

The PAU Survey: Background light estimation with deep learning techniques

L. Cabayol-Garcia^{1*}, M. Eriksen^{1†‡}, A. Alarcón^{2,3}, A. Amara⁴, J. Carretero^{1‡},
 R. Casas^{2,3}, F. J. Castander^{2,3}, E. Fernández¹, J. García-Bellido⁵,
 E. Gaztanaga^{2,3}, H. Hoekstra⁶, R. Miquel^{1,7}, C. Neissner^{1‡}, C. Padilla¹
 E. Sánchez⁸, S. Serrano², I. Sevilla-Noarbe², P. Tallada^{8‡}, L. Tortorelli⁹

¹*Institut de Física d'Altes Energies (IFAE), The Barcelona Institute of Science and Technology, 08193 Bellaterra (Barcelona), Spain*

²*Institute of Space Sciences (ICE, CSIC), Campus UAB, Carrer de Can Magrans, s/n, 08193 Barcelona, Spain*

³*Institut d'Estudis Espacials de Catalunya (IEEC), 08193 Barcelona, Spain*

⁴*Institute of Cosmology & Gravitation, University of Portsmouth, Dennis Sciama Building, Burnaby Road, Portsmouth PO1 3FX, UK*

⁵*Instituto de Física Teórica UAM/CSIC, Universidad Autónoma de Madrid, 28049 Madrid, Spain*

⁶*Leiden Observatory, Leiden University, Leiden, The Netherlands*

⁷*Institució Catalana de Recerca i Estudis Avançats, E-08010 Barcelona, Spain*

⁸*Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), Madrid, Spain*

⁹*Institute for Particle Physics and Astrophysics, ETH Zürich, Wolfgang-Pauli-Str. 27, 8093 Zürich, Switzerland*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

The PAU Survey (PAUS) is an imaging survey using a 40 narrow-band filter camera, named PAU Camera (PAUCam). Images obtained with the PAUCam are affected by scattered light: an optical effect consisting of light multiply reflected that deposits energy in specific detector regions contaminating the science measurements. Fortunately, scattered light is not a random effect, but it can be predicted and corrected for. However, with the current background estimation method around 8% of the PAUS flux measurements are affected by scattered light and therefore flagged. Additionally, failure to flag scattered light results in photometry and photo-z outliers. This paper introduces BKGNET, a deep neural network to predict the background and its associated error. We have found that BKGNET background predictions are very robust to distorting effects, such as scattered light and absorption, while still being statistically accurate. On average, the use of BKGNET improves the photometric flux measurements by 7% and up to 20% at the bright end. BKGNET also removes a systematic trend with magnitude in the *i*-band that is present with the current background estimation method. With BKGNET, we reduce the photometric redshift outlier rate by 35% for the best 20% galaxies selected with a photometric quality parameter.

Key words: techniques: photometric – light pollution – instrumentation: photometers

1 INTRODUCTION

Wide-field surveys are broadly divided into two types: spectroscopic and photometric surveys. The former surveys lead to precise redshift measurements with relatively low

galaxy density, e.g DEEP2 (Newman et al. 2013) and SDSS (Blanton et al. 2017). On the other hand, broad band photometric surveys cover larger contiguous areas but lead to less precise redshift measurements, with a typical uncertainty of 5%, e.g DES (Abbott et al. 2018).

The Physics of the Accelerating Universe Survey (PAUS) is an imaging survey that aims to measure photo-z with a high precision to faint magnitudes ($i_{AB} < 22.5$) while covering large areas of the sky (Martí et al. 2014). This

* E-mail: lcabayol@ifae.es

† E-mail: eriksen@pic.es

‡ Also at Port d'Informació Científica (PIC), Campus UAB, C. Albareda s/n, 08193 Bellaterra (Cerdanyola del Vallès), Spain

is possible thanks to the PAUCam instrument (Castander et al. 2012; Padilla et al. 2016, 2019), an optical camera equipped with 40 narrow bands (NB), covering a wavelength range from 450nm to 850nm (Casas et al. 2016). The NB filter set has 13nm FWHM and a separation between consecutive bands of 10nm. The camera is also equipped with 6 *ugrizY* broad band filters, which are mainly used by external observers. The camera has 18 red-sensitive fully depleted Hamamatsu CCD detectors (Casas et al. 2012), although only the 8 central CCDs are used for NB imaging. Each CCD has 4096x2048 pixels with a pixel scale of 0.26 arcsec/pix. The NB filter set effectively measures a high resolution photometric spectrum ($R \approx 50$). Some of the goals of such an instrument are to perform detailed studies of intermediate-scale cosmic structure (Stohtert et al. 2018), to obtain precise measurement of intrinsic galaxy alignments at $z \sim 0.75$ and contributing to the effective modelling of galaxies in image simulations (Tortorelli et al. 2018).

Thanks to the high wavelength resolution provided by the 40 NB PAUCam filters and based on simulations (Martí et al. 2014), PAUS aimed to reach a photo- z precision $\sigma(z)/(1+z) \sim 0.0035$ for the 50% of the sample, versus the typical 5% for broad band measurements. This is already achieved in Eriksen et al. (2019). Nevertheless, PAUS photo- z catalogue for the full COSMOS sample with $i_{AB} < 22.5$ still contains outliers when compared to the spectroscopic measurements. Some of these outliers arise from biased photometry that may be caused by scattered light, which is the result of light deflecting from the instrument optical path and detected at a different region of the detector, thus contaminating the true flux from astronomical sources imaged in that region. Furthermore, the excess scattered light decreases the signal-to-noise ratio (SNR) and therefore limits the ability to detect faint objects. Nevertheless, scattered light in the PAUCam camera is well localised, appearing only on the edges of the CCDs.

In 2016 the camera was modified in order to mitigate the effect of scattered light by introducing baffles on all the edges of the NB filters of each filter tray. Although this reduced the amount of scattered light, residuals remain. In the latest COSMOS data reduction, around 8% of exposures taken before the camera intervention are flagged as affected by scattered light, and therefore dismissed. After the intervention, this number reduced to 5% of the exposures, such that on average 7% of data in the COSMOS field are lost due to scattered light.

In any imaging survey, estimating the background is a basic step towards measuring the photometry of a source. Errors on such estimation can therefore propagate into errors on the photometry, such as source detection errors or in the estimates of source fluxes. The main source of background is the night sky's intrinsic brightness, which may contain variations due to different effects: moon, airglow and light pollution. There might also be other effects that contribute to the background, for instance cosmic rays, scattered light or instrumental effects such as readout noise, dark current noise or cross-talk (Romanishin

2014).

Historically, there have been different approaches to estimate such background (Bijaoui 1980; Newell 1983). Some example implementations are DAOPHOT (Stetson 1987) and SExtractor (Bertin & Arnouts 1996). DAOPHOT measures a circular aperture around the target source and estimates the background around the mode of the pixels at certain FWHM from such target source. On the other hand, SExtractor meshes the background, reconstructing a 'background map' with the background estimated at each particular mesh location. For this estimation, it takes the histogram of values and applies σ -clipping until convergence at $\pm 3\sigma$ around the median. Teeninga et al. (2015) find that the SExtractor estimator is biased and propose an alternative approach where the background is estimated at a location without nearby sources instead. Another approach, developed by Popowicz & Smolka (2015), is based on the removal of small objects and an interpolation of missing pixels. They claim this performs well for strongly varying backgrounds.

Here we explore an innovative new approach of background estimation using deep learning techniques. In recent years, deep learning has brought revolutionary advances in computer vision and machine learning (Voulodimos et al. 2018). Breakthroughs in the performance of deep learning algorithms as Neural Networks (NN) (Werbos 1982), or Convolutional Neural Networks (CNN) (LeCun et al. 1989; Lecun et al. 1998; Zeiler & Fergus 2013) together with powerful and efficient parallel computing provided by GPU computing (Krizhevsky et al. 2012) have led deep learning to groundbreaking improvements across a variety of applications. Furthermore, the adoption of Rectified Linear Unit (ReLU) activation functions (Krizhevsky et al. 2012; Xu et al. 2015) instead of other non linear alternatives as the sigmoid has reduced the training time and improved the accuracy in many different applications. Some examples of fields where deep learning has shown its power are image sequence recognition (Donahue et al. 2017), super-resolution images (Dong et al. 2016), video classification (Yue-Hei Ng et al. 2015) or natural language processing (Xie et al. 2018).

The number of deep learning projects applied to cosmology is quickly increasing. This includes astronomical object classification (Carrasco-Davis et al. 2018), gravitational wave detection (George & Huerta 2018), point source detection (Vafaei Sadr et al. 2019), cosmic ray detection (Zhang & Bloom 2019) and directly constraining cosmological parameters from mass maps (Fluri et al. 2018a), (Fluri et al. 2018b), (Herbel et al. 2018), among others. PAUS takes between three and five exposures of the same object in 40 NBs. This large amount of data makes PAUS data a unique dataset to apply deep learning techniques, as shown in Cabayol et al. (2019).

In this paper, we present BKGNET, a new deep learning based method to predict scattered light affected backgrounds. The algorithm provides the background behind a target galaxy and its associated error. This algorithm is designed in the PAUS framework, but it can be also be applied to other future imaging surveys such as LSST

(Ivezić et al. 2019) and *Euclid* (Laureijs et al. 2011). The code is available at <https://gitlab.pic.es/pau/bkgnet>.

