

Effective Image Differencing with ConvNets for Real-time Transient Hunting

Nima Sedaghat¹^{*} and Ashish Mahabal²[†]

¹*Department of Computer Science, University of Freiburg, Georges-Koehler-Allee 052, 79110 Freiburg, Germany*

²*Center for Data Driven Discovery, Caltech, 1200 E California Blvd., Pasadena, CA 91125*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT

Large sky surveys are increasingly relying on image subtraction pipelines for real-time (and archival) transient detection. In this process one has to contend with varying PSF, small brightness variations in many sources, as well as artifacts resulting from saturated stars, and, in general, matching errors. Very often the differencing is done with a reference image that is deeper than individual images and the attendant difference in noise characteristics can also lead to artifacts. We present here a deep-learning approach to transient detection that encapsulates all the steps of a traditional image subtraction pipeline – image registration, background subtraction, noise removal, psf matching, and subtraction – into a single real-time convolutional network. Once trained the method works lighteningly fast, and given that it does multiple steps at one go, the advantages for multi-CCD, fast surveys like ZTF and LSST are obvious.

Key words: Transient — Supernova — Deep Learning — Artificial Intelligence — Convolutional Network – ConvNet

1 INTRODUCTION

Time-domain studies in optical astronomy have grown rapidly over the last decade with surveys like ASAS-SN (Pojmański 2014), CRTS (Mahabal et al. 2011; Djorgovski et al. 2011; Drake et al. 2009), Gaia (Gaia Collaboration et al. 2016), Palomar-Quest (Djorgovski et al. 2008), Pan-STARRS (Chambers et al. 2016), PTF (Law et al. 2009) etc. to name a few. With bigger surveys like ZTF (Bellm 2014) and LSST (Ivezic et al. 2008) around the corner, there is even more interest in the field. Besides making available vast sets of objects at different cadences for archival studies, these surveys, combined with fast processing and rapid follow-up capabilities, have opened the doors to an improved understanding of sources that brighten and fade rapidly. The real-time identification of such sources - called transients - is, in fact, one of the main motivation of such surveys. Examples of transients include extragalactic sources such as the supernovae, and flaring M-dwarf stars within our own Galaxy, to name just two types. The main hurdle is identifying all such varying sources quickly (completeness), and without artifacts (contamination). The identification process is typically done by comparing the latest image (hereafter called the science image), with an older image of the same area of the sky (hereafter called the reference image). The

reference image is often deeper so that fainter sources are not mistaken as transients in the science image. Some surveys like CRTS convert the images to a catalog of objects using source extraction software (Bertin & Arnouts 1996), and use the catalogs as their discovery domain, comparing brightness of objects detected in the science and reference images. Other surveys like PTF directly difference the reference and science images after proper scaling and look for transients in the difference images.

The reference and science images differ in many ways: (1) changes in the atmosphere mean the way light scatters is different at different times. This is characterized by the point spread function (PSF), (2) the brightness of the sky changes depending on the phase and proximity to the moon, (3) the condition of the sky can be different (e.g. very light cirrus), and (4) the noise and depth (detection limit for faintest sources) are typically different for the two images. As a result, image differencing is non-trivial, and along with real transients come through a large number of artifacts per transient. Eliminating these artifacts has been a bottleneck for past surveys, with humans having been often employed to remove them one by one – a process called *scanning* – in order to shortlist a set of genuine objects for follow-up using the scarce resources available. Here we present an algorithm based on deep learning that almost completely eliminates artifacts, and is nearly complete (or can be made so) in terms of real objects that it finds. In Sec. 2 we describe prior art for image differencing, and on deep learning in astron-

* E-mail: nima@cs.uni-freiburg.de (NS)

† E-mail: aam@astro.caltech.edu (AAM)

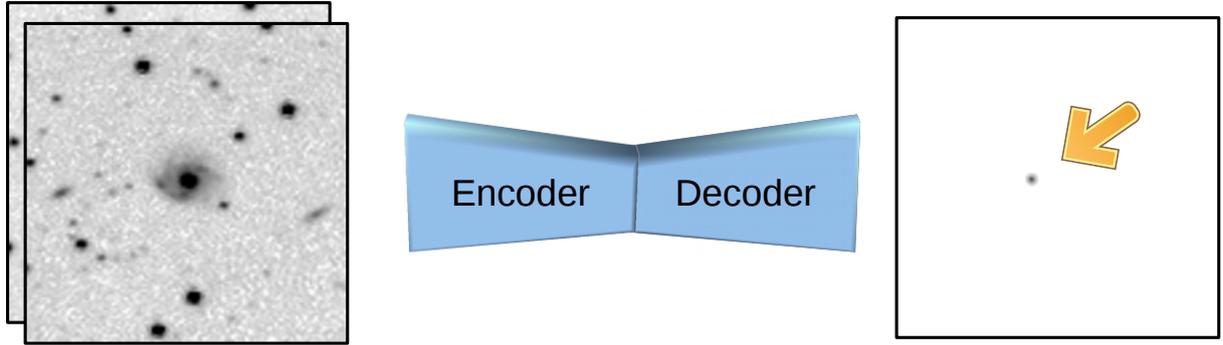


Figure 1. Our CNN-based encoder-decoder network, TransiNet, produces a difference image without an actual subtraction. It does so through training using a labeled set of transients as the ground-truth.

omy. In Sec. 3 we describe the image differencing problem in greater detail, in Sec. 4 we present our method and a generative encoder-decoder network – called *TransiNet* hereafter – based on convolutional networks (ConvNets or CNNs), in Sec. 5 we detail the experiments we have carried out, and in Sec. 7 we discuss future directions.

2 RELATED WORK

For image differencing some of the programs that have been used include Alard & Lupton (1998), Bramich (2008), and PTFIDE (Masci et al. 2017). A recent addition to the list is ZOGY (Zackay et al. 2016) which apparently has lower contamination by more than an order of magnitude. It is to be used with the ZTF pipeline and at least in parts of the LSST pipeline. The main task of such an algorithm is to identify new point sources (convolved by the PSF). The problem continues to be challenging because it has to take in to consideration many complicating factors. Besides maximizing real sources found (true positives), generating as clean an image as possible (fewest false positives) is the quantifiable goal. Please refer to Zackay et al. (2016) for greater detail.

Neural networks in their traditional form have been around since as early as 1980s e.g. Rumelhart & Hinton (1986) and LeCun (1985). Such classical architectures have been used in astronomical applications in the past. One famous example is the star-galaxy classifier embedded into the SExtractor package (Bertin & Arnouts 1996).

The advent of convolutional neural networks (ConvNets: LeCun et al. (1990, 1998)), followed by the advances in parallel computing hardware (Raina et al. 2009), has started a new era in ‘deep’ convolutional networks, specifically in the areas of image processing and computer vision. The applications span from pixel-level tasks such as de-noising to higher-level tasks such as detecting and recognizing multiple objects in a frame. See, e.g. Krizhevsky et al. (2012) and Simonyan & Zisserman (2014).

Researchers in the area of astrophysics have also very recently started to utilize deep learning-based methods to tackle astronomical problems. Deep Learning has already been used for galaxy classification (Hoyle 2016), supernova

classification (Cabrera-Vives et al. 2016), light curve classification (Mahabal et al. 2017; Charnock & Moss 2017), identifying bars in galaxies (Abraham 2017), separating Near Earth Asteroids from artifacts in images (Bue 2017), transient selection post image differencing (Morii et al. 2016), Gravitational Wave transient classification (Mukund et al. 2017), and even classifying noise characteristics (Zevin 2017; Abbott 2017; George et al. 2017).

One aspect of ConvNets that has not received enough attention in the astrophysical research community, is the ability to generate images as output (Rezende et al. 2014; Bengio et al. 2013). We provide here such a generative model to tackle the problem of contamination in difference images (see Fig. 1) and thereby simplify the transient follow-up process.

