Classifying the Source Radius from Microlensing Light Curves using Neural Networks

Soumya Shreeram
This project was supervised by Martin Millon, Eric Paic, and Prof. Frédéric Courbin.

Practical work (TP4b) with *Laboratory of Astrophysics,*
École Polytechnique Fédérale de Lausanne, Route Cantonale, 1015 Lausanne
June 5, 2020

With the dawn of all-sky surveys and time domain astronomy, which are expected to discover thousands of lensed quasars in the coming decade, deep learning provides one of the most feasible techniques to analyze microlensed light-curves. This project aims to classify quasar source sizes into ten categories using simulated microlensing light curves, which closely mimic the COSMOGRAIL light curves in terms of photometric noise, seasonal visibility and duration of the monitoring campaign. This classification is brought about by using three deep learning algorithms: a Convolutional Neural Network (CNN), and two Residual Neural Networks (ResNet-7 and ResNet-18). It was found that the ResNets outperformed the CNN by $> 10\%$. The models were able to best classify the light-curves when the season gaps were fitted using linear interpolation, as opposed to the curves fitted using Gaussian processes regression. Using linearly interpolated light curves, ResNet-7 achieved 63.4% accuracy, while ResNet-18 achieved 66.1% accuracy.

Contents

Email: soumya.shreeram@epfl.ch, GitHub repository link.

I Introduction

The deflection of light rays from a distant source, due to the gravitational field of a massive object, in our line of sight, is called the gravitational lensing effect. Einstein (1911) was among the pioneers to show that gravitational lensing was a measurable effect. After his work on general relativity in 1915, he proposed that the deflection angle, α, which a light ray experiences when passing by a massive body, M, can be given by the formula:

$$\alpha = \frac{4GM}{c^2\xi}, \tag{1}$$

where G is the gravitational constant, c the speed of light, and ξ the distance of the light ray from the centre of the massive body, also known as the impact parameter. He predicted that a light ray, originating from fixed stars behind the sun, which travel past the sun's limb towards us would experience a deflection of 1.75 arc seconds. This phenomenal claim was confirmed during the total solar eclipse on $29th$ May 1919 by Dyson et al. and Eddington. Moving further, Zwicky (1937) highlighted the importance of observing gravitationally lensed galaxies for three major reasons: the ability to detect further objects in the sky, probes to test the theory of general relativity, and aid in determining the galaxy's mass distributions. The following years witnessed an uprise in the theoretical research and observations of lensing events (Refsdal & Bondi, 1964; Liebes Jr, 1964; Dominik, 2011).

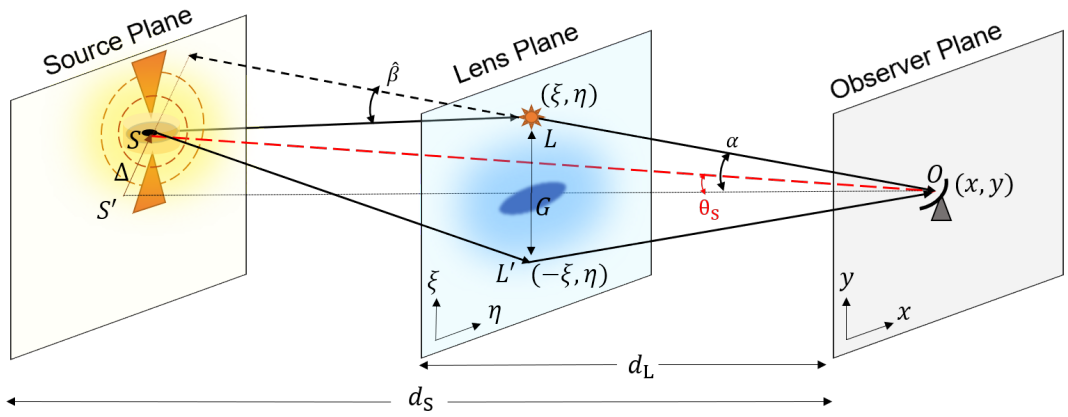

Fig. 1: A pictorial representation of the path taken by light rays emitted from a source S, in this case: a QSO, which makes an unlensed angle θ_S with respect to an observer O on earth. The apparent angle due to gravitational lensing on the observer's plane is denoted as α, while that at the lens plane is denoted as $\hat{\beta}$. The distance to the source and lens plane are d_S and d_L respectively. The lens plane (ξ,η), the observer plane (x,y), and the source plane are normal to the line connecting them $S'O$, which is called the optical axis. For convenience, the origin of the lens plane $(\xi,\eta)=(0,0)$ is chosen to coincide with the large deflecting galaxy G; similarly, O is chosen to be at the origin of the (x,y) plane. SLO corresponds to the path taken by the light ray when it is deflected by both, the galaxy G and the star L. However, $SL'O$ represents the light ray that is deflected only due to the influence of G, i.e. in the absence of microlensing.

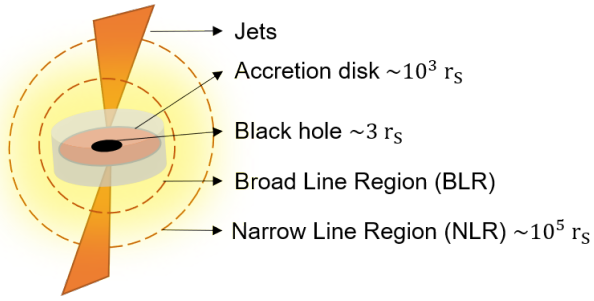

Fig. 2: Illustration the structure of a QSO[a]. It contains a massive black hole at the centre that is surrounded by a thin, optically thick accreting disk of matter and gas extending upto 10^3 r_S[b]. Moreover, there are two long jets of gas shot out in directions normal to the disk plane whose exact geometry is yet debated (Galeev et al., 1979; Miyamoto & Kitamoto, 1991). The inner circular dashed line denotes the region where matter emits lines that are broadened due to strong relativistic effects owing to the high-angular momentum close to the black hole (Laor, 1991), whereas, regions farther from the centre within 10^5 r_S (*outer circular dashed line*) emit narrower emission lines.

[a]Figure inspired from Eric Paic's Master thesis and Aymeric Galan.
[b]The Schwarzschild radius of an object of mass M is given as, $r_s = 2GM/c^2$.

The scenario where the light from a luminous *source* S passes on either side of a *moving lens* L, like a star, and gives rise to a transient brightening of the S in the observers' plane, is called gravitational *microlensing*. It must be noted that the microlensing occurs only when there is macro-lensing, due to a massive object like a galaxy; both these phenomena are illustrated in Fig. 1. The first evidence for microlensing was suggested by Chang & Refsdal (1979) due to the flux variations in the multiply-split images of the quasi-stellar object (QSO) $0957+561$, which was initially observed by Walsh et al. (1979). They claimed that this could be because of the stars deflecting the light in the sparse outer regions of the massive galaxy. Moreover, it was highlighted that the microlensed light curve provides a pathway to decode physical information of the background QSO such as the size and structure of the QSO (see Fig. 2), the ratio of the mass of smooth dark matter to the compact masses in the outer regions of galaxies, studying the foreground motion of stars in the lens plane using time-domain astronomy, relative velocities of the components, and last but not the least, the shear and surface density of the lens galaxy. As an additional advantage, microlensing has also paved the way to discover extra-solar planetary systems (Mao & Paczynski, 1991; Ingrosso et al., 2011).

A handful of approaches have been attempted over the past two decades to obtain physical information about quasars. More specifically, in this project, we are interested in estimating the disk size of the background quasar using microlensing. Kochanek (2004) used the *Bayesian Monte Carlo* (BMC) method that uses a ray-shooting approach (Kayser et al., 1986) to generate a model that mimicked the time-series observations of the microlensed light curves. This method involved producing numerous random realizations of the light curves, using generalized stellar configurations, until

at least one successfully reproduced the data. The best light curve is analysed using the goodness of fit statistics. This method has been successfully applied to multiple quasars over the later years (Morgan et al., 2008, 2012, 2018). The major disadvantage in the BMC method arises from its inability to catch short term variations that could potentially overlook some of the high-frequency microlensing signals (Mosquera & Kochanek, 2011). Another important study uses the flux ratio anomalies in multi-wavelengths, λ, due to microlensing, to estimate the source size $R_{\rm S}$ (Pooley et al., 2007; Blackburne et al., 2011; Floyd et al., 2009). The source size follows a power-law $R_{\rm S} \propto \lambda^{\zeta}$, where the power-law index $\zeta = 4/3$, for a thin accretion disk model (Shakura & Sunyaev, 1973). The major limitation in this method arises due to the strong dependence on the prior parameters (Morgan et al., 2018). Discrepancies between the results for $R_{\rm S}$ in the former two methods have motivated the invention of different methods. Other numerous studies include narrowband photometry (Mosquera et al., 2011), spectroscopic monitoring (Eigenbrod et al., 2008), using the power spectrum of light curves (Eric Paic et al. in prep.), and using deep learning (Vernardos & Tsagkatakis, 2019).

In this project, the aim is to classify microlensed light-curves for a QSO into different accretion disk sizes, using deep learning. The theoretical preliminaries for quasar lensing are introduced in Sec. II, and the basic concepts in microlensing is discussed in Sec. II-A. Sec. III discusses the implementation of three deep learning algorithms: a Convolutional Neural Network (CNN; Sec. III-A), and two Residual Neural Networks (ResNets-7 and ResNet-18; Sec. III-B). The technical details of these models are given in Sec. III-C. Sec. IV details on the generation of the mock light curves that are used for deep learning in this project. These simulated light-curves attempt to mimic the observed microlensed light curves by accounting for season gaps. Sec. IV-A motivates the reason for interpolating the light curves; two distinct methods are attempted for interpolation: a simple linear interpolation (Sec. IV-A1), and a Gaussian processes regression (Sec. IV-A2). Finally, Sec. V summarizes the performances of the deep learning models (Sec. V-A), and gives an account of the classification accuracy for the following cases of light-curves: (A) without season gaps, (B) with season gaps, (C) with gaps fitted using linear interpolations, and lastly, (D) with gaps fitted using Gaussian processes regression (Sec. V-B).

