



HAL
open science

Deep transfer learning for blended source identification in galaxy survey data

S. Farrens, A. Lacan, A. Guinot, A. Z. Vitorelli

► **To cite this version:**

S. Farrens, A. Lacan, A. Guinot, A. Z. Vitorelli. Deep transfer learning for blended source identification in galaxy survey data. *Astronomy and Astrophysics - A&A*, 2022, 657, pp.A98. 10.1051/0004-6361/202141166 . hal-03533667

HAL Id: hal-03533667

<https://hal.science/hal-03533667>

Submitted on 18 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Deep transfer learning for blended source identification in galaxy survey data[★]

S. Farrens¹, A. Lacan², A. Guinot¹, and A. Z. Vitorelli¹

¹ AIM, CEA, CNRS, Université Paris-Saclay, Université Paris Diderot, Sorbonne Paris Cité, 91191 Gif-sur-Yvette, France
e-mail: samuel.farrens@cea.fr

² Université Paris-Saclay, CNRS, CEA, Astrophysique, Instrumentation et Modélisation de Paris-Saclay, 91191 Gif-sur-Yvette, France

Received 23 April 2021 / Accepted 13 October 2021

ABSTRACT

We present BLENDHUNTER, a proof-of-concept deep-transfer-learning-based approach for the automated and robust identification of blended sources in galaxy survey data. We take the VGG-16 network with pre-trained convolutional layers and train the fully connected layers on parametric models of COSMOS images. We test the efficacy of the transfer learning by taking the weights learned on the parametric models and using them to identify blends in more realistic Canada-France Imaging Survey (CFIS)-like images. We compare the performance of this method to SEP (a Python implementation of SEXTRACTOR) as a function of noise levels and the separation between sources. We find that BLENDHUNTER outperforms SEP by $\sim 15\%$ in terms of classification accuracy for close blends (< 10 pixel separation between sources) regardless of the noise level used for training. Additionally, the method provides consistent results to SEP for distant blends (≥ 10 pixel separation between sources) provided the network is trained on data with noise that has a relatively close standard deviation to that of the target images. The code and data have been made publicly available to ensure the reproducibility of the results.

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

1. Introduction

The blending of sources, that is, the apparent overlapping of extended objects in 2D images, has a significant impact on measurements of the morphological and structural properties of galaxies, in particular for ground-based surveys. Objects in close proximity to one another can easily be mistaken for a single source, which can lead to significant detection and/or measurement biases depending on the depth of the survey.

For weak gravitational lensing analysis, it is essential to understand how and to what degree blending impacts shear and photometric redshift measurements (Mandelbaum 2018). This problem becomes even more critical in the current epoch of high-precision cosmology, where systematic effects need to be carefully accounted for. This is particularly important for reliable comparisons between late-time probes, like weak lensing, and early-time probes, such as the CMB.

Blended sources make up a significant fraction of the observed sources in survey images: $>30\%$ for the Dark Energy Survey (DES, Samuroff et al. 2018), $>50\%$ (up to $>60\%$ and $>70\%$ for the Deep and UltraDeep layers, respectively) for Hyper Suprime-Cam (HSC, Bosch et al. 2018), and $>60\%$ for Vera C. Rubin Observatory Legacy Survey of Space and Time (LSST, Sanchez et al. 2021). Simply removing all the blends that have been identified would significantly reduce the sample size and could potentially lead to entangled biases in the shear correlation function (Hartlap et al. 2011). Additional

biases can be introduced by unidentified blends coming from sources below the detection threshold (Hoekstra et al. 2017; Euclid Collaboration 2019). It is therefore necessary to develop an appropriate procedure for dealing with blends in survey data.

The process of handling blended sources in astrophysical images can be broadly divided into four major problems: (a) the detection of objects in the image, (b) the classification of those objects as either single sources or blended sources, (c) the segmentation of pixels from blended sources (often referred to as ‘deblending’), and (d) the rejection of those objects that cannot be easily included in the scientific analysis. This paper focuses on the problem of classifying objects already detected with standard source extraction software as either blends or single sources. Unlike the problem of segmentation, which has been abundantly addressed in the literature (Joseph et al. 2016; Melchior et al. 2018; Reiman & Göhre 2019), little effort has been made to find reliable and automated methods for identifying blended sources in survey data.

Traditional methods for identifying blended sources, such as SEXTRACTOR (Bertin & Arnouts 1996), rely on fixed thresholds to detect multiple peaks in the light intensity profiles of the objects. While this approach may work for a reasonable fraction of the sources, it lacks the flexibility needed to handle blends with a greater discrepancy in the brightness and/or size between the individual objects. This means that a large number of blended sources, which may have a non-negligible impact on the scientific analysis, could be missed.

Machine-learning techniques, in particular deep-learning architectures, have been shown to be incredibly successful when applied to complicated classification problems (see e.g., Kotsiantis 2007; Lecun et al. 2015; Srinivas et al. 2016).

[★] In the spirit of reproducible research, all code and data needed to reproduce the results in this paper have been made publicly available on GitHub (<https://github.com/CosmoStat/BlendHunter>) without any restrictions.

However, the effectiveness of these tools can be difficult to gauge without reliably labelled data. In real astrophysical images, it is not known a priori if the underlying signal for a given detection comes from a single object or indeed from the combination of several. This necessitates the use of simulated data in order to generate a reliable training set, where the number and diversity of overlapping sources are perfectly known. Nevertheless, this introduces the possibility of over-fitting the network to properties specific to the simulation, i.e., overly simplified galaxy models. This in turn can cause the network to be very sensitive to artefacts and more complex structures seen in real observations. Developing simulated galaxy images that contain all of the properties expected in real data is extremely challenging and time consuming. Therefore, the application of networks well suited to transfer learning represents an interesting approach to mitigate this problem.

Transfer learning, in particular deep transfer learning, is a machine-learning approach whereby network weights obtained by training on a given data set are applied to another distinct but similar data set. This can help to prevent over-fitting to the training data and can significantly reduce the time required to apply the method to new data sets. Deep transfer learning has been used in a variety of problems in astrophysics in recent years, including: the classification of compact star clusters (Wei et al. 2020), the separation of low-surface-brightness galaxies from artefacts (Tanoglidis et al. 2021), the classification of planetary nebulae (Awang Iskandar et al. 2020), and the deblending of galaxy images (Arcelin et al. 2021).

