A&A 664, A4 (2022)
https://doi.org/10.1051/0004-6361/202142463

Astronomy & Astrophysics

Finding strong gravitational lenses through self-attention

Study based on the Bologna Lens Challenge

Hareesh Thuruthipilly[1], Adam Zadrozny[1], Agnieszka Pollo[1,2], and Marek Biesiada[1,3]

[1] National Centre for Nuclear Research, Warsaw, Poland
e-mail: Hareesh.Thuruthipilly@ncbj.gov.pl; Adam.Zadrozny@ncbj.gov.pl; Agnieszka.Pollo@ncbj.gov.pl
[2] Jagiellonian University, Kraków, Poland
[3] Department of Astronomy, Beijing Normal University, Beijing 100875, PR China

Received 16 October 2021 / Accepted 11 May 2022

ABSTRACT

Context. The upcoming large-scale surveys, such as the Rubin Observatory Legacy Survey of Space and Time, are expected to find approximately 10^5 strong gravitational lenses by analysing data many orders of magnitude larger than those in contemporary astronomical surveys. In this case, non-automated techniques will be highly challenging and time-consuming, if they are possible at all.
Aims. We propose a new automated architecture based on the principle of self-attention to find strong gravitational lenses. The advantages of self-attention-based encoder models over convolution neural networks (CNNs) are investigated, and ways to optimise the outcome of encoder models are analysed.
Methods. We constructed and trained 21 self-attention-based encoder models and five CNNs to identify gravitational lenses from the Bologna Lens Challenge. Each model was trained separately using 18 000 simulated images, cross-validated using 2000 images, and then applied to a test set with 100 000 images. We used four different metrics for evaluation: classification accuracy, the area under the receiver operating characteristic (AUROC) curve, and TPR_0 and TPR_{10} scores (two metrics of evaluation for the Bologna challenge). The performance of self-attention-based encoder models and CNNs participating in the challenge are compared.
Results. The encoder models performed better than the CNNs. They were able to surpass the CNN models that participated in the Bologna Lens Challenge by a high margin for the TPR_0 and TPR_{10}. In terms of the AUROC, the encoder models with 3×10^6 parameters had equivalent scores to the top CNN model, which had around 23×10^6 parameters.
Conclusions. Self-attention-based models have clear advantages compared to simpler CNNs. They perform competitively in comparison to the currently used residual neural networks. Self-attention-based models can identify lensing candidates with a high confidence level and will be able to filter out potential candidates from real data. Moreover, introducing the encoder layers can also tackle the overfitting problem present in the CNNs by acting as effective filters.

Key words. gravitational lensing: strong – methods: data analysis – techniques: image processing – cosmology: observations

1. Introduction

Strong gravitational lensing is the phenomenon by which a distant galaxy or quasar produces multiple highly distorted images because of the gravitational field of the foreground galaxy or a nearby massive astronomical body. Finding and analysing these strong lenses (SLs) is important; they have diverse applications in cosmology and astrophysics, ranging from estimating the Universe's dark matter distribution to constraining the cosmological models (Koopmans et al. 2006; Covone et al. 2009; Collett & Auger 2014; Cao et al. 2015; Bonvin et al. 2017). Consequently, the current and upcoming surveys have given significant attention to detecting strong gravitational lensing systems. For a detailed review of the applications of strong lensing, we refer to Blandford & Narayan (1992) and Treu (2010).

However, for all these analyses a large sample of SLs is required. Unfortunately, only a few hundred lensing systems have been detected and confirmed by the present astronomical surveys to date. One of the largest lens catalogues available now is from the Sloan Lens ACS Survey (SLACS), with only 130 observed lenses (Bolton et al. 2008). With the upcoming era of advanced missions such as Euclid (Scaramella et al. 2022) and LSST (Ivezić et al. 2019; Verma et al. 2019), the number of observable SLs is expected to reach 10^5, which should be identified from around 10^9 astronomical objects. Similarly, the number of new SLs expected to be in the Square Kilometre Array (SKA) survey will have similar orders of magnitude (McKean et al. 2015). To analyse the enormous amount of data produced from the present and future large-scale surveys, various methods have been tried out, including crowd science (Marshall et al. 2016) and semi-automated methods, for example arc detectors (Lenzen et al. 2004; Cabanac et al. 2007). However, these methods have only had minor success and were too time-consuming to be a practical proposition. Hence, the situation demands better and more effective alternative approaches to detect SLs in future large-scale surveys.

It is worth mentioning that the advancements in artificial intelligence (AI) have opened up a plethora of opportunities and have been widely applied in astronomy and astrophysics (e.g. galaxy classification by Pérez-Carrasco et al. 2019, supernova classification by Cabrera-Vives et al. 2017, and lens modelling by Pearson et al. 2019). A particular class of deep-learning models known as convolutional neural networks (CNNs) has recently been shown to work exceptionally well in finding SLs. Hence, developing deep-learning-based algorithms to detect SLs from

large-scale surveys is an actively investigated area now (Lanusse et al. 2017; Schaefer et al. 2018; Davies et al. 2019; Chianese et al. 2020). For instance, Jacobs et al. (2017) applied CNNs to the data from the Canada-France-Hawaii Telescope Legacy Survey (CFHTLS) to find SLs, and numerous other successful attempts of finding potential SL candidates from the Kilo Degree Survey (KiDS) have been reported (Petrillo et al. 2017, 2019a,b; He et al. 2020; Li et al. 2020). Likewise, various groups have successfully used CNNs to identify strong lens galaxy-scale systems from large-scale surveys, such as the Dark Energy Survey (DES) (Jacobs et al. 2019; Rojas et al. 2021), the Dark Energy Spectroscopic Instrument Legacy Imaging Surveys (Huang et al. 2020, 2021), and Pan-STARRS (Cañameras et al. 2020), and from comparatively small-scale surveys, such as VOICE (Gentile et al. 2021).

An exciting feature of the CNNs is that they can directly take the image as the input and learn the image features, making them one of the most popular and robust architectures currently being used. Generally, the learning capacity of a neural network increases with the number of layers in the network. The network can then learn the low-level features with the first layers and then learn more complex features with increasing depth (Russakovsky et al. 2015; Simonyan & Zisserman 2015). However, increasing the layers in the neural network will result in higher complexity, which in turn may lead to overfitting (Hawkins 2004). In addition, the gradient of the cost function decreases exponentially. Eventually, it vanishes for very deep networks, commonly called the vanishing gradient effect (Hochreiter 1991; Hochreiter et al. 2001). These two problems meant that creating very deep Convolution Networks was a challenging task (Srivastava et al. 2015).

However, the recently introduced idea of residual learning tackles these problems by introducing skip connections between the input and output of a few convolution layers (He et al. 2016). As a result, the CNN learns the difference between the inputs and outputs rather than their direct mapping. Due to the skip connections, the gradients can reach deeper layers, thus tackling the vanishing gradient effect. Recently, He et al. (2015) were able to build models as deep as 1000 layers while increasing classification accuracy for the ImageNet Large-Scale Visual Recognition Challenge 2015. However, the scientific community are constantly looking for alternative and simple solutions that can outperform the existing models with reasonable computational cost.

Recently, there was a breakthrough in natural language processing (NLP) by the introduction of a new self-attention-based architecture known as the transformers (Vaswani et al. 2017). Since then, there have been attempts to adapt the idea of self-attention to build better image processing models (Parmar et al. 2019; Zhao et al. 2020; Tan et al. 2021). The basic idea behind the transformer architecture is the attention mechanism, which has also found a wide variety of applications in machine learning (Zhang et al. 2018; Fu et al. 2019). In the case of NLP, self-attention correlates different positions of a single sequence in order to calculate a representation of the sequence. The idea of self-attention, as the name suggests, is to give relative importance to the input features based on the input features themselves, which helps the network to create a representation of the input with the relatively important features only. Recently, Facebook Inc. (Carion et al. 2020) and Google Brain (Dosovitskiy et al. 2021) have been able to surpass the existing image recognition models with transformer-based architectures. To our best knowledge, the transformer-based models have not been employed in astrophysics yet. In this paper we explore the possibilities of this new architecture in detecting strongly gravitationally lensed systems.

We implemented various self-attention-based encoder models (transformer encoders) to find the gravitational lenses from the Bologna Lens Challenge. We also compared the performance of the encoder models with created CNNs, and the CNNs participated in the challenge. The main objective of our study was to explore how suitable the transformer encoders are for finding strong lenses and how to optimise the performance of transformer encoders. From our analysis, we found that the encoder models perform better than the CNN models compared. We were able to beat the top TPR_0 and TPR_{10} score (two metrics of evaluation for the Bologna Challenge) by a significant margin and to reach the top AUROC reported during the challenge.

The paper is organised as follows. Section 2 briefly describes the data we used to train our models. Section 3 provides a brief overview of the methodology used in our study, including the model's architecture and information on how the models were trained. The results of our analysis are presented in Sect. 4. A detailed discussion of our results with a brief review of the performance of encoder models compared to the CNN models that participated in the challenge is presented in Sect. 5. Section 6 concludes our analysis by highlighting the advantages of the encoder models over CNN models.

2. Data

The data used in this study is from the Bologna Strong Gravitational Lens Finding Challenge (Metcalf et al. 2019). The challenge consisted of two different challenges that could be registered independently. The first challenge was designed to mimic the datasets from surveys such as Euclid consisting of single-band images. The second challenge was designed to resemble data from ground-based detectors with multiple bands. It was roughly modelled on the data from the Kilo-Degree Survey (KiDS) reported in de Jong et al. (2013). However, the simulated images did not strictly mimic the surveys; they were only employed as references to set noise levels, pixel sizes, sensitivities, and other parameters. The distributions of source redshift and Einstein radii in the challenge datasets are shown in Fig. 1. The challenge was opened on November 25, 2016, and closed on February 5, 2017. Surprisingly, automated methods such as CNN and SVM showed far better results than human inspection. During the challenge, these methods were able to classify the images with high confidence where a human would have doubt.

The mock images for the challenge were created using Millennium simulation and GLAMER lensing code (Boylan-Kolchin et al. 2009; Metcalf & Petkova 2014). Sources from the *Hubble* Ultra Deep Field (UDF) decomposed into shapelet functions were used to create the lensed background objects. There were 9350 such sources with redshifts and separate shapelet coefficients in four bands. The visible galaxies associated with the lens were simulated using an analytic model for the surface brightness of these galaxies. In particular, the Sérsic profile: $I(R) = I_0 \exp -kR^{1/n_s}$ was used. The parameters employed to simulate the galaxies were the total magnitude, the bulge-to-disc ratio, the disc scale height, and the bulge effective radius. The magnitude and bulge-to-disc ratio are a function of the passband. Each galaxy was provided with an inclination angle between 0° and 80° and random orientation. An elliptical Sérsic profile describes the bulge with an axis ratio randomly sampled between 0.5 and 1. The Sérsic index, n_s, is given by

$$\log(n_s) = 0.4 \log\left[\max\left(\frac{B}{T}, 0.03\right)\right] + 0.1x, \quad (1)$$

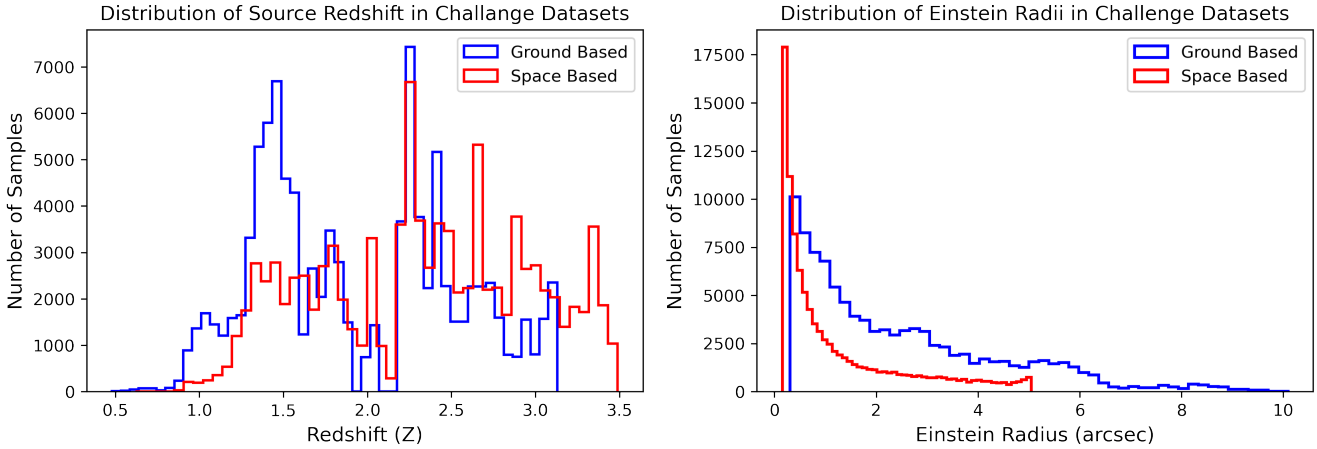

Fig. 1. Distributions of source redshifts and Einstein radii (in arcsec) of simulated gravitational lenses in the Bologna Lens Challenge.

where x is a uniform random number between -1 and 1 and B/T is the bulge-to-total flux ratio. The median redshift of sources in the space-based catalogue was $z_s = 2.35$ and in the ground-based catalogue it was $z_s = 1.81$.