II Quasar Lensing

There exist three different regimes of lensing: strong lensing, weak lensing, and microlensing. Quasars are affected by both, strong lensing (due to the lens galaxy) as well as the weak regimes of microlensing (due to the stars in the lens galaxy). Although, our topic of interest lies in the latter case of microlensing, as further discussed in Sec. II-A, the former two cases are briefly introduced as follows:

- **Strong lensing** occurs when the light rays of the quasar are distorted by a foreground mass distribution, like a galaxy, which is sufficient enough to produce multiple images or arcs; arcs are a consequence of close alignment between the source and the lens along the line of sight. For the case of perfect alignment, when $\theta_{\rm S} = 0$, the arcs form a complete ring, so-called as the Einstein ring $\theta_{\rm E}$, which is given as:

$$\theta_{\rm E} = \sqrt{\frac{4GM}{c^2}\frac{d_{\rm S} - d_{\rm L}}{d_{\rm L}}}, \tag{2}$$

  where M is the mass enclosed by the ring, $d_{\rm L}$ and $d_{\rm S}$ are the distances introduced in Fig. 1 (Schneider et al., 2006, Ch.1, p. 33). $\theta_{\rm E}$ is the characteristic scale for observing multiple images in lensing phenomena.
- **Weak lensing** occurs when the angular distances between the lenses are larger compared to the angular sizes θ_E of each individual lens (Limousin et al., 2005, p. 3). This results in weak distortions and small magnifications (Schneider et al., 2006, Ch.3, p. 269).

The **lens equation** relates the true position of the background source, $\theta_{\rm S}$, with the observed position, α. As shown in Fig. 1, $SS' := \Delta$ denotes the position of the quasar on the two-dimensional source plane, which is defined with respect to the optical axis that intersects the plane perpendicularly. Using the small-angle approximation, the deflection of light rays at the lens plane is given as $\hat{\beta} \sim \sin\hat{\beta} \sim \tan\hat{\beta}$, thereby the lens equation becomes

$$\theta_{\rm S} = \alpha - \hat{\beta}\frac{d_{\rm S} - d_{\rm l}}{d_{\rm S}} = \alpha - \beta(\alpha), \tag{3}$$

where $\beta(\alpha)$ is the *scaled deflection angle* (Schneider et al., 2006, p. 21). It is useful to note that the scaled deflection can be related to the *convergence*[1] κ, and

[1] κ is also called the dimensionless surface mass density.

hence, to the critical surface mass density $\Sigma_{\rm cr}$ using the following relations:

$$\beta(\alpha) = \frac{1}{\pi}\int_{\mathbb{R}} \mathrm{d}^2\alpha' \kappa(\alpha') \frac{\alpha - \alpha'}{|\alpha - \alpha'|^2}, \tag{4}$$

$$\text{where } \kappa(\alpha) := \frac{\Sigma(d_{\rm L}\alpha)}{\Sigma_{\rm cr}}. \tag{5}$$

κ will be revisited while further understanding the concept of magnification. The non-linear mapping in Eq. 3, from $\alpha \rightarrow \theta_{\rm S}$ or in other words, from the lens to the source plane, can be calculated assuming any suitable mass distribution in the lens plane $\Sigma(\xi)$[2]. However, the inverse mapping is non-trivial and requires one to introduce the *deflecting potential* $\psi(\alpha)$, such that $\beta(\alpha) = \nabla\psi$. Therefore, using Eq. 4, ψ satisfies the two dimensional Poisson equation for a source κ, which is given as,

$$\nabla^2\psi(\alpha) = 2\kappa. \tag{6}$$

The aim of introducing this machinery is to emphasize that the solution to the lens equation results in the knowledge of the angular positions of the source in the image or lens plane. As a result, the number of solutions to the lens equation corresponds to the number of images of the source that can be observed on the image plane.

The change of the angular size and shapes of objects in the sky is given by the concept of **magnification**. For an unresolved and small source, the fluxes of the image and unlensed source are defined as the integrals of their brightness distributions, $\mathbf{I}(\alpha)$ and $\mathbf{I}(\theta_{\rm S})$, respectively. The ratio of these two fluxes is called the magnification μ,

$$\mu = [(1-\kappa)^2 - \gamma^2]^{-1}, \tag{7}$$

where γ is called the shear that is responsible for changing the shape of the image (Courbin & Minniti, 2002, p. 6). For a mass distribution with $\kappa \geq 1$, multiple images are produced; thus, κ is a parameter that distinguishes between the weak and strong lensing regime. Since the absolute flux of the unlensed source is unknown, the ratio of fluxes from various images provides a direct way to measure the magnification ratio. This paves the way to create a pixelled *magnification map* representing the magnification in the source plane, due to the foreground stellar distribution that acts as microlenses. An example map is shown in Fig. 3.

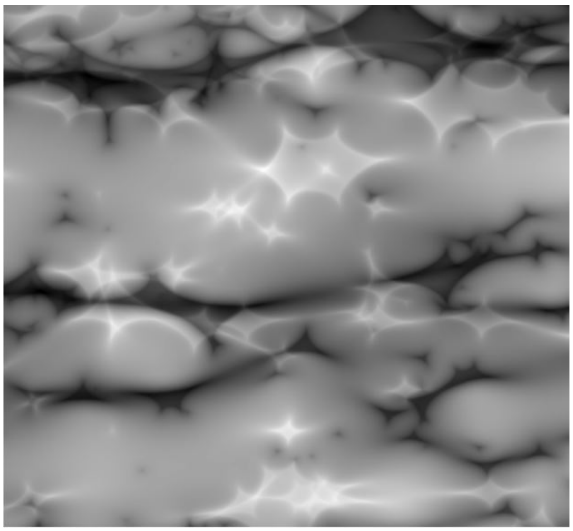

Fig. 3: A simulated 8192×8192 pixelled magnification map of $Q\ J0158-4325$ after taking the difference between the A and B images. The corresponding observed size of the map is $(18.75 \times 18.75)\ R_{\rm E}$ with the sharp caustic lines corresponding to the regions of maximum magnification. Note that $R_{\rm E} = \theta_{\rm E} d_{\rm L}$ is the Einstein radius. An expected light curve that can be generated from such a map is shown in Fig. 9. This figure was taken from Eric Paic's master thesis.

II-A Microlensing

Quasar microlensing occurs due to the foreground stellar mass distribution in the range $10^{-6} \leq m/M_{\odot} \leq 10^3$, which affects the surface brightness of the quasar images. These fluctuations in the surface brightness are variable as a function of time, and they are due to two major causes: intrinsic variability of the quasar, or microlensing. The former cause of fluctuation can be eliminated by observing multiply-imaged quasars; the difference between the light curves of two images at fixed points, say A and B, gets rid of the intrinsic variability, after correcting for time delays. Since the observed pixel amplification is measured in magnitude, the following conversion is used to express the magnification μ in terms of magnitude m:

$$m_{\rm A} - m_{\rm B} = -2.5 \log_{10}\left[\frac{\mu_{\rm A}}{\mu_{\rm B}}\right]. \tag{8}$$

To observe the microlensing signal, the quasar must be monitored over durations lasting from months to years. The characteristic timescale is given by the Einstein time

$$t_{\rm E} = d_{\rm L}\theta_{\rm E}/v_{\perp}, \tag{9}$$

where $v_{\perp}$ is the transverse speed of the lens relative to the optical axis connecting the source

[2] $\Sigma(d_{\rm L}\alpha) = \Sigma(\sqrt{\xi^2+\eta^2})$ from Fig. 1, however, for convenience the case where $\eta = 0$ is considered.

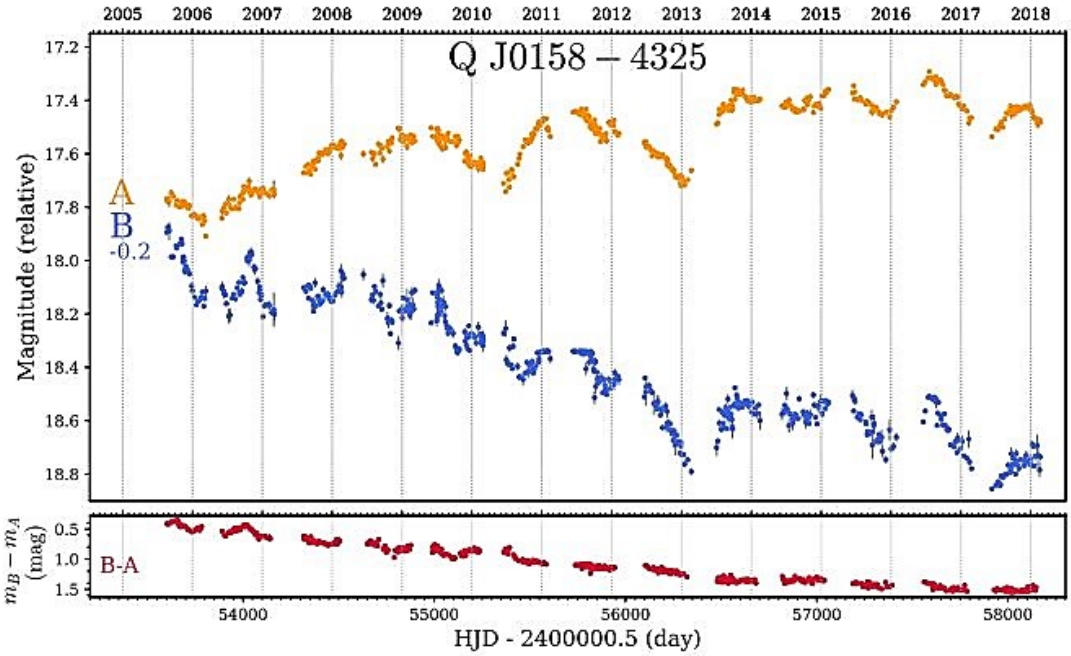

Fig. 4: Light curve generated after 13 years of observations of quasar $Q\ J0158-4325$. Magnitude at images A and B, $m_{\rm A}$ and $m_{\rm B}$, are shown in the top panel. The two imaged are separated by 1.22" on the magnification map and their difference, $m_{\rm B}-m_{\rm A}$ is shown in the bottom panel. Figure taken from Millon et al. (2020).

and observer (Schneider et al., 2006, Ch.4, p. 457). The ingredients needed to describe the microlensing light curve are as follows: $t_{\rm E}$ that contains the physical information about the source, the unlensed flux F_0 that can be measured in the absence of microlensing, t_0 sets an arbitrary time scale, the normalized image position $x \equiv \alpha_0/\theta_{\rm E}$, and the normalized source position $y \equiv \theta_{\rm S}/\theta_{\rm E}$. However, there is a degeneracy between the lens mass M, $d_{\rm L}$, and $v_\perp$ that are required to determine $t_{\rm E}$, as further detailed in (Schneider et al., 2006, Ch.4, p. 454).