This work introduces a proof-of-concept deep transfer learning approach, called BLENDHUNTER (BH), for the automated and robust identification of blended sources in single-band galaxy survey data. This method incorporates a convolutional neural network (CNN) trained on a large database of natural images and a fully connected layer for classification. Simple parametric models are used to train the fully connected layers and the learned weights are in turn used to identify blended sources in more realistic images.

This paper is organised as follows. The following section introduces the properties of the training and testing data. Section 3 presents the transfer learning approach used and how it was trained to identify blended galaxy images. Section 4 provides results on how this approach compares to the state of the art. Finally, conclusions are presented in Sect. 5.

2. Data

Supervised machine-learning approaches, such as that presented in this work, require accurately labelled and representative training data. These labels often correspond to properties that cannot be directly and/or reliably measured from observed data. Simulations, on the other hand, provide a controlled environment where these properties are known a priori.

To train our network, we opted to produce a set of simulated single-band (i.e., monochromatic) galaxy images from a parametric model. This allowed us to control the fraction of blended images in the whole sample and the separation of sources in the blended images. The choice of single-band images was made to restrict the learning to pixel features (i.e., no colour information), which could potentially be of interest for the r -band of the Canada-France Imaging Survey (CFIS)¹, part of the Ultraviolet Near-Infrared Optical Northern Survey (UNIONS), or eventually the Euclid visible band (Cropper et al. 2012). We addition-

ally prepared a sample of realistic CFIS-like galaxy images to test the efficacy of transferring the learned weights to a similar but unseen data set. This section details how the training and testing data sets were generated.

2.1. Blend definition

In order to label our training data, we first needed to make a clear definition of a blend. Typically, this would be some measure of the amount of overlap between the light profiles of the individual sources that constitute the blend. For the purposes of this work, we assume a simple scenario in which isolated (i.e., unblended) sources are composed of a single galaxy centred within a postage stamp of size 51×51 pixels ($9.5 \text{ arcsec} \times 9.5 \text{ arcsec}$). We then define a blend as a postage stamp (of the same size) containing two galaxies, one at the centre and a secondary source at a random position. This definition is fairly agnostic in that we do not require any specific overlap between the light profiles. This choice was made in order to gauge classification accuracy as a function of distance between the two sources.

Given the size of the postage stamps, we expect that standard source-extraction software, such as SExtractor, will be easily able to identify and separate sources near the borders but may struggle as the sources get closer together. To better highlight this specific problem, we additionally separate blends into two categories: ‘close blends’ and ‘distant blends’. We define close blends as postage stamps in which the two sources are separated by less than ten pixels and distant blends as those in which they are separated by ten or more pixels.

To avoid any spurious correlations, blended images are produced by simulating the galaxies individually and then artificially combining them. For simplicity, we only consider blends consisting of two sources. Real images may well contain cases involving more than two sources as well as various artefacts that would make classification more complicated. We leave these issues to be addressed in future work and here focus on testing the applicability of transfer learning to our simple test case.

2.2. The COSMOS catalogue

We use the Cosmological Evolution Survey (COSMOS, Scoville et al. 2007) catalogue as the basis from which to derive our simplistic training data and more realistic testing data. COSMOS was chosen as it provides a representative sample of galaxies in terms of size, ellipticity, luminosity, and morphology. COSMOS is a catalogue of *Hubble* Space Telescope observations of 1.64 deg^2 on the sky with very accurate photometry and morphology. This catalogue contains high-resolution images ($0.05 \text{ arcsec pixel}^{-1}$) with a very small point spread function (PSF) of 0.01 arcsec .

In particular, we use a subset of this catalogue selected and processed for weak lensing purposes (Mandelbaum et al. 2012). This subset of images have a negligible amount of noise and the PSF has been deconvolved. This makes it possible to re-sample the images on a larger pixel scale and to convolve them with a different PSF in order to mimic observations from another instrument. Finally, the galaxies have been fitted by either a Sérsic profile or a bulge + disc profile for which the bulge is represented by a De Vaucouleurs profile (de Vaucouleurs 1948) and the disc by an exponential profile (see Mandelbaum et al. 2012, for details).

The subset of processed COSMOS data has been made available through the Galsim software package (Rowe et al. 2015). GalSim is widely used in the astrophysics community to simulate and manipulate galaxy images and was used

¹ <https://www.cfht.hawaii.edu/Science/CFIS/>

extensively in several weak lensing challenges, such as GREAT3 (Mandelbaum et al. 2014). The package provides all the tools necessary to work with COSMOS data.

2.3. Point spread function

The impulse response or PSF of an instrument encompasses all of the aberrations induced by the optical system along with other distortions arising from the atmosphere, and so on. The PSF induces blurring in observed images, which artificially increases the size of sources and can cause their light profiles to overlap creating blends.

We model the optical PSF as a Moffat profile with $\beta = 4.765$ (Trujillo et al. 2001) and the PSF ellipticity is drawn from real optical variations of the Canada-France-Hawaii Telescope (CFHT) derived from real CFIS data (Guinot et al. 2021). Atmospheric turbulence is modelled by a Kolmogorov profile (Tatarski 2016) with random ellipticity drawn from a Gaussian distribution with average $\mu = 0$ and standard deviation $\sigma = 0.01$. The final PSF is obtained by convolving the two models and has an average size of 0.65 arcsec. Given the relatively small size of the postage stamps, spatial variations of the PSF are neglected. This CFIS-like PSF model is simple but sufficiently realistic for the purposes of this work.

2.4. Parametric model training data

In order to keep our training samples as simple and generic as possible, we use a series of parametric models derived from fits to the COSMOS sample described in Sect. 2.2. This constitutes a range of simulated galaxies with different sizes, shapes, and ellipticities. The parametric models are then convolved with the CFIS-like PSF described in Sect. 2.3. Each image corresponds to a 51×51 pixel postage stamp with a convolved galaxy model at the centre.

