Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **512,** 3464–3479 (2022)
Advance Access publication 2022 February 8

https://doi.org/10.1093/mnras/stac562

Detecting gravitational lenses using machine learning: exploring interpretability and sensitivity to rare lensing configurations

Joshua Wilde,[1★] Stephen Serjeant,[1] Jane M. Bromley,[2] Hugh Dickinson,[1] Léon V. E. Koopmans[3] and R. Benton Metcalf[4,5]

[1] *School of Physical Sciences, The Open University, Walton Hall, Milton Keynes MK7 6AA, UK*
[2] *School of Computing & Communications, The Open University, Walton Hall, Milton Keynes MK7 6AA, UK*
[3] *Kapteyn Astronomical Institute, University of Groningen, PO Box 800, NL-9700AV Groningen, the Netherlands*
[4] *Dipartimento di Fisica e Astronomia, Universitá di Bologna, via Gobetti 93/2, I-40129 Bologna, Italy*
[5] *INAF - Osservatorio di Astrofisica e Scienza dello Spazio di Bologna, via Gobetti 93/3, I-40129 Bologna, Italy*

Accepted 2022 February 24. Received 2022 February 4; in original form 2021 December 10

ABSTRACT

Forthcoming large imaging surveys such as *Euclid* and the Vera Rubin Observatory Legacy Survey of Space and Time are expected to find more than 10^5 strong gravitational lens systems, including many rare and exotic populations such as compound lenses, but these 10^5 systems will be interspersed among much larger catalogues of $\sim 10^9$ galaxies. This volume of data is too much for visual inspection by volunteers alone to be feasible and gravitational lenses will only appear in a small fraction of these data which could cause a large amount of false positives. Machine learning is the obvious alternative but the algorithms' internal workings are not obviously interpretable, so their selection functions are opaque and it is not clear whether they would select against important rare populations. We design, build, and train several convolutional neural networks (CNNs) to identify strong gravitational lenses using *VIS*, *Y*, *J*, and *H* bands of simulated data, with F1 scores between 0.83 and 0.91 on 100 000 test set images. We demonstrate for the first time that such CNNs do not select against compound lenses, obtaining recall scores as high as 76 per cent for compound arcs and 52 per cent for double rings. We verify this performance using *Hubble Space Telescope* and Hyper Suprime-Cam data of all known compound lens systems. Finally, we explore for the first time the interpretability of these CNNs using Deep Dream, Guided Grad-CAM, and by exploring the kernels of the convolutional layers, to illuminate why CNNs succeed in compound lens selection.

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

1 INTRODUCTION

In the near future the *Euclid Space Telescope* (Laureijs et al. 2011) and the Vera Rubin Observatory Legacy Survey of Space and Time (LSST; Abell et al. 2009) will achieve first light. Both *Euclid* and Rubin will be mapping about half the sky in optical/near-infrared bands to AB depths of ∼24–24.5 (*Euclid*) and ∼22–25 (LSST). *Euclid* is expected to find more than 10^5 strong gravitational lens systems (e.g. Collett 2015) and 5000 strongly lensed quasars while LSST expects to find 10^5 galaxy–galaxy lenses, 10^4 strongly lensed quasars and 500 lensed Type Ia supernovae with 100 of these lenses being suitable for gathering time-delay data (Zhan & Tyson 2018; Verma et al. 2019).[1]

With this large number of gravitational lenses, several cosmologically important quantities can be constrained. These include the dark energy equation of state (w) parameter, the Hubble constant (H_0), and dark matter halo substructure. The current tension in H_0 could be eased (or at least illuminated) by measuring the time delays caused by gravitational lensing. Currently H_0 can only be constrained through lensing to a precision of 2.4 per cent using this method (including the error budget for systematics and assuming a spatially flat cosmology) due to the limited number of suitable gravitational lenses that are currently known (Wong et al. 2020). The problem with using time delays from gravitational lenses is that the assumption of the mass distribution using methods such as the mass-sheet transformation can make all observable strong-lensing parameters invariant apart from the time delay (Schneider & Sluse 2014). The time delay of a lens is roughly $\Delta t \propto H_0^{-1}(1 - k_e)$, where k_e is the mean convergence. The only gravitational lens observable which directly constrains this value is the time delay. Properties such as stellar kinematics can provide additional constraints on k_e. Overconstrained models can generate a value for k_e that are precise but not accurate resulting in uncertainties that are far larger than reported (Kochanek 2020). It is possible to break this degeneracy without assuming a mass profile, instead adding spatially resolved kinematics from time delay and non-time delay lens systems resulting in a precision of 2.5 per cent (Birrer & Treu 2020) The amount of data from *Euclid* and LSST will increase the number of suitable gravitational lenses for measuring time delays. This is expected to allow for a determination of H_0 with an accuracy of less than 1 per cent (Liao 2019).

Such a large sample of strong gravitational lenses should also be an excellent source of rare and exotic lens configurations. Compound lenses (multiple lens plane systems) are both rare and extremely

★ E-mail: joshua.wilde@open.ac.uk

[1] The code used in this paper is publicly available at https://github.com/JoshWilde/LensFindery-McLensFinderFace.

valuable to cosmology. Over the course of its mission LSST expects to find ~90 compound lens systems (Mandelbaum et al. 2018). These systems enforce tighter constraints on models allowing for lower uncertainties in measurements of parameters such as $\Omega_{\rm M}$ and w (Collett et al. 2012; Collett & Auger 2014). From as few as 50 compound lenses these values can be measured to within a 10 per cent accuracy (Gavazzi et al. 2008). Other parameters such as w and $\Omega_{\rm m}$ can be determined with a high accuracy from the time delays of 100 lensed quasars (Treu et al. 2013).

However, *Euclid*'s entire non-lensed catalogue will comprise about a billion galaxies, and LSST's will be even larger. Finding strong gravitational lenses, and exotic compound systems in particular, will therefore be a 'needle in a haystack in a field of haystacks' data mining problem. In this paper, we define compound systems to be a gravitational lens containing two sets of strongly lensed images. The imaging data are too numerous to rely on human inspection alone (e.g. Marshall et al. 2016; More et al. 2016; Sonnenfeld et al. 2020), but machine learning is a natural approach to attempt to solve this problem.

Machine learning was originally inspired by the processes of neurons within the brain and was initially developed as artificial neural networks (McCulloch & Pitts 1943). These could only perform simple tasks such as playing checkers (Samuel 1960). These artificial neural networks were further improved upon with algorithms such as back-propagation (Rumelhart, Hinton & Williams 1986) allowing for information to be transferred between hidden layers. Taking inspiration from how the human eye processes information, convolutional neural networks (CNNs) were created (LeCun et al. 1989, 1998). These included convolutional layers at the start of the network that forced the CNNs to learn information about how features were related within the image. These layers tend to start off by learning simplistic features such as edges, then they progress into patterns and shapes, eventually detecting complex features within the data.

Over the past decade, machine learning and in particular CNNs have become more prominent following from advances in GPUs, the creation of large accessible data sets (Deng et al. 2009), and the success of the deep learning CNN AlexNet (Krizhevsky, Sutskever & Hinton 2012). As it has become popular in the mainstream, machine learning has been increasingly used in astronomy, in particular to classification problems, such as using CNNs and multiband images to classify dwarf galaxies (Müller & Schnider 2021); using CNNs to assign Fanaroff–Riley classifications to radio galaxies (Scaife & Porter 2021); using an auto-encoder for morphological classification of galaxies (Spindler, Geach & Smith 2021); using a U-Net to perform source detection, segmentation, and classification (Hausen & Robertson 2020); using Recurrent Neural Networks (RNNs) to correct classifications in maps (Maggiori et al. 2017); and classifying galaxies using t-distributed stochastic neighbour embedding (T-SNE; Zhang et al. 2020). Denoising auto-encoders have also been used for image deconvolution (Lauritsen et al. 2021).

In recent years machine learning has been applied to search for gravitational lenses. Various machine learning techniques have been used for this application, which can be broken down into two main categories: supervised and unsupervised learning. Supervised machine learning approaches require large amounts of data in order to sensibly train. Since the number of lenses discovered is much smaller than the number of lenses need to train machine learning approaches, simulated data are often used for this task (e.g. Hezaveh, Levasseur & Marshall 2017; Jacobs et al. 2017; Huang et al. 2019; Jacobs et al. 2019; Khramtsov et al. 2019; Petrillo et al. 2019; He et al. 2020). Other methods can include using data augmentation to boost the size of a data set. In preparation for upcoming surveys and telescopes a great deal of work has been done to create automated lens finding tools to make the process of finding gravitational lenses in this large amount of data more effective. Within the *Euclid* telescope consortium, there have been strong gravitational lens challenges, in which participants used various methods to classify simulated images as lenses and non-lenses (Metcalf et al. 2019). Machine learning technologies deployed on lens finding include Support Vector Machines (SVMs; Hartley et al. 2017), auto-encoders (Cheng et al. 2020), transfer learning (Hezaveh et al. 2017), ResNets (Lanusse et al. 2017; Huang et al. 2019; Petrillo et al. 2019), and CNNs (Pearson, Pennock & Robinson 2018). The most common architecture used is CNN. These range from the very simplistic to the very complex.

Nevertheless, this technology still presents the community with a fundamental interpretability problem. How exactly are these lens-finding algorithms working? What features of the lenses are they learning and responding to? What categories of gravitational lens will they be good at finding, and are there strong lens categories or configurations that would be systematically missed by these machine learning algorithms? A naive and traditional approach would be to regard machine learning, and CNNs in particular, as black boxes whose performance can just be assessed with simulated inputs. Such an approach would only be as good as the extent to which simulations reproduce real data, and 'it just works' is not usually considered a sufficient level of insight in the physical sciences. Moreover, awareness of inadvertent algorithmic biases is increasingly seen as a core problem in machine learning (e.g. Obermeyer et al. 2019), and tools are now being developed to shed light on the internal workings of these ostensible black box algorithms.

One does not need to dig very deep to discover that interpretability is a non-trivial problem. Deep learning models often rely on many millions of trainable parameters that depend on each other in very complicated ways. Historically, this complexity has made interpreting the operation of deep-learning models very challenging. Moreover, CNNs can be sensitive to very subtle features of their input data, so images which look like noise to humans are able to generate high-scoring results in CNNs (Nguyen, Yosinski & Clune 2015). Similarly, images of objects such as cats can be classified with high probability of being elephants by imposing an elephant skin texture over the image (Geirhos et al. 2018). One of the major areas of research in machine learning at the current time is the development of techniques to interpret the results of machine learning algorithms. There are several ways in which machine learning is analysed to develop an understanding of how the algorithm is behaving, such as tactically changing the input values in a known way and evaluating how the output changes (Zeiler & Fergus 2014). Other methods include using back-propagation to update the input image based on a target output value rather than the model weights. This updates an image to strongly activate the target class, which can reveal information that the model has learnt such as making forearms when trying to create an image of a dumbbell (Mordvintsev, Olah & Tyka 2015).

There are various methods that attempt to highlight areas of the image that influence the decision making of the CNN. The simplest of these approaches is the generation of saliency maps (Simonyan, Vedaldi & Zisserman 2013), including occlusion maps (see Section 5.1 and Zeiler & Fergus 2014) and *class activation maps* (see Section 5.6 and Selvaraju et al. 2017) like GradCAM and guided GradCAM. Alternative approaches to interpreting deep networks include deriving *class-generated images* (CGIs) using tools like Google's Deep Dream (Mordvintsev et al. 2015).