The structure of this paper is as follows. In section 2, we describe the PAU Survey and the PAUCam camera and present the modelling of scattered light using sky flats. In section 3, we introduce convolutional neural networks and the specific network we have developed, as well as defining the training and testing process. Sections 4 and 5 contain the results obtained for simulated and real PAUCam images, respectively. In section 6, we validate the network predictions on real target locations and we conclude and summarize in section 7.

2 MODELLING SCATTERED LIGHT

PAUCam images are affected by scattered light, which appears on the edges of some CCDs and increases the amount of background in the affected regions. This can lead to an incorrect estimate of the background if not properly modelled, thus biasing the photometry. Moreover the elevated background lowers the SNR. In this section, we present the PAUCam scattered light model we are using throughout the paper.

2.1 The PAUS observations

PAUS has been observing since 2015B and as of 2019A, PAUS has taken data for 160 nights. The current data cover 10 deg² of the CFHTLS fields¹ W1, W2; 20 deg² in W3 and 2 deg² of the COSMOS field². The PAUS data are stored at the Port d'Informació Científica (PIC), where the data are processed and distributed (Tonello et al. 2019). In this paper we focus only on the data from the COSMOS field, which were taken in the semesters 2015B, 2016A, 2016B and 2017B (the low efficiency was caused by bad weather). The COSMOS field observations comprise a total of 9749 images, 343 images for each NB. From these images, 4928 were taken before the camera intervention and 4821 after. The basic exposure times in the COSMOS field are 70, 80, 90, 110 and 130 seconds from the bluest to the reddest.

The current PAUDM pipeline (Serrano et al. prep; Castander et al. prep), similarly to DAOPHOT, uses an annulus to predict the background around a target source. It does so by calculating the median of the pixels within a ring placed around the source. However, this algorithm requires a (fairly) flat background for an accurate estimate. This is not the case in the presence of scattered light as either the annulus is contaminated by scattered light, or the source itself is. Other effects may contribute too, such as undetected sources, cosmic rays, cross-talk, etc. In order to minimise the effect of any of these artifacts, we perform 3σ clipping of the pixels in the annulus before computing the median. The default PAUDM radii for the annulus are $r_{\text{in}} = 30$ and $r_{\text{out}} = 45$

¹ http://www.cfht.hawaii.edu/Science/CFHTLS_Y_WIRCam/cfhtlsdeepwidefields.html

² <http://cosmos.astro.caltech.edu/>

pixels (Serrano et al. prep). Throughout this paper we use these values to compare this commonly used approach to our deep learning algorithm.

2.2 Sky flats

Figure 1 shows four PAUCam images in the NB filter NB685 before the camera intervention (first and second on the left) and after the camera intervention (third and fourth images). They show scattered light on the edges of the CCD, displaying a spatially varying amount of scattered light. The pattern is the same for the pair of images taken using the NB685 filter before the installation of the baffles. The pattern remains similar for the pair of images obtained after the camera intervention. Comparison to images taken in other filters show that the pattern depends on the filter used.

One way to quantify and model the scattered light is to create background pixel maps per NB. This is done with the following steps:

- i. *Select images*: Select a group of NB images from the same bands, since they have the same scattered light pattern.
- ii. *Compute median*: For each of the images, compute the median background level in the central regions, μ_{BKG} , which are unaffected by scattered light.
- iii. *Estimate ratios*: Divide every image by its median to obtain a pixel ratio map.
- iv. *Mask sources*: Mask the images sources by masking all pixels above a given pixel ratio threshold.
- v. *Combine images*: Combine all individual pixels maps with a median to get a single sky flat for all the selected images.

If the background was flat and poissonian, all pixels in the ratio map should fluctuate along unity. However, if the image is affected by scattered light, the sky flat in affected regions will have a value above unity. We can understand this ratio as approximately the percentage of extra light (scattered light) compared to the flat background. Notice that this model takes into account that scattered light depends on the amount of light falling on the CCD. The procedure in step (v) can be written as

$$\text{skyflat}(x, y) = \text{median}_j \left[\frac{I_j(x, y)}{\mu_{\text{BKG}}} \right], \quad (1)$$

where I_j is image j and the median is over the selected images (step i).

To determine the amount of scattered light we can follow the previous procedure to step [iii]. This way we obtain normalised background images that should fluctuate around unity if they contain a flat background, but would have values above one if they are affected by scattered light. The top panel in Figure 2 shows some of these normalised images for the NB685 filter. It shows the background pixel value from side to side of the image before (black dashed line) and one after (orange solid line) the camera intervention. The plot shows an increasing background on the edges of the CCD before the camera intervention. After the intervention, the amount of scattered light is

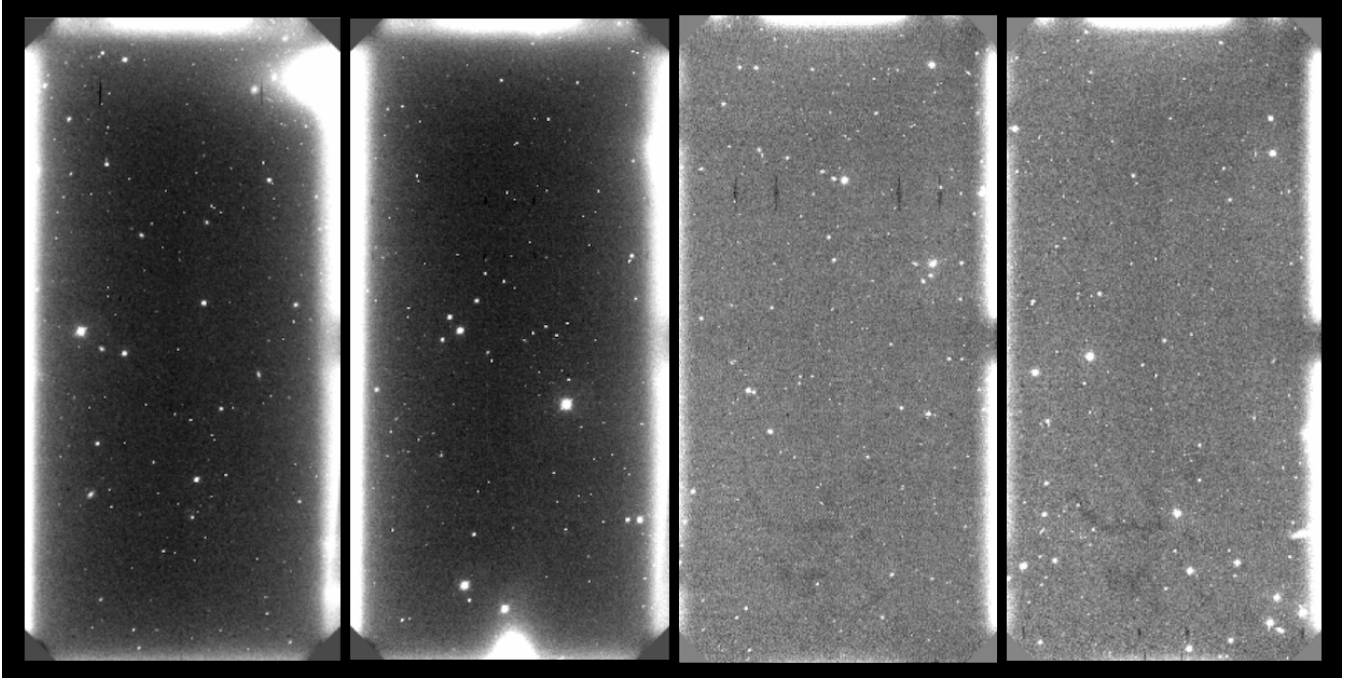


Figure 1. Images taken with the PAUCam, corresponding to NB685. *Left:* The first two images correspond to PAUCam images before the camera intervention. Notice that both exhibit the same scattered light pattern. *Right:* The two images on the right correspond to PAUCam images after the intervention. Again, both present the same scattered light pattern, but different to the first two images on the left. This shows the changes in scattered light patterns with the intervention.

reduced. Unfortunately it is still present and thus needs to be accounted for.

We can use all the normalised background images in a given NB to create a general sky flat for that band (also splitting before/after the intervention). The bottom panel in Figure 2 shows the resulting mean of each sky flat (one per band) as a function of NB. The mean of the sky flat gives information about the amount of scattered light in a given band. We can clearly see the effect of the intervention on the amount of scattered light, which is reduced.

2.3 Sky flat as scattered light correcting method

If the sky flat modelling is sufficiently accurate, it can be used to correct the scattered light on PAUCam images. Assuming all images from a given NB follow the same scattered light pattern scaled by the CCD sky background, a way of correcting scattered light would be

$$\tilde{I}(x, y) = I(x, y) - (\text{sky flat}(x, y) - 1)\mu_{\text{BKG}}, \quad (2)$$

where we subtract from a given target image ($I(x, y)$) the sky flat scaled by the mean background of such image (μ_{BKG}). Notice that instead of subtracting the sky flat, we subtract the sky flat without the flat sky background. This way, the regions without scattered light are barely affected.