3 PROBLEM FORMULATION

We cast the transient detection problem as an image generation task. In this approach, the input is composed of a pair of images (generally with different depth, and seeing aka FWHM of the PSF) and the output is an image containing, ideally, only the transient at its correct location and with a proper estimation of the difference in magnitudes. In this work we define a transient as a point source appearing in the second/science input image, and not present in the first/reference image. Such a generative solution as we propose naturally has at its heart registration, noise-removal, sky subtraction, and PSF-matching.

In the computer vision literature, this resembles a segmentation task, where one assigns a label to each pixel of an image, e.g. transient vs. non-transient. However our detections include information about the magnitude of the transients and the PSF they are convolved with, in addition to their shape and location. Therefore the pixel-values of the output are real-valued (or are in the same space as the inputs, making it a different problem than a simple segmentation) (see Fig. 2). To this end, we introduce an approach that is based on deep-learning, and train a convolutional neural network (ConvNet) to generate the expected output based on the input image pair.

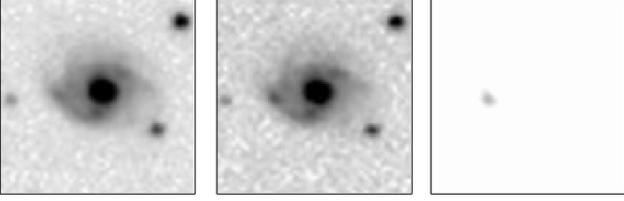


Figure 2. Examples of the reference (left) and science (center) images. The image on the right is the ground truth output defined for this image pair. It contains the image of a single transient, completely devoid of background and noise. The profile of the transient is the best match to reality our model can produce.

We formulate the problem as follows. Let us consider (I_1, I_2) as the reference-science pair:

$$I_1 = I_0 * \phi_1 + S_1 + N_1 \quad (1)$$

$$I_2 = (I_0 + I_t) * \phi_2 + S_2 + N_2 \quad (2)$$

where I_0 is the underlying unconvolved image of the specific region of the sky; ϕ_1 and ϕ_2 represent the PSF models; S_1 and S_2 are the sky levels and N_1 , N_2 represent the noise. Note that for the sake of readability, we have illustrated the effect of noise as a simple addition operation. However in reality the noise is ‘applied’ per pixel throughout the workflow.

I_t is the ideal model for the transient, and can be seen as an empty image with an ideal point-source on it. Based on our formulation of the problem, the answer we seek is $I_t * \phi_2$, which represents the image containing the transient, in the same seeing conditions as the science image. This involves PSF matching for taking the first image from $I_0 * \phi_1$ to I_0 and then to $I_0 * \phi_2$, for the subtraction to work.

Note that in eqs. (1) and (2), for the sake of clarity, the two images are assumed to be registered. In the real problem that the network is trying to solve, 2 is replaced by:

$$I_2 = D\{(I_0 + I_t) * \phi_2\} + S_2 + N_2 \quad (3)$$

in which $D\{\}$ represents spatial inconsistency, which in its simplest form consists of one or more of small rotation, translation and scaling.

4 METHOD

We tackle the problem using a deep-learning method, in which an encoder-decoder convolutional neural network is responsible for inferring the desired difference image based on the input pair of images.

4.1 Network Architecture

We illustrate TransieNet in Fig. 3. It is a fully-convolutional encoder-decoder architecture inspired by the one introduced in Sedaghat et al. (2017). Ten convolutional layers are responsible for the contraction throughout the encoder, and learn features with varying levels of detail in a hierarchical manner. The expansion component of the network consists of 6 up-convolutional layers which decode the learned features, step by step, and generate estimates of the output with different resolutions along the way. We compute and

back-propagate errors computed based on all different resolutions of the output during training. But in the end and for evaluation purposes, we only consider the full-resolution output. This multi-scale strategy helps the network learn better features with different levels of detail. We use an L1 loss function at each output:

$$E = \frac{1}{N} \sum_{n=1}^N |y_n - \hat{y}_n| \quad (4)$$

where \hat{y} and y represent the prediction and the target (ground truth) respectively, and N is the number of samples in each mini-batch – see Sec. 4.3. The reason behind the choice of L1 loss over its more popular counterpart, L2 or Euclidean loss, is that the latter introduces more blur into the output, ruining pixel-level accuracy – see Sedaghat et al. (2017), Mathieu et al. (2015).

4.2 Data Preparation

Neural networks are in general data-greedy and require a large training dataset. TransieNet is not an exception and in view of the complexity of the problem – and equivalently the architecture – needs a large number of training samples: reference-science image pairs + their corresponding ground-truth images. Real astronomical image pairs with transients are not so publicly available. Difficulty of providing proper transient annotations makes them even scarcer. The best one could do is to manually (or semi-automatically) annotate image pairs, and find smart ways to estimate a close-to-correct ground truth image: a clean difference image with background-subtracted gradients. Although as explained in Sec. 4.2.1 we implement and prepare such a real training set, it is still too small (~ 200 samples) and if used as is, the network would easily overfit it.

One solution is to use image augmentation techniques, such as spatial transformations, to virtually increase the size of the training set. This trick, though necessary, is still not sufficient in our case with only a few hundred data samples – the network eventually discovers common patterns and overfits to the few underlying real scenes.

An alternative solution is to generate a large simulated (aka synthetic) dataset. However, relying only on the synthetic data makes the network learn features based on the characteristics of the simulated examples, making it difficult to transfer the knowledge to the real domain.

Our final solution is to feed the network with both types of data: synthetic samples mixed with real astronomical images of sky with approximate annotations. This along with online augmentation, makes a virtually infinite training set, which has the best of both worlds. We describe details about the datasets used and the training strategy in the following sub-sections.

4.2.1 Real Data

For real examples we make use of data from the Supernova Hunt project (Howerton 2017) of the CRTS survey. In this project image subtraction is performed on pairs of images of galaxies in search of supernovae. While this may bias the project towards finding supernovae rather than generic transients, that should not affect the end result as we mark

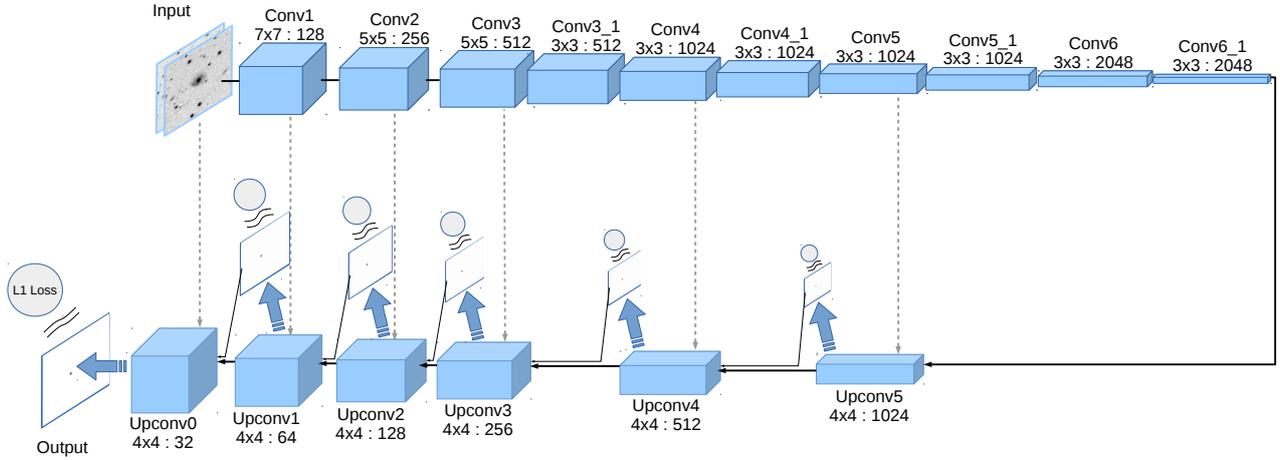


Figure 3. Our suggested fully convolutional encoder-decoder network architecture. The captions on top/bottom of each layer show the kernel size, followed by the number of feature maps. Each arrow represents a convolution layer with a kernel size of 3×3 and stride and padding values of 1, which preserves the spatial dimensions. Dotted lines represent the skip connections. Low-resolution outputs are depicted on top of each up-convolution layer with the corresponding loss. After each (Up-)convolution layer there is a ReLU layer which is not displayed here.

the transients found, and the ground-truth images contain just the transients. If anything, finding such blended point sources should make finding point sources in the field (i.e. away from other sources) easier. Unlike most other surveys, the CRTS images are obtained without a filter, but that too is not something that directly concerns our method. We gathered 214 pairs of publicly available jpeg images from SN Hunt and split this dataset in to training, validation and test subsets of 102, 26 and 86 members respectively. The reference images are typically made by stacking ~ 20 older images of the same area. The science image is a single 30-second exposure. The pixels are $2''.5 \times 2''.5$ and thus comparable or somewhat bigger than the typical PSF. Individual images are 120×120 pixels, and at times not perfectly registered.