Understanding the decision making processes of machine learning algorithms is arguably the next important advance in the application of machine learning to astronomy. If one can understand the decisions the algorithms are making then one might be able to directly infer which image features or astrophysical phenomena the algorithms may be selecting for or actively selecting against. This will help quantify which categories of gravitational lenses that the algorithms can find and those they cannot. For example, one of the possible categories of gravitational lens that could be missed by machine learning are source galaxies that are not blue, particularly if trained on existing surveys such as SLACS (Bolton et al. 2008) which pre-selects early-type foreground galaxies and star-forming emission line background systems. The machine learning methods could, for example, be focusing on colour more than the general morphology of the lens system. CNNs have been shown to perform worse on galaxy–galaxy lenses that differ from the colour and point spread function (PSF) of lenses from their training data (Li et al. 2021; Jacobs et al. 2022). If one does not investigate which rare lens configurations the machine learning methods do not detect, there is a risk of losing interesting and important gravitational lens systems in the seas of data generated by future telescopes, as well as drawing spurious conclusions about the populations of sources and lenses, and drawing misleading cosmological inferences from them. Indeed there is already a precedent for human volunteers finding a strong lens with a red background source that would be missed by a SLACS-like automated lens finder (Geach et al. 2015).

In this paper, we present new CNNs trained for finding strong gravitational lenses, and assess the networks' characteristics and performance with a wide variety of these interpretability tools. As a critical test case, we examine in particular whether the exotic subclass of compound lenses is detected by our CNNs, and attempt to determine why. In Section 2, we describe the three different data sets we use within this paper. In Section 3, we describe the network architecture and training process for the CNNs used within this work. Section 4 describes the results of the CNNs on the three different data sets. Section 5 investigates the interpretability of the CNNs and what they interpret as a lens and non-lens. Section 6 gives a discussion on this paper and Section 7 is the conclusion. A flat Lambda cold dark matter cosmology is used throughout with $H_0 = 70\ \mathrm{km\,s^{-1}\,Mpc^{-1}}$ and matter density $\Omega_\mathrm{m} = 0.3$.

2 DATA

2.1 Single lenses

There is no empirical data set with the image characteristics and size of *Euclid*, so training on simulations is inevitable. The data used in this paper are simulated *Euclid* images from the VIS and NISP instruments, produced by the Bologna Lens Factory (Metcalf, private communication). There are 100 000 images in the training set and an additional 100 000 images in the test set. The training set consists of ∼72 000 lenses and ∼28 000 non-lenses. The test set consists of ∼71 000 lenses and ∼29 000 non-lenses. The NISP data consists of three bands: *Y*, *J*, and *H*, while the VIS imaging is with a very broad-band visual filter (Laureijs et al. 2011). These images are 66 × 66 pixels with a pixel scale of 0.3 arcsec/pixel. The VIS images are 200 × 200 pixels with a pixel scale of 0.1 arcsec/pixel. An example of VIS images from this data set is shown in Fig. 1.

The gravitational lens can be anywhere within the image. The PSF has an approximately three-fold symmetry (Schmitz et al. 2020). The images are scaled linearly between 0 and 1 before being input to the CNNs.

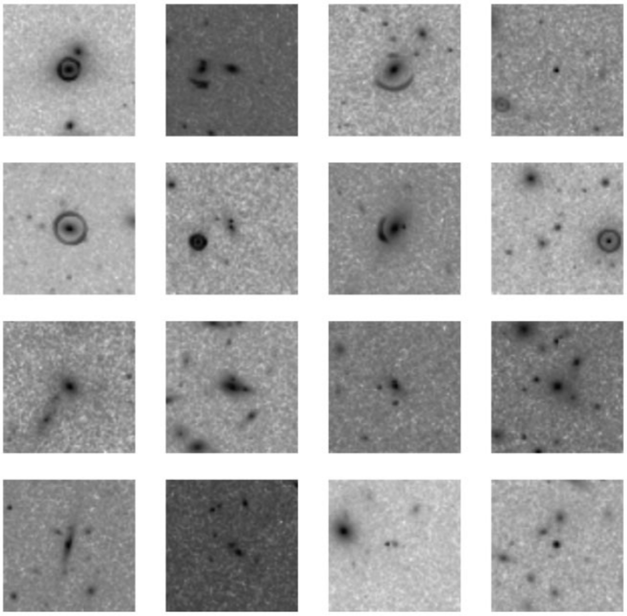

Figure 1. Log-scale *VIS*-band images from the training set. Top two rows show simulated single lenses. Bottom two rows show simulated non-lenses.

2.2 Compound lenses

The simulated systems in Section 2.1 did not contain compound lenses, and very few compound lensing systems have been discovered to date, so further simulations were required. Simulated images containing compound lenses were created using the LENSTRONOMY Python package (Birrer & Amara 2018) and SKYPY (Amara et al. 2021). Redshifts and magnitudes for the lens and source galaxies were randomly sampled in pairs from set of parameter values that were used to generate the single-lens training data. The source was sampled from source magnitude and source redshift distributions and the two-lens galaxies were sampled from lens magnitude and lens redshift distributions. From these sampled magnitudes the galaxy velocity dispersions were calculated using the Faber–Jackson relationship (Faber & Jackson 1976) and finally the Einstein radius of each lens were calculated using equation (1):

$$\theta = 4\pi \left(\frac{\sigma_\mathrm{v}}{c}\right)^2 \frac{D_\mathrm{ls}}{D_\mathrm{s}}, \tag{1}$$

where θ is the Einstein radius, D_ls is the lens–source angular diameter distance, D_s is the observer–source angular diameter distance, and σ_v is the lensing galaxy velocity dispersion. The other parameters of the galaxies were sampled from a distribution of SLACS lens data (Bolton et al. 2008; Auger et al. 2009; Newton et al. 2011; Shu et al. 2017; Denzel, Mukherjee & Saha 2021). A uniform distribution was used to assign these SLACS variables to the galaxies; these parameters are shown in Table 1. The lower redshift lens is more massive than the higher redshift lens two-thirds of the time. The *NISP* bands have a simulated exposure time of 264 s and the *VIS* band have a simulated exposure time of 1800 s to match the single-lens data for *Euclid*-Wide. To approximate the *Euclid* instruments' imaging characteristics, the simulated images were convolved with Gaussian PSFs with full width at half-maximum

Table 1. Parameters used to generate simulated compound lenses.

Parameter	Lower bound	Upper bound	Units
R sersic	0.2	7	arcsec
n sersic	1.5	20	–
X position of source	−5	5	arcsec
Y position of source	−5	5	arcsec
Angle	0	π	rad
Axis ratio	0.3	1	–
X position of compound arc lens galaxy	X position of source − 5	X position of source + 5	arcsec
Y position of compound arc lens galaxy	Y position of source − 5	Y position of source + 5	arcsec
X position of double ring lens galaxy	X position of source − 0.25	X position of source + 0.25	arcsec
Y position of double ring lens galaxy	Y position of source − 0.25	Y position of source + 0.25	arcsec
Gamma	−0.5	0.5	–

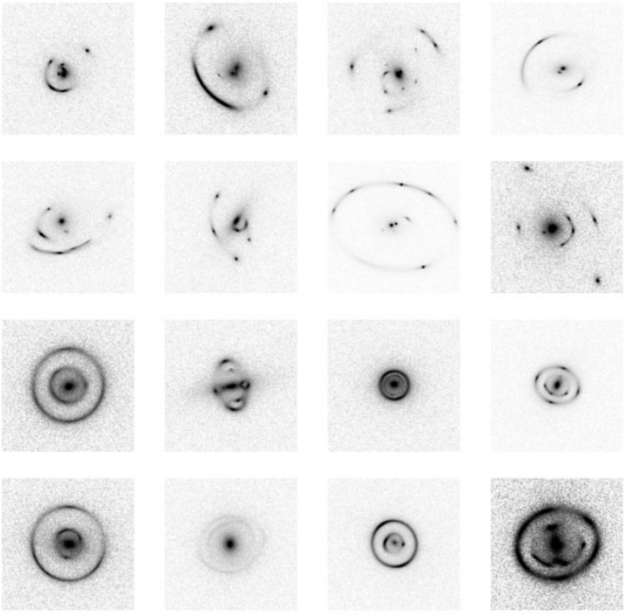

Figure 2. *VIS*-band images of compound lenses. Top two rows show simulated compound arcs. Bottom two rows show simulated double rings.

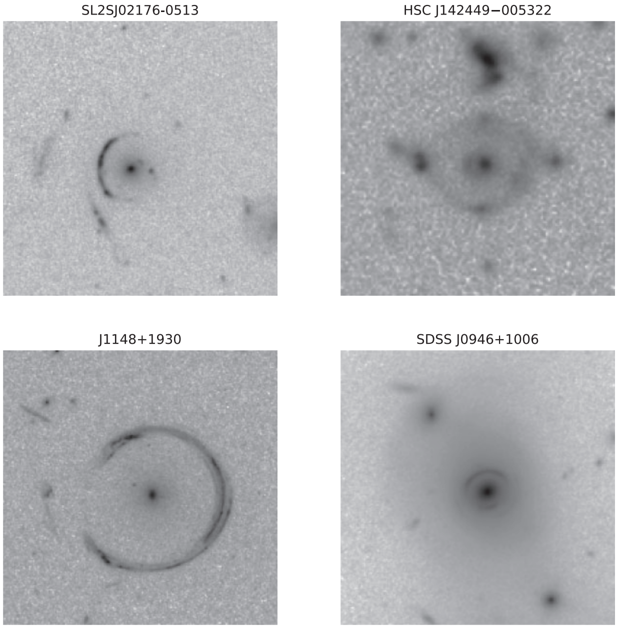

Figure 3. *F814W*-band images of the four known compound lenses.

appropriate for the VIS and NISP instruments (Laureijs et al. 2011).

Two data sets of 10 000 images were created. One used parameters that produce a set of compound lenses that are mainly arcs, while the other used a parameter set with tighter constraints on the galaxy positions generating mostly double Einstein rings (see Table 1). An example of these simulated compound lenses can be seen in Fig. 2.

2.3 Known compound lenses

The CNNs described in this paper were applied to four known compound lenses SL2SJ02176−0513 (Tu et al. 2009), J1148+1930 (Schuldt et al. 2019), and SDSS J0946+1006 (Gavazzi et al. 2008) from the *Hubble Space Telescope (HST)* and HSC J142449−5322 (Tanaka et al. 2016) from Hyper Suprime-Cam (HSC). The archival *F814W*-band images for these compound lenses are shown in Fig. 3. A 20 × 20 arcsec postage stamp was extracted with the compound lens centred in the image. The pixel scales of these compound lens postage stamps were resampled to the same pixel scale as the training data, 0.1 arcsec/pixel for the bluest band, and 0.3 arcsec/pixel for the rest. These data differ from the data on which the CNNs have been trained. These known compound lenses do not necessarily have four bands of imaging available. The CNN architectures require four bands of input data, so missing bands were populated by duplicating data from adjacent bands. The bands used for each compound lens are shown in Table 2. Each image was normalized between 0 and 1 prior to being input into the models. Each band was resampled to the pixel dimensions of the CNN as described in Section 3.

Table 2. Known compound lenses image bands to expected *CNN* bands.