The image splitting due to microlensing is not directly observable due to their small angular scales, therefore, the sum of the magnifications of all the micro-images over $t_{\rm E}$ makes microlensing an observable and dynamic phenomenon. An example light curve is shown in Fig. 4 for the two-imaged $Q\ J0158-4325$, hereafter $J0158$. $J0158$ is at redshift $z_{\rm S} = 1.29$ and the lens is located at $z_{\rm L} \approx 0.317$ (Faure et al., 2009; Morgan et al., 1999), with an Einstein radius, $R_E = 3.414 \times 10^{16}$ cm (Mosquera & Kochanek, 2011). This QSO has been monitored over a period of 13 years and it was confirmed that the source is subject to significant microlensing. Further details regarding the time delay corrections can be found in literature (Morgan et al., 2012; Millon et al., 2020). Before providing details on the generation of mock light curves that are used for deep learning, details are given on the models built in this project in the next section.

III Implementation of Deep Learning

A neural network (NN) consists of multiple, connected neurons that are each activated to produce an output. The activation of the neuron is due to the input data or due to a previously activated neuron that is connected to it. Every neuron output is a scalar function $y(\mathbf{x};\mathbf{w})$ that depends on the input vector, $\mathbf{x} = \{x_n\}_{n=1}^{N}$, weight vector, $\mathbf{w}$, and bias vector, θ, as

$$y = \sum_{n=1}^{N}(x_n\ w_n - \theta_n), \tag{10}$$

where the summation index n runs over all the N data points of the input vectors (Bouchain, 2006). The output from every neuron is fed into a non-linear *activation function*, $f(y)$, that generates a continuous number between 0 and 1. Some examples of activation functions are sigmoid, softmax, tanh, or Rectified Linear Unit (ReLU); amongst which ReLu is the most widely used activation function (Lau & Lim, 2017). The *learning* process constitutes the goal of finding the weights $\mathbf{w}$ for every layer of the network such that the NN displays the desired output behaviour. This is achieved by minimizing the cost function E that is expressed as,

$$E = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{11}$$

where i iterates over all N output neurons; y_i and $\hat{y}_i$ are the desired and computed outputs of the neurons respectively (Svozil et al., 1997, p. 45). Minimization

of E is accomplished by calculating the gradient of the multi-variable parameter space. After every run through the entire data set, so-called as an *epoch*, the weights and biases of the entire network are updated with the aid of a *loss function*. This part of the training process is called *back-propagation*.

A particular class of learning for classifying data is called as *Supervised Learning* (SL), which is associated with adjusting $\mathbf{w}$ by comparing the predicted results outputted by the model with the true values that one would expect, known as *test* or *validation* labels. Hence, for SL one would need to provide the NN with *training* and *testing* data sets. The parameterized model is typically trained for multiple epochs until it achieves the highest possible accuracy (or minimum loss) when applied to the arbitrary *test* data set (Schmidhuber, 2015, p. 4). The ultimate aim is to generate a model that can be generalized to other data sets. Generalization means that the model performs well on new and unseen data. A measure to check if the model generalizes well in SL is by applying the model on an unseen data set. An over-trained or *overfitted* model will memorize the training data set and will not generalize well for the new data set (Svozil et al., 1997). Overfitting is a common problem that can be avoided by optimizing model parameters.

In this project, the python programming package used to generate neural networks is *Keras* (Chollet et al., 2015). Three NNs were built to classify the microlensed light curves into 10 categories of source radii. Before discussing the details of the three networks, Sec. III-A and Sec. III-B provides an overview on CNN and ResNet.

III-A Convolutional Neural Network (CNN)

CNN's are built using convolutional layers; the output of the neuron in a layer is a fundamental unit called a *feature map*. Feature maps are results of the convolution of the input image with *filter matrices* or *filters*. Each 2D layer contains multiple feature maps that are produced by convolutions of the input image with the smaller filters that are projected on the previous layer. CNN is built on three novel ideas: invariance under distortions and space translations in the input data sample, shared weights, and locally receptive fields. Locally receptive fields mean that a unit in every layer receives the output from a set of units in a small neighbourhood of the previous layer (Bouchain, 2006).

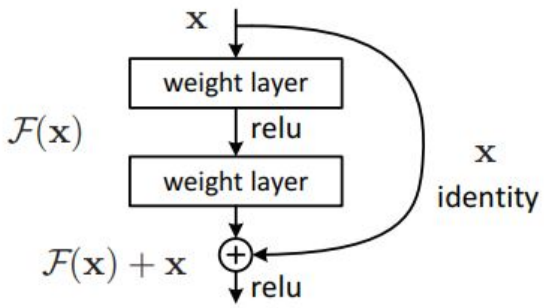

Fig. 5: A figure showing a building block for ResNet, taken from He et al. (2016a). The first mapping after the input layer is $\mathcal{F}(\mathbf{x})$ is recast into $\mathcal{F}(\mathbf{x})+\mathbf{x}$ after the second, stacked, non-linear layer.

The subsequent layers of a CNN extract a particular set of features from the input tensor and eventually recombine these features in the higher layers; this constitutes the learning process. Overall, the pillars of the CNN are the convolutional layers and *sub-sampling layers*. The sub-sampling layer also called as pooling layers, reduces the resolution of the feature maps while increasing their number; this is made possible because of spatial invariance in the NN, allowing the important features of the input image to already be captured by the top layers in the CNN. More so, this makes the location of the feature in the original image itself unimportant (LeCun et al., 1995).

III-B Residual Neural Network (ResNet)

The *level* of features, i.e. low/mid/high frequency, that a NN can classify depends on the number of stacked layers, so called as the depth of the network. The residual neural network, *ResNet*, is a NN that attempts to ease the training process for deeper networks by reformulating the layer of the network as *residual functions*. These residual functions learn with reference to the input given to the layer rather than arbitrarily learning with unreferenced functions (He et al., 2016a). This is brought about by implementing *skip* or *shortcut connections*. Shortcut connections allow for flow of information via short paths that connect the top layers with the bottom layers (Tong et al., 2017). These shortcut connections are not merely skipping one or more layers. In the case of the ResNet, the shortcut connections performs a so-called *identity* mapping. This can be understood from Fig. 5 which shows an example block of an *identity* shortcut connection. The input is directly added to the outputs of the second, stacked layer. These connections do not introduce additional parameters or complexity because the connecting layers

are of the same shape. The output from a ResNet for a particular block with weights W_i is given as:

$$\mathbf{y} = \mathcal{H}(\mathbf{x}) + \mathcal{F}(\mathbf{x}, \{W_i\}), \tag{12}$$

where $\mathbf{y}$ and $\mathbf{x}$ are the input and output vectors of the block in Fig. 5, $\mathcal{F}(\mathbf{x}, \{w_i\})$ is the residual mapping that needs to be learned, and when $\mathcal{H}(\mathbf{x}) = \mathbf{x}$ it is the identity shortcut mapping. Since in Fig. 5 the block has two layers, $\mathcal{F} = w_2 f(w_1 x)$, where f represents the activation function *ReLu* that is applied posterior to the first layer. It must be noted that the $\mathcal{F} + \mathbf{x}$ operation in Eq. 12 is made possible if and only if the two vectors have equal dimensions. If they are not of the same dimension, a linear projection W_s can be performed and the resulting mapping $\mathcal{H}$ in Eq. 12 becomes, $\mathcal{H}(\mathbf{x}) = W_s\mathbf{x}$. The biases are ignored here for simplicity (He et al., 2016a,b).

The main advantage tackled by ResNet is the *degradation* or the *vanishing gradient problem*, where the increase in depth causes a saturation and further degradation in the accuracy of the model. This problem was a bottleneck in further improving very deep CNNs (Bengio et al., 1994). Previous work in this field has shown that the use of identity shortcut connections in deep CNNs allows for faster convergence of models, better performance, and reduction in the number of parameters (Kim et al., 2016).

III-C The three Neural Networks: CNN, ResNet-7, and ResNet-18

The models built in this project are CNN, ResNet-7 and ResNet-18, as shown in Fig. 6, Fig. 7, and Fig. 8, respectively. Since the aim of these deep learning models is to catch variability in the microlensing light curves that happens on all time scale, both the CNN and ResNets are capable of executing the job successfully. The CNN is a simple convolutional neural network containing 7 total layers, and ResNet-7 and ResNet-18 contain a total of 7 and 18 layers, respectively. They are all fed with $32,000$ training and 8000 validation light curves. The details on the generation of these simulated light curves for ten radius categories are detailed in the following Sec. IV. The section aims to provide the details about the models built in this project.

- **CNN classification**: The simple CNN model, as shown in Fig. 6, contains a total of seven layers: an input layer, four convolutional layers, one hidden fully connected layer, and an output layer. The four convolutional layers have filter sizes $(32, 32, 44, 44)$ and optimized kernel sizes $(21, 21, 15, 15)$ respectively. The first hidden, fully connected layer contains 1500 nodes. A mini-batch size of 60 was used with a learning rate of 1×10^{-4} and decay 1.25×10^{-6}. The model was compiled using the optimizer *Adam*. It is worth noting the close resemblance of this model with the ResNet-7, shown in Fig. 7. This CNN model was a primarily built to compare its performance with its counterparts, ResNet-7 and ResNet-18, as will be further discussed in Sec. V-A.

- **ResNet-7 classification**: The input layer is reshaped and sliced into 2D arrays containing 1515×1 pixels. The ResNet-7 contains an input layer followed by two residual blocks each containing two convolutional layers, similar to the residual block shown in Fig. 5. The four convolution layers have filter sizes $(32, 32, 44, 44)$ and optimized kernel sizes $(21, 21, 15, 15)$ respectively. The activation functions used posterior to every layer is summarized in Fig. 7. Finally, the NN is fed into two fully connected layers, containing 1500 hidden and 10 output nodes respectively. The model is compiled by using the optimizer *Adam* with the loss function *categorical cross-entropy* (Chollet et al., 2015). A mini-batch size of 60 was used with a learning rate of 1×10^{-4} and decay 1.25×10^{-6}.

- **ResNet-18**: The ResNet-18 built in this project is shown in Fig. 8. Each residual block consists of three convolution layers with filter sizes $(32, 32, 32)$ and kernal sizes $(10, 20, 50)$ respectively. These residual blocks are repeated five times before joining the two fully connected layers. The connected layers contain a hidden layer with 1000 nodes, and the output layer with 10 nodes. The model compiler, loss function, learning rate, and decay were set to the same values as ResNet-7.

IV Generation of Mock Light Curves

Before generating the mock light curves, it is necessary to produce the convolved magnification map. The technique undertaken in this project to generate 5×10^5 light curves is similar to the one detailed in past literature (Wambsganss, 1999; Kochanek, 2004; Mediavilla et al., 2011), which uses the inverse ray-shooting technique (Vernardos et al., 2014). The maps used in this project were taken from GERLUMPH, which were generated using GPU−D (Vernardos et al., 2014).