To prepare the ground truth, we developed an annotation tool. The user needs to roughly define the location of the transient in the science image, by comparing it with the reference image, and put a circular aperture around it. Then the software models the background and subtracts it from the aperture to provide an estimate of the transient’s shape and brightness. Simple annulus-based estimates of the local background (Davis 1987; Howell 1989) or even the recent Aperture Photometry Tool (Laher et al. 2012), are not suitable for most of the samples of this dataset since the transients, often supernovae, naturally overlap their host galaxies. Therefore we use a more complex model and fit a polynomial of degree 8 to a square-shaped neighborhood of size $2r \times 2r$ around the aperture, where r is the radius of the user-defined aperture. Note however that the model fitting is performed only after masking out the aperture, to exclude the effect of the transient itself – the points are literally excluded from model-fitting – rather than being masked and replaced with a value such as zero. This method works reasonably well even when the local background is complex. Fig. 4 illustrates the process.

The annotations on real images are not required to be accurate, as the main responsibility of this dataset is to provide the network with real examples of the sky. This lack of accuracy is compensated by the synthetic samples with precise positions.

4.2.2 Synthetic Data

To make close-to-real synthetic training samples, we need realistic background images. Existing simulators, such as Sky-maker (Bertin 2009), do not yet provide with a diverse set of galaxy morphologies, and therefore are not suitable for our purpose. Instead, we use images from the Kaggle Galaxy Zoo dataset¹, based on the Galaxy Zoo 2 dataset (Willett et al. 2013), for our simulations. To this end, we pick single images as the background image and create a pair of reference-science images based on it.

This method also makes us independent of precise physical simulation of the background, allowing us to focus on simulations only at the image level – even for the ‘foreground’, i.e. transients. This may result in some samples that do not exactly resemble a ‘normal’ astronomical scene, in terms of the magnitude and location of the transient, or the final blur of the objects. But that is even better in a learning-based method, as the network will be trained on a more general set of samples, and less prone to over-fitting to specific types of scenes. Fig. 5 illustrates details of this process.

We first augment the background image using a random

¹ <https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge>

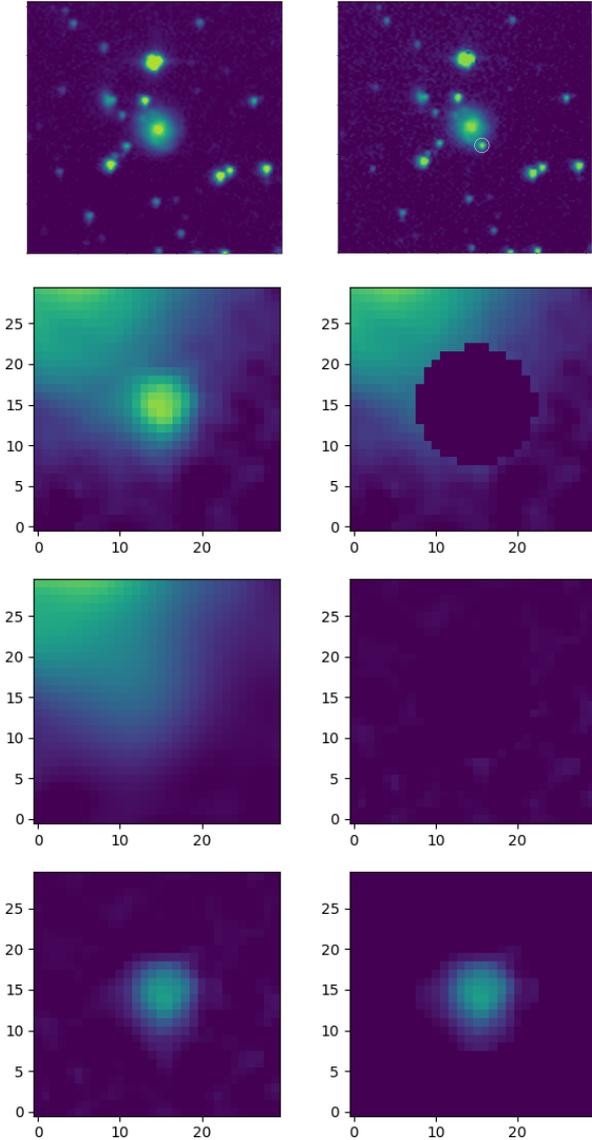


Figure 4. An exemplar transient annotation case. From top-left to bottom-right the first two images are the input ref/science pair. Number 3 illustrates the $2r \times 2r$ neighborhood of the transient and on 4 the user-defined aperture is masked out. Image 5 shows the polynomial model fit to the ‘masked-neighborhood’. Note that since the blank aperture is excluded from the fitting process, there’s no dark region in the results. In number 6 the estimated background is subtracted from the masked neighborhood to form a measure of how well the background has been modeled: the more uniform and dark this image is, the better the polynomial has modeled the background. Finally in number 7 the estimated background is subtracted from the neighborhood and the transient stands out. In number 8 the transient is cropped out of 7 using the user-defined aperture.

spatial transformation:

$$R \sim U(0, 2\pi) \quad (5)$$

$$T \sim N\left(\mu = 0, \Sigma = \begin{bmatrix} 0.03 & 0 \\ 0 & 0.03 \end{bmatrix}\right) \quad (6)$$

here R & T represent rotation and translation (shift) respec-

tively. U shows a normal distribution and N is a 2D normal distributions. T is then a 2D vector and its values show a translation proportional to the dimensions of the image.

At the next step simulated transients are added to the science (second) image as ideal point sources, with random locations and magnitudes, to form $I_0 + I_t$. The transient locations are again sampled from a 2D gaussian distribution. The distribution parameters are adjusted such that transients, although scattered all around the image, happen mostly in the vicinity of galaxies at the center of the image, to resemble real supernovae:

$$(X_t, Y_t) \sim N\left(\mu = 0, \Sigma = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}\right) \quad (7)$$

In most of our experiments we simulated only a single transient. But in cases where we had more of them, we made sure they were apart from each other by at least half of the bigger dimension of the image. The *amplitude* of the simulated source is also randomly chosen as:

$$A_t \sim N(\mu = 10, \sigma = 0.3) \quad (8)$$

This value, after being convolved with the (sum-normalized) PSF, will constitute the flux of the transient (F_t). We can select a specific range of A_t for training – to fine-tune the network – based on the range of transients (and their relative brightening) that we expect to find for a given survey.

The two images are then convolved with *different* gaussian PSFs, generated based on random kernel parameters, with a random eccentricity, limited by a user-defined maximum:

$$\sigma_{\phi, x} \sim U(\sigma_{\phi, m}, \sigma_{\phi, M}) \quad (9)$$

$$\sigma_{\phi, y} = \sigma_{\phi, x} \sqrt{1 - ecc^2} \quad (10)$$

$$ecc \sim N(\mu = 0, \sigma = ecc_{max}) \quad (11)$$

where $\sigma_{\phi, x}$ and $\sigma_{\phi, y}$ are the standard deviations of the 2D gaussian function along x and y directions respectively and $[\sigma_{\phi, m}, \sigma_{\phi, M}]$ is the range from which $\sigma_{\phi, x}$ is uniformly sampled. The PSF is then rotated using a random value, θ_{ϕ} , uniformly sampled from $[0, 2\pi]$. This should also help catch asteroids that would leave a very short streak.