Figure 3 shows the original CCD image (left), after correcting with the sky flat (middle) and the sky flat used for correction (right). Visually, the scattered light pattern in the original image (left) disappears after applying the sky flat

correction (middle). However, although the correction seems visually almost perfect, this method has a drawback. Even though scattered light follows approximately a pattern given a band, there might be fluctuations due to other external conditions. For example the weather, moon and observing conditions may induce variations between different observations in a NB. To be more precise on the correction, one should create a template per band and per night, such that the observing conditions are similar. However, for creating a sky flat per night, there might be an insufficient number of images to create an accurate modelling. Bright stars also contribute to scattered light and this cannot be corrected with the sky flat. Figure 4 shows the background level for a specific image in NB685 before and after the correction with the sky flat. In this case, the image is corrected without considering any split on night to generate the sky flat. This means that all images, despite being observed on different night and with different observing conditions are used to build the sky flat. The image without correction displays large peaks at both edges and those are clearly corrected by the sky flat. However, both sides of the CCD still have some strange behaviours, peaks and drops that are caused by scattered light residuals.

3 BKGNET: A DEEP LEARNING BASED METHOD TO PREDICT THE BACKGROUND

In this section we start by describing the basics of deep learning algorithms and define some of the terminology. We then

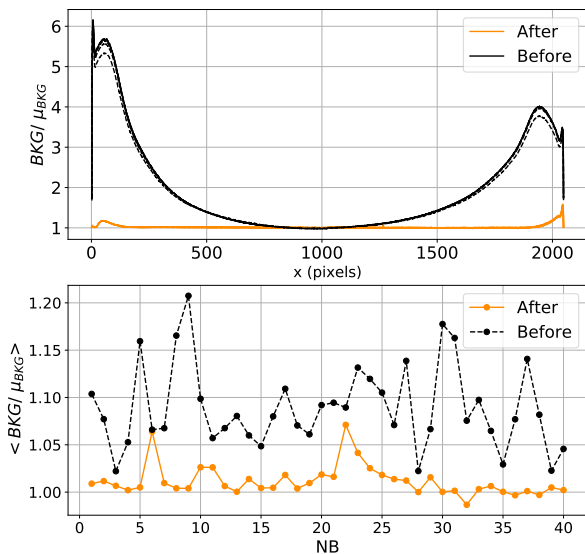


Figure 2. *Top:* Normalised background light content in each pixel as a function of the pixel position in the image for different images before (black dashed line) and after (orange solid line) the camera intervention. Each pixel value is divided by the mean background in the image. Regions without scattered light should fluctuate around unity. Regions affected by scattered light should be above unity. *Bottom:* Given a narrow photometric band, mean value of the normalised background curves considering all the images taken in that band.

describe our training and test samples and define the network.

3.1 Convolutional Neural Networks

Machine learning methods are data analysis techniques where the algorithm learns from the data. In particular, one of the most popular class of algorithms are neural networks (Werbos 1982), which are designed to recognise patterns, usually learned from training data (*supervised method*). They are mainly used for regression and classification problems (Alexander et al. 2019). Deep learning is a subset of machine learning that refers to a development of neural network technology, involving a large number of layers.

Deep learning methods, and in general any supervised machine learning method, model a problem by optimising a set of trainable weights that fit the data. This is done in three stages: *forward propagation*, *back propagation* and *weight optimisation*. The network starts with the *forward propagation*. At this stage, the input data propagates through all the network layers and then, the network gives a prediction for each of the input samples. After that, by comparing with the known true value, which is technically called *label*, the network estimates a prediction error with a given loss function. After that, *back propagation* takes place. *Back propagation* consists of computing the contribution

of each weight on the prediction error. Such contributions are calculated with the partial derivative of the loss with respect to each of the weights. The *weight optimisation* is the weights correction based on the quantities calculated in the *back propagation* to reduce the error in the next iteration.

In this work, we will use a Convolutional Neural Network (CNN; Lecun et al. 1998; Zeiler & Fergus 2013). Our network contains four differentiated types of layers:

Convolutional layer: This layer makes the network powerful in image and pattern recognition tasks. It has a filter, technically named *kernel* and is usually 2-dimensional, which contains a set of trainable weights used to convolve the image. The outcome of this layer is the input image convolved with the kernel. In a given convolutional layer, one can convolve the input with as many kernels as desired. Each of these convolutions will generate a convolved image, which we refer to as *channel*. All of them together are the input of the next layer.

Pooling layers: This layer reduces the dimensionality of the set of convolved images. It applies some function (e.g. sum, mean, maximum) to a group of spatially connected pixels and reduces the dimensions of such group. For example, it takes 2 consecutive pixels and converts them to the mean of both. Although we use it to handle the amount of data generated after the convolutions, it also regularises the model to avoid learning from non-generalisable noise and details in the training data (also known as overfitting).

Fully connected layer: This layer is usually the last layer of the network. Its input is the linearised outcome of the previous ones (in our network, convolutions + poolings). It applies a linear transformation from the input to the output. The slope and bias of the linear transformation are the learning parameters.

Batch normalisation layer: In this layer the network normalises the output of a previous activation layer. It subtracts the mean and divides by the standard deviation. Batch normalisation helps to increase the stability of a neural network and avoids over-fitting problems.

After each convolution and fully connected layer there is an activation function that transforms the outcome. Activation functions are non-linear functions that map the outcome of a layer to the input of the following one. An example of an activation function is the Rectified Linear Unit (ReLU) (Krizhevsky et al. 2012), although we use a variation of this function called LeakyReLU (Xu et al. 2015), with which we find better results. Other terms that one needs to be familiar with are *epoch* and *batch*. An *epoch* is an iteration over the complete training dataset. It is common practice to avoid feeding the network with all the training sample together. Instead, the training data is divided in groups of a certain size and each of these groups is called *batch*. Feeding the network in batches helps it learn faster as in every iteration over a batch, it back-propagates updating all the weights. Then, instead of updating once per epoch, it updates as many times as there are batches. The amount of variation allowed per iteration is regulated by the *learning rate*.

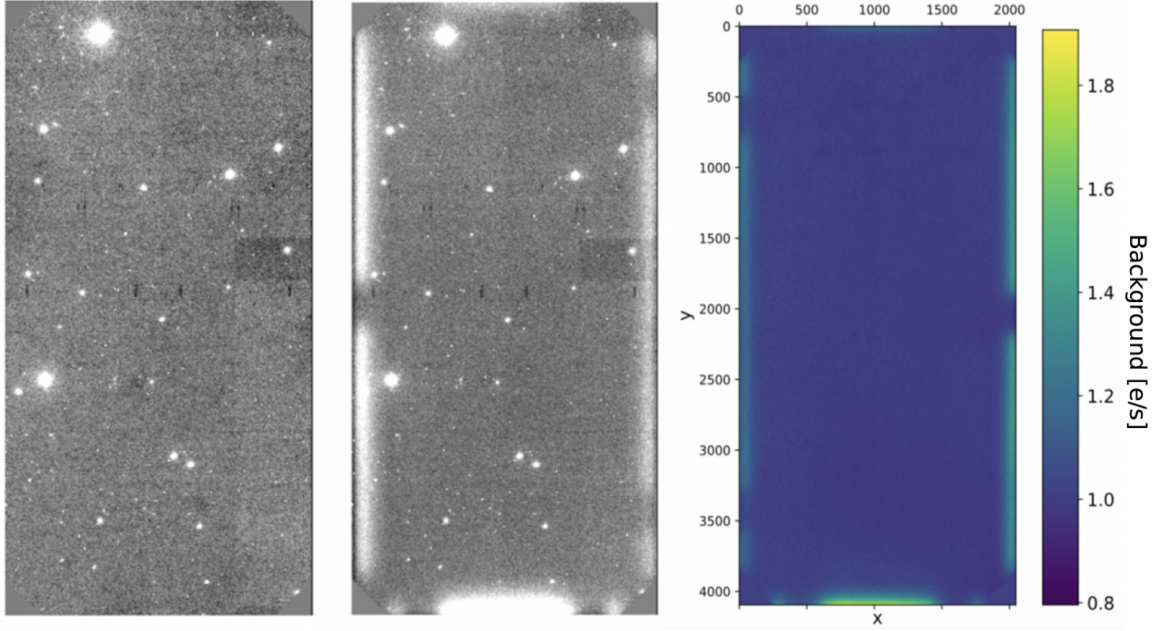


Figure 3. *Left:* Image taken in NB685 with a scattered light pattern on the edges after correcting scattered light with a sky flat. *Middle:* Image taken in NB685 with a scattered light pattern on the edges. *Right:* The sky flat generated with equation 1 considering all images taken the same observation night as the original image.

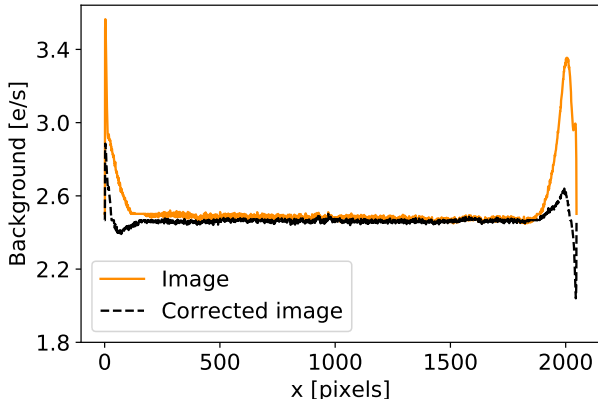


Figure 4. Background pixel values across the image. The original image (orange solid line) displays high peaks on the edges caused by scattered light. After correcting with the sky flat (dashed black line) the peaks are reduced, but some residuals remain.