2.1. Space-based

The images for the space-based detector were simulated by Metcalf et al. (2019) to roughly mimic the observations by the Euclid telescope in the visible channel. Metcalf et al. (2019) set the pixel size as 0.1 arcsec, and applied a Gaussian point spread function (PSF) with an FWHM of 0.18 arcsec to simulate the images. The reference band for background and foreground galaxies was the SDSS i, overlapping with the broader Euclid VIS band. The training set consisted of 20 000 images, and the challenge set consisted of 100 000 potential lens candidates.

2.2. Ground-based

The ground-based images consisted of simulated images from four bands (u, g, r, and i), and the reference band was the r band. In the challenge set, 85% of the images were purely simulated. The other 15% were actual images chosen from a preliminary sample of bright galaxies directly from the KiDS survey. These real images were added to the challenge set for more realism. Some images had masked regions where removed stars, cosmic rays, and bad pixels were present. The noise for the mock images was simulated by adding normally distributed numbers with the variance given by the weight maps from the KiDS survey. The example images of a mock simulated lens for the challenge are shown in Fig. 2. For a detailed review of how the data was created, we refer to Metcalf et al. (2019).

An exciting result reported from the challenge was that colour information was crucial for finding strong lenses. All the methods that participated in the challenge performed better on the data from the ground-based observatories, which had four photometric bands (u, g, r, i), than on the data from the space-based detectors, which had a single band. Consequently, it was advised by Metcalf et al. (2019) to add even low-resolution information from other instruments or telescopes to the higher resolution data in one band to improve the detection rates significantly. In other words, multiple bands make a significant difference, and future surveys will perform better if they have information provided in multiple bands. Hence, for our study, we chose the data from the ground-based observatories with four photometric bands (u, g, r, i) to study the attention-based models' ability to detect strong gravitational lenses. Since we are also interested in exploring the transformer architecture's optimisation and analysing the transformers' performance, a better data structure was preferred to compare transformer models.

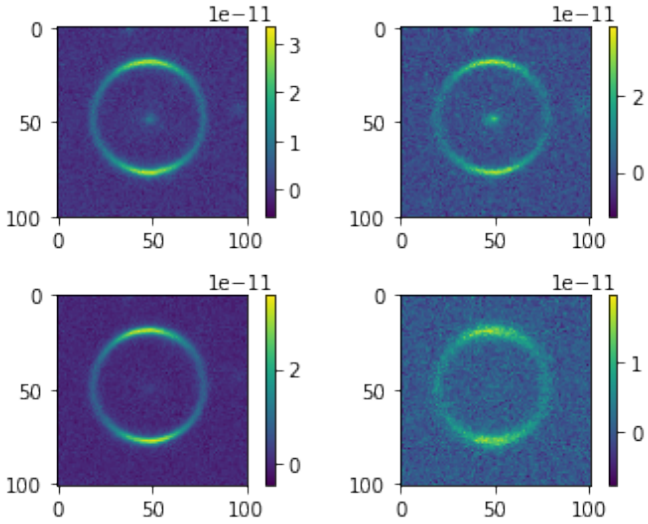

Fig. 2. Typical image of a mock simulated lens for the challenge. Bands are shown in the following order: u (*top left*), g (*top right*), r (*bottom left*), and i (*bottom right*).

2.3. Data pre-processing

The simulated datasets of the Ground-Based Bologna Strong Gravitational Lens Finding Challenge were provided in the FITS format and were available for download to the public[1]. The challenge datasets contained 100 000 potential strong lens candidates, and the training set contained 20 000 images along with other information, such as the Einstein area in rad^2 and number of pixels in the lensed image above $1 \times \sigma$. In this work we did not use additional information about the images. We only used whole images (101×101 pixels) in all four photometric bands (u, g, r, i) as an input to the model and information about the lens present or not as the desired output for training the models. During training

[1] http://metcalf1.difa.unibo.it/blf-portal/gg_challenge.html

the 20 000 images were split into two parts. We used a dataset of 18 000 to train the network, which was used for validation. Before training the models, each image was re-scaled and rotated by $n\pi/2$, where $n \in (0, 1, 2, 3)$, to enrich the dataset.

3. Methodology

3.1. Convolution neural networks

The concept of using CNNs to analyse image-like data was first proposed by Lecun et al. (1998). However, a breakthrough for image recognition by CNNs did not happen until Krizhevsky et al. (2012) created an architecture that won the ImageNet Large-Scale Visual Recognition Challenge 2012. Since then, CNNs have been extensively employed in various research disciplines following the proposed architecture. A regular CNN can be thought of as consisting of two parts. The first part consists of convolution layers, and the second part of fully connected layers that resemble the usual artificial neural networks (ANNs). The main advantage of using convolution layers is that they can learn the local spatial correlation in the data, so using multiple convolution layers will help us to detect the features in the data independent of their position (Mallat 2016).

During the training, the input image is convolved with a number of small kernels (or features maps, typically of dimension 3×3). These individual kernels are optimised during training. The final part of the CNN is the fully connected (FC) layers, which resemble the ANNs. They are used to consolidate the information contained in the feature maps to generate the output. However, a convolutional neural network is restricted by the size of its kernels to collect spatial information from the data. Hence it may lead to deviations due to the ignorance of global information. Since the CNN models have high complexity and a large number of trainable parameters, they are usually prone to overfitting. In addition, the depth of the CNN models is limited by the vanishing gradient effect, where the gradient of the CNN layers vanishes as we go deeper.

3.2. Self-attention

The introduction of attention mechanisms in machine learning has the potential to revolutionise machine learning, and it has been found particularly useful in NLP. Depending on the task at hand, various attention mechanisms can be employed. Among them, self-attention is one of the highly used attention mechanisms for image analysis. For a review of various attention mechanisms, we refer to Yang (2020); Niu et al. (2021). During the application of self-attention, each point in the feature map generated by the convolution layer is considered a random variable, and the paring covariances are determined, so that the value of each prediction can be improved or minimised based on its similarity to other points in the feature map. In other words, the central idea of self-attention is to assign relative importance to the features of the input based on the input itself.

In general, the attention function can be defined mathematically as

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2}$$

where Q, K, V are vectors and $\sqrt{d_k}$ is the dimension of the vector key (K). The softmax function, by definition, is the normalised exponential function that takes an input vector of K real numbers and normalises it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. As we compute the normalised dot product between the query (Q) and the key (K), we get a tensor (QK^T) that encodes the relative importance of the features in the key to the query (Vaswani et al. 2017). For self-attention, the vectors (Q), (V), and (K) are identical. Hence multiplying the tensor (QK^T) by vector (V) results in a vector that encodes the relative importance of features inside the input vector.

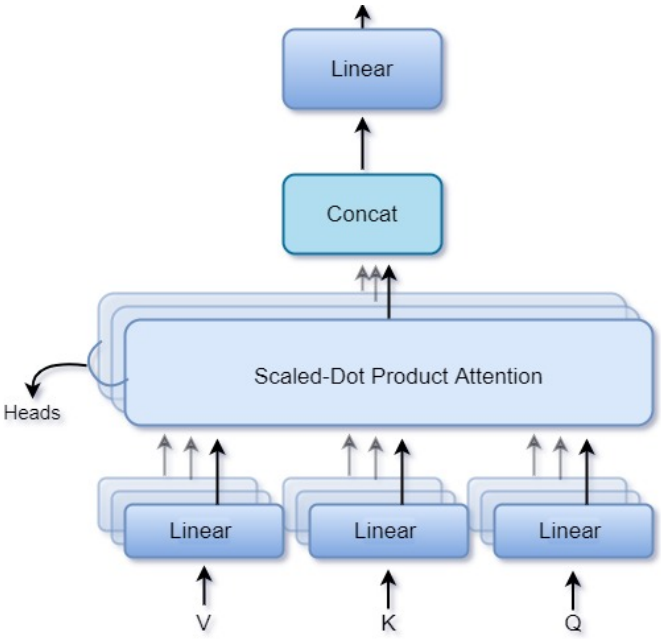

Fig. 3. Scheme of the multi-head attention layer.

A physical interpretation of self-attention applied to feature vectors can be thought of as filtering the input features based on the correlation in the input. The structure of a multi-head attention layer is given in Fig. 3. It is possible to provide the self-attention with more power by creating several layers and dividing the input vector into smaller parts (H, number of heads). Each attention layer is called a head, which applies self-attention to one part of the divided input.

3.3. Positional encoding

If we pass the input directly to the attention layers, the input order or the positional information is lost as transformer models are permutation invariant. So to preserve the information regarding the order of features, we use positional encoding, and the lack of positional encoding will lower the performance of a transformer model. Following the work of Vaswani et al. (2017), we used fixed positional encoding defined by the function

$$\text{PE}_{(\text{pos},2\text{i})} = \sin\left(\text{pos}/12\,800^{\frac{2\text{i}}{d_{\text{model}}}}\right), \tag{3}$$

$$\text{PE}_{(\text{pos},2\text{i}+1)} = \cos\left(\text{pos}/12\,800^{\frac{2\text{i}}{d_{\text{model}}}}\right), \tag{4}$$

where pos is the position, i is the dimension, and d_{model} is the dimension of the input feature vector. Each dimension of the positional encoding corresponds to a sinusoid function. For a detailed description of positional encoding and its importance we refer to Vaswani et al. (2017); Liutkus et al. (2021); Su et al. (2021); Chen et al. (2021).

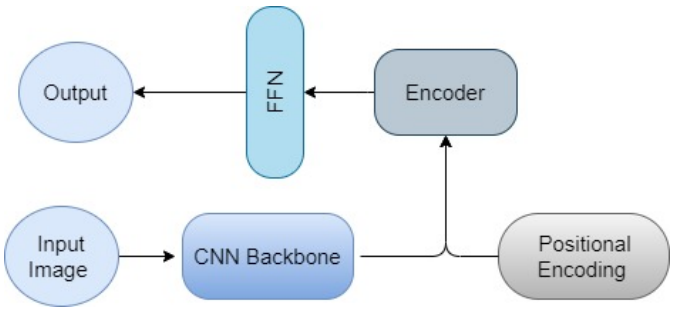

Fig. 4. Scheme of the architecture of the transformer encoder.

3.4. Transformer encoder

The Transformer models we constructed to detect SLs were inspired by the DEtection TRansformer (DETR) created by Facebook (Carion et al. 2020). As shown in Fig. 4, the transformer encoder has a very simple architecture and contains three main parts, which are desctibed below.

The first component of the architecture is a simple CNN used to extract the features from the image. The output from the CNN backbone is a vector with dimensions H×W×D, where D is the number of filters in the last convolution layer. The encoder demands a sequence as input, hence we have to reshape the output of the CNN to a D×HW feature map. As mentioned above, the transformer architecture is permutation-invariant, hence for the second component we add the output of the CNN backbone with fixed positional encoding before processing it to the transformer encoder layer. After the CNN backbone, we process the self-attention-based encoder layers, and filter the relevant features extracted by the CNN. The encoder layer has a standard architecture and consists of a multi-head self-attention module and a feed-forward network (FFN). The third part of the model is a FFN that is similar to the regular CNNs and that learns the features filtered by the encoder layers. The model's output is a single neuron with a sigmoid activation function that predicts the probability that the input image is a lens.

We created 21 encoder models with different structures to study how the hyperparameters in the encoder will affect the model's performance. We used the exponential linear unit (ELU) function as the activation function for all the layers in these models. We initialise the weights of our model with the Xavier uniform initialiser, and all layers are trained from scratch by the ADAM optimiser with the default exponential decay rates (Glorot & Bengio 2010; Kingma & Ba 2015).

3.5. Lens detector

Among the created encoder models, the best performance was given by the encoder model that uses a CNN backbone similar to the LASTRO-EPFL model, from the Bologna Lens Challenge (Schaefer et al. 2018). Here we present the two best architectures: Lens Detector 15 and Lens Detector 16, which outperformed all the other models during our analysis. Lens Detector 15 was first trained for 300 epochs with an initial learning rate of $\alpha = 10^{-4}$ and again trained for another 100 epochs starting with a learning rate of $\alpha = 10^{-5}$. This version of the lens detector gave high scores in all three evaluation metrics for the challenge. The architecture of Lens Detector 15 is given in Fig. 5. In the spirit of reproducible research, our code for Lens Detector 15 is publicly available[2].