Precisely modeling the difference between reference and science images, and adjusting the PSF parameters’ distributions accordingly, would be achievable. However, as stated before, we prefer to keep the training samples as general as possible. Therefore in our experiments $[\sigma_{\phi, m}, \sigma_{\phi, M}]$ is set to $[2, 5]$ for both images. These numbers are larger than typically encountered, and real images should fare better. The ecc_{max} value is set to 0.4 and 0.6 for reference and science images, respectively, to model the more isotropically blurred seeing of reference images.

The sky and noise levels are different for the reference and science images. We choose to model these difference in our simulations since in contrast to the previous parameters, ignoring them would make learning easier for the network – and that’s exactly what we want to avoid. We model the sky level, S , as a constant value, add it to the image and only after that ‘apply’ the Poisson noise to every pixel:

$$I_n(x, y) = poisson(\lambda = I(x, y) + S) \quad (12)$$

where *poisson* is a function returning a sample from a Poisson distribution with the given λ parameter, S is the sky model, and I_n is the noisy version of input I .

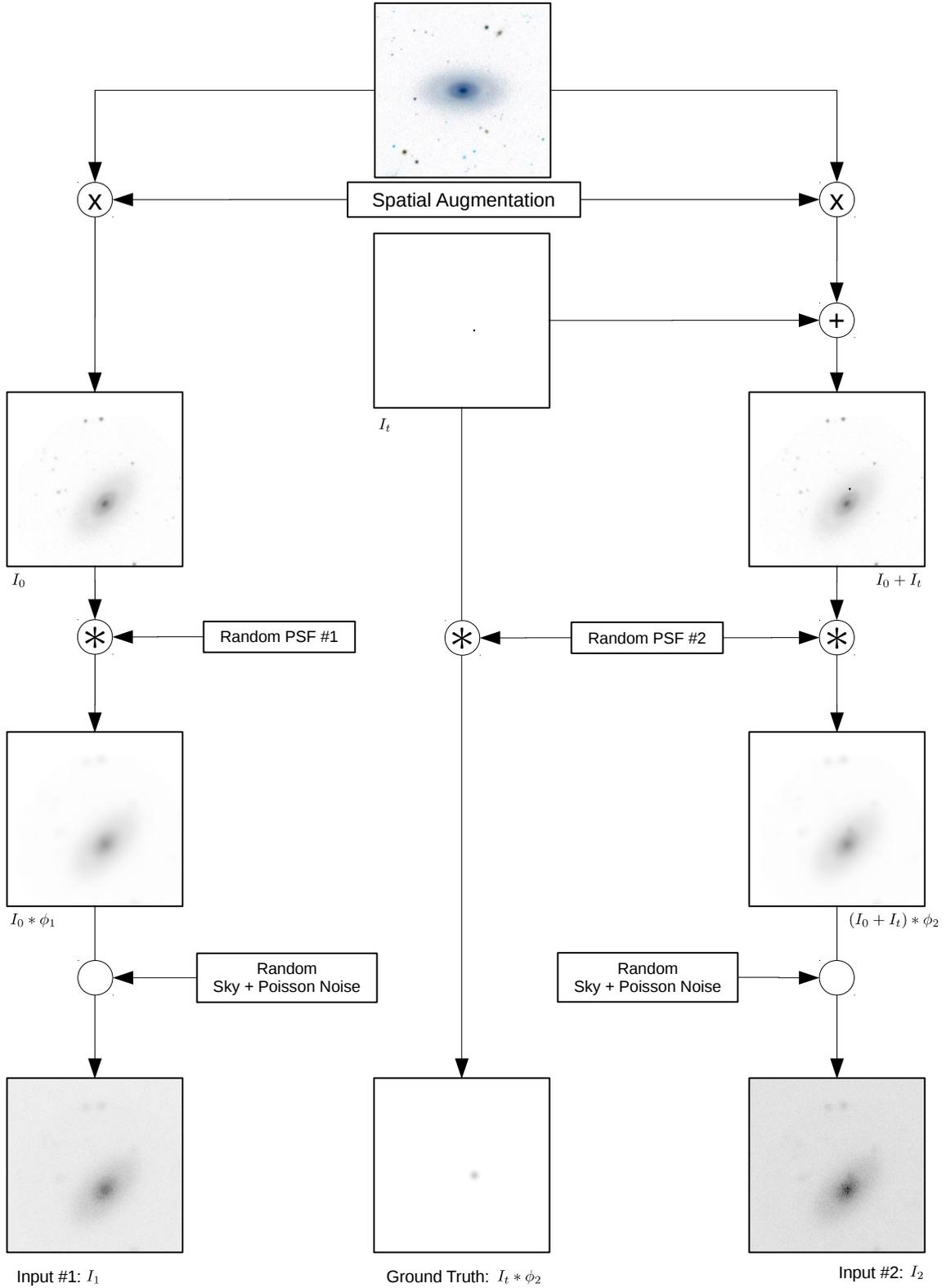


Figure 5. The synthetic sample generation procedure. The notations used here are described in Equations (1) and (2).

Then we perform a pairwise augmentation (rotation, scaling and translation), such that the two images are not perfectly registered. This forces the network to also learn the task of registration on the fly.

The ground truth image is then formed by convolving the ideal transient’s image, I_t , with the same PSF as applied to the science image. No constant sky value or noise are applied to this image. This way the network learns to predict transient locations and their magnitudes in the same seeing conditions as the science image, in addition to noise removal and sky subtraction.

4.3 Training Details

We have two networks, one generic, and the other specialized to the SN Hunt dataset. Each shares the 90K images from Kaggle zoo, with transients inserted to create *synthetic* science images. The *real* data currently is just the ~ 100 SN Hunt image pairs earmarked for training. The zoo and SN Hunt images are further rotated, shifted, and scaled to augment the dataset, and also make the network more robust. Training is done in small batches of 16. We use ADAM for optimization using the Caffe framework (Jia et al. 2014). First 90K iterations are common to both networks. Images used are 140×140 . Then, for the generic network fine tuning is done using batches that contain 12 CRTS and 4 zoo images. The image size here is 256×256 (with images scaled where needed), and the number of iterations is 50K. We put 12 real images and 4 synthetic images in each batch during this second round of training to prevent over-fitting to the small-sized real dataset. We start with a learning rate of $3e^{-4}$ and drop it in the second round by a factor of 0.3 every 20K iterations. Running on an NVIDIA GTX 1070 along with 16 CPU cores, the whole training process takes a day and half to complete.

For the specialized SN Hunt network, on the other hand, the entire fine-tuning is done using 8K iterations on CRTS images (we emphasize that we are still using just the ~ 100 image-pairs, modified in many ways). That way it is better at recognizing transients in real data.

4.4 The Attention Trick

In this specific type of application, the target images mainly consist of black regions (i.e. zero intensity pixels), and non-zero regions take up only a small number of pixels. Therefore mere use of a simple L1 loss does not generate and propagate big enough error values back to the network, when the network has learned to remove the noise and generates blank images. So the network spends too long a time focusing on generating blank images instead of the desired output, and in some cases fails to even converge. The trick we use to get around this issue is to conditionally boost the error on the interesting regions. The realization of this idea is to simply apply the mapping $[0, 1] \rightarrow [0, K]$ on the ground-truth pixel values. K represents the boosting factor and we set it to 100 in our experiments. This effectively boosts the error in non-zero regions of the target, virtually increasing the learning rate for those regions only. The output of the network is later downscaled to lie in the normal range. Note that increasing the total learning rate is not an alternative solution, as the network would go unstable and would not even converge.

5 EXPERIMENTS AND RESULTS

We have run TransiNet on samples from CRTS SN Hunt and the Kaggle zoo dataset. The network weights take up about 2GB of memory. Once read, on the NVIDIA GTX 1070 the code runs fast: 39ms per sample, which can be reduced to 14ms if samples are passed to the network as batches of 10. The numbers were calculated by running tests on 10000 images three times. Fig. 6 depicts samples from running TransiNet on the CRTS test subset. The advantage of TransiNet is that the “image differencing” produces a noiseless image ideally consisting of just the transient, and thus robust to artifacts and removes the need for human scanners.

