Introduction

Objective

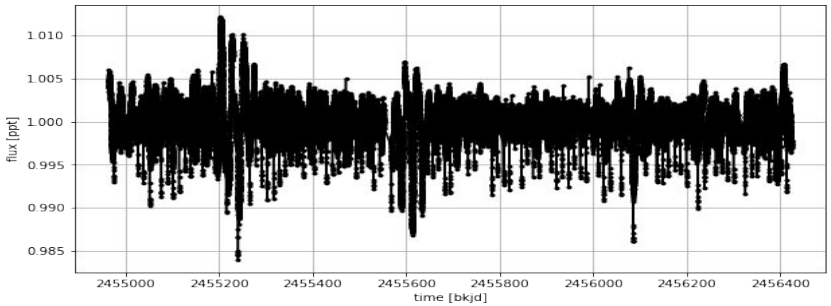
Our goal is to apply deep learning algorithms and GPU parallelization methods for developing an efficient and robust method of identifying HSP candidates in Kepler data.

Kepler Light Curves & Current Methods

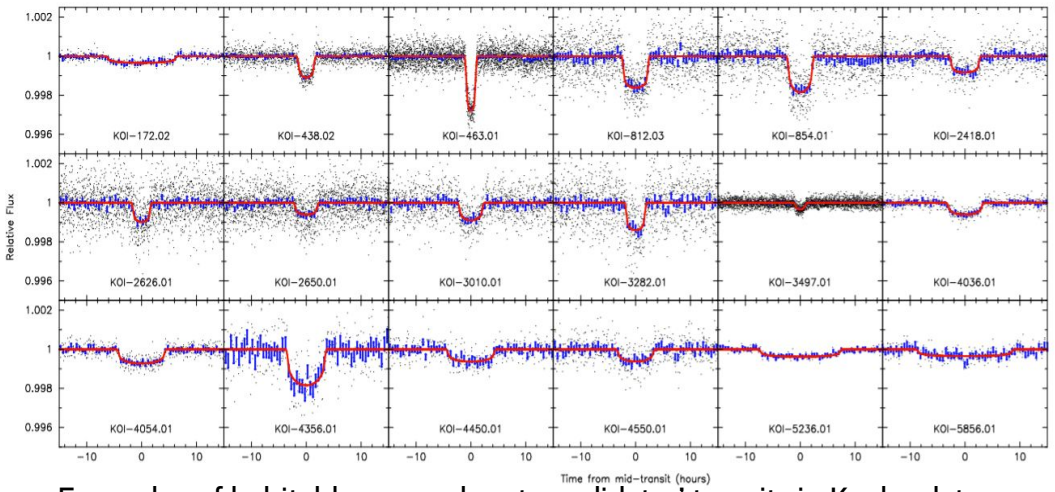
- Kepler mission collected photometric observations of stars as light cueves
- Current methods of searching for exoplanets involve identifying threshold crossing events (TCEs), periodic “dips” in the light curve from a transiting planet
- TCEs are manually examined to eliminate false positives caused by noise, eclipsing binaries, or stellar activity
- Current methodology:
  - Statistical methods like Box Least Squares (BLS) imprecise for small and shallow transits, inefficient for large numbers of candidates
  - Expensive hardware such as supercomputers required for analysis

Habitable Small Planets (HSPs)

- Exoplanets with radii < 2.0 Earth radii, orbiting within a star’s habitable zone (HZ)
- Characterized by longer orbital periods (100 to 400 days), shallow and wide transits -- this makes their transits more difficult to identify with traditional methods
- HSPs and HZs provide a great opportunity to study the occurrence, development, and evolution of Earthlike worlds
- Fewer than 60 HSPs currently known, out of over 4,281 confirmed exoplanets -- finding more would allow for stronger population studies



An example of raw lightcurve data from the Kepler Input Catalog (KIC).



Examples of habitable-zone planet candidates’ transits in Kepler data. From *Torres et. al 2011*

Methods 1.1: Preprocessing Kepler Light Curve Data

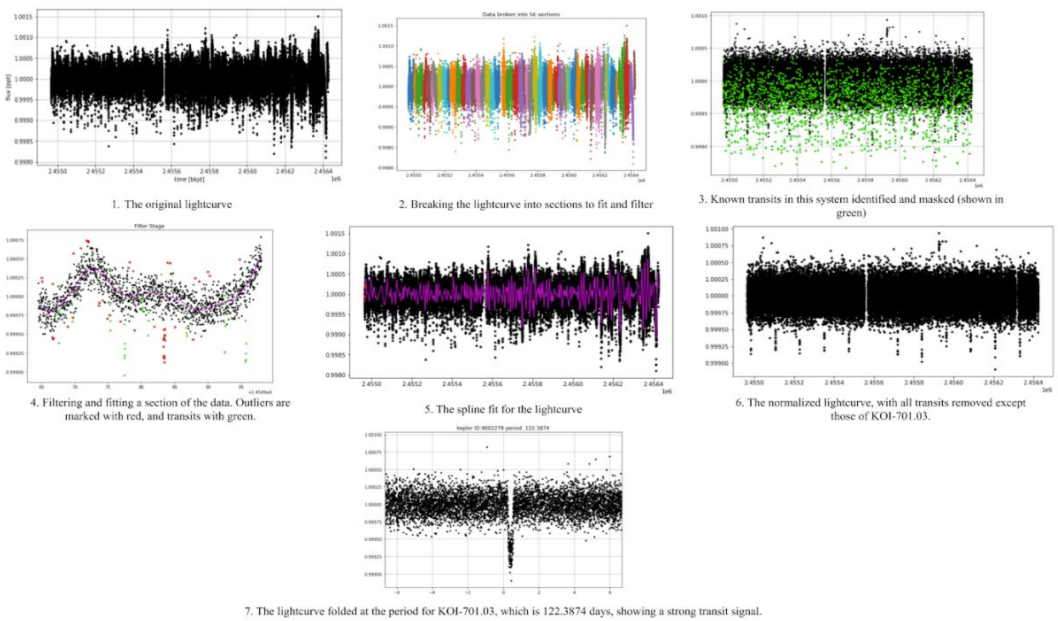
Normalization and Fitting

1. Light curves were broken into sections at locations of gaps
2. Known transits were masked and removed to prevent interference with fitting
3. Spline Fitting was automatically performed with Bayesian Information Criterion process
4. Fits were manually verified and readjusted if necessary
5. Discrepant data manually removed