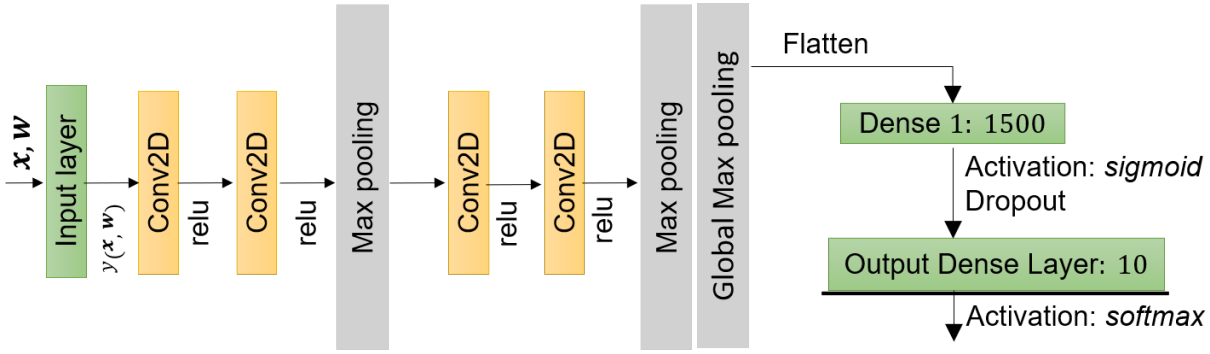

Fig. 6: The CNN model with 7 layers. The network contains an input layer, four convolutional layers, one hidden dense fully connected layer, and an output layer. The activation function used posterior to the convolutional layers is *relu*. The dropout ratio after the first dense layer was set to 0.4.

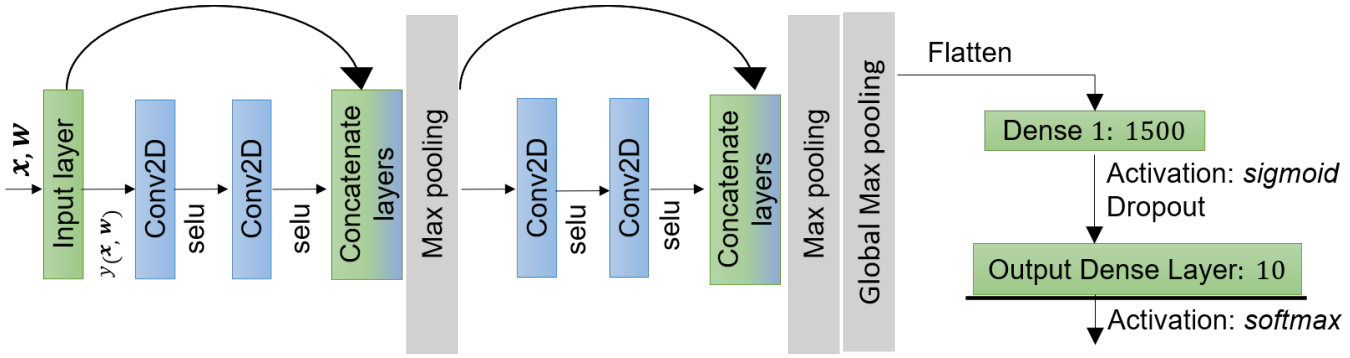

Fig. 7: The ResNet-7 model with forward skip connections. A two-convolutional layer residual block is concatenated with a prior layers via forward skip connections. This procedure is repeated twice before joining the dense connected layers that contain a hidden layer and an output layer. The dropout ratio after the first dense layer was set to 0.4.

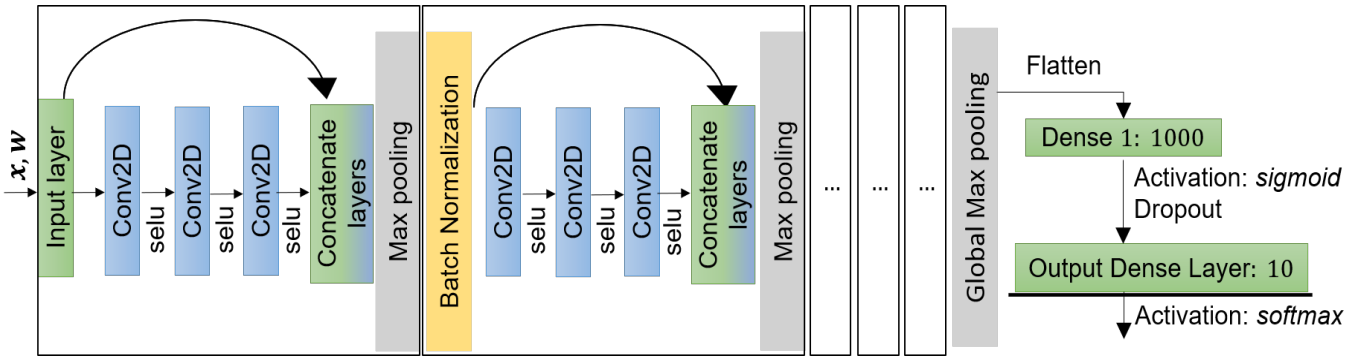

Fig. 8: The ResNet-18 model with 18 layers. Each residual block consists of three stacked convolutional layers along with a max pooling layer. The outputs from every residual block are normalized; there are five blocks in total before the outputs join the fully connected layers. The dropout ratio of the first dense layer was set to 0.7.

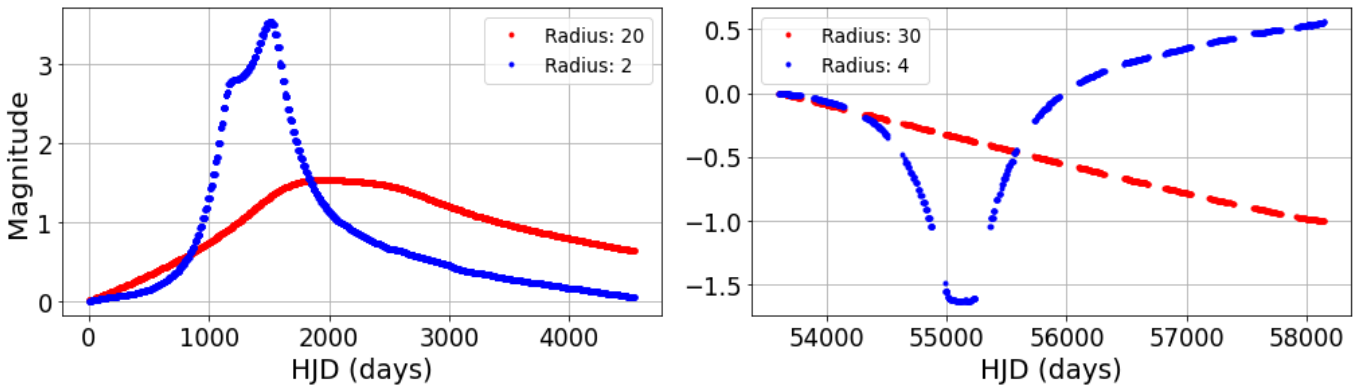

Fig. 9: The simulated light curves depict the magnification that is calculated along the light trajectory using the magnification maps, which are generated using the inverse-ray shooting method detailed by Vernardos et al. (2014); Vohl et al. (2015). Two arbitrary light-curves of the 10 categories of $R_{\rm S}$ are shown. The units of the source radius are in pixels, where one pixel corresponds to 0.07 $R_{\rm S}$. **Left**: The magnification is calculated along the trajectory as a function of continuous-time. **Right**: The magnification is calculated along the random trajectory as a function of time by accounting for the season gaps.

The computation of each map depends on parameters that are classified into three groups (Vernardos et al., 2014):

1) **External parameters**: A lens mass model, either analytical (Witt et al., 1995) or numerical (see Faure et al. for more details), is assumed such that the observables like the positions of the doubly-imaged quasar, times delays, etc., are reproduced. From this, the values of κ, γ, and the smooth matter fraction parameter, which is defined as $s = \kappa_*/\kappa$, are extracted for every position on the lens plane. This enables the calculation of the corresponding μ, using Eq. 7, for every pixel in the magnification map. The parameters κ, γ, and s for the image A and B of $J0158$ are given in Tab. I. It must be noted that due to the mass-sheet degeneracy (Falco et al., 1985), these values cannot be uniquely determined. However, for simulation purposes, the approximations of the macro-model parameters provide sufficient estimations about the source or the lens-galaxy.

$f_{\rm M/L} = 0.9$	κ	γ	κ_*/κ
A	0.23	0.39	0.81
B	0.72	1.03	0.92

TABLE I: The external parameters that are used while generating the magnification maps. These values were taken from Morgan et al. (2008). Note that $f_{\rm M/L}$ is the fraction of constant mass-to-light (M/L) ratio that is used to parameterize the lens models

2) **Microlensing parameters**: As the stellar microlens masses have a negligible effect on the magnification probability distribution (MPD), the mass spectrum was assumed to have a prior mean microlens mass $\langle M \rangle = 0.3$ $\rm M_\odot$. The number of simulated stars N_* in the lens plane is typically between $\sim 10^3 - 10^5$ (Vernardos & Fluke, 2014). Lastly, the relative transverse speed of the lens with respect to the source and the observer is set to $v_\perp = 500$ km/s.
3) **Map characteristics**: The map resolution is 8192×8192 pixels corresponding to an observed scale of (18.75×18.75) $R_{\rm E}$, as shown in Fig. 3. We use the source size found by Mosquera & Kochanek (2011), $R_{\rm ref} = 0.048$ $R_{\rm E}$ for reference, where $R_{\rm ref} = 1.64 \times 10^{16}$ cm is the characteristic size of the source $J0158$. One pixel corresponds to 0.003 $R_{\rm E}$ or 0.067 $R_{\rm ref}$. These values were obtained by assuming a thin-disk brightness profile for the source accretion disk (Shakura & Sunyaev, 1973).

The parameter of interest in this project is the size of the source $R_{\rm S}$. Since microlensing depends on $R_{\rm S}$, a magnification map was generated for each value of $R_{\rm S}$, for the A and B images. These maps were convolved and their difference was taken to obtain the desired microlensed magnification map (see Fig. 3). Ten maps were generated for source radii in the range between $R_{\rm S} = [0.13,\ 6.67]$ $R_{\rm ref}$; when $R_{\rm S}$ is converted to the pixel scale, it corresponds to the following categories: $[2, 4, 10, 15, 20, 30, 40, 60, 80, 100]$ pixels.

