Emergent Scientist **2**, 1 (2018)
© J. Pearson et al., Published by EDP Sciences, 2018
https://doi.org/10.1051/emsci/2017010

Emergent Scientist

Available online at:
www.emergent-scientist.edp-open.org

RESEARCH ARTICLE OPEN ACCESS

Auto-detection of strong gravitational lenses using convolutional neural networks

James Pearson, Clara Pennock, and Tom Robinson*

The University of Nottingham, University Park, Nottingham NG7 2RD, UK

Received: 23 July 2017 / Accepted: 23 November 2017

Abstract. We propose a method for the automated detection of strong galaxy-galaxy gravitational lenses in images, utilising a convolutional neural network (CNN) trained on 210 000 simulated galaxy-galaxy lens and non-lens images. The CNN, named LensFinder, was tested on a separate 210 000 simulated image catalogue, with 95% of images classied with at least 98.6% certainty. An accuracy of over 98% was achieved and an area under curve of 0.9975 was determined from the resulting receiver operating characteristic curve. A regional CNN, R-LensFinder, was trained to label lens positions in images, perfectly labelling 80% while partially labelling another 10% correctly.

1 Introduction

All massive objects have gravitational fields that distort spacetime around them, with more massive objects producing stronger distortions. Light rays travel along the shortest path (a geodesic), and those passing through this distorted region change direction as they are curved around the object. Such large objects are called gravitational lenses, and include galaxies and galaxy clusters. These are capable of strong, noticeable lensing effects in the form of arcs of light around the lens, which are the result of light rays originating from a light source behind the lens, such as a distant galaxy. Furthermore, when the source and lens are perfectly aligned along an observer's line of sight the source light can be distorted to form a pattern around the lens known as an Einstein ring or cross.

The arcs and rings formed from gravitational lensing provide a useful way to probe the early Universe [1,2], as the distorted light is a magnified view of distant obscured galaxies. If the distortion can be removed, then the original appearance of the source galaxy can be obtained along with the mass profile of the lensing galaxy [3]. Gravitational lensing can help constrain the inner mass density profiles of galaxies [4], and when combined with redshift measurements, gravitational lensing features have the potential to aid galaxy evolution models.

The radius of the Einstein ring is dependent on the total mass of the lensing object, with the level of distortion providing information on the lensing galaxy's projected mass density. This, along with other techniques (e.g. galaxy rotation curves [5,6]), allows the proportion of dark matter in that region to be determined as well as an approximation of the deprojected mass density, for use in dark matter simulations, constraining cosmological models [7–9] and testing theories of modified gravity [10]. Redshift measurements of the source galaxy can also aid the study of the expanding Universe, and moreover, help to constrain the nature of dark energy using gravitational time delays to measure the Hubble constant [11,12].

To date, there are only a few hundred detected examples of massive single galaxies lensing light from distant galaxies. Such systems are known as strong galaxy-galaxy lenses [13]. One survey designed to detect gravitational lenses is the Sloan Lens ACS (SLACS) survey [14], which uses the spectroscopic data from the Sloan Digital Sky Survey [15] to identify potential gravitational lenses, which are then confirmed by images from the Hubble Space Telescope. Strong galaxy-galaxy lens systems have also been found in other surveys, including the Dark Energy Survey [16], which is currently aiming to observe 300 million galaxies, by covering 5000 square degrees of the sky over five years (2013–2018).

Telescopes currently in construction are to begin operation in the near future, with resulting surveys expected to generate many thousands of strong galaxy-galaxy lens systems. For example, the ground-based Large Synoptic Survey Telescope [17] due to become operational in 2019 will conduct a ten year survey mapping billions of stars and galaxies, producing around 30 terabytes of data each night. Meanwhile, the European Space Agency's Euclid telescope [18] is due to be launched in 2020. Euclid's aim is to study dark matter and dark energy by measuring the acceleration (and hence expansion history) of the

* e-mail: tomrobinson@hotmail.co.uk

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Universe out to a redshift of $z = 2$. It will cover 15000 square degrees (approximately one third of the sky) over its planned six year mission, producing images of around 2 billion galaxies, and detecting an anticipated 10000 strong lensing systems.

However, until recently, finding strong lenses involved painstaking manual inspection of every image, requiring large groups of people and taking long periods of time to complete. As a result, the development of an autonomous gravitational lens detection method has been necessary to cope with the vast catalogue of images set to be produced by these future surveys. Several automated methods have already been proposed, using geometrical quantification of the lensed images [19,20] along with machine learning techniques [21,22].

Machine learning has become a widespread technique in developing sophisticated computer algorithms, able to analyse large amounts of data far more rapidly than humans. It has a wealth of applications, including face and object recognition, online adaptive advertising, speech recognition, search engines and medical diagnosis [23].

Other methods involve analysing images taken in multiple colour bands, using differences in the relative intensities to distinguish between galaxy lenses and gravitational arcs [13,24]. Another technique incorporated machine learning and spectroscopic analysis in order to distinguish between the light from the source and lensing galaxies [25]. Likewise, analysis of galaxy spectroscopy using another form of machine learning has been separately used to classify a range of astronomical objects, not just strong galaxy-galaxy lens systems [26].

In this paper we summarise our method for the automatic detection of gravitational lenses, which involves the use of machine learning, specifically through the creation of a convolutional neural network (CNN). CNNs in particular are mainly used in face and object recognition, by applying convolutional matrices (filters) to input images in a similar manner to image processing techniques [27]. The program we have created classifies each input image containing galaxies into two categories: those that contain lenses and those that do not. We have trained and tested our CNN on a set of simulated images, and through controlling simulation parameters the variability in the accuracy of the CNN could then be ascertained. Such a method does not require the inspection of spectroscopic data for every object or the morphological object classification used in the papers previously mentioned, and a pre-trained CNN would be able to classify thousands of images extremely quickly.

Since starting this project, multiple papers have been published on the use of CNNs in strong gravitational lens detection. The first method [28] was trained to identify Luminous Red Galaxy lenses with Einstein radii $\geq$1.4 arcsec and was applied specifically to the kilo degree survey [29] for lensing galaxy redshifts of $z \leq 0.4$. While their method is similar, our project benefits in being more general, as it aims to identify lenses regardless of lensing galaxy type and at much greater redshifts, as well as for lenses without any constraint on the Einstein radii.

