TJAA Vol. **6**, *Issue*1, *p.*18–27 (2025). DOI: 10.55064/tjaa.1708479 *Research Article*

Classification of Eclipsing Binary Light Curves with Deep Learning Neural Network Algorithms

Burak Ulaş[1] ⋆
[1] *Department of Space Sciences and Technologies, Faculty of Sciences, Çanakkale Onsekiz Mart University, 17100 Çanakkale, Türkiye*

Accepted: June 24, 2025. Revised: June 24, 2025. Received: May 28, 2025.

Abstract
We present an image classification algorithm utilising a deep learning convolutional neural network architecture, which categorises the morphologies of eclipsing binary systems based on their light curves. The algorithm trains the machine with light curve images generated from the observational data of eclipsing binary stars in contact, detached and semi-detached morphologies, whose light curves are provided by Kepler, ASAS and CALEB catalogues. The structure of the architecture is explained, the parameters of the network layers and the resulting metrics are discussed. Our results show that the algorithm, which is selected among 132 neural network architectures, estimates the morphological classes of an independent validation dataset, 705 true data, with an accuracy of 92%.

Key words: (stars:) binaries: eclipsing — methods: data analysis — techniques: image processing

1 Introduction

Deep learning techniques strengthen their solid ground in various areas from art to science every day. In the present, countless machine and deep learning applications let the researchers achieve faster and more precise results in their studies, as well as changing daily life. Convolutional neural networks, a specialised architecture in deep learning algorithms using neural networks, give an opportunity to use powerful methods in some processes such as image recognition and classification. The prototype of these networks, the neocognitron, was proposed by Fukushima (1980). Lecun et al. (1998) introduced the convolutional networks and demonstrated their robust performance in handling variations in 2D shapes. They also noted the advantages of fast learning in their handwriting experiment. The improvement in both hardware and software technology allows taking giant leaps in the usage and development of convolutional neural networks. For instance, a famous architecture, AlexNet (Krizhevsky et al. 2012) classified 1.2 million images with an accuracy value of about 85% in the ImageNet computer vision challenge. CoAtNet (Dai et al. 2021) also reached 91% accuracy by improving the model capacity and introducing the hybrid models.

Eclipsing binary stars are stellar systems showing light variations in their light curves due to occultations of the companions light. Their importance arises from being tools for deriving the crucial stellar parameters precisely, and therefore, allowing researchers to determine the structure of the stars in realistic estimations. Guinan (1993) remarked that the analyses of their light curves enable us to estimate the important parameters like mass, radius, luminosity, effective temperature as well as atmospheric properties. These systems appear in several morphological types, mainly contact, detached, and semi-detached (Kopal 1955; Bradstreet 2005), which are classified based on Roche lobe geometry. While stellar evolution can drive transitions between these morphological classes through mass transfer and Roche lobe overflow, the classification itself is geometric, determined by whether and how the component stars fill their respective Roche lobes. Thus, accurate morphological classification is essential for determining fundamental stellar parameters, understanding mass transfer processes, and predicting evolutionary outcomes of binary systems, which constitute a significant fraction of stellar populations in our galaxy.

Researchers made efforts in detecting, fitting and classifying the light curves of binary systems using machine and deep learning algorithms. Wyrzykowski et al. (2003) proposed an algorithm using artificial neural networks and detected 2580 binary systems in Large Magellanic Cloud based on the OGLE data (Udalski et al. 1998). Prša et al. (2008) presented an artificial neural network trained with data points of 33235 light curve samples of detached eclipsing binaries for deriving some physical parameters of eclipsing binary stars selected from several databases. The authors remarked that the success rate of the algorithm is more than 90% for OGLE and CALEB data sets, respectively. Kochoska et al. (2020) evaluated different fitting methods and concluded that machine learning techniques are useful tools for estimating the initial parameters of the binaries. The preliminary results of a systematic classification for the light curve morphologies of eclipsing binaries from TESS (Ricker et al. 2015) based on a machine learning technique, were published by Birky et al. (2020). Ulaş (2020) suggested a deep learning image classification algorithm for the classification of light curve morphologies of ASAS-SN eclipsing binaries with an accuracy value of 92%. Lately, Čokina et al. (2021a) introduced a two-class (detached and overcontact) classification based on the 491425 synthetic light curve data generated by ELISa software (Čokina et al. 2021b). The authors accomplished 98% accuracy with their combined deep learning architecture. Bódi & Hajdu (2021) applied a machine learning algorithm that utilised locally

⋆ burak.ulas@comu.edu.tr

linear embedding to classify the morphologies of OGLE binaries. Szklenár et al. (2022) classified variable stars, including eclipsing binaries, based on their visual characteristics, using a multi-input neural network trained on OGLE-III data. Prsa & Wrona (2024) developed a neural network, trained on synthetic light curves, enabling robust and efficient posterior sampling for modelling thousands of observed EB systems. More recently, a combined classification scheme (Long Short-Term Memory and Convolutional Neural Network) that classifies light curves by making use of both the Roche geometry-based morphology and the presence of spot-induced light curve modulations was proposed by Parimucha et al. (2024) and a classification of over 60,000 eclipsing binaries in the Zwicky Transient Facility database using machine learning algorithms was made by Healy et al. (2024).

Applications of the machine learning techniques on the astrophysical data are in progress and promise novel results in the area. The potential of the subject motivated us to apply the method to the classification of eclipsing binary light curves according to their morphologies. In the following section, we introduce the details and structure of the light curve data used in the study. Sec. 3 deals with the details of our code and the architecture of the convolutional neural network. The results are discussed and the concluding remarks are given in the last section.

2 Light Curve Data

The algorithm needs light curve images corresponding to certain morphological classes of binary stars to train the machine and perform the classification. Therefore, we collected real light curve data to construct light curve images of eclipsing binary stars. The data for eclipsing binary stars with known morphological types are provided from three main data sources in this study: Kepler Eclipsing Binary Catalog (Kirk et al. 2016), All Sky Automated Survey (ASAS, Pojmanski 1997) and Catalog and Atlas of Eclipsing Binaries (CALEB, former EBOLA, Bradstreet et al. 2004).

Kepler light curves were accessed through the Kepler Eclipsing Binary Catalog (Kirk et al. 2016). The authors catalogued some basic properties of 2920 binary systems and indicated a parameter for their morphological classes. The catalog provides light curves consisting of long cadence Kepler observations processed by the Kepler Science Operations Center pipeline (Jenkins et al. 2010). It applies advanced systematic error correction using Cotrending Basis Vectors (CBVs) and provides PDCSAP (Pre-search Data Conditioning Simple Aperture Photometry) fluxes that are detrended by using the common features in the CBVs. We made use of these PDCSAP light curves in the classification of Kepler data. The parameter c, introduced by Matijevič et al. (2012) using the locally linear embedding method, is a classification criterion for contact ($0.7<c<0.8$), detached ($c<0.5$) and semi-detached ($0.5<c<0.7$) binary systems. We note that Kepler systems with $c\geq0.8$ are predominantly ellipsoidal variables or uncertain types that typically show only ellipsoidal modulation without clear eclipse features. Since our deep learning algorithm requires well-defined eclipse features for accurate morphological classification, we excluded all Kepler systems with $c\geq0.8$ from our analysis. For consistency across all three databases, we adopt the term 'contact' throughout this work, noting that it corresponds to systems with $0.7<c<0.8$ in the Kepler classification scheme, and to contact binaries as defined in the ASAS and CALEB catalogs. We eventually collected 1913 binary systems (239 contact, 1253 detached and 421 semi-detached) with corresponding c parameters. To construct the light curve images, the orbital phase and detrended flux values were used as provided in the catalog. No additional processing, such as further detrending or outlier removal, was applied to these preprocessed data in our analysis.