Lens name	*VIS*	*Y*	*J*	*H*
SL20S J02176−0513	F606	F606	F814	F814
HSC J142449−005322	*R*	*I*	*Z*	*Y*
SDSS J1148+1930	F475	F606	F606	F814
SDSS J0946+1006	F160	F336	F438	F814

3 NETWORK DESIGN

This section details the creation, training, and testing of three CNN models; OU-66, OU-200, and OU-T-SNE (OU stands for Open

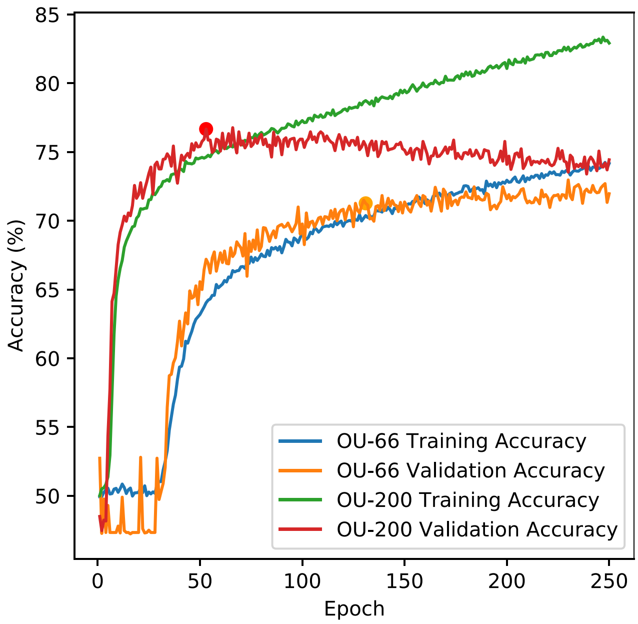

Figure 4. The accuracy of the training and validation data over the course of training. The final recorded weights are shown with a dot.

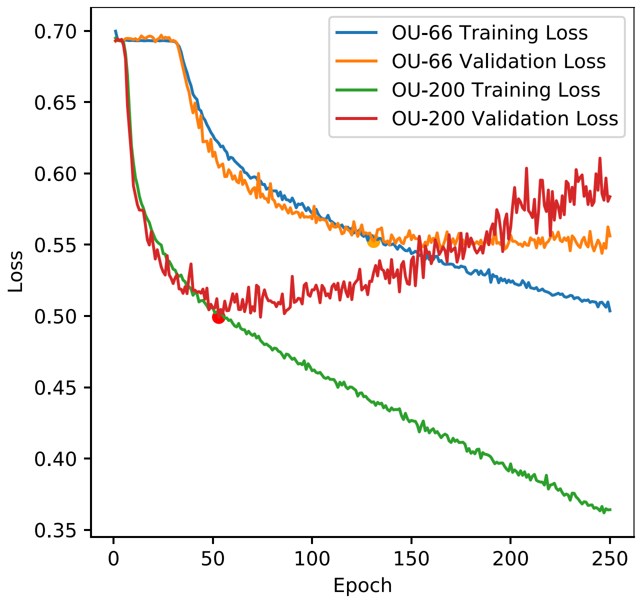

Figure 5. The loss of the training and validation data over the course of training. The final recorded weights are shown with a dot.

University in this case). The architectures for both of these CNNs can be seen in Tables 3 and 4 and graphical representations in Figs 7 and 8.

3.1 OU-66, OU-200, and variants

These CNNs were created and trained using the Python package PYTORCH (Paszke et al. 2019). Using these model designs a total of seven CNNs were created for this task, each accepting different configurations of input data. These CNNs can be segregated into two architectures depending on the pixel dimensions of the input data, if the data was 66 × 66 pixels the OU-66 architecture was

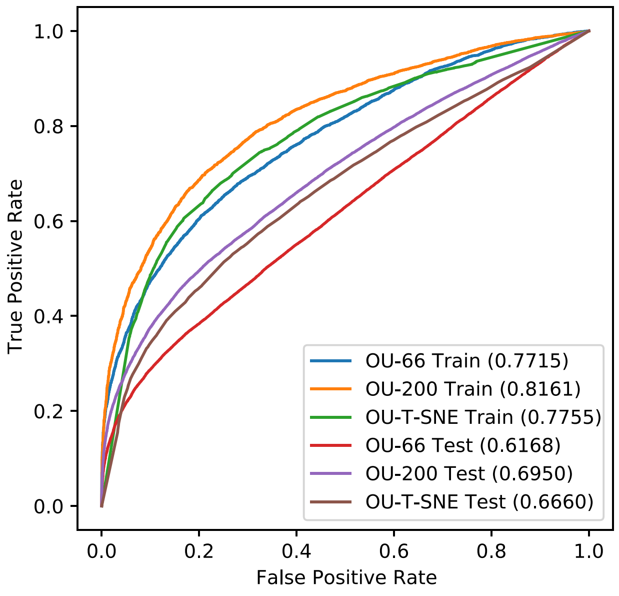

Figure 6. Receiver Operating Characteristic (ROC) curve for each model on the training and testing data.

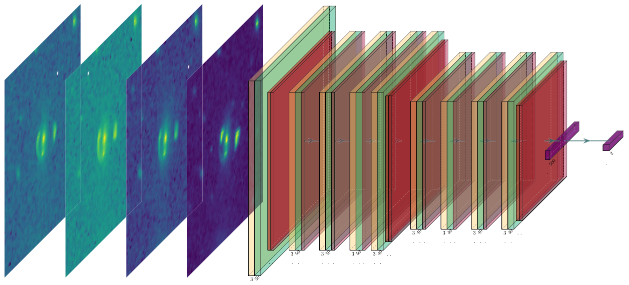

Figure 7. The architecture of OU-66. Yellow squares correspond to Conv2D layers, teal squares correspond to ReLU layers, orange layers correspond to MaxPool2D layers, and red squares correspond to Dropout layers.

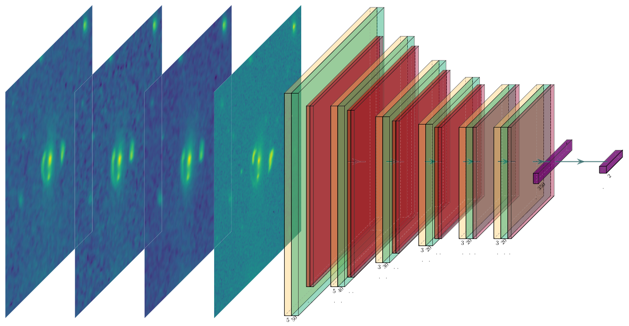

Figure 8. The architecture of OU-200. Yellow squares correspond to Conv2D layers, teal squares correspond to ReLU layers, orange layers correspond to MaxPool2D layers, and red squares correspond to Dropout layers.

Table 3. OU-66 architecture. The ReLU activation functions are considered layers in this table and follow all Conv2D layers.

Layer	Type	Kernel/pool size	Kernel/neuron count	Stride	Padding	Activation/ dropout probability	Input shape	Output shape	Parameter count
0	Conv2D	(3,3)	50	(1,1)	Valid	ReLU	(4,66,66)	(50,64,64)	1850
2	MaxPool2D	(2,2)	–	(1,1)	–	–	(50,64,64)	(50,32,32)	0
3	Dropout	–	–	–	–	0.2	(50,32,32)	(50,32,32)	0
4	Conv2D	(3,3)	50	(1,1)	Valid	ReLU	(50,32,32)	(50,30,30)	22 550
6	Dropout	–	–	–	–	0.2	(50,30,30)	(50,30,30)	0
7	Conv2D	(3,3)	50	(1,1)	Valid	ReLU	(50,30,30)	(50,28,28)	22 550
9	Dropout	–	–	–	–	0.2	(50,28,28)	(50,28,28)	0
10	Conv2D	(3,3)	50	(1,1)	Valid	ReLU	(50,28,28)	(50,26,26)	22 550
12	Dropout	–	–	–	–	0.2	(50,26,26)	(50,26,26)	0
13	Conv2D	(3,3)	40	(1,1)	Valid	ReLU	(50,26,26)	(40,24,24)	18 040
15	MaxPool2D	(2,2)	-	(1,1)	–	–	(40,24,24)	(40,12,12)	0
16	Dropout	–	–	–	–	0.2	(40,12,12)	(40,12,12)	0
17	Conv2D	(3,3)	40	(1,1)	Valid	ReLU	(40,12,12)	(40,10,10)	14 440
19	Dropout	–	–	–	–	0.2	(40,10,10)	(40,10,10)	0
20	Conv2D	(3,3)	40	(1,1)	Valid	ReLU	(40,10,10)	(40,8,8)	14 440
22	Dropout	–	–	–	–	0.2	(40,8,8)	(40,8,8)	0
23	Conv2D	(3,3)	40	(1,1)	Valid	ReLU	(40,8,8)	(40,6,6)	14 440
25	Dropout	–	–	–	–	0.2	(40,6,6)	(40,6,6)	0
26	Conv2D	(3,3)	30	(1,1)	Valid	ReLU	(40,6,6)	(30,4,4)	10 830
28	MaxPool2D	(2,2)	–	(1,1)	–	–	(30,4,4)	(30,2,2)	0
29	Dropout	–	–	–	–	0.2	(30,2,2)	(30,2,2)	0
30	Dense	–	500	–	–	ReLU	120	500	60 500
32	Dense	–	2	–	–	Softmax	500	2	1002
Total parameter count									203 192

Table 4. OU-200 architecture. The ReLU activation functions are considered layers in this table and follow all Conv2D layers.

Layer	Type	Kernel/pool size	Kernel/neuron count	Stride	Padding	Activation/ dropout probability	Input shape	Output shape	Parameter count
0	Conv2D	(5,5)	50	(1,1)	Valid	ReLU	(4,200,200)	(50,196,196)	5050
2	MaxPool2D	(2,2)	-	(1,1)	–	–	(50,196,196	(50,98,98)	0
3	Dropout	–	–	–	–	0.2	(50,98,98)	(50,98,98)	0
4	Conv2D	(5,5)	50	(1,1)	Valid	ReLU	(50,98,98)	(40,94,94)	50 040
6	MaxPool2D	(2,2)	–	(1,1)	–	–	(40,94,94)	(40,47,47)	0
7	Dropout	–	–	–	–	0.2	(40,47,47)	(40,47,47)	0
8	Conv2D	(3,3)	40	(1,1)	Valid	ReLU	(40,47,47	(30,45,45)	10 830
10	MaxPool2D	(2,2)	–	(1,1)	–	–	(30,45,45)	(30,22,22)	0
11	Dropout	–	–	–	–	0.2	(30,22,22)	(30,22,22)	0
12	Conv2D	(3,3)	30	(1,1)	Valid	ReLU	(30,22,22)	(20,20,20)	5420
14	MaxPool2D	(2,2)	–	(1,1)	–	–	(20,20,20)	(20,10,10)	0
15	Dropout	–	–	–	–	0.2	(20,10,10)	(20,10,10)	0
16	Conv2D	(3,3)	20	(1,1)	Valid	ReLU	(20,10,10)	(20,8,8)	3620
18	Dropout	–	–	–	–	0.2	(20,8,8)	(20,8,8)	0
19	Conv2D	(3,3)	20	(1,1)	Valid	ReLU	(20,8,8)	(20,6,6)	3620
21	Dropout	–	–	–	–	0.2	(20,6,6)	(20,6,6)	0
22	Dense	-	350	–	–	ReLU	720	350	252 350
24	Dense	–	2	–	–	Softmax	350	2	702
Total parameter count									331 632

used, if the data was 200 × 200 pixels the OU-200 architecture was used. These architectures are outlined in Tables 3 and 4, and are based upon a CNN designed and trained on a similar problem by Davies, Serjeant & Bromley (2019). Changes were made to this model including adding regularization with dropout (Srivastava et al. 2014), adding another fully connected layer, replacing 15 × 15 convolutional kernels with two 5 × 5 convolutional kernels, and replacing 5 × 5 convolutional kernels with two 3 × 3 convolutional kernels. The replacement of single large convolutional kernels with two smaller convolutional kernels is designed to reduce the computational time required, but the transformations being made are equivalent. The ReLU activation function (Agarap 2018) is used after each convolutional layer this returns zero if the input is negative otherwise returns the input value. The way that inputs were handled was also changed by scaling them to be between 0 and 1, rather than normalizing them to the maximum pixel value in each image.

Seven CNNs OU-J, OU-Y, OU-H, OU-JYH, OU-VIS, OU-66, and OU-200 are used to create the classifer OU-T-SNE in Section 3.3.