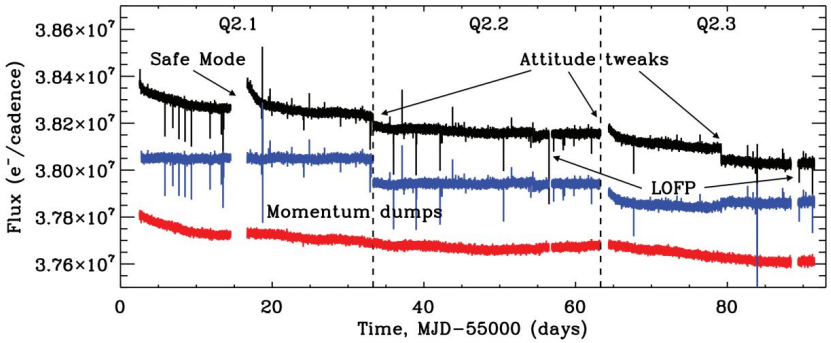
Challenges Encountered

- Presence of “Altitude Tweaks” (sudden vertical translation of flux trend) interfered with fit quality
- “Loss of Fine Pointing” (LOFP) events resulted in significant outliers
- “Safe Mode” events in Kepler telescope disrupted flux trend
- Systematic underfitting fails to account for short-period rotationally variate stars
- Light curves with periodic noise and interference were difficult to normalize

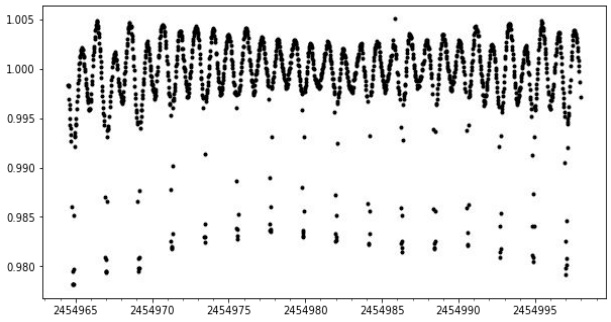
Preprocessing Steps

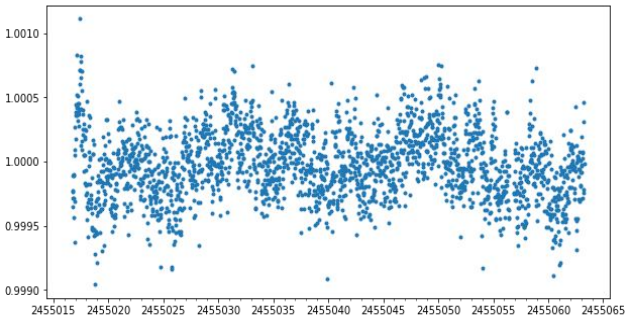


1. The original lightcurve
2. Breaking the lightcurve into sections to fit and filter
3. Known transits in this system identified and masked (shown in green)
4. Filtering and fitting a section of the data. Outliers are marked with red, and transits with green.
5. The spline fit for the lightcurve
6. The normalized lightcurve, with all transits removed except those of KOI-701.03.
7. The lightcurve folded at the period for KOI-701.03, which is 122.3874 days, showing a strong transit signal.