The inverse ray-shooting method, as detailed by Vernardos et al. (2014), shoots a large number of light rays through the lens plane and maps them on the source plane, thereby measuring the magnification along the trajectory of the light rays. The map must reach the desired magnification in the regions of interest, so a large number of rays need to be shot ($\approx 10^6$). A

total of 5×10^3 light curves are produced from the corresponding magnification maps for every radius; so a total of 5×10^4 light curves were produced for the 10 categories of $R_{\rm S}$. This is made possible by choosing the initial point $\mathbf{r}_0$ and using the effective velocity $\mathbf{v_e} = v_\perp(\cos\theta, \sin\theta)$ along the trajectory to calculate the magnification as a function of time, as shown in Fig. 9 (*left*). In this case, the trajectory direction θ and the start position of the trajectory on the lens plane are assumed to be independent, uniformly distributed variables. Furthermore, the magnification is calculated as a function of time accounting for the *season gaps*, which corresponds to the uneven sampling of the light curves caused due to the real-time observing methods. The light curve generated with season gaps is shown in Fig. 9 (*right*).

IV-A Interpolating the Light Curves

As shown in Fig. 4, to produce a light curve the QSO is observed for multiple cycles and they contain season gaps. To build models that can extract physical information from the observed microlensed light curves, the season gaps must be interpolated. Therefore, before using the simulated light curves with season gaps to train the deep learning algorithms, the aim that we wish to achieve by fitting these curves is to imitate the strategy that is adopted for real data. Additionally, as shown further in Sec. V-A, fitting the light curves allows us to attain a higher classification accuracy while using deep learning. The simulated light curves were fitted using linear interpolation in Sec. IV-A1 and using the Gaussian processes regression in Sec. IV-A2, as discussed below.

1) Linear Interpolation

The 5×10^4 light curves generated for the ten categories of $R_{\rm S}$ were fitted using a simple linear interpolation, as shown in Fig. 10. The reasoning behind interpolating the light curves is to predict the data that is not measured during the season gaps, using the remaining observed data. However, a simple linear interpolation deems to be inadequate if we were to estimate a microlensing event withing the season gaps. Therefore, a more resourceful attempt at fitting the light curves was brought about by using Gaussian Processes regression.

2) Gaussian Processes

Following the work by Rumelhart et al. (1986), there has been an increasing interest for non-linear empirical modelling of data. One other method of non-linear

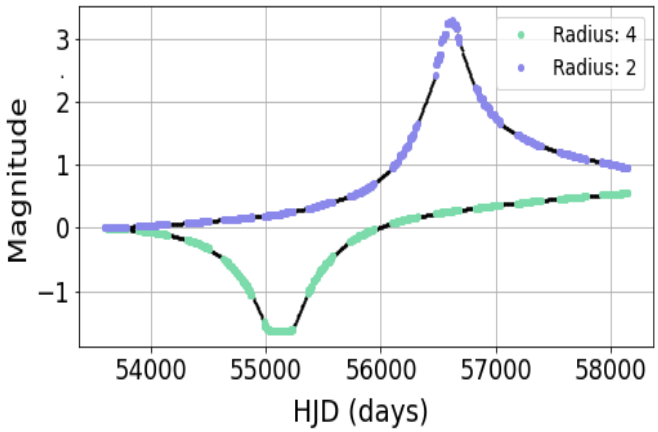

Fig. 10: Linear interpolation was used to fit the season gaps in the light curves. The figure shows two arbitrary light curves corresponding to the categories of $R_{\rm S} = 4$ pixels (*light-blue*) and $R_{\rm S} = 2$ pixels (*light-green*), which correspond to physical values of 0.13 $R_{\rm ref}$ and 0.25 $R_{\rm ref}$, respectively. The black points correspond to the curve fitted to the colored data points using linear interpolation.

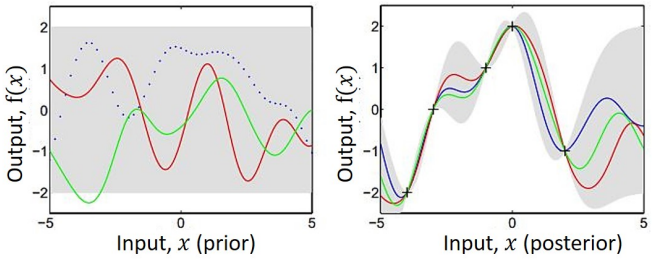

Fig. 11: Left: Three random functions are drawn from a prior GP. **Right**: The addition of the five observed data points conditions the prior GP function, now called the posterior function. The shaded grey area in both plots represents 2×standard deviation around the mean for every point of $f(\mathbf{x})$ (prior and posterior). Figure taken from Rasmussen, 2003, p. 15.

modelling is *Bayesian interference*, which considers a parameterized function $f(\mathbf{x}; \mathbf{w})$, where the parameters $\mathbf{w}$ are adapted to suit the input vector, $\mathbf{X}_N \equiv \{\mathbf{x}^{(n)}\}_{n=1}^N$, and its corresponding target values $\mathbf{y}_N \equiv \{y_n\}_{n=1}^N$. The posterior probability distribution expressing the interference of $f(\mathbf{x})$ with the input data is given by,

$$P(f(\mathbf{x})|\mathbf{X}_N, \mathbf{y}_N) = \frac{P(\mathbf{y}_N|f(\mathbf{x}), \mathbf{X}_N)P(f(\mathbf{x}))}{P(\mathbf{X}_N|\mathbf{y}_N)}, \quad (13)$$

where $P(\mathbf{y}_N|f(\mathbf{x}), \mathbf{X}_N)$ is the probability distribution assumed by the target values, given the modelling function $f(\mathbf{x})$. $P(f(\mathbf{x}))$ represents the prior distribution assumed by the model (MacKay, 1997, p. 2). The prior distribution is an implicit choice that is adapted to obtain the desired target values $\mathbf{y}_N$. Unlike the usual parameterized modelling, where $\mathbf{w}$ is changed

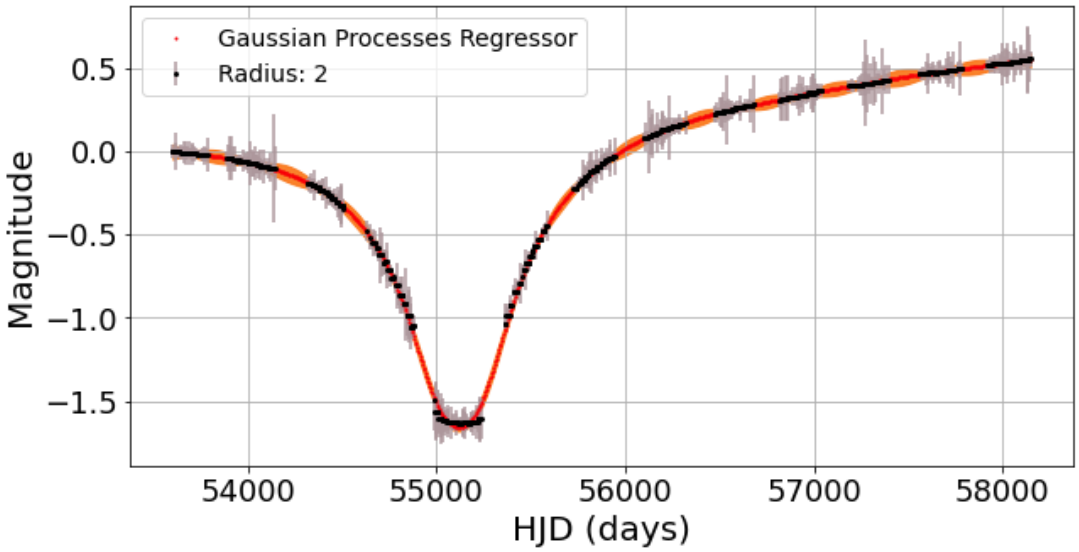

Fig. 12: A light curve corresponding to the category $R_s = 2$ pixels $= 0.13\ R_{\rm ref}$ is interpolated using the Gaussian processes regressor (*red line*). The *grey lines* signify the error-bars on the *black*-colored data points; these error-bars accounts for the noise added to the light curves. The *orange fill ups* around the interpolated line correspond to the 1-sigma standard deviation from the best-fit.

such that $f(\mathbf{x};\mathbf{w})$ is the best-fitting model to the data $(\mathbf{X}_N, \mathbf{y}_N)$, *Gaussian processes* (GP) directly adapts the prior distribution, as shown in Fig. 11. The prior distribution is specified by a mean, $m(\mathbf{x})$, and covariance function, $k(\mathbf{x}.\mathbf{x}')$. Therefore, we can write GP as,

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}.\mathbf{x}')). \tag{14}$$

Modelling the data is directly concerned with the problem of finding suitable properties for $k(\mathbf{x}.\mathbf{x}')$. All the 5×10^4 light curves were modelled using GP; a light curve from the data set is shown in Fig. 12. To specify the prior function, $m(\mathbf{x})$ was set to a constant $(=0)$ and the Matérn covariance function was used, which has a functional form:

$$k(\mathbf{x}.\mathbf{x}') = \frac{A}{\Gamma(\nu)2^{\nu-1}}(\phi)^{\nu}\ K_{\nu}(\phi), \tag{15}$$

$$\text{where, } \phi = \frac{\sqrt{2\nu}}{l} d(\mathbf{x},\mathbf{x}'), \tag{16}$$

and ν is the smoothness parameter, l is the characteristic length scale, A is the amplitude, $d(\cdot,\cdot)$ is the Euclidean distance, $K_\nu(\cdot)$ is the modified Bessel function, and $\Gamma(\cdot)$ is the gamma function (Rasmussen, 2003, p. 93). In this project, $\nu = 1.5$, $l = 200$ and the fitting was implemented by the function `GaussianProcessesRegressor` using the python package `scikit-learn` (Pedregosa et al., 2011). The amplitude of $k(\mathbf{x}.\mathbf{x}')$ was $A = 2$ and the white noise added to the system using the `noise_level` argument was set to five. The code can be found on GitHub (Notebook 7).

V Results and Discussions

The models were trained and evaluated for the following variations of light curves:

(A) Light curves without season gaps
(B) Light curves with season gaps
(C) Light curves with season gaps that are fitted with linear interpolation (LI)
(D) Light curves with season gaps that are fitted with Gaussian processes regression (GPR)

Hereafter, these four cases will be referenced using the enumerator A, B, C, or D. First, we test the performance of the three models: CNN, ResNet-7, and ResNet-18, on the simplest case A, as discussed in Sec. V-A, thereby choosing the best two of the three deep learning models. We then proceed to evaluate the classification accuracy achieved for each of the cases B, C, and D in Sec. V-B.