Another separate paper was published soon after this [30], which detailed the use of deep residual networks (a new advanced version of CNNs) rather than a standard CNN for strong galaxy lens finding, in a method called CMU DeepLens. They found the method was easier to train, using 20000 simulated LSST-like images, and achieved a high degree of accuracy. However, they also observed that their method did not perform significantly better than that in the previous paper, despite their own being notably more complex, due to the limitations in their simulations.

In Jacobs et al. (2017) [31], four separate CNNs were trained using catalogues produced using two different methods, and they showed promising results, with all networks identifying over 90% accuracy. Multiple CNNs have been tested by another group [32], in which a classic CNN was compared against other architectures. They too achieved high accuracy in all architectures, highlighting how complex models were not necessary to achieve the result. However, both papers noted that such high accuracies were likely due to a lack of challenging complexity in their simulations.

CNNs have also been used to analyse images of confirmed lenses, estimating the lensing parameters not only accurately, but much quicker than previous methods. Hezaveh et al. (2017) [33] have created a network capable of obtaining parameters around ten million times as fast, although currently they apply to only a specific, simple density profile. Despite this, it is clear that CNNs can significantly reduce the time taken to perform such tasks with no notable increase in uncertainty, as detailed in Levasseur et al. (2017) [34].

2 Methods

In gravitational lensing theory (see [35]), the mapping between the source and image planes is provided by the lens equation

$$y = x - \alpha(x), \tag{1}$$

where x and y represent positions on the lens plane and source plane respectively, and α is the deflection angle, which is dependent on the lens mass distribution.

For our simulations, we used the isothermal ellipsoid model widely used in gravitational lensing [36], obtained from Keeton (2001) [37], which models a galaxy mass distribution with an outer flat rotation curve. The deflection angle $\alpha = (\alpha_1, \alpha_2)$ for this model is given by

$$\alpha_1 = \frac{bq}{\sqrt{1-q^2}} \tan^{-1} \frac{\sqrt{1-q^2}x_1}{\psi + q^2 s} \tag{2}$$

$$\alpha_2 = \frac{bq}{\sqrt{1-q^2}} \tanh^{-1} \frac{\sqrt{1-q^2}x_2}{\psi + q^2 s}, \tag{3}$$

where $\psi^2 = q^2(s^2 + x_1^2) + x_2^2$, q is the ratio between the semi-minor and semi-major axes, b is a normalisation factor related to the Einstein radius and s is the core radius, set to

zero here. As this project focuses on single galaxies as lenses, it would be extremely unlikely for light from a source to be lensed by multiple galaxies along a telescope's line of sight. As such, the lens plane was chosen to consist of a single galaxy, utilising the aforementioned model.

2.1 Convolutional neural networks

A neural network is a computational model inspired by information processing in biological neural networks, such as in the human brain [38,39]. They consist of neurons, which are basic units of computation. Generally, a neural network consists of layers of neurons, where every neuron in one layer is connected to every neuron in the neighbouring layers, with information travelling forward between them when classifying given inputs (information only travels backwards during training). Outputs are computed using the weights associated with each given input and some non-linear function f, called the activation function.

The activation function is used to introduce non-linearity into the output of a neuron, as to match real-world data. The functions most used throughout this project are the rectified linear unit (ReLU) and the softmax function (see [40] for more implementation details).

Convolutional neural networks are similar to neural networks, the main differences being that the inputs have have a grid-like topology, such as images, and that each neuron in one layer is not necessarily connected to every neuron in the neighbouring layer.

Every CNN contains 4 basic stages, a convolution stage, to extract features from the input image, creating a feature map later used to classify objects in the image. A non-linearity stage to introduce non-linearity into the CNN. A pooling stage which reduces the dimensions of each feature map whilst retaining the most important information, thus reducing the number of computations in the network. Finally a classification stage, which takes the form of a neural network. The final outputs are passed through the softmax activation function to obtain a probability that an object belongs to a certain class. These stages can all be repeated multiple times within the CNN in different orders, provided it starts with a convolutional stage and ends with the softmax activation function. For example, the structure of our CNN is shown in Figure 1. The final classification layer holds the class labels and the type of error function used in training.

The architecture of our CNN was determined by starting with the most basic design, with one of each type of stage. Then trial and error (i.e., adding and removing stages, as well as changing the properties of each stage, such as the number of epochs or convolutions) was used to determine the structure that gave both the optimal training time and accuracy.

After the architecture of the CNN has been determined, it can then be trained to detect and classify certain objects in images. This is done using back propagation, an iterative process of minimising an error function, with adjustments to the weights between neurons being made in a sequence of steps. This uses a set of input data which has a known output; the training set. In neural computing literature

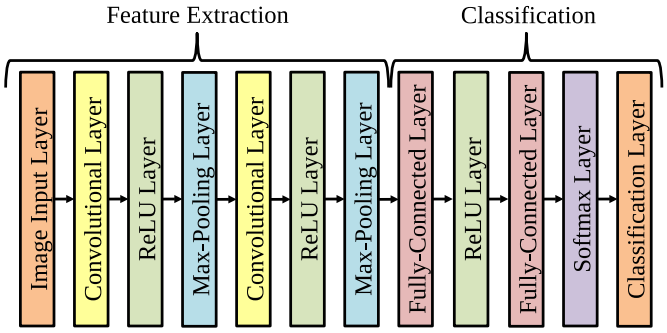

Fig. 1. Structure of our CNN LensFinder. Each max-pooling layer downsamples the images, allowing the next convolutional layer to identify more abstract features in the resulting lower-resolution images. The features extracted are then put through two neighbouring fully connected layers (multi-layered perceptrons) and a prediction is made on how likely the image belongs to a specific class.

back propagation can mean a variety of different things, but throughout this report it shall refer to the training of a CNN using the stochastic gradient descent algorithm applied to an error function [41]. The error function used in our CNN was the cross entropy function for mutually exclusive classes, often paired with the softmax activation function [40].