ASAS (Pojmanski 1997) variable star database was also used to gather light curve data and morphological classes of the eclipsing binary stars. The variability class of the targets in the catalogue were determined by Pojmanski (2002) using an approach based on multidimensional parametric space as well as an extended method using certain Fourier coefficients. We were able to collect data and morphological classes of 5907 binary systems through the database query service (ASAS). The phases for the light curves were calculated by adopting the times of minimum and orbital period values from the ACVS (ASAS Catalog of Variable Stars) list given by the author. The magnitudes were also converted to normalised fluxes by deriving the maximum magnitudes for the corresponding light curves for the systems.

The CALEB data were achieved via the catalogue's web page. The author catalogued light curves and observational properties of 305 individual stars with their morphological classes. Since the catalogue contains light curves in several filters for many stars, the actual number of data exceeds the above-mentioned value. 1632 light curves from the database were included in our study. The light curve images were constructed by using the orbital phase and flux values given by the catalogue.

The 256×256 pixel light curve images were generated by plotting the data in the 0.25-1.25 phase interval, where phase 0.0 corresponds to the primary minimum. For Kepler and CALEB targets, the phase values were taken directly from the respective databases, while for ASAS binaries, they were calculated using the corresponding times of minima, as previously described. The total number of data decreased after eliminating the light curves which (1) show very large scatter that obscured the eclipse features (assessed visually rather than using a fixed quantitative criterion due to the heterogeneous nature of the data sources), (2) have very few data points to adequately sample the orbital phase and (3) do not resemble the light curve of an eclipsing binary system. This visual inspection approach was necessary because the three databases have significantly different observational characteristics.

Kepler provides high-precision space-based photometry with a regular long cadence, while ASAS and CALEB contain ground-based observations with varying cadences and photometric precision. A uniform quantitative scatter criterion would have been inappropriate across such diverse data sources. The incorrect orbital period values, especially in the ASAS Catalog, were also responsible for the decrease in the number of light curves. Additionally, the entire dataset from three databases was checked by eye to prevent misclassification. The data in each class was also balanced. Namely, we limit the number of light curve images in each morphological class to be equal, thus, it is one-third of the total number of data in a given database (e.g. 2286 light curves from ASAS contain 762 images from each individual class; contact, detached and semi-detached).

We randomly chose 657, 2286 and 585 images from the final datasets of Kepler, ASAS and CALEB, respectively. Note

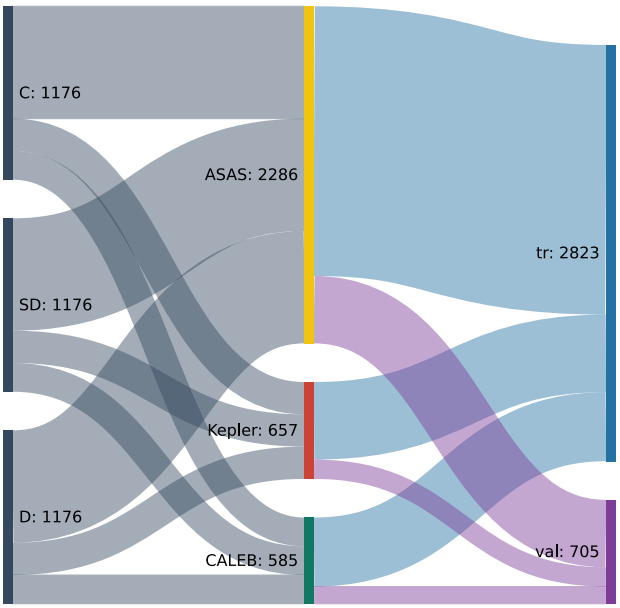

Figure 1. Sankey diagram showing the distribution of 3528 light curve image data in three nodes: morphology, database and type of the dataset. C, D and SD refer to contact, detached and semi-detached morphologies, respectively. **tr** indicates training set, while **val** remarks the validation data. See text for details. Diagram created using SankeyMATIC.

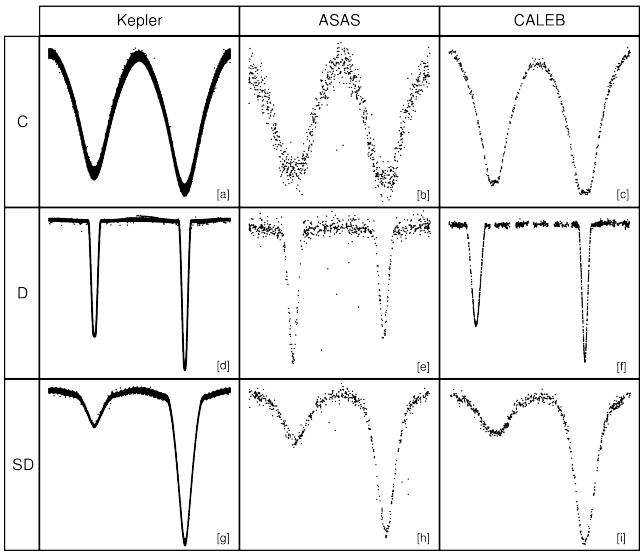

Figure 2. Selected images from the training set generated by using the light curve data of KIC 12458133 (a), ASAS 065227-5524.6 (b), V572 Cen (c), KIC 06545018 (d), ASAS 075602-4454.8 (e), QX Car (f), KIC 03954798 (g), ASAS 101553-6012.9 (h) and TZ Lyr from three different databases. C, D and SD refer to contact, detached and semi-detached morphologies, respectively. Figure created using gnuplot.

that the 585 CALEB light curves in our final dataset correspond to some systems observed in different photometric filters. While this could potentially introduce a bias toward systems with multi-band coverage, we addressed this concern through our class balancing procedure. By ensuring equal representation across morphological classes (contact, detached, and semi-detached), we prevented any systematic overrepresentation of individual systems. Moreover, the inclusion of multi-filter observations serves to enhance the model's ability to recognize morphological features that are wavelength-independent, contributing to more robust classification. Therefore, a total of 3528 light curves from all databases were selected to use for image classification. The number of the light curves in the training set is 2823, while the validation set covers 705 light curves, about 20% of the whole sample, following the Pareto principle (Moore 1897; Juran & Godfrey 1999).

