H])—in separate experiments. This flexible setup allowed us to assess how the models perform when learning parameters individually versus jointly.

The sections that follow provide a detailed explanation of each model's architecture, design rationale, and training configuration.

StarNet Convolutional Neural Network (CNN) Model

StarNet is a one-dimensional convolutional neural network (1D-CNN) developed to estimate fundamental stellar parameters—effective temperature ($T_{\rm eff}$), surface gravity ($\log g$), and metallicity ([Fe/H])—directly from normalized stellar spectra [2]. It follows a supervised learning paradigm, where the model is trained on spectra with known parameter labels, learning a mapping from flux vectors to physical quantities.

The architecture of StarNet consists of seven layers in total. Starting from the in-

put layer, which receives a 1D flux array (typically with 7214 wavelength points), the model applies two sequential convolutional layers with 4 and 16 filters, respectively. These layers are designed to extract localized spectral features. A max pooling layer follows, which downsamples the signal and reduces dimensionality while preserving the most relevant patterns. The final part of the network comprises three fully connected (dense) layers with 256, 128, and 3 neurons, respectively. The last layer outputs predictions for $T_{\rm eff}$, $\log g$, and [Fe/H].

The model uses the ADAM[11] optimizer for training and minimizes a mean squared error (MSE) loss function. ReLU [12](Rectified Linear Unit) activations are used across all layers except the output. To accelerate convergence and improve weight initialization, the He normal initialization method is applied. A validation set is held out during training to monitor model performance and prevent overfitting.

StarNet can be trained on either real spectra from surveys like APOGEE or synthetic spectra generated via stellar atmosphere models. When using synthetic data, Gaussian noise is deliberately added to the input to simulate real observational uncertainties and enhance the model's generalization capability. One of StarNet's key strengths is its robustness to low signal-to-noise (S/N) data, attributed to its training on individual visit spectra rather than only on high-quality stacked spectra.

By combining convolutional layers for local feature detection and fully connected layers for global integration, StarNet achieves high accuracy and generalizability in predicting stellar parameters across a diverse range of stars [2].

(See Figure: 5.1)

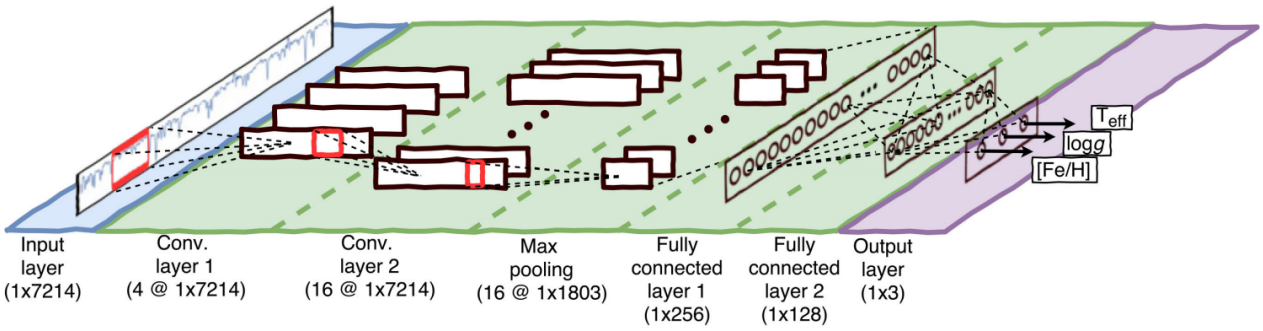


Figure 5.1: StarNet Architecture

VGG Regression Model Architecture

To estimate stellar parameters from one-dimensional spectra, a deep learning model was developed based on a modified VGG-style convolutional neural network (CNN). The architecture is tailored for regression tasks and designed to extract meaningful features from normalized 1D spectral inputs.

The model accepts input in the form of a sequence of vectors, where each vector represents flux-related measurements at a given wavelength. The input shape is defined as (batch size, sequence length, input channels). Before feeding the data into the network, the input tensor is permuted to match PyTorch's 'Conv1d' format: (batch size, channels, sequence length).

The architecture begins with a feature extraction block composed of three convolutional layers. The first convolutional layer maps the input channels to 64 filters using a kernel size of 3 and includes padding to maintain spatial dimensions. This is followed by a ReLU activation and a max-pooling layer with a stride of 2, which reduces the sequence length by half. The second convolutional layer increases the feature dimensionality to 128 and is again followed by ReLU activation and max pooling. The third and final convolutional layer outputs 256 channels, which are subsequently passed through a global average pooling layer implemented via 'AdaptiveAvgPool1d(1)'. This compresses each feature map into a single scalar, reducing the output to a fixed-size feature vector of shape (batch size, 256) regardless of input sequence length.