V-A Comparison of the performance of the NNs

The accuracy of a model is a measure of the number of test light curves that are correctly classified by the NN. An equivalent measure can be obtained by calculating the error rate or loss $\mathcal{L}$ of the trained model. A desirable result would be for the model to achieve the highest test accuracy or equivalently; minimize the test loss, also called as the validation loss. The validation loss corresponds to the failure rate of the model to classify test data (validation data). Of the total 5×10^4 light curves, 20% of the curves are separated for evaluating the classification accuracy of the model (10^4 curves). The problem of overfitting, which was

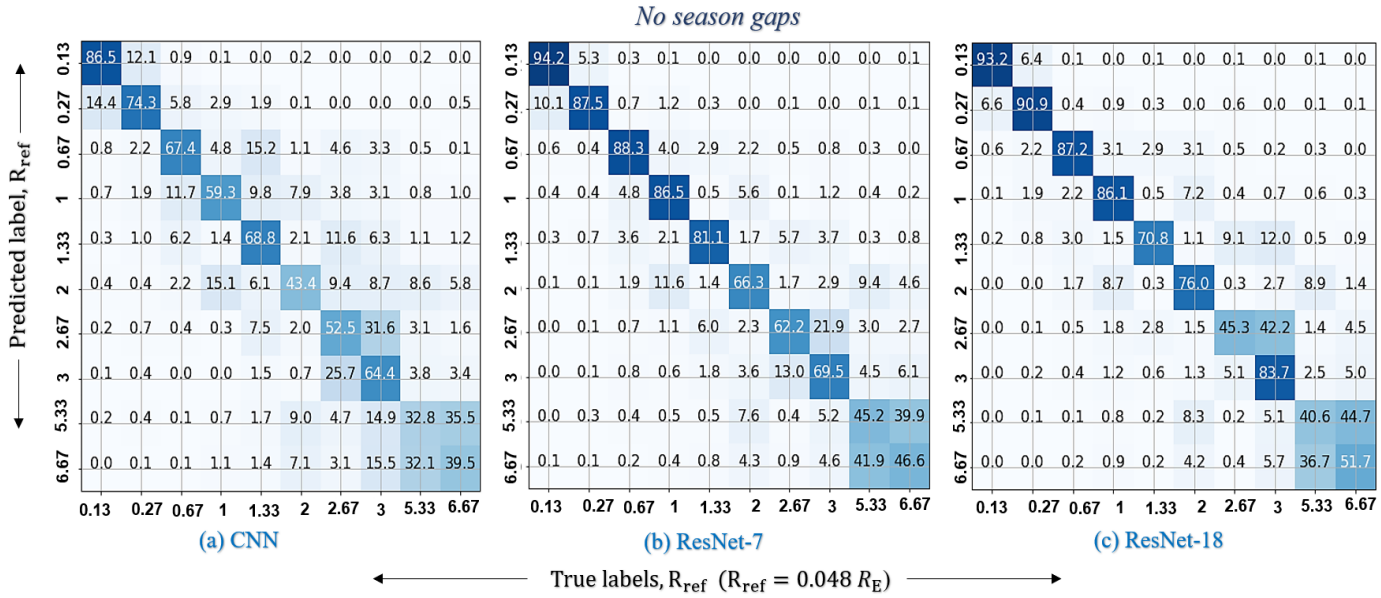

Fig. 13: The confusion matrices obtained by training the models: CNN, ResNet-7, and ResNet-18, for case A (light curves without season gaps). CNN, ResNet-7, and ResNet-18 achieved accuracies of 61.4%, 73.3% and 71.9% respectively. The labels represent the ten categories of source radius R_S of $J0158$ into which the data is classified. Here, $R_{ref} = 0.048\ R_E$, the so-called reference radius, corresponds to the source radius that has been found by Mosquera & Kochanek (2011). All the models share the common feature of being able to classify smaller source radii $R_S \leq R_{ref}$ with higher accuracy.

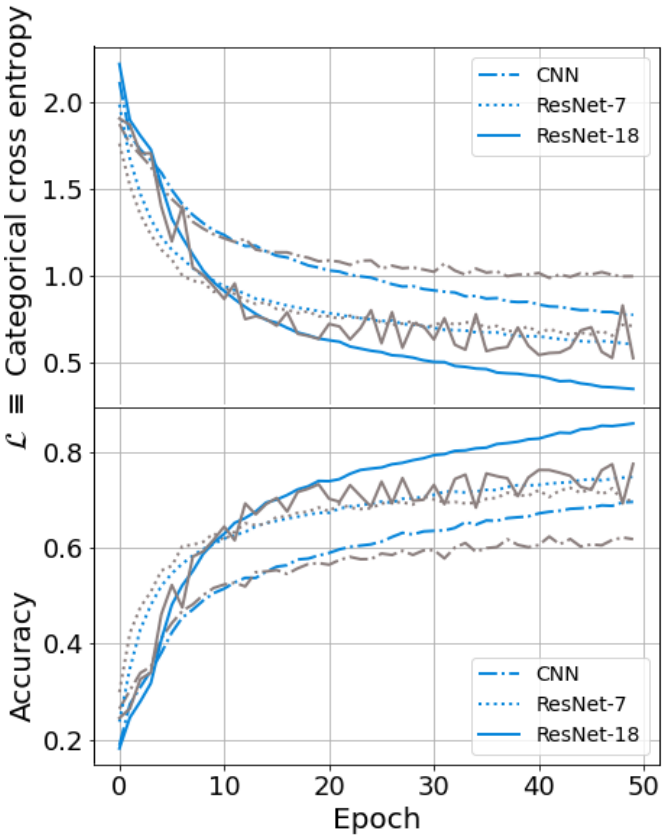

Fig. 14: The loss and accuracy curves obtained by training the three models: CNN (dot-dashed line), ResNet-7 (dotted line), and ResNet-18 (solid-line). The data set contains light curves without any season gaps. The blue lines denote the training curve, and the grey lines represent the validation curves.

briefly mentioned in Sec. III, graphically translates to a large gap between the training and validation loss values. Therefore, the ideal case for a well-trained model is when the gap between the training loss and the validation loss is small. On the contrary, the case of underfitting is described as the scenario where the training loss value is large (Goodfellow et al., 2016, p. 101).

The data set for case A is used to compare the performance of the three neural networks introduced in this project. The loss and accuracy curves for the three models are shown in Fig. 14. The *blue* and *grey* lines denote the training and validation accuracy-loss curves, respectively. ResNet-18 achieved a higher validation accuracy (lower validation loss) of 73.3% (0.6%) compared to ResNet-7 71.9% (0.7%); however, they both perform better compared to CNN, which achieved 61.4% (1%). The phenomenon of overfitting becomes evident at a very early epoch ≈ 15 for ResNet-18. Another useful measure to evaluate the performance of multi-class deep learning models is the **confusion matrix**. The diagonal elements show the truth value for the multiple-categories into which the data is classified. The off-diagonal elements show undesired correlations. The confusion matrices for the three models: CNN, ResNet-7, and ResNet-18, for case A is shown in Fig. 13. As the two ResNets attain higher classification accuracies for most categories of source radii, they are favoured over the CNN for further evaluation of the cases B, C, and D.

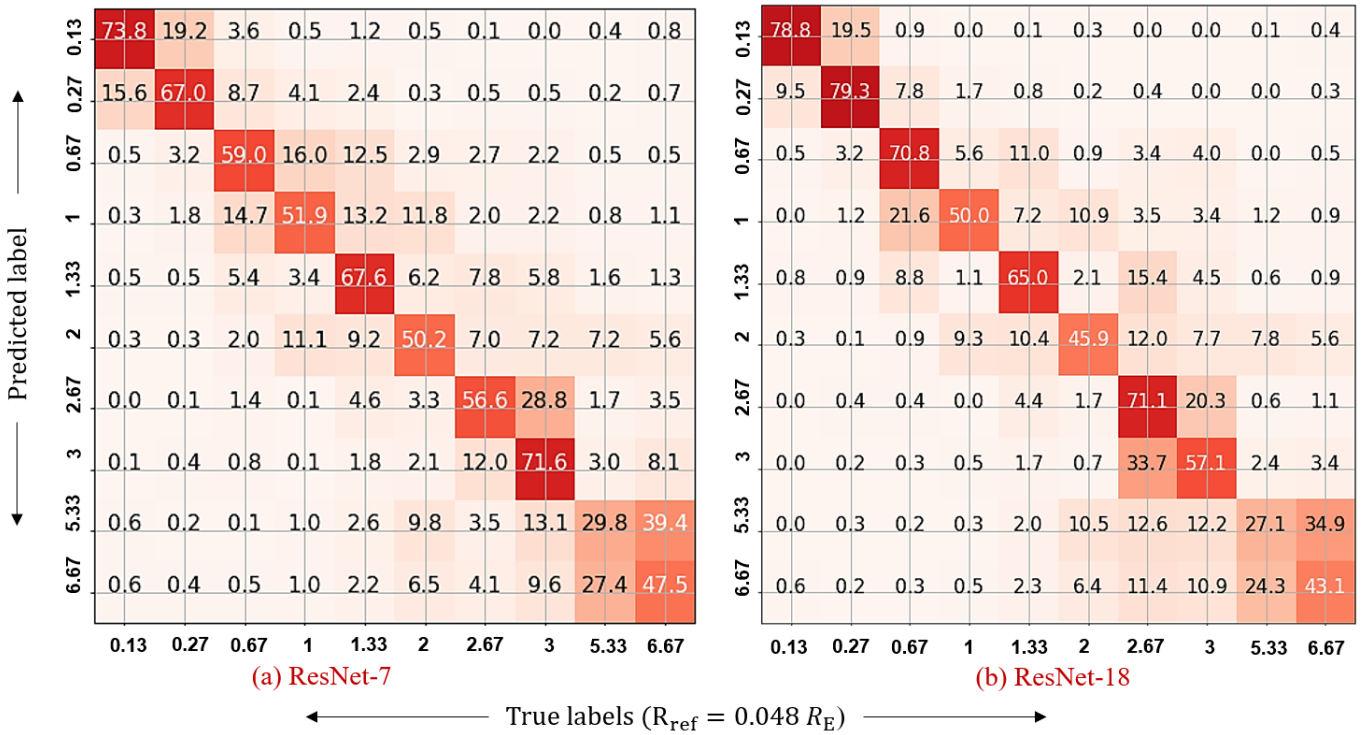

Fig. 15: The confusion matrices that are generated by training the ResNet-7 and ResNet-18 with light curves containing season gaps, which are not interpolated (case B). ResNet-7 and ResNet-18 reached an accuracy of 56.6% and 61.4%, respectively. As usual, $R_{ref} = 0.048\ R_E$, is the so-called reference radius Mosquera & Kochanek (2011).