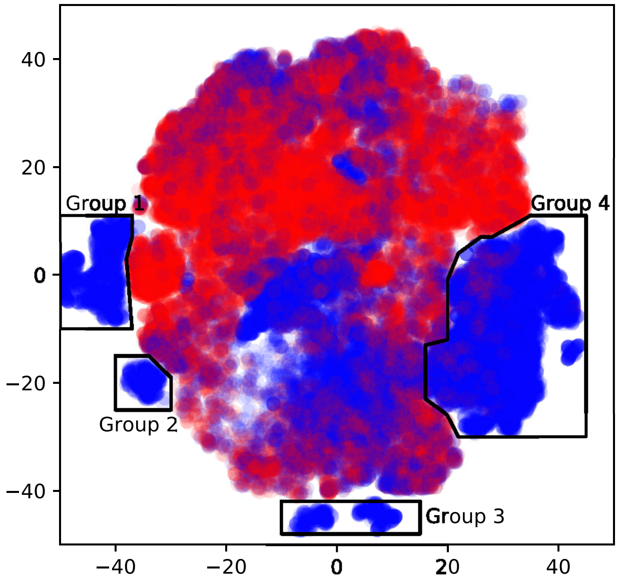

Figure 9. T-SNE embedding of the seven different model outputs for the training set. This is a dimensionality reduction of the 7D array into 2D. The position of the image in this 2D reduction is used to classify the image as a lens or non-lens. Blue dots represent lenses and red represent non-lenses.

These CNNs take input from the bands in their names; OU-66 and OU-200 take all four bands and resize the input to the corresponding size in their name, while the variations take only subsets of the bands. CNN variations that use near-infrared data only are based on OU-66, while OU-VIS is based on OU-200. The outputs of these models are used to generate OU-T-SNE.

3.2 CNN training

Each CNN is initialized with He Normalization (He et al. 2015). Images from each band were combined into a single input tensor with dimensions shown in Tables 3 and 4. The CNNs were trained for 250 epochs with a learning rate[2] of 3×10^{-4} using the Adam optimizer (Kingma & Ba 2014) and a batch size of 250 for OU-66, and 125 for OU-200. They were each trained on the Open University GPU cluster[3] for approximately 7 h.

The *Euclid* strong-lensing data challenge used two distinct data sets. The first of these sets (COMP) contains 60 000 images and was provided to participants to allow them to design, train, refine and validate their challenge entries. The second data set (EVAL) contains 100 000 images that were reserved for final evaluation of the individual challenge entries. While designing the CNNs in this paper, the COMP data set was further subdivided into three subsets – the ‘training set’, the ‘validation set’, and the ‘COMP test set’. The training set contains 45 000 images, the validation set contains 3000 images and the COMP test set contains 12 000 images. If the CNN took input from both the *VIS* and the *NISP* bands, the bands were resampled to the input size of the CNN using the resize function from the Python package SCIKIT-IMAGE (van der Walt et al. 2014). Attempting to balance the non-lens class by rotating images caused the CNN to achieve spurious accuracies greater than 90 per cent by learning the orientation of the PSF, despite the asymmetry of the PSF being not discernable by eye in the images. To avoid this problem the class imbalance was mitigated by applying a weighting to the loss function. Categorical cross-entropy loss is shown in equation (2) where M is the number of classes, y is 1 if the class label (c) is the correct classification for the image (i), p is the predicted value for the image (i) in class (c). A categorical cross-entropy loss was used instead of a binary classifier to allow the CNN to generate schemata for both lenses and non-lens. In contrast, training a binary classifier will only yield an internal representation or schema for its target class i.e. *either* for lens *or* non-lens. The training data was shuffled during training.

$$\text{loss} = -\sum_{c=1}^{M} y_{\text{i},c} \ln(p_{\text{i},c}). \tag{2}$$

During training the time taken for each epoch, the epoch number, the training loss, the training accuracy, the validation loss, and the validation accuracy were recorded. The parameters show the learning process of the CNN. Eventually the CNNs are expected to have memorized the entire training set. This leads to the training accuracy increasing over time until it memorizes the entire data set. At this stage in CNN can often no longer generalize to unseen data. The validation data set is not seen whilst the CNN is training this acts as a proxy for the CNN’s ability to generalize to unseen data. The training and validation loss represent the same concepts in terms of categorical cross-entropy loss instead of accuracy. The CNN weights for each trained model are for the epoch which has the lowest loss. The evolution of the training and validation accuracies during training are shown in Fig. 4, and the corresponding loss evolution is shown in Fig. 5. The model weights were saved whenever the model reached a new minimum validation loss. The ROC curves for these models are shown in Fig. 6.

3.3 OU-T-SNE

The OU-T-SNE model is a collection of seven different models. These models have the same architectures as OU-66 and OU-200 but with different inputs. There are five variations of OU-66, including three where the input is a single *NISP* band, one with all the *NISP* bands, and finally the original OU-66 configuration. In addition, OU-200 and a variation of OU-200, where the input is a single-band VIS image, are used. These seven models are trained independently and the output of each model for the lens class is combined into a seven-element vector.

The 7D combined model outputs were dimensionally reduced to 2D using a T-SNE (Maaten & Hinton 2008). This embedding is shown in Fig. 9 where blue dots represent inputs containing simulated lenses and red dots represent non-lenses. Inspecting the members of the clusters within this embedding revealed that they represent different types of lenses and non-lenses. To classify the images within this embedding this space was divided into a 100 × 100 grid. Each square within the grid was assigned a classification value between 0 and 1 based upon the fraction of lenses within the square. The test data were classified by deriving their locations in the embedded space and using the classification value assigned to the grid element at that location. In some cases, the embeddings for elements of the test set fell in grid elements that contained zero entries from the training data, and therefore no corresponding classification value. In such cases, the classification value was estimated using the mean of the eight surrounding grid values if they are defined. All of the test data

[2] The learning rate is a dimensionless scale factor that determines the step size in the optimization.

[3] Node specification: two INTEL XEON GOLD 5118 processors with 12 cores of 24 threads (2.30 GHz); 3D controllers NVIDIA Corporation GP100GL (Tesla P100 PCIe 12GB); CPU memory 251 GB.

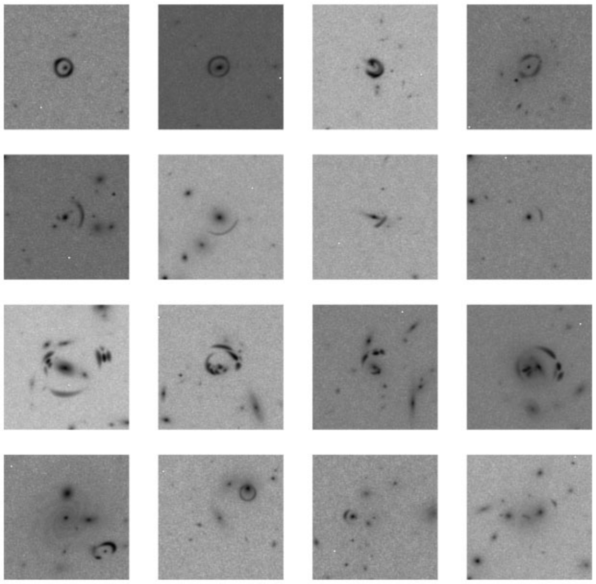

Figure 10. Examples of images within the groups shown in Fig. 9. Row 1: Images from group 1. Row 2: Images from group 2. Row 3: Images from group 3. Row 4: Images from group 4.

within a square were assigned the same value for their classification strengths.

The T-SNE algorithm has clustered model predictions into a 2D space as shown in Fig. 9; here four groups are highlighted. Examples of images in these groups are shown in Fig. 10. Group one consists of mainly small centred Einstein rings. Group two mainly consists of arcs placed above, below, and to the right of the central galaxy. Group three mainly consists of gravitational lenses with multiple components. Group four mainly consists of off-centred gravitational lenses.

4 RESULTS

The real-valued model outputs were converted into binary classifications by thresholding. Any output value below 0.5 is classified as a non-lens and output values that are greater than or equal to 0.5 are classified as lenses.

Based on comparisons between the classification results and the known ground truths for each of the EVAL set images, the performances of the models were evaluated by maximizing the F_β score. The F_β score (Chinchor 1992), defined as

$$F_\beta = \frac{\left(1+\beta^2\right) \times (\text{precision} \times \text{recall})}{(\beta^2 \times \text{precision} + \text{recall})} \tag{3}$$

is a weighted average of precision, defined as

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{4}$$

recall defined as

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{5}$$

and accuracy defined as

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}} \tag{6}$$

The ROC curve for the training set and the EVAL set are shown in Fig. 6. When the area under the ROC curve (AUROC) is 1 the model is a perfect classifier, when the AUROC is 0.5 the model is a random classifier.

Table 5. Metrics used to evaluate each model used in this paper, when the threshold is set to 0.5.

	F_β	Precision	Recall	AUROC	Accuracy
OU-66	0.9727	0.9758	0.2180	0.617	0.3308
OU-200	0.9687	0.9697	0.4423	0.695	0.5143
OU-T-SNE	0.9423	0.9431	0.4846	0.666	0.5371

The goal for this challenge was to try to create a fairly complete (albeit slightly contaminated) lens sample, which could be the basis of follow-ups. Therefore, we used $\beta = 0.03$ in model evaluation, which places greater weight on recall than precision. Other values of β could be well justified for other science goals, such as where the presence of false positives would represent a critical failure (e.g. JWST observations).

The data sets described in Section 2 were classified by the models OU-66, OU-200, and OU-T-SNE. The metrics used to evaluate these models on the EVAL data set are shown in Table 5 and the recalls of these models on all the data sets used in this paper are shown in Table 6. The differences in recall between the single lenses and the compound lenses could be due to the differences between the training data and the compound lens data. The arcs and rings in the compound lens data tend to be redder than the training data, there are no red rings or arcs in the training data. The colour difference could be causing the model to struggle with some lens configurations. Moreover, the compound lens images do not contain any additional background galaxies unlike those in the COMP and EVAL data sets.

The model outputs for each of the known compound lenses are shown in Table 7.

5 INTERPRETABILITY OF THE CLASSIFICATIONS

This section describes a set of investigations which were conducted in order to help explain how OU-66 and OU-200 make their decisions when classifying images. Five main techniques are used in this paper to develop an understanding of the model:

(i) Covering parts of the image and recording how the model responds (i.e. occlusion mapping; Zeiler & Fergus 2014).

(ii) Visually inspecting the images which highly activate kernels within the model (Erhan et al. 2009).

(iii) Generating and inspecting images to highly activate either of the target classes (Simonyan et al. 2013; Yosinski et al. 2015).

(iv) Applying *Deep Dream* (Mordvintsev et al. 2015) to amplify features within simulated images that correspond to 'lens' and 'non-lens' features.

(v) Applying *Grad-CAM* (Selvaraju et al. 2017) to highlight regions of input images that strongly activate for the 'lens' class.

In this section, the *NISP* bands are combined into an RGB image where red is the *H* band, green is *J* band, and blue is *Y* band. Yellow is a summation of the *H* and *J* bands, cyan is a combination of *J* and *Y* bands, and magenta is a combination of *H* and *Y* bands. The majority of the code used to generate the representations shown in this section is adapted from (Ozbulak 2019).

Table 6. Recall of the models when the threshold is set to 0.5.

	OU-H	OU-J	OU-Y	OU-JYH	OU-VIS	OU-66	OU-200
Train data	0.4476	0.3636	0.2585	0.3170	0.2597	0.3807	0.4004
Test data	0.4772	0.3989	0.2752	0.3555	0.2495	0.2180	0.44.23
Arcs	0.0706	0.5111	0.4083	0.7771	0.1684	0.6291	0.7505
Double rings	0.0908	0.5393	0.4667	0.7461	0.1618	0.4742	0.5567

Table 7. Model predictions on known *HST* and HSC data.

