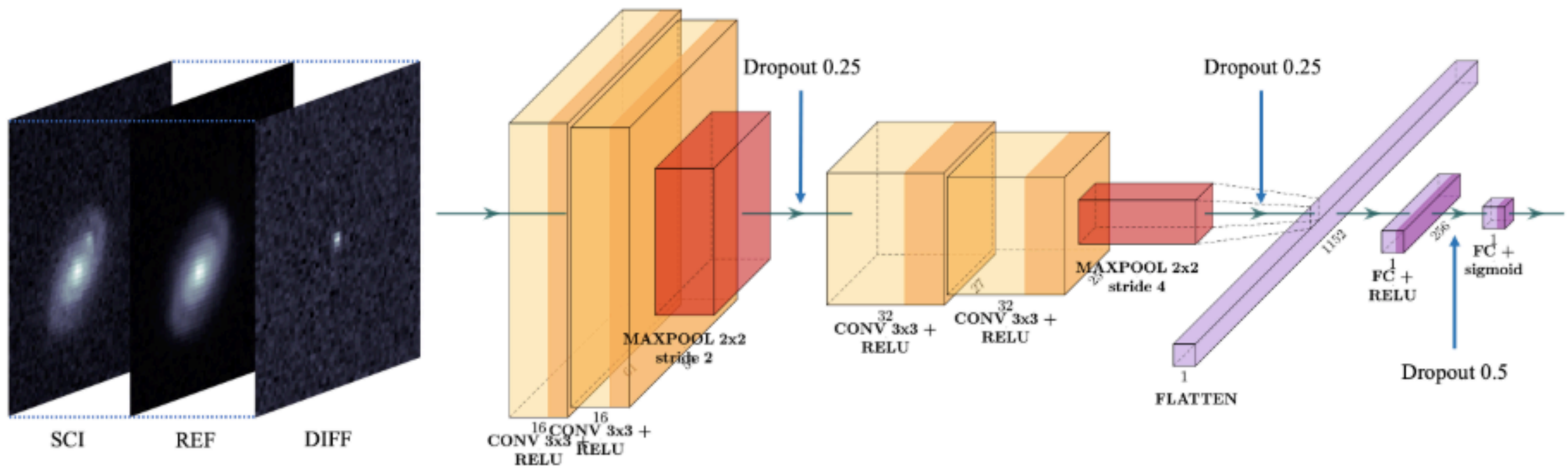
ML Status and Plans

Coordinating overall ML for ZTF
Help (and problems) always wanted
Thursday 2 PM meetings

Ashish Mahabal <aam@astro.caltech.edu>
ZTF ML Lead
Lead Computational Scientist, CD3, Caltech
ZTF Team Meeting, UW, 20190905

rb -> drb ('braai')

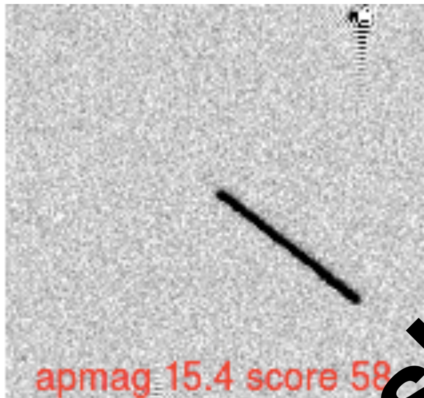
rf -> CNN



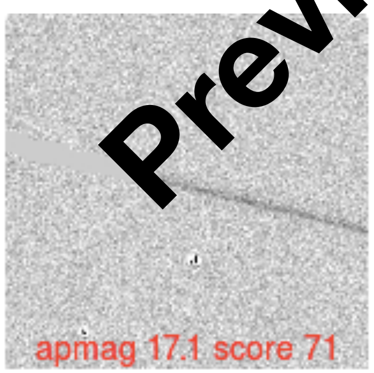
Duev et al.
Next talk

Deep Learning with AStreaks

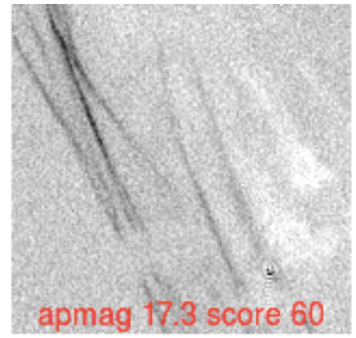
Previously



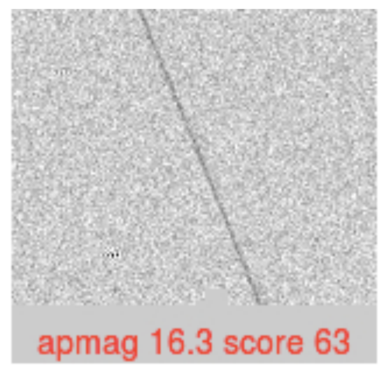
This is how a real asteroid would look. Short streak.



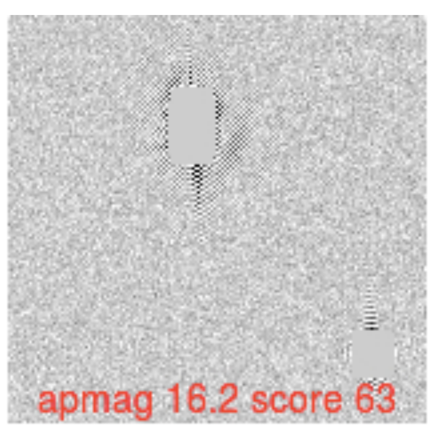
A satellite trail. Note that part of it is masked out, and the unmasked trail is longer.



These are ghosts and dementors



Another satellite trail



A masked bright star

What kind of streak do you see?

- Asteroid (short streak)
- Satellite (long streak - could be partially masked)
- Masked bright star
- Dementors and ghosts
- Cosmic rays
- Naked stars
- Yin-Yang (multiple badly subtracted stars)
- Skip (Includes 'Not Sure' and seemingly 'Blank Images')

Need some help with this task?

Done

Show the project tutorial

DeepStreaks

CNNs

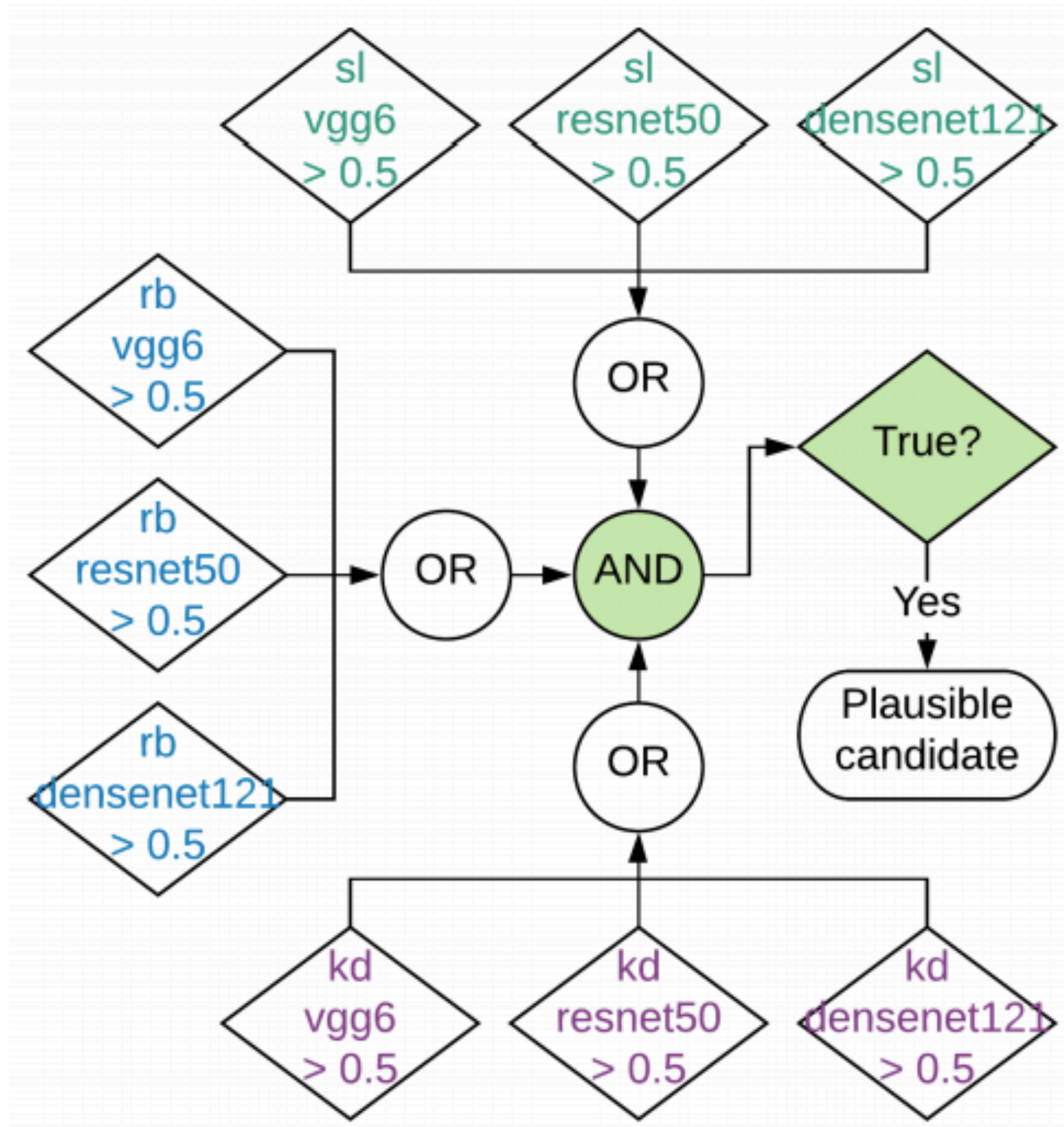
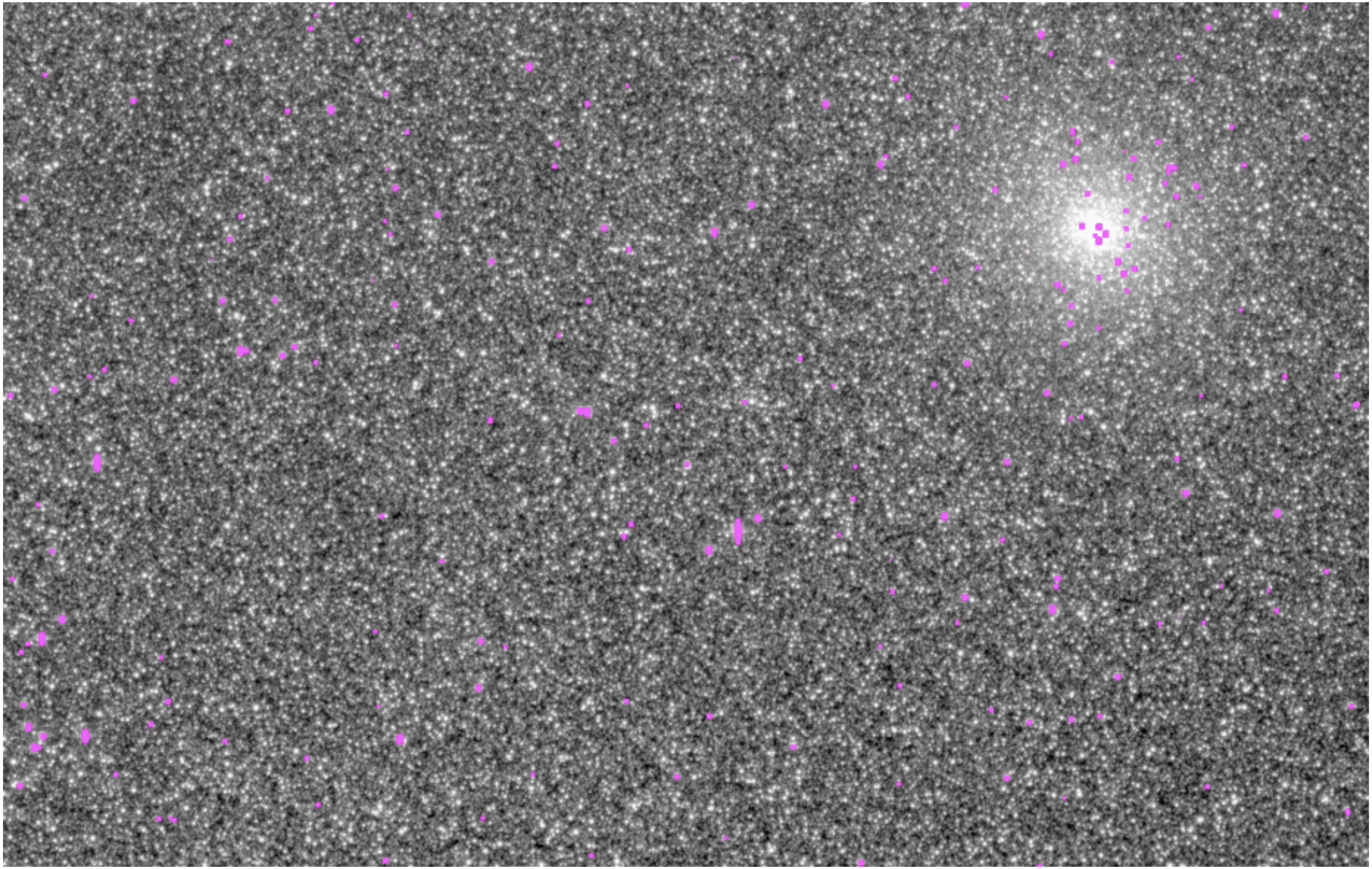


Figure 2. Decision logic used by **DeepStreaks** to identify plausible streaks. The problem is split into three simpler sub-problems, each solved by a dedicated group of classifiers assigning real vs. bogus ("rb"), short vs. long ("sl"), and keep vs. ditch ("kd") scores. At least one member of each group must output a score that passes a pre-defined threshold. See Section 2.1 for details.