For half of the sample, a second galaxy model is placed at a random position within the postage stamp to produce blends according to the definition provided in Sect. 2.1. We pad all of the images with zeros (7 pixels in every direction) to avoid issues when the secondary source is close to the border of the postage stamp. The complete sample is comprised of 80 000 noiseless, padded postage stamps, half of which are isolated galaxies and the other half are blends.

The final step in producing our simulated training sets is to add random Gaussian noise. In order to test how sensitive blend identification is to noise, we generate seven different noise standard deviations ($\sigma_{\text{noise}} = 5, 10, 15, 20, 25, 30, 35$). We additionally created ten realisations of the noise for each value of σ_{noise} in order to test the stability of the training. Each realisation of each noise level is treated as an independent training set. In other words, we train the network 70 times and obtain 70 sets of weights.

2.5. Realistic testing data

We generate a sample of realistic CFIS-like images to test our transfer learning approach. To do so, we take the real denoised and deconvolved COSMOS images described in Sect. 2.2, crop them to 51×51 pixel postage stamps, and convolve them with the CFIS-like PSF described in Sect. 2.3. The flux is rescaled in order to reproduce a 300 s exposure at the 3.6m CFHT telescope. The images are re-sampled at the resolution of the CFIS survey (0.187 arcsec). Similarly to the training images, blends are created for half of the sample by adding a secondary source at a

Table 1. Summary of the data used for training and testing.

Data set	Parametric training set	Realistic testing set
N_{isolated}	40 000	40 000
N_{blended}	40 000	40 000
σ_{noise}	5, 10, 15, 20, 25, 30, 35	14.5
$N_{\sigma \text{ real}}$	10	1

random position in the postage stamp and then all the postage stamps are zero padded in the same way. Finally, Gaussian noise with $\sigma_{\text{noise}} = 14.5$ was added to the images to replicate the S/N of CFIS data.

The final sample of 80 000 realistic postage stamps is considerably more complex than the parametric models described in Sect. 2.4 and more closely approximate the conditions expected in real images. For example, the light profiles do not necessarily have central symmetry because of star forming regions or complex morphology. Out of the 80 000 postage stamps, 4838 correspond to close blends (i.e., the sources are less than ten pixels apart) and the remaining 35 162 correspond to distant blends. Table 1 summarises all of the data sets used for training and testing.

N_{isolated} is the number of postage stamps containing isolated sources, N_{blended} is the number of postage stamps containing blended sources, σ_{noise} is the amount of noise added to the postage stamps, and $N_{\sigma \text{ real}}$ is the number of realisations for each noise level.

3. Deep transfer learning framework for blend identification

The objective of transfer learning is to learn a set of weights from a given training set and then to transfer these to a related but independent problem. This is interesting from the perspective of blend identification given that we do not have a large sample of labelled data and training with a small sample of known blends may significantly bias the learned weights. Additionally, the use of pre-trained weights considerably reduces the time required to train a network, thereby making the application of a deep learning approach to galaxy survey data more feasible.

For the purposes of this work we test the applicability of transfer learning to the problem of blend identification in a two-step process. In the first step, CNN weights learned from natural images are used to extract generic features from the galaxy images. The fully connected layers are then trained on galaxy image features derived from simple parametric models. The objective being to capture the general properties of blended images and not specific features of the individual galaxies. In the second step, the weights learned from simple parametric models are applied to more realistic CFIS-like images to test the classification accuracy.

The architecture we choose to implement our deep transfer problem was that of VGG-16. For simplicity we refer to our specific VGG-16 setup for the problem of blend identification as BLENDHUNTER.

3.1. The VGG-16 network

VGG-16 is a deep convolutional network with 16 weight layers developed by the Visual Geometry Group (VGG) at the University of Oxford (Simonyan & Zisserman 2014). The network was ranked first in the ImageNet Large-Scale Visual Recognition

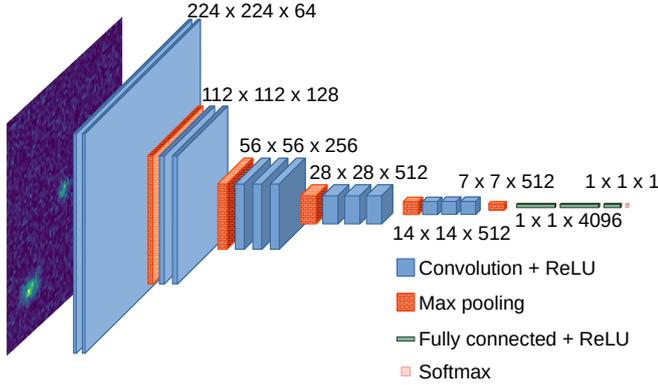


Fig. 1. Visual representation of the VGG-16 network. Convolutional layers with ReLU activation are shown in solid blue, max pooling layers are shown in brick-pattern red, fully connected layers are shown as green bars, and the output softmax layer is shown as the last box in pink.

Challenge (ILSVRC) in 2014 (Russakovsky et al. 2015). The main feature of this architecture was the increased depth of the network compared to the state of the art at the time. In VGG-16, three-channel images (RGB) are passed through five blocks of convolutional layers, where each block is composed of increasing numbers of 3×3 filters. The stride (i.e., the amount by which the filter is shifted) is fixed to 1, while the convolutional layer inputs are padded such that the spatial resolution is preserved after convolution (i.e., the padding is 1 pixel for 3×3 filters). The blocks are separated by max-pooling (i.e., down-sampling) layers. Max-pooling is performed over 2×2 windows with a stride of 2. The five blocks of convolutional layers are followed by three fully connected layers. The final layer is a soft-max layer that outputs class probabilities. The full network architecture used is shown in Fig. 1.

The VGG-16 network was chosen for the work presented here for several reasons. Firstly, the network can be implemented with weights pre-trained on the ImageNet database (Deng et al. 2009) in order to save computation time and resources. The diversity of this data set has allowed the network to learn a variety of generic image features applicable to most image-classification tasks. Finally, VGG-16 has already been applied in a variety of astrophysical problems, including galaxy morphology classification (Wu et al. 2019; Zhu et al. 2019), glitch classification in gravitational wave data (George et al. 2018), and coronagraph image classification (Shan et al. 2020). The VGG-16 network was implemented in Python via the TensorFlow-Keras neural network API (Chollet et al. 2015; Abadi et al. 2015), which includes ImageNet pre-trained weights.