The output of the convolutional block is flattened and passed through a fully connected linear layer that maps the 256-dimensional feature vector to the target regression space. Depending on the task, this may include one or more stellar parameters such as effective temperature (T_{eff}), surface gravity ($\log g$), or metallicity ([Fe/H]).

Overall, this VGG-inspired 1D CNN model combines local feature extraction with global summarization to provide a robust, compact representation of spectral features. Its design allows it to learn patterns across different wavelength regions while maintaining a relatively small number of parameters. The model is particularly well-suited for spectroscopic regression problems, including stellar parameter estimation from normalized flux spectra.(See Figure: 5.2)

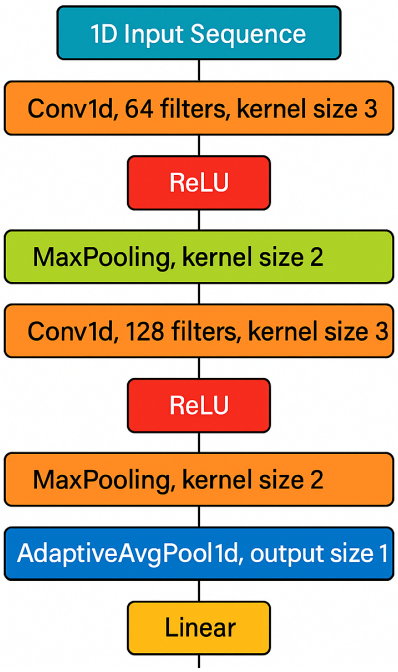


Figure 5.2: VGG Style Architecture

5.1 Components

To understand the internal workings of the CNN models, we detail the following components and techniques used in our implementations:

1. **Loss Function:**

The StarNet model is trained using the Mean Squared Error (MSE) loss function, which is well-suited for regression tasks involving continuous output variables such as stellar parameters. This loss function measures the average of the squared differences between the predicted values and the true labels, ensuring that larger errors are penalized more heavily.

Mathematically, the MSE loss is defined as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

where N is the number of training samples, y_i denotes the ground truth value, and $\hat{y}_i$ is the corresponding model prediction. During training, the model optimizes its parameters by minimizing this loss over the training set.

2. **Optimizer:**

To train both models effectively, we used the Adam optimizer, a popular choice in deep learning due to its ability to adaptively adjust learning rates for each parameter. Adam combines the benefits of two earlier optimization methods—AdaGrad, which works well for sparse gradients, and RMSProp, which is effective for non-stationary objectives. This makes it particularly suitable for training deep neural networks on large, high-dimensional datasets such as stellar spectra.

The optimizer updates model weights using estimates of both the first moment (mean) and the second moment (uncentered variance) of the gradients. The mathematical formulation of the Adam optimizer is as follows:

$$
\begin{aligned}
m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t \\
v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\
\hat{m}_t &= \frac{m_t}{1-\beta_1^t} \\
\hat{v}_t &= \frac{v_t}{1-\beta_2^t} \\
\theta_t &= \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}
\end{aligned}
$$

where g_t is the gradient at time step t, m_t and v_t are the first and second moment estimates, β_1 and β_2 are decay rates typically set to 0.9 and 0.999 respectively, ϵ is a small constant (e.g., 10^{-8}) for numerical stability, and α is the learning rate.

3.**Learning Rate:** The learning rate plays a crucial role in guiding how quickly or slowly a model updates its weights during training. In this work, we set the learning rate to 0.0007 for the StarNet-based model and 1×10^{-3} for the VGG-style CNN. These values were selected through preliminary experimentation to balance convergence speed with training stability. A smaller learning rate allows the model to approach the optimal solution more gradually and accurately, though it typically requires more training epochs to reach convergence.

4. **1D Convolution:**

One-dimensional (1D) convolution is a fundamental operation used in neural networks to extract localized patterns from sequential data, such as time series or spectra. It works by sliding a small filter (also called a kernel) over the input sequence and computing a dot product at each position. This operation allows the network to detect local features irrespective of their exact location in the input.

Mathematical Formulation: Given an input sequence $x = [x_1, x_2, \ldots, x_n]$ and a kernel $w = [w_1, w_2, \ldots, w_k]$ of size k, the 1D convolution output $y = [y_1, y_2, \ldots, y_{n-k+1}]$

(assuming no padding and unit stride) is computed as:

$$y_i = \sum_{j=1}^{k} w_j \cdot x_{i+j-1} \quad \text{for } i = 1, 2, \ldots, n-k+1$$

Example: Let the input be:

$$x = [2,\ 1,\ 3,\ 0,\ 1]$$

and the kernel (filter) be:

$$w = [1,\ 0,\ -1]$$

Then the convolution output is:

$$y_1 = (1 \cdot 2) + (0 \cdot 1) + (-1 \cdot 3) = 2 + 0 - 3 = -1$$
$$y_2 = (1 \cdot 1) + (0 \cdot 3) + (-1 \cdot 0) = 1 + 0 - 0 = 1$$
$$y_3 = (1 \cdot 3) + (0 \cdot 0) + (-1 \cdot 1) = 3 + 0 - 1 = 2$$

So the output of the 1D convolution is:

$$y = [-1,\ 1,\ 2]$$

Interpretation: This operation helps highlight changes in the sequence. For example, the kernel used above acts similarly to an edge detector, emphasizing where the input values increase or decrease.