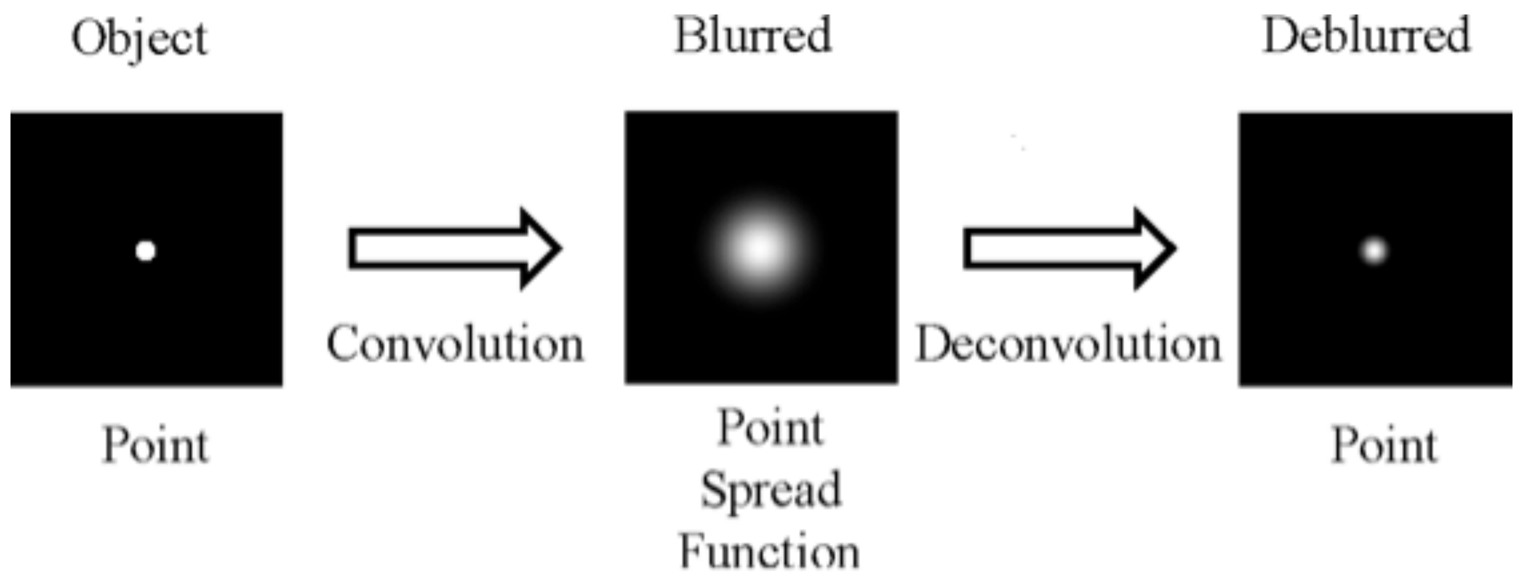
Duev et al.
Next talk

Crowded field (shown by Frank)



Deconvolution

SURF project
Shubhranshu Singh (IIT-G)
Mentor: A Mahabal
Co-mentor: D Duev



If the PSF is known:

- Wiener Filtering
  - Based on Fourier transform of the signal, the PSF, and the noise
- CLEAN
  - Approximate point sources by delta-functions
- Maximum Entropy
  - Fits the data with maximum entropy
- Richardson-Lucy
  - Iterative method - Bayesian methodology

If the PSF is unknown - Deep Learning

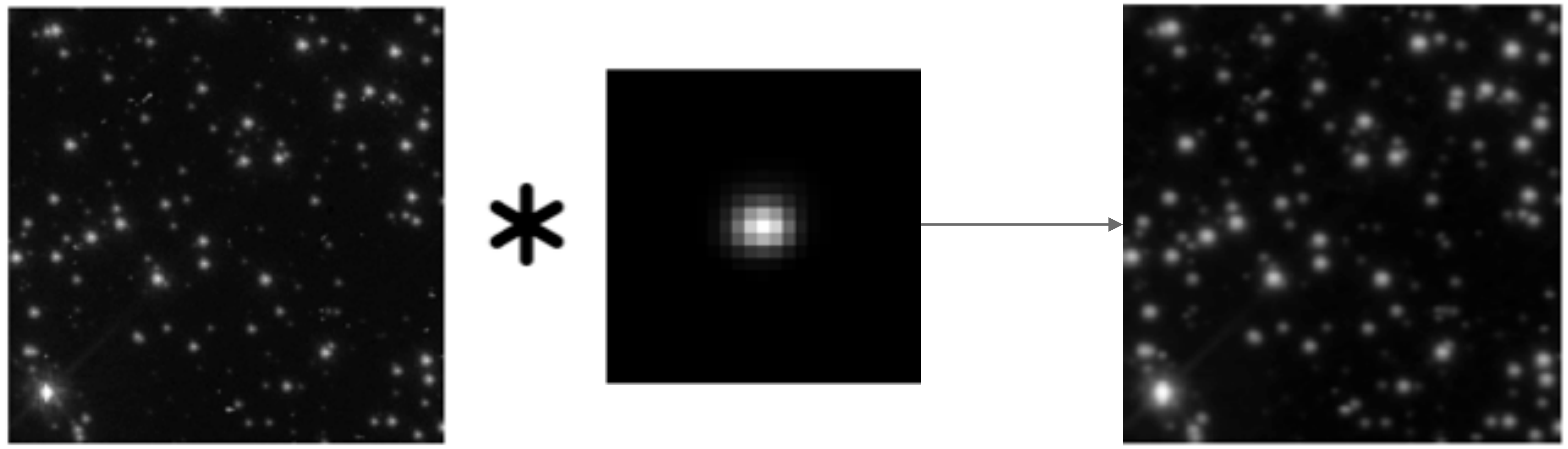


Image pairs of convolved and clean images required

Clean images: HST's WFC3 (normalised to [0, 1])
Convolved images - use ZTF PSF

HST data (For training)
- 2K by 4K FITS images
- UV filter with 32 bits per pixel

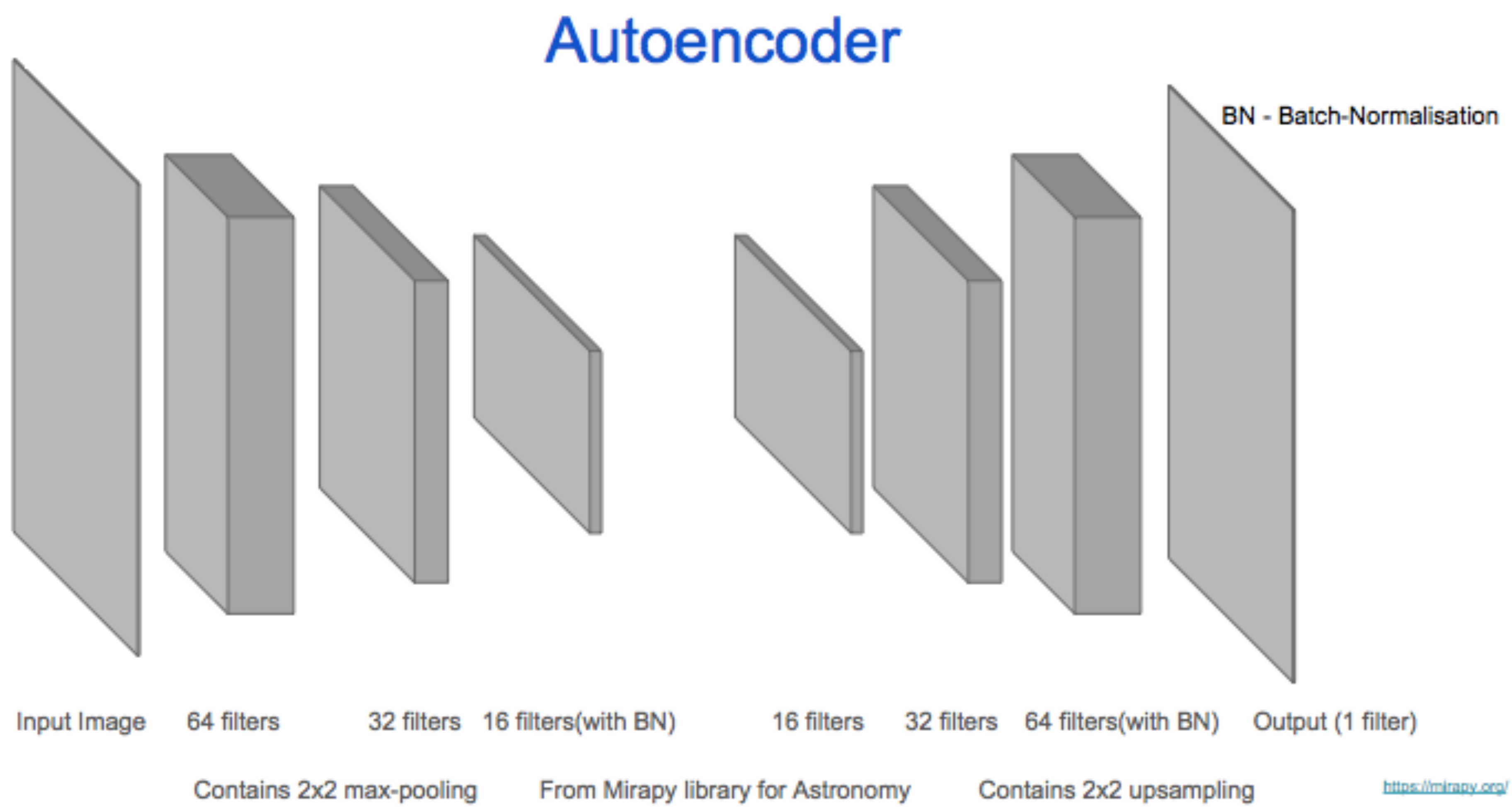
ZTF data (For testing)
- 3K by 3K FITS images
- g/r filter with 32 bits per pixel

During training - cropped images used

Simulated data - Using PyRAF mkobjects and starlist commands

1000, 256x256 images with 5-15 gaussian sources
Background with poisson noise

Shubhranshu Singh



Input images - 64x64, 256x256 and 512x512 for different experiments.
500 epochs, Adam optimizer, learning rate of 10^{-4} (for 256x256 images)

	Training set	Validation set	Test set
HST data	1450 images	162 images	180 images
Simulated data	810 images	90 images	100 images

Shubhranshu Singh

Typically used - L2-norm and L1-norm
Modified loss function - mixture of L1-norm and multi-scale structure similarity (MS-SSIM)
MS-SSIM - Calculated using mean and variance of the data, at various scales.