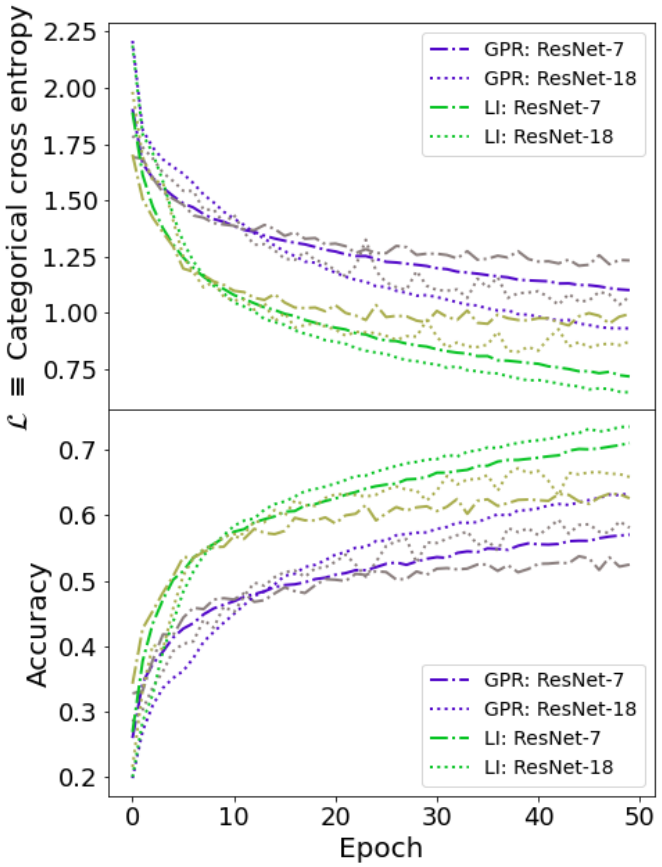

Fig. 16: Loss-accuracy curves for cases C (purple) and D (green), obtained by training the two models: ResNet-7 (dot-dashed line) and ResNet-18 (dotted line).

V-B Evaluating the Classification Accuracy

The confusion matrices that were generated by training the ResNet-7 and ResNet-18 using the light curves for the cases B, C, and D, are shown in Fig. 15, 17, and 18, respectively. For case B, the light curves are inputted into the network without interpolating the season gaps. This would mean that the model must learn the kinks in the light curves that are generated between two discontinuous cycles. This resulted in the drop of accuracy for case B: 16.7% for ResNet-7 and 10.5% for ResNet-18, compared to case A. The case C, where the light curves were fitted using linear interpolation, showed considerable improvement in the classification accuracy, which can be seen from the confusion matrix (Fig. 17), compared to the former case B. However, the same trend of improvement in the classification accuracy that is arising from fitting the light curves is not evident in case D, which used GPR, as summarized in Tab. II. This can be further understood from Fig. 16, which shows the loss-accuracy curves for the cases C and D, for the two models: ResNet-7 and ResNet-18. The *top panel* of Fig. 18 shows the loss-curves, while the *bottom-panel* shows the accuracy-curves. The training-curves and validation-curves for each model in case C (LI) are plotted in *green* and *light-green* respectively, and those for case D (GPR) are plotted in *purple* and *grey*, respectively.

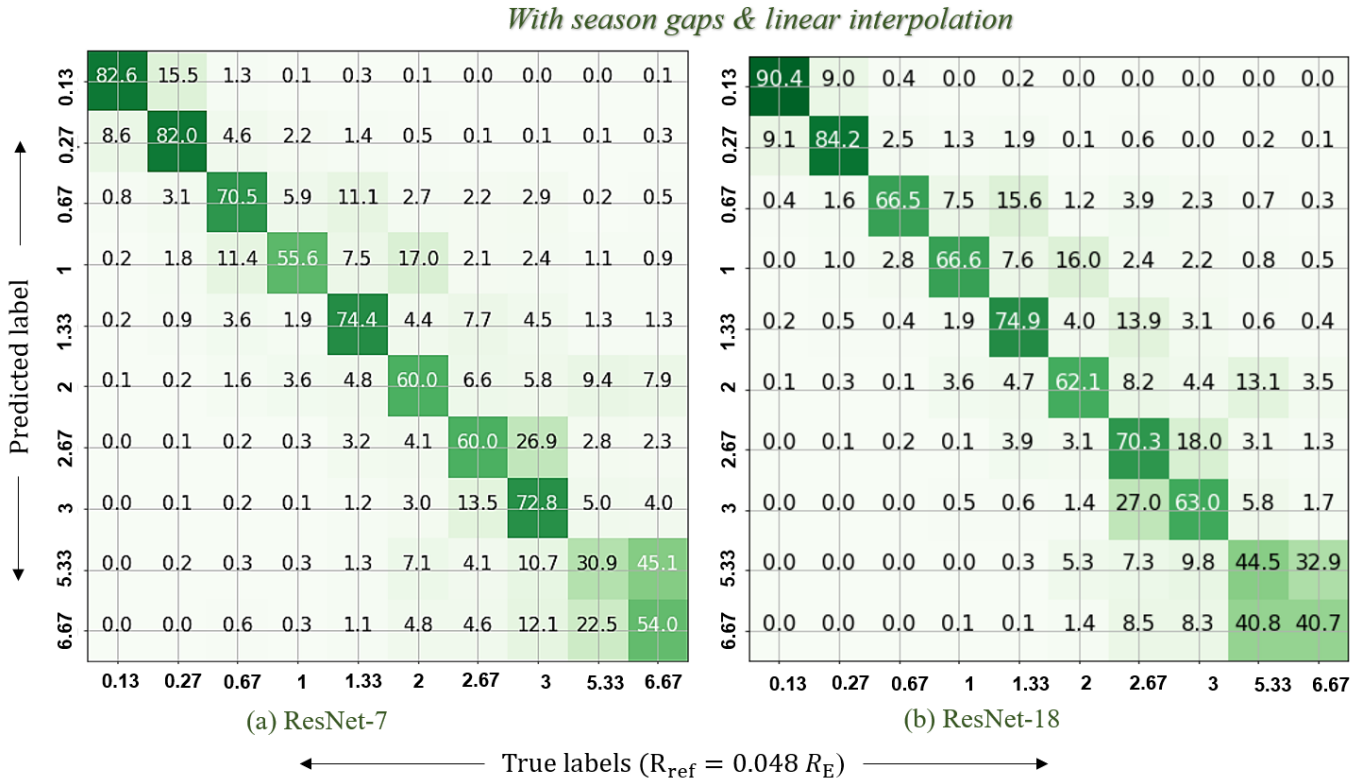

Fig. 17: The confusion matrices for ResNet-7 and ResNet-18 that are generated for case C, where the season gaps of the light curves were fitted using linear interpolation. The true and predicted labels represent the ten categories of source radius R_S of $J0158$ into which the data is classified. Both the models perform a better classification for the categories with $R_S \leq R_{ref}$; however, ResNet-18 achieved an overall higher accuracy of 66.1% than ResNet-7 with an accuracy of $63.4\%\%$.

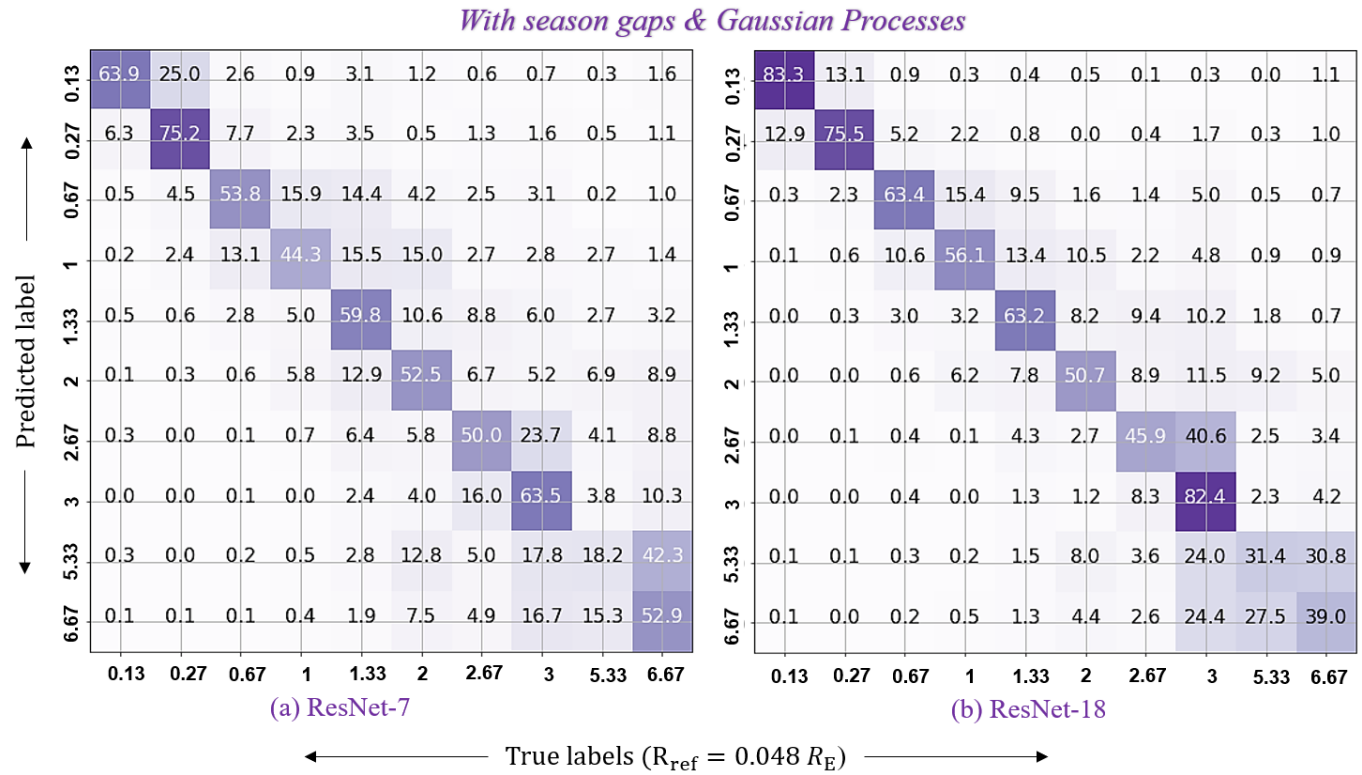

Fig. 18: The confusion matrices that are generated by training the ResNet-7 and ResNet-18 for case D, where the season gaps of the light curves were fitted using the Gaussian Processes Regression algorithm (detailed in Sec. IV-A2). ResNet-7 and ResNet-18 achieved an accuracy of 53.2% and 56.93%, respectively. The categories $(0.13, 0.67, 1, 1.33)$ R_{ref} are better classified by ResNet-18 than ResNet-7.