3.2 Algorithm scheme

The goal of this work is to build a deep learning based algorithm capable of learning the underlying behavior of scattered light and other distorting effects present in the PAUCam images and do background and error predictions in the location of the target sources. The network is named BKGNET³ and it is built using the PyTORCH library (Paszke et al. 2017). BKGNET has two main blocks: a convo-

lutional neural network (CNN) and a linear neural network. Figure 5 shows the BKGNET architecture. The CNN block handles the information coming from the image itself, as the background we want to recover is encoded in the pixel values. The inputs are 120x120 pixels stamps containing the target galaxy in the center. These stamps are sampled from PAUCam images. Our network contains 5 convolutional layers (red layers). Each convolution is followed by a pooling layer (yellow layer). Between each convolution and pooling layer there is a batch normalization layer (blue layer). The numbers on each of the convolutional layer represent the layer’s dimension. The first number corresponds to the number of channels. The second and third numbers are the dimension of the stamp in that layer. In each convolutional layer, the network learns to capture different features in the image. The network gradually picks up features as the input goes deeper through it. Once the stamp has propagated through the CNN, its outcome is linearised to a set of values that should represent the content of the stamp faithfully.

The next stage of BKGNET is the linear neural network. Here, we feed the network with the set of linear values representing the stamp (CNN’s output) together with extra information on the galaxy, i.e. its position in the original image, its magnitude, the NB used to observe the galaxy and a before/after intervention flag informing the network when the galaxy was observed. This information is not spatially related as, for example, the pixels in the image. Therefore, it is more convenient to use a linear neural network rather than a CNN.

There are 80 different combinations of band + intervention flag, each one of them corresponding to a scattered light

³ <https://gitlab.pic.es/pau/bkgnet>

pattern. We use the PYTORCH embedding module to encode each pattern in the ten trainable parameters that best define the pattern. Therefore, the band and intervention information is given to the network in the form of a (80x10) trainable matrix that encodes the scattered light pattern.

3.3 Data: training and test samples

BKGNET's inputs are stamps with the target galaxy in the center. However, to train the network we use empty CCD positions, meaning regions where there are no target sources. This way, we can estimate the ground truth background value at the central CCD region (where there is supposed to be a target galaxy) and train the network to recover this value. The estimation of the true background values used as training sample labels is done by computing the mean background inside a circular aperture of a given fixed radius in the central region of the stamp. Therefore, these measurements have an associated uncertainty that directly depends on the aperture radius. Assuming that the background is purely Poissonian, then

$$\sigma_{\text{label}}^2 = \frac{N_a b}{t_{\text{exp}}}, \quad (3)$$

where t_{exp} is the exposure time, b is the background estimated as the mean of the pixels inside the aperture, i.e. the background label, and N_a is the number of pixels inside the circular aperture, directly related with the choice of aperture radius. Although the radius is a free parameter, we fixed it to 8 pixels. To select empty stamps for the training sample we identify sources by cross-correlating the sky coordinates of a given image location with the sky coordinates of the sources in COSMOS catalogue (Laigle et al. 2016).

In any deep learning algorithm, the training and test samples should be as similar as possible. Our training sample does not contain target galaxies whereas the test does. We therefore add simulated galaxies in the center of the empty training stamps. The simulated galaxies are constructed with parameters based on PAUS data: Sersic profile, $r50$, $I50$ and magnitude in the i -band. The Sersic profile describes the surface brightness profile (I). The radius r that contains 50% of the light intensity ($I50$) is $r50$. These simulated galaxies may differ from the real ones. For this reason, we mask the central 16 x 16 pixels in both training and test samples. Although the simulated galaxy is now masked, it is still important to include it, as for some profiles the galaxy light extends outside the masked region. Without the simulated galaxy, BKGNET fails on testing bright sources.

We normalise the stamp before feeding the network. There are different ways of doing this. We apply a normalisation stamp by stamp, where we use the mean and the standard deviation of each stamp to normalise it. We have chosen this normalisation method as it performs better on our dataset.

We use all the PAUCam images in COSMOS to train and validate the network. We have 4928 PAUCam images

before the intervention and 4821 after (see sec. 3.3 for details). For each of them, we sample around 40 stamps per CCD image, giving a total of around 400,000 stamps. We use 90% of them for training and the remaining 10% for validation.

3.4 Loss function

Supervised deep learning algorithms are trained comparing the true value with the algorithm's prediction. The agreement between the prediction and the true is evaluated with a loss function. The choice of loss function depends on the kind of problem one is facing, (e.g. classification, regression). A typical loss function for classification problems is the cross-entropy loss, whereas in regression problems the mean squared error is commonly used. With BKGnet we want the network to associate an uncertainty to each prediction. In supervised deep learning, there are some methods based on Bayesian statistics that deal with uncertainties associated with the predictions (Kendall & Gal 2017; Kendall et al. 2017).

The method we use assumes that the distribution $p(\mathbf{y}|f^{\mathbf{w}}(\mathbf{x}))$ is Gaussian, where \mathbf{y} are the background label values, \mathbf{x} are the inputs and $f^{\mathbf{w}}(\mathbf{x})$ are the network background predictions. Therefore, the loss function is defined

$$\text{Loss} = -\log p(f^{\mathbf{w}}(\mathbf{x})) = \frac{(f^{\mathbf{w}}(\mathbf{x}) - y)^2}{\sigma^2} + 2 \log \sigma. \quad (4)$$

In this way, we train the network to provide both, the background prediction $f^{\mathbf{w}}(\mathbf{x})$ and σ . Notice that the second term on the right hand side prevents the network from predicting a large error that minimises the first term.

Note that with the loss in Equation 4, the network provides an error on the quantity $f^{\mathbf{w}}(\mathbf{x}) - y$, which has an associated uncertainty $\sigma_{\text{pred}}^2 + \sigma_{\text{label}}^2$. Therefore, the error on the prediction is

$$\sigma_{\text{pred}} = \sqrt{\sigma_{\text{bkgnet}}^2 - \sigma_{\text{label}}^2}, \quad (5)$$

where σ_{bkgnet}^2 is the error provided by the network and σ_{label}^2 is the error of the background label. The error of the background label is defined in Equation (3).

4 TESTING BKGNET ON SIMULATIONS

To test the performance of BKGNET we first apply it to simulated data. This allows us to examine how well we can predict the background with the network and also allows us to explore what data are needed and how this data is treated before feeding the network. Throughout the rest of the paper we compare the BKGNET predictions with those obtained by calculating the background inside an annulus around the target source before, and after correcting the image with the sky flat.

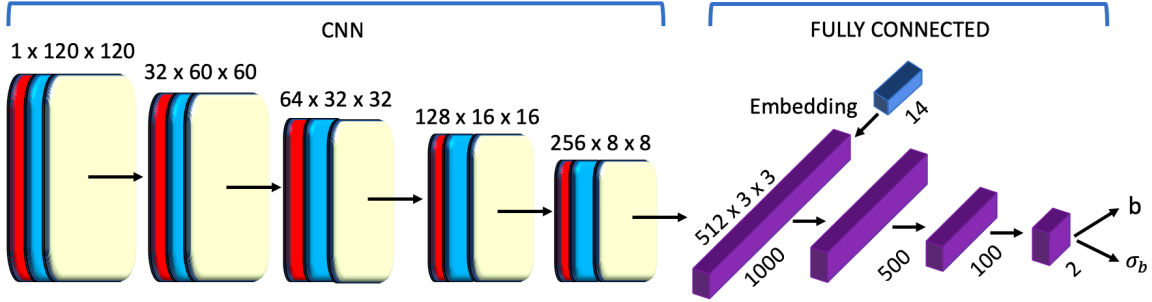


Figure 5. BKGnet scheme: The first set of layers correspond to a Convolutional Neural Network to which one inputs the images. The CNN output is then embedded with extra information: the (x,y) position of the input in the original CCD, the i -band magnitude of the target galaxy and ten trainable numbers encoding information about the band and the intervention flag.

4.1 Simulated PAUCam background images

The simulated PAUCam images are generated using the sky flats as follows.

Select the sky flat: We select the sky flat according to the band that is considered.

Convert to photons: We scale the sky flat with the exposure time in that band, t_{exp} .

Scale the sky flat: We scale the sky flat by a factor A , so that they are normalised to have mean background around unity. To obtain simulated images, we sample scaling factors A from the real distribution of backgrounds in PAUCam images and we use these to scale the sky flat.

Add noise: We add Poisson sky noise.

Back to electrons: We return to the original image units by dividing by the exposure time t_{exp} .

The final simulated image $I_{\text{sim}}(x,y)$ can thus be expressed as

$$I_{\text{sim}}(x,y) = A \cdot \frac{t_{\text{exp}} \cdot \text{skyflat}(x,y) + P(t_{\text{exp}} \cdot \text{skyflat}(x,y))}{t_{\text{exp}}}, \quad (6)$$

where $P(\cdot)$ indicates the realisation of Poisson noise.