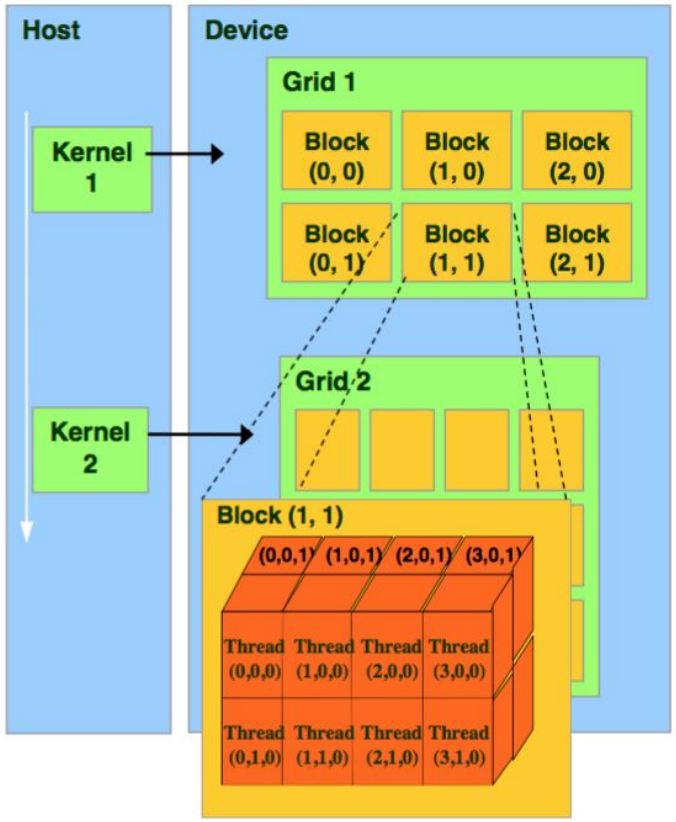


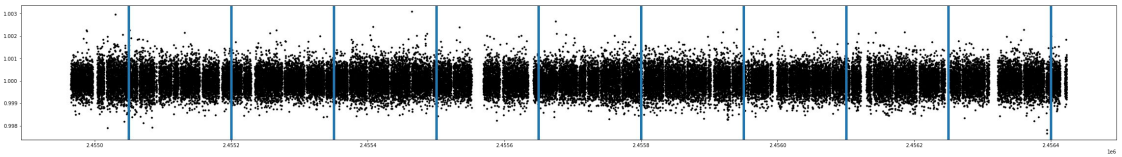
Difficult-to-Process Examples

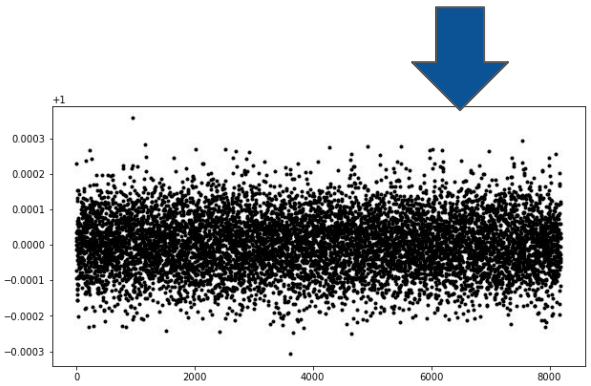




Methods 2.1: GPU Constant Width Folding & Binning







x 100000

What is Constant Width Folding, and Why?

- "Fast Folding":
  - Light curves were folded on 100,000 periods between 100-200 days
  - Every n points in a fixed point-size window averaged ("binned") to 8,192 pixels (1 pixel ~ 30 min)
- GPU Parallelization:
  - While CPU operations serialized, GPU performs all binning simultaneously by assigning each thread independent task
  - Implemented CUDA programming language for fastfolding

- **GPU Runtime Performance:** ~70 seconds for 100,000 period folds
- CPU Runtime Performance: ~60 minutes for 100,000 period folds

Considerations

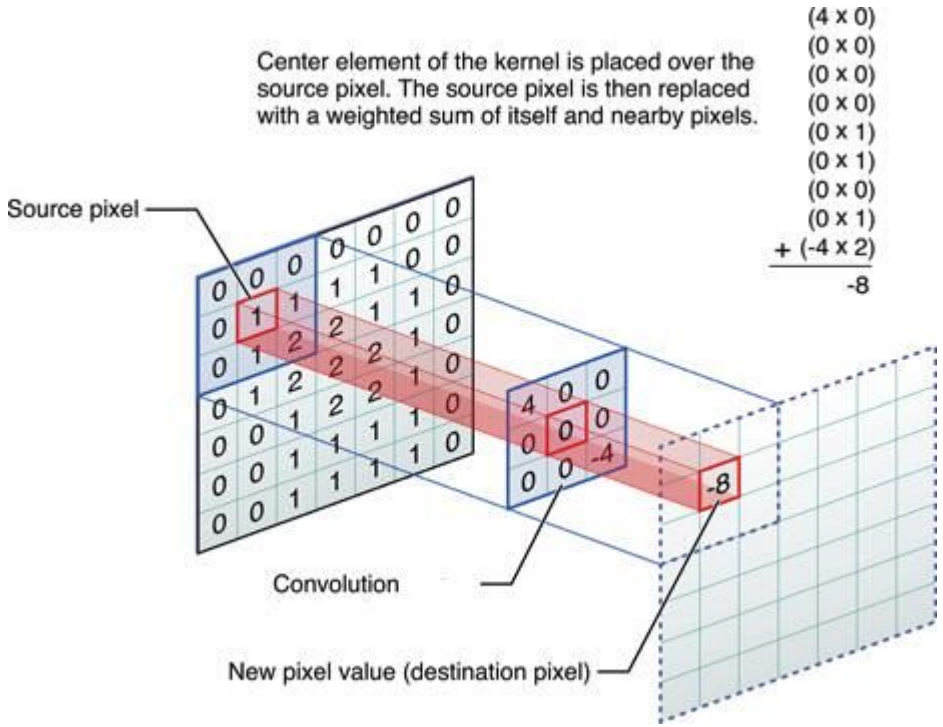
- Constant width folding eliminates the need for calculating a specific trial period, which often fails to detect shallow and long transits
- 8,192 is the minimum size where folding with any period in the 100-200 day range would not wash out short transits (since each point represents ~30 min)
- Windows taken based on fixed points rather than time intervals because this results in uneven sections, false positive detections

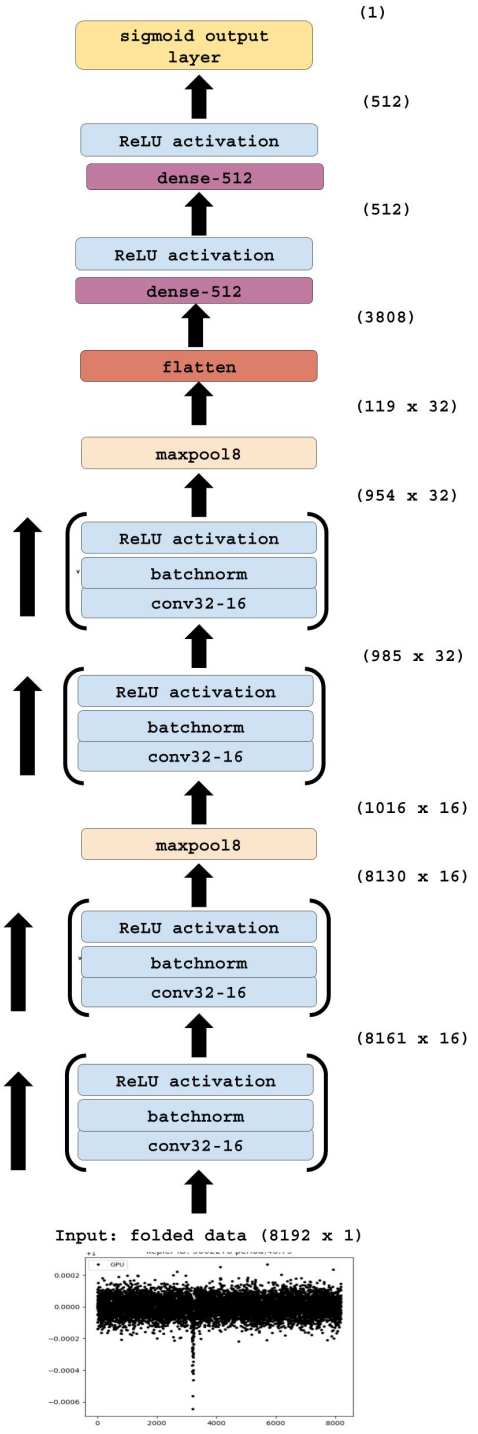
Methods 3.1: Convolutional Neural Network (CNN) Design

Why Convolutional Neural Networks (CNNs)?