Zhao, Hang, et al. "Loss functions for image restoration with neural networks."
Wang, Zhou, Eero P. Simoncelli, and Alan C. Bovik. "Multiscale structural similarity for image quality assessment."

$$\mathcal{L}^{MS-SSIM}(I) = 1 - \text{MS-SSIM}(I)$$

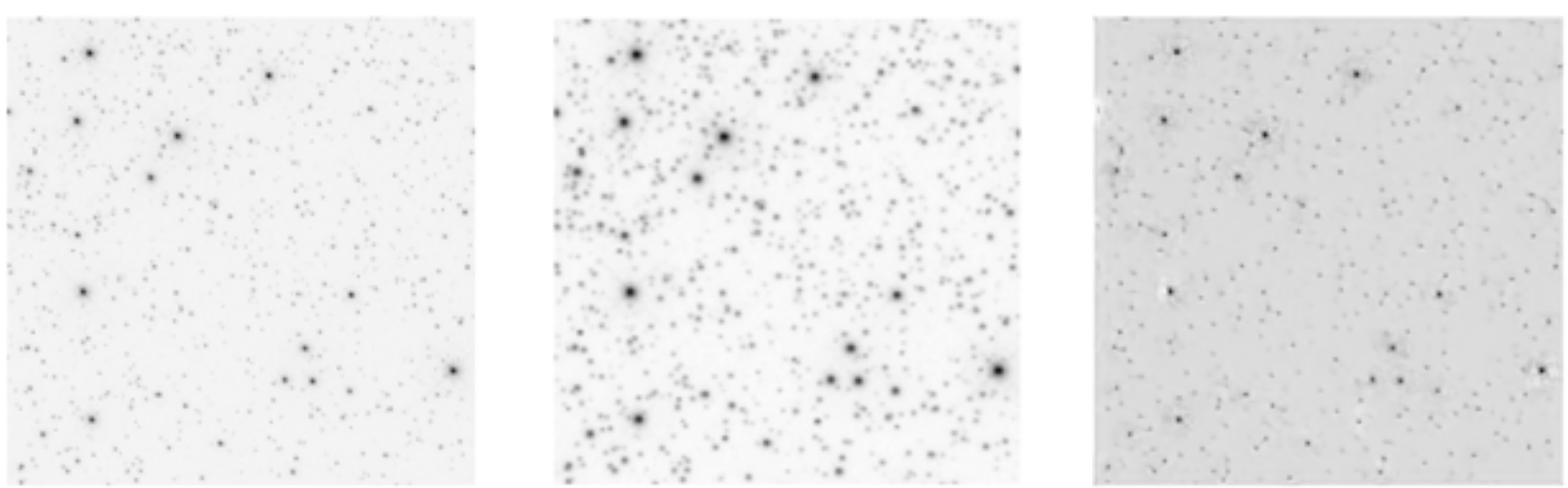
$$\mathcal{L}^{Mix} = 0.8 \cdot \mathcal{L}^{MS-SSIM} + 0.2 \cdot \mathcal{L}^{\ell_1}$$

Results on Simulated Data

	Convolved Image	**Original model**	**Added layer**	**Deeper Model**
Average PSNR	43.99 dB	48.5 dB	51.74 dB	47.83 dB
Max. PSNR	46.80 dB	51.23 dB	55.03 dB	51.84 dB
Min. PSNR	40.11 dB	37.87 dB	40.06 dB	33.6 dB

PSNR - Peak Signal-to-Noise Ratio

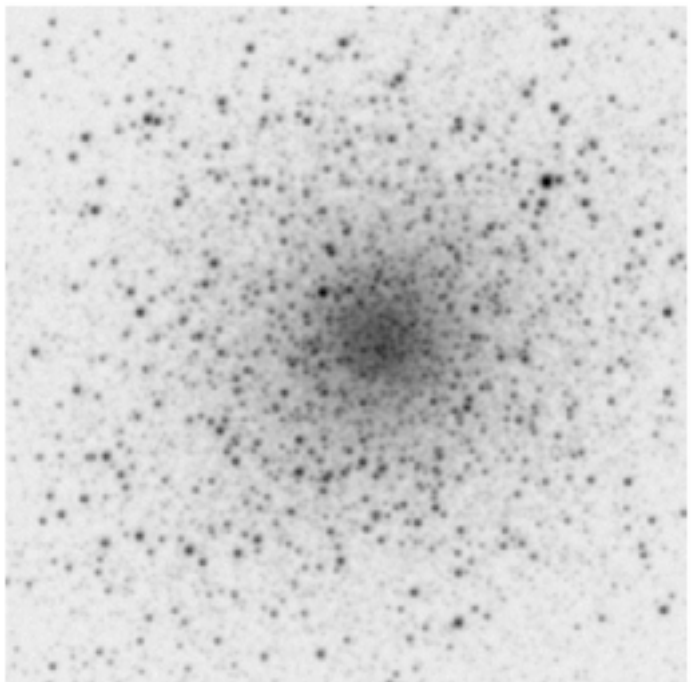
HST data + ZTF PSF



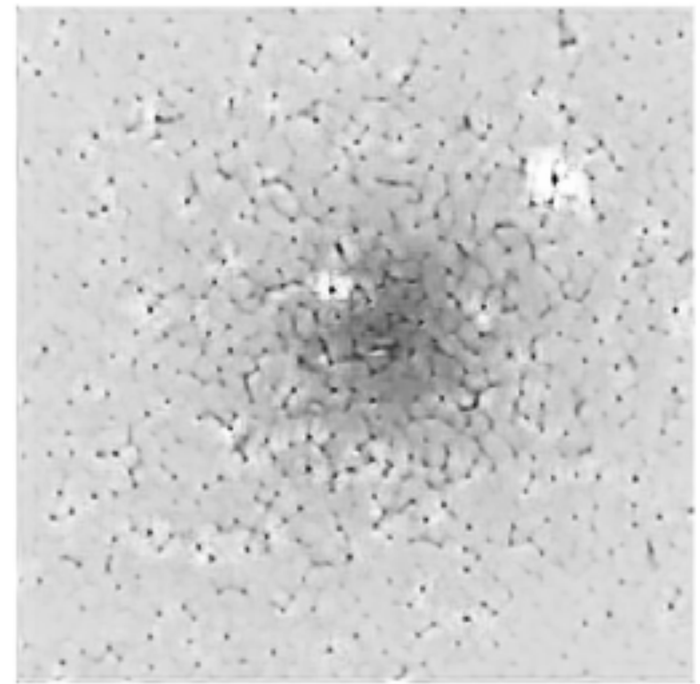
deconvolved

Shubhranshu Singh

Autoencoder results

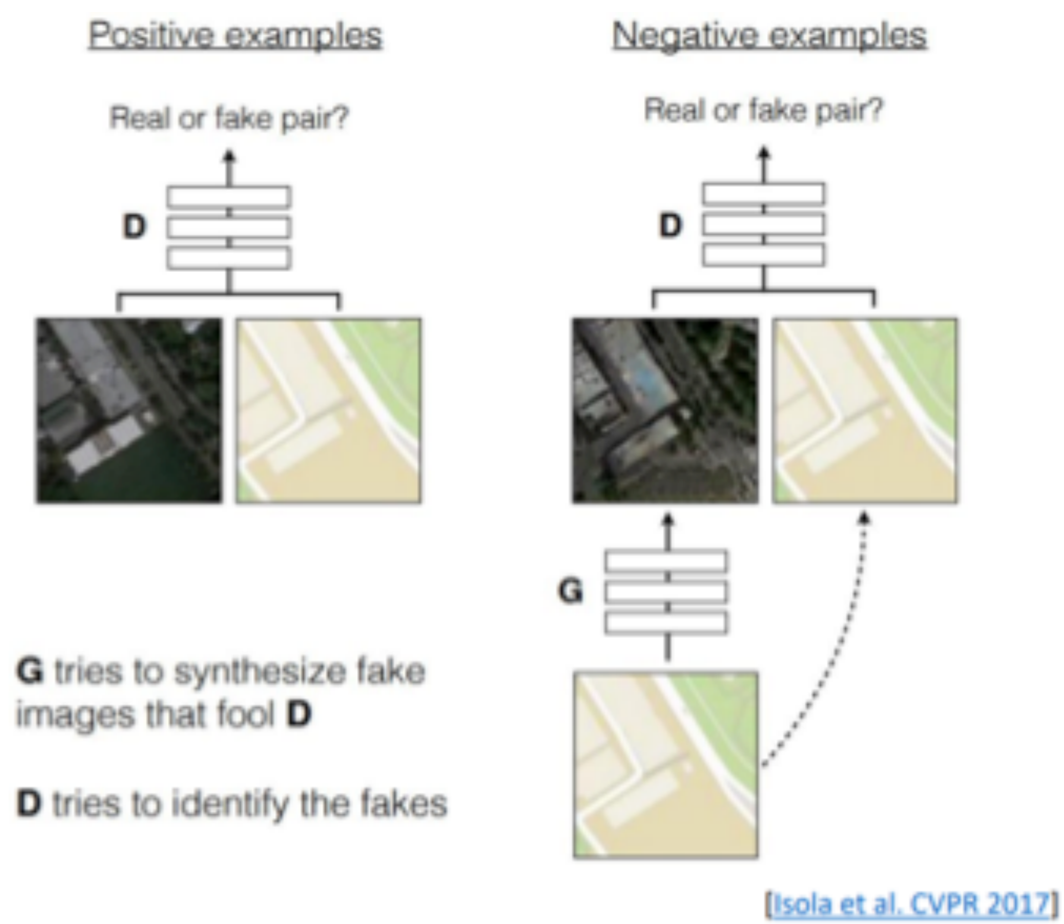


ZTF image

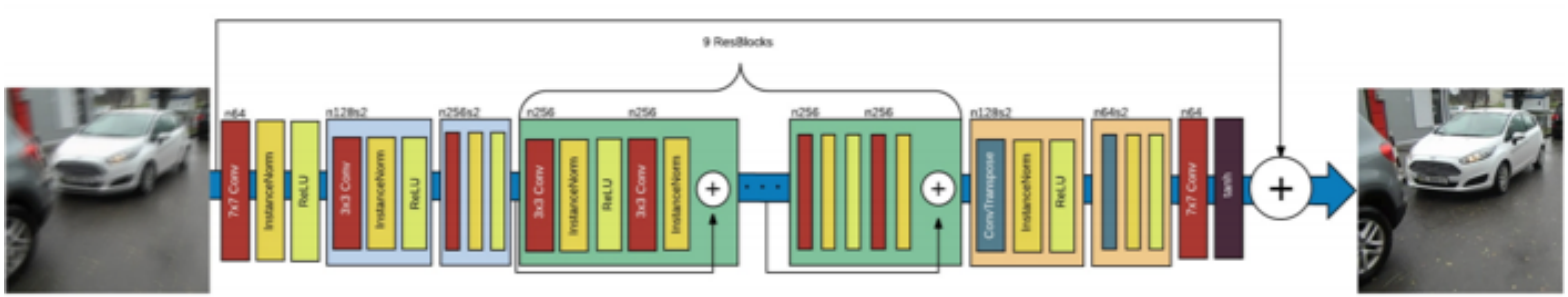


Shubhranshu Singh

Generative Adversarial Network (GAN)