4.2 BKGnet predictions on simulations

Throughout this section, we train and test on stamps without target galaxies (empty positions). This allows us to test whether it is possible to predict the background with this network's assembly. We also fix the band we are testing to NB685 after the camera intervention. This choice is a compromise between having a considerable amount of scattered light without being completely dominated by it. Before the intervention, the amount of scattered light in some of the CCD images is very large and might not be an adequate choice to test the network. On the other hand, after the intervention, some of the CCDs barely contain scattered light, and those would not be a good choice either. NB685 contains a considerable amount of scattered light and therefore it is a representative example. We do not need to simulate all bands, as here we only want to test the viability of the scattered light prediction with BKGNET and to have a better understanding of the network's behaviour. To quantify

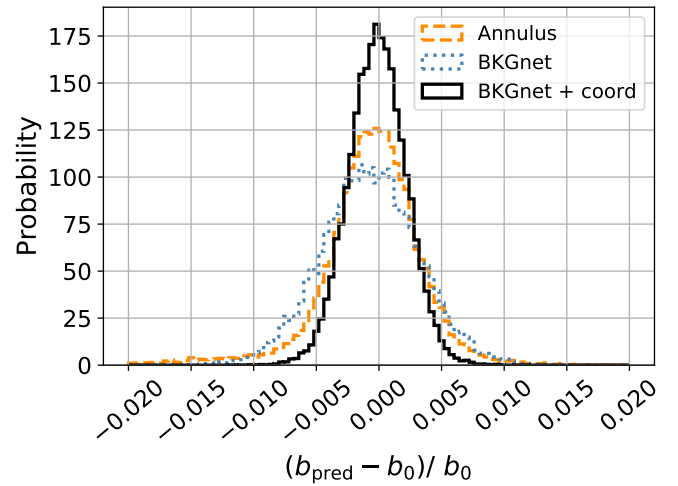


Figure 6. Relative error distributions for the BKGNET (green without coordinate information and orange with coordinate information) and the annulus predictions. b_0 is the background label and b_{pred} is the background prediction, either for the annulus or for BKGNET.

the background prediction accuracy, we use

$$\sigma_{68} \equiv 0.5(b_{\text{quant}}^{84.1} - b_{\text{quant}}^{15.9}), \quad (7)$$

where we use the 84.1 and 15.9 quantiles. This is equivalent to 1σ for a normal distribution, but less affected by outliers.

Figure 6 compares the accuracy with which BKGNET predicts the background to the PAUS default approach that uses an annulus around the object (and computing the median inside plus an additional σ -clipping step). The plot shows the relative error distribution of the predictions for both methods. We have tested BKGNET with and without embedding the image coordinates of the galaxy. The BKGNET performance improves significantly with the coordinate information. This is not surprising, because the amount of scattered light depends on the CCD position (see Section 2). Although scattered light is encoded in the image, the CCD position also includes essential information for the prediction. The network might need it to create something similar to the sky flat. Only with information coming with

the stamps, BKGNET achieves a $\sigma_{68} = 0.0038$. Including the coordinate information, this improves to $\sigma_{68} = 0.0022$. Therefore, the network improves by 70% with the coordinates embedding. The default background estimate shows tails on both sides of the distribution, and yields $\sigma_{68} = 0.0033$, which means BKGNET improves the estimate by 42%.

Figure 7 shows the spatial background map (left) and the relative error on the prediction of this map with the annulus background predictions (right) and the BKGNET background predictions (middle). The precision is lower at the edges for the annulus-based method, where scattered light is present. This indicates that the tails in Figure 6 are caused by scattered light. On the other hand, one can see that BKGNET is able to account for the presence of scattered light.

5 BKGNET ON PAUCAM IMAGES

In the previous section, we have shown that BKGNET is able to accurately predict strongly scattered light backgrounds in simulated blank images. However, in real PAUCam data other complications, such as cosmic rays, electronic cross-talk, read-out noise and dark current may affect the performance. Moreover, correlations between pixels might be introduced during the data reduction process. To examine the impact of these real-life effects we use actual PAUCam images. To assess the accuracy of our measurements, we test network on empty stamps, i.e. without target galaxies.

We will use all the images available in COSMOS, but split the data into those obtained before the camera intervention (in 2016A) and after, yielding 4928 images and 4821 images, respectively. As these numbers are similar, we can easily balance the number of stamps before and after the intervention in our training sample. Although the training sample does not contain target galaxies in the center, sources might be placed in other stamp's positions. To avoid outliers in the training set, e.g. a stamp with a bright star covering most of the background or a bright object too close to the center, we filter the training stamps based on the maximum pixel value. All stamps with a pixel containing more than 100,000 counts are excluded from the training sample.

We also exclude 40 images from each subsample before training the network. These 80 images are not used to train the network, but are kept to test it. This is important, as we need to test the network on images it has never seen before. To generate the test set, instead of sampling randomly from the CCDs, we sample stamps consecutively in intervals of 60 pixels. This ensures that we test all CCD regions, including regions affected by scattered light.

Figure 8 shows the results when we use BKGNET to predict the background on PAUCam images in empty regions. We also show results when the background is estimated using an annulus, and when we first correct the background variations using a sky flat ('annulus + sky'). Figure 8 shows the value of σ_{68} (Equation (7))

	BEFORE		AFTER	
	filtered	sources	filtered	sources
Annulus	0.011	0.011	0.014	0.014
+ sky flat	0.011	0.011	0.011	0.013
BKGnet	0.008	0.008	0.011	0.011

Table 1. Average σ_{68} of the relative error in the background prediction across all the bands for BKGNET trained before and after the camera intervention. We list the results for the data sets without filtering out stamps affected by sources ('sources'), and if we remove these ('filtered').

of the relative error distribution on the prediction for the 40 different bands. Because we are using the relative error, the comparison between the results before and after the intervention is not representative, as the background levels are different. For instance, in the first filter tray (NB455-NB515), the background before the intervention is between 3 and 5 times higher than after.

We focus first on the results before the camera intervention (left panel in Figure 8). Images before the camera intervention contain more scattered light than those after (see Fig. 2). This makes the sky flat modelling more unstable than the modeling of images after the intervention. We find that correcting with the sky flat does not improve the annulus result in every band. In the bluest NBs, i.e. those with the highest amount of scattered light, the sky flat seems to decrease the accuracy of the background prediction. On the other hand, BKGNET improves the accuracy compared to the other two methods, especially on the bluest filter tray. On average considering all bands, the network reduces the σ_{68} by a 37% compared with the sky flat and up to a 50% if we only consider the 8 bluest NBs.

If we consider the results after the camera intervention (right panel of Figure 8), we see that the sky flat improves the annulus prediction in all the bands. This is expected from the top panel in Figure 2, which shows that scattered light trends are stable after the camera intervention. Before the intervention the sky flat fails in the bluer bands, which no longer happens after the camera intervention. Nevertheless, BKGNET performs even better: on average, after the intervention it achieves an 18% improvement compared to the sky flat correction.

Table 1 lists the average value of σ_{68} of the relative error in the background prediction for the three methods: annulus, annulus + sky flat and BKGNET. When training and testing, we first exclude the stamps with a maximum pixel value above 100,000. By doing this, we avoid stamps with very bright nearby sources that might bias the prediction. To examine the impact of this step, we list also the results when contaminated stamps are included ('sources') in Table 1. These results show that the filtering does not make a difference before the intervention, but the performance improves somewhat for the correction that uses the skyflat. The small difference suggests that scattered light is the main source of bias. For the images without bright sources taken after the intervention, BKGNET and the sky flat give the same σ_{68} . Therefore, it is possible that