5. **Activation Function:** ReLU (Rectified Linear Unit) that we have used as activation function,is defined as:

$$\text{ReLU}(x) = \max(0, x)$$

ReLU has tendency to introduce non-linearity to our model and helps in vanishing gradient problem, which leads to faster and effective training.

6. MaxPooling (1D):

In one-dimensional convolutional neural networks, MaxPooling is used to downsample the feature vector along the temporal or sequential dimension. It reduces the input size by selecting the maximum value within a specified window, thereby decreasing computational cost and helping the model retain the most prominent features.

$$\text{Input Vector} = [3,\ 5,\ 2,\ 8,\ 6,\ 1,\ 7,\ 4]$$

Applying 1D MaxPooling with a size of kernel 2 and stride 2:

$$\text{Pooled Output} = [\max(3,5),\ \max(2,8),\ \max(6,1),\ \max(7,4)] = [5,\ 8,\ 6,\ 7]$$

This operation reduces the original 8-element vector to 4 elements, effectively summarizing local information and reducing overfitting by limiting the number of downstream parameters.

The selection of each component in our CNN architecture—from the activation functions to the use of the Adam optimizer—was carefully made to improve the model's capacity to learn complex patterns in the data and accurately predict stellar parameters.

Chapter 6

Results

This section outlines the experimental design and presents the results of our investigation into stellar parameter prediction using convolutional neural networks (CNNs). The goal of the study is to estimate key stellar parameters—including effective temperature (T_{eff}), surface gravity ($\log g$), and metallicity ([Fe/H])—from spectroscopic data. We evaluate two deep learning models: one inspired by the StarNet architecture and another based on a custom VGG-style CNN.

Both models were trained and tested on observed spectra from the CARMENES survey as well as synthetic spectra generated from BT-Settl models. To measure predictive accuracy, we employ widely used regression metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE). These metrics provide a clear understanding of how closely the model outputs align with the true parameter values.

Additionally, we compare the performance of the models across different training conditions and input types to assess the impact of architectural choices and data preprocessing steps. This comparison helps highlight the strengths and limitations of each approach in both synthetic and observational domains.

6.1 Performance:

To evaluate how well our models predict continuous stellar parameters such as T_{eff}, $\log g$, and [Fe/H], we employ several standard regression metrics. These include Mean Absolute Error (MAE), Mean Squared Error (MSE). Each of these metrics captures different aspects of model performance, providing a more complete understanding of prediction accuracy.

6.1.1 Mean Absolute Error (MAE)

The Mean Absolute Error measures the average magnitude of prediction errors, without considering their direction. It is a straightforward metric that gives a sense of how much the predictions typically deviate from the true values.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

Lower MAE values indicate better performance. Since it treats all errors equally, MAE is less sensitive to outliers compared to MSE.

6.1.2 Mean Squared Error (MSE)

The Mean Squared Error calculates the average of the squared differences between predicted and actual values:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

By squaring the errors, MSE penalizes larger deviations more than smaller ones. This makes it particularly useful when large errors are undesirable. However, it can be more sensitive to outliers than MAE.

Model	**Parameter**	**MAE**	**MSE**
StarNet	T	85.65	5974.49
StarNet	TG	71.33	4568.72
StarNet	TGM	97.03	2395.98
VGG Style CNN	T	52.68	4763.94
VGG Style CNN	TG	45.87	3362.30
VGG Style CNN	TGM	105.23	6756.42

Table 6.1: Regression performance of different models on stellar parameter prediction (e.g., T_{eff})

Chapter 7

Challenges and Future Work

This chapter outlines the primary challenges encountered during the study and proposes future directions to enhance the model's flexibility, accuracy, and real-world applicability.

7.1 Challenges

7.1.1 Handling Variable-Length Spectra

The CARMENES spectra are captured across multiple overlapping echelle orders, which results in final stitched spectra of varying lengths. Since standard convolutional neural networks require inputs of fixed size, all spectra were interpolated to a uniform length for compatibility. While this preprocessing step is necessary, it may distort important absorption lines or lead to inconsistent feature alignment across samples, limiting the network's ability to learn generalizable patterns.