A disadvantage of using a CNN is that it can only be used on images which are the same size as those on which it was trained. This can be improved upon by using transfer learning [42] from a pre-trained CNN to a regional convolutional neural network (R-CNN) [43]. An R-CNN is a MATLAB detection framework that uses a trained CNN to classify regions within an image. The R-CNN only processes regions likely to contain an object, instead of using a sliding window to process every single region, thus reducing the computational cost. An advantage of using an R-CNN is that it puts a bounding box around positive classifications and gives a probability that it is a specific object class. Not only is this labelling convenient for the user, R-CNNs have the additional advantages of being applicable to wide-field images and reducing the false positive rate (FPR) of the CNN. For further information on CNNs and the training of them see [40].

2.2 Simulation process

We generated two sets of simulated images with 210 000 in each, varying signal-to-noise and morphological parameters. The CNN was trained on one group and then tested on the other. This trained CNN was then re-trained as an R-CNN using both simulated and real images.

To begin, images of galaxies were simulated, with half containing gravitational arcs. An image size of 56×56 pixels was deemed appropriate, based on two criteria: the increase in training time for higher resolution images, and the diameters of expected Einstein rings determined from the Euclid telescope's resolution (0.1 arcsec per pixel).

The light profiles of the lenses were modelled as elliptical galaxies described by a Sérsic surface brightness profile,

Table 1. Parameters varied in the simulation process.

Parameter	Range
Source galaxy redshift	1.1–3.1
Lensing galaxy redshift	1–2
Source offset from lens centre (pixels)	0–4
R_{Ein}/R_e	0.1–1.0
I_{0l}/I_{0s}	1.5–6
Signal-to-noise ratio	10–500

$$I(R) = I_0 \exp\left[-b_n\left(\frac{R}{R_e}\right)^{\frac{1}{n}}\right], \quad (4)$$

where I_0 is the peak brightness. R_e is the effective radius which encloses half the total light of the galaxy, $n = 4$ is the Sérsic index and $b_n = 7.669$ is a constant that describes the shape of the light profile.

Several parameters were used to control the appearance of the images, notably the position offset (in pixels) of the source peak intensity from the lens centre, the ratio of the lens Einstein radius to the galaxy's effective radius (R_{Ein}/R_e), the ratio of the lens intensity to source intensity (I_{0l}/I_{0s}), and the lens and source redshifts. These were utilised to simulate the lensing effects and the foreground (lens) and background (source) galaxies. By allowing all of these values to randomly vary with a uniform distribution between certain limits, shown in Table 1, a wide range of images was obtained. Maintaining 1–5 additional sources in all the training images avoided overcrowding and replicated the galaxy density in typical lens snapshots.

The apparent luminosities, angular sizes and distances to the galaxies were calculated based on their redshifts, which in turn affected the clarity and size of the Einstein ring. The two-dimensional pixelated lens plane was created using equations (1)–(3), by lensing the source plane galaxy surface brightness. Both source and lensing galaxy brightness profiles were created using equation (4), and as galaxies are unlikely to be spherically symmetric, both radius R and effective radius R_e depend on the angle from the semi-major axis. The foreground galaxy was also allowed to vary in position and rotation, both of which affecting the appearance of the Einstein ring.

For elliptical galaxies, the Faber-Jackson relation describes how the luminosity scales with velocity dispersion [44]. Following this, the mass is approximated as being almost directly proportional to the luminosity, so this was applied to the peak intensity of the lensing galaxy, and this combined with I_{0l}/I_{0s} determined the source peak intensity. Additional foreground galaxies were also added that did not produce lensing effects, in order to make the images more realistic.

For the most part the simulated galaxies were generated with RGB colour. Determining the colour of the galaxies was based on the proportion of red ellipticals to blue spiral galaxies in a given region, which is related to the redshift at which they are observed. Buitrago et al. (2013) [45] show an approximately linear relation between the number of spiral galaxies and redshift up to $z = 2$, with the number of ellipticals decreasing linearly as a result. This was employed in our program, in which each galaxy was assigned to one of two groups, ellipticals or spirals. The likelihood of being assigned to a given group was determined using a random distribution related to this linear relation. Colours were assigned to each group, with ellipticals assigned colours from red to yellow, and spirals assigned variations around blue. The variability is used to address the differences in galaxy colours, as well as account for other potential galaxies (e.g. lenticulars) and the effects of redshifted light. These colours were not expected to truly accurately match real observations, but were included to provide an indication of how well the CNN would cope with classifying colour images.

To account for the background noise seen in real images, the simulations included the addition of Gaussian noise distribution, which allowed for the control of the signal-to-noise ratio (SNR). As the program aimed to detect lensing arcs, their collective intensities acted as the "signal". By specifying the SNR,

$$SNR = \frac{S}{\sigma\sqrt{N}}, \quad (5)$$

where N is the number of pixels whose intensities were summed to produce the signal S, the Gaussian noise was added with the standard deviation σ.

For efficiency, each image was produced twice, one with and one without lensing effects. 210 000 of each "Lens" and "NoLens" images were created, used for both training and testing. Some examples of the images produced are shown in Figure 2, along with some real images of strong galaxy-galaxy lens systems for comparison.

2.3 CNN construction and training

Our script was written in MATLAB [46], and utilised the deep learning toolbox. The structure of the final CNN, to which we have attributed the name LensFinder, is shown in Figure 1. It combines two convolutional and max-pooling layers, two fully connected layers, a softmax layer and finally a classification layer. In addition to this are three ReLU Layers, which add non-linearity to the data as necessary. The training process itself was optimised by varying certain training options. The learning rate was maintained at a low value of 0.001 to ensure accurate training (although increasing training duration), however the number of training iterations (epochs) was kept minimal to reduce the total training time, with a maximum of 20 epochs allowed.