Lens name	OU-66	OU-200
SL2OS J02176−0513	0.9993	0.9852
HSC J142449−005322	0.9302	0.7104
J1148+1930	1.0000	0.9953
SDSS J0946+1006	0.6873	0.8228

5.1 Occlusion maps

Occlusion mapping involves recording the change in models' responses to input images as some of their pixels are masked out. If the model output for the 'lens' class using the masked image decreases relative to that when using its unmasked counterpart then it is more likely that the occluded pixels are associated with a lensing feature (Zeiler & Fergus 2014).

In the case of OU-200 the mask was 4 × 4 pixels for OU-200, while for OU-66 it was 1 × 1 pixels. Model outputs were recorded with the mask placed at positions on a grid with 4 pixel intervals in the horizontal and vertical directions for OU-200 and 1 pixel intervals for OU-66.[4]

The blue pixels in Figs 11, 12, and 13 indicate areas where occlusion reduces the model output for the 'lens' class, suggesting that the occluded pixels form part of a feature that is associated with lensing. Conversely, red pixels in Figs 11, 12, and 13 indicate areas where occlusion increases the model output for the lens, suggesting the occluded pixels form part of a feature associated with the 'non-lens' class.

5.1.1 Single lenses

Fig. 11 shows the computed occlusion maps for both models (rescaled as necessary) overlaid on log-scaled VIS images. The top two rows show the maps for OU-66 and the bottom two rows show the maps for OU-200.

For OU-66 and OU-200, the occlusion maps for images for which the models strongly predict the 'lens' class tend to highlight the arcs and rings in the image in blue. This suggests that the model has learnt to associate rings and arcs with gravitational lensing. For images for which the models strongly predict the 'non-lens' class, the main patterns seen are a red region in the centre of the image, surrounded by a faint disjointed blue ring roughly 10 arcsec in diameter. It is possible that these faint blue rings arise because the training data contain a large number of 'lens' images that have central lenses with roughly the same Einstein radius as these blue rings. This could result in the model trying to find a lens at this position within the noise of the data when a lens is not detected by the model. Spiral galaxy examples are also shown; the occlusion maps show arcs within the spiral arms, but these are identified as non-lens features by the model. This might mean that the model does not blindly associate arcs with lensing but also considers the context in which the arcs appear.

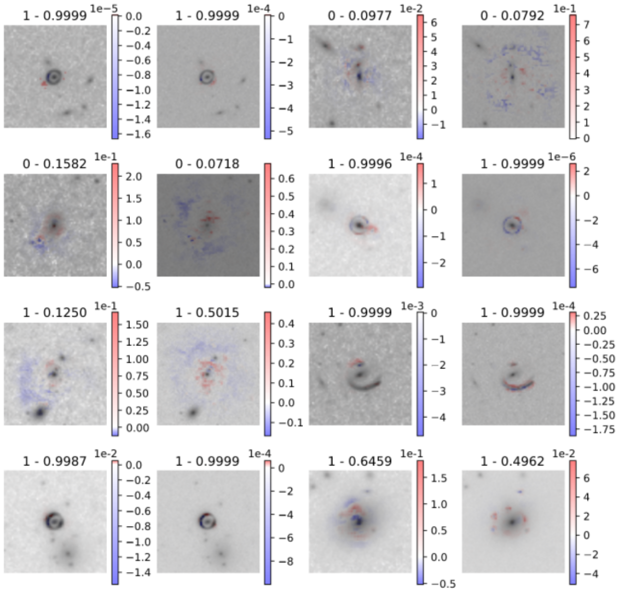

Figure 11. Occlusion maps of SGLC2 data on OU-66 and OU-200. The first and third columns show the occlusion maps for OU-66, the second and fourth columns show the occlusion maps for OU-200 for the same images to their left. The left number above the image indicates if the image is a lens (1) or a non-lens (0) and the right number is the CNN output for that image. The occlusion maps are independently scaled to highlight the change in output for each image, where the minimum value is blue, the maximum value is red, and the value of 0 is white. Blue indicates that the model associates the occulted feature with lensing and red indicates that the model associates the occulted feature with the non-lensing class.

5.1.2 Compound lenses

The occlusion maps for the simulated compound arcs are shown in the top two rows and double rings are shown in the bottom two rows of Fig. 12. The compound lens images for which the model predicts $\lesssim 0.3$ for the 'lens' class tend to have a central red ring or filled circle indicating that the model considers the pixels in this area to disfavour a lensing feature. Surrounding this area is a large blue region indicating that the model could be looking for rings and arcs at this diameter, which is similar to that of the faint blue rings shown in the 'non-lens' images Fig. 11. The images in this data for which the model predicts $\gtrsim 0.99$ are similar to the examples shown in Fig. 11. The occlusion maps highlight lensing areas along the arcs

[4] Different grid sizes were used for the two models because the inputs for OU-200 include resampled images from the NISP camera which have a native pixel size three times larger than that of VIS. Occluding fewer than four pixels in the OU-200 inputs would leave redundant information from the resampled NISP images unmasked.

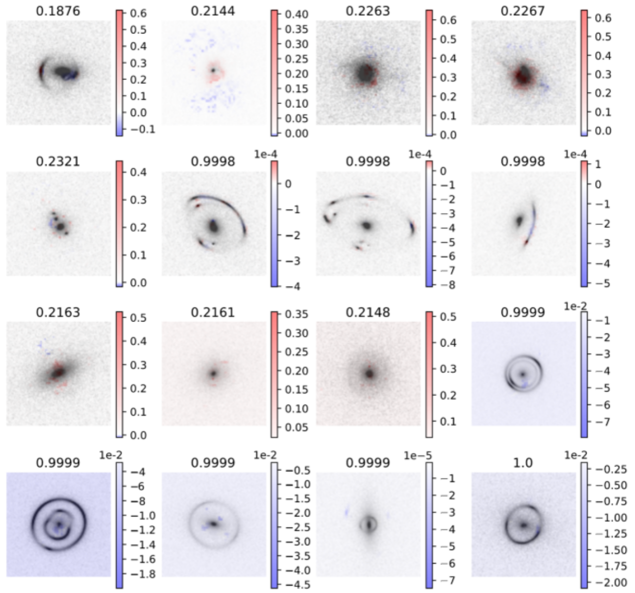

Figure 12. Occlusion maps of simulated compound arcs and simulated double rings. The top two rows show the occlusion maps for simulated compound arcs and the bottom two rows show the occlusion maps for simulated double rings. The number above the image is the model output for that image. The occlusion maps are independently scaled to highlight the change in output for each image, where the minimum value is blue, the maximum value is red, and the value of 0 is white. Blue indicates that the model associates the occulted feature with lensing and red indicates that the model associates the occulted feature with the non-lensing class.

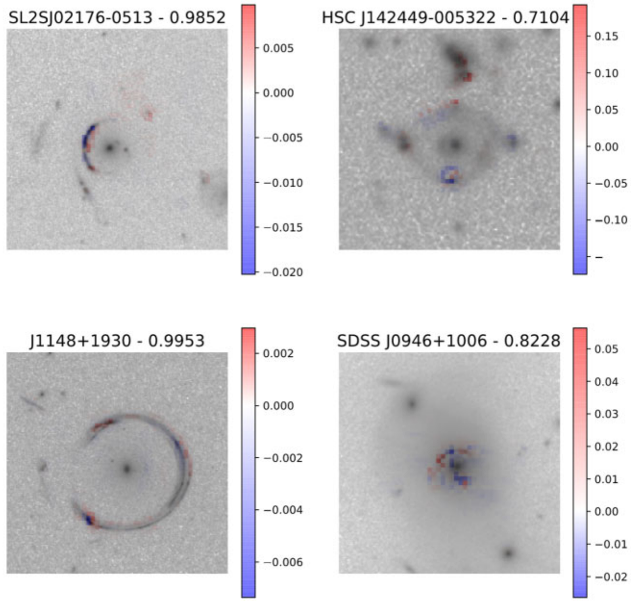

Figure 13. Occlusion maps of known compound lenses. The left number shows the lens name for the image, and the right number is the model output for that image. The occlusion maps are independently scaled to highlight the change in output for each image, where the minimum value is blue, the maximum value is red, and the value of 0 is white. Blue indicates that the model associates the occulted feature with lensing and red indicates that the model associates the occulted feature with the non-lensing class.

and rings, with a strong non-lensing feature on the boundary of these arcs.

The double ring images for which the model predicts $\lesssim 0.3$ for the 'lens' class in Fig. 12 tend to have strong non-lensing features in the centre of the image and there are occasionally low-level lensing features at the same radius as seen in the compound arc images. The images in this data for which the model predicts $\gtrsim 0.99$ produce occlusion maps which are qualitatively different from those shown in Fig. 11; these maps tend to be relatively robust to the occlusion map process. There is a small number of mask locations that have a dramatic effect on the occlusion map that are either in the centre of the image or along the arcs of the lens. Masking the rest of the image induces a roughly equal suppression of the 'lens' class output value.

5.1.3 *Known compound lenses*

The occlusion maps for how OU-200 responds to the known compound lenses are shown in Fig. 13.

The occlusion map for SL2SJ02176-0513 highlights the arc with a strong lens response and the boundary of the arc has a strong non-lens response. The occlusion map for HSC J142449-005322 appears to be highlighting the quadruply lensed quasars in the image. This is unexpected behaviour as the training data only contains rings and arcs. The occlusion maps suggest that OU-200 may be identifying each quasar as an individual Einstein ring in the image, instead of identifying the two Einstein rings surrounding the central galaxy. In particular a clear ring-like feature is visible in the occlusion map close to the bottom quasar image. Another possibility is that no single 4 × 4 pixel square significantly affects the output of this image because of how large the Einstein rings are. Hence, the model can compensate based on areas where the image is not occluded.

The occlusion map for J1148+1930 highlights three main sections of the central Einstein ring. This is similar to the occlusion maps shown in 11, where the model seems to detect the arcs in the image as lensing features. Again, the boundaries of the arcs seem to be detected as non-lens features.

The occlusion map for SDSS J0946+1006 is difficult to diagnose. This is probably due to the spatial extension of the lensing features being relatively small compared to the size of the occlusion mask. It appears that the top arc is considered mainly as a non-lens feature and the bottom arc is mainly considered to be a lens feature. The fainter arcs of the large Einstein ring do not appear to be making any significant changes to the lens classification.

5.2 Images that activate the kernels

Convolutional kernels are data structures associated with individual layers of a CNN that associate the trainable parameters (weights) of that layer with adjacent sets of pixels in its inputs.

For the first layer in the CNN, the kernel parameters map directly to adjacent sets of pixels in the input images. In later layers, each parameter maps to a larger region of input image pixels.

As the CNN trains, kernels in the first layer learn representations simple shapes and colour relationships that are directly visible in the image pixels. Kernels in subsequent layers learn how to combine these into more abstract representations including, for example, adjacency and the global context of a simple feature within the image. Identifying which features the kernels have learnt over the course of training could help illuminate what features are important to the models' classification.

