# Transient Photometry with Star Trails

**David Thomas***
Institute for Computational and Mathematical Engineering
Kavli Institute for Particle Astrophysics and Cosmology
Stanford University
dthomas5@stanford.edu

## Abstract

We develop a novel experiment that allows telescopes to detect transient events lasting from seconds down to milliseconds. This is made possible by rotating the telescope during the exposure and using deep learning to detect bursts in star trails. We use high fidelity physics simulations to generate telescope images and inject them with transient bursts. We present a deep neural network that can localize the bursts and a binary classifier that can reliably detect the images with transient sources. This work will enable telescopes to probe an entirely new regime for transient astrophysics.

## 1 Introduction

Current telescopes are blind to transient events, such as a burst of flux from a star, that last for seconds or less. A typical exposure time is around one minute, and a sequence of exposures is necessary to measure changes over time. In this work, we develop a technique that allows existing telescopes, with only trivial modifications, to finally see short duration astrophysical events.

The first component of our solution is to rotate the telescope with respect to its target during the exposure (or turning off the tracking that counteracts the Earth's rotation). This smears the sources in the image from circles into trails (referred to as star trails). In a trail we can detect changes to the source within a single image. We can track the source's movement within a trail pixel by pixel, which allows us to improve the time resolution by four orders of magnitude - from minutes to milliseconds[2]! However to realize this potential, we must extract the transient stellar events from crowded and noisy images with many trails, some of which intersect. We develop a deep neural network that extracts transient events and completes our solution.

The input to the network are simulated trail images, where the value in each pixel is a photon count; the output is an image which only contains excess flux from transient sources, again the value in each pixel is a photon count. We expect transient events to be rare and need a mechanism for converting the output images into binary detections. We develop a second neural network to make this determination on the images output by the first network.

There are multiple short duration astrophysical phenomena that produce signatures in the optical regime of the electromagnetic spectrum. Pulsars, gamma ray bursts, fast radio bursts, and stellar flares are areas of active research (Levin et al., 2017; Goldstein et al., 2017; Chatterjee et al., 2017; Van Doorsselaere et al., 2017). The Large Synoptic Survey Telescope, currently under construction, will rapidly scan the sky and capture around ten million sources in a single image. Our method will enable such a facility to see into the transient regime and propel multiple science communities.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

[2]This measurement comes from the 15 arcsec per second rotation speed of the Earth and the 0.21 arcsec Large Synoptic Survey Telescope camera pixel pitch.

CS230: Deep Learning, Winter 2018, Stanford University, CA.

## 2 Related work

There are authors who considered using star trails for science and even photometry many decades ago. Harlan & Walker (1965) and Walker (1971) used star trails for site testing. Then Howell & Jacoby (1986) conceived of using charge coupled devices for time resolved photometry, which is closer to our core idea. Given the major advances in telescopes and software capabilities over the past thirty years, it is an appropriate time to re-examine these ideas.

Over the past three years deep learning has found many applications in astrophysics. (Dieleman et al., 2015) used a convolutional neural network to predict galaxy morphologies in the Galaxy Zoo project . Schawinski et al. (2017) used a generative adversarial network to recover astrophysical features in images beyond the deconvolution limit. Lanusse et al. (2018) developed a method to find galaxy-galaxy strong lenses. Hezaveh et al. (2017) reduced the time to analyze strong gravitational lenses by seven orders of magnitude. Shallue & Vanderburg (2018) identified new exoplanets. George & Huerta (2017) improved gravitational wave detection. Mahabal et al. (2017) classified stellar light curves. Finally, Sedaghat & Mahabal (2017) used deep learning for images subtraction, which is closest to our application. Most of the applications so far have improved analysis on existing datasets, we believe our method is the first to introduce a new experiment.