LensFinder was trained with a randomised 210 000-image set (retaining equal numbers of lenses and non-lenses) drawn from the 420 000 image set. Simulations were varied using the four control parameters (SNR, position offset of the source peak intensity, R_{Ein}/R_e and I_{0l}/I_{0s}). The trained CNN was saved, and tested with the remaining 210 000 images. Producing the image catalogue took around two hours, and training times for the CNN ranged

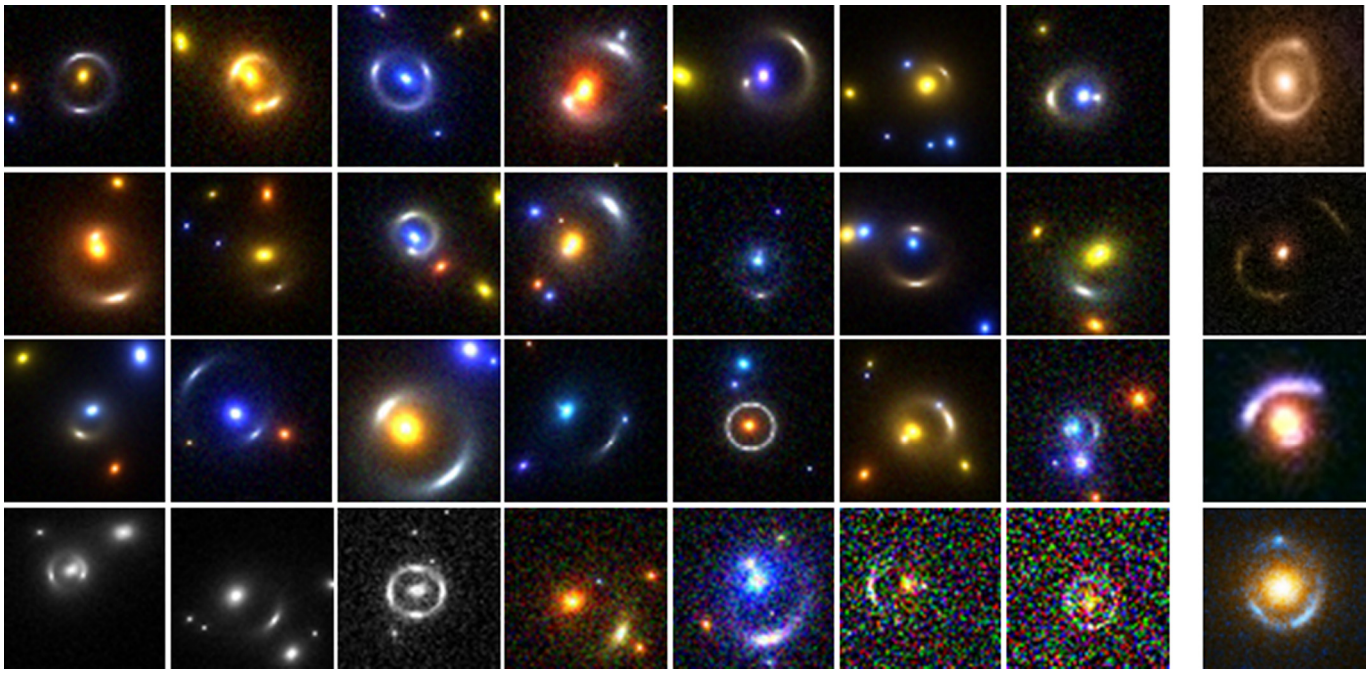

Fig. 2. Left: sample of simulated 56×56 pixel images of strong galaxy-galaxy lenses. Along the bottom row, the first three show examples of greyscale images produced compared to the other RGB images, while the others are examples of images with decreasing signal-to-noise ratios. Right: real images of strong galaxy-galaxy lenses. Image Courtesies from top to bottom: NASA, ESA, C. Faure (Zentrum für Astronomie, University of Heidelberg) and J.P. Kneib (Laboratoire d'Astrophysique de Marseille) (top two images); Kavan Ratnatunga (Carnegie Mellon Univ.) and NASA/ESA; and NASA, ESA, A. Bolton (Harvard-Smithsonian CfA) and the SLACS Team.

from 30 min to several hours. However, the testing of image catalogues was much faster, consistently taking less than 10 min to classify the entire catalogue. The test images were passed through the CNN to gauge the accuracy and the results were then recorded.

3 Results

Accuracy is defined as the ratio of correct classifications the CNN makes to the total number of images tested against it. For the CNN layer structure we have designed (see Fig. 1), an accuracy of $98.12 \pm 0.26\%$ can be expected from any newly trained CNN. However, as the 420 000 images are randomly selected during each iteration, the accuracy of any individual CNN will be different from the mean average accuracy of the CNN layer structure. The accuracy of the trained CNN we present here (LensFinder) is $98.19 \pm 0.12\%$. Errors are estimated by iterating the entire training and testing process seven times. The accuracies are calculated via the mean of these iterations and the error from the first standard deviation of the mean result.

Accuracy is a competent method for considering the overall capabilities of a CNN, however tells us nothing about how confident a prediction may be. The prediction threshold for a classification is 0.5, thus predictions around this value have a very low certainty in being accurate. To understand the true capabilities of LensFinder, the decimal probabilities of an image containing or not containing a lens are shown in Figure 3.

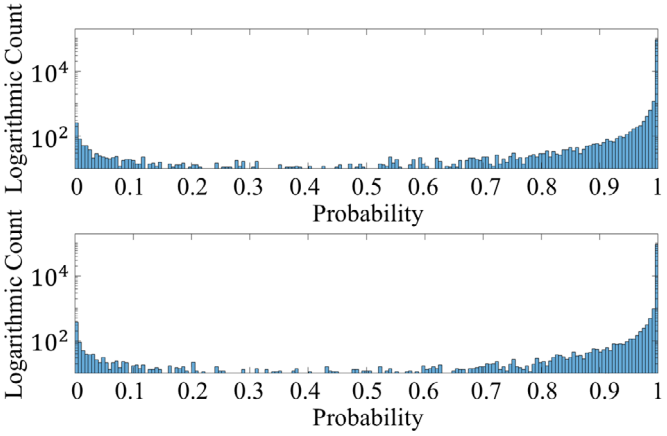

Fig. 3. Histograms representing prediction data from the CNN. Top: decimal probability of an image containing a lens. Histogram represents 105 000 images which do contain lenses. Bottom: decimal probability of an image not containing a lens. Histogram represents 105 000 images which do not contain lenses.