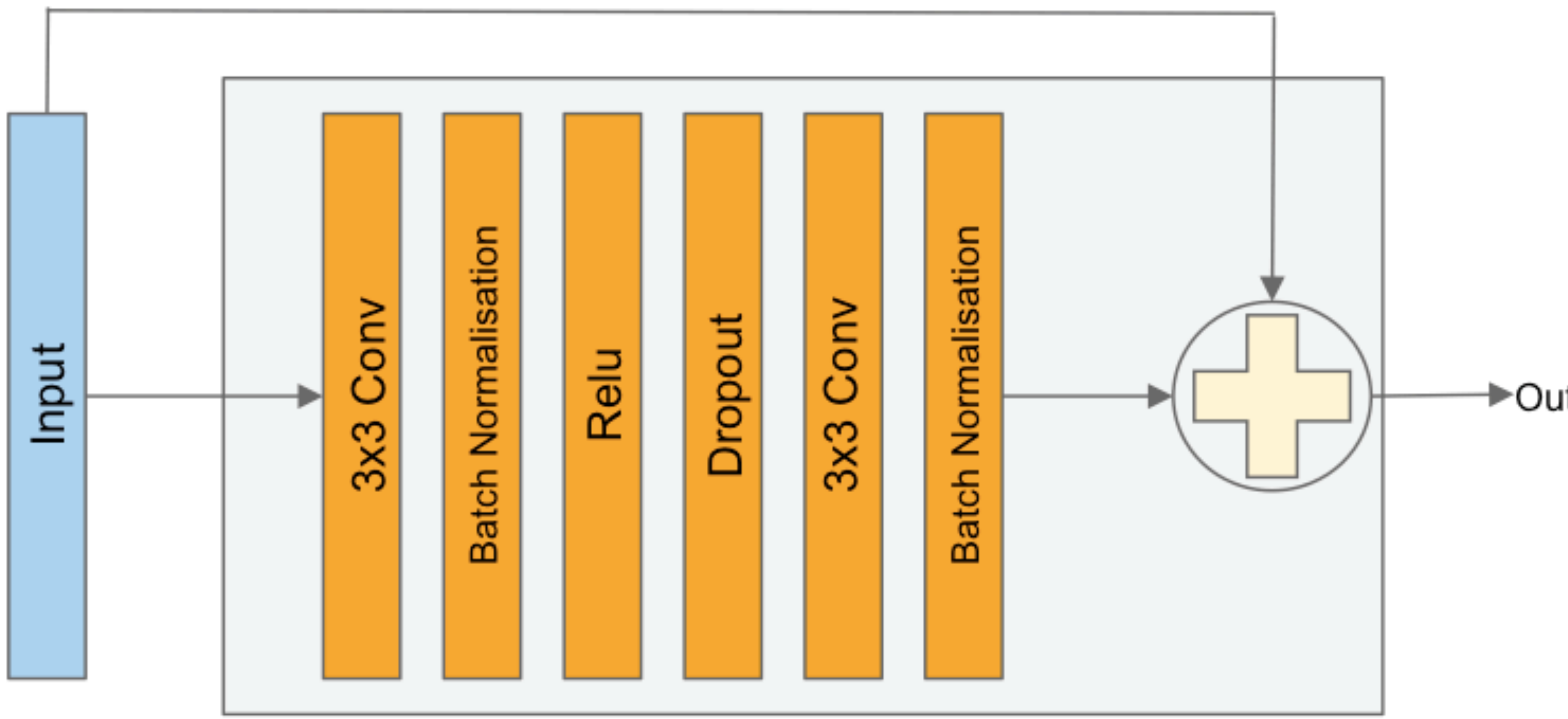
Convolved deconvolved



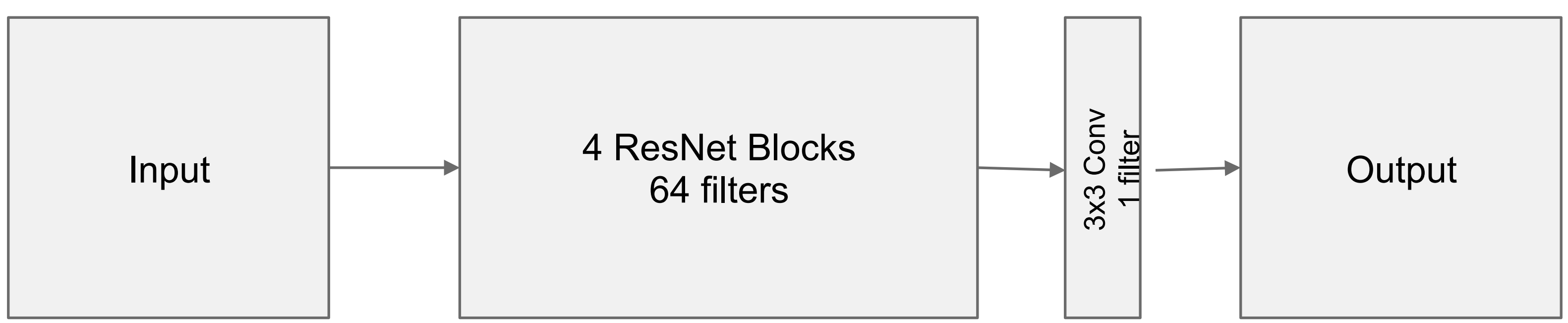
deblurGAN

Shubhranshu Singh

ResNET

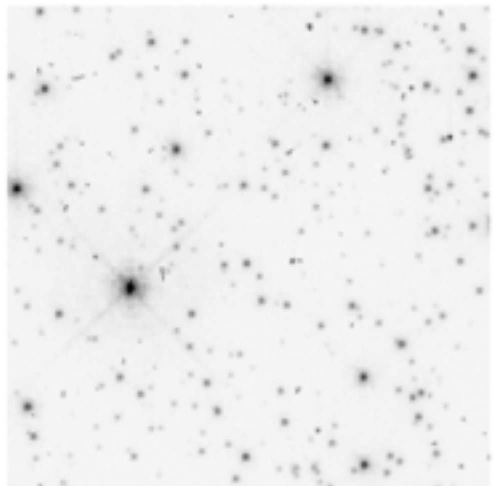


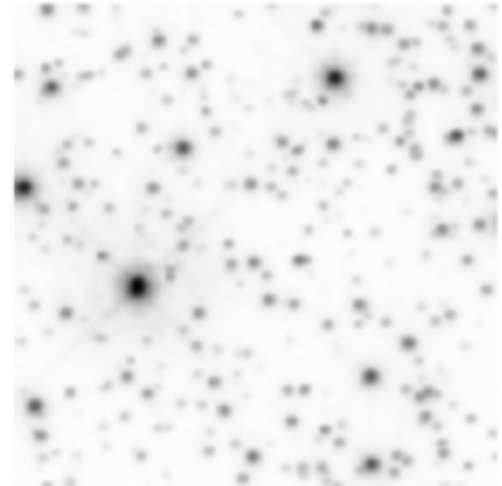
He, Kaiming, et al. "Deep residual learning for image recognition."