[2] https://github.com/hareesht23/Lens-Detector

Lens Detector 16 was created by stacking two Lens Detector 15 models in parallel and combining their outputs through an additional dense layer connected to a single neuron to give the output. The architecture of Lens Detector 16 is shown in Fig. 6. Lens Detector 16 was first trained for 100 epochs with an initial learning rate of $\alpha = 10^{-4}$ and again trained for another 100 epochs starting with a learning rate of $\alpha = 5 \times 10^{-5}$. Furthermore, the model was trained for 50 epochs with $\alpha = 10^{-5}$ and after that with $\alpha = 5 \times 10^{-6}$ for another 200 epochs.

We created the Space Lens Detector model to identify strong lenses from the space-based dataset. The Space Lens Detector has a similar structure to Lens Detector 15. The only difference is the use of four heads in the encoder layers. The model was trained with an initial learning rate of $\alpha = 10^{-4}$ using the ten-fold validation and iterating for 20 epochs in each fold.

3.6. Metrics for evaluation

We first start with a brief overview of the performance metrics we used to quantify the performance of the lens classification. We used classification accuracy as the metric to compare the created transformer models. The classification accuracy is calculated as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{5}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives. Apart from the classification accuracy, another popular figure of merit for a classifier is the area under the receiver operating characteristic (AUROC) curve (Metcalf et al. 2019). The receiver operating characteristic curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) as a function of the threshold. The TPR is the ratio of detected lenses to the total number of lenses:

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{6}$$

The TPR measures how well the classifier detects lenses from the whole population of objects. The FPR can be understood as a contamination rate in the classification and defined as the fraction of non-lens images incorrectly identified as lenses:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{7}$$

The AUROC assesses the overall ability of a classifier to distinguish between classes. A perfect classifier will have AUROC = 1.0 with TPR = 1.0 and FPR = 0.0 for any threshold, whereas a random classifier will have AUROC = 0.5 with TPR = FPR for any threshold. For the Bologna Lens Challenge the participants were instructed to optimise AUROC rather than the accuracy. In addition, two more figures of merit were also considered for the competition, which are TPR_0 and TPR_{10}. The TPR_0 is defined as the highest TPR reached, as a function of the p threshold, before a single false positive occurs in the test set of 100 000 cases. This is the point where the ROC meets the FPR = 0 axis. If the classifier assigns a high probability for a non-lensed image to be a lens, even for one case, the TPR_0 will go low. This means that the TPR_0 parameter measures the confidence in the purity of the samples identified by a model. Similarly, the TPR_{10} is defined as the TPR at the point where fewer than ten false positives are made, which is also a measure of confidence in the true samples mined out by a model with a slight impurity. A high TPR_0 and TPR_{10} indicate that the classifier can clearly distinguish between lensed and non-lensed images.

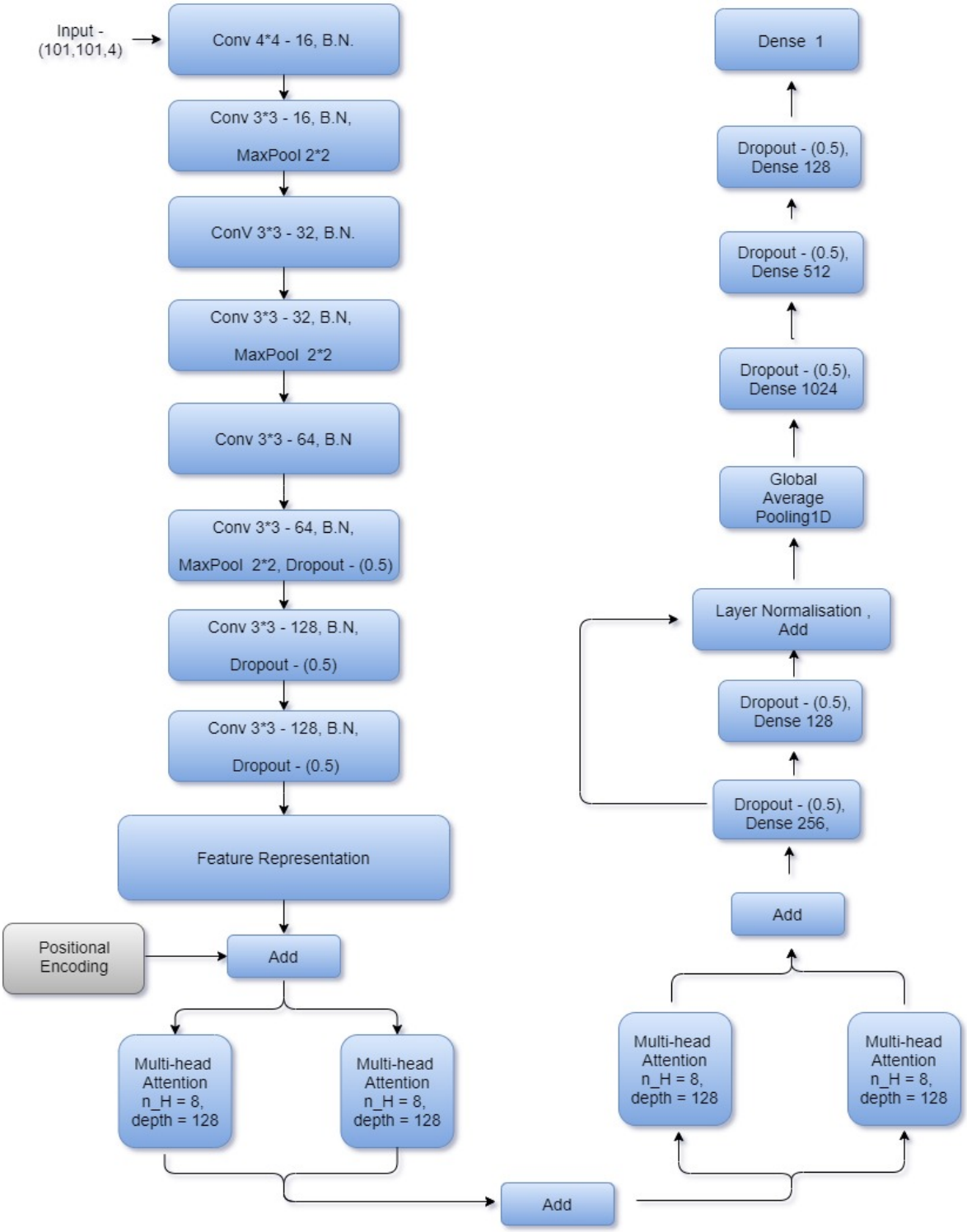

Fig. 5. Scheme of the architecture of Lens Detector 15.

4. Results

We created 5 convolution models to use as the backbones for the encoder models and 21 encoder models to study how the hyperparameters of the encoder layer affect the performance. Since each architecture was implemented as a regression model, a probability of 0.5 was set as the threshold for classifying an image as a lens or not. Thus, input images with a prediction value less than 0.5 were classified as non-lensed images labelled zero and vice versa. Table 1 describes the architecture and total accuracy, AUROC, TPR_0, and TPR_{10} of all created models.

Among the created encoder models the highest accuracy was achieved by Lens Detector 15 and the highest AUROC, TPR_0, and TPR_{10} were achieved by Lens Detector 21, Lens Detector 16, and Lens Detector 9, respectively. From the presented models here, we would like to highlight Lens Detector 15 as the best model since it performs well in all categories and has the

Table 1. Architecture, accuracy, AUROC, TPR_0, and TPR_{10} of all the models in chronological order of creation.

Model name	Model structure	Accuracy	AUROC	TPR_0	TPR_{10}
CNN 1	5 CNN Layers	88.21	0.951	0.000	0.07
CNN 2	4 CNN Layers	86.74	0.915	0.000	0.4
CNN 3	8 CNN Layers	88.51	0.968	0.033	0.37
CNN 4	3 CNN Layers	88.49	0.956	0.000	0.68
CNN 5	25 CNN Layers	91.26	0.974	0.004	0.004
Lens Detector 1	CNN 1+1 H_{16}+1(E)	89.57	0.961	0.000	0.643
Lens Detector 2	CNN 2 + 1 H_{16} + 1(E)	88.13	0.950	0.001	0.001
Lens Detector 3	CNN 2 + 2 H_{16} + 1(E)	88.00	0.962	0.018	0.018
Lens Detector 4	CNN 2 + 2 H_{32} + 1(E)	88.12	0.952	0.121	0.124
Lens Detector 5	CNN 2 + 4 H_{64} + 2 (E)	88.46	0.955	0.125	0.133
Lens Detector 6	CNN 2 + 4 H_{128} + 4(E)	89.51	0.957	0.003	0.004
Lens Detector 7	CNN 3 + 8 H_{128} + 2(E)	91.45	0.968	0.000	0.410
Lens Detector 8	CNN 4 + 2 H_{384} + 2 (E)	89.43	0.954	0.000	0.758
Lens Detector 9	3 CNN Layers + 2 H_{384} + 2 (E)	89.61	0.959	0.000	0.789
Lens Detector 10	5 CNN Layers + 8 H_{128} + 2 (E)	90.58	0.970	0.180	0.23
Lens Detector 11	5 CNN Layers + 8 H_{128} + 4 (E)	90.45	0.966	0.219	0.34
Lens Detector 12	8 CNN Layers + 8 H_{128} + 4 (E)	89.82	0.960	0.040	0.680
Lens Detector 13	8 CNN Layers + 8 H_{128} + 4 (E)	91.94	0.975	0.175	0.525
Lens Detector 14	8 CNN Layers + 8 H_{128} + 4 (E)	91.95	0.975	0.002	0.539
Lens Detector 15	8 CNN Layers + 8 H_{128} + 4 (E)	92.99	0.978	0.140	0.48
Lens Detector 16	16 CNN Layers + 8 H_{128} + 8 (E)	90.97	0.962	0.225	0.24
Lens Detector 17	16 CNN Layers + 8 H_{128} + 8 (E)	92.19	0.973	0.00	0.717
Lens Detector 18	16 CNN Layers + 8 H_{128} + 8 (E)	92.09	0.976	0.113	0.590
Lens Detector 19	16 CNN Layers + 16 H_{128} + 8 (E)	90.03	0.961	0.114	0.115
Lens Detector 20	25 CNN Layers + 8 H_{128} + 4 (E)	91.26	0.973	0.212	0.223
Lens Detector 21	8 CNN Layers + 8 H_{128} + 4 (E)	92.79	0.98	0.00	0.64

Notes. The encoder models are named 'Lens Detector' followed by a number. The model structure describes if the model uses transfer learning in the CNN backbone or not. Generally, the term 'XH_Y' in the model structure means there are x heads with dimension y in one encoder layer. Similarly, the term '$Z(E)$' denotes that there are Z encoders in the structure.

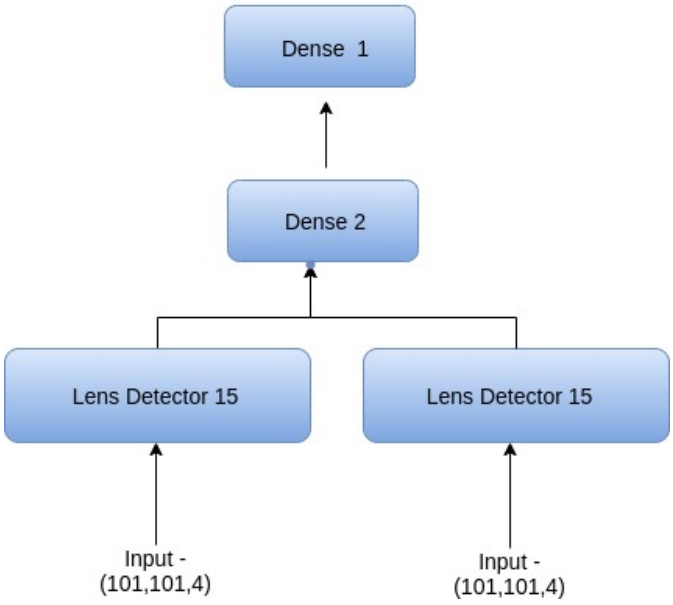

Fig. 6. Scheme of the architecture of Lens Detector 16.