3.2. Training and validation

We trained BlendHunter in two phases using the 70 sets of 80 000 simulated images described in Sect. 2.4. We divided each data set in the following way: 36 000 images were used for training, 36 000 for validation, and the remaining 8000 for testing.

3.2.1. Convolutional layers

In the first phase, we simply initialise the convolutional layers (i.e., the first 13 weight layers) with the pre-trained ImageNet weights. As mentioned in the previous section, the purpose of using pre-trained weights is to save processing time. Therefore,

this part of the network is not trained with our data and can be seen as a simple feature extractor.

As ImageNet weights are used, the VGG-16 network expects RGB images as inputs. Therefore, we rescaled the simulated images to the range $i \in \mathbb{Z} : 0 \leq i \leq 255$ and the monochromatic pixel values were repeated across the three RGB channels. The final images are saved as portable network graphics (PNG) files which are fed into the network.

Given the relatively small amount of training data, we additionally implemented data augmentation in order to obtain more diversified features from the convolutional layers of the network. Augmenting our simulated images simply means creating additional images with minor changes such as flips, translations, or rotations. Specifically, we used the Keras image pre-processing modules to include a shear range and zoom range of 0.2, as well as horizontal flipping. We note that data augmentation was not used on the test images.

Figure 2 shows some examples of features extracted from one of the simulated images with the convolutional layers. The earlier convolutional blocks provide more general features while the later blocks provide more specific features.

3.2.2. Fully connected layers

In the second phase, we train the fully connected layers (i.e., the last three weight layers) on top of the previously stored features. These layers consist of a layer with linear transformation, followed by a layer with dropout and finally a layer with rectified linear unit (ReLU) activation. We chose this non-linear activation function, widely used with deep neural networks, because it allows for faster convergence. We used dropout alongside data augmentation to avoid over-fitting (Srivastava et al. 2014). This is a way of cancelling some activations to prevent the network from learning random correlations in the images. We employed a dropout rate of 0.1 for our experiments. For the training, we proceeded with batches of 250 images at a time over 500 epochs. An epoch corresponds to passing the entire training sample through the network and a batch size of 250 proved to be the most computationally efficient for the gradient descent.

We use the binary cross entropy loss function in Eq. (1) :

$$\text{BCE} = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)), \quad (1)$$

where y_i is the true label (1 for blended, 0 otherwise) for the i^{th} image and $p(y_i)$ is the probability returned by the network of the image being blended or not. We selected this loss function to optimise as it is typically used in this type of binary classification problem.

The Adam algorithm (Kingma & Ba 2014) was used to minimise our loss function. In contrast to a classic stochastic gradient descent (SGD) algorithm, it computes individual learning rates for each weight and is relatively efficient in the optimisation of deep neural networks. In order to adapt the learning rate, Adam requires the estimations of the first (the mean) and second (the uncentred variance) moments of the gradient using exponentially moving averages (Eqs. (2) and (3), respectively), with g_t the objective function gradient at time-step t , and β_1 and β_2 the exponential decay rates for the moment estimates:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \quad (2)$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \quad (3)$$

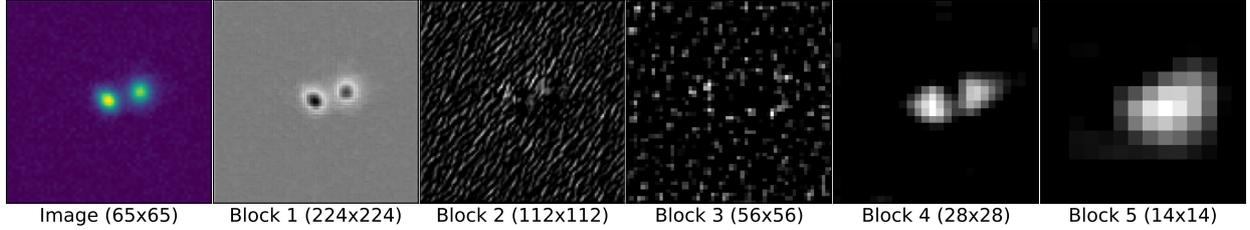


Fig. 2. Examples of features extracted from the VGG-16 convolutional layers pre-trained with ImageNet weights. *Leftmost panel:* input postage stamp containing blended sources. The *following panels* show example features extracted from various convolution blocks.

Adam takes advantage of both RMSprop (Tieleman & Hinton 2012) and AdaGrad (Duchi et al. 2011) methods. In addition to being robust and less time-consuming, it can therefore be applied to a wider selection of optimisation problems. It also requires almost no tuning of its parameters. We set β_1 and β_2 to their default values, respectively 0.9 and 0.999. Finally, the weights are updated according to Eq. (4), where w_t are the fully connected network weights at time-step t , η is the step size, \hat{m}_t and \hat{v}_t are the bias-corrected estimators of the first and second moments, and ϵ is a value set to 10^{-8} to prevent division by 0.

$$w_t = w_{t-1} - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}}. \quad (4)$$

At the end of each epoch, both training and validation losses were computed, and the weights were updated every time the validation loss decreased. Training was stopped when the validation loss had not decreased after ten epochs. The network converged after around 70 epochs on average.

We began training with a learning rate of 1×10^{-3} . As choosing the right learning rate can be challenging, we decided to reduce the learning rate by a factor of 0.5 every time the validation loss did not decrease after five epochs. A small learning rate would make it possible to avoid big jumps in gradient descent. Otherwise, in this case, it could fail to converge and settle around a local minimum. No weight decay was implemented in this phase.

Tuning deep neural network hyperparameters with a considerable amount of parameters to learn can prove to be very time-consuming. This is why we focused on the hyperparameters that would have the most impact on the results. Several tests were made such as changes to the regularisation, weight initialisation, dropout rate, learning rate, and optimiser. However, no significant improvement in accuracy was observed. Switching to the SGD optimiser or increasing the dropout rate led to worse performance overall.