- CNNs are frequently applied to extract features present in an image, and when individual pixels in the data are interrelated.
- Convolutional Kernels in CNNs are trained and optimized to detect features (i.e. prolonged dips in a flux trend in light curves)
- CNNs are optimal in this case, as other NN designs have layers that are fully connected, which is redundant in this case.
    - Spatial matching by CNNs reduce the number of trainable parameters, improves overall performance and training speed.

Principle of Convolutional Kernels



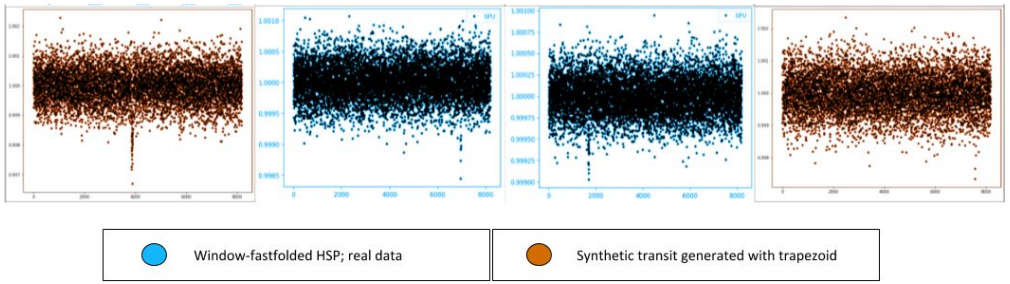


CNN Design

Methods 3.2: Initial Neural Network Design and Training

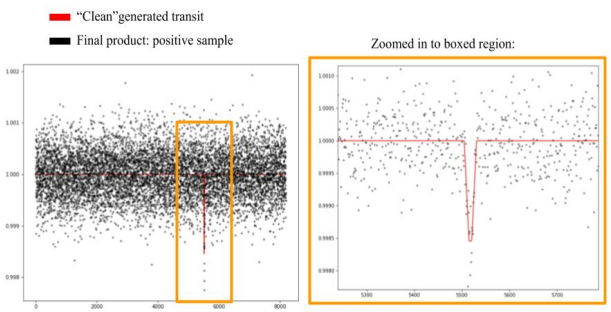
Generating Training Data

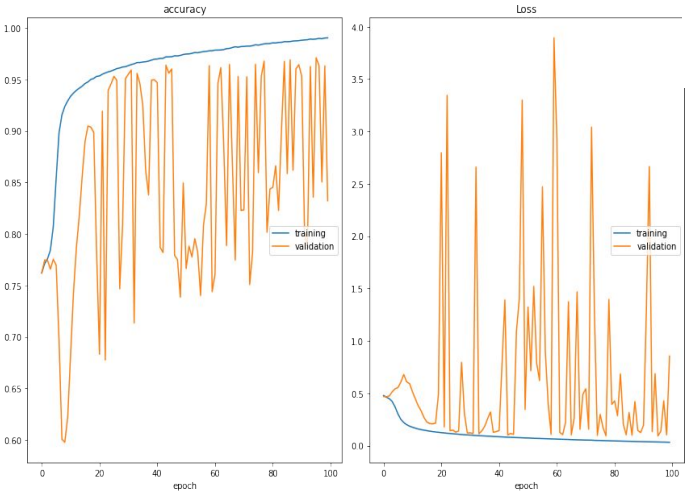
- Small population of known HSPs means not enough real data to train a CNN with
- Solution: create representative artificial dataset by generating trapezoidal "base transits" using distribution of known transits' properties



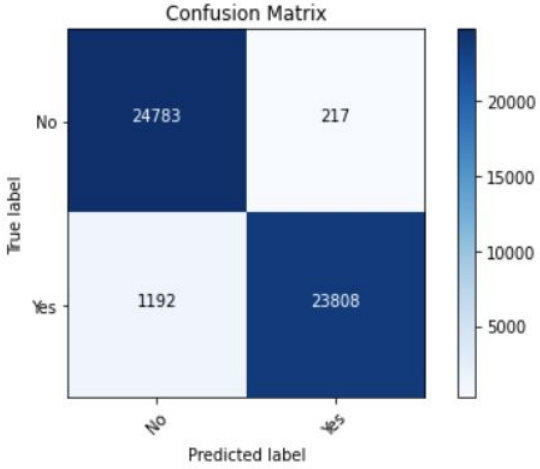
Training & Testing Results

- Trained for 100 epochs, batch size 32
- Adam optimization algorithm, learning rate of 0.000001
- Trained on 100,000 synthetic samples
- High training (99.04%) and validation (>97%) accuracy, but heavy fluctuation (over 10 percent at times)





Training phase--accuracy increases overall, but fluctuates often, most likely from too-high learning rate



Confusion matrix from testing 50,000 artificially generated samples with the CNN.
Model correctly identified samples as either having a transit signal or having no signal 97.18% of the time, at a high precision of 99.10%. However, it was more likely to fail on shorter-duration signals.
Solution: generate a larger number of weaker and shorter-duration signals for the training set, further adjusting the distribution of synthetic duration and depth to better reflect real distributions.