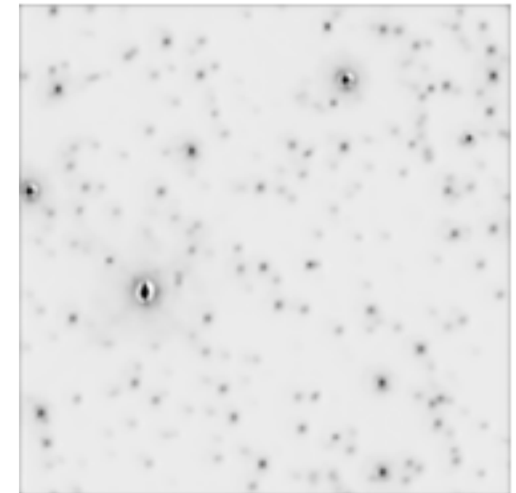
Total parameters - 261,761

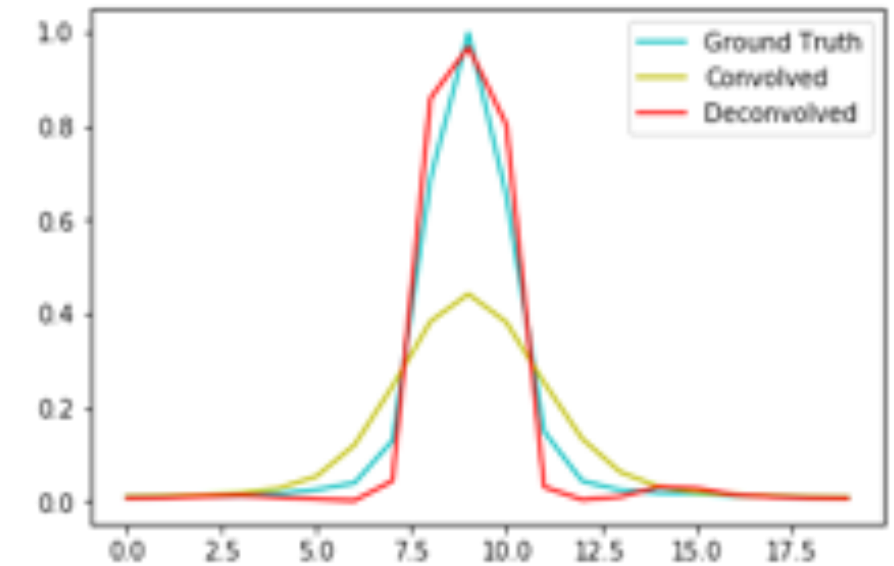
Shubhranshu Singh

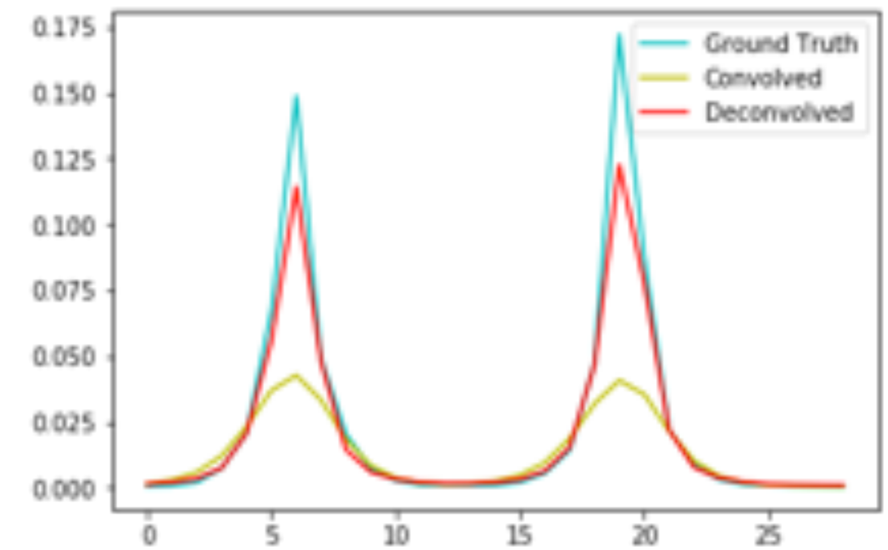


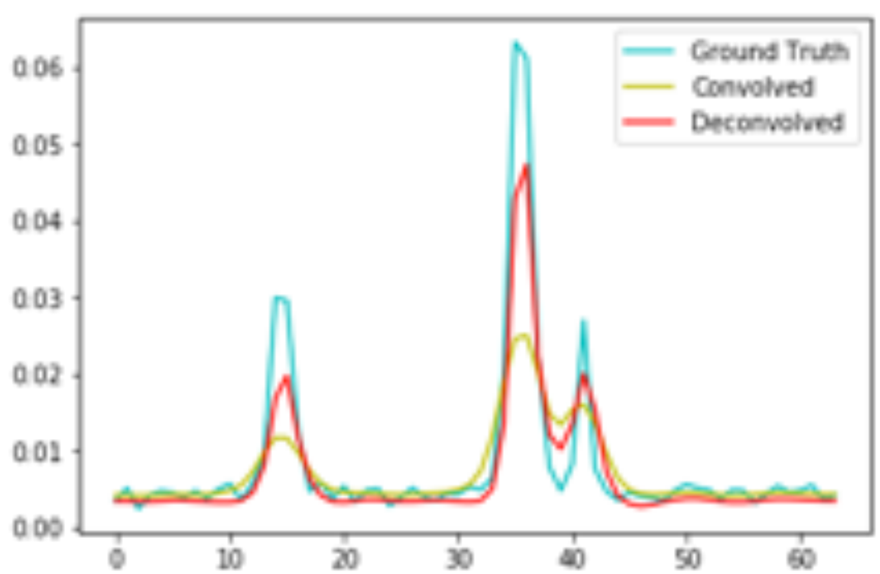


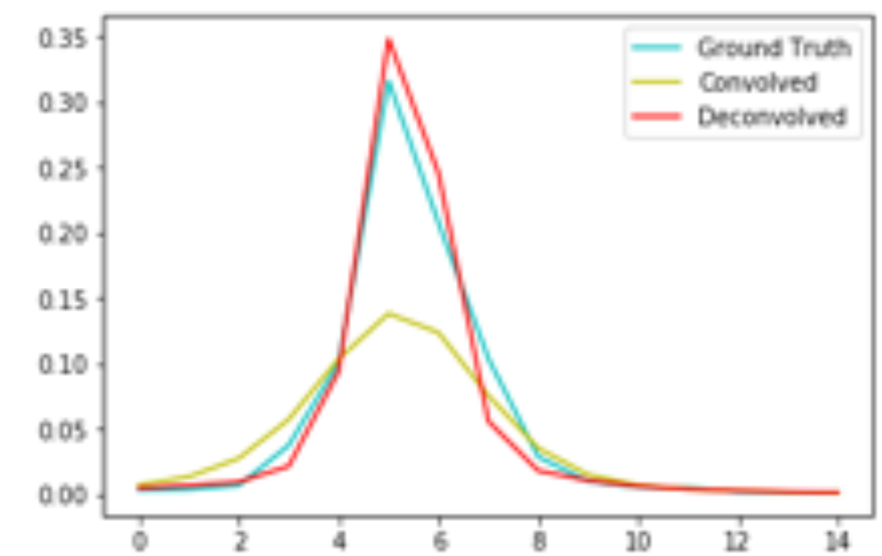
deconvolved



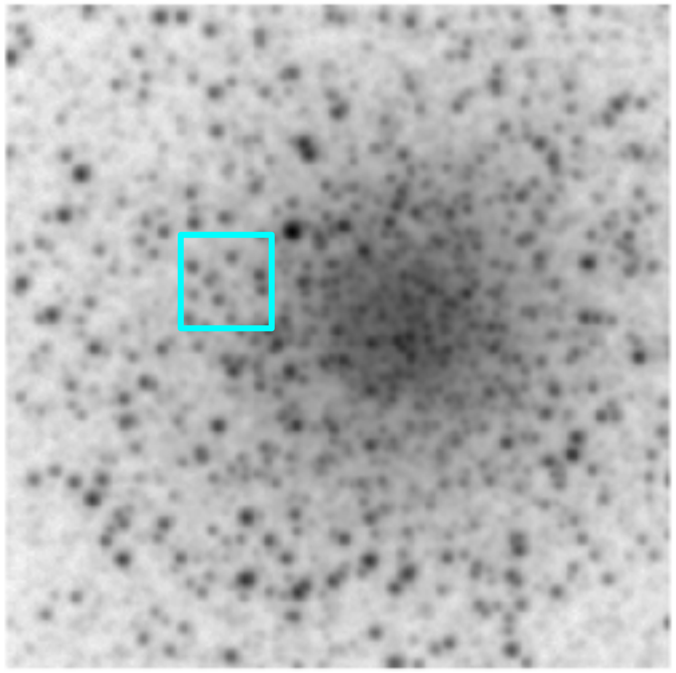


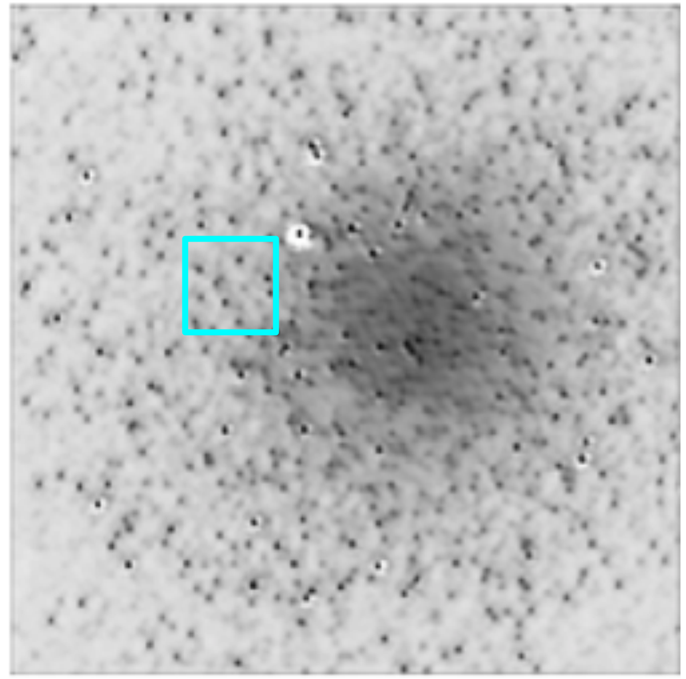


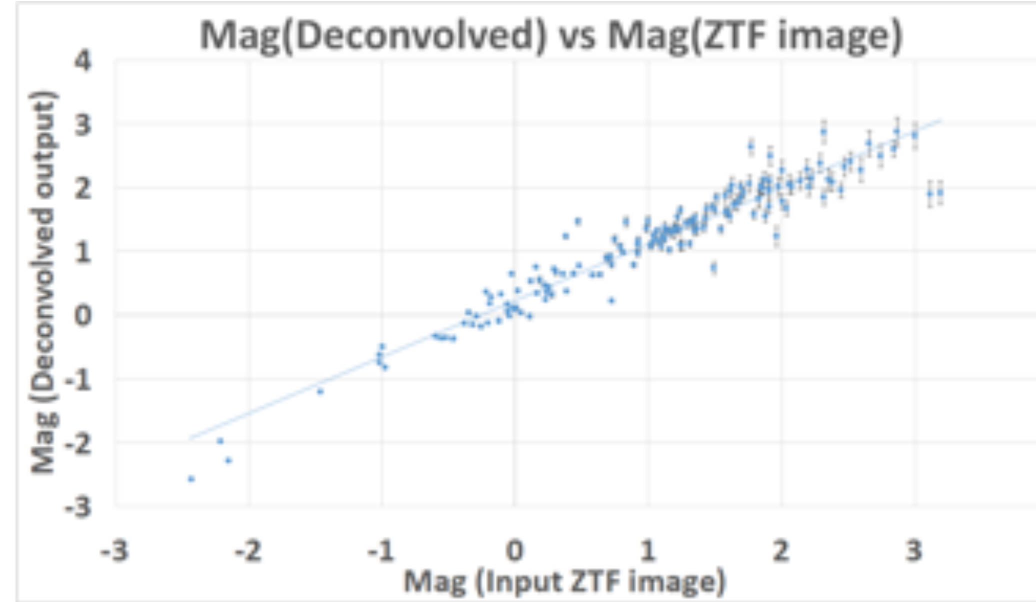


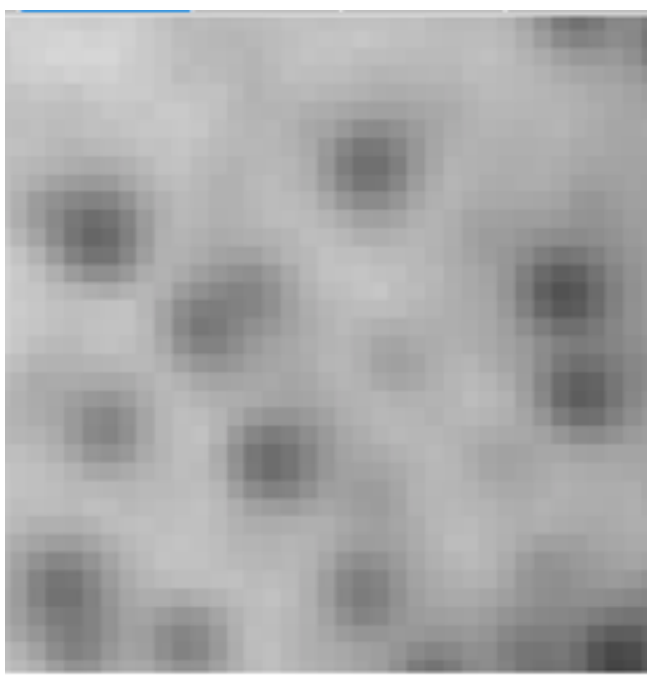


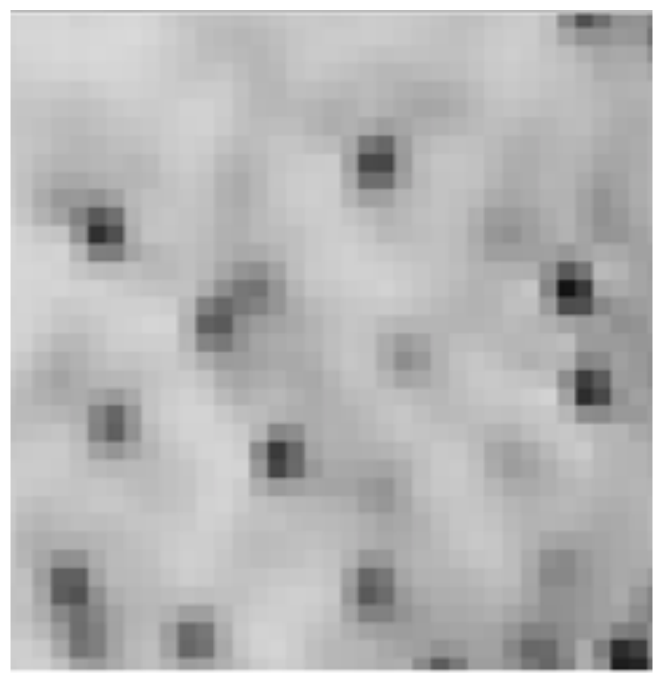
Shubhranshu Singh

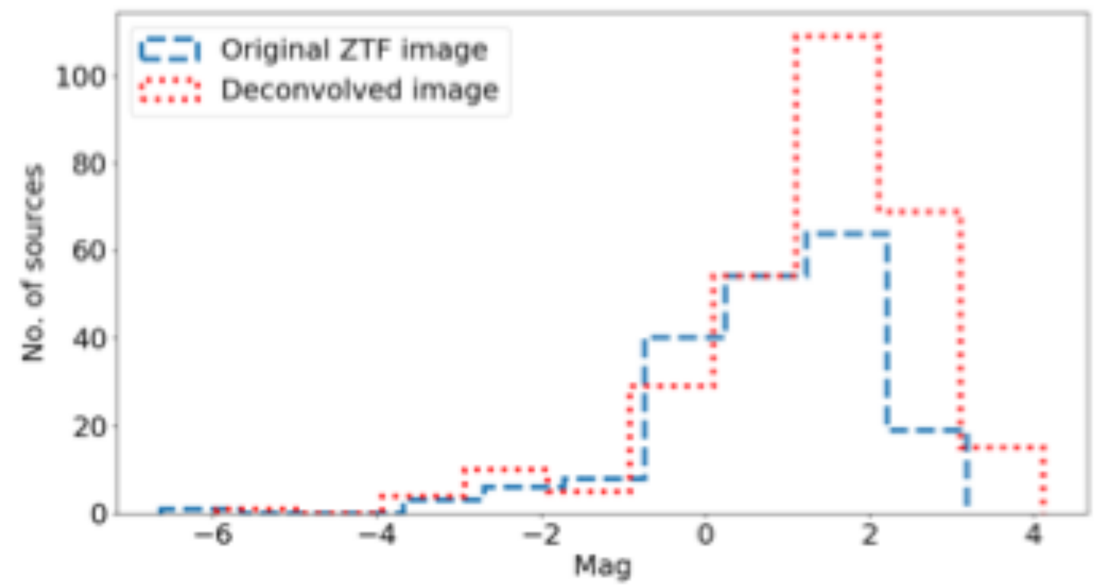










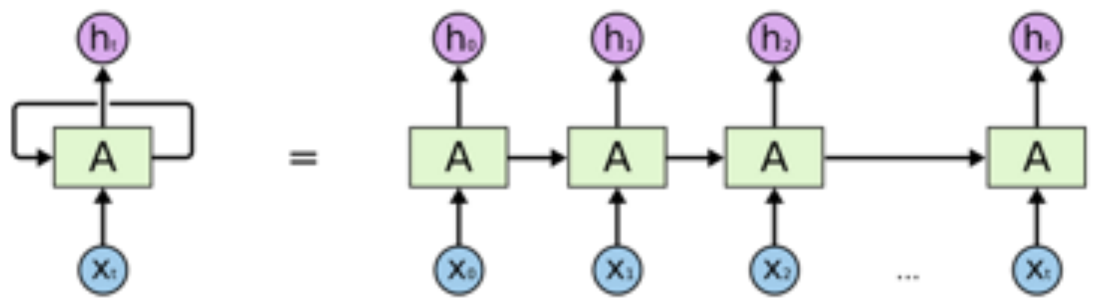


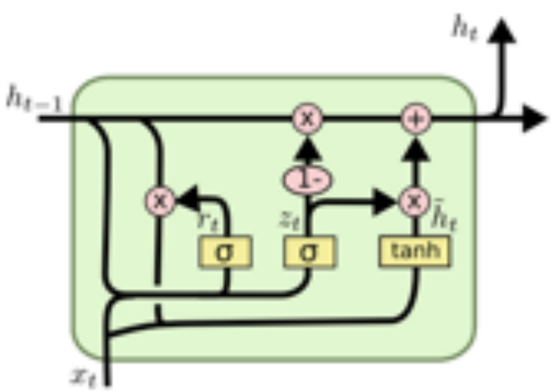
Shubhranshu Singh

Remove artifacts near bright sources
Use a better PSF model
Improve the ResNet architecture
Make a pipeline for processing 3k x 3k ZTF images

RNN - Vinu (ZTF)

SURF project
Vinu Sankar (IIT-G)
Mentor: A Mahabal
Co-mentor: M Graham





$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Image courtesy: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Wavenet

Dilation
Long range
But needs to be deep.
Seq2seq

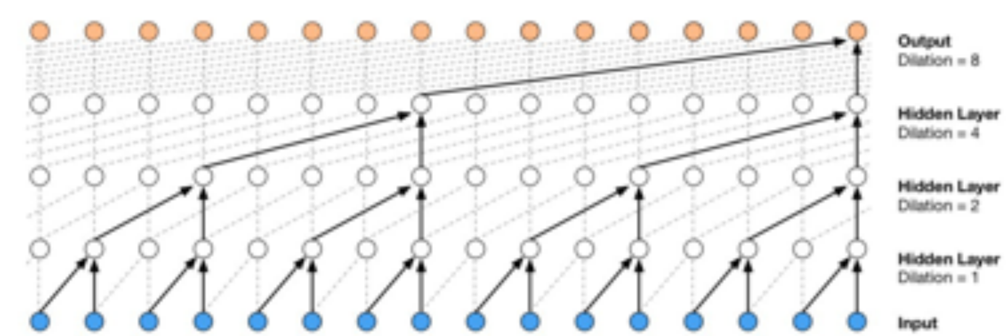
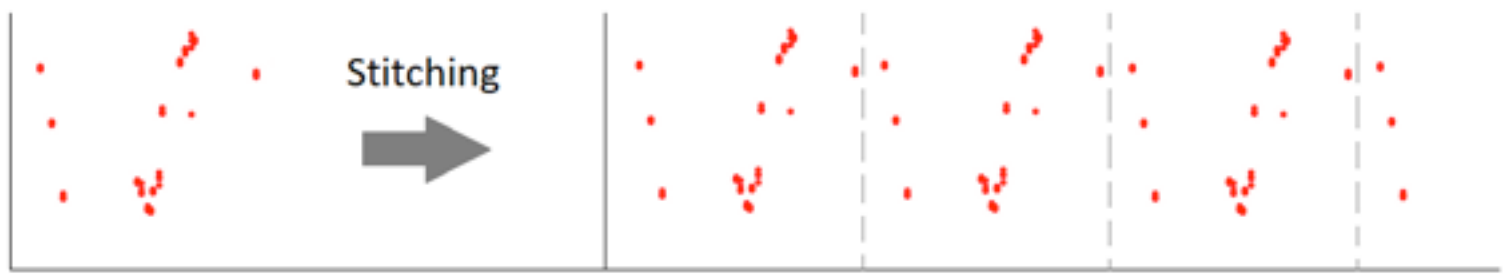