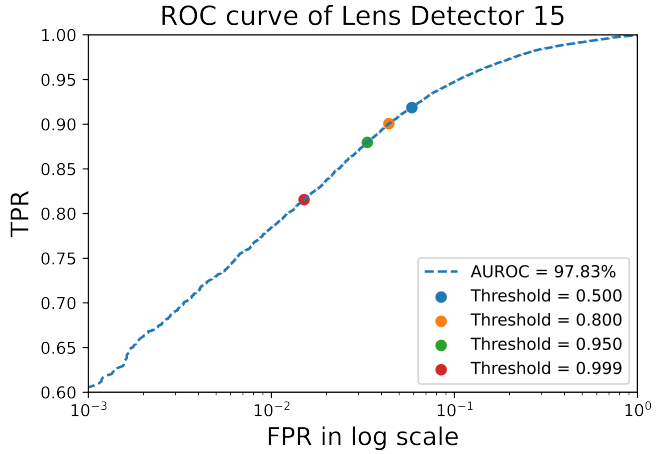

Fig. 7. Receiver operating characteristic (ROC) curve of Lens Detector 15.

highest classification accuracy. The receiver operator characteristic (ROC) curve of Lens Detector 15 is shown in Fig. 7. Similarly, Lens Detector 13, Lens Detector 18, and Lens Detector 20 can also be considered highly performing classifiers. All of these models scored an AUROC equivalent to the second-best model that participated in the challenge and a better TPR_0 and TPR_{10} compared to all other models that participated in the challenge. The ROC curves of all the encoder models are presented in Fig. A.1.

Even though we chose 0.5 as the threshold for identifying a candidate as a strong lens, in reality such a threshold is not practical since the number of lenses to visually inspect after the run

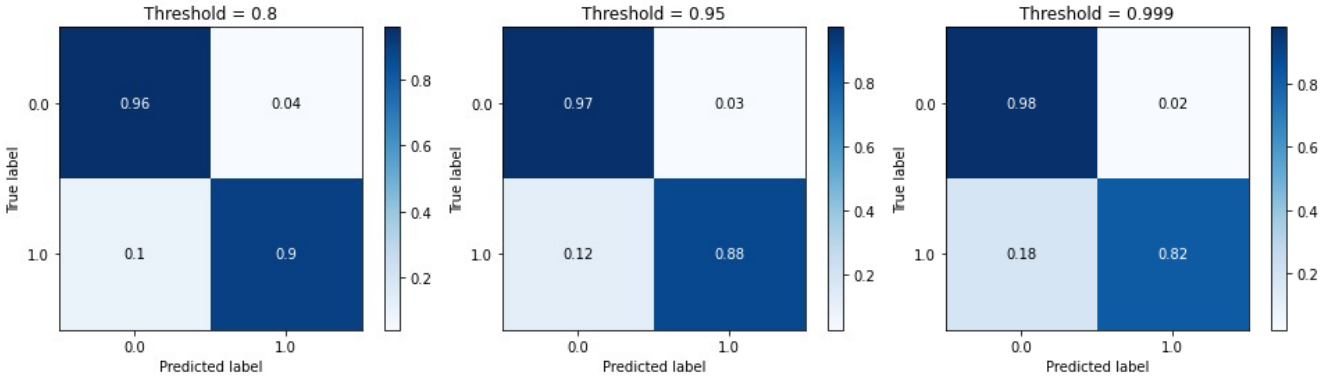

Fig. 8. Confusion matrix of Lens Detector 15 plotted for various thresholds. Class 0 represents the non-lensed images, and Class 1 represents the lensed images. *Lower right*: square in each confusion matrix represents the true positives for which Lens Detector 15 identified strong lenses correctly. *Upper left*: square in each confusion matrix represents the true negatives for which Lens Detector 15 identified non-strong lenses correctly. *Lower left*: square in each confusion matrix represents the false negatives or the missed true lenses by Lens Detector 15. *Upper right*: square in each confusion matrix represents the false positives or the non-lenses identified by Lens Detector 15 as strong lenses.

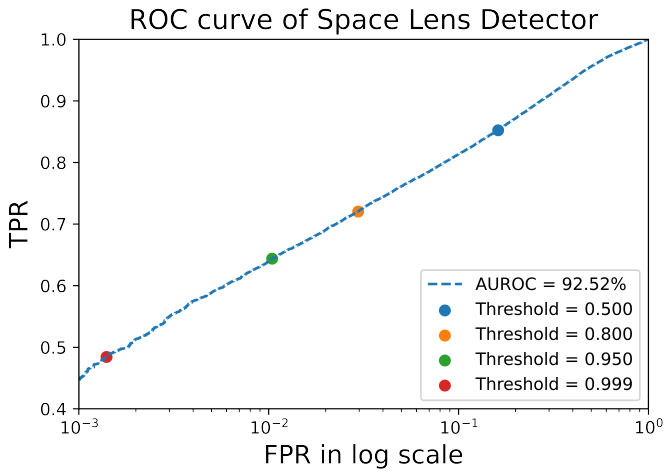

Fig. 9. ROC curve of the Space Lens Detector.

of the network could be unrealistically high. In order to validate the performance of Lens Detector 15, we plotted the confusion matrix of the lens detector with varying thresholds (see Fig. 8). For the ground-based data, we were able to mine out more than 80% of the true lenses with a threshold as high as 0.999.

To identify strong lenses from the space-based dataset, we created the Space Lens Detector, which scored an AUROC = 0.925. The corresponding AUROC is greater than the second-best AUROC reported in the Bologna Lens Challenge and a little below the top AUROC (0.93) of the Bologna Lens Challenge. The Space Lens detector scored a TPR_0 of 0.039 and TPR_{10} of 0.166, which is comparable to the performance of the other machine learning techniques that participated in the challenge. Since it is clear that the attention-based encoder models can identify the SLs from single-band images, we did not try to improve the scores further. The ROC curve of the Space Lens Detector is shown in Fig. 9.

Similarly to the models that participated in the challenge, encoder models also performed better on the data from the ground-based observatories, which had four photometric bands (u, g, r, i), compared to the data from the space-based detectors, which had a single band. Our results support the argument presented in Metcalf et al. (2019) that multiple bands make a significant difference and improve the detection.

5. Discussion

5.1. Transformers and models from the Bologna Lens Challenge

The Bologna Lens Challenge was intended to improve the efficiency and biases of tools used to find strong gravitational lenses on galactic scales. It was clear from the challenge that automated methods such as CNNs and SVM have a clear advantage compared to conventional methods. The performance of all these models was evaluated using AUROC, TPR_0, and TPR_{10} scored on the challenge set. An SVM model named Manchester SVM won the competition in the TPR_0 category with a score of 0.22. (Metcalf et al. 2019; Hartley et al. 2017). The model named CMU Deep Lens received an AUROC of 0.981 and a TPR_{10} score of 0.45, the highest in their respective categories, and thus won the competition (Metcalf et al. 2019; Lanusse et al. 2017). Another variant of the model, named CMU Deep Lens, also received an AUROC of 0.98 during the challenge. These models were 46 layers deep ResNets with around 23×10^6 parameters (Lanusse et al. 2017). Another model worth mentioning from the challenge is LASTRO EPFL, an eight-layer CNN that won the competition for the space-based dataset in the AUROC category. For a detailed look at the models that participated in the challenge we refer to Hartley et al. (2017) for the SVM, Lanusse et al. (2017) for CMU-DeepLens, and Schaefer et al. (2018) for LASTRO EPFL.

We would like to compare the performance of these models to the performance of the encoder models to exhibit the advantages of encoder models over the CNNs and SVM models. As mentioned earlier, we focused on the data from ground-based observatories. Here we are only comparing the performance of the created encoder models and that of the models that participated in the challenge only for the ground-based observatories data. The values reported in Metcalf et al. (2019) are used here for the comparison.

During the challenge, the TPR_0 was used to strongly penalise the classifiers with discrete ranking because their highest classification level was not conservative enough to eliminate all false positives, and they were likely to get $\mathrm{TPR}_0 = 0$. For the other models that participated in the challenge, maximising the TPR_0 was a tough challenge, also for encoder models. However, the encoder models performed very well compared to the CNN models that participated in the challenge. The results of TPR_0 for the

Table 2. Comparison of encoder models and models that participated in the Bologna Lens Challenge, listed in decreasing order of TPR_0.

Name	AUROC	TPR_0	TPR_{10}	Model type
Lens Detector 16	0.962	0.225	0.24	Transformer
Manchester SVM	0.93	0.220	0.35	SVM/Gabor
Lens Detector 11	0.966	0.219	0.34	Transformer
Lens Detector 15	0.978	0.140	0.48	Transformer
CMU-DeepLens Resnet-ground3	0.98	0.09	0.45	CNN
LASTRO EPFL	0.97	0.07	0.11	CNN

Table 3. Comparison of encoder models and models that participated in the Bologna Lens Challenge, listed in decreasing order of TPR_{10}.

Name	AUROC	TPR_0	TPR_{10}	Model type
Lens Detector 9	0.959	0.00	0.789	Transformer
Lens Detector 8	0.954	0.00	0.758	Transformer
Lens Detector 17	0.973	0.00	0.717	Transformer
CMU-DeepLens Resnet-ground3	0.98	0.09	0.45	CNN
Manchester SVM	0.93	0.220	0.35	SVM/Gabor
LASTRO EPFL	0.97	0.07	0.11	CNN

top three encoder models and the top three models that participated in the challenge are listed in Table 2. We would like to note two models; Lens Detector 16 achieved a TPR_0 of 0.225 and Lens Detector 11 reached a TPR_0 of 0.219, which are very high compared to the CNNs that participated in the challenge.

The next parameter used to evaluate the models in the challenge was TPR_{10} for which the encoder models showed a high range of supremacy over all other models that participated in the challenge. Particularly, Lens Detector 9 achieved $TPR_{10} = 0.79$. The results of TPR_{10} for the top three encoder models and the top three models that participated in the challenge are listed in Table 3. Three of our models were able to score a TPR_{10} above 0.70, which is very high compared to the top TPR_{10} reported during the challenge. Table 1 clearly shows that most encoder models achieved a higher score in this category compared to the other models that participated in the challenge.

Now looking at the third parameter of merit used in the Bologna Lens Challenge, which is the AUROC, we can see that Lens Detector 21 was able to reach the highest reported AUROC in the Bologna Lens Challenge (Metcalf et al. 2019). The top three encoder models and the top three models that participated in the challenge that scored the highest AUROC are listed in Table 4. However, the CMU-DeepLens, was a 46 layer deep ResNet with around 23×10^6 parameters (Lanusse et al. 2017), whereas Lens Detector 21 had only 3×10^6 parameters and achieved an AUROC of 0.9809, which is very close to the performance of CMU Deep Lens (AUROC = 0.9814).

5.2. Insights into transformers

An initial glance at the results in Table 1 shows that encoder models perform better than CNN models. However, the encoder models depend on the CNN backbone to extract the features, and as a result the performance of the encoder models depends upon the CNN backbone. A detailed look at the results indicates that

Table 4. Comparison of encoder models and models that participated in the Bologna Lens Challenge, listed in decreasing order of AUROC.

Name	AUROC	TPR_0	TPR_{10}	Model type
CMU-DeepLens Resnet-ground3	0.981	0.09	0.45	CNN
Lens Detector 21	0.981	0.00	0.64	Transformer
CMU-DeepLens Resnet-Voting	0.980	0.02	0.10	CNN
Lens Detector 15	0.978	0.140	0.48	Transformer
Lens Detector 18	0.976	0.113	0.59	Transformer
LASTRO EPFL	0.97	0.07	0.11	CNN

the encoder model is only good as its CNN backbone. However, the encoder model always performs better than its CNN backbone. For example, the lowest accuracy achieved among the encoder models was for Lens Detector 3, which has better performance than CNN 2. A similar trend can be observed for other encoder models, which use trained CNNs as their backbone and perform better than the trained CNNs. These observations show that the encoder models can achieve accuracy that is better by a small percentage than a CNN with the same number of convolution layers.

However, an interesting question that should be addressed is what happens if we use deeper CNN backbones. We compare the performance of a deeper CNN and an encoder model with a deeper CNN backbone. Model CNN 5 with 25 convolution layers and Lens Detector 20 with the same number of convolution layers give same AUROC (0.97) and the same accuracy (0.91). We can see that the two models perform equally well. Using self-attention in deeper CNNs did not significantly improve the AUROC or accuracy. However, the TPR_0 score and TPR_{10} score of Lens Detector 20 (0.212 and 0.223, respectively) is higher compared to CNN 5 (0.004 and 0.004, respectively). One probable reason why self-attention does not improve the accuracy and AUROC of deeper CNNs is that the CNN backbone will learn more about the image's micro-scale features in deeper layers. Hence, the model will miss the long-range correlations of the original image found by the encoder layer.

In this section we speak of the number of hyper-parameters in the encoder layer. On analysing the results from the encoder models, we can see that increasing the number of heads and the depth of the encoder increases the model's performance. However, increasing the number of heads and the depth of the encoder also increases the number of trainable parameters in the model. During the training period, it was found that increasing the number of parameters in the encoder layer helps the model to learn faster. This points to an exciting aspect of the encoder models. The encoder's performance is proportional to the number of trainable parameters in the encoder layer or specifically in the multi-head attention layer. The higher the number of trainable parameters, the better the learning curve and performance. However, the performance saturates beyond a limit for a given CNN backbone.