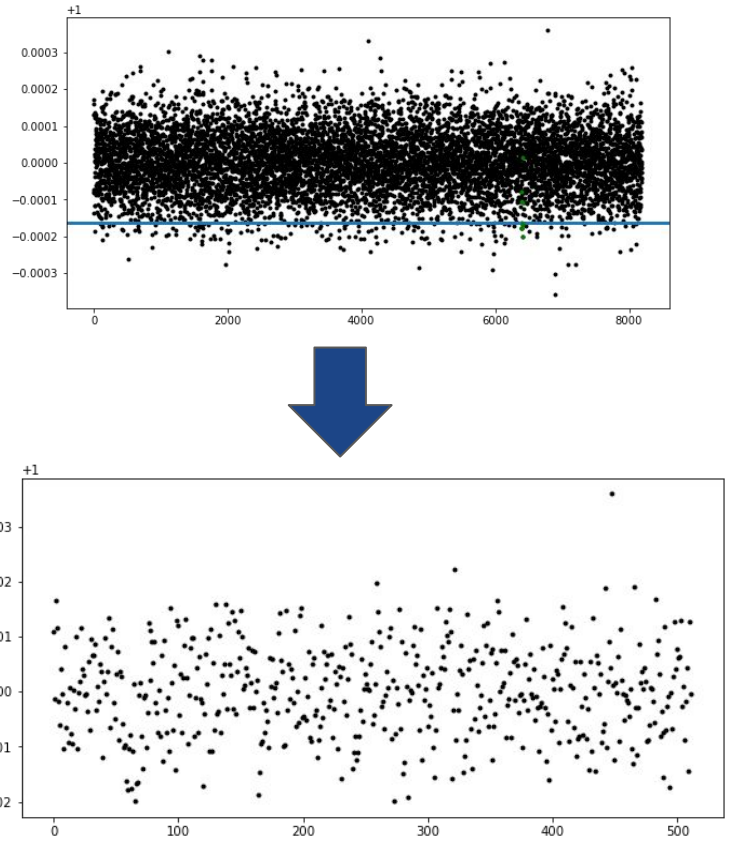
Takeaways and Changes

- Learning rate was too high-algorithm prematurely "jumps out" of some local minima, missing solutions
- Fluctuations in validation could also be from oscillations when convolutional layers are stuck on changing a certain parameter in failed attempts to optimize loss function
- Could try adding more convolution layers, increasing each convolution layer's size, and decreasing learning rate

Methods 3.3: Sigma Clipping

What and Why

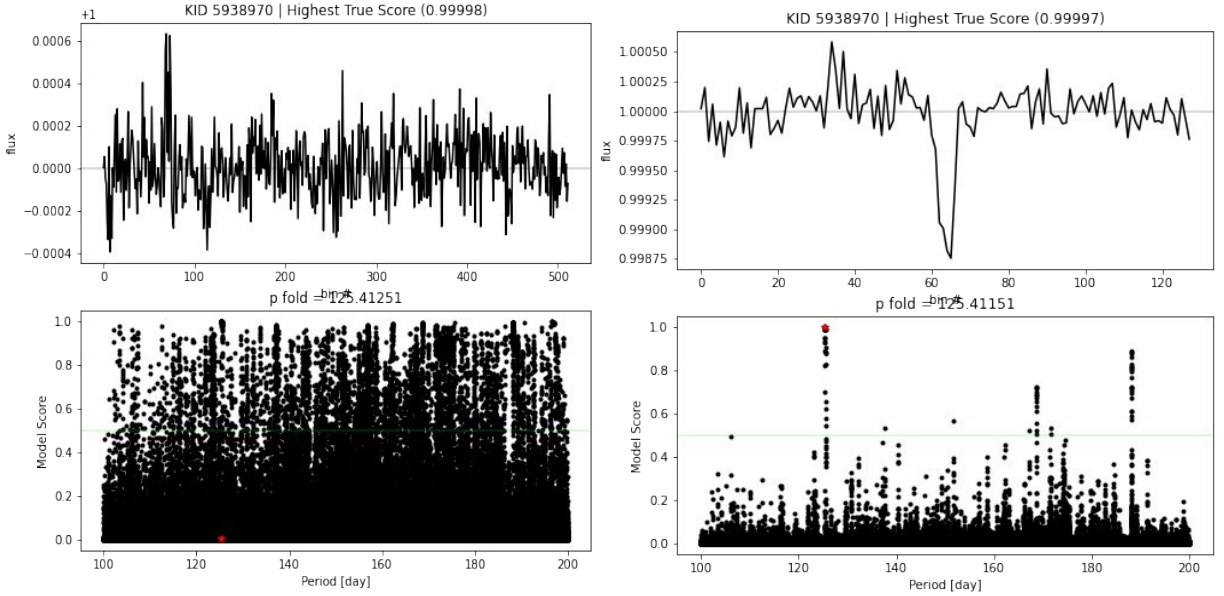
- Random outliers often interfere with testing, cause false detections.
- Sigma clipping: windows with three or more consecutive points less than 2σ were saved
- Rather than testing all 8192 points, only need to test 128 points



Analysis and Comparison

- 128 point window chosen based on analysis of HSP transit duration distribution
- Sigma clipping implemented with CUDA GPU parallelization for speed
- Significant reduction in false positive detections, improves analysis and period estimation
- Sigma clipping able to identify all candidates in a sample of 30 existing HSPs