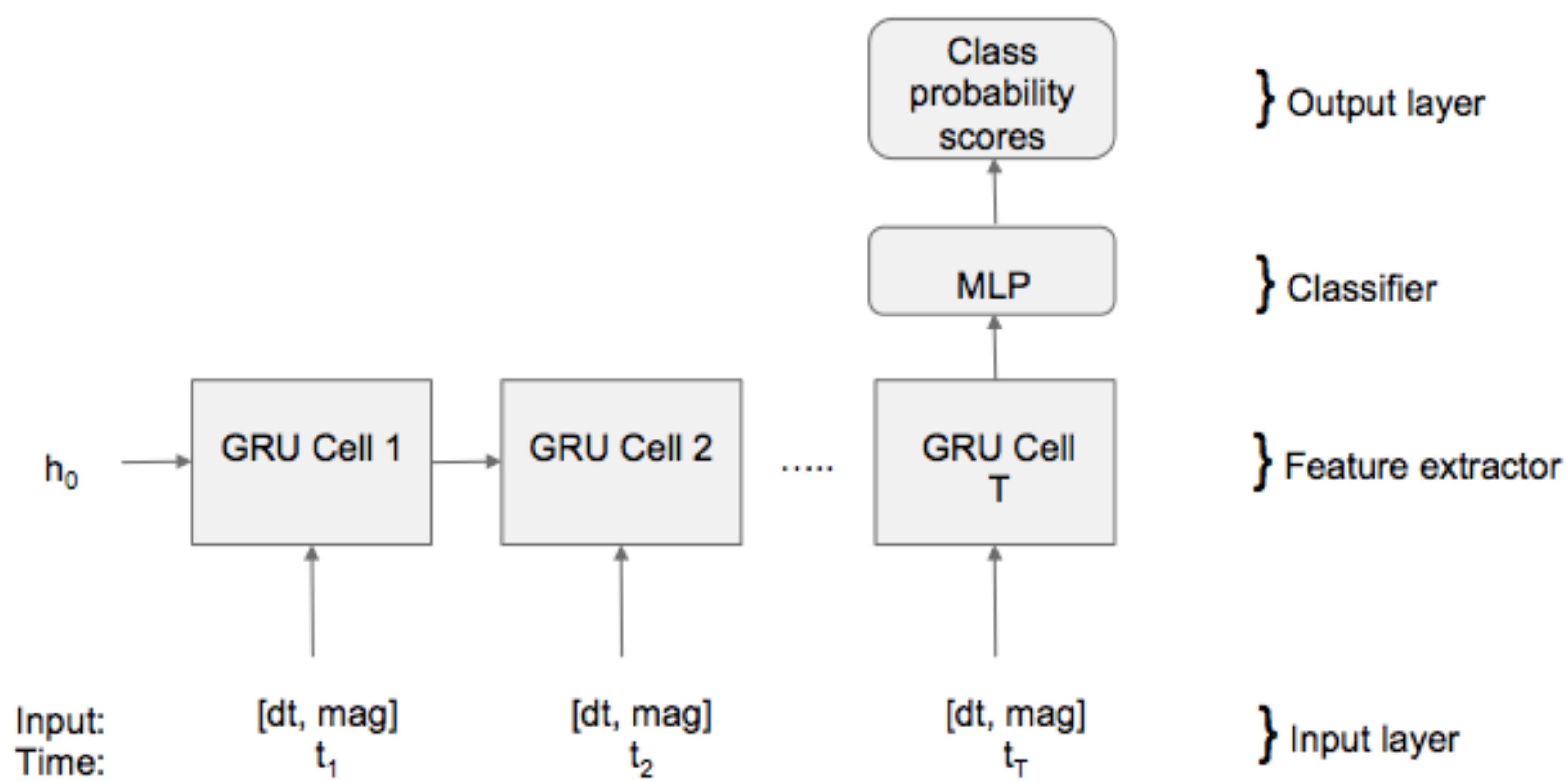
Figure 3: Visualization of a stack of *dilated* causal convolutional layers.

Error associated in mag.
Irregular temporal gaps, sparse data.
Padding inputs for RNNs.
RNNs tend to forget long lcs.

Tackling challenges



dt as input to RNN.
Use GRUs or LSTMs[1] instead of vanilla RNN cell.
Drop points with dt > 4 months.
Stitching instead of padding.



Vinu Sankar

Static RNN models

- Zero-padding
- Stitching
- GRUCell, FastGRNNCell, various hidden units 16, 32, 64, 128

Dynamic RNN models

- Bucketing in batches
- Masking

Various inputs

- [dt, mag], [dt, dm], [$t-t_0$, dt, dm, mag, magerr], [dt, mag, dm/dt, $t-t_0$]…
- Inputs tried with/without normalizing
- Drop points with dt > 4 months

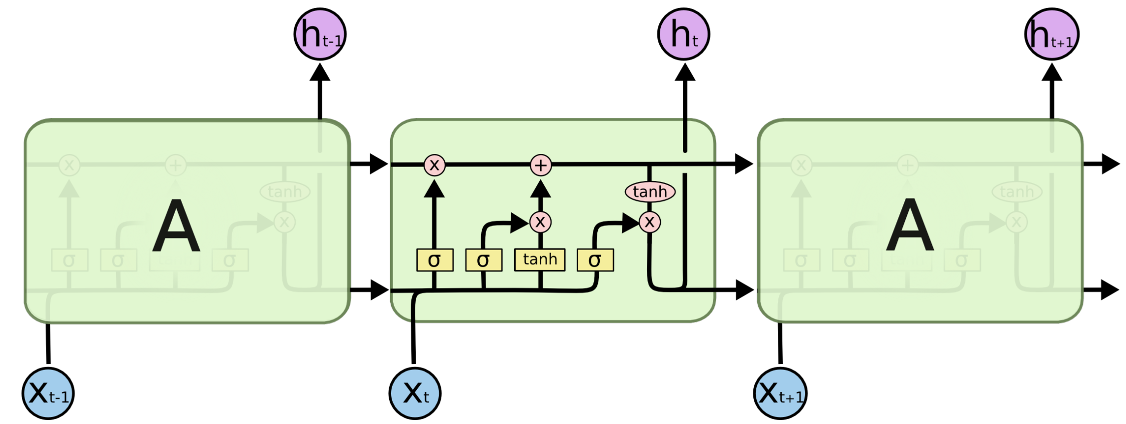
Best model:

Static RNN with stitching, drop dt>120, input [dt, mag] (normalized)

Vinu Sankar

RNNs and delta-ts

using for multiple filters



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

$\boldsymbol{X}$: Input time series (2 variables);
$\boldsymbol{s}$: Timestamps for $\boldsymbol{X}$;

$\boldsymbol{M}$: Masking for $\boldsymbol{X}$;
$\boldsymbol{\Delta}$: Time interval for $\boldsymbol{X}$.

$$\boldsymbol{X} = \begin{bmatrix} 47 & 49 & NA & 40 & NA & 43 & 55 \\ NA & 15 & 14 & NA & NA & NA & 15 \end{bmatrix} \qquad \boldsymbol{M} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{s} = \begin{bmatrix} 0 & 0.1 & 0.6 & 1.6 & 2.2 & 2.5 & 3.1 \end{bmatrix} \qquad \boldsymbol{\Delta} = \begin{bmatrix} 0.0 & 0.1 & 0.5 & 1.5 & 0.6 & 0.9 & 0.6 \\ 0.0 & 0.1 & 0.5 & 1.0 & 1.6 & 1.9 & 2.5 \end{bmatrix}$$

Figure 2. An example of measurement vectors x_t, time stamps s_t, masking m_t, and time interval δ_t.

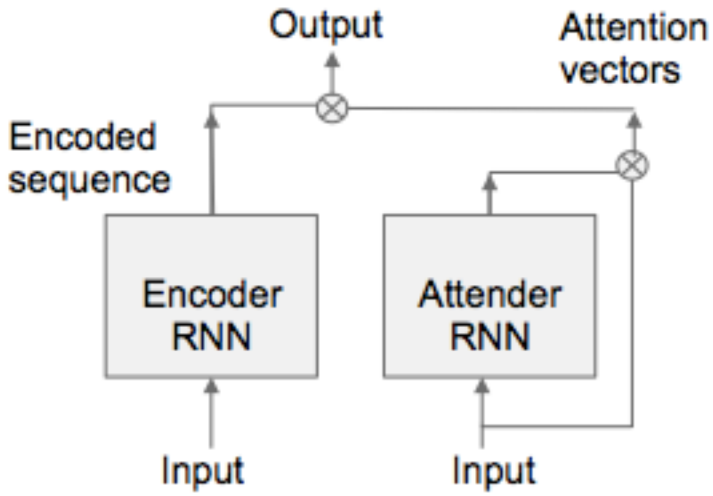
Che et al. 2018 Nature DOI:10.1038/s41598-018-24271-9

Naul et al. (encoder-decoder)

Best model: Static RNN with stitching, drop dt>120, input [dt, mag] (normalized)

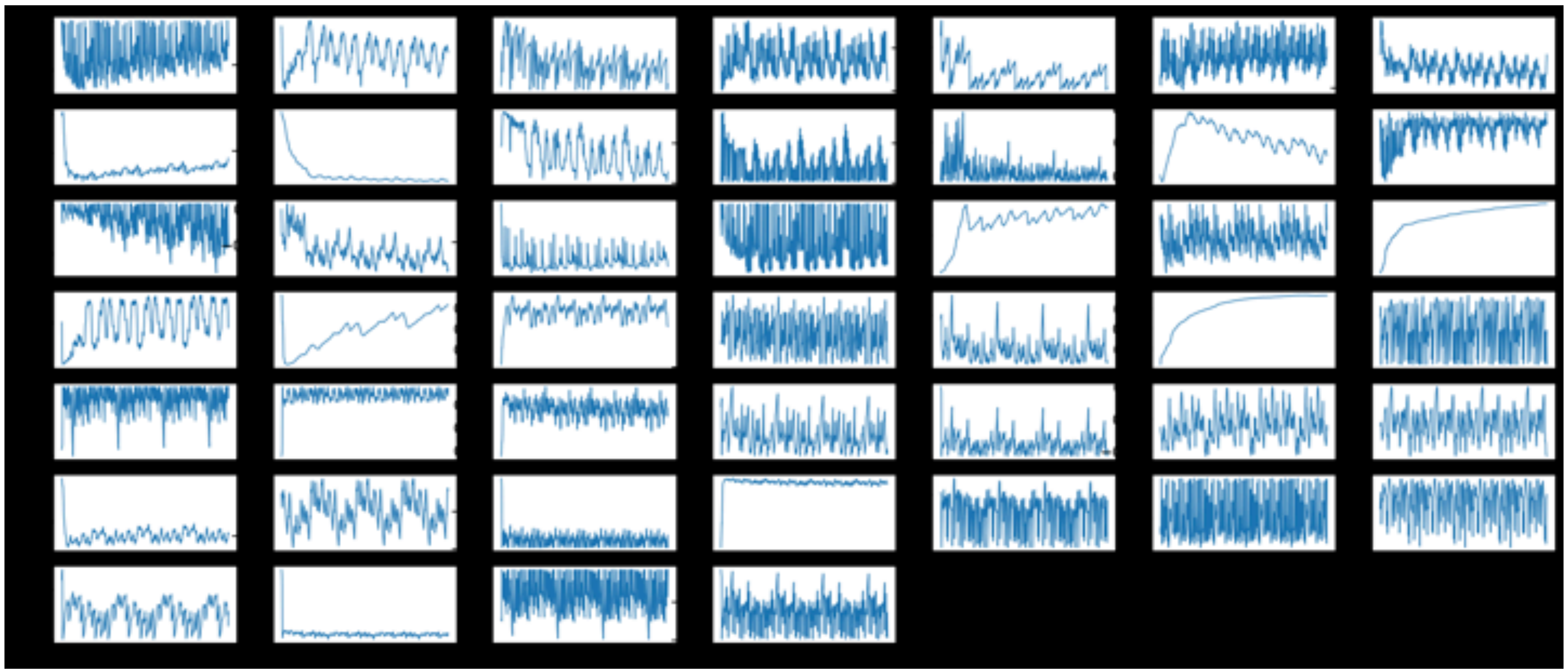
96% for easy classes

Should certain delta-ts be ignored?