It should be noted that we employed a two-way data split (training and validation) rather than the three-way split (training, validation, and test) that is sometimes utilized in machine learning applications. This methodological choice was made due to the relatively limited size of our total dataset (3528 images) and the need to maintain a balanced representation across morphological classes. In our implementation, the validation set of 705 images serves as an independent test set for final performance evaluation, as these data were completely withheld during the training process and were not used for any model optimization decisions. This approach ensures that our reported 92% accuracy represents performance on genuinely unseen data, though we acknowledge that having a separate test set would provide an additional layer of validation. Fig. 1 is a Sankey diagram showing the relation among the morphologies, databases and datasets based on the number of light curves. Nine samples of data with different morphologies in the training set from three databases are illustrated in Fig. 2.

We acknowledge that the three data sources also exhibit significantly different observational characteristics in terms of cadence, precision, and sampling patterns. To address these, we applied consistent preprocessing: using phase-folded light curves, plotting them in the 0.25-1.25 phase range, normalizing fluxes and filtering out unsuitable light curves. We then converted all data into the same size (256×256 pixel) images to create a uniform visual representation. Rather than applying aggressive harmonization techniques like rebinning or smoothing, which may risk altering genuine astrophysical features, we relied on the CNN's capacity to learn robust patterns from these images, aided by Gaussian blur during augmentation to handle dataset-specific noise. While differences between datasets may still influence results, our model's 92% accuracy across a mixed validation set suggests it has learned features resilient to these variations.

3 Architecture of the Neural Network

A Python (Van Rossum & Drake 2009) code (ebclass) was written to set a deep learning neural network algorithm and thus train the machine to classify the light curve images generated from the light curve data. The code proceeds with seed fixing for NumPy, Python and TensorFlow to avoid randomness and make the results reproducible, and yet randomness that may arise from the calculations on the GPU still remains. It must be noted that when running the code on a GPU, randomness may alter the results slightly from one run to another due to the parallel operations, as remarked by the Keras team. The problem can be solved by conducting the calculations on a

CPU, however, neural networks are computationally expensive, and it takes an extremely long time to achieve the results on a CPU.

The light curve image data in three folders (C: contact, D: detached and SD: semi-detached) were indicated in the code and the total number of data files inside those directories was commanded to display on the output. The sizes of input images were also defined. We applied data augmentation by adding a random Gaussian blur to images in the training dataset. Augmentation enriches information about data by applying certain operations to a given dataset, and it helps prevent overfitting (Shorten & Khoshgoftaar 2019). As some augmentation methods (e.g. flip and rotation) may cause changes in the shape of the light curve, we avoided employing further augmentation to our data. Training and validation directories were defined to direct the algorithm to the targeted data, which was intended to be dealt with. The data generation process resizes the images to 128×128 pixels to save on computing time and converts to greyscale since the colour is not a distinctive feature for our data. In data generation, `class_mode` argument was `categorical` which defines 2D *one-hot* encoded labels that contain one *hot* (1) among all other *cold* (0) values (Harris & Harris 2007).

The backbone of the algorithm is the sequential model, a stack of layers, which includes several hyperparameters forming the convolutional neural network architecture, which consists of convolutional, pooling, fully connected and output layers. Convolutional layers use kernels with the size of (3, 3), referring to the size of the convolutional window (Chollet et al. 2015). Rectified Linear Unit (ReLU) function, $f(x)$=max$(0, x)$, was selected as the activation function for the layers, which assigns 0 for the values smaller than zero and returns the input value for non-negative inputs (Goodfellow et al. 2016). The training is done by using the stochastic gradient descent algorithm (Ruder 2016) to achieve the converged result and then ReLU provides relatively more effortless optimisation and calculation since it is a piece-wise linear function consisting of two linear segments. The convolution operation was done by applying a $L2$ regularisation penalty (Cortes et al. 2009). The regularisation term is

$$\lambda \sum_{i=1}^{N} \omega_i^2 \tag{1}$$

where λ (=0.001 in our case), ω and N are the regularisation parameters, weight and the number of features, respectively. The term adds the squared weights to the loss function and controls the weights to be relatively small values, thus, preventing the model from overfitting and structural complexity. The `padding` hyperparameter was adjusted to `same`, which guarantees that the feature map is the same size as the input (Chollet et al. 2015). The stride value was left default, (1, 1), which corresponds to the filter moves one pixel at a time. We also applied max pooling operation (Christlein et al. 2019) with a pool size of (2, 2) between convolutional layers. It basically downsamples the input data by taking the maximum values within the pool size. The pooling also helps avoid overfitting and lowers the computation time. The above processes lead to the feature extraction and the next stage, the flattening operation, is necessary to convert the 3D tensor output from the final convolutional layer into a 1D vector required as input for the dense layers (Basha et al. 2020). Flattening converts the data into a 1-dimensional array, the shape which is mandatory to make the algorithm be able to perform the classification. A Dropout layer with a rate of 0.5 follows the flattening, which avoids the model from overfitting, as mentioned by Srivastava et al. (2014). The last steps of the convolutional neural network include fully connected layers where the classification takes place. All the input neurons are connected to the neurons in the present layer at this stage (Géron 2017), therefore, the dimensionality of the first dense layer is equal to the filter number of the last convolutional layer. The dimension was set to 3 in the output layer of the network since we have three classes (contact, detached and semi-detached). Probabilistic distribution was determined using the softmax activation function as it is appropriate for multiclass classifications using the categorical cross-entropy loss function, which is

$$L = -\sum_{i=1}^{n} p(x_i) \log_e(q(x_i)) \tag{2}$$

given by Zhou et al. (2021), where $p(x_i)$ and $q(x_i)$ denote real and predicted distributions, and n is the number of classes. Our architecture consists of 7 trainable layers (5 convolutional and 2 dense layers) and 7 non-trainable layers (5 max pooling, 1 flatten, and 1 dropout layer). Note that these numbers are specific to our architecture and would vary with different network designs. The Adam algorithm (Kingma & Ba 2015) was chosen as the optimiser, which uses the stochastic gradient descent method. Adam is appropriate for multiclass problems and can be adjusted with the learning rate hyperparameter. Learning rate is basically referring to the step size (Murphy 2012) in the convergence of the learning process. Tuning this parameter plays an important role in obtaining reliable results during calculations. Large values can result in straying from convergence, while small values may extend the training time. We control the learning process by monitoring the validation loss through `EarlyStopping` callback, which is known to boost the performance of algorithms (Yao et al. 2007). The arguments of early stopping were arranged to stop training when no decrement in validation loss is observed in 20 consecutive epochs, and therefore, training was prevented from overfitting. Another callback, `ModelCheckpoint`, was also included in the code to save the best model having the maximum validation accuracy in a model file. We compiled our model using cross-entropy loss, as mentioned before, based on accuracy evaluation. The final operation, fitting the model, was done by specifying the number of training samples per iteration (`batch_size=32`), generators, and the callbacks mentioned above. Additionally, in our code, we stored the number of filters in convolutional layers and learning rate values in variables (`l1`, `l2`, `l3`, `l4` and `lrate`) to be able to test various architectures quicker, only by changing the set of variables.