## 3 Dataset and Features

We ultimately look to deploy our method on the upcoming Large Synoptic Survey Telescope (LSST), so we produce a dataset of extremely realistic simulated LSST images for our network. We employ three simulators to produce them. The first is the LSST Operations Simulator (Delgado et al. 2014) which we use to generate a sequence of observation targets in the 'r' band. The second is the LSST Catalog Simulator (Connolly et al. 2014) which produces large catalogs - which are files with all the sources, around 100,000 for a single sensor at LSST depth. Finally we use the LSST Photon Simulator (*PhoSim*, Peterson 2014; Peterson et al. 2015) to map (observation, catalog) into (image).

We make significant modifications to PhoSim to allow for star trails and their labels[3]. We design a new I/O interface so that it can support catalogs with transient objects. We model transient objects as stars with bursts of flux that are parameterized by their duration and a multiplicative factor of the default brightness. We create four new simulation modes to simulate these bursts:

- Standard: simulates static images with no trailing.
- Trailing without Bursts: simulates trail images but does not simulate bursts.
- Trailing with Bursts: simulates trail images and bursts.
- Only Bursts: produces images that only contain the bursts. It runs the simulation as if the other sources and their photons are present to account for coupling and interference effects. These images align perfectly with the *trailing with bursts* images down to single pixels and serve as labels.

The PhoSim simulations are very computationally expensive ( 24 cpu-hour per one simulated image) because they model the atmosphere, optics, and electronics with high fidelity. We augment our data in multiple ways to get around this limitation. We simulate 100 backgrounds at full density in *trailing without bursts* mode, and simulate 10 sparse foregrounds for each of the backgrounds in *trailing without bursts, trailing with bursts*, and *only bursts* modes. Adding these three foregrounds to the background results in three images: one with bursts, one without bursts, and one that is a label for the one with bursts (the label for the one without bursts is an image of zeros). Then we randomly crop a 512 x 512 region from these 4000 x 4072 images. We down sample them to 256 x 256. Then we randomly rotate them by 0, 90, 180, or 270 degrees and or flip them. We take the log of the images to bring the max value from 100,000 down to 11. Finally we add Poisson noise to help our network further generalize. We are left with two (input, label) pairs, one with bursts and one without bursts.

We use 90 backgrounds in the training set and the remaining 10 backgrounds for the dev and test sets. We produce 180,000 images for the training set and 200 each for the dev and test sets. Half of the

[3]Our modifications to PhoSim can be found on Bitbucket: bitbucket.org/davidthomas5412/phosim_release.

samples contain bursts; half of the samples do not contain bursts and have a zero matrix as their label. We also design a test set to characterize the performance of our method in variables astronomers can appreciate. This set is partitioned into bins that allow us to measures the sensitivity of our method to the magnitude (brightness) of the transient source and the duration of the burst.

## 4 Methods and Results

We first tried to modify existing statistical methods to detect the transient bursts. Zackay et al. (2016) present a nice formalism for finding significant differences between telescope images (which are warped from the atmosphere, optics, and electronics). Appendix A describes the mathematical changes we make to their model. Not only did we show in our milestone that these methods produce unwanted image artifacts and are incapable of bringing our novel method to fruition, they also required two exposures and knowledge of the Point Spread Function (PSF). We turned to deep learning to achieve reasonable burst extraction performance and to drop the need for reference images or assumptions about the PSF.

In our initial testing we down-sampled the images to 64 x 64 so we could quickly find out which general architectures and loss functions worked best[4]. We use Pytorch (Paszke et al., 2017) to build, train, and evaluate neural networks. The network architectures we experimented with were inspired by the UNet (Ronneberger et al., 2015) and SegNet (Badrinarayanan et al., 2015) architectures. We chose these image segmentation networks because image segmentation shares two important similarities with our regression problem. Pixel decisions depend heavily on the local context so there is less need to mix spatial resolutions with inception-like layers or use fully connected layers. The primary information driving decisions is geometrical in nature which makes the image segmentation networks, which use fewer parameters, more appropriate than those used in object classification. The network is composed of an initial down-sequence of levels that repeatedly decrease the width and height of the tensors and increase the number of channels, followed by a up-sequence of levels that increase the width and height of the tensors and decrease the number of channels. The down-sequence levels have skip connections to the up-sequence levels. Each level is comprised of two blocks of convolution-relu-batchnorm followed by either a maxpool or up-sample for down and up levels respectively. The target output is a 64 x 64 image of the transient bursts.