We know that LensFinder predicts the classification of images with an accuracy of 98.19%, thus 98.19% of the classifications are above the 0.5 threshold. At the edges of each plot, there is a clear data spike corresponding to at least 100 000 correctly predicted images (95%), with certainties $\geq$0.986. Evidently the majority of classifications are based on highly certain predictions. Data spikes at the incorrect edges of the plot represent 0.3% of data, thus may be attributed to anomalous cases. The histograms show

incredibly similar trends, suggesting that LensFinder is equally capable of identifying images with and without lenses. Both histograms have strong peaks, followed by a rapid tail-off, confirming that predictions are dominated by confident probabilities.

Early tests showed that a CNN trained and tested against catalogues containing fewer than 100 000 images gave significantly lower accuracies. We concluded that the optimal image catalogue must contain approximately 100 000–500 000 images to maximise CNN accuracy, while maintaining time efficiency. During the development of LensFinder we tested two methods of training a CNN. Training method 1 (TM1) trained the CNN using a collection of thirteen 10000 image catalogues. In each 10000 image catalogue, one of the four control parameters were given a specific value, while the other three were randomised uniformly between two chosen values. This method trained the CNN heavily for extreme examples of lensing. Training method 2 (TM2) trained the CNN using a catalogue of images simulated by smoothly varying all four control parameters together. These parameters were all randomised uniformly, creating a catalogue of 130 000 images with non-equal parameters. TM2 therefore trained the CNN against a wide number of lensing scenarios.

Ultimately, LensFinder was trained using an amalgamation of the two methods. The two 210 000 image catalogues which trained and tested LensFinder each consisted of 160 000 images simulated using the smooth variation method of TM2. Alongside this were five of the 10000-image catalogues used in TM1. The TM1 catalogues chosen were those which gave the poorest results, such as images with low SNR and faint source galaxies compared to those in the foreground. LensFinder produced accuracies in excess of its original 98.19% accuracy for all but one of the thirteen TM1 test catalogues. Source offset position and SNR remained constant, regardless of parameterisation. Despite the training catalogue attempting to prepare LensFinder for extreme cases, e.g. significantly low SNRs, there is still a noticeable drop in accuracy when tested against such images. For SNR, intensity ratio and Einstein radius, this is unsurprising, as the images are either noisy, dusty or faint. Accuracy drops for both high and low values of source offset; we hypothesise that this is because the central positions produce a wide variety of features, rather then many examples of the same type, which better suits LensFinder.

Another method used to determine the effectiveness of LensFinder was the creation of a receiver operating characteristic (ROC) curve. This consisted of LensFinder's true positive rate (TPR),

$$TPR = \frac{N_{\text{true positives}}}{N_{\text{true positives}} + N_{\text{false negatives}}} \quad (6)$$

(the proportion of correct lens classifications compared to the total number of lenses in the test catalogue), plotted against its FPR,

$$FPR = \frac{N_{\text{false positives}}}{N_{\text{false positives}} + N_{\text{true negatives}}} \quad (7)$$

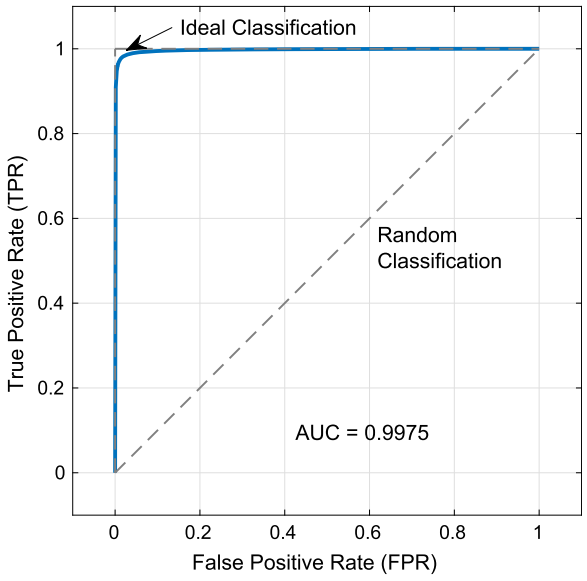

Fig. 4. ROC curve for the LensFinder CNN. An ideal result would be to have $TPR = 1$ for $FPR = 0$, while a result gained from random classification would produce a 45 degree line. Clearly LensFinder is much closer to the former.

(the proportion of non-lenses misclassified as lenses compared to the total number of non-lenses). The result can be seen in Figure 4, which again shows the high accuracy of LensFinder. This can be measured using the area under the curve (AUC), which ranges from 0.5 (random classification) to 1.0 (ideal classification), and LensFinder was found to have an AUC of 0.9975. Figure 5 shows the logarithmic version of the ROC curve, and as can be seen, for a FPR of 1% LensFinder achieved a TPR of 0.971, which is also a promising result.

The final output from the network was labelling from the R-CNN (R-LensFinder). For this, a set of 100 images were selected, both real and simulated, and bounding boxes were manually placed around any lenses in the images. R-LensFinder used this set to provide further training in recognising lenses in wider-field and real images. The R-CNN had a perfect labelling rate of 80% (correctly labelling the lenses in the images), with an additional 10% partially correct (correctly labelling the position of the lenses, but then either labelling additional sources incorrectly, or the bounding box is too large or small). As for the remaining images, there was a false negative rate of 9% as a result of R-LensFinder not labelling any part of the lensed images. Only one image was labelled completely incorrectly, and thus, R-LensFinder has a FPR of 1%. R-LensFinder is clearly able to identify and label the location of lenses in the images.

The final images tested on R-LensFinder were eight real examples of astronomical features taken by the ESO and ESA. The images included not only examples of real gravitational lenses, but also images which contained no lenses, such as nebulae. An example of these tests are shown in Figure 6. Initially, no training was provided against real examples of gravitational lenses, nor negative training against astronomical objects such as stars and spiral galaxies (training with no bounding boxes around them).