The aim of the neural network is to minimise the cross-entropy type loss function and reach the maximum accuracy value. Accuracy is a measure of how model predictions are close to the true labels in all classes, while loss, a cost function, is an indicator of the correctness of the predictions (Sammut & Webb 2017). Specifically, in zero-one loss, 0 and 1 refer to correct and incorrect classifications, respectively. To achieve the best result based on the corresponding values, we employ a total of 132 different convolutional neural network architectures (Fig. 3) with three different learning rate values (10^{-3}, 10^{-4}, 10^{-5})

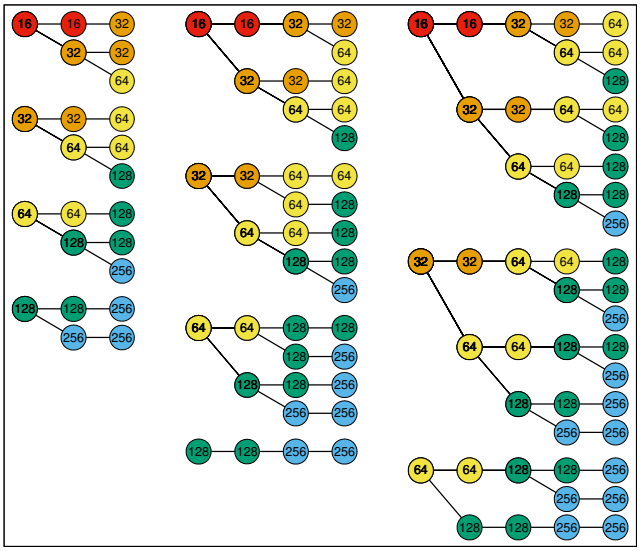

Figure 3. Schematic representation of 44 different architectures with their filter numbers which are employed with three specific learning rate values (10^{-3}, 10^{-4}, 10^{-5}) separately, and correspond to 132 different networks.

which were run using NVidia T4 GPU accelerator provided by Kaggle platform. Although the higher accuracy values were reached in some other architectures, the optimum result was achieved in the network of 5 convolutional layers having 32, 32, 64, 128, 256 filters, respectively (see §4). The final accuracy and loss values for the models with validation accuracies larger than 0.9 are represented in Fig. 4. Models with the learning rate value of 10^{-3} are not included in the figure, since their validation accuracy never exceeded 0.9.

4 Results and Conclusion

An accuracy of 92% was achieved in architecture with 5 convolutional layers having 32, 32, 64,128 and 256 filters. The learning rate was 10^{-4}, and it takes 142 epochs for the architecture to achieve the result, whose training loss (0.233) is slightly lower than the validation loss (0.257), and the training and validation accuracies (0.937 and 0.936) are close. Thus, the model can be considered prevented from overfitting and underfitting compared to other models shown in Fig. 4. A visualisation of the final architecture is shown in Fig. 5. The learning curves, accuracy and loss values for both training and validation versus epoch, are plotted in Fig. 6. The trends of the curves imply a typical good fit. The Keras model file containing the final architecture is provided through a GitHub repository.

A plot of the confusion matrix (Fig. 7) for the validation dataset reveals the details of the classification result. 217 of contact, 228 of detached and 201 of semi-detached systems out of a total of 705 were correctly classified. The maximum misclassification was seen in semi-detached binaries; 22 of them are classified as detached systems. Selected light curves among the best and the worst classified data for each of the morphological classes are given in Fig. 8. True positive (TP), true negative (TN), false positive (FP) and false negative (FN) values calculated based on the confusion matrix are listed in Table 1.

Our training data combines morphological classifications

Table 1. True positive (TP), true negative (TN), false positive (FP) and false negative (FN) values of classification for validation dataset covering 705 light curves.

	TP	*TN*	*FP*	*FN*
C	217	458	12	18
D	228	448	22	7
SD	201	445	25	34

Table 2. Classification report. C, D and SD refer to contact, detached and semi-detached systems, respectively.

	Precision	Recall	F1 score	number of data
C	0.948	0.923	0.935	235
D	0.912	0.970	0.940	235
SD	0.889	0.855	0.872	235
Average	0.916	0.916	0.916	705

derived from different methodological approaches. The Kepler dataset uses the morphology parameter (c), ASAS classifications are based on Fourier coefficient analysis and the classes of CALEB database appear to be based on detailed individual analyses. This heterogeneity in classification methods could potentially introduce inconsistencies in our training labels. However, we suggest that our deep learning approach may benefit from this diversity in several ways. First, by training on data classified through different methods, our model may learn more robust and generalized morphological features. Second, the model's 92% accuracy on this diverse validation set indicates that it has successfully captured the core characteristics of light curves that are consistent across various classification schemes. Nevertheless, we recognize that future work should investigate the impact of this heterogeneity more systematically, perhaps by training separate models on each dataset. Such analysis would help quantify any systematic biases and could inform strategies for harmonizing morphological classifications across different surveys.

The classification report, indicating metrics for the classification of the validation dataset, is shown in Table 2. Precision is the ratio of true positives to the total number of true and false positives (TP/(TP+FP)), a measure of how trustworthy the model is in predicting positive samples (Ting 2010). Recall is defined as the ratio of the number of correctly classified positives to the total number of positives, TP/(TP+FN), and it focuses on positive samples. F1 score is the harmonic mean of precision and recall. In addition to these metrics, the subset accuracy of the classification, calculated using `accuracy_score` function of `scikitlearn` library (Pedregosa et al. 2011), is 92%. This is simply the percentage of correctly classified samples (Tsoumakas & Vlahavas 2007):

$$\frac{1}{|D|}\sum_{i=1}^{|D|} I(Z_i = Y_i) \qquad (3)$$

where Y_i and Z_i are actual and predicted labels, while $|D|$ is the number of multilabel examples and I takes the value of 0 or 1 for false or true statements, respectively.

An important astrophysical consideration in our classification results is the O'Connell effect, where stellar spots or other surface inhomogeneities cause unequal heights