Another interesting observation was the difference in the number of trainable parameters in the CNNs and self-attention-based encoder models. For example, an eight-layer CNN will have $4{\times}10^6$ parameters, whereas the encoder with eight CNN layers and four encoder layers with eight heads each will have $3{\times}10^6$ parameters. In CNNs, most of the parameters come from the connections between the flattened output of the last convolution layer to the following dense layer. The weights in these layers

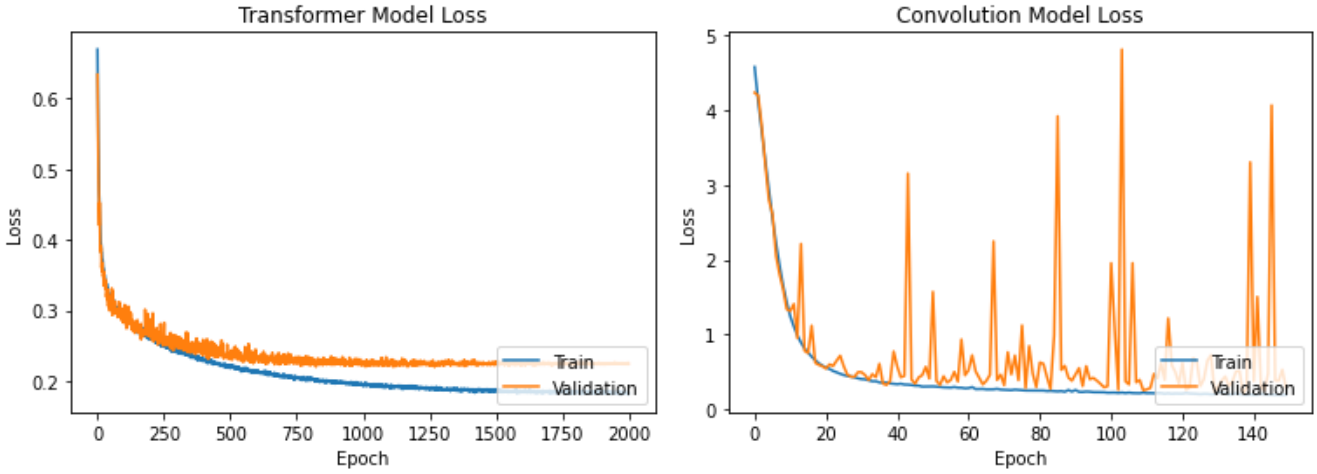

Fig. 10. Variation of loss function with epochs for Lens Detector 13 and CNN 3, respectively. Lens Detector 13 uses CNN 3 as its CNN backbone.

help the CNNs to learn the features of the image. However, for a transformer network most of the trainable parameters are in the attention layers, which are only trying to learn the long-range correlations in the data and effectively act as a filter. This is one of the reasons why transformer networks can prevent overfitting. However, transformer models did not show any advantages over CNNs for the time taken to test and train the models.

We also tried different learning rates from different ranges to find the optimal learning rate. We found that increasing the learning rate above 0.001 considerably reduced the performance of the lens detectors. With an initial learning rate of 0.01, the models were not able to learn the features of the lenses from the training set. The optimal learning rate for our model was found to be between $\alpha = 5 \times 10^{-5}$ and $\alpha = 2 \times 10^{-4}$; we chose $\alpha = 10^{-4}$ as the initial learning rate for our models.

Another striking feature worth pointing out is that, unlike convolution layers increasing the number of parameters in the encoder layers has a very slight effect on overfitting the model. Since the self-attention layers act as the filters for features extracted by CNN, an increase in the number of parameters in the encoder layers helps the models to filter the features faster and effectively without causing the overfitting of the model. The effect of self-attention layers in filtering and smoothing the learning curve can also be seen in comparing the loss curve of the CNN and encoder models presented in Fig. 10.

We also tried transfer learning by using an already trained CNN as the backbone of the encoder model. Surprisingly, however, the encoder models that do not use transfer learning performed slightly better than the models that use transfer learning. Since a trained CNN model has already learned to extract specific features of an image, the encoder model with that CNN backbone is restricted to minimise the loss function in only a small part of the hyperspace. So, the self-attention layers can only filter the features and improve the accuracy by a small percentage (e.g. CNN 2 86.74% and Lens Detector 2, with CNN 2 as the backbone, 88.13%). Nevertheless, for a model without transfer learning, there is a possibility for the CNN part in the encoder model to learn more features than a solo CNN about the image and improve the accuracy. For example, Lens Detector 7, which used the trained CNN 3 as its backbone, scored an accuracy of 91.45%. In contrast, Lens Detector 15, which used a CNN backbone similar to CNN 3 and without any transfer learning, scored an accuracy of 92.99%. However, this result cannot be generalised since it also depends on the trained CNN backbone.

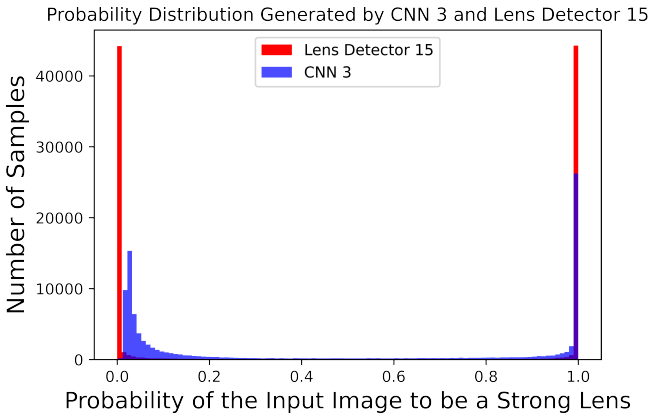

Fig. 11. Comparison of the output probabilities of CNN 3 and Lens Detector 15 for the ground-based challenge dataset. In this histogram values leaning towards zero represent the lack of a strong lens in the image, and values leaning towards one indicate the presence of a strong lens in the image.

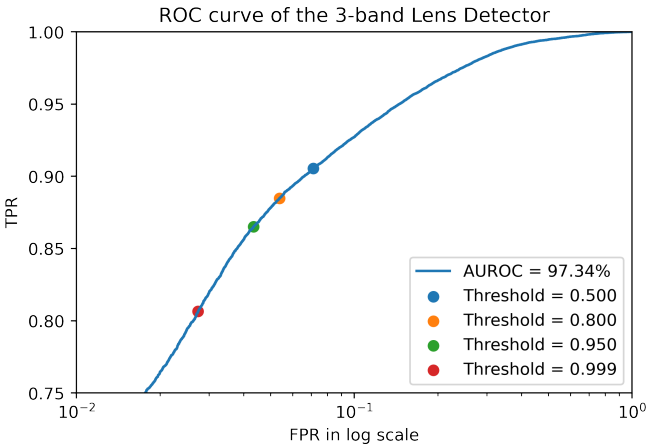

Fig. 12. ROC curve of the 3-band Lens Detector.

An updated version of the CNN models that participated in the challenge and had better scores in every category compared to their previous versions has recently been reported by Magro et al. (2021). They used the same CNNs that participated in the

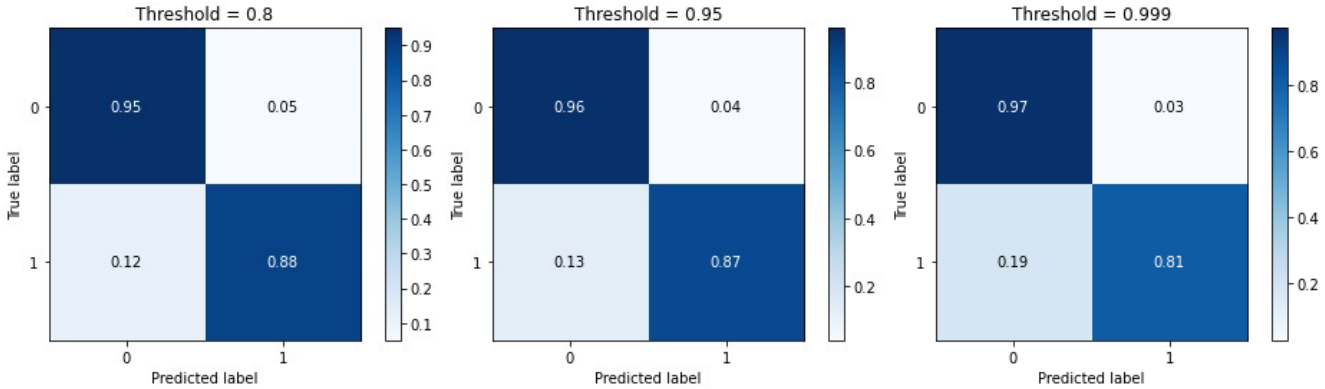

Fig. 13. Confusion matrix of the 3-band Lens Detector on the challenge dataset for various thresholds. Class 0 represents the non-lensed images, and Class 1 represents the lensed images. The *lower right* represents TP in each confusion matrix, the *lower left* represents FN, the *upper left* represents TN, and the *upper right* represents FP.

challenge and retrained the networks with different epochs. Even though the models had improved scores, it was evident from the report that the performance of the models is highly dependent on the number of epochs. In other words, the CNNs reported in the Bologna Lens Challenge had lower stability. We have to monitor the training to achieve better results carefully. In contrast, the encoder models are highly stable compared to the CNNs. As shown in Fig. 10, we were able to train the encoder models up to 2000 epochs without any sign of overfitting. Interestingly, the fluctuations in the validation loss were very stable up to the end.

5.3. Lens detectors for strong lens detection

It is worth pointing out that encoder models can identify SLs and non-SLs better than their CNN counterparts. The probability distribution of finding a lens in the challenge dataset is depicted in Fig. 11. The encoder model can assign a probability for an input to be a lens ($p \approx 1$) or non-lens ($p \approx 0$) with greater confidence than the CNN. Furthermore, from Fig. 11, it is clear that the transformer models can approximately mimic a perfect classifier by assigning a probability of 0 to non-lensed images and a probability of 1 to lensed images. This feature of the encoder model will be beneficial and applicable in the upcoming large-scale surveys to narrow down the potential lensing systems with great confidence. Figure 8 shows that Lens Detector 15 was able to identify 82% of the true lenses with a probability greater than 0.999, which explains the peak near the point 1.0 on the x-axis of Fig. 11.

Here, we used all four bands available for training the models. However, the u-band images are usually not used to search for strong lenses because, in the real scenario, the u-band images often have lower image quality. It is also possible that they are not available for fainter galaxies. In the literature, for detecting the strong lenses, the images from the g, r, and i bands are used for training machine learning models. Sometimes along with a three-channel CNN, another single-band CNN is also created, and the combined predictions of these two CNNs are used to shortlist the real lens candidates (Petrillo et al. 2019a,b; Li et al. 2020). To test the adaptability of the encoder model for three bands, we removed the u band and retrained the Lens Detector 15 model from scratch with the images from the g, r, and i bands and named it the 3-band Lens Detector. The retrained Lens Detector 15 achieved an AUROC = 0.974, which is comparable to the AUROC when the u band was present. The ROC curve and the confusion matrix for various thresholds are presented in Figs. 12 and 13. Comparing the confusion matrices in Figs. 8 and 13, we can see that removing the u band slightly reduces the number of true positives for a given threshold since the information available in the u band was gone. For example, for a threshold =0.8, Lens Detector 15 identified 90% of the true positives and the 3-band Lens Detector identified 88% of the true positives. This result also validates the argument presented in Metcalf et al. (2019) to include even low-resolution information in one band to improve the detection rates.

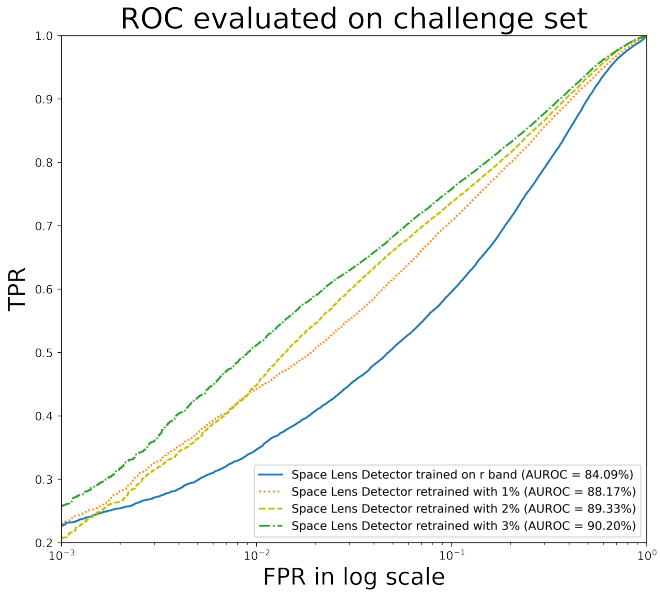

Fig. 14. ROC curve of Space Lens Detector trained on r band and tested on the space-based dataset. Improved ROC curves of the model retrained using 200, 400, and 600 samples from the space dataset are also plotted.