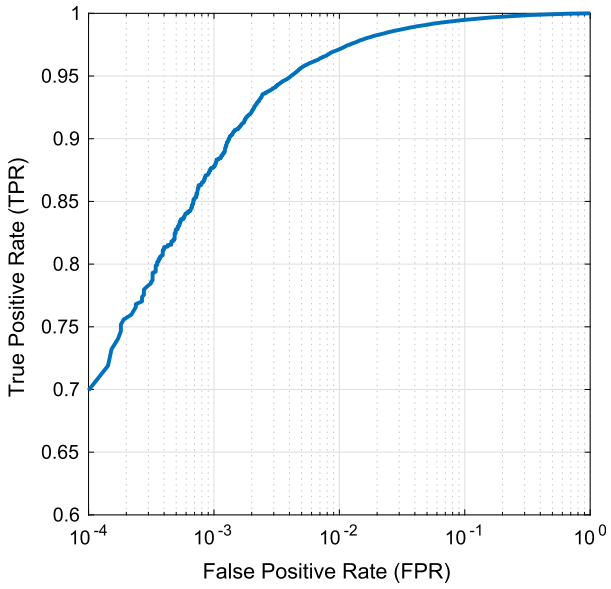

Fig. 5. Logarithmic ROC curve for the LensFinder CNN. This indicates that for a FPR of 1%, a TPR of 97% is achieved.

The top image in Figure 6 shows the labelling provided by R-LensFinder for the initial basic training set. False labelling and incorrectly-sized boxes were the most prominent issues across all images tested. To improve this result, we added several real images to the training set, including both positive training for natural lenses, and negative training against stars with large diffraction spikes, spiral galaxies and nebulae, and this solved many of the issues across the majority of the test images. Figure 6 shows the remarkable capability of R-LensFinder to identify and label examples of gravitational lensing from snapshot images alone. Despite this success however, there are some evident limitations in the capabilities of R-LensFinder. For many of the images tested, there was still a significant proportion of falsely labelled images (generally caused by the colour variations from diffraction spikes around stars, or arc-like dust features in spiral galaxies).

4 Discussion

LensFinder was developed to accept any image catalogue then classify the images automatically. LensFinder has succeeded in doing so to a notably high accuracy. Despite this, any CNN is only as good as the images which it is trained against, therefore the capabilities of LensFinder must be weighed against the quality of the training catalogue. Overall, we believe the simulated images are sufficiently comparable to real data. Figure 2 indicates a striking resemblance between our simulated lenses and real observational data.

CNN classifications are based on shape recognition, so using observationally-motivated surface brightness profiles for the simulations was essential. Minor factors also add to the realism of the final images: The angular resolution of surveys such as Euclid will typically produce lens images with ring radii of 10–20 pixels. By choosing to train

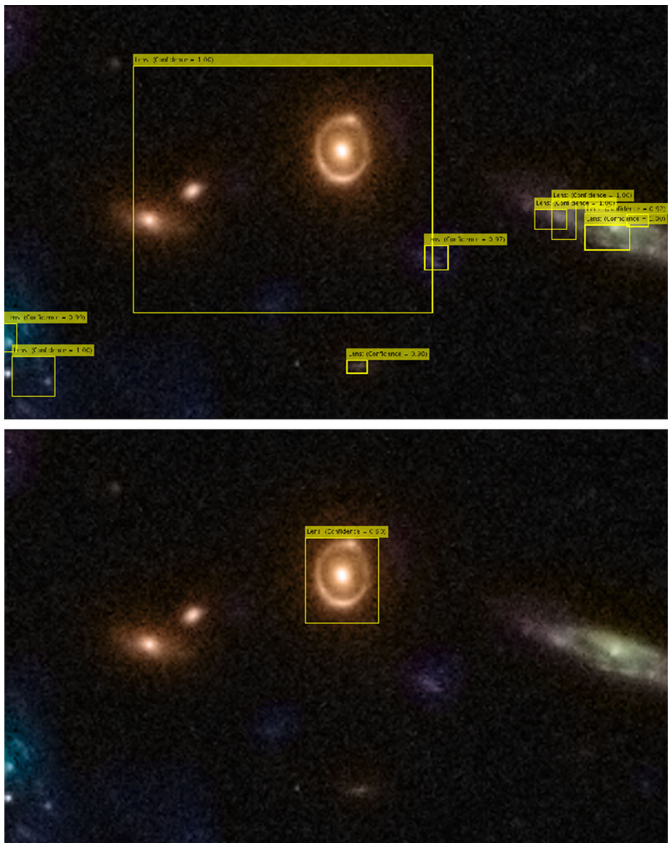

Fig. 6. Images showing the effect of negative training on an R-CNN. In this comparison we are considering only the sizes and positions of the bounding boxes, as the R-CNN has classified them all as lenses with confidences ≥90%. The top image has had no negative training, while the bottom image has. Negative training has had a significant positive effect as all false positives have been removed, and the correct label box has been cropped significantly. Credit to ESA for the original image.

LensFinder against 56×56 pixel snapshots the scale of features in the images fit well with real data, which in turn better prepares LensFinder to classify real images.

By simulating light sources which can vary in position, brightness, size and orientation, a more natural snapshot can be created. Some detectable lens features appear to be very similar to point-like foreground galaxies, therefore it is crucial to differentiate between a lens feature and a foreground light source. Despite our efforts, certain astronomical objects, such as spiral galaxies, appear very similar to the lens features which are produced by our simulations. It is therefore possible that LensFinder would falsely identify structured objects, such as spiral galaxies, nebulae, proto-planetary disks or even stars, as lenses. This problem is not unique to our CNN, as others have also had false positives as a result of such objects in real images [28,30]. The solution to this problem would be to negatively train LensFinder against these features by adding examples of such astronomical objects, either natural or simulated, to the "NoLens" training catalogue. Figure 6 is evidence to this, as adding examples of other astronomical objects has clearly had a positive effect on the final results.

Currently it is uncertain how LensFinder would adapt to real structured objects, despite being well trained against other foreground sources.