- **Performance:** ~40 seconds for 100,000 folds



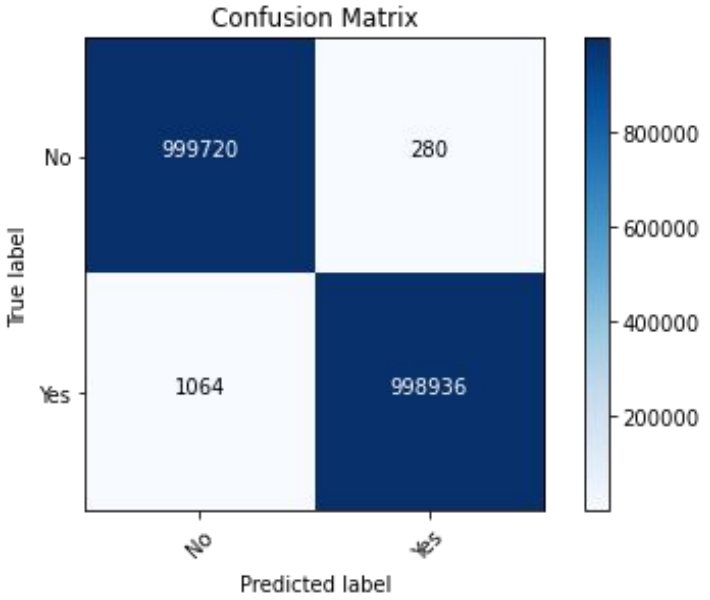
Methods 3.4: Modifying the Neural Network

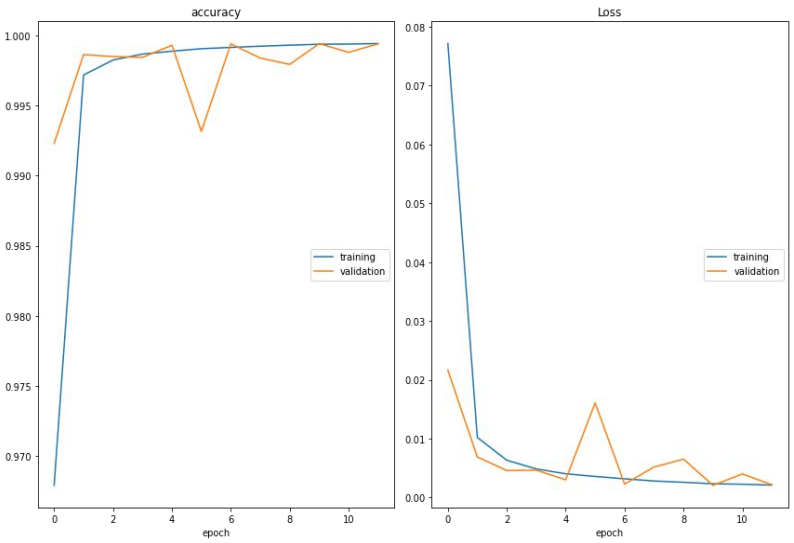
Changes Made

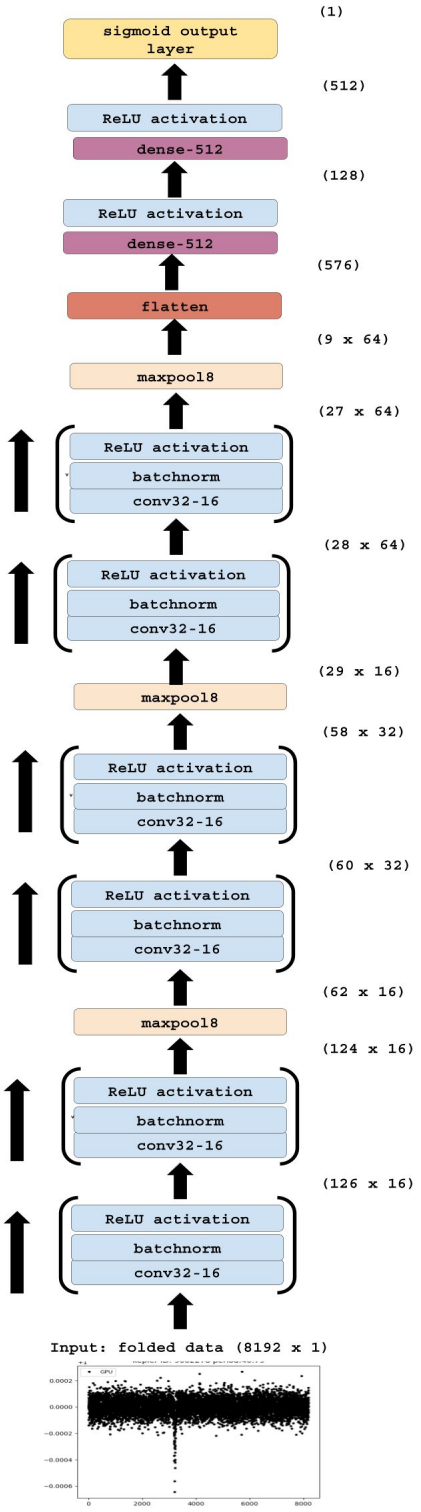
- Rather than receiving input of 8192, decrease input to 128 points to accommodate sigma clipping
- Increased training size from 200,000 samples to 2,000,000 samples
- Preserved methods of training set generation

Analysis

- Significant improvement in validation loss, accuracy
  - Validation Loss: 0.9997
  - Validation Accuracy: 0.9993
  - Recall Score: 0.9998
  - F1 Score: 0.9993
  - RO Score: 0.9993
- Enhanced training speed and low epochs required