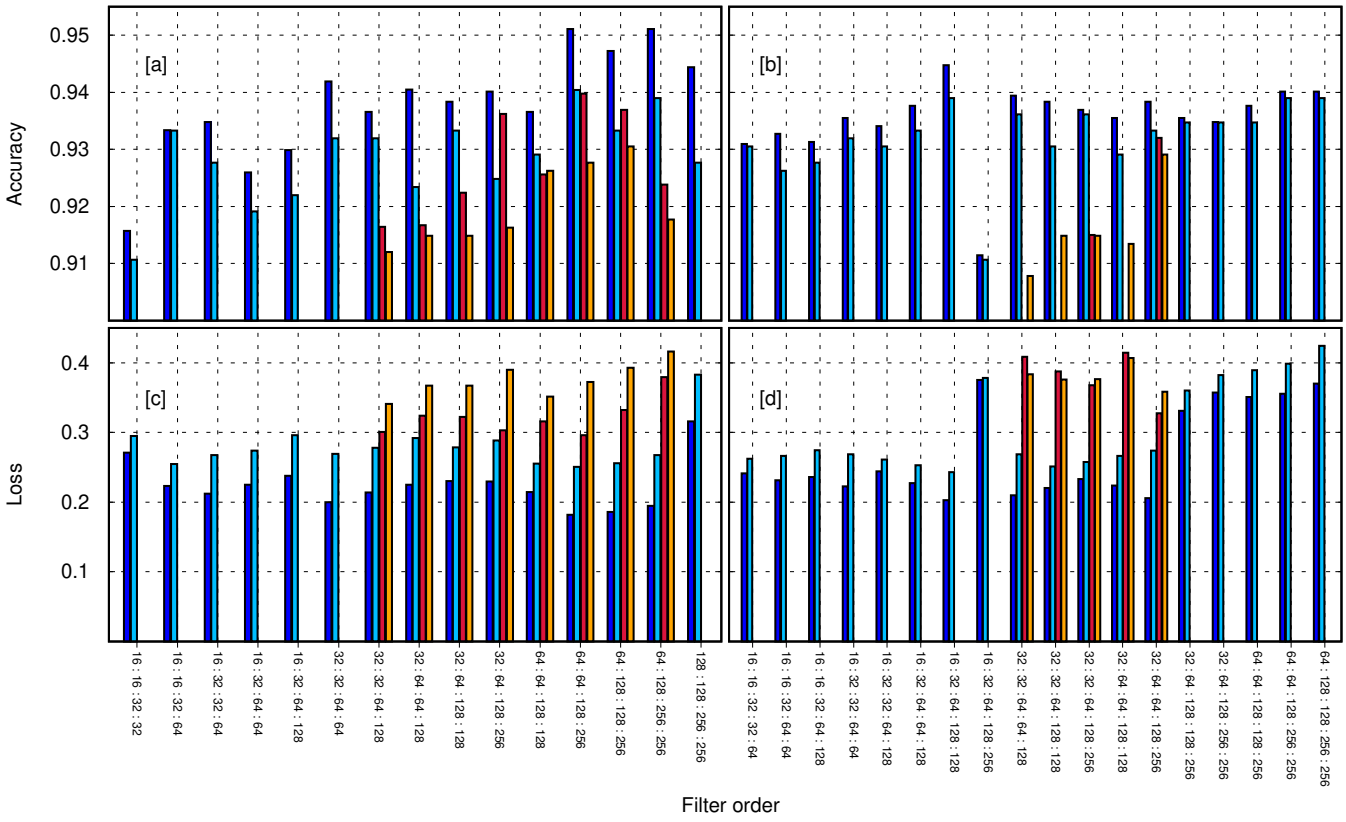

Figure 4. Final Accuracy (a and b) and Loss (c and d) values for different architectures whose validation accuracies are larger than 0.9. Filter orders refer to the filter numbers of convolutional layers. Blue and light blue refer to the training and validation datasets with the learning rate value of 10^{-4}. Red and orange bars represent the training and validation results when the learning rate was set to 10^{-5}.

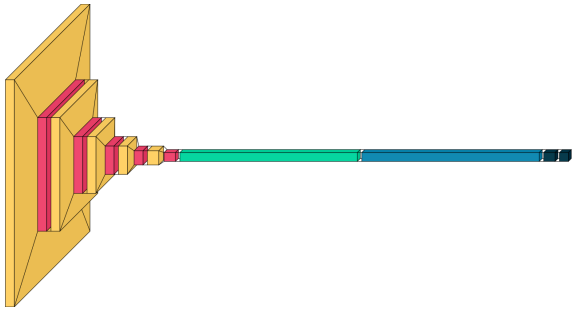

Figure 5. Visualisation of the final neural network architecture. Orange, red, green, blue and black colours refer to Convolutional, Max pooling, Flatten, Dropout and Dense layers. Figure created using `visualkeras` for Keras/TensorFlow (Gavrikov 2020).

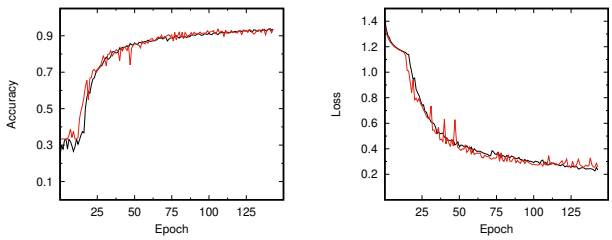

Figure 6. Learning curves, variation of training (black) and validation (red) accuracy and loss with epoch.

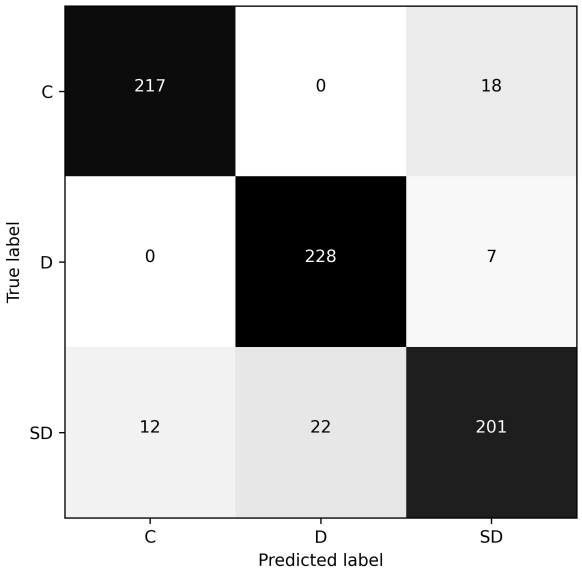

Figure 7. Confusion matrix for validation dataset (705 images) as obtained by using *metrics* module of Scikit-learn (Pedregosa et al. 2011) based on Keras model file containing the final network architecture. C, D and SD refer to contact, detached and semi-detached morphologies, respectively.