It should be noted that despite recent developments in machine learning architecture, significant progress can be made in creating more complex simulations. The CMU DeepLens method published earlier this year [30] also found issue in their simulations. While their method was easy to train, and they achieved a 90% TPR for Einstein radii larger than 1.4″ and for a 1% FPR, they state that the results are "optimistic estimates" due to their limited complexity in simulating images. Because of this, they were unable to show a significant improvement in detections compared to simpler methods, and that their model would only be able to outperform them once more complex simulations have been created. In Schaefer et al. (2017) [32], a high degree of accuracy was achieved in all four of their CNN architectures, reaching AUCs of 97.7% and 94.0% for ground-based and space-based data respectively. They again relay that this was likely due to their simulations lacking the wider variety of astronomical objects that would potentially cause false positives. A similar case emerged in Jacobs et al. (2017) [31], where the simulated images used were produced using one of two methods. Of the four CNNs used, three produced accuracies of over 98% when both trained and tested images were produced by the same method, yet they performed poorer when tested on images produced using the method not used in training. This indicated that CNNs identify features specific to the simulated training catalogue, which may bias their results.

The random nature of our simulations has produced a significant number of unrealistic simulations. No full Einstein ring has ever been imaged, yet ∼15% of the training catalogue contains full, and very bright Einstein rings. This ratio is clearly far from realistic and therefore could positively skew the data as the images are too basic. It is likely however, that retaining a small portion of basic images helps to maximize training efficiency.

The colour-magnitude of galaxies is simulated randomly and independently for all sources in the images, generally in a red or blue hue (as in real data). However, some anomalous sources produce unnaturally coloured galaxies which may be vivid shades of purple, yellow or green. Our colours are simulated using a three band filter (RGB), however, real surveys may use other combinations of filters to generate colour depending on the wavelength of the data. While this appears to be a flaw in the catalogue, the CNN training is dependent on the colour difference between the sources in the images. We expect LensFinder to detect features regardless of their colour, as it is already well trained against many colour variations. Despite this, the analysis of image colours can significantly improve the detection process [13,24,25,28], due to the expected colours of galaxies having a redshift dependence, and as such this would present an avenue to explore for further research in this project.

Additional circumstances which LensFinder has no training against include source spiral galaxies and multiple lens sources. The mass distribution and surface brightness models for spirals are very different to those of ellipticals, therefore the shape and brightness of lens features would also vary. In addition, our simulations account for only one source galaxy, assuming all other sources are foreground galaxies. In many cases, light from other background sources are also affected by the lens galaxy. As our simulations do not account for either possibility, it is unknown how LensFinder would adapt to these types of unseen images.

With accuracies consistently in excess of 97%, LensFinder not only makes very accurate predictions about our simulated images, but highly confident predictions as well. LensFinder has a false positive/negative rate of ∼1% each. The FPR is slightly higher than the false negative rate, however, this is potentially better for the purpose of LensFinder. Galaxy-galaxy lenses are rare, therefore it is better to detect all potential lenses, than confidently rule them out. The R-CNN's purpose is to remove the majority of the false positives, by labelling features directly in images. With a FPR of only 1% R-LensFinder has succeeded in doing so.

A disadvantage of our method is the requirement for a GPU and large amounts of computing power. The supervised classification pipeline using logistic regression and HOG edge detection [22] published earlier this year had the advantage of working without these, however their simulations once again presented an issue; notably their source galaxies had a fixed redshift of $z = 2$, and similar to our method they used 50% lens and 50% non-lens images in training, which is unrealistic. The alternative method involving PCA [21] achieved a completeness of 90% for $SNR \geq 30$, and performed well when tested on real data. While ours gives a higher accuracy, we have only tested LensFinder on our less complex simulations, and so may perform poorer when tested on real observations.

Fully training a single version of LensFinder takes considerable time, however the comparative time taken to test a new image catalogue against a pre-trained version of LensFinder justifies the use of a CNN. Generally, 210 000-image catalogues are tested completely in under 10 min, hence LensFinder is both faster and more accurate than human inspection. For many images in the test catalogue, human inspection alone was insufficient to definitively categorise an image. We approximate that 5% of the catalogue contained images with lens features unidentifiable to the human eye, such as due to low SNR or from lensing galaxy light obscuring the lensing arcs. As LensFinder's false negative rate is far lower than this, clearly LensFinder is more capable of classification than human inspection alone. We attribute this success to the catalogue of images which LensFinder is trained against. The variations in the catalogue give a highly diverse mix of images, which allows LensFinder to be capable with obvious lens features such as Einstein rings, but also provides a wealth of training against small features such as point-like duplications. LensFinder is also adapted to cope with extremely noisy images and faint sources.

While parameters were controlled in the development of the CNN, due to time constraints we were unable to test how the performances of the final CNN and R-CNN changed as a function of SNR, source intensity and other parameters. This would have allowed us to discern the strengths and weaknesses, and hence refine, our design.

R-LensFinder has clearly not matched the capabilities of LensFinder, however a perfect labelling rate of 80% remains an impressive result. The false positive/negative rate is higher than expected, likely caused by a lack of training, particularly against faint lenses. Generally however, R-LensFinder has achieved its purpose, to aid in the detection of lens features, and reduce the FPR of LensFinder. Figure 6 represents an important result, proving that an R-CNN can easily adapt to real data. Despite this success, R-LensFinder requires far more training to ensure that all of the false labels are removed. In particular, R-LensFinder needs to be exposed to fainter Einstein rings, and images with large variations in the colours of features (such as spiral galaxies, or diffraction spikes around stars), as these appear to cause R-LensFinder the most trouble.

Despite LensFinder's accuracy of 98.19% and AUC of 0.9975, this success is dependent upon our images. While our simulated catalogue bares a strong resemblance to real lenses, it is limited in its realism, such as in the colour of the images and that they consist only of ellipticals. This highlights a major hurdle in identify lenses using machine learning, as to achieve significant progress, more complex and realistic simulations are required. R-LensFinder introduces an additional layer of classification to LensFinder, which adds confidence to the capabilities of LensFinder as a whole. R-LensFinder has also laid the groundwork for direct identifications, as real lenses have been correctly identified and labelled automatically.

5 Dead end

Ultimately the structure and training options of LensFinder were determined by trial and improvement. To save training time the initial learn rate was modified, but an error developed that prevented the CNN from training beyond 50%. Hence it was decided that the training options would be kept as MATLAB defaults, and instead a compromise between training time and accuracy was decided based on the CNN layer structure.