Since we already tested the encoder model on the space-based dataset and obtained comparable results with models that participated in the Bologna Lens Challenge, it is clear that encoder models can be adapted for a single-band analysis. However, another interesting question to be investigated is the ability of the encoder models to find strong lenses from a different data distribution than the model has been trained on. We retrained the Space Lens Detector using the data in the r band from the ground-based data and tested it on the space-based challenge dataset to investigate this aspect. The retrained Space Lens

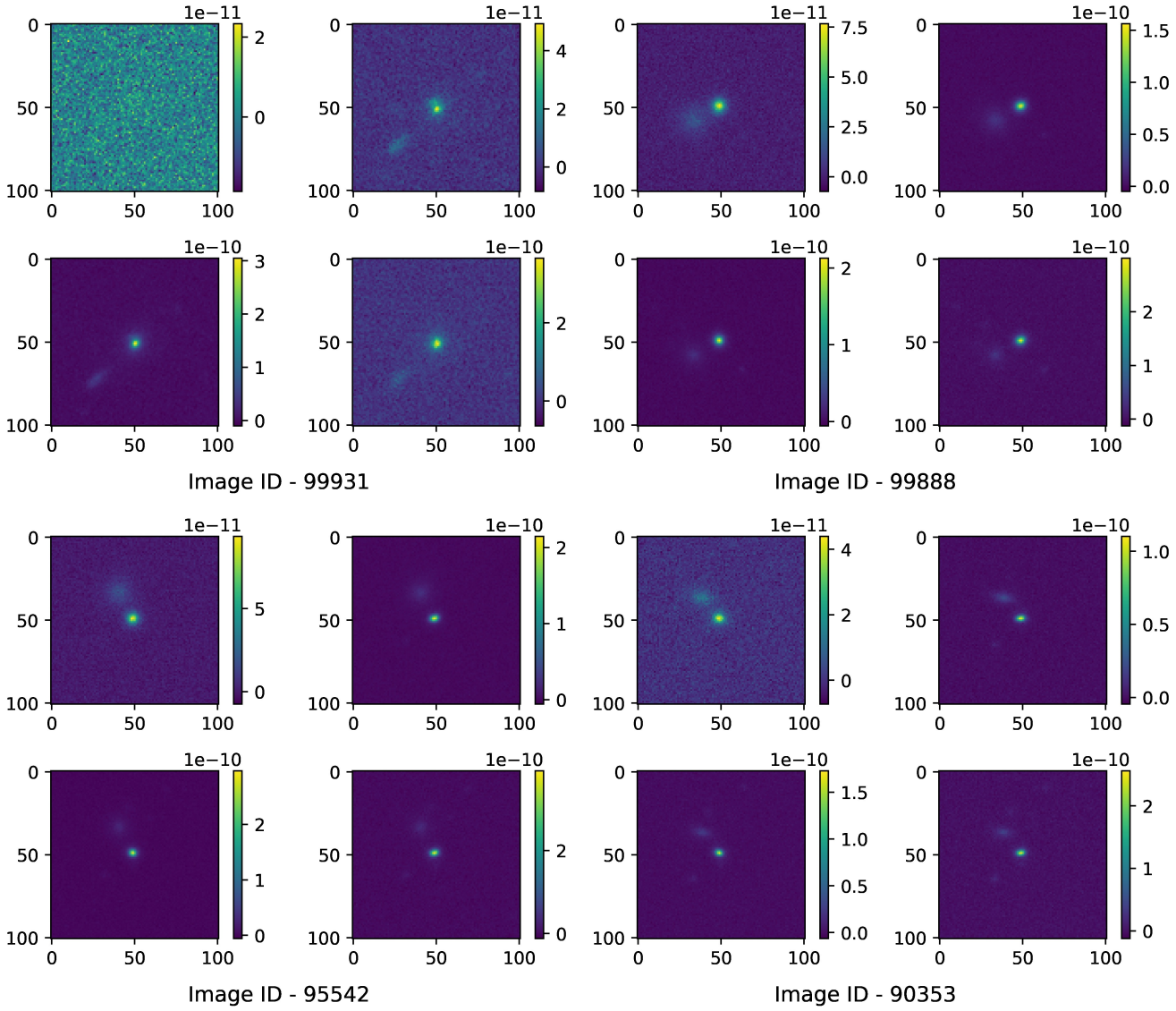

Fig. 15. Four examples of false positives found by the encoder models. The channels shown are *u* (*top left*), *g* (*top right*), *r* (*bottom left*), and *i* (*bottom right*). Image ID from the test data is given below each set of images.

Detector scored AUROC = 0.84, which shows the model has the minimum capacity to distinguish lenses from a different distribution. If we train the network again with 200 samples from the space-based dataset (1% of the space training set), the AUROC improves by 0.88. With 400 samples (2%), AUROC becomes 0.89, and with 600 samples (3%), AUROC improves to 0.902. Each retraining was done independently of the others. The ROC curve of the Space Lens Detector trained with *r* band and was tested on the space-based dataset, and the improved ROC curves of the retrained model are shown in Fig. 14. The capacity of the model to identify the lenses improves if we train the model with very few samples from a different distribution, which indicates the adaptability of the encoder model in the presence of new data distributions. This feature also shows that the encoder models trained on simulated data can be optimised to detect strong lenses from real data using even a small sample of detected lenses.

Even though the encoder model performs better than the convolution models and the other models that participated in the challenge, the encoder models that have been trained here have a slight gap with a perfect classifier. We carefully examined the frequent false positives and the false negatives reported by various encoder models. Some of the images that have been identified as false positives and false negatives are given in Figs. 15 and 16. Looking at these false positives and false negatives, we can see that the encoder models are trying to find whether the input image has an arc-like structure or multiple distorted images. If we suppose the input image has any of these characteristics in at least one of the bands, the detector identifies the image as a strong lens. Similarly, if both these features are missing, then the detector classifies the image as a non-lens. In order to improve the performance of the models, we need the model to be trained on more realistic and complex data.

Since strong gravitational lensing is a rare phenomenon, we need to have a closer look at the cases of false negatives. In real surveys false positives can be filtered out from a candidate sample created by an automated classifier, whereas missing a true SL is not favoured. Looking at the relations between the parameters of the strong lens and the model's performance is a possible way to search for a space of parameters where it is more difficult to find lenses. We sorted the test dataset into ascending order

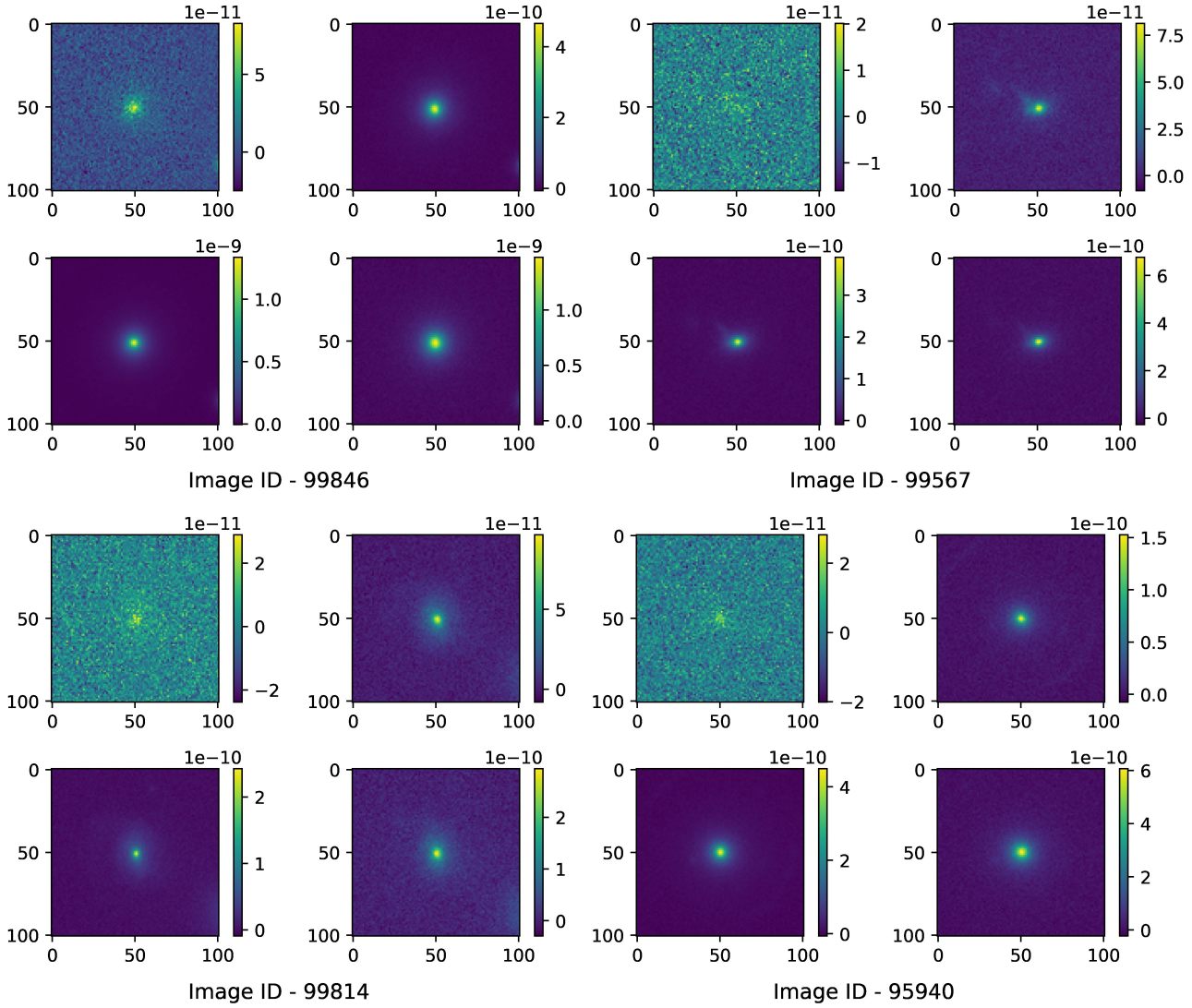

Fig. 16. Four examples of false negatives found by the encoder models. The channels shown are *u* (*top left*), *g* (*top right*), *r* (*bottom left*), and *i* (*bottom right*). Image ID from the test data is given below each set of images.

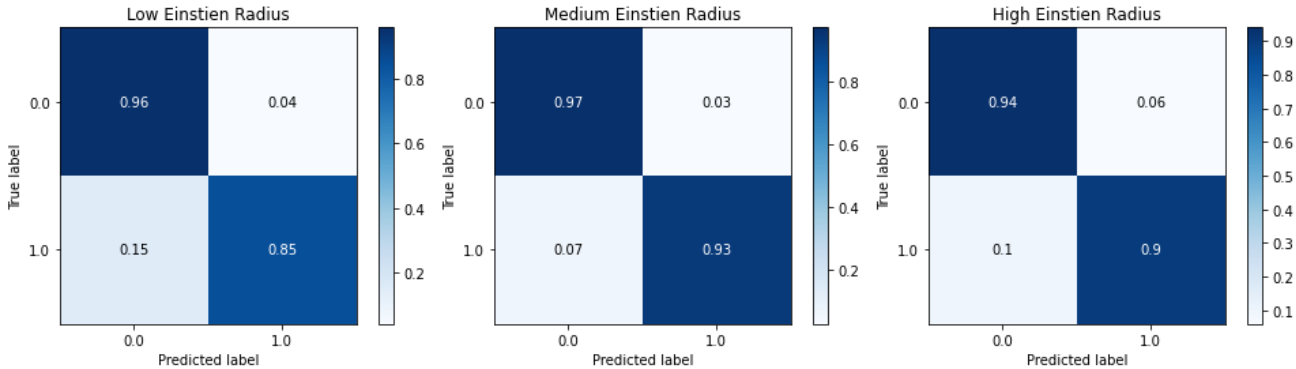

Fig. 17. Confusion matrix of Lens Detector 15 with 0.8 as the threshold plotted for small Einstein radius (0.3011–0.873 arcsec), medium Einstein radius (0.873–3.547 arcsec) and high Einstein radius (3.547–10.08 arcsec). In the confusion matrix, the *lower right* represents TP, the *lower left* represents FN, the *upper left* represents TN, and the *upper right* represents FP.

in Einstein radii and divided it into three subclasses depending on the Einstein radius. The first quartile (25 000) and the third quartile (75 000) were used as the subclass boundaries. Looking at the confusion matrices (threshold = 0.8) from Fig. 17, we can see that the best performance of Lens Detector 15 is for the intermediate bin (0.873–3.547 arcsec). The performance of Lens Detector 15 is low for the first bin (0.3011–0.873 arcsec) and the third bin (3.547–10.08 arcsec). This means that Lens Detector

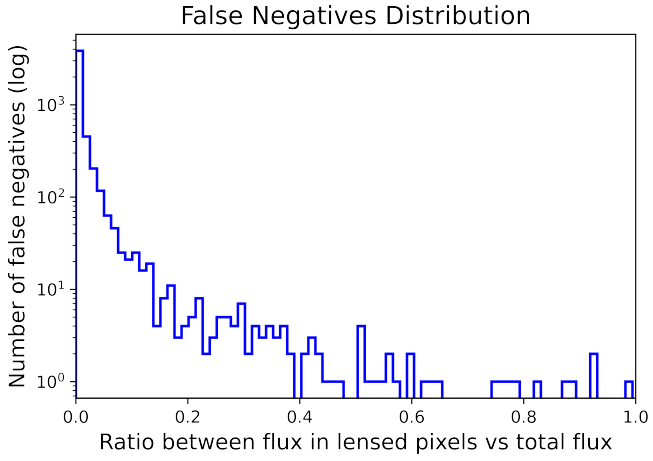

Fig. 18. Distribution of false negatives as a function of the ratio of flux in the lensed pixels to the total flux.