The network takes approximately 630 s to train on a sample of 80 000 images using a standard Intel(R) Core(TM) i7-6900K CPU (3.20 GHz, 32GB).

4. Results

4.1. SExtractor benchmark

We compared the performance of BLENDHUNTER with SExtractor, as the latter is the most widely used tool in the community for identifying and handling blended sources in astronomical images. The objective was to test the reliability of our approach compared to the state of the art. Specifically, we made use of the SExtractor Python wrapper SEP (Barbary 2016) for our tests. We note that SEP does many things beyond

Table 2. SEP parameter settings.

Parameter	Value
THRESH_TYPE	RELATIVE
DETECT_THRESH	1.5
DETECT_MINAREA	5
FILTER	Y
FILTER_NAME	kernel_3x3.conv (default)
DEBLEND_NTHRESH	32
DEBLEND_MINCONT	0.005

blend identification and many of these steps cannot easily be isolated; however we tried to make the comparison as fair and consistent as possible.

SEP implements a multi-thresholding technique to decompose detected sources into subsources (when possible). This method takes two input parameters: the number of bins to decompose the light profile and the minimum contrast value between the main peak and a given subpeak. The contrast is evaluated based on the flux of each peak (see Fig. 2 in Bertin & Arnouts 1996). The set of parameters we used for SEP is shown in Table 2 (all the other parameters are kept to their default values). The value of 0.005 for DEBLEND_MINCONT (the minimum contrast parameter for deblending) may be considered a little high compared to that commonly used in the literature, but here we chose to favour reliable identification over increasing the number of blends found at the cost of also increasing the number of spurious detections.

Given our loose definition of blended sources (see Sect. 2.1), we chose not to rely exclusively on the deblending flags provided by SEP. Instead, we also check that the sources are found at the right positions (within a two pixel radius) to make sure we do not extract noise features. Additionally, as some of the sources in the postage stamp do not technically overlap, when SEP correctly identifies the number of sources (i.e., 1 or 2) we take this as a correctly labelled postage stamp.

The process for labelling postage stamps as either isolated or blended sources using SEP can be summarised as follows:

If a single source is detected:

- If the source is flagged as a blend and is at the expected position, the image is labelled as a blend.
- Otherwise, the image is labelled as an isolated source.

If two sources are detected:

- If both sources are detected at the right positions, the image is labelled as a blend (this stands even if one of the two sources is itself detected as a blend).
- If the sources are incorrectly identified, the image is labelled as an isolated source.

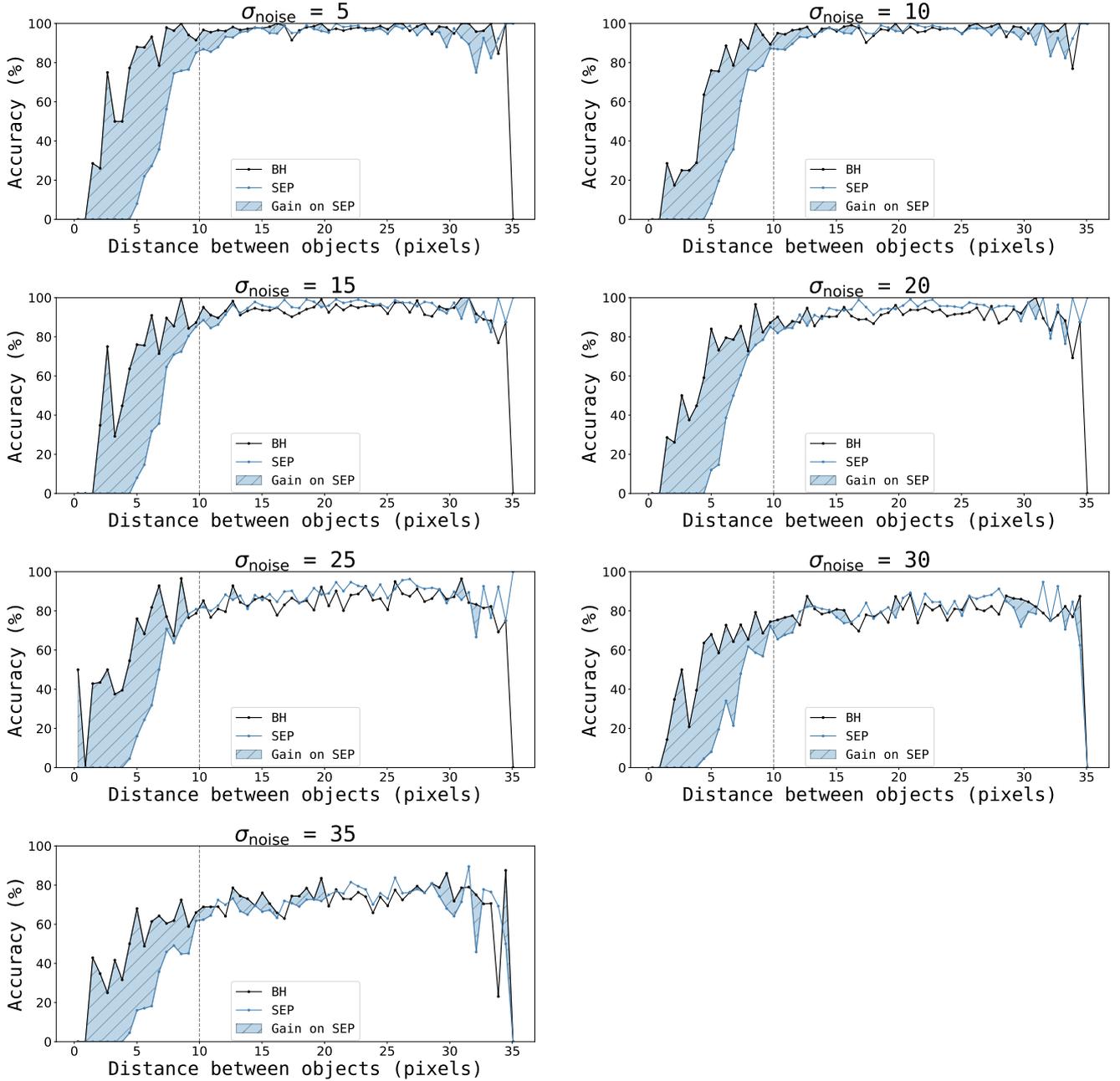


Fig. 3. Classification accuracy of BLENDHUNTER (black solid line) vs. SEP (blue dashed line) as a function of separation between sources. Results are from the blended samples of the COSMOS parametric model testing sets. Each panel shows one realisation of a given noise standard deviation, σ_{noise} . The blue shaded area shows the gain in accuracy of BLENDHUNTER with respect to SEP.