Time Steps

Activation



Using transformers[1]
Combining CNN + RNN
Train on ZTF + CRTS
Test on ZTF

Vinu Sankar

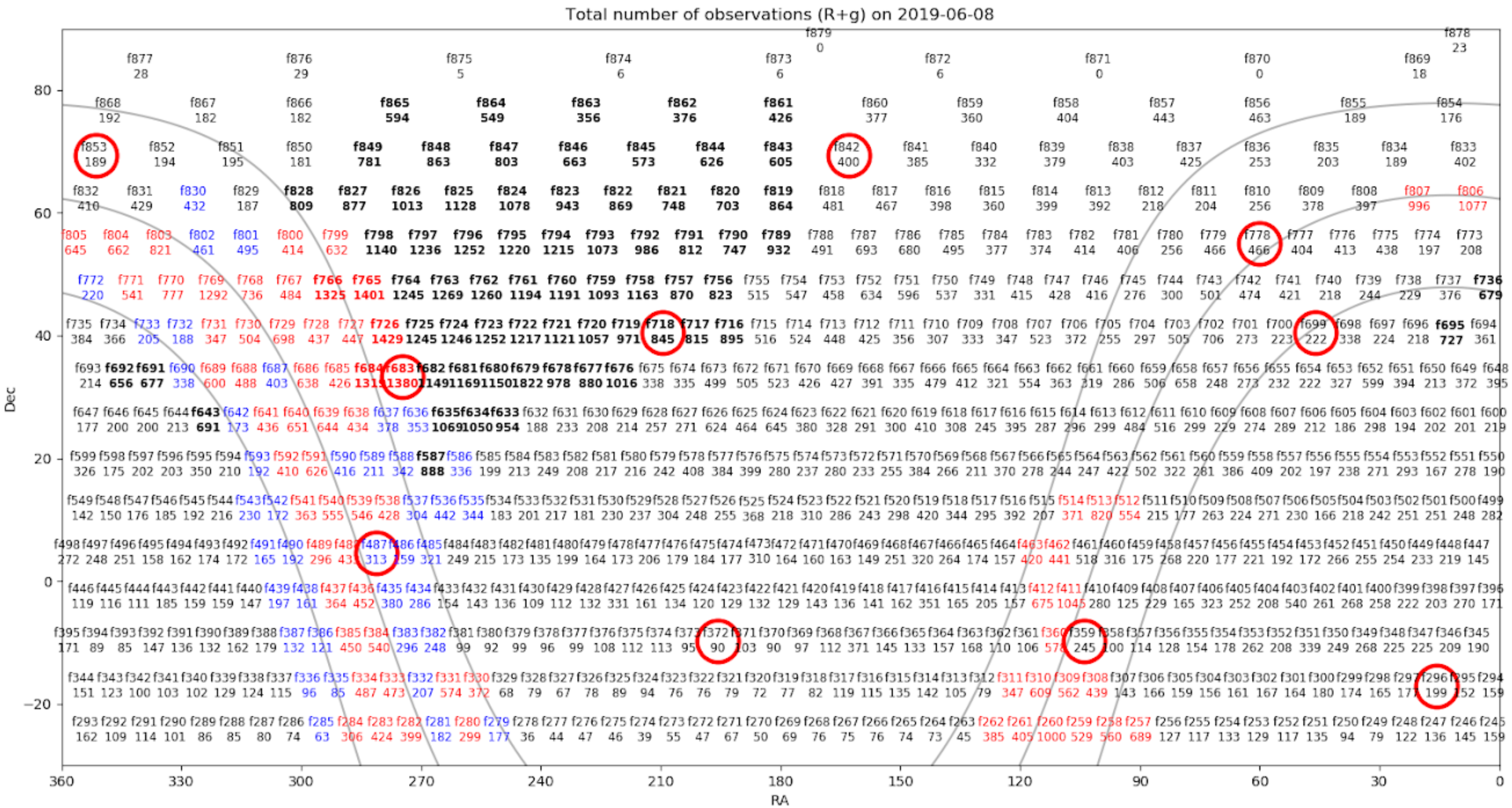
Plans

- Classification: Variable sources [+architecture] (JKM, UW)
- Visualization to understand/improve transient classification
- BTS (Adam Miller)
- Deep Coadds (Danny Goldstein)
- Zooniverse with light curves (Richard Walters)
- RNN (visiting postdocs)
- Transfer learning (UNC)
- Asteroid light curves? (Rex)
- Babamul [broker]
- Gaia/ALERCE synergy

Help always needed
Thursday 2 PM meetings
Coordinating overall ML for ZTF

aam@astro.caltech.edu

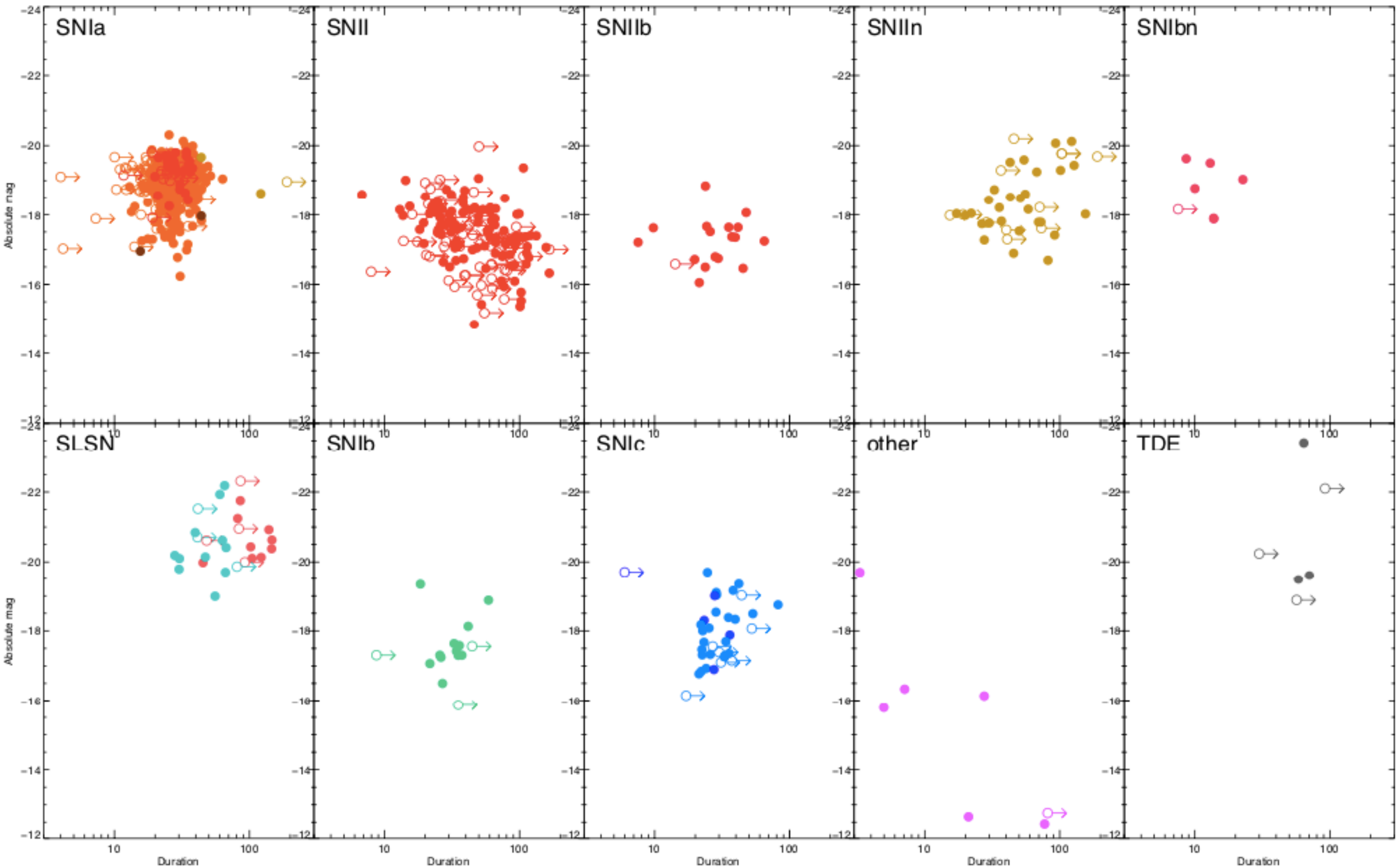
10 fields (Jan)



features/variability -> classification

[Hackday possibility]

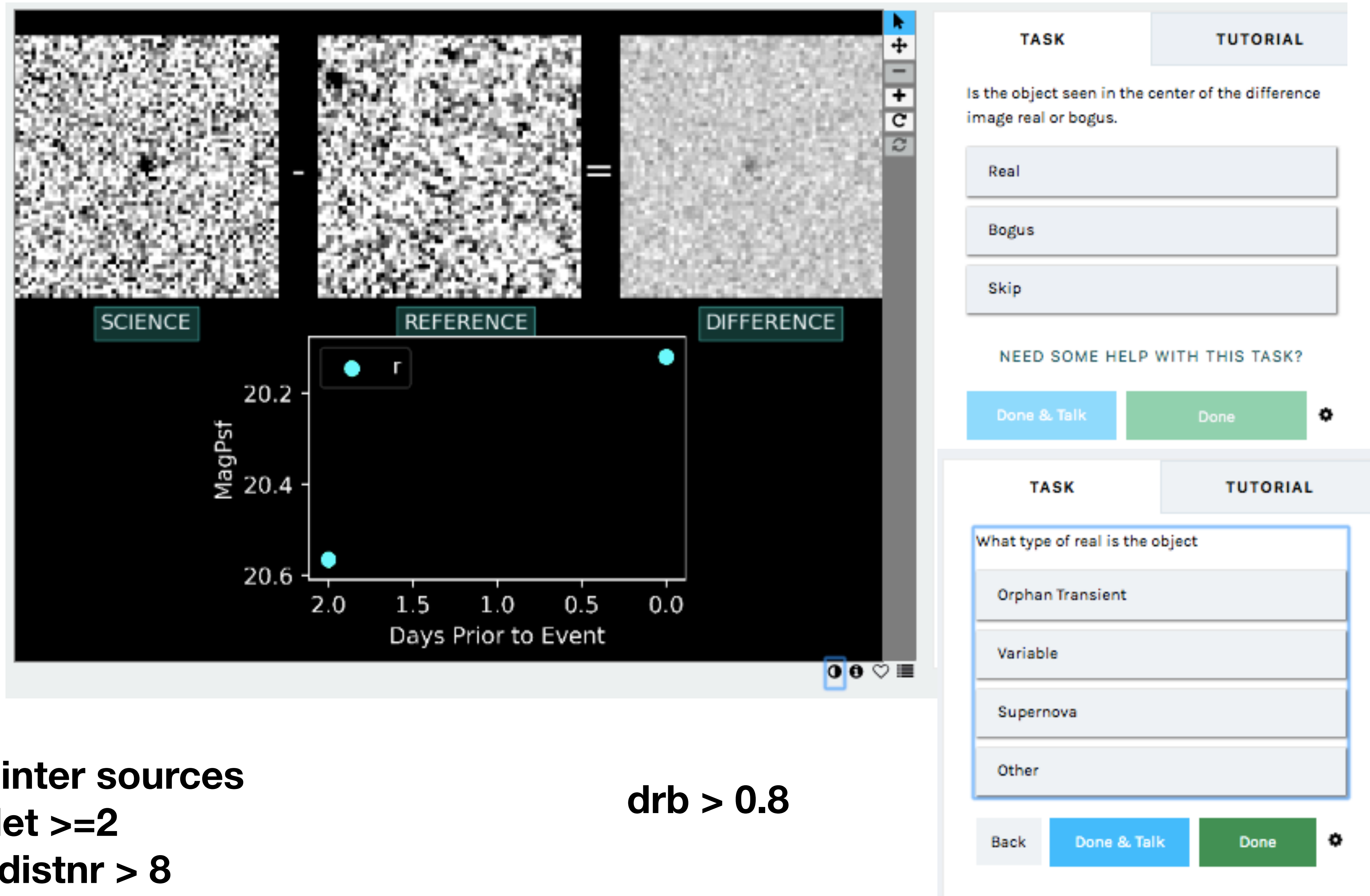
Timescale-Luminosity plot (Dan, Anna, ...)



advanced dynamic plotting -> filtering/discovering ...