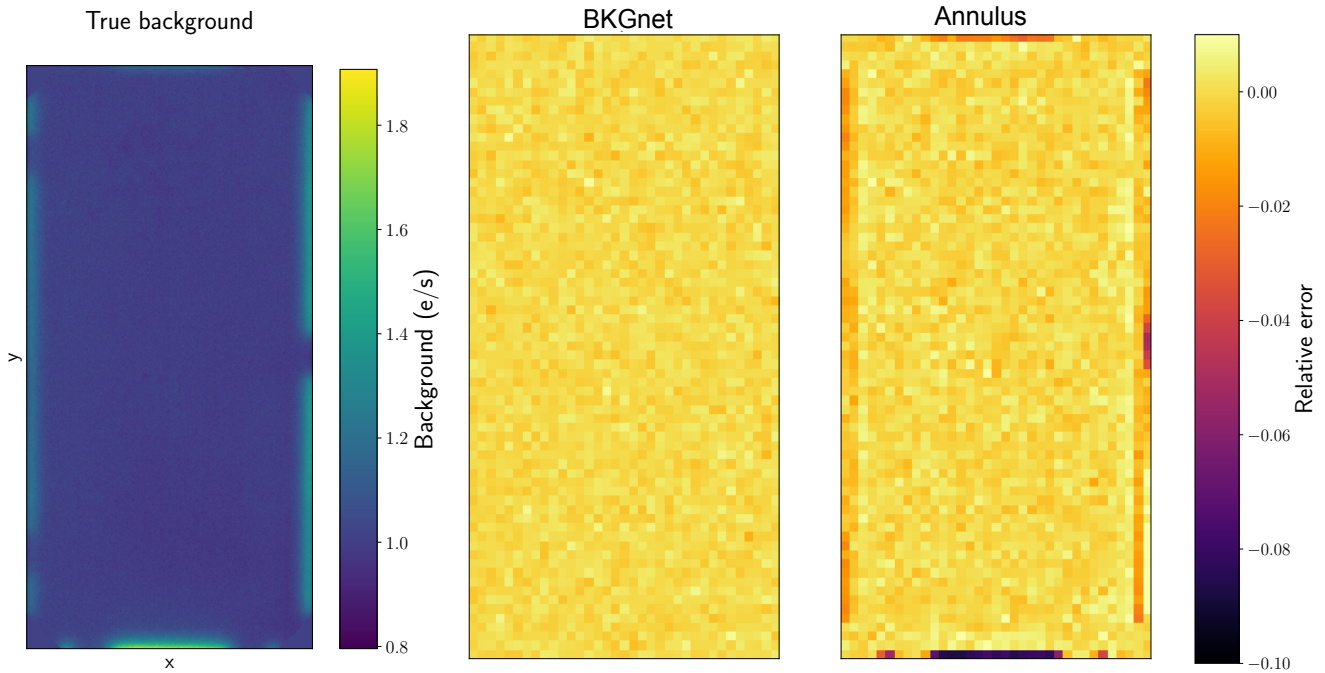


Figure 7. *Left:* CCD reconstruction with the true background values used to train the network. We sample these background values consecutively and we reconstruct the original image by placing each value in the position it was sample from. *Middle:* Accuracy on the background prediction with BKGNET in the different image positions. We can see there are no spatial patterns. *Right:* Accuracy on the background prediction with the annulus in the different image positions. We can see there are no spatial patterns.

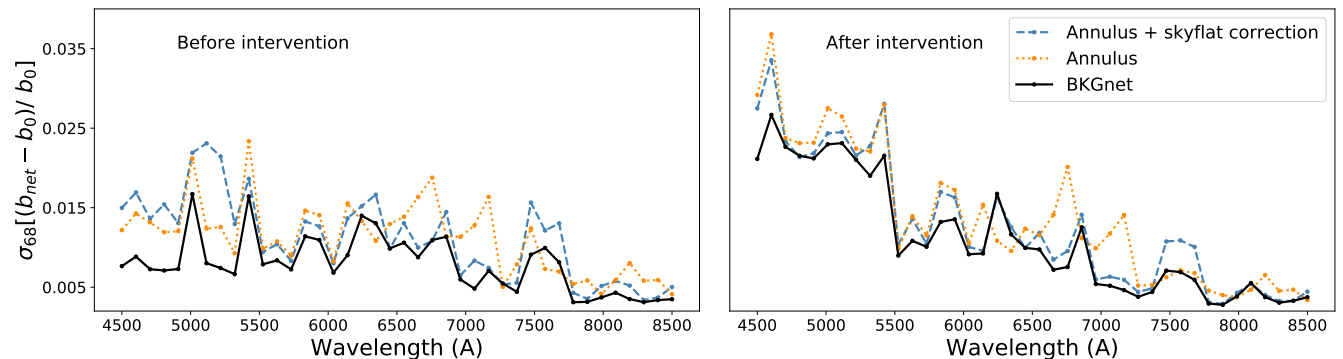


Figure 8. σ_{68} of the relative error in the background prediction for the 40 NBs. *Left:* Before the intervention. *Right:* After the camera intervention. In almost all cases BKGNET performs better than the default approach that employs an annulus to estimate the background.

BKGNET learns the underlying behaviour of scattered light in a similar way as the sky flat. However, as the network also sees the stamp, the correction it infers is more flexible than applying an sky flat. This indicates that BKGNET is able to learn how to estimate the background in the presence of other artifacts (e.g. sources or cosmic rays).

BKGNET also provides an estimate for the uncertainty associated with the background prediction. To test the accuracy of this estimate we use the empty stamps and study the distribution of $(b_{\text{net}} - b_{\text{true}})/\sigma$, where $(b_{\text{net}}$ and σ are the network predictions. If the errors are correct, this distribution should be a Gaussian with zero mean and unit variance.

Figure 9 shows the theoretical Gaussian we should re-

cover and the measured distributions for the annulus and BKGNET predictions. The BKGNET results fit the theoretical Gaussian, which means that our errors are robust. In contrast, the annulus predictions underestimate the uncertainties by 47%. Therefore, BKGNET provides a more reliable estimate of the uncertainty in the background determination.

6 BKGNET VALIDATION

The results presented in section 5 show that, compared to the annulus-based methods, BKGNET yields better background estimates (see Fig 8 and Tab 1), while also providing accurate estimates for the associated uncertainty.

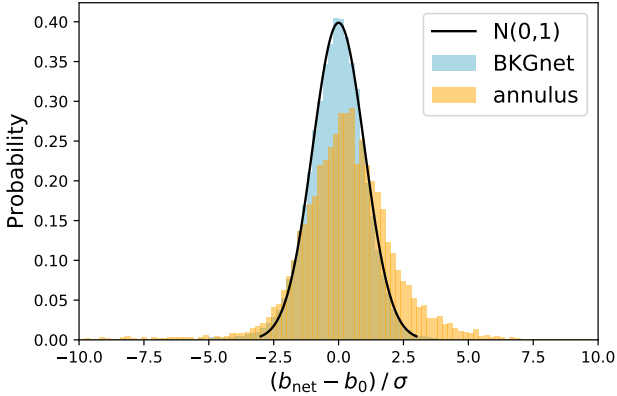


Figure 9. The distribution of $(b_{\text{net}} - b_0)/\sigma$, where b_{net} is the background prediction and b_0 are the true backgrounds. σ is the uncertainty in the prediction. We expect the distribution to be a Gaussian centered on zero with unit variance. We show the distribution for the annulus (orange) and for BKGNET (blue).

However, these tests were done on stamps without galaxies. Here we increase the realism of the problem and quantify the performance of BKGNET at galaxy positions.

6.1 Generating the PAUS catalogue with BKGnet predictions

We use BKGNET to estimate the background for galaxies in the COSMOS field. We compare the results to those from the PAUDM catalogue, which uses an annulus to determine the background. These catalogues contain around 12 million flux measurements, approximately half before and half after the intervention. To obtain the galaxy fluxes we need to subtract the background from the PAUS raw signal measurements,

$$F = S - N_a b, \quad (8)$$

where F is the net galaxy flux, S is the total signal measured inside the aperture, N_a is the number of pixels inside the aperture and b is the predicted background per pixel. When the background is estimated with an annulus, the error on the net flux is

$$\sigma^2 = (S - b) + N_a \sigma_b^2 + N_a^2 \left(\frac{\pi}{2}\right) \frac{\sigma_b^2}{N_b}, \quad (9)$$

where b and σ_b are the background and the background error in that region and N_b is the number of pixels inside the annulus. The $\pi/2$ factor arises from that fact that we use the median of the pixels inside the annulus instead of the mean⁴.

For BKGNET the error on the galaxy flux is

$$\sigma^2 = (S - b) + N_a (b + RN^2) + N_a^2 \sigma_b^2, \quad (10)$$

where RN is the read-out noise.

Equations 9 and 10 reflect the differences in the flux

⁴ <http://wise2.ipac.caltech.edu/staff/fmasci/ApPhotUncert.pdf>

uncertainty when the background is measured with an annulus or with BKGNET. In general, there are three main contributors to the flux uncertainty: the uncertainty in the net galaxy flux, the uncertainty in the background estimate, and the uncertainty introduced by the background subtraction. For both background estimation methods we assume that the uncertainty in the net galaxy flux is captured by shot noise. For BKGNET, the background uncertainty is also described by shot noise (Eq. 10), but we add a read-out noise contribution to the background error. For the PAUDM measurements, the background uncertainty is given by the mean variance per pixel (Eq. 9). Therefore, for PAUDM, this term would also account for other error contributions besides shot noise. The third terms in Eqs. 9 and 10 are the contributions from background subtraction uncertainties. In PAUDM, this is determined by the subtraction of a background measured in the annulus. In contrast, in Equation 10 we use the uncertainty provided by the network within the aperture where the flux is estimated.

6.2 Validating the catalogs

For a flat background using an annulus to estimate the background is viable method. Scattered light only affects objects near the edges of the image. Hence, for most of the galaxies in PAUS data the background should be (approximately) flat and we should not expect large differences between the BKGNET and the PAUDM catalogues. Comparing the fluxes estimated with Equation 8, we find a 2% difference between the two approaches. On the other hand, the uncertainties estimated with BKGNET (Eq. 10) are 4% lower than for PAUDM.

We need to determine which catalogue provides better photometry estimates. To do so, we use the fact that PAU-Cam takes multiple observations of the same object in all NB filters. We can compare different exposures of the same object, which should be comparable once the background noise is subtracted. This is formulated as

$$D \equiv \frac{(e_1 - e_2)}{\sqrt{(\sigma_1^2 + \sigma_2^2)}}, \quad (11)$$

where e_i are different exposures of the same object and σ_i the associated uncertainties. The distribution of D should be a Gaussian with unit variance if the photometry is robust. We call this the duplicates test.