The baseline network for testing loss functions had only two up and down levels. We compared $L_1$, weighted $L_1$, and $L_2$ losses. The weighted $L_1$ loss multiplies the loss in each of the pixels with transient burst photons by a factor of 100. We train the network over one epoch of 18,000 images. Then we evaluate the different networks over the dev set. Figure 1 shows the qualitative results. The network trained with the $L_1$ loss failed to make progress on this task. The weighted $L_1$ loss does much better but predicts a lot of excess flux because it is penalized more for missing flux than assuming flux. We conclude that the $L_2$ loss is the best to use moving forward.

| | | |
|---|---|---|
| baseline<br>$2.16 \times 10^{-2}$ | no batchnorm<br>$3.42 \times 10^{-1}$ | one extra level<br>$1.72 \times 10^{-2}$ |
| two extra levels<br>$1.80 \times 10^{-2}$ | 5x5 convolutions<br>$2.86 \times 10^{-2}$ | with dropout<br>$1.52 \times 10^{-2}$ |
| 1 fewer convolution per block<br>$2.23 \times 10^{-2}$ | 0.1 learning rate<br>1.65 | 0.001 learning rate<br>$2.57 \times 10^{-2}$ |

Table 1: Each cell in this table corresponds to a different experiment we ran on the down-sampled images and corresponding network. The number in the cell is the average $L_2$ loss on the dev set.

We ran nine experiments to test how different architecture and learning parameters impact the mean $L_2$ loss on the dev set after one epoch of training. Table 1 shows our results. We found that a deeper network with dropout for regularization and a learning rate of 0.01 performs best.

We drew on our experience experimenting with networks on the down-sampled images when we designed our network for the full problem. This network consists of three down and up levels and three strategically placed dropouts with a keep probabilities of 0.9. We train this network for five epochs over our training set of 180,000 images. The average training and dev losses in the last epoch

[4]Our neural network code can be found on Github: `github.com/davidthomas5412/CS230`.

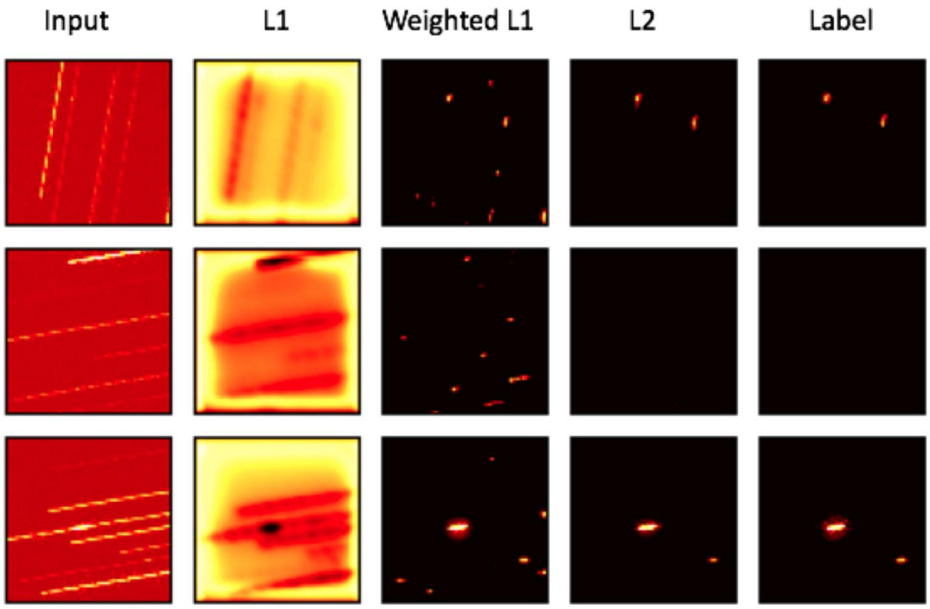


Figure 1: Each row corresponds to a different sample. The first column contain the input; the middle three columns contain the output of the networks trained with the $L_1$, weighted $L_1$, and $L_2$ losses respectively; the final column contains the label. The second column is on a separate color scale because the values are too faint to register on the color scale used for all the other images.