15 has difficulty finding SLs with small and large Einstein radii. A similar result for CNNs has also been reported by Li et al. (2020); Cañameras et al. (2020).

Analysing the parameters of the false negatives, we found that another important parameter that impacts the identification of strong lenses is the ratio of the flux in lensed pixels to the total flux. For a probability threshold of 0.8, we had 4981 false negatives in the Bologna Ground-Based Challenge. All of them had a very low ratio of the flux in lensed pixels to the total flux. Out of 4981 false negative cases, 4775 cases had a flux ratio of the source to the lens lower than 0.1. Similarly, 3667 out of 4981 cases in the sample of false negatives had a flux ratio of the source to the lens lower than 0.01. The Bologna Ground-Based Challenge dataset contained 10,818 true strong lenses with a flux ratio of the source to the lens lower than 0.01. Thus, Lens Detector 15 could identify 66% of the true strong lenses with a very low flux ratio (lower than 0.01). These results indicate that Lens Detector 15 will have trouble distinguishing strong lens candidates for a very low flux ratio between the source and the lens. The reason is that the distortions of the source galaxy due to lensing may not be bright enough to be detected by the models. This result is similar to the results reported by Li et al. (2020); Cañameras et al. (2020) where the CNN models have low performance on fainter SLs samples. The distribution of false negatives as a function of the ratio of flux in the lensed pixels to the total flux is plotted in Fig. 18.

We also would like to comment on another reported CNN model on the Bologna Lens Challenge, which is the LensCNN, achieving a total accuracy of 0.8749 (TP 0.8817 and TN 0.8682). It is the only CNN model where classification accuracy for the Bologna Lens challenge has been reported (Davies et al. 2019). Our results show that all of our encoder models have surpassed the LensCNN in total accuracy. Furthermore, the LensCNN model has also reported an AUROC of 0.96 on the challenge data, which is exceeded by most of our encoder models. In this context, it is worth mentioning that the LensCNN had approximately 10×10^6 parameters. Lens Detectors outperformed the LensCNN with just 3×10^6 of the parameters.

5.4. Performance on real data

Since all our models have been trained on the simulated dataset provided by the Bologna Lens Challenge, it is critical to check if the trained model can identify strong lenses from real data. Ideally, we expect the encoder models to learn the general features of the strong lenses from the simulated data and to retrieve the potential lens candidates from the real data. Recently, Petrillo et al. (2019b) have trained a three-band CNN (g, r, and i bands) and a single-band CNN (r band) on the data simulated using the real images from the KiDS survey and applied it to the KiDS DR4 data to identify potential strong lens candidates. To obtain a reasonable number of true positives and so as not to contaminate the sample with a large number of false positives, they chose 0.8 as the threshold for identifying a candidate as a strong lens for each CNN. Using these criteria, they shortlisted 3500 cases as potential strong lenses, and Petrillo et al. (2019b) conducted a visual inspection to validate these candidates.

The potential candidates were classified into three classes, and each class was assigned a numerical score: Sure lens, 10 points; Maybe lens, 4 points; No lens, 0 points. As a result, the highest score that any candidate can obtain is 70, when all human classifiers think it is undoubtedly a lens. Using these criteria, Petrillo et al. (2019b) shortlisted 1983 potential strong lens candidates from the data selected by the two CNNs. The FITS files, probability scores of the CNNs reported in Petrillo et al. (2019b), and numerical scores of visual inspection for each candidate are available publicly[3], and we chose this dataset to study the performance of the encoder model on real data.

Since the 1983 lens candidates were found together using a single-band CNN and a three-band CNN, some of the candidates found by the single-band CNN were not detected by the three-band CNN. Specifically, 946 candidates were missed by the three-band CNN (which means they were assigned a probability of less than 0.8) in the final sample of 1983 candidates. Similarly, 526 candidates identified by the three-band CNN were missed by the single-band CNN. To analyse the performance of the encoder model, we used the 3-band Lens Detector and tested it on the lens candidates found by Petrillo et al. (2019b). After evaluating the model on real data, we created three classes using the visual inspection score as the reference. Class 0 sources have a low probability of being a lens (score less than 20 out of 70, and predictive value less than 0.8). Class 1 sources have an intermediate probability of being a lens (score between 20 and 50, and predictive value between 0.8 and 0.95). We are highly confident that Class 2 sources are strong lenses (score greater than 50, and a predictive value greater than 0.95). Using the probability scores predicted by the three-band CNN and using the visual scores, we plotted the confusion matrix for the three-band CNN along with the 3-band Lens Detector, which is given in Fig. 19.

Looking at the confusion matrix in Fig. 19, we can see that the encoder model can classify low (Class 0) and high (Class 2) probability cases similarly to a human expert. However, for class 1, which represents the cases with an intermediate probability of being a strong lens, the 3-band Lens Detector performs poorly. This is to be expected since the 3-band Lens Detector is trained as a binary classifier, and from Fig. 11 it is clear that the Lens Detector tends to assign very high or very low probability scores. Since there are three classes and the number of samples in each class is different, in order to compare the three-band CNN and the 3-band Lens Detector, we can calculate the weighted f_1 score by taking the mean of all per class f_1 scores while considering each class's support. Here, support refers to the number of actual occurrences of the class in the dataset. Using the visual scores as a reference, the weighted f_1 score of the three-band CNN is

[3] https://www.astro.rug.nl/lensesinkids/

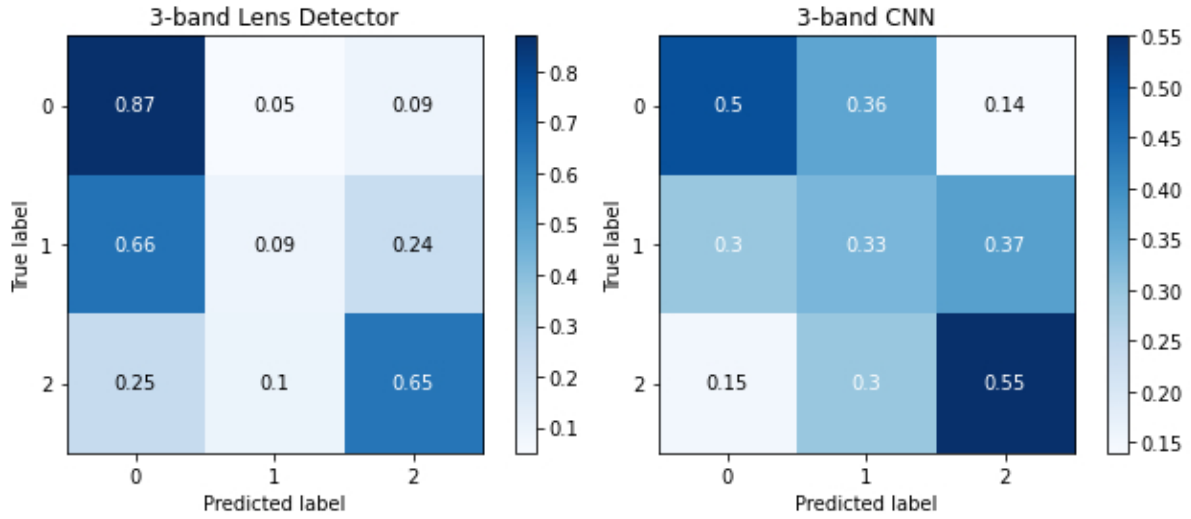

Fig. 19. Confusion matrix of the 3-band Lens Detector and three-band CNN for the classification of the 1983 KiDS DR4 lens candidates (Petrillo et al. 2019b). The candidates are classified into three categories based on the visual inspection scores by Petrillo et al. (2019b), which is treated as the correct classification. The candidates are also classified into three categories based on the probability values generated by the 3-band Lens Detector introduced in this work and three-band CNN from Petrillo et al. (2019b). The overlap between these classifications is shown in the confusion matrices.

0.601, and the weighted f_1 score of the 3-band Lens Detector is 0.822, which indicates that the 3-band Lens Detector is performing similarly to a human visual expert on the shortlisted SL candidates. Here we have assumed that the output probabilities assigned by the three-band CNN and the team of visual experts are independent.

With these results, we cannot claim that the 3-band Lens Detector is better than the three-band CNN presented in Petrillo et al. (2019b) or vice versa since we are testing the model on the already shortlisted candidates by the three-band CNN. However, we can claim that self-attention-based encoder models can detect strong lenses from the real data in competition with the CNNs. Another factor to be noted here is that the 3-band Lens Detector was trained on a complete data distribution compared to the training set of the three-band CNN, which was derived from the actual KiDS DR4 data. As mentioned earlier, the data in the Bologna Lens Challenge used KiDS as a reference, and they did not strictly mimic the data from KiDS. Thus, the data used in the Bologna Lens Challenge have a different data distribution compared to the KiDS DR4 data. Above we showed that the encoder models can adapt to different data distributions and improve their performance if we retrain, even with a small sample set from the new data distribution. Thus, the performance of the encoder model can be significantly improved if one retrains the 3-band Lens Detector with the data derived from the KiDS DR4 data.

6. Conclusions

We have presented a novel machine learning approach known as the self-attention-based encoders to detect strong gravitational lenses. We have explored this new architecture's possibilities to understand better how to apply the transformer models for image analysis using the data from the Bologna Lens challenge. Currently, most of the automated techniques employed to find strong lenses are based on CNNs. However, as noted by Metcalf et al. (2019), CNNs are prone to overfitting the training set. Here we showed that the self-attention-based architectures provide better stability and are less likely to overfit than CNNs. Another advantage of a self-attention-based encoder over a CNN is that it performs better with a fewer number of trainable parameters. Hence, self-attention-based encoder models can be considered a better alternative to CNNs and other automated methods.

Here we have described the 21 encoder models we created to study the application of self-attention-based models for SL detection using the data from the Bologna Lens Challenge. We have presented the three best encoder models, which provide more reliable performance than those participating in the Bologna Lens Challenge. Lens Detector 21 scored an AUROC of 0.9809, which is equivalent to the top AUROC achieved in the challenge. Similarly, Lens Detector 16 scored a TPR_0 0.225 higher than any model that participated in the challenge, and surpassed the top TPR_0 (0.09) achieved by the CNNs by a high margin. We consider Lens Detector 15 to be the best encoder model as it scored 0.14 and 0.48 respectively for TPR_0 and TPR_{10}, outperforming the CNN models to a greater extent and also scoring an AUROC of 0.9783, which is very close to the top AUROC.

From our analysis, we were able to point out that the encoder models have more stability than CNNs, which minimises the need for human interaction or monitoring. Similarly, the encoder models were better than the CNN models in classifying lenses and non-lenses by assigning high probability scores for the lens ($p \approx 1$) and non-lens ($p \approx 0$) systems. In addition, the architecture we proposed here is very simple and robust and has a high resistance to overfitting. We could train models as deep as 25 layers and for 2000 epochs without any sign of overfitting. With a simple 8-layer deep CNN, we were able to surpass the performance of a 46-layer deep RNN and surpass all the other models to a great extent.

We tested our model on the 1983 potential strong lens candidates from the KiDS DR4 data found in Petrillo et al. (2019b). We were able to closely mimic a human visual expert in identifying the strong lenses. Even though we cannot claim to outperform the CNN model presented in Petrillo et al. (2019b), we confirm that the encoder models can perform well on the real data. Since we have tested the network on a different data distribution than it was trained on, we expect to improve the performance of

the encoder model if the training and testing data distribution is similar. In the future we are planning to train the encoder models on more complex data derived from the real data and test them on the real data to find more potential lens candidates. Even though we have glimpsed at the adaptability of the encoder models to different data distributions, further studies are needed to establish the full scope of this architecture.