[Hackday possibility]

Planned Zooniverse Campaign

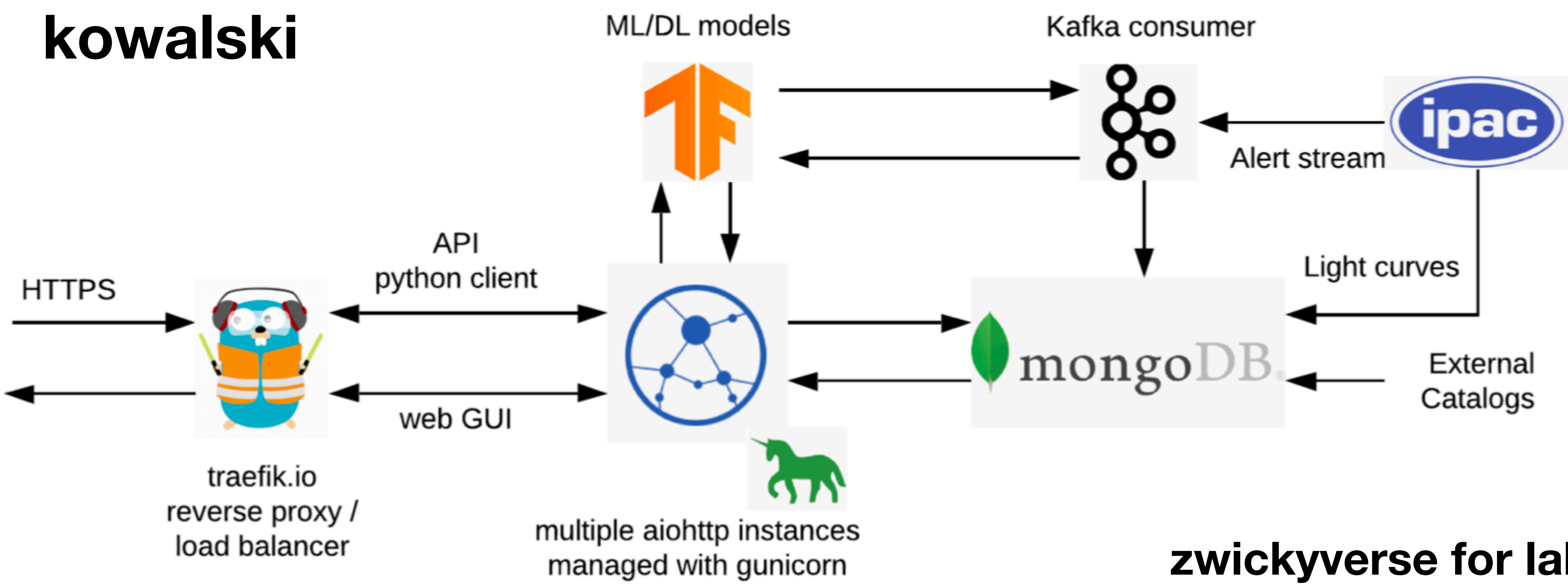


- **Fainter sources**
- **ndet >=2**
- **ssdistnr > 8**
- **Exactly 1 source within 30 arcsec**

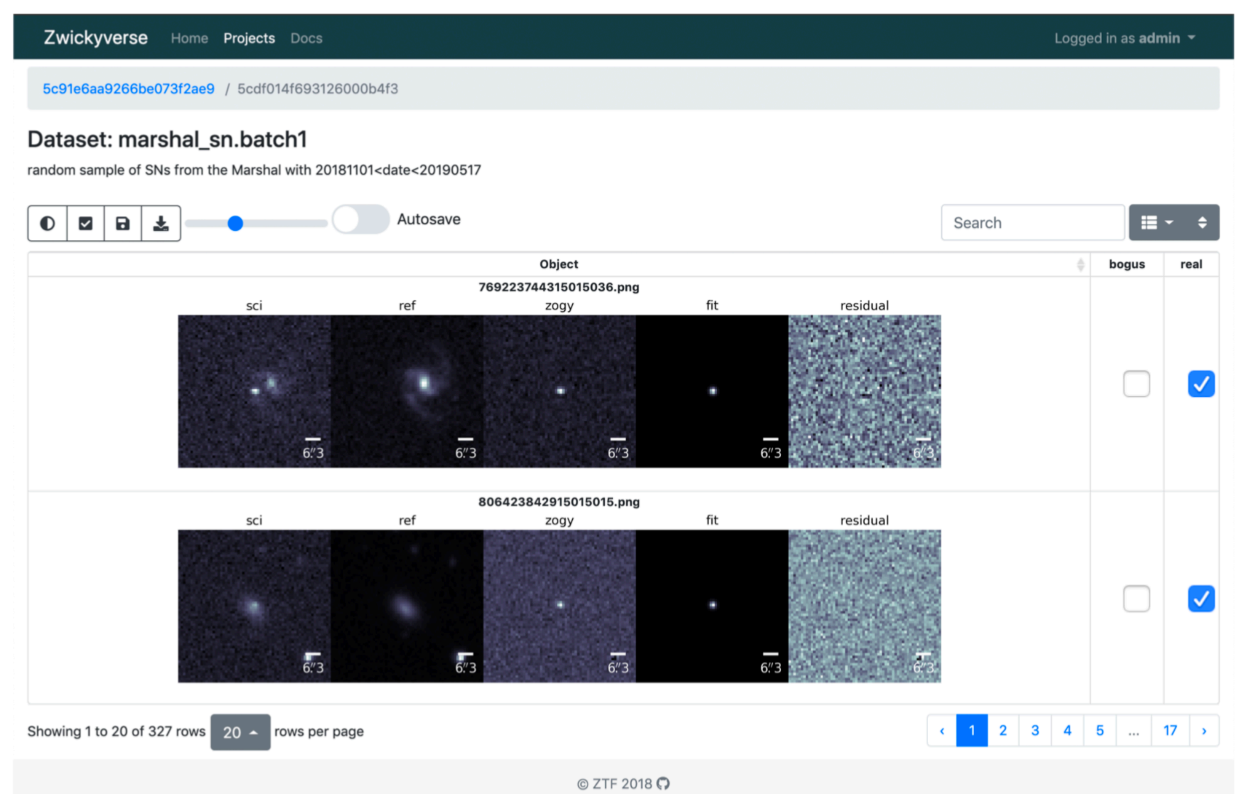
drb > 0.8

ZTF Brokering architecture

kowalski



zwickyverse for labeling



Data:
- 125M alerts, 2.5B light curves

Brokers:
- Alerce, ANTARES, Lasair, MARS, …
- GROWTH, AMPEL, kowalski

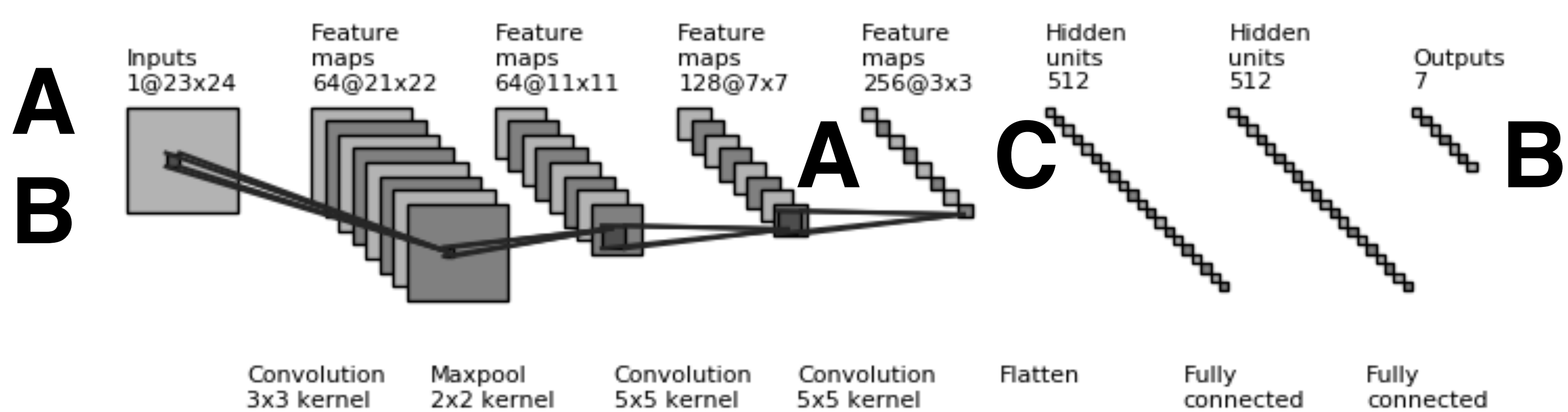
Followup:
- SEDM, BTS
- TNS

Visualization for interpretability

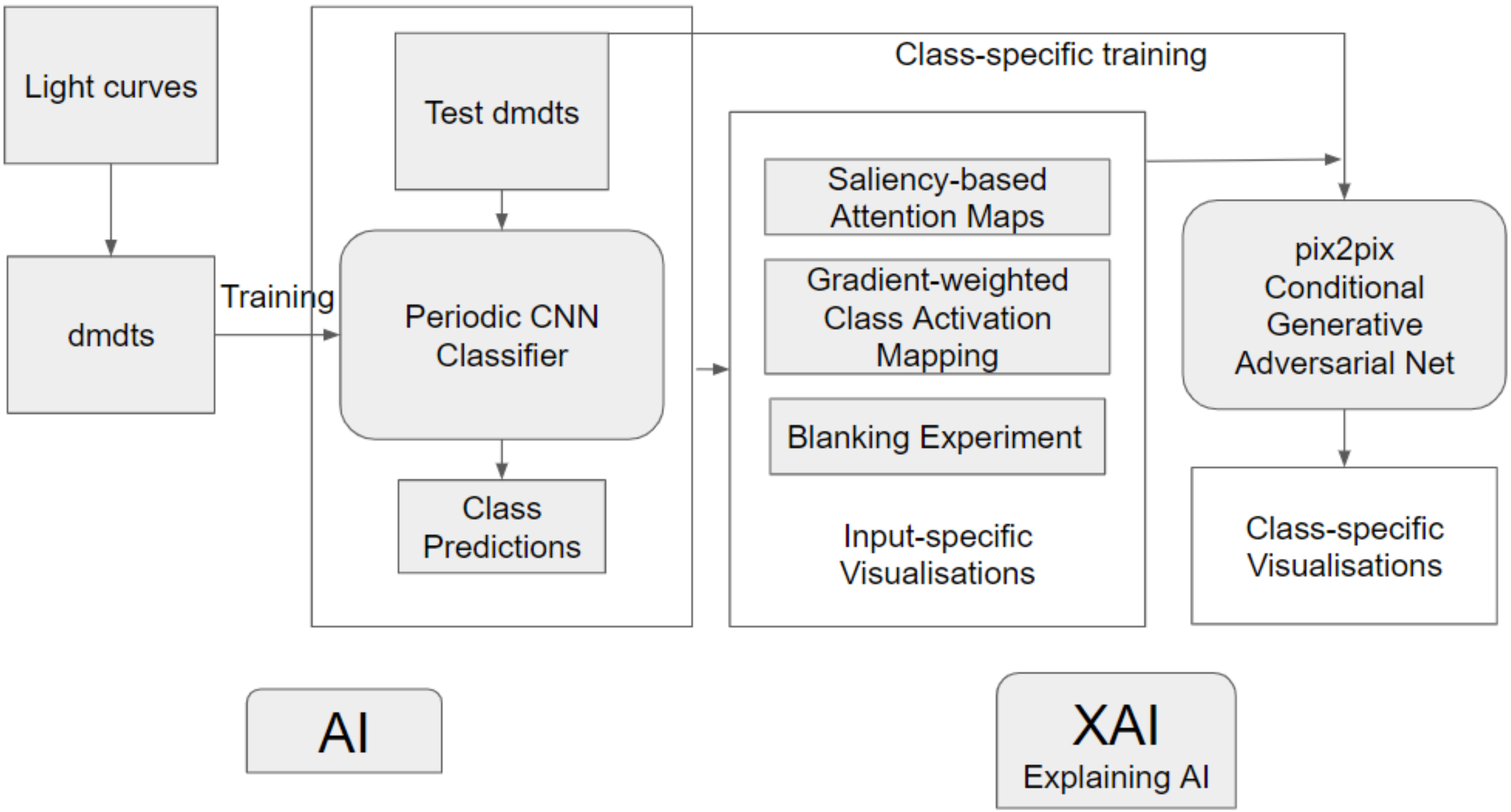
A. Activation Maximization
   - Initial layer filters easy to visualize
   - Generate input image that activates later filters

B. Saliency Maps
   - Gradient of o/p category wrt input image
   - Understanding attention of the classifier

C. Class Activation Maps
   - Gradients based on first dense layer
   - Spatial information still intact



https://raghakot.github.io/keras-vis/


Light curves
dmdts
Training
Test dmdts
Periodic CNN Classifier
Class Predictions
Class-specific training
Saliency-based Attention Maps
Gradient-weighted Class Activation Mapping
Blanking Experiment
Input-specific Visualisations
pix2pix Conditional Generative Adversarial Net
Class-specific Visualisations
AI
XAI
Explaining AI

Meet Gandhi