are $1.86 \times 10^{-2}$ and $9.01 \times 10^{-3}$ respectively. The average test loss is $1.90 \times 10^{-2}$. There is a noticeable difference between the training and dev losses. We believe this stems from the different observations and corresponding backgrounds in the training and dev and test sets.

The network described above picks out the transient bursts, but we would ultimately like to answer whether there was a detection or not. We build a binary classification network that is trained on the output from the core network over the training set. This network is constructed to find the region with the highest flux concentration and determines whether it should be considered a detection. The layers are averagepool-convolution(7x7)-maxpool-convolution(1x1)-softmax, where the first convolution increases the number of channels from 1 to 16 and the second convolution decreases the number of channels from 16 to 2. The auxiliary network uses a cross entropy loss and is trained for five epochs over our training set of 10,000 output images and evaluated on dev and test sets of 100 output images. We tried other architectures but found that this relatively simple one generalized best. The network has 98.6 % accuracy on the final epoch of training and 97 % accuracy on both the dev and test sets.

## 5 Discussion

The extraction and detection networks working in tandem achieve almost perfect accuracy. Figure 2 shows six sample images and their input, labels, and the output of the extraction network. The image triplet in the second row on the right is one of the samples the network misclassified. It is difficult to see the small burst of flux present in the label. This is an intrinsically difficult burst to extract and detect. The other 100 misclassified images we analyzed were similarly difficult. Given that many of the bursts are too difficult for us to detect by eye, and that the network fails on intrinsically difficult images, we believe there is limited room for further improvement.

A single image from the LSST is 3.6 Gigapixels - equivalent to over 1500 HDTVs. Our network operates on 512 x 512 images. We would need to apply it over 10,000 times to cover a single LSST image. If the probability that our network produces a false positive is p, then the probability that there is at least one false detection over the entire image would be greater than $1 - (1 - p)^{10,000}$. This is why it is essential to analyze and potentially mitigate the number of false positives at the expense of

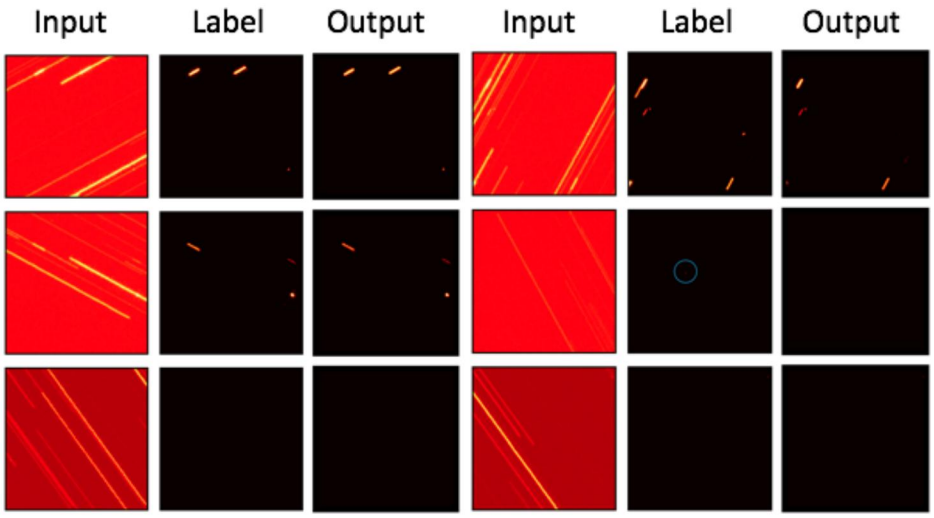


Figure 2: Six samples from the final network. The sample on the right side of the second row gets misclassified. The blue circle surrounds the small burst that is in the label but not present in the output.

false negatives. Fortunately, only 12 % of our network's misclassifications are false positives. This is already a favorable ratio of false positives to false negatives.