In the upcoming era of big data in astronomy, automated methods are expected to play a crucial role. Better and alternative automated methods have to be consistently investigated to advance the scientific study in this scenario. From our study it is clear that the search for strong lenses in the current and upcoming wide-field surveys such as KiDS (Kuijken et al. 2019), HSC (Aihara et al. 2019), DES (Abbott et al. 2021), LSST (Ivezić et al. 2019; Verma et al. 2019), *Euclid* (Scaramella et al. 2022), and WFIRST (Koekemoer 2019) can be achieved using self-attention-based encoder models with better performance compared to CNNs.

Acknowledgements. We are thankful to the Referee for their helpful comments, which allowed us to improve the paper significantly. Authors and NCNR are grateful for financial support from MNiSW grant DIR/WK/2018/12 and NCN grants UMO-2017/26/M/ST9/00978 and UMO-2018/30/M/ST9/00757.

References

Abbott, T. M. C., Adamów, M., Aguena, M., et al. 2021, ApJS, 255, 20
Aihara, H., AlSayyad, Y., Ando, M., et al. 2019, PASJ, 71, 114
Blandford, R. D., & Narayan, R. 1992, ARA&A, 30, 311
Bolton, A. S., Burles, S., Koopmans, L. V. E., et al. 2008, ApJ, 682, 964
Bonvin, V., Courbin, F., Suyu, S. H., et al. 2017, MNRAS, 465, 4914
Boylan-Kolchin, M., Springel, V., White, S. D. M., Jenkins, A., & Lemson, G. 2009, MNRAS, 398, 1150
Cabanac, R. A., Alard, C., Dantel-Fort, M., et al. 2007, A&A, 461, 813
Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., & Maureira, J.-C. 2017, ApJ, 836, 97
Cañameras, R., Schuldt, S., Suyu, S. H., et al. 2020, A&A, 644, A163
Cao, S., Biesiada, M., Gavazzi, R., Piórkowska, A., & Zhu, Z.-H. 2015, ApJ, 806, 185
Carion, N., Massa, F., Synnaeve, G., et al. 2020, in Computer Vision – ECCV 2020, eds. A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Cham: Springer International Publishing), 213
Chen, P.-C., Tsai, H., Bhojanapalli, S., et al. 2021, in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (Online and Punta Cana, Dominican Republic: Association for Computational Linguistics), 2974
Chianese, M., Coogan, A., Hofma, P., Otten, S., & Weniger, C. 2020, MNRAS, 496, 381
Collett, T. E., & Auger, M. W. 2014, MNRAS, 443, 969
Covone, G., Paolillo, M., Napolitano, N. R., et al. 2009, ApJ, 691, 531
Davies, A., Serjeant, S., & Bromley, J. M. 2019, MNRAS, 487, 5263
de Jong, J. T. A., Verdoes Kleijn, G. A., Kuijken, K. H., & Valentijn, E. A. 2013, Exp. Astron., 35, 25
Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al. 2021, in 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021
Fu, J., Liu, J., Tian, H., et al. 2019, in IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (Los Alamitos, CA, USA: IEEE Computer Society), 3141
Gentile, F., Tortora, C., Covone, G., et al. 2021, MNRAS, 510, 500
Glorot, X., & Bengio, Y. 2010, in Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS), 9
Hartley, P., Flamary, R., Jackson, N., Tagore, A. S., & Metcalf, R. B. 2017, MNRAS, 471, 3378
Hawkins, D. M. 2004, J. Chem. Information Computer Sci., 44, 1
He, K., Zhang, X., Ren, S., & Sun, J. 2015, in IEEE International Conference on Computer Vision (ICCV), 1026
He, K., Zhang, X., Ren, S., & Sun, J. 2016, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770
He, Z., Er, X., Long, Q., et al. 2020, MNRAS, 497, 556
Hochreiter, S. 1991, Ph.D. thesis Technische Universität München, Germany
Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. 2001, in A Field Guide to Dynamical Recurrent Neural Networks, eds. S. C. Kremer, & J. F. Kolen (USA: IEEE Press)
Huang, X., Storfer, C., Ravi, V., et al. 2020, ApJ, 894, 78
Huang, X., Storfer, C., Gu, A., et al. 2021, ApJ, 909, 27
Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111
Jacobs, C., Glazebrook, K., Collett, T., More, A., & McCarthy, C. 2017, MNRAS, 471, 167
Jacobs, C., Collett, T., Glazebrook, K., et al. 2019, ApJS, 243, 17
Kingma, D. P., & Ba, J. 2015, in 3rd International Conference on Learning Representations, ICLR 2015, (San Diego, CA: USA) Conference Track Proceedings, eds. Y. Bengio & Y. LeCun
Koekemoer, A. M. 2019, AAS Meeting Abs., 234, 222.02
Koopmans, L. V. E., Treu, T., Bolton, A. S., Burles, S., & Moustakas, L. A. 2006, ApJ, 649, 599
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Advances in Neural Information Processing Systems 25, eds. F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (USA: Curran Associates, Inc.), 1097
Kuijken, K., Heymans, C., Dvornik, A., et al. 2019, A&A, 625, A2
Lanusse, F., Ma, Q., Li, N., et al. 2017, MNRAS, 473, 3895
Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998, Proc. IEEE, 86, 2278
Lenzen, F., Schindler, S., & Scherzer, O. 2004, A&A, 416, 391
Li, R., Napolitano, N. R., Tortora, C., et al. 2020, ApJ, 899, 30
Liutkus, A., Cífka, O., Wu, S., et al. 2021, Proc. Mach. Learn. Res., 139, 7067
Magro, D., Zarb Adami, K., DeMarco, A., Riggi, S., & Sciacca, E. 2021, MNRAS, 505, 6155
Mallat, S. 2016, Phil. Transa. R. Soc. A, 374, 20150203
Marshall, P. J., Verma, A., More, A., et al. 2016, MNRAS, 455, 1171
McKean, J., Jackson, N., Vegetti, S., et al. 2015, in Advancing Astrophysics with the Square Kilometre Array (AASKA14), 84
Metcalf, R. B., & Petkova, M. 2014, MNRAS, 445, 1942
Metcalf, R. B., Meneghetti, M., Avestruz, C., et al. 2019, A&A, 625, A119
Niu, Z., Zhong, G., & Yu, H. 2021, Neurocomputing, 452, 48
Parmar, N., Ramachandran, P., Vaswani, A., et al. 2019, in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, (Vancouver, BC: Canada) 68
Pearson, J., Li, N., & Dye, S. 2019, MNRAS, 488, 991
Pérez-Carrasco, M., Cabrera-Vives, G., Martinez-Marin, M., et al. 2019, PASP, 131, 108002
Petrillo, C. E., Tortora, C., Chatterjee, S., et al. 2017, MNRAS, 472, 1129
Petrillo, C. E., Tortora, C., Chatterjee, S., et al. 2019a, MNRAS, 482, 807
Petrillo, C. E., Tortora, C., Vernardos, G., et al. 2019b, MNRAS, 484, 3879
Rojas, K., Savary, E., Clément, B., et al. 2021, A&A, submitted, [arXiv:2109.00014]
Russakovsky, O., Deng, J., Su, H., et al. 2015, Int. J. Comput. Vis., 115, 211
Scaramella, R., Amiaux, J., Mellier, Y., et al. 2022, A&A, 662, A112
Schaefer, C., Geiger, M., Kuntzer, T., & Kneib, J.-P. 2018, A&A, 611, A2
Simonyan, K., & Zisserman, A. 2015, in 3rd International Conference on Learning Representations, ICLR 2015 (San Diego, CA: USA) Conference Track Proceedings, eds. Y. Bengio, & Y. LeCun
Srivastava, R. K., Greff, K., & Schmidhuber, J. 2015, CoRR, abs/1505.00387 [arXiv:1505.00387]
Su, J., Lu, Y., Pan, S., Wen, B., & Liu, Y. 2021, CoRR, abs/2104.09864 [arXiv:2104.09864]
Tan, A., Nguyen, D. T., Dax, M., Nießner, M., & Brox, T. 2021, in Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, 9799
Treu, T. 2010, ARA&A, 48, 87
Vaswani, A., Shazeer, N., Parmar, N., et al. 2017, in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, (Long Beach, CA: USA) 5998
Verma, A., Collett, T., Smith, G. P., Strong Lensing Science Collaboration, & the DESC Strong Lensing Science Working Group. 2019, ArXiv e-prints [arXiv:1902.05141]
Yang, X. 2020, J. Phys. Conf. Ser., 1693, 012173
Zhang, H., Goodfellow, I. J., Metaxas, D. N., & Odena, A. 2018, CoRR, abs/1805.08318 [arXiv:1805.08318]
Zhao, H., Jia, J., & Koltun, V. 2020, CoRR, abs/2004.13621 [arXiv:2004.13621]

Appendix A: ROC of encoder models

ROC curves of all the encoder models presented in Table 1 are displayed in Fig. A.1. The inset gives the AUROC of each model in order to facilitate comparison of the models.

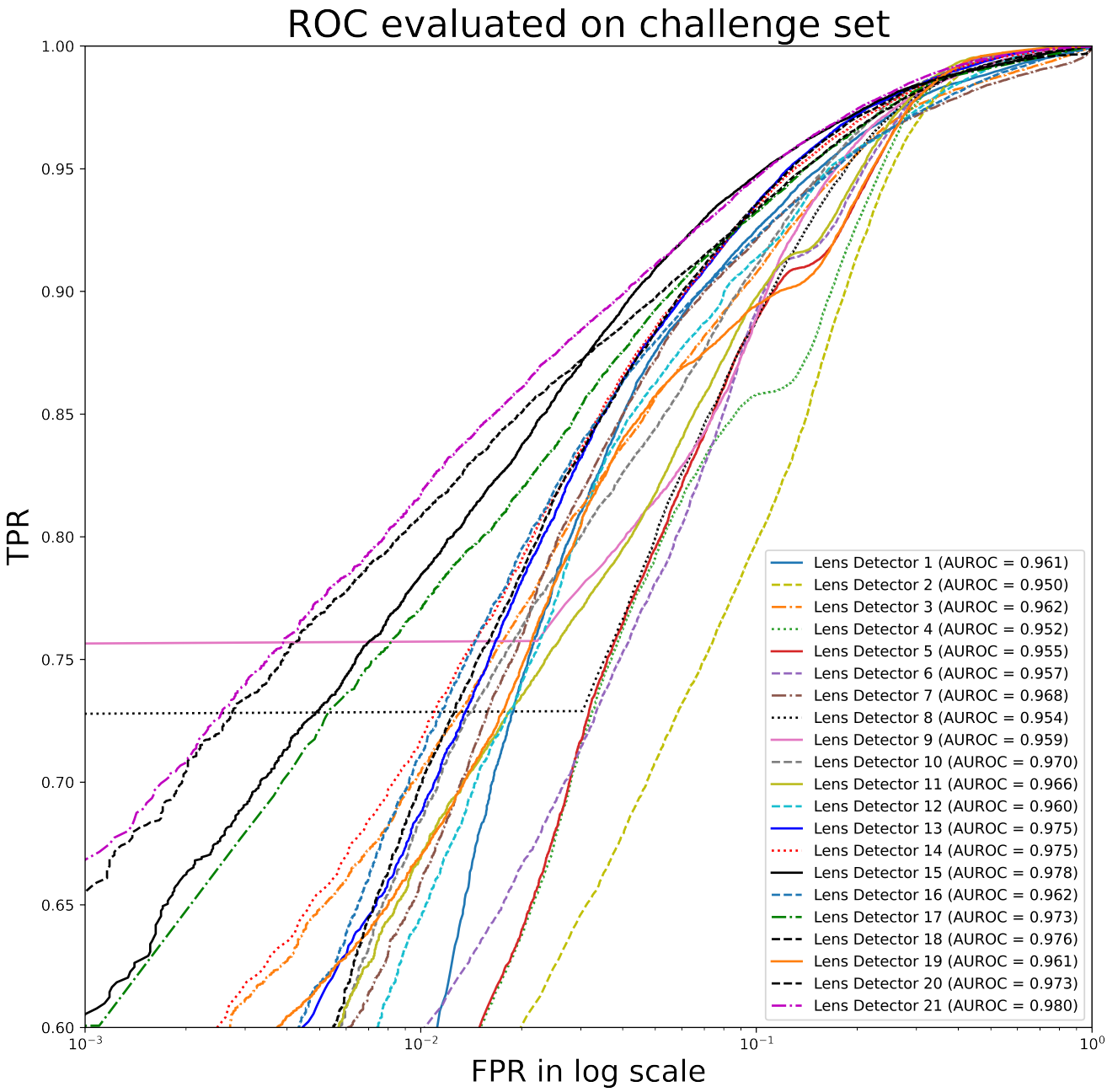

Fig. A.1. ROC curves of all the encoder models.