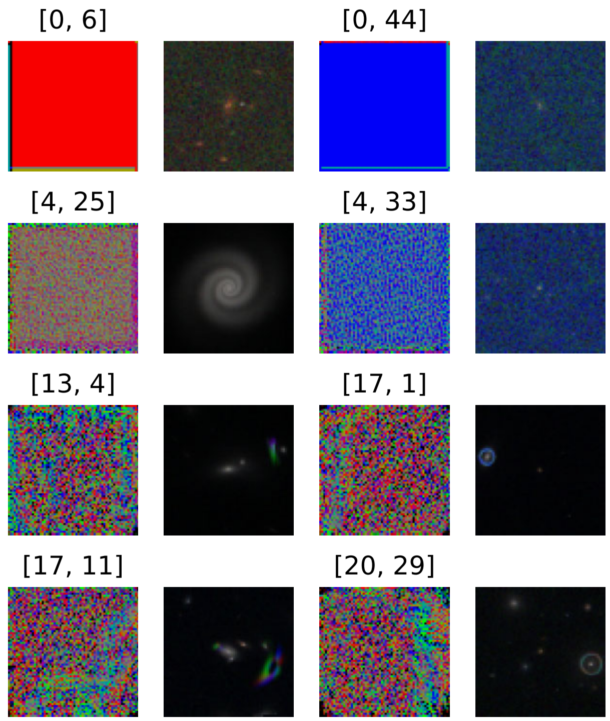

Figure 14. OU-66, first and third columns: Deep Dream creations that highly activate the kernel. Second and fourth columns: The NISP image from the single-lens data set that most activates the kernel. The square brackets indicate the layer number and kernel number for the kernel.

One of the ways to do this is to calculate the response of each kernel to each image. This is done by taking the mean value of the feature map generated by a kernel.

The mean value of a feature map (before apply ReLU activation) generated by a kernel is how the activation of a kernel is evaluated. This is the kernel activation value (KAV). This is done to condense the output of a kernel into a single value. Images where the KAV is high, trigger a significant response in the kernel these are considered to highly activate that particular kernel. By comparing the images that have a high KAV for a given kernel, it can be inferred what a given kernel responds to. As to be expected with a CNN in the early layers of these models identify broad features of the image, such as the predominant colour of the image or if a large central galaxy is present. Looking deeper into the model more complex features start to emerge such as the position of a gravitational lens in the image or if the lensing feature is an arc or ring.

Groups that could be identified from this approach include large central galaxies, large Einstein rings, arcs, small blue Einstein rings, off-centred Einstein rings, the position of large arcs, and the position of small blue Einstein rings. The large central galaxies that take up the majority of the image tend to be white, they are both spiral and elliptical, and the spiral galaxies can be either face or edge on.

Figs 14 and 15 show in the right column the image which has the highest KAV. These figures show a range of the groups described above. In Fig. 14 [layer 0 kernel 6] shows an example of the group where the majority of galaxies within the image are red. [layer 0 kernel 44] and [layer 4 kernel 33] show a group where the noise level is high in the blue bands. [layer 4 kernel 25] shows the group that consists of large spiral galaxies that take up the majority of the image. [layer 13 kernel 4] and [layer 20 kernel 29] show the group where the lensing feature is in the right third of the image. [layer 17 kernel 1] shows the group where the lensing feature is in the left third of the image. [layer 17 kernel 11] is the group where the lensing feature is in the bottom-right corner of the image. In Fig. 15, [layer 4 kernel 29] shows the group which consists of a white pale blue central elliptical galaxy. [layer 8 kernel 1] shows the group where the galaxies in the image are red. [layer 12 kernel 18] shows the group which consists of central Einstein rings.

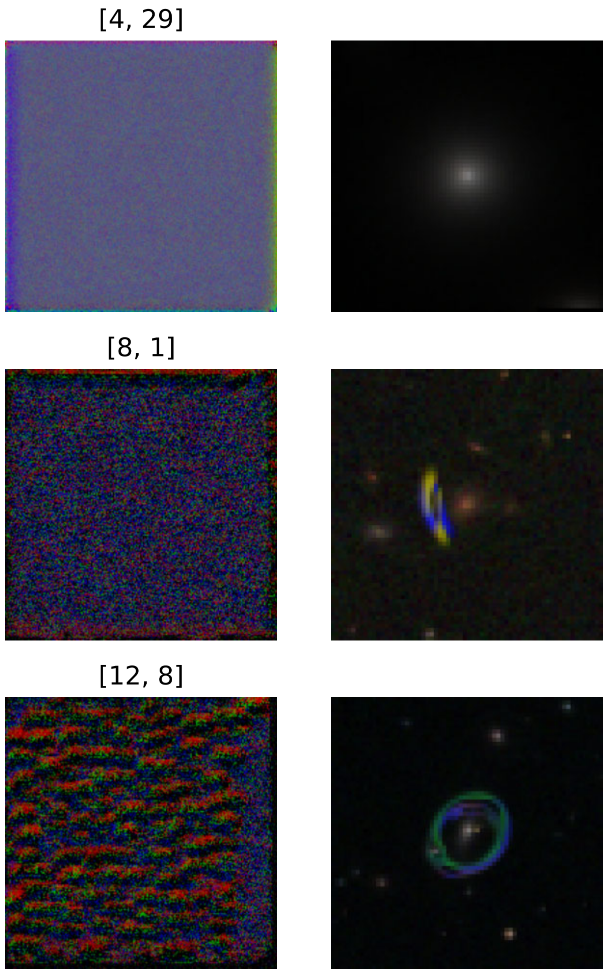

Figure 15. OU-200, first and third columns: Deep Dream creations that highly activate the kernel. Second and fourth columns: The NISP image from the single-lens data set that most activates the kernel. The square brackets indicate the layer number and kernel number for the kernel.

5.3 Class-generated images

To understand what image features the model may be associating with lenses and non-lenses images were generated based upon the target class for the model (Simonyan et al. 2013; Yosinski et al. 2015). The model weights were frozen and the output was maximized for the target class. Through back-propagation the input image was updated. This results in an updated image that activates the target class more strongly than the unmodified input image. This process was repeated to saturate the activation of the target class. This artificially generated new features in the image that cause a strong response for the target class. It is likely that these new features emulate those that the model has learnt to identify within the target classes.

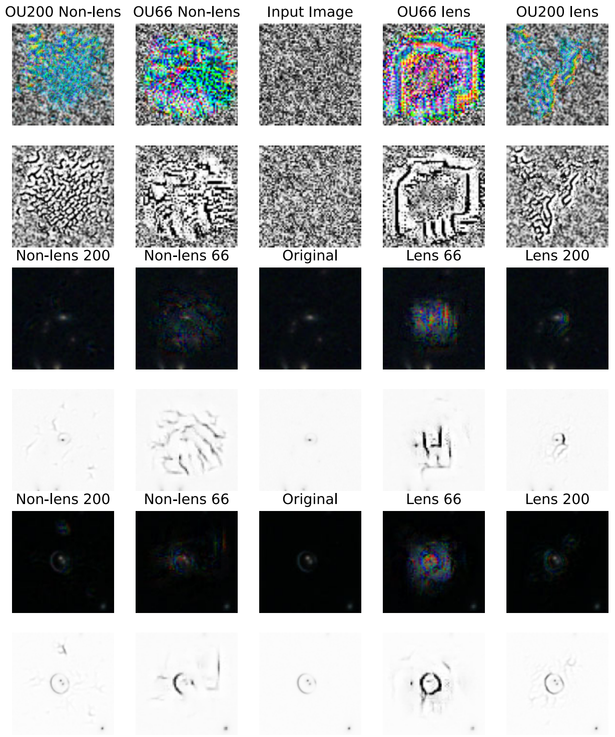

Figure 16. Rows 1 and 2: An example of CGIs generated by both models for each target class. Rows 3 and 4: Deep Dream images created by both models for each target class starting from an image of a non-lens. Rows 5 and 6: Deep Dream images created by both models for each target class starting from an image of a lens. The *NISP* bands for the Deep Dream images have had their brightness increased by 150 per cent for clarity of the features generated.

For OU-66, all four channels of the original input to this model are identical 66 × 66 pixel arrays containing random uniform noise on [0,1]. This process is repeated for OU-200 and the image is scaled up to 200 × 200 pixels.

Modified images were generated from random noise inputs using OU-66 and OU-200 for both classes. A learning rate of 0.1 was applied and the input image was updated by back-propagation 5000 times. The images produced at the end of this process can be seen on the top two rows of Fig. 16. Both models produce similar outputs when generating a non-lens, creating a large blue and green blob in the centre of the image. This suggests that the model considers a non-lens to have large blue features in the centre of the image. The models generate different images for a lens. OU-66 creates a large blue Einstein ring that appears to be spatially offset in the four image bands. The area inside the ring is predominately red indicating that the model expects to see a red source galaxy in this region. OU-200 creates two separate structures, a small blob in the top left of the image and another diagonal feature going from bottom left to top right. The majority of the generated features are blue, but there are strong yellow and red curved features diagonally across the centre of the image. This could suggest that OU-66 is creating large central Einstein rings and that OU-200 is creating arc features within the image.

5.4 Deep Dream

Deep Dream is a technique that is similar to CGIs (Mordvintsev et al. 2015), but instead of the input being random data one of the simulated single-lens images is used.

The learning rate used was 0.001 and this process was allowed to run for 1000 iterations. Examples of this technique being applied to a lens image and a non-lens image are shown in the bottom four rows in Fig. 16.

Rows 3 and 4 of Fig. 16 show Deep Dream images that result when the input image was a simulated non-lens, and ‘non-lens’ was used as the target class. Deep Dream images created using OU-200 for ‘non-lens’ images and target class differ very little from their inputs. In contrast, the corresponding Deep Dream images for OU-66 exhibit blue green diagonal features that could be representing galaxies. When ‘lens’ is the target class both models created features around the central galaxy when using ‘non-lens’ input images. OU-66 created a ‘blocky’ arc below the central galaxy, while OU-200 created a rounder arc to the right of the central galaxy. Both models have a colour gradient across these generated features suggesting that the model is activated by a change in colour across features.

Rows 5 and 6 of Fig. 16 show Deep Dream images that were generated when the input image was a simulated single lens, and ‘non-lens’ was used as the target class. In the OU-200 Deep Dream images the Einstein ring becomes red suggesting that this model heavily relies on colour for classification. Several round artefacts appear in the image, which might indicate that the model has learnt to associate multiple flux peaks in an image with the ‘non-lens’ class. The Deep Dream image for OU-66 also fragments the Einstein ring into two sections and appears to straighten both sections. The resultant image begins to resemble edge-on spiral galaxies, indicating that the model may have learnt that the presence of such galaxies in images is also likely to indicate the ‘non-lens’ class. This is reasonable since the majority of gravitational lenses in the simulated training data (and in reality) are large elliptical galaxies. Unlike OU-200, the Deep Dream image for OU-66 shows little evidence for colour modification which indicates that the model places less weight on colour and bases its classifications primarily on morphological information.

When ‘lens’ was the target class both models make the Einstein ring brighter and bluer and tend to join previously incomplete arcs into complete Einstein rings. OU-66 often creates a red and yellow feature inside the Einstein ring. This feature could indicate that the model associates red central galaxies with images in the ‘lens’ class. Again this is consistent with the prevalence of gravitational lenses that are the haloes of red elliptical galaxies.

5.5 Generating images to highly activate kernels

Although it is often used to produce images that activate a specific target class, the Deep Dream technique can also be used to generate images that activate a specific kernel within the model (see e.g. Simonyan et al. 2013; Yosinski et al. 2015). This will be referred to as generated kernel images (GKI).

The central columns of Figs 14 and 15 show images that activate the kernels illustrated in the left-hand columns of those figures. The descriptions below outline some of the features these kernels detect and what they could relate to in the input images.

As can be seen in Fig. 14, [layer 0 kernel 6] creates a solid red block with some other colours along the edge. [layer 0 kernel 44] creates a solid blue block with some other colours along the edge. [layer 4 kernel 25] creates a mixture of pink and green pixels. [layer 4 kernel 33] creates a blue noise that appears very similar to the images which are highly activated by this corresponding kernel. [layer 13 kernel 4], [layer 17 kernel 1], [layer 17 kernel 11], and [layer 20 kernel 29], all of these Deep Dream images have the same overall feature, where the majority of the image is red apart from a section of the image which is very blue and green. The difference between these GKIs is the location and shape of these blue features. The position of these blue features corresponds to the location of the lens within the highly activating image.