If more than two sources are detected:

- If at least the right number of sources are detected at their expected positions, the image is labelled as a blend.
- If the number of detected sources is larger than the number of true sources, the image is flagged as a SEP failure and is not included in the analysis. We note that this occurred for less than 1% of the images.

4.2. Results from parametric models

In the following, we define classification accuracy as the percentage of postage stamps correctly labelled as either isolated or blended sources. Here we take the weights obtained via the train-

ing procedure described in Sect. 3.2 and apply them to the test sample of 8000 parametric model postage stamps (see Sect. 2.4) to obtain classification labels.

Figure 3 shows the classification accuracy of BLENDHUNTER (black solid line) versus SEP (blue dashed line) as a function of separation between sources. Results are from the blended samples of the COSMOS parametric model testing sets (i.e., sets of 4000 postage stamps). Each panel shows one realisation of a given noise standard deviation, σ_{noise} . The blue shaded area shows the gain in accuracy of BLENDHUNTER with respect to SEP. For close blends (i.e., where the two sources are less than ten pixels apart), BLENDHUNTER significantly outperforms SEP, showing gains as high as $\sim 80\%$ in classification accuracy.

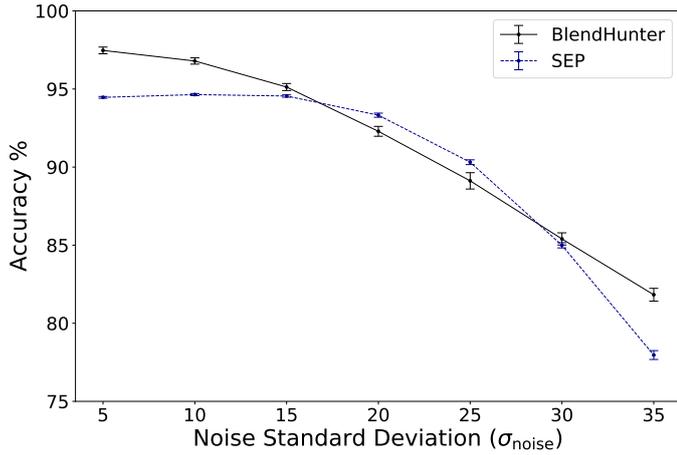


Fig. 4. Overall classification accuracy of BLENDHUNTER (black solid line) versus SEP (blue dashed line) with respect to σ_{noise} on the COSMOS parametric model testing set. The points are taken from the average accuracy from ten realisations of each noise level and the error bars from the standard deviation.

For distant blends, the two techniques appear to be consistent to within a few percent and correctly label the majority of the postage stamps.

In Fig. 4 we display the overall accuracy as a function of the standard deviation of the noise for BLENDHUNTER and SEP on the full parametric model testing set (i.e., sets of 8000 postage stamps). The points are taken from the average classification accuracy from the ten noise realisations and the error bars from the standard deviation. As expected, both approaches perform almost perfectly for low noise levels and their performance drops off as the noise increases. Overall, the two approaches appear fairly consistent, however BLENDHUNTER drops off less rapidly for higher noise levels indicating it may be slightly more robust in higher noise regimes.

4.3. Results from realistic CFIS-like images

Here we test the weights obtained from the parametric models with various noise levels and apply them to the test sample of 80 000 realistic CFIS-like postage stamps (see Sect. 2.5) to obtain classification labels.

Figure 5 shows the relative classification accuracy as function of the standard deviation of the training noise (i.e., the noise level of the sample used for training) for BLENDHUNTER with respect to SEP on the realistic CFIS-like postage stamps. Points show the average relative classification accuracy corresponding to the ten noise realisations for each training noise level and the error bars are the standard deviation. SEP is simply run once on the full data set. For the whole set of images (i.e., all 80 000 postage stamps), we can see that BLENDHUNTER is able to slightly outperform SEP (only by a few percent) around $\sigma_{\text{noise}} = 14.5$, which is the true noise standard deviation of the CFIS-like sample. The classification accuracy then degrades if the network is trained on noise levels that differ significantly. In the subset of close blends (4838 postage stamps), BLENDHUNTER significantly outperforms SEP regardless of the training conditions. This is consistent with the results from the parametric model data. Finally, for distant blends (35 162 postage stamps), BLENDHUNTER matches the performance of SEP when trained on the same noise level as the target images and under-performs by a maximum of 5% when trained on a

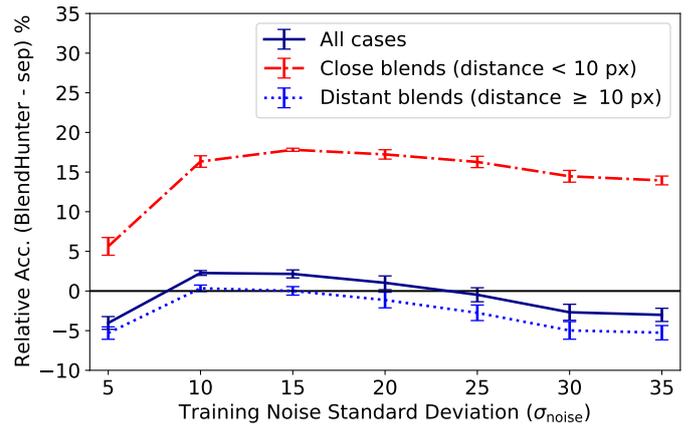


Fig. 5. Relative classification accuracy for BLENDHUNTER with respect to SEP on the realistic CFIS-like postage stamps, with $\sigma_{\text{noise}} = 14.5$. The points are taken from the average relative classification accuracy from the ten training noise realisations and the error bars from the standard deviation. For close blends (red dot-dashed line), BLENDHUNTER outperforms SEP for any level of noise in the training data.

noise level that differs. The overall accuracy of SEP in each case is: 91% over the whole data set, 69% for close blends, and 94% for distant blends.