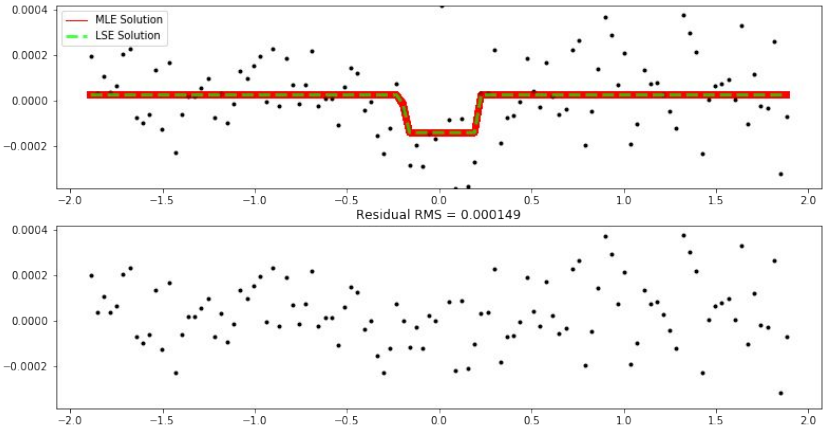




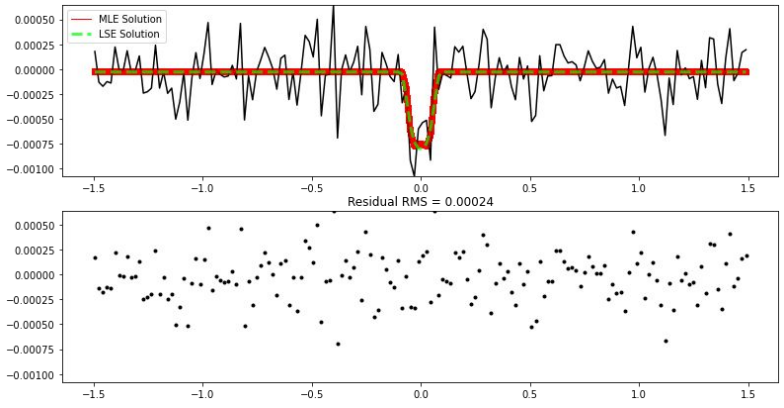
Results: Proposed HSP Candidates

Candidates and Properties

Candidate	Period (days)	t0 (bkjd)	Planet Radius (Earth radii)	Inclination (radians)
Candidate 1	192.470	2455142.497	1.247	1.569
Candidate 2	162.601	2455089.91	1.634	1.565

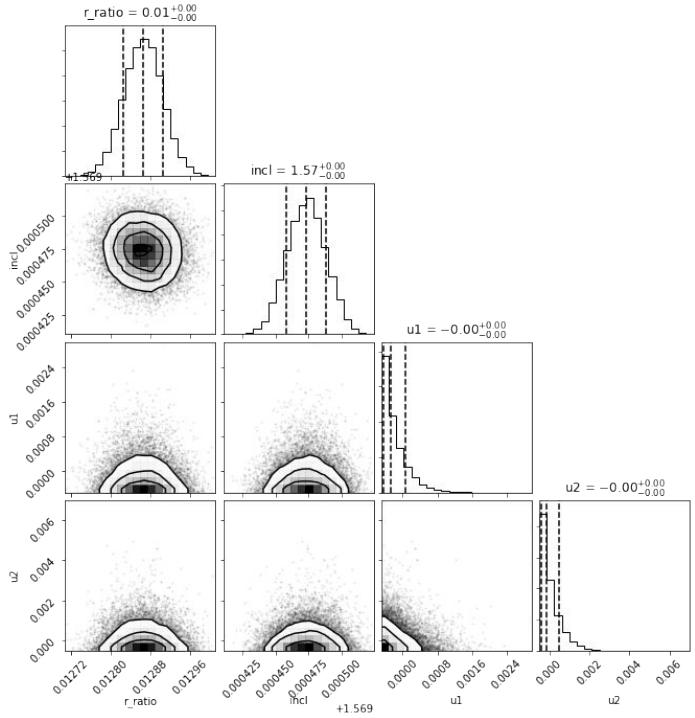


Candidate 1

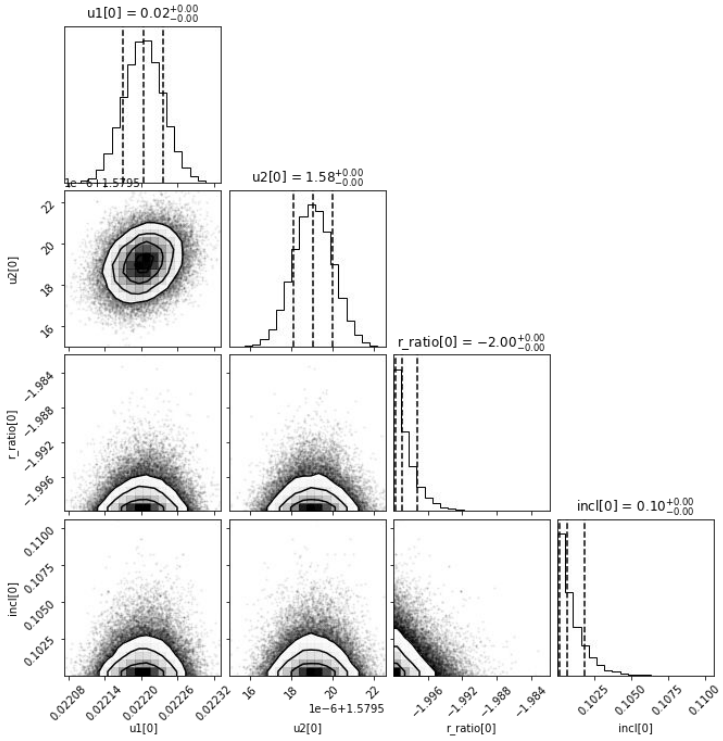


Candidate 2

Analysis



Candidate 1



Candidate 2

"Corner plots" showing the correlation of fixed variables: inclination (incl, in radians); and limb darkening parameters *u1* and *u2*

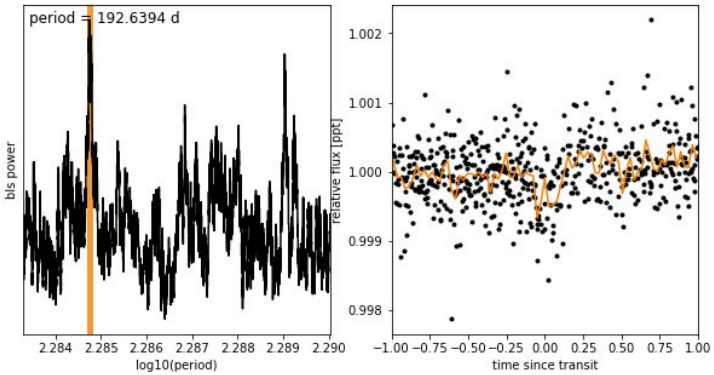
Validating Candidates

- Candidates were verified to not be "harmonic multiples" of:
  - Confirmed exoplanet candidates
  - Rotationally variable stars
  - Eclipsing binaries
  - Stellar Rotation Period
- Light curve of candidates renormalized
- Tested with statistical BLS (Box-Least Squares) method to confirm derived period, t0 parameters
- Each transit of candidates verified to:
  - Be consistent throughout
  - Not be coincidental with noise