With increasing CCD sizes it is much more likely than not that there will be multiple transients in a single image. Since the SN Hunt images, or the zoo images used, rarely have multiple transients, the networks may not be ideal when looking for such cases. However, because of the way the network is trained - with the output as pure PSF-like transients, it is capable of finding multiple transients though we did not train it explicitly with such cases. This is demonstrated in Fig. 7 which depicts an exemplar sample from the zoo subset. Here we introduced four transients, and all were correctly located. Another side effect - a good one - is that the network rejects non-PSF like additions, including Cosmic Rays. In addition to the four transients, we had also inserted 10 single pixel Cosmic Rays in the science image shown in Fig. 7 and all were rejected. An example from the SN Hunt set is shown in Fig. 8 which happens to have two astrophysical objects - the second is likely an asteroid. Here too, the network has detected both transients. Locating new asteroids is as useful as locating transients to help make the asteroid catalog more complete for future linking and position predicting.

5.1 Quantitative Evaluation

We provide below quantitative evaluations of TransiNet performance.

5.1.1 Precision-Recall Curve

Precision-recall curves are the de-facto evaluation tool for detectors. They capture *TruePositives* (TP, or ‘hits’, the number of correctly detected objects), *FalsePositives* (FP, or ‘false alarms’), and *FalseNegatives* (FN, or ‘misses’, the number of missed real objects) for all possible thresholds. This allows users to either set a fixed threshold, or a dynamic threshold (e.g. 5σ above the background level, or 70% of the max in a difference image etc.)

$$\text{Precision} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalsePositive}} \quad (13)$$

$$\text{Recall} = \frac{\text{TruePositive}}{\text{TruePositive} + \text{FalseNegative}} \quad (14)$$

Low-SNR Detections & Blank Outputs The output of TransiNet is an image with real-valued pixels. Therefore each pixel is more likely to contain a non-zero real value, even in the ‘dark’ regions of the image, or when there is no transient to detect at all. Thus, we consider low-SNR detection images

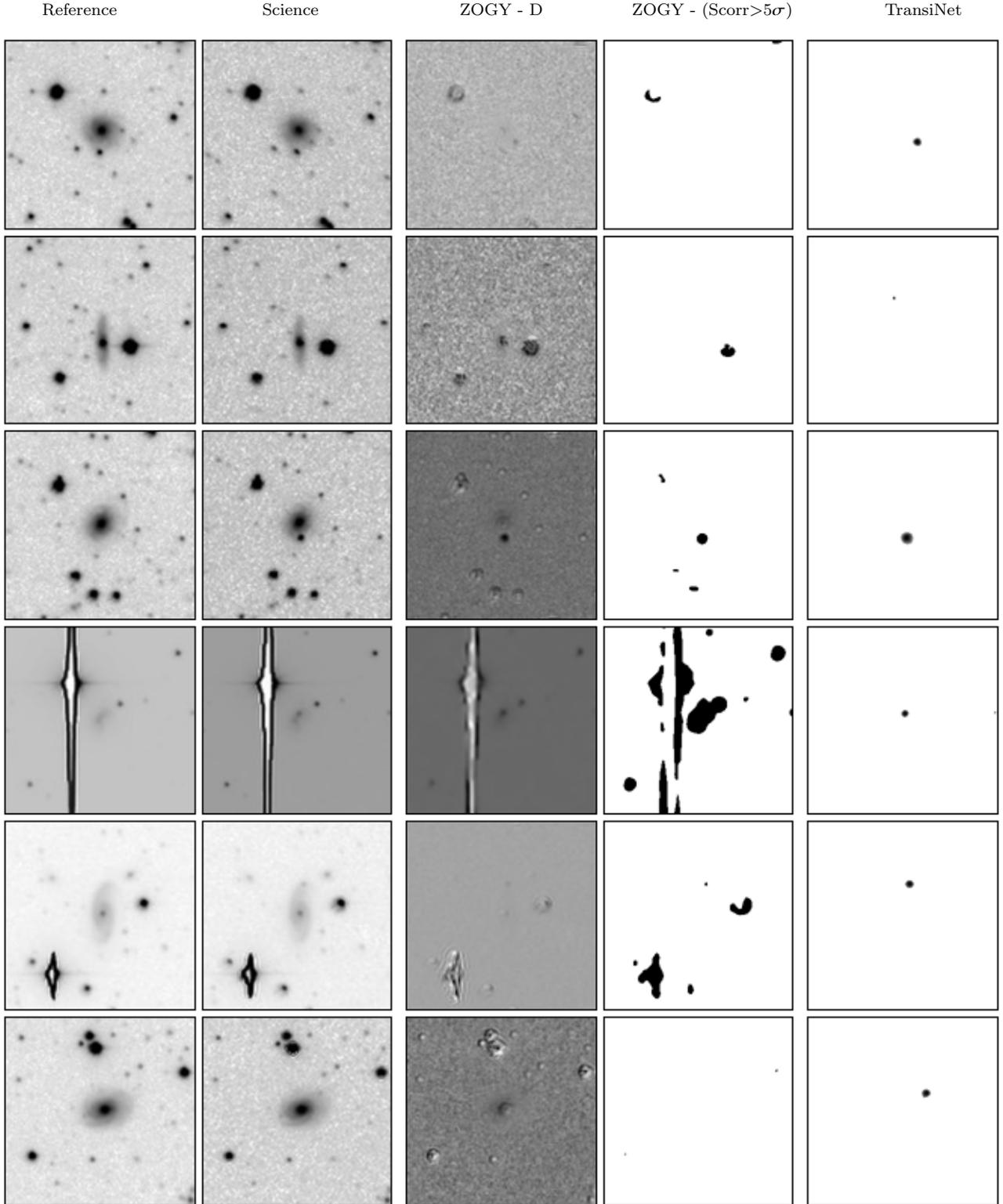


Figure 6. Image subtraction examples using ZOGY and TransiNet for a set of CRTS Supernova Hunt images. The first column has the deep reference images, second column contains the science images which have a transient source and are a shallower version of the reference images. The third column contains the ZOGY D images, and the fourth has the ZOGY Scorr images i.e. “the matched filter difference image corrected for source noise and astrometric noise” (Zackay et al. 2016). The fifth column has the thresholded versions of ZOGY SCORR, as recommended in that paper. The sixth column shows the difference image obtained using TransiNet. All images are mapped to the $[0,1]$ range of pixel values, with a gamma correction on the last column for illustration purposes. TransiNet has a better detection accuracy, and is also robust against noise and artifacts. It is possible that ZOGY could be tuned to perform better, and on a different dataset provide superior results - the reason for the comparison here is to simply show that TransiNet does very well.

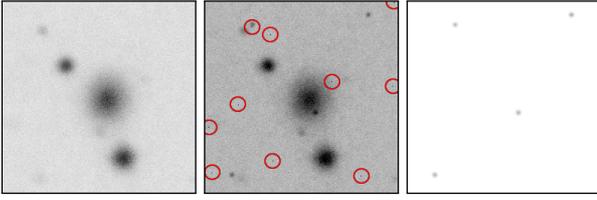


Figure 7. An exemplar multi-transient case from the zoo dataset. The reference image (left), science image (middle) with 10 single-pixel Cosmic Ray events, indicated by red circles, and four transients, and the network prediction (right) with all transients detected cleanly, and all CRs rejected.

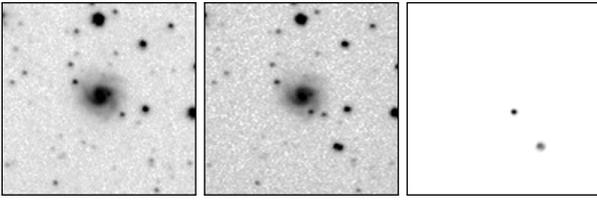


Figure 8. An exemplar multi-transient case from the CRTS SN Hunt dataset. The science image (middle) has two transients, and the network prediction (right) finds them both though never trained explicitly to look for multiple transients.

as blank images. The outputs of the network (detection images) which have a standard deviation (σ) lower than 0.001 were marked as blank images during our experiments and unconsidered thereafter.