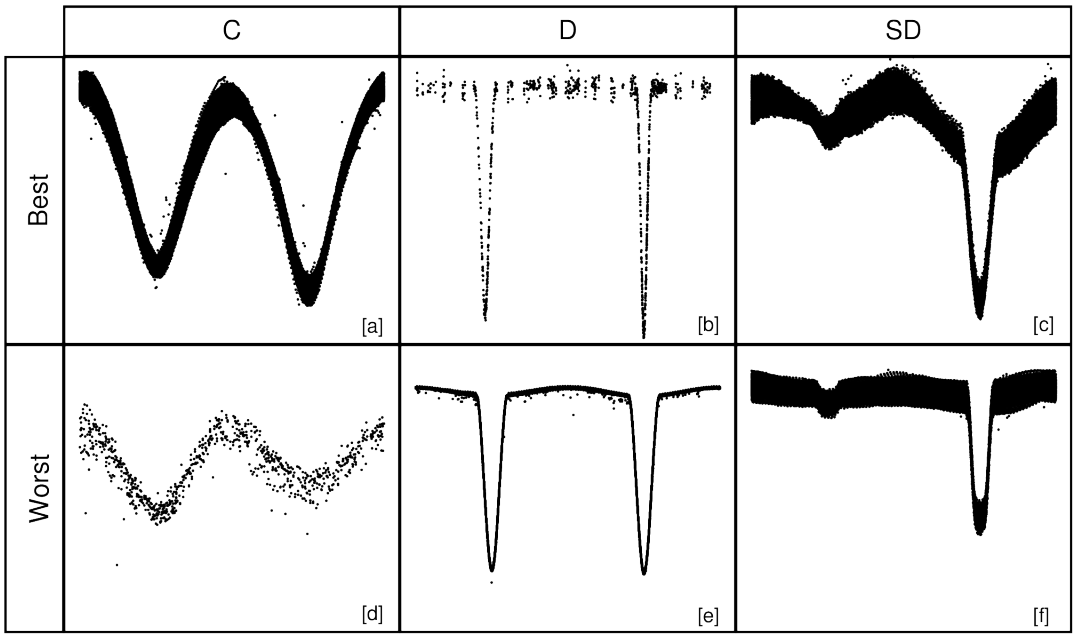

Figure 8. Selection from the best and the worst classified light curves with their estimated morphological classes. The algorithm correctly classified KIC 10723143 (a), BW Aqr (b) and KIC 06852488 (c) with the probability of 99.9% as contact, detached and semi-detached, respectively. The semi-detached binaries ASAS 125523-7322.2 (d) and KIC 10191056 (e) were estimated as contact and detached systems with 99.9% probability. KIC 03530668 (e) was also misclassified as 93.1% semi-detached, while its actual class is detached. C, D and SD refer to contact, detached and semi-detached morphologies, respectively.

of the light curve maxima. The effect can make semi-detached systems appear more similar to contact binaries, as both may exhibit asymmetric light curves with unequal maxima. This effect likely contributes to some of the misclassifications observed in our confusion matrix (Fig. 7). Notably, 12 semi-detached systems were classified as contact binaries, and 22 were classified as detached systems. The former misclassification could be attributed to semi-detached systems with significant spot activity producing light curves that resemble the continuous variations seen in contact binaries, while the latter might occur when spots reduce the apparent depth of one minimum, making the system appear more detached. While our CNN model has become somewhat robust to these variations through exposure to diverse light curves in the training set, the O'Connell effect remains a fundamental challenge for morphological classification based solely on light curve shape. Future improvements to our approach could include incorporating light curves from multiple photometric bands to help distinguish wavelength-dependent spot-induced asymmetries from geometric effects. Training on light curves from different time intervals to help the model learn to distinguish transient spot effects from stable morphological features. The presence of the O'Connell effect underscores the inherent ambiguity in morphological classification based on photometry alone and highlights the value of spectroscopic follow-up for definitive classification of eclipsing binary systems.

Furthermore, it is worth looking up the filters and the output of convolutional layers (feature maps) of the final architecture in terms of how the machine sees and processes the light curves through the network. As an example, we demonstrate the feature maps for the light curve of KIC 03954798 as proceeded along with the filters of the first and the fourth convolutional layers in Fig. 9 and Fig. 10. For the human eye, the deeper the layer is, the harder the light curve perception is.

Besides fixing the seeds in our code for reproducibility, following the 2.0 version of the reproducibility checklist for machine learning given by Pineau et al. (2020), we addressed the details of our model in the previous section. The algorithm was explained in detail with necessary mathematical descriptions. The application platform and infrastructure used were also denoted. The sample size of the data was given, the number of examples in the training and validation sets was specified, and the data preparation process was denoted (§2). The dataset and the code executing the classification are downloadable. We defined and provided the metrics of the classification and indicated the classification report, which refers to the quality of the classification.

The scientific importance of our neural network algorithm arises from its capacity to provide a means to distinguish the morphological types of eclipsing binary systems with high accuracy, only using their light curve images. It is also significantly faster than other conventional methods performing the same process, such as workflows that involve testing at least two morphological models using widely known light curve analysis software (e.g., `PHOEBE` (Prša & Zwitter 2005) which is based on the Wilson-Devinney method (Wilson & Devinney 1971); `JKTEBOP` (Southworth et al. 2005) based on `EBOP` of Popper & Etzel (1981); or `ELISa` which is written by Čokina et al. 2021b), and then comparing the results to select the best fit. The determination of the morphological class is vital in the analysis of an eclipsing binary light curve in order to yield physically meaningful results, therefore, our algorithm can be applied to a light curve image before its analysis to establish a