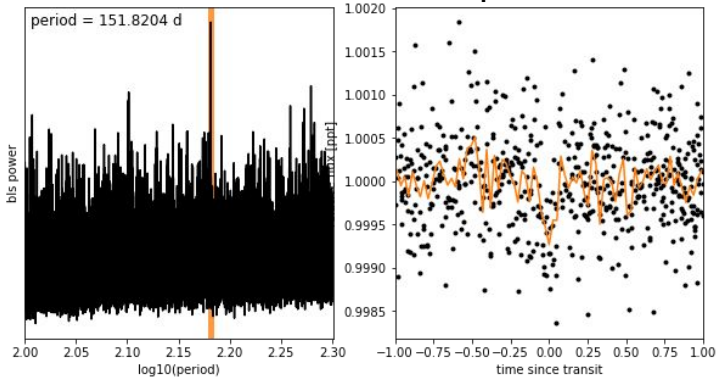
Conclusions

Summary

- Though there is room for improvement, the use of convolutional neural networks for detecting HSP candidates certainly has potential to be more efficient and effective than our current implementation
- The method demonstrated viability for identifying transit candidates by successfully recovering the transits of all known HSPs with 100-200 day periods



Candidate 1 - Boxed Least Squares Results



Candidate 2 - Boxed Least Squares Results

Shortcomings, Limitations, and Uncertainties

- Our CNN model is currently only optimized for finding planets with periods in the 100-200 day range; modifications and retraining may be necessary to expand the scope of the search (since known HSPs have been found with periods from ~8 to >300 days)
- Periodic noise (ex. from rotationally variable stars, or activity) sometimes makes quality normalization impossible
- Without a better way to remove noise, there is a limit to how much we can automate the normalization process, which comes at the cost of processing quality and speed

Future Steps

- There are many ways to verify our potential HSP candidates, and we will use them to continue examining the signals we found:
  - Comparing candidates with each target's known planets for the distribution and separation of their periods
  - Attempting to recover the transit with BLS
- Using a Fourier transform filter to model and remove high frequency noise may help-this is a possible solution to dealing with periodic noise that we plan on exploring
- Expanding beyond the Kepler catalog-apply this CNN and method to data from TESS as well

References and Acknowledgements

Papers and Journals

- Torres, G., Kane, S. R., Rowe, J. F., Batalha, et al. 2017, "Validation of Small Kepler Transiting Planet Candidates in or near the Habitable Zone". *The Astronomical Journal*, 154(6), 264. https://doi.org/10.3847/1538-3881/aa984b
- Torres, G., Kipping, D. M., Fressin, F., et al. 2015, "Validation of 12 Small Kepler Transiting Planets in the Habitable Zone". The Astrophysical Journal, 800(2), 99. https://doi.org/10.1088/0004-637x/800/2/99
- Kipping, D. M., 2013. "Efficient, Uninformative Sampling of Limb Darkening Coefficients for Two-Parameter Laws". *Monthly Notices of the Royal Astronomical Society*, 435(3), 2152–2160. https://doi.org/10.1093/mnras/stt1435
- Kopparapu, R. K., Ramirez, R., Kasting, et al. 2013, "Habitable Zones Around Main-Sequence Stars: New Estimates". *The Astrophysical Journal*, 765(2), 131. https://doi.org/10.1088/0004-637x/765/2/131
- Thompson, S. E., Coughlin, J. L., Hoffman, K., et al. 2018, "Planetary Candidates Observed by Kepler . VIII: A Fully Automated Catalog with Measured Completeness and Reliability Based on Data Release 25". *The Astrophysical Journal Supplement Series*, 235(2), 38. https://doi.org/10.3847/1538-4365/aab4f9
- Kovács, G., Zucker, S., & Mazeh, T, 2002. "A Box-Fitting Algorithm in the Search for Periodic Transits". Astronomy & Astrophysics, 391(1), 369–377. https://doi.org/10.1051/0004-6361:20020802
- Gajdoš, P., Vaňko, M., & Parimucha, Š, 2019. "Transit Timing Variations and Linear Ephemerides of Confirmed Kepler Transiting Exoplanets". Research in Astronomy and Astrophysics, 19(3), 041. https://doi.org/10.1088/1674-4527/19/3/41
- R. A. García, S. Hekker, D. Stello, et al. 2011, "Preparation of Kepler light curves for asteroseismic analyses", *Monthly Notices of the Royal Astronomical Society: Letters*, Volume 414, Issue 1, 11 June 2011, Pages L6–L10, https://doi.org/10.1111/j.1745-3933.2011.01042.x

Databases

- This research made use of the exoplanet package and its dependencies PyMC3 for the inference engine and modeling framework, AstroPy for units and constants, Kipping 2013 for the reparameterization of the limb darkening parameters for a quadratic law, and Luger et al. (2018) for the light curve calculation.
- This research made use of Lightkurve, a Python package for Kepler and TESS data analysis (Lightkurve Collaboration, 2018), as well as its dependencies AstroPy and AstroQuery.
- This research has made use of the NASA Exoplanet Archive, which is operated by the California Institute of Technology, under contract with the National Aeronautics and Space Administration under the Exoplanet Exploration Program.
- This research has made use of the SIMBAD database, operated at CDS, Strasbourg, France. 2000,A&AS,143,9, "The SIMBAD astronomical database", Wenger et al.