These results indicate that BLENDHUNTER is not overly sensitive to the training without perfect knowledge of the noise in the target sample. This is a strong indication that the transfer learning has been successful and is promising for the prospect of applying this approach to real survey data. Additionally, the performance of BLENDHUNTER is noticeably more robust for close blends, where galaxy profiles significantly overlap. It should be stressed that these results are based on the relatively simple blend definition provided in Sect. 2.1, which does not take into account all of the complexities that could be encountered for real extended sources.

We additionally examined the confusion matrices for the predictions from the realistic CFIS-like test set. Figure 6 shows the fraction of blends correctly classified as blends (i.e., true positives) or isolated sources correctly classified as isolated sources (i.e., true negatives) as a function of the training noise standard deviation. The first panel shows the performance of BLENDHUNTER compared to SEP for the sample of close blends (green lines) along with an equivalent number (4838) of isolated sources (red lines). The second panel shows the equivalent plot for distant blends with the remaining sample of isolated sources. The third panel shows the results for all 80 000 postage stamps.

The results show that BLENDHUNTER is significantly better at identifying true blends than SEP regardless of the noise level used for training. This is particularly noticeable for the sample of close blends. However, BLENDHUNTER also produces more incorrect labels for isolated sources and this worsens as the training noise level increases beyond that of the target sample.

5. Conclusions

We present a proof-of-concept deep transfer learning approach for the automated and robust identification of blended sources in galaxy survey data called BLENDHUNTER. This technique uses convolution layers pre-trained on natural images from ImageNet, thus significantly reducing the time required for training. The fully connected layers are then trained on simple

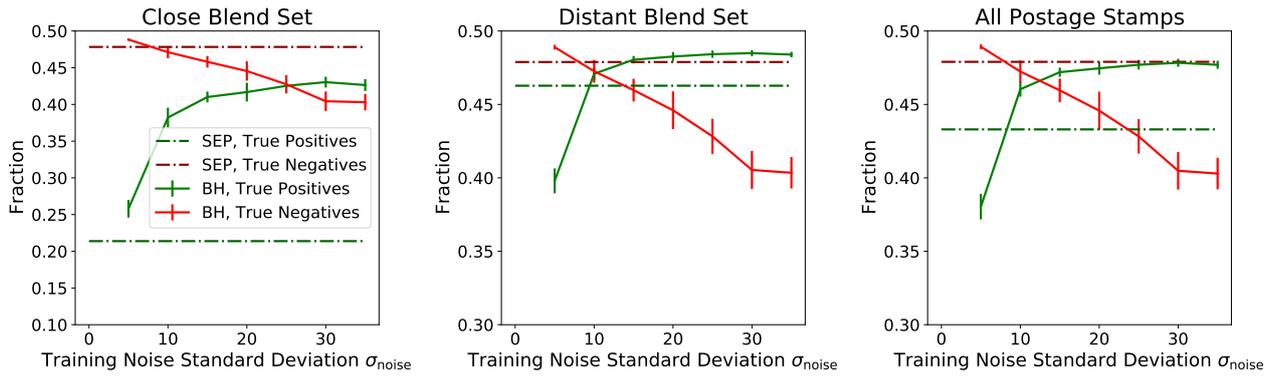


Fig. 6. Classification confusion matrices of BLENDHUNTER (solid lines) compared to SEP (dot-dashed lines) for the realistic CFIS-like test data. Each panel shows the fraction of blends correctly classified as blends (i.e., true positives, green lines) or isolated sources correctly classified as isolated sources (i.e., true negatives, red lines) as a function of the training noise standard deviation. *Left panel:* sample of close blends along with an equivalent number of isolated sources. *Middle panel:* sample of distant blends along with the remaining isolated sources. *Right panel:* results for the full set of postage stamps.

parametric models derived from COSMOS data for various levels and realisations of Gaussian random noise.

Comparison with the community standard SEP (a Python implementation of SExtractor) on 70 sets of 8000 parametric models demonstrates that BLENDHUNTER is significantly more sensitive to blends when the images are very noisy or when galaxy pairs are very close (<10 pixels). In the low-noise distant blend regime, the performance of BLENDHUNTER appears to be consistent with that of SEP.

We additionally produced a sample of 80 000 more realistic CFIS-like images derived from real COSMOS data. Results demonstrate that the BLENDHUNTER weights, learned on the parametric models, can successfully be transferred to this more complex data set and achieve >90% classification accuracy, outperforming SEP by a few percent on the full sample when the appropriate weights are used. The results also indicate that BLENDHUNTER is capable of achieving results that are roughly consistent with those of SEP even when weights are used that have been trained with a significantly different noise standard deviation.

BLENDHUNTER notably outperforms SEP by 5–15% for blends in which the galaxies are separated by less than ten pixels. This is an interesting result as these are precisely the cases that generate biases in galaxy shape measurements (MacCrann et al. 2022).

Overall the results are promising and indicate that it may be possible to adapt this approach to more accurately identify blended sources in real survey data. The next steps in this direction would entail the generation of more realistic testing data that contain some artefacts and images with more than two sources. Further work is also required to reduce the number of false negatives, i.e., incorrect labels for isolated sources. Finally, additional tests should be performed to determine if and to what degree the use of pre-trained weights in the CNN layers can help prevent over-fitting the network to the training sets. We leave the investigation and implementation of these steps for future work.