We perform one final test to characterize our results. We produce datasets where one of the magnitude or duration of the star and transient burst is held fixed (15th magnitude, 300 millisecond duration) while the other varies. The results are shown in Figure 3. It is incredible that there is no sign of a degradation of the performance of our networks out to an extremely faint 18th magnitude and down to an extremely brief 5 milliseconds. We never anticipated that it would work so well through this whole range, as it extends further than the network was trained on.

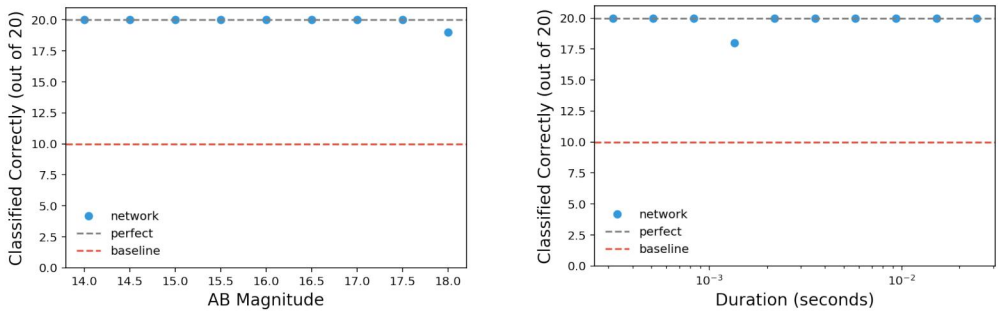


Figure 3: The performance of our network on sources of different brightness and bursts of different durations. The blue dots count the number of images at that magnitude or duration that are correctly classified out of 20.

## 6 Conclusion/Future Work

The future is bright for short duration transient astrophysics. We have presented a new method that leverages existing telescopes to scan a new time regime in the sky. We have shown that this method is very accurate out to 18th magnitude and down to 5 milliseconds. We are in the process of writing a proposal for time on the Blanco 4-m Telescope at the Cerro Tololo Inter-American Observatory. We look forward to applying this method and searching the sky for transient astrophysics.

## 7 Contributions

David Thomas implemented the code, ran the simulations, and trained and evaluated the neural networks. Professor Steven Kahn from the Stanford Physics Department advised David in this work.

## References


Badrinarayanan, V., Handa, A., & Cipolla, R. 2015, arXiv preprint arXiv:1505.07293

Chatterjee, S., Law, C. J., Wharton, R. S., et al. 2017, Nature, 541, 58, doi: 10.1038/nature20797

Connolly, A. J., Angeli, G. Z., Chandrasekharan, S., et al. 2014, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 9150, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 14

Delgado, F., Saha, A., Chandrasekharan, S., et al. 2014, in Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, Vol. 9150, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 15

Dieleman, S., Willett, K. W., & Dambre, J. 2015, MNRAS, 450, 1441, doi: 10.1093/mnras/stv632

George, D., & Huerta, E. A. 2017, ArXiv e-prints. https://arxiv.org/abs/1711.07966

Goldstein, A., Veres, P., Burns, E., et al. 2017, ApJ, 848, L14, doi: 10.3847/2041-8213/aa8f41

Harlan, E. A., & Walker, M. F. 1965, PASP, 77, 246, doi: 10.1086/128210

Hezaveh, Y. D., Levasseur, L. P., & Marshall, P. J. 2017, Nature, 548, 555, doi: 10.1038/nature23463