Case		CNN	ResNet-7	ResNet-18
A	Loss (%)	1.0	0.6	0.7
	Accuracy (%)	61.4	71.9	73.6
B	Loss (%)	-	1.2	1.0
	Accuracy (%)	-	56.6	61.4
C	Loss (%)	-	1.0	0.9
	Accuracy (%)	-	63.4	66.1
D	Loss (%)	-	1.2	1.1
	Accuracy (%)	-	53.2	56.9

TABLE II: Summary of the loss and accuracy values for the models CNN, ResNet-7 and ResNet-18, which were trained and validated for the four cases: A, B, C, and D. Owing to the outperformance of the ResNets compared to the CNN in case A, the remaining cases B, C, D were classified using the two ResNet models alone.

It is observed that there is an total 10.2% and 9.2% *drop* in the accuracy achieved by ResNet-7 and ResNet-18, respectively, for case D (GPR) compared to case C (LI), as summarized in Tab. II. This is a consequence of the fact that the GPR fitted curves smoothens the overall light-cuves, which results in a loss of the high-frequency features and adds an extra layer of variability. Since the ResNets were trained with the fitted curves, rather than the simulated curves with the season gaps *filled* with the fitted GPR points, this could result in a drop in the classification accuracy, thereby making it a topic of further investigation for the future. This drop in the classification accuracy can also be seen when comparing Fig. 18 to Fig. 17.

VI Conclusions and Future Work

Microlensing light curves were simulated from the magnification maps for the quasar Q $J0158$, using the inverse ray-shooting method. 5×10^5 light-curves were generated for the ten categories of source radii, between the range $(0.13 - 6.67)$ $R_{\rm ref}$. The light curves were generated using season gaps to mimic observed data sets; these gaps were fitted using linear interpolation and Gaussian processes regression (GPR). Three deep learning models: CNN, ResNet-7, and ResNet-18 were used to classify the microlensing light curves into the ten categories of source radius. The performance of the three models was first tested on the simple data set, where the light curves did not contain season gaps. It was observed that ResNet-7 and ResNet-18, which achieved an accuracy of 71.9% and 73.6% respectively, outperformed the CNN, which achieved an overall accuracy of 61.4%. Therefore, the ResNets were used for further classification of $R_{\rm S}$ for the cases of light curves: (i) with season gaps, (ii) with gaps and the light curves are linearly interpolated and (iii) with gaps and the light curves are fitted using GPR.

The classification accuracy was measured by plotting confusion matrices. A diagonal confusion matrix implies that the model can successfully extract the information about the size of the background source. More so, if the model fails to provide a precise and accurate estimate of a category for $R_{\rm S}$, it yet provides an estimate on the range of radii (categories) for $R_{\rm S}$ that must hold the final result. It was observed in this project, that the models were able to best classify the light curves when they were linearly interpolated, as seen in Tab. II. However, the failure in attaining a high classification accuracy for curves fitted using GPR was recognized; it owed to the fact that the light curves used for training the models have to be modified, such that only the season gaps had to be *filled* using *fitted GPR* points, rather the whole fitted GPR light curve. Acknowledging the wealth of information that can be extracted from microlensing light curves, this project aimed to develop an efficient, non-linear model that can be used to extract information about the source size of the background quasar from microlensing light curves.

Future work on this project involves the following:

1) Applying the network to the observed light curve of $J0158$. The first step in executing this procedure would involve performing a detailed error analysis on the model, which can be done as follows: (i) building a new Baysian NN using Gaussian processes, (ii) modifying the network such that finding $R_{\rm S}$ becomes a regression problem rather than a discrete classification problem that is used in this project, or (iii) adding fake white noise to generate multiple instances of the real light curve and asking the network to classify it.
2) Generating magnification maps and hence, light curves that catch the short-time (high-frequency) variation in the microlensing signal.
3) Adopting the approach taken by Vernardos & Tsagkatakis (2019), where they increase the parameter space by generating maps for a range of κ and shear γ that allows the study of many objects simultaneously rather than focusing on a particular case of κ, γ for a single QSO. Moreover, we can further extend this idea to accommodate for a range in the parameter space of the transverse relative velocity of the lens $v_\perp$.

All the code for this project is made available at the following GitHub repository: https://github.com/SoumyaShreeram/Microlensing_with_NeuralNets.

Acknowledgements: Special thanks to Martin Millon and Eric Paic for taking their time to guide me through this project, and to Prof. Courbin for introducing me to the project and providing me with this opportunity to delve deeper in the field of quasar microlensing.

This document contains 5777 words.

References

Bengio, Y., Simard, P., & Frasconi, P. 1994, IEEE transactions on neural networks, 5, 157
Blackburne, J. A., Pooley, D., Rappaport, S., & Schechter, P. L. 2011, The Astrophysical Journal, 729, 34
Bouchain, D. 2006, Institute for Neural Information Processing, 2007
Chang, K. & Refsdal, S. 1979, Nature, 282, 561
Chollet, F. et al. 2015, Keras, https://keras.io
Courbin, F. & Minniti, D. 2002, Gravitational lensing: an astrophysical tool, Vol. 608 (Springer Science & Business Media)
Dominik, M. 2011, General Relativity and Gravitation, 43, 989
Dyson, F. W., Eddington, A. S., & Davidson, C. 1920, Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 220, 291
Eddington, A. S. 1919, The Observatory, 42, 119
Eigenbrod, A., Courbin, F., Sluse, D., Meylan, G., & Agol, E. 2008, Astronomy & Astrophysics, 480, 647
Einstein, A. 1911, Annals of Physics, 340, 898
Einstein, A. 1915, session area preuss. Akad. Wiss., Vol. 47, No. 2, pp. 831-839, 1915, 47, 831
Falco, E., Gorenstein, M., & Shapiro, I. 1985, The Astrophysical Journal, 289, L1
Faure, C., Anguita, T., Eigenbrod, A., et al. 2009, Astronomy & Astrophysics, 496, 361
Floyd, D. J., Bate, N., & Webster, R. 2009, Monthly Notices of the Royal Astronomical Society, 398, 233
Galeev, A., Rosner, R., & Vaiana, G. 1979, The Astrophysical Journal, 229, 318
Goodfellow, I., Bengio, Y., & Courville, A. 2016, Deep Learning (MIT Press), http://www.deeplearningbook.org
He, K., Zhang, X., Ren, S., & Sun, J. 2016a, in Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778
He, K., Zhang, X., Ren, S., & Sun, J. 2016b, in European conference on computer vision, Springer, 630–645
Ingrosso, G., Novati, S. C., De Paolis, F., et al. 2011, General Relativity and Gravitation, 43, 1047
Kayser, R., Refsdal, S., & Stabell, R. 1986, Astronomy and Astrophysics, 166, 36
Kim, J., Kwon Lee, J., & Mu Lee, K. 2016, in Proceedings of the IEEE conference on computer vision and pattern recognition, 1646–1654
Kochanek, C. S. 2004, The Astrophysical Journal, 605, 58
Laor, A. 1991, in Iron Line Diagnostics in X-ray Sources (Springer), 205–208
Lau, M. M. & Lim, K. H. 2017, in 2017 2nd international conference on control and robotics engineering (ICCRE), IEEE, 201–206
LeCun, Y., Bengio, Y., et al. 1995, The handbook of brain theory and neural networks, 3361, 1995
Liebes Jr, S. 1964, Physical Review, 133, B835
Limousin, M., Kneib, J.-P., & Natarajan, P. 2005, Monthly Notices of the Royal Astronomical Society, 356, 309
MacKay, D. J. 1997
Mao, S. & Paczynski, B. 1991, The Astrophysical Journal, 374, L37
Mediavilla, E., Mediavilla, T., Muñoz, J., et al. 2011, The Astrophysical Journal, 741, 42
Millon, M., Courbin, F., Bonvin, V., et al. 2020, arXiv preprint arXiv:2002.05736
Miyamoto, S. & Kitamoto, S. 1991, The Astrophysical Journal, 374, 741
Morgan, C. W., Eyler, M. E., Kochanek, C., et al. 2008, The Astrophysical Journal, 676, 80
Morgan, C. W., Hainline, L. J., Chen, B., et al. 2012, The Astrophysical Journal, 756, 52
Morgan, C. W., Hyer, G. E., Bonvin, V., et al. 2018, The Astrophysical Journal, 869, 106
Morgan, N. D., Dressler, A., Maza, J., Schechter, P. L., & Winn, J. N. 1999, The Astronomical Journal, 118, 1444
Mosquera, A. M. & Kochanek, C. S. 2011, The Astrophysical Journal, 738, 96
Mosquera, A. M., Muñoz, J. A., Mediavilla, E., & Kochanek, C. S. 2011, The Astrophysical Journal, 728, 145
Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825
Pooley, D., Blackburne, J. A., Rappaport, S., & Schechter, P. L. 2007, The Astrophysical Journal, 661, 19
Rasmussen, C. E. 2003, in Summer School on Machine Learning, Springer, 63–71
Refsdal, S. & Bondi, H. 1964, Monthly Notices of the Royal Astronomical Society, 128, 295
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1986, nature, 323, 533
Schmidhuber, J. 2015, Neural networks, 61, 85
Schneider, P., Kochanek, C., & Wambsganss, J. 2006, Gravitational lensing: strong, weak and micro: Saas-Fee advanced course 33, Vol. 33 (Springer Science & Business Media)
Shakura, N. I. & Sunyaev, R. A. 1973, Astronomy and Astrophysics, 24, 337
Svozil, D., Kvasnicka, V., & Pospichal, J. 1997, Chemometrics and intelligent laboratory systems, 39, 43
Tong, T., Li, G., Liu, X., & Gao, Q. 2017, in Proceedings of the IEEE International Conference on Computer Vision, 4799–4807
Vernardos, G. & Fluke, C. J. 2014, Astronomy and Computing, 6, 1
Vernardos, G., Fluke, C. J., Bate, N. F., & Croton, D. 2014, The Astrophysical Journal Supplement Series, 211, 16
Vernardos, G. & Tsagkatakis, G. 2019, Monthly Notices of the Royal Astronomical Society, 486, 1944

Vohl, D., Fluke, C. J., & Vernardos, G. 2015, Astronomy and Computing, 12, 200
Walsh, D., Carswell, R. F., & Weymann, R. J. 1979, Nature, 279, 381
Wambsganss, J. 1999, Journal of Computational and Applied Mathematics, 109, 353
Witt, H. J., Mao, S., & Schechter, P. L. 1995, The Astrophysical Journal, 443, 18
Zwicky, F. 1937, Physical Review, 51, 290