Once our CNN design was completed, we also experimented with the composition of images within catalogues, as is described in the results section. Despite initial successes using TM1, we found that CNNs are far better trained when exposed to diverse mixtures of lenses, rather than tailored examples of extreme lenses. We propose that any image catalogue should consist of images simulated by both TM1 and TM2 for a completely trained CNN.

6 Conclusion

We have presented the development of a convolutional neural network, capable of automatically detecting strong galaxy-galaxy lenses from snapshot images. This method is well suited to analysing the large data sets expected to be produced by galaxy surveys in the near future, and ultimately detects lenses far more quickly and accurately than previous methods. Our code, LensFinder, was written in MATLAB. LensFinder was trained and tested using a catalogue containing a total of 420 000 simulated images. Repeated training using this catalogue gave a mean accuracy of $98.12 \pm 0.26\%$, with the final CNN having an accuracy of $98.19 \pm 0.12\%$. Prediction data showed that 95% of the images were correctly classified (as either containing a lens or not) with at least 98.6% certainty, indicating that the CNN has been well-trained. This was supported by LensFinder achieving an AUC of 0.9975 and a TPR of 0.97 for a FPR of 1%.

An R-CNN was also used to reduce the FPR of the CNN. Testing indicated that R-LensFinder labels images correctly up to 90% of the time. Furthermore, R-LensFinder has proved capable of correctly identifying and labelling real gravitational lenses. Overall, LensFinder has largely been a success, producing highly accurate classifications of galaxy images in a comparatively short time. There are certainly developments to be made, including increasing image variability in the training catalogue. Adapting LensFinder to identify more types of lenses, and to do so in greyscale, would also make LensFinder appropriate for a wider range of applications.

References

1. J.-P. Kneib, et al., ApJ **607**, 697 (2004)
2. J. Richard, et al., MNRAS **413**, 643 (2011)
3. S. Dye, et al., MNRAS **388**, 384 (2008)
4. L.V.E. Koopmans, et al., Astro2010: the astronomy and astrophysics decadal survey, science white papers, no. 159 (2009)
5. N. Hashim, et al., arXiv:1407.0379 (2014)
6. E. Louis Strigari, Phys Rep **531**, 1–88 (2013)
7. J. Schwab, et al., ApJ **708**, 750 (2009)
8. G. Jiang, C.S. Kochanek, ApJ **671**, 1568 (2007)
9. C. Ma, T.-J. Zhang, ApJ **730**, 74 (2011)
10. J. Enander, E. Mörtsell, JHEP **2013**, 31 (2013)
11. V. Bonvin, et al., MNRAS **465**, 4914 (2017)
12. G. Adam Riess, et al., ApJ **826**, 56 (2016)
13. M. Maturi, et al., A&A **567**, A111 (2014)
14. S. Adam Bolton, et al., ApJ **638**, 703 (2006)
15. B. Abolfathi, et al., arXiv:1707.09322 (2017)
16. Dark energy survey collaboration and others, arXiv:astro-ph/0510346 (2005)
17. Z. Ivezic et al., AAS Bull. Am. Astron. Soc. **41**, 366 (2008)
18. R. Laureijs et al., 2011, Reference: ESA/SRE(2011)12
19. C.R. Bom, et al., A&A **597**, 13 (2017)
20. G. Seidel, M. Bartelmann, A&A **472**, 341 (2007)
21. R. Joseph, et al., A&A **566**, A63 (2014)
22. C. Avestruz, et al., arXiv:1704.02322 (2017)
23. M. Mohammed, et al., *Machine learning: algorithms and applications* (CRC Press, Boca Raton, Florida, 2016)
24. R. Gavazzi, et al., ApJ **785**, 144 (2014)
25. F. Ostrovski, et al., MNRAS **465**, 4325 (2017)
26. D. Baron, D. Poznanski, MNRAS **465**, 4530 (2016)
27. E.P. Alessandro Villa, et al., Articial neural networks and machine learning–ICANN 2016: Proceedings of 25th International Conference on Articial Neural Networks, volume 9887, Springer, Barcelona, Spain, 2016
28. C.E. Petrillo, et al., MNRAS **472**, 1129–1150 (2017)
29. T.A. Jelte de Jong, et al., Exp. Astron. **35**, 25 (2013)

30. F. Lanusse, et al., MNRAS, arXiv:1703.02642 (2017)
31. C. Jacobs, et al., MNRAS **471**, 167 (2017)
32. C. Schaefer, et al., A&A 9 (2017)
33. D. Yashar Hezaveh, et al., Nature **548**, 555557 (2017)
34. L.P. Levasseur, et al., ApJL **850** (2017)
35. S. Mollerach, E. Roulet, *Gravitational lensing and microlensing* (World Scientific, Singapore, 2002)
36. A. Kassiola, I. Kovner, ApJ **417**, 450 (1993)
37. C.H. Keeton, arXiv:astro-ph/0102341 (2001)
38. F. Rosenblatt, *Principles of neurodynamics: perceptrons and the theory of brain mechanisms* (Spartan Books, Washington, DC, 1962)
39. E. David Rumelhart et al., *Parallel distributed processing, explorations in the microstructure of cognition: foundations*, volume 1 (MIT Press, Cambridge, Massachusetts, 1986)
40. I. Goodfellow, Y. Bengio, A. Courville, *Deep learning* (MIT Press, Cambridge, Massachusetts, 2016), http://www.deeplearningbook.org
41. C.M. Bishop, *Pattern recognition and machine learning* (Springer, New York, 2006)
42. S.J. Pan, Q. Yang, IEEE TKDE **22**, 1345 (2010)
43. R. Girshick et al., Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14, IEEE Computer Society, 2014, 580 p.
44. R.H. Sanders, MNRAS **407**, 1128 (2010)
45. F. Buitrago, et al., MNRAS **428**, 1460 (2013)
46. MATLAB. version 9.0 (R2016a), The MathWorks Inc., Natick, Massachusetts, 2016

Cite this article as: James Pearson, Clara Pennock, Tom Robinson, Auto-detection of strong gravitational lenses using convolutional neural networks, Emergent Scientist **2**, 1 (2018)