Howell, S. B., & Jacoby, G. H. 1986, 98, 802, doi: 10.1086/131828

Lanusse, F., Ma, Q., Li, N., et al. 2018, MNRAS, 473, 3895, doi: 10.1093/mnras/stx1665

Levin, L., Armour, W., Baffa, C., et al. 2017, ArXiv e-prints. https://arxiv.org/abs/1712.01008

Mahabal, A., Sheth, K., Gieseke, F., et al. 2017, ArXiv e-prints. https://arxiv.org/abs/1709.06257

Paszke, A., Gross, S., Chintala, S., et al. 2017

Peterson, J. R. 2014, Journal of Instrumentation, 9, C04010, doi: 10.1088/1748-0221/9/04/C04010

Peterson, J. R., Jernigan, J. G., Kahn, S. M., et al. 2015, ApJS, 218, 14, doi: 10.1088/0067-0049/218/1/14

Ronneberger, O., Fischer, P., & Brox, T. 2015, ArXiv e-prints. https://arxiv.org/abs/1505.04597

Schawinski, K., Zhang, C., Zhang, H., Fowler, L., & Santhanam, G. K. 2017, MNRAS, 467, L110, doi: 10.1093/mnrasl/slx008

Sedaghat, N., & Mahabal, A. 2017, ArXiv e-prints. https://arxiv.org/abs/1710.01422

Shallue, C. J., & Vanderburg, A. 2018, AJ, 155, 94, doi: 10.3847/1538-3881/aa9e09

Van Doorsselaere, T., Shariati, H., & Debosscher, J. 2017, ApJS, 232, 26, doi: 10.3847/1538-4365/aa8f9a

Walker, M. F. 1971, PASP, 83, 401, doi: 10.1086/129147

Zackay, B., Ofek, E. O., & Gal-Yam, A. 2016, ApJ, 830, 27, doi: 10.3847/0004-637X/830/1/27

## A Proper Image Subtraction for Trail Images

Here we extend the formalism introduced by Zackay et al. (2016) to trail images. We introduce the new kernel $K$, which corresponds to the trail a point source would create in the trail image. We update the statistical model with the changes:

$$N_{|\mathcal{H}_0} = T \otimes P_n + \epsilon_n \rightarrow T \otimes K \otimes P_n + \epsilon_n \tag{1}$$

$$N_{|\mathcal{H}_1(\alpha,q)} = T \otimes P_n + \alpha\delta(q) \otimes P_n + \epsilon_n \rightarrow T \otimes K \otimes P_n + a\delta(q) \otimes P_n + \epsilon_n \tag{2}$$

Then proceeding with the likelihood analysis we end up with the modified fourier space statistical products:

$$\hat{D} = \frac{\hat{P}_r\hat{N} - \hat{P}_n\hat{R}}{\sqrt{\sigma_n^2|\hat{P}_r|^2 + \sigma_r^2|\hat{P}_n|^2}} \rightarrow \frac{\hat{P}_r\hat{N} - \hat{P}_n\hat{K}\hat{R}}{\sqrt{\sigma_n^2|\hat{P}_r|^2 + \sigma_r^2|\hat{P}_n\hat{K}|^2}}$$

$$\hat{P}_D = \frac{\hat{P}_r\hat{P}_n\sqrt{\sigma_r^2 + \sigma_n^2}}{\sqrt{\sigma_n^2|\hat{P}_r|^2 + \sigma_r^2|\hat{P}_n|^2}} \rightarrow \frac{\hat{P}_r\hat{P}_n\sqrt{\sigma_r^2 + \sigma_n^2}}{\sqrt{\sigma_n^2|\hat{P}_r|^2 + \sigma_r^2|\hat{P}_n\hat{K}|^2}}$$

$$\hat{S} = \frac{\hat{D}\overline{\hat{P}_D}}{\sqrt{\sigma_r^2 + \sigma_n^2}}$$

These are the products we use to compute $S_{corr}$ for image subtraction between a reference image and a corresponding trail image.