In Fig. 15, [layer 4 kernel 29] creates a pale blue feature with a low level of noise. [layer 8 kernel 1] creates an image with a black background with a mixture of noise that is red, green, and blue. [layer 12 kernel 8] is very similar to [layer 8 kernel 1]; however, this has several horizontal 'sand ripples' that are green and red which are slightly offset vertically from each other. This feature could be trying to detect horizontal changes in colour.

5.6 Guided Grad-CAM

Guided Grad-CAM (Selvaraju et al. 2017) is a common technique used for CNN interpretability. It is well known for its association with the ImageNet challenge, where it has highlighted areas of images that correspond to areas that highly influence the determination of an output class in various models. The technique works by taking a weighted average of all the feature maps in the last convolutional layer with ReLU applied. This creates a Grad-CAM image highlighting locations in the image where the CNN detects features of interest. This Grad-CAM image only shows features that increase the model output value for the target class. Guided back-propagation (Springenberg et al. 2014) is applied to the image, which sets the desired class to 1 and all other outputs to 0. Only the positive gradients are back-propagated through the model (negative gradients are clipped at zero) to show areas of the image that have a positive effect on the classification. Once both processes are complete the results of guided back-propagation and Grad-CAM are multiplied together to generate a Guided Grad-CAM image.

Guided Grad-CAM was applied to a set of images for the OU-200 model. In Fig. 17, the input images are shown in rows 1, 3, 5, and 7. Rows 2, 4, 6, and 8 show the pixels from the input image when the pixel value in that position in the Guided Grad-CAM image is greater than or equal to 0.05. The majority of these Guided Grad-CAM images highlight the gravitational lens feature in the image. Guided Grad-CAM finds parts of both rings in four of the six double ring images in Fig. 17. For the known compound lenses, Guided Grad-CAM selects large sections of the gravitational lensing features for SL2S J02176−0513 and SDSS J1148+1930. For SDSS J0946+1006, Guided Grad-CAM mainly selects the brightest lensing arc and for SL2SJ02176−0513 small sections of the Einstein ring are selected. In most of the known compound lens images Guided Grad-CAM does not find features from both lenses in the image, with the exception of SDSS J0946+1006 where a small area of the fainter lens is clearly highlighted.

6 DISCUSSION

In this paper, several deep learning models have been applied to classify simulated single lenses and non-lenses, simulated compound lenses, and previously discovered compound lenses. Models that process multiple bands (in particular those that include the bluer *VIS* band) with higher angular resolution like OU-200 tend to improve the recall for the gravitational lens class. Providing high-resolution multiband input images provides the model with additional information during training, which allows it to make more accurate predictions when determining whether an image contains a lens. OU-200 was shown to adapt well to a simulated compound lens data set that the model had not been trained on. This indicates that training models to accurately identify single-lens systems does not necessarily preclude their use for identifying compound lens systems. Multiband models (OU-66 and OU-200) are able to classify all the previously discovered compound lenses shown in this paper as lenses. These models can adapt from simulated images to observational data including *HST* and HSC images, suggesting that they will need little retraining to perform well on real images in future upcoming surveys. This has previously been shown by the CNN described in (Petrillo et al. 2019) which was trained on KiDS data and performed second best in the strong gravitational lens challenge (Metcalf et al. 2019) without retraining, demonstrating that these CNNs can find gravitational lenses outside of the data they have been trained on.

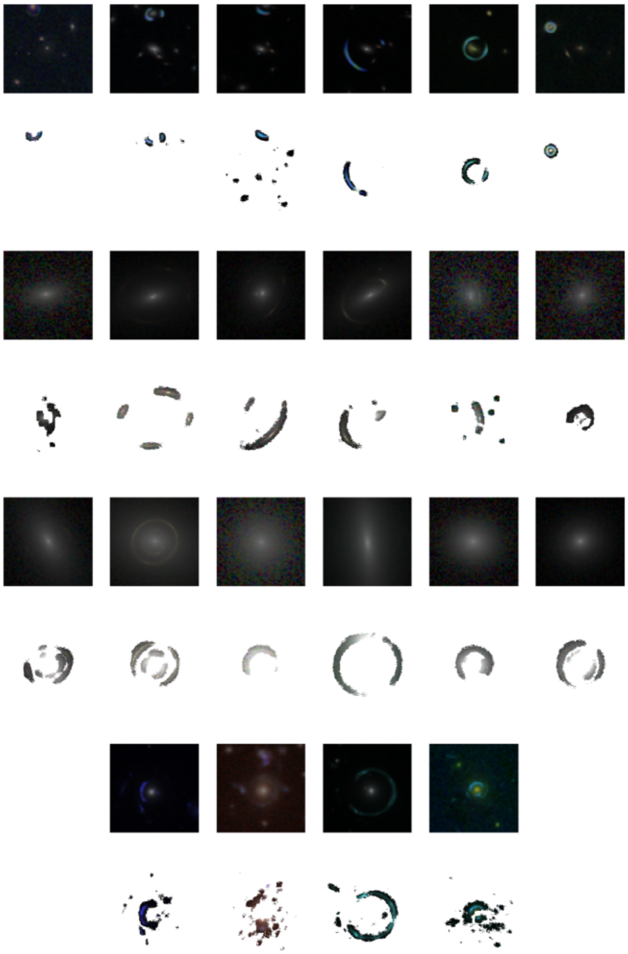

Figure 17. Top row: Guided Grad-CAM of single lens. Second row: Guided Grad-CAM of compound arcs. Third row: Guided Grad-CAM of double rings. Bottom row: Guided Grad-CAM of known compound lenses.

In Section 5, the factors affecting the classifications made by the models in this paper were investigated. Features associated with lensing such as Einstein rings and arcs, significantly influence the output of these models, which implies that they have correctly learnt to associate these features with the lens class. Based on the Deep Dream images that were generated to activate specific kernels (Section 5.5), the CGIs (Section 5.3), and the Deep Dream images (Section 5.4) it is clear that colour is important to the models' classifications of the images. Inspecting the GKIs revealed kernels in the deep layers of the models are strongly activated by blue regions on red backgrounds at the location of the gravitational lens. For OU-66, the CGIs (Section 5.3) exhibit blue ring-like structures enclosing a red central region. This indicates that this model has learnt that blue ring-like structures surrounding central red features, which are typical of many gravitational lensing systems, are more likely to indicate the image contains a gravitational lens. Pessimistically, this might indicate that this model would struggle to classify images with red source galaxies as containing a gravitational lens and more generally that the model may have a colour bias. This colour bias has also been reported in (Petrillo et al. 2019; Jacobs et al. 2022). This is not necessarily a bad thing for the model to have learnt, since the majority of known gravitational lens systems have blue source galaxies and red lensing galaxies. Nevertheless, it selects against systems that are not typical of currently known lenses.

The only way to begin to understand the biases within methods such as CNNs for gravitational lens detection is to investigate the interpretability of these methods. This is possible through a variety of means such as investigating the parameters learnt by these models directly as shown in this paper or by manipulating the input images via a sensitivity probe (Jacobs et al. 2022). The use of a sensitivity probe has been able to show how susceptible CNNs can be to minor modifications to the input data. The indications of small decisions significantly affecting the performance of CNNs are present. Features such as the PSF can significantly affect the ability of a CNN to learn data (as we discuss later on) or the ability of a CNN to adapt to new data.

Accordingly, OU-200 has learnt relevant details about typical gravitational lenses, but this also shows that this model may not adapt well when presented with a gravitational lens system where the source galaxy is not blue. The class-targeted Deep Dream images (Section 5.4) also exhibit blue, ring-like features surrounding red central regions. Moreover, when targeting the ‘lens’ class with a lens image as input, the lens is emphasized and becomes bluer. When targeting the non-lens class using a lens image as input the Einstein ring becomes fragmented in OU-66 and redder in OU-200, indicating that the shapes of features are more important to OU-66 when classifying images, while instead, colour information is more strongly weighted by OU-200. OU-200 has been shown in Fig. 16 to actively associate the ‘non-lens’ class with red Einstein rings. Whereas the CGIs of the ‘lens’ class generate blue Einstein rings, suggesting that the model may not classify images with red Einstein rings as lenses.

This hints at a fundamental limitation to the current approaches to lens finding with missions and surveys such as *Euclid* and LSST: The training sets used for the algorithms may inadvertently select against non-negligible populations of strong gravitational lenses that exist on the sky. There are now several known examples of gravitational lensing systems that exhibit red Einstein rings (Geach et al. 2015), and there are growing numbers of known gravitationally lensed dusty star-forming galaxies (e.g. Negrello et al. 2010, 2017; Spilker et al. 2016; Rivera et al. 2019; Neri et al. 2020; Urquhart et al. 2022). The surface density of submm-bright gravitational lenses (e.g. 500 μm flux densities $S_{500} \geq 100$ mJy) is relatively low, with only one lens per 7.5 deg^2 (e.g. Negrello et al. 2017), compared to e.g. ∼50 lenses per square degree in COSMOS (e.g. Faure et al. 2008; Jackson 2008), but these are only the lensed submm galaxies that are easy to identify in blank-field surveys. At submm flux densities ∼10 × fainter, the predicted surface density of strong lenses is expected to be about two orders of magnitude higher, e.g. ∼10 strong lenses per deg^2 with $S_{500} \geq 10$ mJy (e.g. Trombetti et al. 2021). This implies ∼ 150 000 strong submm-wave lenses over the entire *Euclid* survey area, comparable to the total expected to be found with the *Euclid* VIS instrument (e.g. Collett 2015). As one probes fainter submm luminosities and lower dust obscurations, it is reasonable to expect the background sources to become increasingly detectable at near-infrared wavelengths, suggesting the possibility of a non-negligible exotic population of lensed red galaxies in wide-field surveys such as *Euclid*. Furthermore, extreme ultra-high-redshift-lensed galaxies will be preferentially detected only at near-infrared wavelengths, and though very rare on the sky (e.g. Marchetti, Serjeant & Vaccari 2017; Vikaeus et al. 2021), will be high-priority targets for follow-ups. Our results show that it is not yet clear whether such red background-lensed sources will be systematically selected against with default lens finders.

On the other hand, the sensitivity to morphological features has proven promising for the discovery of compound lens systems. The Guided Grad-CAM images shown in Fig. 17 highlight the lensing features in the images for the majority of the examples shown. This indicates that the model has learnt to look for indicative features in the image such as rings and arcs. The CNNs presented in this paper appear to be sufficiently flexible that they correctly classify exotic systems with multiple lensing planes, despite no such systems appearing in the data sets on which they were trained. This flexibility is encouraging because it suggests that similar exotic systems that exist in the real Universe will not be missed when the models are applied to real upcoming observational data. Nevertheless, while the successes in detecting compound lenses are encouraging, it is likely that improvements could be made by training the networks explicitly to recover them.

When initially training the CNNs in this paper there was a class imbalance in the training data. To balance this data set the non-lenses were rotated by 90, 180, and 270 deg and flipped along the horizontal axis then rotated by 90, 180, and 270 deg. This caused the model to classify images which had not been augmented as lenses, and any image which had been augmented was classified as a non-lens. The model had learnt to classify the images based upon their orientation. It was realised that the model could determine the orientation image based on the orientation of the simulated *Euclid* PSF, i.e. its three-fold symmetry meant the model could separate images based on the augmentations that had been applied to it. This highlights the sensitivity of machine learning techniques like CNNs to very subtle, and scientifically irrelevant, image features that would be completely imperceptible for human inspectors. We would like the network outputs to be invariant with respect to the astrophysical orientation on the sky, but the detector orientation imposes rotational asymmetry by virtue of the PSF shape. One possible solution could be to add the detector orientation on the sky as an auxiliary input to the network training and the prediction. This would reintroduce the ability to learn rotational invariance from the training data by augmenting them using rotation, or allow the application of group-equivariant CNNs (Scaife & Porter 2021).