Figure 10 shows the results of the duplicates test as a function of wavelength. We estimate $\sigma_{68}[D]$ (Eq. 10) for each NB with the BKGNET (black line) and PAUDM (orange line) catalogues. It is possible to flag photometric outliers based on an ellipticity parameter to detect strongly varying backgrounds. The dashed lines in Figure 10 show the results when we exclude such flagged objects. The difference is small for BKGNET, but we see a clear improvement for the PAUDM measurements. The improvement is particularly prominent for NB755 (at 7500Å), which is affected by telluric absorption of O₂ in the atmosphere. In principle, the calibration should account for this. We also note that the affected objects are flagged. Interestingly BKGNET seems to know how to deal with these objects. This shows that BKGNET is more

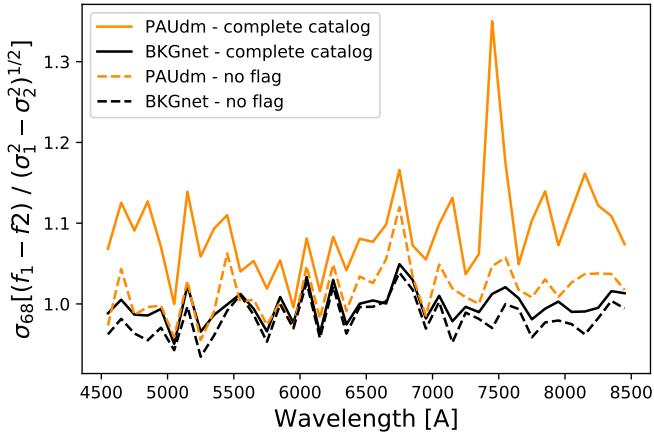


Figure 10. BKGNET validation with the duplicates distribution test. We plot the width of the distribution defined in Equation (11) as a function of wavelength for the catalogue generated with BKGNET (black line) and the current PAUDM catalogue (orange line). The dashed line corresponds to the results excluding all objects flagged in PAUDM. The solid line includes all objects.

robust towards various sources of bias, not only scattered light. When we consider all NBs, we find $\langle \sigma_{68}[D] \rangle = 1.00$ for BKGNET, which is what we would expect for correct photometry. On the other hand, the current PAUDM catalogue yields $\langle \sigma_{68}[D] \rangle = 1.10$, i.e. it overestimates the uncertainties.

The measurement uncertainties should depend on the brightness of the source. To explore this we show $\sigma_{68}[D]$ as a function of Subaru i_{Auto} magnitude in Figure 11. In the PAUDM catalogue there is a strong trend with magnitude. At the bright end, the fluxes differ by more than 20% compared to the expectation. This trend disappears when we predict the background and uncertainties with BKGNET. To explore the origin of the trend further we used the background prediction from BKGNET but the errors from the annulus. As the blue dotted line in Figure 11 show, we find the same trend with magnitude. This implies that it is caused by the estimated uncertainties for the annulus. Moreover, the blue dotted line lies below the PAUDM line. The only difference between these two curves is the background value prediction (not the error). Therefore, the predictions with PAUDM are more accurate than those with the annulus.

To further validate the BKGNET catalogue we run BCNZ2 (Eriksen et al. 2019) using the fluxes determined using BKGNET. For this test, we exclude the objects flagged in the PAUDM catalogue, in order to use exactly the same objects as (Eriksen et al. 2019). However, as shown in Figures 10 and 11, we do not need to exclude these objects. The photo- z s are compared to secure spectroscopic estimates from zCOSMOS DR3 (Lilly et al. 2007) with $i_{\text{AB}} < 22.5$. We split the sample based on a quality parameter defined as:

$$Q_z \equiv \frac{\chi^2}{N_f - 3} \left(\frac{z_{\text{quant}}^{99} - z_{\text{quant}}^1}{\text{ODDS}(\Delta z = 0.01)} \right), \quad (12)$$

where $\chi^2/(n_f - 1)$ is the reduced chi-squared from the tem-

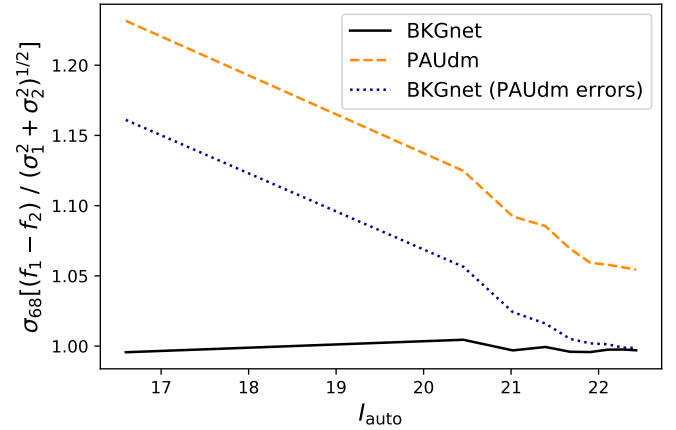


Figure 11. BKGNET validation with the duplicates distribution test. We plot the width of the distribution defined in Equation (11) as a function of i_{Auto} in the Subaru i band for the catalogue generated with BKGNET (black solid line), the current PAUDM catalogue (orange dashed line) and a mixed catalogue with the predictions from BKGNET and the errors from PAUDM (blue dotted line).

plate fit, the z_{quant} are the percentiles of $(z_{\text{photo}} - z_{\text{spec}})/(1 + z_{\text{spec}})$. The ODDS is defined as

$$\text{ODDS} \equiv \int_{z_b - \Delta z}^{z_b + \Delta z} dz p(z), \quad (13)$$

where z_b is the peak in $p(z)$ and Δz defines a redshift interval around the peak. In PAUS, a galaxy is considered an outlier if

$$|z_{\text{photo}} - z_{\text{spec}}| / (1 + z_{\text{spec}}) > 0.02. \quad (14)$$

Notice that this outlier definition is much more strict than in other broad-band photometric surveys.

Table 2 lists the outlier rate and the photometric redshift precision obtained with BCNZ2 for the two catalogues. To quantify the redshift precision we use σ_{68} (Eq. (7)). We find that the photometric redshift precision does not improve significantly between the two catalogues, but we find an reduction in the outlier rate. If we consider the complete sample (100%) this improvement is small. This might be because in the full sample the outliers are dominated by photo- z outliers, rather than outliers on the photometry itself. However, if we cut using the Q_z parameter to get the best 20% and 50% of the sample, we notice that the outlier rate reduces significantly. These should be dominated by photometry outliers. For the best 50% of the sample we reduce the number of outliers by 25%, whereas for the best 20% of objects this improvement rises to a 35%. This shows once more that BKGNET is a statistically accurate method that is also robust.

7 CONCLUSIONS

We have presented a new method to predict the background for astronomical images that are affected by scattered light. The algorithm has been developed to predict the

Percentage	Outlier percentage		$10^3 \sigma_{68}$	
	BKGNET	PAUDM	BKGNET	PAUDM
20	3.5	5.4	2.0	2.1
50	3.8	5.1	3.6	3.7
80	10.4	11.3	5.8	6.0
100	16.7	17.5	8.4	8.6

Table 2. Photo- z outlier rate and accuracy obtained with BCNz2 for the BKGNET and the PAUDM catalogues. The percentages correspond to the samples selected by the photo- z quality parameters Qz .

background on images taken with PAUCam. The edges of PAUCam images are affected by scattered (see Fig. 1), especially in the bluer bands. In 2016, the camera was modified to reduce the amount of scattered light. While the amount of scattered light decreased drastically, PAUCam images still contain a significant amount of scattered light (see Fig. 2).

For each band, the scattered light follows the same spatial pattern within the CCD and scales approximately linear with the background level. We have constructed sky flats and background pixel maps by combining images taken with the same NB and normalised by their background level. These sky flats show the scattered light variation across the CCD and can be used to correct for scattered light (see Fig. 3). Nevertheless, background fluctuations due to external conditions (e.g. moon, seeing, airmass) can trigger differences on scattered light from night to night. To accurately correct scattered light with sky flats, we would need to generate a sky flat per NB and night. However, even then fluctuations during the night or a small number of available images in a given band can lead to inaccurate corrections.

We therefore developed BKGNET, a deep learning based algorithm that predicts the background and its associated uncertainty behind target sources accounting for scattered light and other distorting effect. BKGNET consists of a Convolutional Neural Network followed by a linear neural network (see Fig. 5). In the training set we use empty stamps, i.e. without a target galaxy, so that we can estimate the true background and use it for training. We need to simulate target galaxies in the training sample before masking the central region, otherwise the network fails when applied to bright and large sources.

The second stage of BKGNET is a linear network. Its input is the output of the CNN, together with the embedding of the target source position on the image together with the NB and a flag whether the data were taken before or after the camera intervention. These two last quantities are included as a (80x10) matrix. Here 80 is the number of band + intervention flag combinations and each of these combinations has ten associated trainable parameters. The network learns to find ten parameters to define each scattered light pattern.

We first tested the predictions on PAUCam empty stamps, i.e. without target galaxies. For data taken before

the intervention, BKGNET improves over the sky flat + annulus prediction by 37%. The sky flat correction fails in many of the bands, specially on the bluer filter tray, which is affected the most by scattered light (left panel of Fig. 8). For data taken after the intervention, BKGNET improves over the sky flat + annulus prediction by 17% (right panel of Fig. 8).