Binarization and Counting of Objects Evaluations at a series of thresholds is the essence of a precision-recall curve and helps reveal low SNR contaminants, while digging for higher completeness (see Fig. 9).

The thresholding

$$\hat{y}_{ij} = \begin{cases} 0 & \hat{y}_{ij} < \tau \\ 1 & \hat{y}_{ij} \geq \tau \end{cases} \quad (15)$$

results in the binary image, \hat{Y} , on which we obtain ‘connected’ regions to count detected objects with full connectivity (Fiorio & Gustedt 1996; Wu et al. 2005). For this specific kind of evaluation, we also convert the ground truth image (y) to a similar binary-valued image, Y , using a fixed threshold.

Let P be the set of all positives, i.e. the objects in \hat{Y} , and G the set of all objects in Y . Then we have

$$TP = P \wedge G \quad (16)$$

where \wedge is used here to denote spatial intersection, such that TP is the set of objects in P that have a spatial intersection with a member of G . TP is the set of *True Positives*. We conversely define $TP' = G \wedge P$ which is of the same cardinality as TP and includes the set of objects in G that have been detected. Then we also have:

$$FP = P - TP \quad (17)$$

$$FN = G - TP' \quad (18)$$

in which FP and FN stand for *False Positives* and *True*

Positives respectively. Now we can rewrite Equations (13) and (14) in a more compact and formal form as:

$$Precision = \frac{|TP|}{|TP| + |FP|} \quad (19)$$

$$Recall = \frac{|TP|}{|TP| + |FN|} \quad (20)$$

where $|\cdot|$ represents the cardinality of the set. We also define *completeness* and *contamination* measures as follows:

$$Completeness = \frac{|TP|}{|TP| + |FN|} = Recall \quad (21)$$

$$Contamination = \frac{|FP|}{|TP| + |FP|} = 1 - Precision \quad (22)$$

Figs. 10 and 11 depict the precision-recall curves corresponding to the two versions of TransiNet before and after blanking. Each curve is obtained by sweeping the threshold (τ) in the pixel-value domain. Starting from the minimum (0), \hat{Y} is set to 1 everywhere, resulting in a 100% recall (everything that is to be found is found) with a close-to-zero precision (too many false positives), which is equivalent to total contamination. But as we increase τ , fewer pixels in \hat{Y} ‘fire’, generally resulting in a lower recall (some misses) and higher precision (far fewer contaminants) – see Fig. 9. To generate the curves we sampled 101 logarithmically-distributed values for τ from the range $[10^{-4}\sigma, 100\sigma]$, where σ is the standard deviation of the pixel values in each detection image (y). Also the ground truth images were binarized with a fixed threshold of 10^{-3} .

The sharp and irregular behavior of the curve at around 75% of recall on the CRTS dataset, is due to the low contamination levels in the output: transients are detected with a high significance. Contaminants, if any, have a much lower intensity, and their number goes up only when one pushes for high completeness to the lower significance levels.

5.1.2 Relative Magnitude of the Transient

Thanks to the freedom in generation of synthetic samples with different parameters, we can evaluate the performance of the network with transients of different magnitudes. However, for this evaluation we use *relative magnitudes*, as opposed to the absolute intensities used during training. This would make it easier to quantitatively determine the ability of the network to detect faint transients without contamination. In the future we hope to incorporate similar process during training as well.

We define the relative magnitude as the difference of magnitudes at the location of transient, with and without the transient:

$$mag_{rel} = -2.5 \log_{10}(F_{rel}) \quad (23)$$

$$F_{rel} = \frac{F_t + F_{local}}{F_{local}} \quad (24)$$

where F_t is the absolute flux of the transient, and F_{local} represents the flux of the background, before having the transient. The latter is measured inside an FWHM-sized square neighborhood around the location of the transient.

Fig. 12 depicts the performance of the detector for several relative magnitudes, in terms of the precision-recall curve. With higher visibility, the curve approaches the ideal

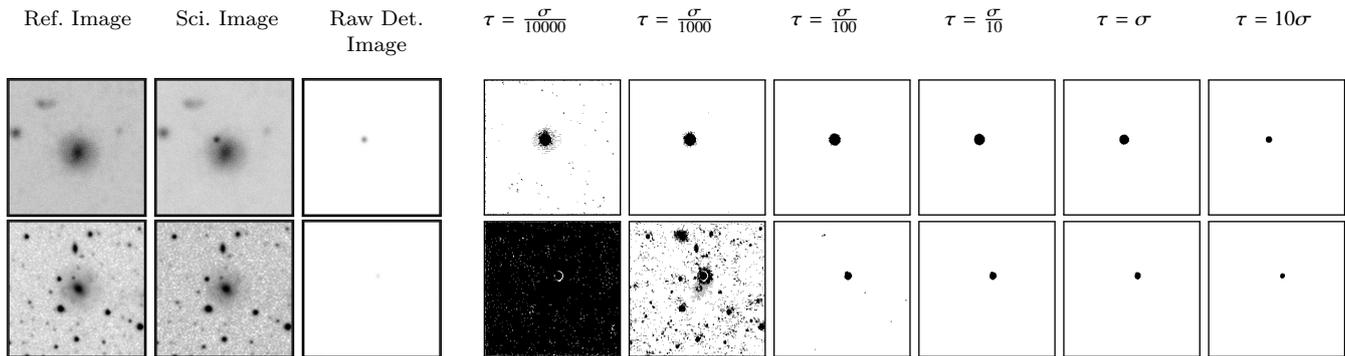


Figure 9. Visualization of the thresholding process used for generation of precision-recall curves. Each row illustrates exemplar levels of thresholding of a single detection image: first row is chosen from the synthetic subset and the second row from CRTS. Outputs of the network are normally quite clean, and contaminants practically appear only after taking the threshold down below the noise level. This is particularly visible in the second row, where the transient has been of a low magnitude, and so the detection image has a low standard deviation (σ). Thus $\frac{\sigma}{100}$ is still too low and below noise level.

form. Considering that during the training phase, the network has rarely seen transients with such low magnitudes as the ones in the lower region of this experiment, it is still performing well. We expect it to gain much better results by broadening the range of simulated transient amplitudes during training.

5.1.3 Robustness to Spatial Displacements

We analyze the robustness of the TransiNet to pairwise spatial inconsistencies between the science and reference images. That way small rotations, WCS inconsistencies etc. do not give rise to Yin-Yang like ‘features’ and lead to artifacts. To this end, for a subset of image pairs, we exert manual shift, rotation and scaling to one of the images in each pair, and pass them through the network. Fig. 13 depicts the results of these experiments as plots of completeness and contamination vs. the manual perturbation.

5.1.4 Performance of the two TransiNet Networks

Table 1 summarizes the testing results for the two TransiNet networks. For new surveys one can start with the generic network, and as events become available, fine-tune the network with specific data.

5.2 Comparison with ZOGY

Given the generative, and hence very different, nature of our ‘pipeline’, it is difficult to compare it with direct image differencing pipelines. We have done our best by comparing the output of TransiNet and of ZOGY for synthetic as well as real images. We used the publicly available MATLAB version of ZOGY, and almost certainly we used ZOGY in a sub-optimal fashion. As a result this comparison should be taken only as suggestive. More direct comparisons with real data (PTF and ZTF) are planned for the near-future. Fig. 6 depicts the comparison for a few of the SN Hunt transients.

Both pipelines could be run in parallel to choose an ideal set of transients, since the overhead of TransiNet is minuscule.

Network	Transients	TP	FP	FN	Prec.	Recall
Synth/Zoo	100	100	0	0	100.0	100.0
CRTS/SNH	86	65	1	21	98.4	75.5

Table 1. Hits and misses for TransiNet for the Synthetic and SN Hunt networks. TransiNet does very well for synthetics. One reason could be that there isn’t enough depth variation in the reference and science images. But for CRTS too the output is very clean for the recall of 76% that it achieves. The lower (than perfect) recall could be due to a smaller sample, larger pixels, large shifts in some of the cutouts etc. Fine tuning with more data can improve performance further. The fixed thresholds used for the synthetic and SN Hunt networks are 40 and 20 respectively.