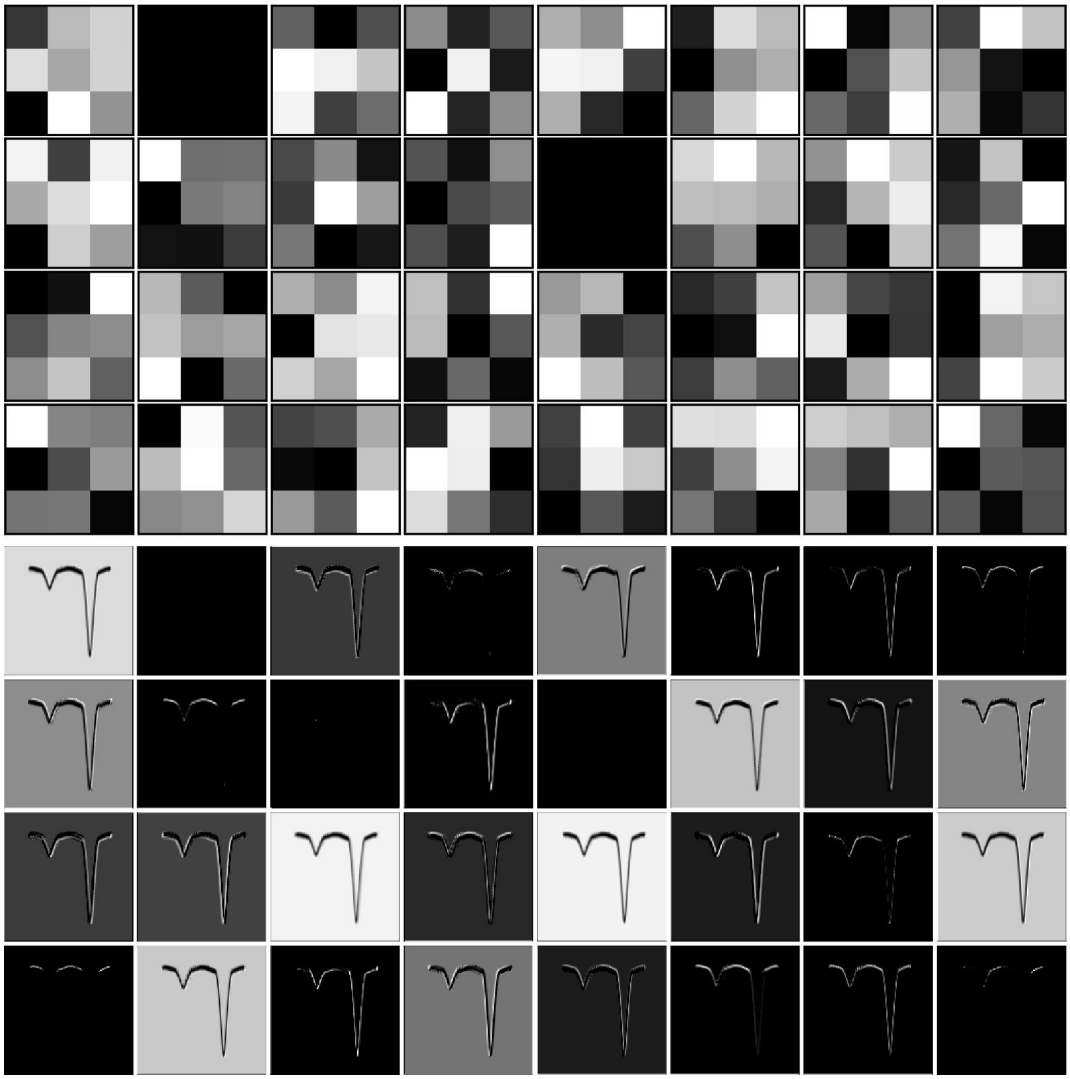

Figure 9. 32 (3×3) filters of the first convolutional layer (upper panel) and 32 feature maps of the light curve image of KIC 03954798 as output from the first convolutional layer with corresponding filters (lower panel). Figure created using `Matplotlib` (Hunter 2007).

rapid and reliable morphological assumption for the light curve solution.

When it comes to comparing our results to other studies using machine learning algorithms related to the morphologies, our accuracy is found to be close to that of investigations in the literature. Although they were not to deal with the morphology alone, the accuracy in the three-layer artificial neural network by Prša et al. (2008), which focused on detached morphological classification, was higher than 90%. An image classification algorithm proposed by Ulaş (2020) was also reached an accuracy value of 91%. Čokina et al. (2021a) achieved 98% accuracy through their combined classifier, which was trained with synthetic light curve data constructed using `ELISa` software for detached and overcontact morphologies. Finally, our results are comparable to the performance reported by Parimucha et al. (2024), who achieved an accuracy of 94% based on two classes. We did not run our code using the images generated from the light curve of the above-mentioned studies, since a classification owes its resulting accuracy to properly collected training data as well as the architecture. A complete change in the training set most probably requires modification in the network architecture and hyperparameters to achieve the same accuracy, over 90%.

The accurate information on the classes of training samples plays a vital role in the quality of the results. Therefore, in a future study, we plan to improve our algorithm by collecting light curve images with more accurate information on their types, namely the light curves of the systems having the morphological classes determined by analyses through human-controlled software, since hands-on modelling is the finest approach as Kochoska et al. (2020) concluded. This is projected to be done by a detailed survey of the literature for individual analyses of eclipsing binary light curves. Thuswise, the volume of training and validation samples, another crucial parameter, is also aimed to be increased. The increasing volume of space telescope observations of binary stars provides a growing pool of potential training data; however, the utility of these data for machine learning applications depends critically on accurate morphological classification of each system. Finally, our code and collected data are public, therefore, it is open

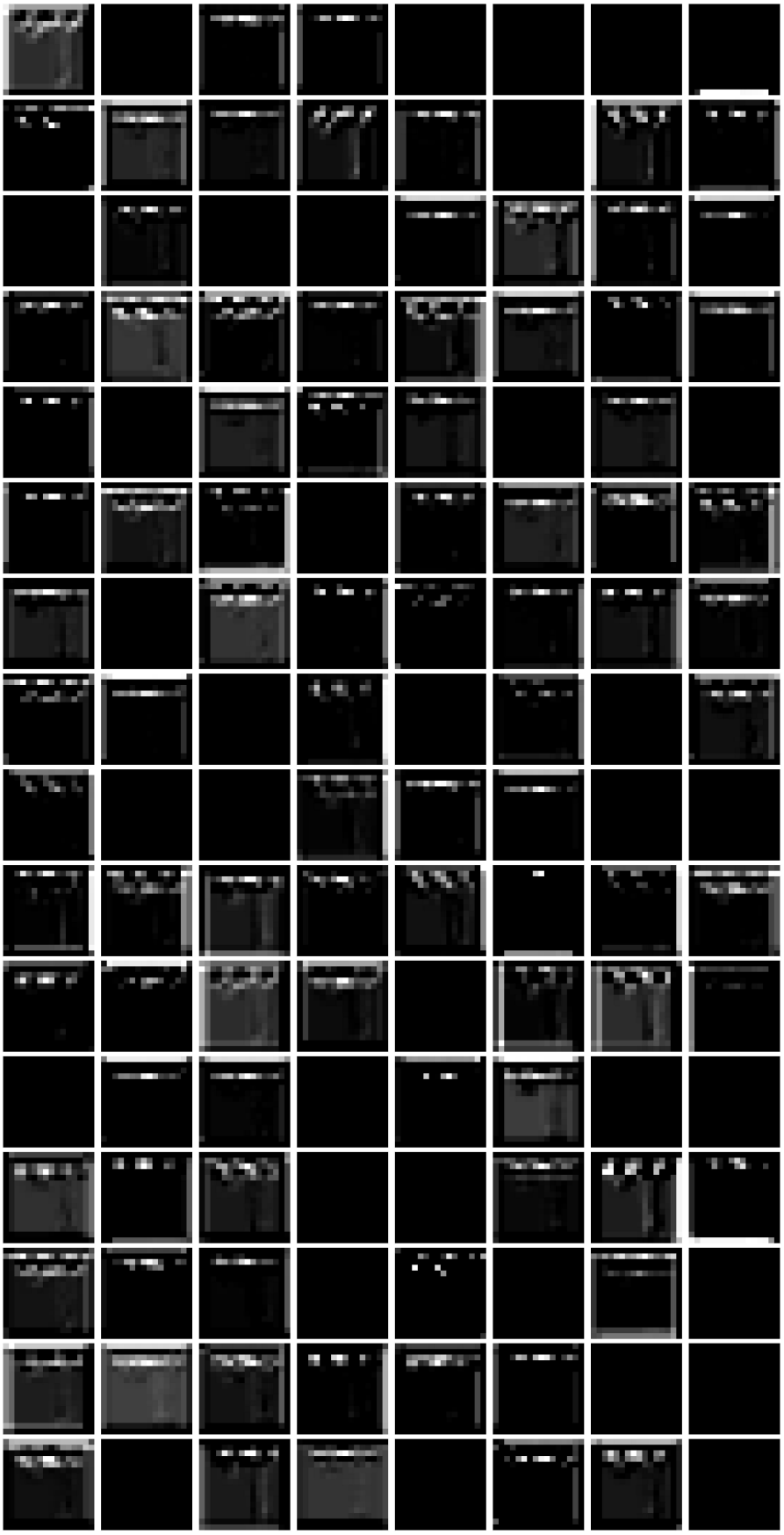

Figure 10. Same as the lower panel of Fig. 9, but for the fourth convolutional layer with 128 filters. Note that human perception for the light curve is almost lost.

to be improved by tuning the hyperparameters or altering the architecture.