Acknowledgements. The authors wish to acknowledge the COSMIC project funded by the CEA DRF-Impulsion call in 2016, the CrossDisciplinary Program on Numerical Simulation (SILICOSMIC project in 2018) of CEA, the French Alternative Energies and Atomic Energy Commission. The Euclid Collaboration, the European Space Agency and the support of the Centre National d’Etudes Spatiales. This work was also supported by the ANR AstroDeep project – grant 19-CE23-0024-01. This work has made use of the CANDIDE Cluster at the Institut d’Astrophysique de Paris and made possible by grants from the PNCG and the

DIM-ACAV. This work is based on data obtained as part of the Canada-France Imaging Survey, a CFHT large program of the National Research Council of Canada and the French Centre National de la Recherche Scientifique. Based on observations obtained with MegaPrime/MegaCam, a joint project of CFHT and CEA Saclay, at the Canada-France-Hawaii Telescope (CFHT) which is operated by the National Research Council (NRC) of Canada, the Institut National des Science de l’Univers (INSU) of the Centre National de la Recherche Scientifique (CNRS) of France, and the University of Hawaii. This research used the facilities of the Canadian Astronomy Data Centre operated by the National Research Council of Canada with the support of the Canadian Space Agency. AZV would like to thank LSC/PMR/EP at University of São Paulo for providing additional computing power. The authors also wish to thank Xinyu Wang and Alexandre Bruckert for initial efforts that were expanded upon in this work.

References

- Abadi, M., Agarwal, A., Barham, P., et al. 2015, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, software available from tensorflow.org
- Arcelin, B., Doux, C., Aubourg, E., Roucelle, C., & LSST Dark Energy Science Collaboration 2021, *MNRAS*, **500**, 531
- Awang Iskandar, D. N. F., Zijlstra, A. A., McDonald, I., et al. 2020, *Galaxies*, **8**, 88
- Barbary, K. 2016, *J. Open Source Softw.*, **1**, 58
- Bertin, E., & Arnouts, S. 1996, *A&AS*, **117**, 393
- Bosch, J., Armstrong, R., Bickerton, S., et al. 2018, *PASJ*, **70**, S5
- Chollet, F., et al. 2015, *Keras* <https://keras.io>.
- Cropper, M., Cole, R., James, A., et al. 2012, in *Space Telescopes and Instrumentation 2012: Optical, Infrared, and Millimeter Wave*, eds. M. C. Clampin, G. G. Fazio, H. A. MacEwen, J. Oschmann, & M. Jacobus, *SPIE Conf. Ser.*, **8442**, 84420V
- de Vaucouleurs, G. 1948, *Ann. Astrophys.*, **11**, 247
- Deng, J., Dong, W., Socher, R., et al. 2009, *Proc. CVPR*
- Duchi, J., Hazan, E., & Singer, Y. 2011, *J. Mach. Learn. Res.*, **12**, 2121
- Euclid Collaboration (Martinet, N., et al. 2019, *A&A*, **627**, A59
- George, D., Shen, H., & Huerta, E. A. 2018, *Phys. Rev. D*, **97**
- Guinot, A., Kilbinger, M., Farrens, S., et al. 2021, *A&A*, submitted
- Hartlap, J., Hilbert, S., Schneider, P., & Hildebrandt, H. 2011, *A&A*, **528**, A51
- Hoekstra, H., Viola, M., & Herbonnet, R. 2017, *MNRAS*, **468**, 3295
- Joseph, R., Courbin, F., & Starck, J.-L. 2016, *A&A*, **589**, A2
- Kingma, D. P., & Ba, J. 2014, Adam: A Method for Stochastic Optimization, cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015
- Kotsiantis, S. 2007, *Super. Mach. Learn.: Rev. Class. Tech.*, **31**, 249
- Lecun, Y., Bengio, Y., & Hinton, G. 2015, *Nature*, **521**, 436
- MacCrann, N., Becker, M. R., McCullough, J., et al. 2022, *MNRAS*, **509**, 3371
- Mandelbaum, R. 2018, *ARA&A*, **56**, 393
- Mandelbaum, R., Hirata, C. M., Leauthaud, A., Massey, R. J., & Rhodes, J. 2012, *MNRAS*, **420**, 1518
- Mandelbaum, R., Rowe, B., Bosch, J., et al. 2014, *ApJS*, **212**, 5
- Melchior, P., Moolekamp, F., Jerdee, M., et al. 2018, *Astron. Comput.*, **24**, 129
- Reiman, D. M., & Göhre, B. E. 2019, *MNRAS*, **485**, 2617

- Rowe, B. T. P., Jarvis, M., Mandelbaum, R., et al. 2015, *Astron. Comput.*, **10**, 121
- Russakovsky, O., Deng, J., Su, H., et al. 2015, *Int. J. Comput. Vis.*, **115**, 211
- Samuroff, S., Bridle, S. L., Zuntz, J., et al. 2018, *MNRAS*, **475**, 4524
- Sanchez, J., Mendoza, I., Kirkby, D.P., Burchat, P.R., & LSST Dark Energy Science Collaboration 2021, *JCAP*, **2021**, 043
- Scoville, N., Ausser, H., Brusa, M., et al. 2007, *ApJS*, **172**, 1
- Shan, J.-H., Feng, L., Yuan, H.-Q., et al. 2020, *Chin. Astron. Astrophys.*, **44**, 507
- Simonyan, K., & Zisserman, A. 2014, CoRR [arXiv:1409.1556]
- Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., et al. 2016, *Front. Robot. AI*, **2**, 36
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, *J. Mach. Learn. Res.*, **15**, 1929
- Tanoglidis, D., Ćiprijanović, A., & Drlica-Wagner, A. 2021, *Astron. Comput.*, **35**
- Tatarski, V. I. 2016, *Wave propagation in a turbulent medium* (Courier Dover Publications)
- Tieleman, T., & Hinton, G. 2012, *Lecture 6.5–RmsProp: Divide the gradient by a running average of its recent magnitude* (COURSERA: Neural Networks for Machine Learning)
- Trujillo, I., Aguerri, J., Cepa, J., & Gutiérrez, C. 2001, *MNRAS*, **328**, 977
- Wei, W., Huerta, E. A., Whitmore, B. C., et al. 2020, *MNRAS*, **493**, 3178
- Wu, C., Wong, O. I., Rudnick, L., et al. 2019, *MNRAS*, **482**, 1211
- Zhu, X.-P., Dai, J.-M., Bian, C.-J., et al. 2019, *Ap&SS*, **364**, 55