6 FUTURE DIRECTIONS

We have shown how transients can be effectively detected using TransiNet. In using the two networks we described, one with the Kaggle zoo images, and another with CRTS, we cover all broad aspects required, and yet for this method to work with any specific project, e.g. ZTF, appropriate tweaks will be needed, in particular labeled examples from image differencing generated by that survey. Also, the assumptions during simulations can be improved upon by such examples. Using labeled sets from surveys accessible to us is definitely the next step. Since the method works on the large pixels that CRTS has, we are confident that such experiments will improve the performance of TransiNet.

The current version produces convolved transients to match the shape and PSF of the science image. One can modify the network to produce just the transient location and leave the determination of other properties to the original science and reference images as they contain more quantitative detail.

Further the network could be tweaked to find variable sources too. But for that a much better labeled non-binary training set will be needed. In the same manner, it can also be trained to look for drop-outs, objects that have vanished in recent science images but were present in the corresponding reference images. This is in fact an inverse of the transients problem, and somewhat easier to do.

In terms of reducing the number of contaminants even further, one can provide as input not just the pair of science

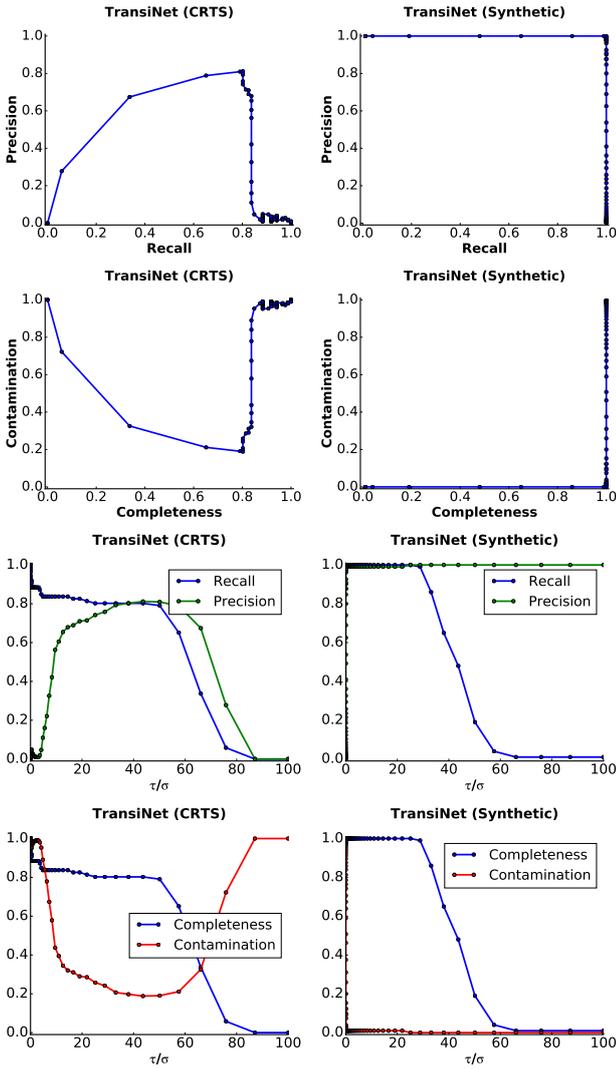


Figure 10. Plots showing precision-recall (row 1), completeness-contamination (row 2), and their dependence on threshold (rows 3 and 4) for TransiNet before the blanking is done to remove ultra low-SNR detections (see text). The two columns show CRTS (left) and synthetic (right) subsets. For CRTS a threshold can be picked where 80% transients are detected with little contamination. Not unexpectedly, the performance is better for the synthetic images.

and reference images, but also pairs of the rotated (by 90, 180, 270 degrees) and flipped (about x- and y-axes) versions. The expectation is that the transient will still be detected (perhaps with a slightly different peak, extent), but the weak contaminants, at least those that were possibly conjured by the weights inside the network, will be gone (perhaps replaced by other – similarly weak ones – at a different location), and the averaging of the detections from the set will leave just the real transient.

Another way to eliminate inhomogeneities in network weights is to test it with image pairs without any transients. While most image pairs do not have any transient except in a small number of pixels, such a test can help streamline the network better.

In order to detect multiple transients, one could cut the image in to smaller parts and provide these subimages for

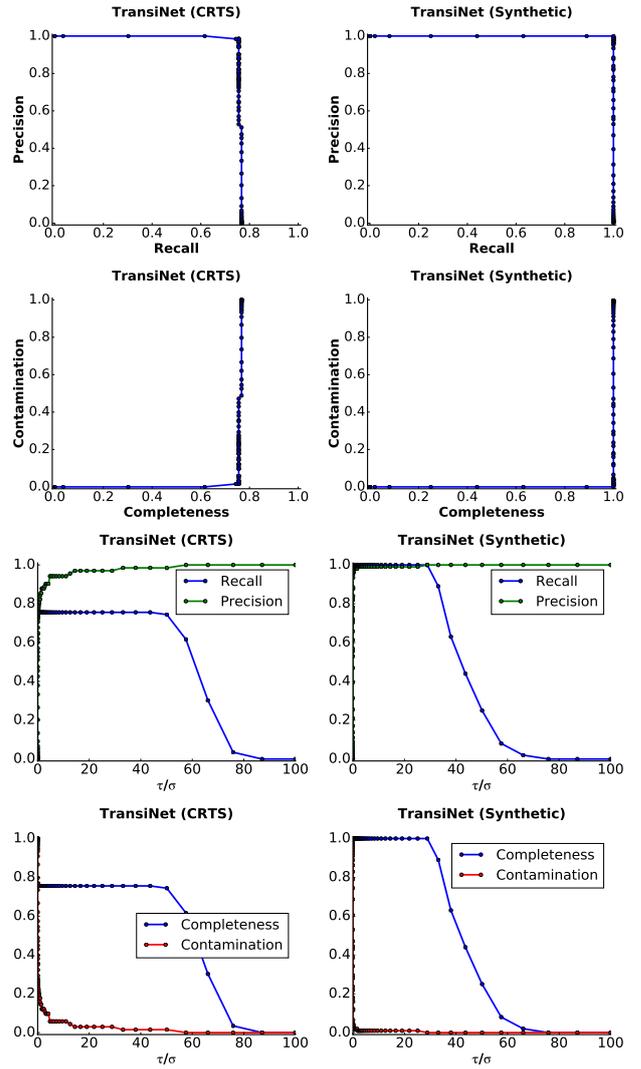


Figure 11. The plots are as in Fig. 10, but post-blanking. The plots are now closer to ideal. In this scenario, for CRTS, we never go above a completeness of 80% detections, but all those detections are clean, and the ones we miss are the really low significance ones below the blanking threshold of 0.001. Fig. 9 shows a single transient field for each type at different thresholds.

detection. Another possibility is to mask the ‘best’ transient, and rerun the pipeline to look for more transients iteratively until none is left. An easier fix is to train the network for larger images, and for multiple transients in each image pair.

Another way to improve the speed of the network is to experiment with the architecture, and if possible obtain a more lightweight network with a smaller footprint that performs equally well. Finally, the current network used jpegs with limited dynamic range as inputs. Using non-lossy FITS images should improve performance of the network.

7 CONCLUSIONS

We have introduced a generative method based on convolutional neural networks for image subtraction to detect transients. It is superior to other methods as it has a higher

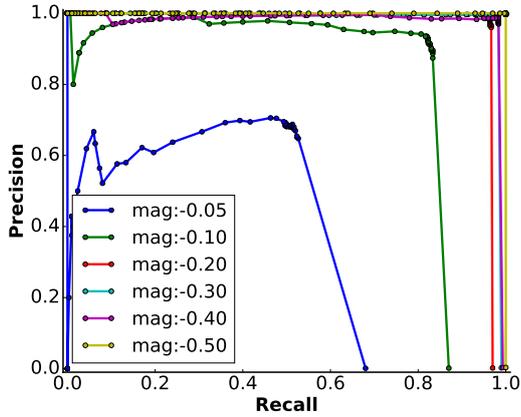


Figure 12. Precision-recall curves for a range of magnitudes. These are for the synthetic transients where we had control over the relative magnitudes. The network misses more transients as the relative magnitude goes lower. This is not unexpected as the network has not seen such faint samples during training. The sharp vertical transitions reflect the clean nature of the detection images.