The application of machine learning to lens finding in this paper has been predicated on the intractability of exhaustive human inspection, but even an apparently perfect performance on simulated data need not imply that there is no useful contribution to be made by human volunteers (e.g. Marshall et al. 2016; More et al. 2016; Sonnenfeld et al. 2020). The sample of candidate gravitational lenses is likely to be of a tractable size for volunteers or experts to examine every one. This may identify underperformance or other problems for known rare subpopulations, which may in turn be used to refine or retrain automated lens finders, in a virtuous circle between human and machine vision. Human inspection is also currently the only widely applied method for finding new, unanticipated categories of exotic systems. Machine learning approaches are in development to detect anomalies in astronomical imaging, such as training a generative adversarial network (GAN) to encode the space of commonly occurring features in real astrophysical image (e.g. Storey-Fisher et al. 2021). Atypical images can then be identified as those for which the discriminator of the GAN rejects the image. This has succeeded in identifying images affected by instrumental defects and/or artefacts, but there are still formidable obstacles in representing the diversity of (for example) galaxy morphologies, and 'interesting' anomalies are still a matter of human judgement.

7 CONCLUSIONS

One of the main advantages of very large samples of strong gravitational lenses is the potential for discovery of rare lens configurations, among which the compound lens systems have the added bonus of providing new cosmological parameter constraints (e.g. Gavazzi et al. 2008). This places the detection of these systems at a premium within survey projects such as *Euclid* and LSST. Our analysis has shown that existing lens finding methodologies should be able to recover compound lensing systems even without further training, albeit not with 100 per cent recall. Ultra-high-redshift-lensed sources will be at a similar high premium, but it is less clear that these red-lensed systems will be detectable by the lens finder presented in this paper. It seems clear that, in order to be able to exploit the advantages of a very large strong-lensing sample, attention must be paid to training the lens finding algorithms to accommodate the anticipated interesting subsets. For the 'unknown unknowns', however, the data mining problem remains intractable by definition, and there may be no option other than a coordinated programme of human inspection such as through citizen science (e.g. Marshall et al. 2016; More et al. 2016; Sonnenfeld et al. 2020).

ACKNOWLEDGEMENTS

JW was supported by the Science and Technology Facilities Council under grant number ST/P006760/1, the DISCnet Centre for Doctoral Training in Data-Intensive Science. SS and HD were partly supported by the ESCAPE project; ESCAPE - The European Science Cluster of Astronomy & Particle Physics ESFRI Research Infrastructures has received funding from the European Union's Horizon 2020 - Research and Innovation Framework Programme under grant agreement no. 824064. SS also thanks the Science and Technology Facilities Council for financial support under grant ST/P000584/1.

DATA AVAILABILITY

The code used in this paper is available at https://github.com/JoshWilde/LensFindery-McLensFinderFace.

REFERENCES

Abell P. A. et al., 2009, preprint (arXiv:0912.0201)
Agarap A. F., 2018, preprint (arXiv:1803.08375)
Amara A. et al., 2021, J. Open Source Softw., 6, 3056
Auger M., Treu T., Bolton A., Gavazzi R., Koopmans L., Marshall P., Bundy K., Moustakas L., 2009, ApJ, 705, 1099
Birrer S., Amara A., 2018, Phys. Dark Universe, 22, 189
Birrer S., Treu T., 2020, A&A, 649, A61
Bolton A. S., Burles S., Koopmans L. V., Treu T., Gavazzi R., Moustakas L. A., Wayth R., Schlegel D. J., 2008, ApJ, 682, 964
Cheng T.-Y., Li N., Conselice C. J., Aragón-Salamanca A., Dye S., Metcalf R. B., 2020, MNRAS, 494, 3750
Chinchor N., 1992, Proc. 4th Message Underst. Conf., MUC-4 Evaluation Metricserence. Assoc. Comput. Linguist., USA, p. 22
Collett T. E., 2015, ApJ, 811, 20
Collett T. E., Auger M. W., 2014, MNRAS, 443, 969
Collett T. E., Auger M. W., Belokurov V., Marshall P. J., Hall A. C., 2012, MNRAS, 424, 2864
Davies A., Serjeant S., Bromley J. M., 2019, MNRAS, 487, 5263
Deng J., Dong W., Socher R., Li L.-J., Li K., Fei-Fei L., 2009, IEEE Conference on Computer Vision and Pattern Recognition, 248
Denzel P., Mukherjee S., Saha P., 2021, MNRAS, 506, 1815
Erhan D., Bengio Y., Courville A., Vincent P., 2009, Tech. Rep. Univ. Montréal, 1341, 1
Faber S., Jackson R. E., 1976, ApJ, 204, 668
Faure C. et al., 2008, ApJS, 176, 19
Gavazzi R., Treu T., Koopmans L. V., Bolton A. S., Moustakas L. A., Burles S., Marshall P. J., 2008, ApJ, 677, 1046
Geach J. E. et al., 2015, MNRAS, 452, 502
Geirhos R., Rubisch P., Michaelis C., Bethge M., Wichmann F. A., Brendel W., 2019, International Conference on Learning Representations
Hartley P., Flamary R., Jackson N., Tagore A., Metcalf R., 2017, MNRAS, 471, 3378
Hausen R., Robertson B. E., 2020, ApJS, 248, 20
He K., Zhang X., Ren S., Sun J., 2015, 2015 IEEE International Conference on Computer Vision (ICCV), p. 1026
He Z. et al., 2020, MNRAS, 497, 556
Hezaveh Y. D., Levasseur L. P., Marshall P. J., 2017, Nature, 548, 555
Huang X. et al., 2019, ApJ, 894, 78
Jackson N., 2008, MNRAS, 389, 1311
Jacobs C. et al., 2019, MNRAS, 484, 5330
Jacobs C., Glazebrook K., Collett T., More A., McCarthy C., 2017, MNRAS, 471, 167
Jacobs C., Glazebrook K., Qin A., Collett T., 2022, Astron. Comput., 38, 100535
Khramtsov V. et al., 2019, A&A, 632, A56
Kingma D. P., Ba J., 2015, International Conference on Learning Representations (ICLR)
Kochanek C., 2020, MNRAS, 493, 1725
Krizhevsky A., Sutskever I., Hinton G. E., 2012, Adv. Neural Inf. Process. Syst., 25, 1097
Lanusse F., Ma Q., Li N., Collett T. E., Li C.-L., Ravanbakhsh S., Mandelbaum R., Póczos B., 2017, MNRAS, 473, 3895
Laureijs R. et al., 2011, preprint (arXiv:1110.3193)
Lauritsen L., Dickinson H., Bromley J., Serjeant S., Lim C.-F., Gao Z.-K., Wang W.-H., 2021, MNRAS, 507 1546
LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D., 1989, Neural Comput., 1, 541
LeCun Y., Bottou L., Bengio Y., Haffner P., 1998, Proc. IEEE, 86, 2278
Li R., 2021, ApJ, 923, 16
Liao K., 2019, ApJ, 871, 113
Maaten L. v. d., Hinton G., 2008, J. Mach. Learn. Res, 9, 2579
Maggiori E., Charpiat G., Tarabalka Y., Alliez P., 2017, Proc. IEEE, 55, 4962
Mandelbaum R. et al., 2018, preprint (arXiv:1809.01669)
Marchetti L., Serjeant S., Vaccari M., 2017, MNRAS, 470, 5007
Marshall P. J. et al., 2016, MNRAS, 455, 1171
McCulloch W. S., Pitts W., 1943, Bull. Math. Biol, 5, 115
Metcalf R. B. et al., 2019, A&A, 625, A119
Mordvintsev A., Olah C., Tyka M., 2015, https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html URL (accessed 22/07/2020)
More A. et al., 2016, MNRAS, 455, 1191
Müller O., Schnider E., 2021, The Open J. Astrophys., 4, 3
Negrello M. et al., 2010, Science, 330, 800
Negrello M. et al., 2017, MNRAS, 465, 3558
Neri R. et al., 2020, A&A, 635, A7
Newton E. R., Marshall P. J., Treu T., Auger M. W., Gavazzi R., Bolton A. S., Koopmans L. V., Moustakas L. A., 2011, ApJ, 734, 104
Nguyen A., Yosinski J., Clune J., 2015, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), p. 427
Obermeyer Z., Powers B., Vogeli C., Mullainathan S., 2019, Science, 366, 447

Ozbulak U., 2019, https://github.com/utkuozbulak/pytorch-cnn-visualizations (accessed 14/02/2020)

Paszke A. et al., 2019, Proceedings of the 33rd International Conference on Neural Information Processing Systems, p. 8026

Pearson J., Pennock C., Robinson T., 2018, Emergent Scientist, 2, 1

Petrillo C. et al., 2019, MNRAS, 482, 807

Rivera J. et al., 2019, ApJ, 879, 95

Rumelhart D. E., Hinton G. E., Williams R. J., 1986, Nature, 323, 533

Samuel A. L., 1959, IBM Journal of Research and Development, 210

Scaife A. M., Porter F., 2021, MNRAS, 503, 2369

Schmitz M. A. et al., 2020, A&A, 636, A78

Schneider P., Sluse D., 2014, A&A, 564, A103

Schuldt S., Chirivì G., Suyu S., Yıldırım A., Sonnenfeld A., Halkola A., Lewis G., 2019, A&A, 631, A40

Selvaraju R. R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D., 2017, IEEE International Conference on Computer Vision (ICCV), 618

Shu Y. et al., 2017, ApJ, 851, 48

Simonyan K., Vedaldi A., Zisserman A., 2014, International Conference on Learning Representations

Sonnenfeld A. et al., 2020, A&A, 642, A148

Spilker J. S. et al., 2016, ApJ, 826, 112

Spindler A., Geach J. E., Smith M. J., 2021, MNRAS, 502, 985

Springenberg J. T., Dosovitskiy A., Brox T., Riedmiller M., 2015, International Conference on Learning Representations

Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., 2014, J. Mach. Learn. Res, 15, 1929

Storey-Fisher K., Huertas-Company M., Ramachandra N., Lanusse F., Leauthaud A., Luo Y., Huang S., Prochaska J. X., 2021, MNRAS, 508, 2946

Tanaka M. et al., 2016, ApJ, 826, L19

Treu T. et al., 2013, preprint (arXiv:1306.1272)

Trombetti T., Burigana C., Bonato M., Herranz D., De Zotti G., Negrello M., Galluzzi V., Massardi M., 2021, A&A, 653, A151

Tu H. et al., 2009, A&A, 501, 475

Urquhart S. et al., 2022, MNRAS, 511, 3017

van der Walt S. et al., 2014, PeerJ, 2, e453

Verma A., Collett T., Smith G. P.Collaboration S. L. S., et al., Collaboration S. L. S., 2019, preprint (arXiv:1902.05141)

Vikaeus A., Zackrisson E., Schaerer D., Visbal E., Fransson E., Malhotra S., Rhoads J., Sahlén M., 2022, MNRAS

Wong K. C. et al., 2020, MNRAS, 498, 1420

Yosinski J., Clune J., Nguyen A., Fuchs T., Lipson H., 2015, preprint (arXiv:1506.06579)

Zeiler M. D., Fergus R., 2014, Proc. Eur. Conf. Comp. Vis., Visualizing and Understanding Convolutional Networks. Springer, Cham, p. 818

Zhan H., Tyson J. A., 2018, Rep. Prog. Phys., 81, 066901

Zhang X., Feng Y., Chen H., Yuan Q., 2020, ApJ, 905, 97

This paper has been typeset from a TEX/LATEX file prepared by the author.