7.1.2 Synthetic vs Observed Spectra Discrepancy

Another key challenge lies in the difference between synthetic data (e.g., BT-Settl spectra) and actual observational data (e.g., CARMENES). Despite preprocessing steps such as normalization and resolution matching, there remain inconsistencies due to instrumental noise, calibration errors, and real atmospheric effects. These domain discrepancies can negatively affect the generalization performance of models trained solely on synthetic spectra.

7.2 Future Work

7.2.1 Architectures for Variable-Length Input

To better accommodate variable-length spectral data, future models could move beyond fixed-size convolutional architectures. Sequence-aware models such as Long Short-Term Memory (LSTM) networks, Transformer-based architectures [13], and permutation-invariant frameworks like *Deep Sets* [14] are promising alternatives. These models can learn directly from the unaltered spectral sequences, reducing the risk of losing important structural information during interpolation.

7.2.2 Fusion of VIS and NIR Bands

In this work, the visible (VIS) and near-infrared (NIR) spectra were processed independently. A joint learning framework that fuses both spectral ranges could leverage complementary information from the two bands and improve the model's ability to capture subtle patterns across the full wavelength span.

7.2.3 Improving Synthetic-to-Observed Generalization

To bridge the domain gap between synthetic and observed spectra, transfer learning approaches can be employed. This involves pretraining the model on large synthetic datasets and fine-tuning it using a smaller set of real observations. Additionally, domain adaptation techniques could help align the feature distributions between the two data sources.

7.2.4 Uncertainty Estimation

Adding uncertainty quantification to the predictions can enhance their scientific utility. Future models may incorporate Bayesian neural networks or apply Monte Carlo dropout to provide confidence intervals for each estimated stellar parameter. This would be especially beneficial in flagging unreliable predictions and guiding follow-up observations.

Conclusion

Tackling these challenges through improved model design and learning strategies will lead to more robust, accurate, and interpretable models. Such advancements would expand the practical use of deep learning in stellar spectroscopy and help process the next generation of large-scale astronomical surveys.

Bibliography

[1] V. M. Passegger, A. Bello-García, J. Ordieres-Meré, J. A. Caballero, A. Schweitzer, A. González-Marcos, I. Ribas, A. Reiners, A. Quirrenbach, P. J. Amado, M. Azzaro, F. F. Bauer, V. J. S. Béjar, M. Cortés-Contreras, S. Dreizler, A. P. Hatzes, Th. Henning, S. V. Jeffers, A. Kaminski, M. Kürster, M. Lafarga, E. Marfil, D. Montes, J. C. Morales, E. Nagel, L. M. Sarro, E. Solano, H. M. Tabernero, and M. Zechmeister. The carmenes search for exoplanets around m dwarfs: A deep learning approach to determine fundamental parameters of target stars. *Astronomy amp; Astrophysics*, 642:A22, September 2020.

[2] S Fabbro, K A Venn, T O'Briain, S Bialek, C L Kielty, F Jahandar, and S Monty. An application of deep learning in the analysis of stellar spectra. *Monthly Notices of the Royal Astronomical Society*, 475(3):2978–2993, December 2017.

[3] M. Ness, D. W. Hogg, H. W. Rix, A. Y. Q. Ho, and G. Zasowski. The cannon: A data-driven approach to stellar label determination. *The Astrophysical Journal*, 808(1):16, 2015.

[4] A. Bressan, P. Marigo, L. Girardi, B. Salasnich, C. Dal Cero, S. Rubele, and A. Nanni. Parsec: Stellar evolutionary tracks and isochrones. *Monthly Notices of the Royal Astronomical Society*, 427(1):127–145, 2012.

[5] C. A. L. Bailer-Jones, M. Irwin, G. Gilmore, and T. von Hippel. Automated classification of stellar spectra using neural networks. *Monthly Notices of the Royal Astronomical Society*, 292(1):157–166, 1997.

[6] C. A. L. Bailer-Jones. Data-driven stellar parameter determination from low-resolution spectra. *Astronomy & Astrophysics*, 357:197–209, 2000.

[7] K. Wang, P. Guo, and A.-L. Luo. Automated stellar spectral classification using deep convolutional neural networks. *Monthly Notices of the Royal Astronomical Society*, 465(4):4311–4321, 2017.

[8] H. Li, K. Pan, and J. Duan. Spectral classification with convolutional neural networks. *Publications of the Astronomical Society of the Pacific*, 129(975):124101, 2017.

[9] Ronald W. Schafer. What is a savitzky-golay filter? [lecture notes]. *IEEE Signal Processing Magazine*, 28(4):111–117, 2011.

[10] Sumio Watanabe. A widely applicable bayesian information criterion, 2012.

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[12] Abien Fred Agarap. Deep learning using rectified linear units (relu), 2019.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[14] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets, 2018.