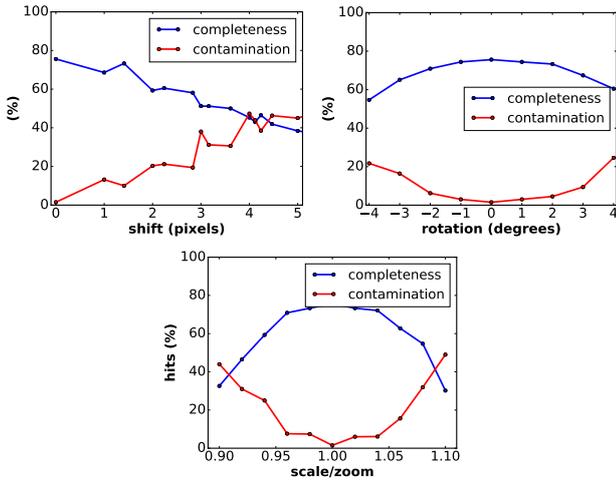


Figure 13. Robustness of the network to shift (top-left), rotation (top-right), and scaling (bottom) between the reference and science images. Ideally there will be no misalignments, but some can creep in through improper WCS, or changes between runs etc. The peak detectability is close to 75% across the board.

completeness at lower thresholds, and at the same time has fewer contaminants. Once the training is done with appropriate labeled datasets, execution on individual images is fast. It can operate on images of any size (after appropriate training), and can be easily incorporated in to real-time pipelines. While we have not explicitly tested the method on high-density fields (e.g. closer to the plane of the Galaxy) it will be possible to get good performance once a corresponding labeled dataset is used for training. We hope surveys like ZTF, LSST as well as those with larger pixels like ASAS-SN (Shappee et al. 2014), Evryscope (Law et al. 2015) etc. adapt and adopt the method. It is also possible to extend the

method to other wavelengths like radio and use for surveys including SKA and its path-finders.

ACKNOWLEDGEMENTS

AM was supported in part by the NSF grants AST-0909182, AST-1313422, AST-1413600, and AST-1518308, and by the Ajax Foundation.

REFERENCES

- Abbott B.P. et. al. p., 2017, Phys. Rev. Lett. 118, 221101
 Abraham S. et. al. i., 2017, Submitted
 Alard C., Lupton R. H., 1998, *ApJ*, 503, 325
 Bellm E., 2014, in Wozniak P. R., Graham M. J., Mahabal A. A., Seaman R., eds, The Third Hot-wiring the Transient Universe Workshop. pp 27–33 ([arXiv:1410.8185](https://arxiv.org/abs/1410.8185))
 Bengio Y., Yao L., Alain G., Vincent P., 2013, in Advances in Neural Information Processing Systems. pp 899–907
 Bertin E., 2009, *Memorie della Societa Astronomica Italiana*, 80, 422
 Bertin E., Arnouts S., 1996,] 10.1051/aas:1996164, 117, 393
 Bramich D. M., 2008, *MNRAS*, 386, L77
 Bue Brian et al. i., 2017, In Prep.
 Cabrera-Vives G., Reyes I., FÄurster F., EstÄlvez P., Maureira J.-C., 2016, *IJCNN*, pp 251–
 Chambers K. C., et al., 2016, preprint, ([arXiv:1612.05560](https://arxiv.org/abs/1612.05560))
 Charnock T., Moss A., 2017, *ApJ*, 837, L28
 Davis L., 1987, National Optical Astronomy Observatories. Revised October
 Djorgovski S. G., et al., 2008, *Astronomische Nachrichten*, 329, 263
 Djorgovski S. G., et al., 2011, preprint, ([arXiv:1110.4655](https://arxiv.org/abs/1110.4655))
 Drake A. J., et al., 2009, *ApJ*, 696, 870
 Fiorio C., Gustedt J., 1996, *Theoretical Computer Science*, 154, 165
 Gaia Collaboration et al., 2016, *A&A*, 595, A1
 George D., Shen H., Huerta E. A., 2017, preprint, ([arXiv:1706.07446](https://arxiv.org/abs/1706.07446))
 Howell S. B., 1989, *Publications of the Astronomical Society of the Pacific*, 101, 616
 Howerton S. C., 2017, *CRTS SNhunt: The First Five Years of Supernova Discoveries*
 Hoyle B., 2016,] <https://doi.org/10.1016/j.ascom.2016.03.006>
 Ivezić Z., et al., 2008, preprint, ([arXiv:0805.2366](https://arxiv.org/abs/0805.2366))
 Jia Y., Shelhamer E., Donahue J., Karayev S., Long J., Girshick R., Guadarrama S., Darrell T., 2014, arXiv preprint [arXiv:1408.5093](https://arxiv.org/abs/1408.5093)
 Krizhevsky A., Sutskever I., Hinton G. E., 2012, in Advances in Neural Information Processing Systems. pp 1097–1105
 Laher R. R., Gorjian V., Rebull L. M., Masci F. J., Fowler J. W., Helou G., Kulkarni S. R., Law N. M., 2012, *Publications of the Astronomical Society of the Pacific*, 124, 737
 Law N. M., et al., 2009, *PASP*, 121, 1395
 Law N. M., et al., 2015, *PASP*, 127, 234
 LeCun Y., 1985, in , Proceedings of Cognitiva 85, Paris, France
 LeCun Y., Boser B. E., Denker J. S., Henderson D., Howard R. E., Hubbard W. E., Jackel L. D., 1990, in Advances in Neural Information Processing Systems. pp 396–404
 LeCun Y., Bottou L., Bengio Y., Haffner P., 1998, 86, 2278
 Mahabal A. A., et al., 2011, *Bulletin of the Astronomical Society of India*, 39, 387
 Mahabal A., Sheth K., Gieseke F., Pai A., Djorgovski S. G., Drake A., Graham M., the CSS/CRTS/PTF Collaboration 2017, preprint, ([arXiv:1709.06257](https://arxiv.org/abs/1709.06257))

- Masci F. J., et al., 2017, *PASP*, **129**, 014002
- Mathieu M., Couprie C., LeCun Y., 2015, arXiv:1511.05440
- Morii M., et al., 2016, *PASJ*, **68**, 104
- Mukund N., Abraham S., Kandhasamy S., Mitra S., Philip N. S., 2017, *Phys. Rev. D*, **95**, 104059
- Pojmański G., 2014, Contributions of the Astronomical Observatory Skalnaté Pleso, **43**, 523
- Raina R., Madhavan A., Ng A. Y., 2009, in Proceedings of the 26th Annual International Conference on Machine Learning. ACM, pp 873–880
- Rezende D. J., Mohamed S., Wierstra D., 2014, arXiv preprint arXiv:1401.4082
- Rumelhart D. E., Hinton G. E., 1986, *NATURE*, 323, 9
- Sedaghat N., Zolfaghari M., Brox T., 2017, Technical report, Hybrid Learning of Optical Flow and Next Frame Prediction to Boost Optical Flow in the Wild. arXiv:1612.03777
- Shappee B. J., et al., 2014, *ApJ*, **788**, 48
- Simonyan K., Zisserman A., 2014, preprint, ([arXiv:1409.1556](https://arxiv.org/abs/1409.1556))
- Willett K. W., et al., 2013, *MNRAS*, **435**, 2835
- Wu K., Otoo E., Shoshani A., 2005, Lawrence Berkeley National Laboratory
- Zackay B., Ofek E. O., Gal-Yam A., 2016, *ApJ*, **830**, 27
- Zevin M. e. a., 2017, *Class. Quantum Grav* 34, 6

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