BKGNET also predicts the uncertainty associated with the background prediction. For that, we use the log likelihood of a Gaussian centered at the background true value as loss function (Eq. 4). To validate BKGNET, we test on empty positions and estimate the difference between the prediction and the background label, divided by the estimated uncertainty. For the annulus, we find that the errors are underestimated by 47% (Fig 9). On the other hand, with BKGNET this quantity is normally distributed around zero with unit variance, showing that the uncertainties are correctly estimated.

We generated a PAUS catalogue for the COSMOS field using BKGNET to predict the background. To validate the catalogue we took advantage of having multiple measurements of the same object. The resulting distribution of differences in flux measurements should be a Gaussian of unit variance (Eq. 11). The results demonstrate that BKGNET improves the photometry with respect to the current background subtraction algorithm. We test the performance for the full catalogue and a catalogue where we exclude all objects flagged in the current catalogue version. When excluding flagged objects, we find very similar results with BKGNET catalogue and the current catalogue. However, when testing the full catalogue, we find a large improvement for BKGNET. It specially improves the results in a region with high atmospheric absorption, demonstrating that it is more robust against sources of bias while still being statistically accurate. It also removes a strong systematic trend with i -band magnitude, that disappears when the uncertainties are estimated with the network.

Finally, as the aim of PAUS is to provide accurate redshifts for large samples of galaxies, we have run the BCNz2 code using the BKGNET catalogue. BKGNET reduces the outlier rate by a 25% and 35% respectively for the best 50% and 20% photo- z samples, while the accuracy is not affected. Our results provide the first building block of an end-to-end pipeline to analyze photometric images. The complete pipeline would subtract the background, predict the flux and measure the photometric redshift. This first step focuses on the background to understand the background noise in PAUCam images, and more concretely the behaviour of scattered light.

ACKNOWLEDGEMENT

Funding for PAUS has been provided by Durham University (via the ERC StG DEGAS-259586), ETH Zurich, Leiden University (via ERC StG ADULT-279396 and Netherlands Organisation for Scientific Research (NWO) Vici grant 639.043.512) and University College London. The PAUS participants from Spanish institutions are partially supported

by MINECO under grants CSD2007-00060, AYA2015-71825, ESP2015-88861, FPA2015-68048, SEV-2016-0588, SEV-2016-0597, and MDM-2015-0509, some of which include ERDF funds from the European Union. IEEC and IFAE are partially funded by the CERCA program of the Generalitat de Catalunya. The PAU data center is hosted by the Port d'Informació Científica (PIC), maintained through a collaboration of CIEMAT and IFAE, with additional support from Universitat Autònoma de Barcelona and ERDF. CosmoHub has been developed by PIC and was partially funded by the "Plan Estatal de Investigación Científica y Técnica y de Innovación" program of the Spanish government. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan V GPU used for this research. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 776247. AA is supported by a Royal Society Wolfson Fellowship.

REFERENCES

- Abbott T. M. C., et al., 2018, *ApJS*, **239**, 18
- Alexander S., Gleyzer S., McDonough E., Toomey M. W., Usai E., 2019, arXiv e-prints, p. [arXiv:1909.07346](https://arxiv.org/abs/1909.07346)
- Bertin E., Arnouts S., 1996, *Astronomy and Astrophysics Supplement Series*, **117**, 393
- Bijaoui A., 1980, *A&A*, **84**, 81
- Blanton M. R., et al., 2017, *AJ*, **154**, 28
- Cabayol L., et al., 2019, *MNRAS*, **483**, 529
- Carrasco-Davis R., et al., 2018, arXiv e-prints, p. [arXiv:1807.03869](https://arxiv.org/abs/1807.03869)
- Casas R., et al., 2012, in High Energy, Optical, and Infrared Detectors for Astronomy V. p. 845326, doi:10.1117/12.924640
- Casas R., et al., 2016, in Ground-based and Airborne Instrumentation for Astronomy VI. p. 99084K, doi:10.1117/12.2232422
- Castander F. J., et al., 2012, in Ground-based and Airborne Instrumentation for Astronomy IV. p. 84466D, doi:10.1117/12.926234
- Castander F., Eriksen M., Serrano S., et al. in prep.
- Donahue J., Hendricks L. A., Rohrbach M., Venugopalan S., Guadarrama S., Saenko K., Darrell T., 2017, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39**, 677
- Dong C., Loy C. C., He K., Tang X., 2016, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**, 295
- Eriksen M., et al., 2019, *MNRAS*, **484**, 4200
- Fluri J., Kacprzak T., Refregier A., Amara A., Lucchi A., Hofmann T., 2018a, *Phys. Rev. D*, **98**, 123518
- Fluri J., Kacprzak T., Refregier A., Amara A., Lucchi A., Hofmann T., 2018b, *Phys. Rev. D*, **98**, 123518
- George D., Huerta E. A., 2018, *Physics Letters B*, **778**, 64
- Herbel J., Kacprzak T., Amara A., Refregier A., Lucchi A., 2018, *J. Cosmology Astropart. Phys.*, **2018**, 054
- Ivezić Ž., et al., 2019, *ApJ*, **873**, 111
- Kendall A., Gal Y., 2017, preprint(arXiv:1703.04977), p. [arXiv:1703.04977](https://arxiv.org/abs/1703.04977)
- Kendall A., Gal Y., Cipolla R., 2017, preprint(arXiv:1705.07115), p. [arXiv:1705.07115](https://arxiv.org/abs/1705.07115)
- Krizhevsky A., Sutskever I., Hinton G. E., 2012, in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. NIPS'12. Curran Associates Inc., USA, pp 1097–1105, <http://dl.acm.org/citation.cfm?id=2999134.2999257>
- Laigle C., et al., 2016, *ApJS*, **224**, 24
- Laureijs R., et al., 2011, arXiv e-prints, p. [arXiv:1110.3193](https://arxiv.org/abs/1110.3193)
- LeCun Y., Boser B., Denker J. S., Henderson D., Howard R. E., Hubbard W., Jackel L. D., 1989, *Neural Computation*, **1**, 541
- Lecun Y., Bottou L., Bengio Y., Haffner P., 1998, *Proceedings of the IEEE*, **86**, 2278
- Lilly S. J., et al., 2007, *ApJS*, **172**, 70
- Martí P., Miquel R., Castander F. J., Gaztañaga E., Eriksen M., Sánchez C., 2014, *MNRAS*, **442**, 92
- Newell E. B., 1983, in Astronomical Measuring Machines Workshop. pp 15–40
- Newman J. A., et al., 2013, *The Astrophysical Journal Supplement Series*, **208**, 5
- Padilla C., et al., 2016, in Ground-based and Airborne Instrumentation for Astronomy VI. p. 99080Z, doi:10.1117/12.2231884
- Padilla C., et al., 2019, *AJ*, **157**, 246
- Paszke A., et al., 2017
- Popowicz A., Smolka B., 2015, *MNRAS*, **452**, 809
- Romanishin W., 2014, An Introduction to Astronomical Photometry Using Ccds. Createspace Independent Pub, <https://books.google.es/books?id=0nbMoQEACAAJ>
- Serrano S., Castander F., Fernandez E., et al. in prep.
- Stetson P. B., 1987, *Publications of the Astronomical Society of the Pacific*, **99**, 191
- Stoherth L., et al., 2018, *MNRAS*, **481**, 4221
- Teeninga P., Moschini U., Trager S. C., Wilkinson M. H. F., 2015, in 2015 IEEE International Conference on Image Processing (ICIP). pp 1046–1050, doi:10.1109/ICIP.2015.7350959
- Tonello N., et al., 2019, *Astronomy and Computing*, **27**, 171
- Tortorelli L., et al., 2018, preprint(ArXiv:1805.05340),
- Vafaei Sadr A., Vos E. E., Bassett B. A., Hosnie Z., Oozeer N., Lochner M., 2019, *MNRAS*, **484**, 2793
- Voulodimos A., Doulamis N., Doulamis A., Protopapadakis E., 2018, *Computational Intelligence and Neuroscience*, **2018**, 1
- Werbos P. J., 1982, in Drenick R. F., Kozin F., eds, System Modeling and Optimization. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 762–770
- Xie Y., Le L., Zhou Y., Raghavan V. V., 2018, in Gudivada V. N., Rao C., eds, Handbook of Statistics, Vol. 38, Computational Analysis and Understanding of Natural Languages: Principles, Methods and Applications. Elsevier, pp 317 – 328, doi:https://doi.org/10.1016/bs.host.2018.05.001, <http://www.sciencedirect.com/science/article/pii/S0169716118300026>
- Xu B., Wang N., Chen T., Li M., 2015, arXiv e-prints, p. [arXiv:1505.00853](https://arxiv.org/abs/1505.00853)
- Yue-Hei Ng J., Hausknecht M., Vijayanarasimhan S., Vinyals O., Monga R., Toderici G., 2015, Cornell Univ. Lab.
- Zeiler M. D., Fergus R., 2013, arXiv e-prints, p. [arXiv:1311.2901](https://arxiv.org/abs/1311.2901)
- Zhang K., Bloom J., 2019, *The Journal of Open Source Software*, **4**, 1651

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.