References

Abadi M., et al., 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, https://www.tensorflow.org/

Basha S. S., Dubey S. R., Pulabaigari V., Mukherjee S., 2020, Neurocomputing, 378, 112

Birky J., Davenport J., Brandt T., 2020, in American Astronomical Society Meeting Abstracts #235. p. 170.20

Bódi A., Hajdu T., 2021, ApJS, 255, 1

Bradstreet D. H., 2005, Society for Astronomical Sciences Annual Symposium, 24, 23, ADS

Bradstreet D. H., Steelman D. P., Sanders S. J., Hargis J. R., 2004, in American Astronomical Society Meeting Abstracts #204. p. 05.01

Chollet F., et al., 2015, Keras, https://keras.io

Christlein V., Spranger L., Seuret M., Nicolaou A., Král P., Maier A., 2019, in 2019 International Conference on Document Analysis and Recognition (ICDAR). pp 1090–1096, doi:10.1109/ICDAR.2019.00177

Čokina M., Maslej-Krešň'akov'a V., Butka P., Parimucha Š., 2021a, Astronomy and Computing, p. 100488

Čokina M., Fedurco M., Parimucha Š., 2021b, A&A, 652, A156

Cortes C., Mohri M., Rostamizadeh A., 2009, in Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. UAI '09. AUAI Press, Arlington, Virginia, USA, p. 109–116

Dai Z., Liu H., Le Q. V., Tan M., 2021, preprint, , ADS

Fukushima K., 1980, Biological Cybernetics, 36, 193

Gavrikov P., 2020, visualkeras, https://github.com/paulgavrikov/visualkeras

Géron A., 2017, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1st edn. O'Reilly Media, Inc., p. 357

Goodfellow I. J., Bengio Y., Courville A., 2016, Deep Learning. MIT Press, Cambridge, MA, USA, p. 171

Guinan E. F., 1993, in Leung K.-C., Nha I.-S., eds, Astronomical Society of the Pacific Conference Series Vol. 38, New Frontiers in Binary Star Research. p. 1

Harris D., Harris S., 2007, Digital Design and Computer Architecture. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, p. 82

Harris C. R., et al., 2020, Nature, 585, 357

Healy B. F., et al., 2024, ApJS, 272, 14

Hunter J. D., 2007, Computing in Science & Engineering, 9, 90

Jenkins J. M., et al., 2010, ApJ, 713, L87

Juran J. M., Godfrey A. B., 1999, Juran's quality handbook. 5e, McGraw Hill, http://books.google.de/books?id=beVTAAAAMAAJ

Kingma D. P., Ba J., 2015, in Bengio Y., LeCun Y., eds, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings. p. 1, http://arxiv.org/abs/1412.6980

Kirk B., et al., 2016, The Astronomical Journal, 151, 68

Kochoska A., Conroy K., Hambleton K., Prša A., 2020, Contributions of the Astronomical Observatory Skalnaté Pleso, 50

Kopal Z., 1955, Annales d'Astrophysique, 18, 379, ADS

Krizhevsky A., Sutskever I., Hinton G. E., 2012, in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12. Curran Associates Inc., Red Hook, NY, USA, p. 1097–1105

Lecun Y., Bottou L., Bengio Y., Haffner P., 1998, Proceedings of the IEEE, 86, 2278

Matijevič G., Prša A., Orosz J. A., Welsh W. F., Bloemen S., Barclay T., 2012, The Astronomical Journal, 143, 123

Moore H., 1897, The ANNALS of the American Academy of Political and Social Science, 9, 128

Murphy K. P., 2012, Machine Learning. MIT Press, London, England, p. 247

Parimucha Š., Gajdoš P., Markus Y., Kudak V., 2024, Contributions of the Astronomical Observatory Skalnate Pleso, 54, 167

Pedregosa F., et al., 2011, Journal of Machine Learning Research, 12, 2825

Pineau J., Vincent-Lamarre P., Sinha K., Larivière V., Beygelzimer A., d'Alché-Buc F., Fox E., Larochelle H., 2020, preprint, (arXiv:2003.12206), ADS

Pojmanski G., 1997, Acta Astronomica, 47, 467, ADS

Pojmanski G., 2002, AcA, 52, 397, ADS

Popper D. M., Etzel P. B., 1981, AJ, 86, 102

Prsa A., Wrona M., 2024, in American Astronomical Society Meeting Abstracts. p. 423.07

Prša A., Guinan E. F., Devinney E. J., DeGeorge M., Bradstreet D. H., Giammarco J. M., Alcock C. R., Engle S. G., 2008, The Astrophysical Journal, 687, 542

Prša A., Zwitter T., 2005, ApJ, 628, 426

Ricker G. R., et al., 2015, Journal of Astronomical Telescopes, Instruments, and Systems, 1, 014003

Ruder S., 2016, ArXiv, abs/1609.04747

Sammut C., Webb G. I., eds, 2017, Loss. Springer US, Boston, MA, p. 781, doi:10.1007/978-1-4899-7687-1-499

Shorten C., Khoshgoftaar T., 2019, Journal of Big Data, 6, 60

Southworth J., Smalley B., Maxted P. F. L., Claret A., Etzel P. B., 2005, MNRAS, 363, 529

Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, Journal of Machine Learning Research, 15, 1929

Szklenár T., Bódi A., Tarczay-Nehéz D., Vida K., Mező G., Szabó R., 2022, ApJ, 938, 37

The Pandas Development Team 2020, pandas-dev/pandas: Pandas, doi:10.5281/zenodo.3509134, https://doi.org/10.5281/zenodo.3509134

Ting K. M., 2010, in Sammut C., Webb G. I., eds, , Encyclopedia of Machine Learning. Springer US, Boston, MA, p. 781, doi:10.1007/978-0-387-30164-8_652, https://doi.org/10.1007/978-0-387-30164-8_652

Tsoumakas G., Vlahavas I., 2007, in Kok J. N., Koronacki J., Mantaras R. L. d., Matwin S., Mladenič D., Skowron A., eds, Machine Learning: ECML 2007. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 406–417

Udalski A., Szymanski M., Kubiak M., Pietrzynski G., Wozniak P., Zebrun K., 1998, Acta Astron., 48, 147, ADS

Ulaş B., 2020, preprint, , ADS

Van Rossum G., 2020, The Python Library Reference, release 3.8.2. Python Software Foundation

Van Rossum G., Drake F. L., 2009, Python 3 Reference Manual. CreateSpace, Scotts Valley, CA

Wes McKinney 2010, in Stéfan van der Walt Jarrod Millman eds, Proceedings of the 9th Python in Science Conference. pp 56 – 61, doi:10.25080/Majora-92bf1922-00a

Wilson R. E., Devinney E. J., 1971, ApJ, 166, 605

Wyrzykowski L., et al., 2003, Acta Astron., 53, 1, ADS

Yao Y., Rosasco L., Caponnetto A., 2007, Constr. Approx, pp 289–315

Zhou Z., Huang H., Fang B., 2021, Journal of Computer and Communications, 09, 1

Access:

M25-0103: Turkish J.A&A — Vol.6, Issue 1.