Enhancing Unknown Near-Earth Object Detection with Synthetic Tracking and Convolutional Neural Networks

Kevin Phan
EO Solutions
Justin Fletcher
Space Systems Command (A&AS)

ABSTRACT

This study conducts an examination of synthetic tracking as a fundamental technique for satellite detection. Using synthetic data generated by SatSim, we assess the efficacy of standalone synthetic tracking in detecting moving objects across varying brightness levels. Performance evaluation is conducted using metrics precision, recall, and F1 score, aiming to understand the performance of synthetic tracking. Particularly, we emphasize assessing the detection rates of dimmer objects. By implementing a convolutional neural network model through transfer learning, we notably enhance detection metrics and improve sensitivity in detecting dimmer objects compared to standalone synthetic tracking. We demonstrate a robust pipeline for maximizing detection of satellites with unknown velocities using sidereal data by combining synthetic tracking with convolutional neural networks while maintaining high precision.

1. INTRODUCTION

Astrometric localization involves extracting stars from images and aligning them with real-world stars, enabling precise satellite localization down to arcseconds. Successful astrometric fitting is crucial for subsequent near-Earth object (NEO) tracking and for space domain awareness (SDA). When an object's velocity and location are known, a telescope can follow the movement of the object and accurately record its position. This is known as rate tracking. However, the challenge lies in detecting unknown objects due to uncertainties in velocities and astrometric data. Unknown objects result from untracked maneuvers, new launches, and breakup events – establishing orbits for these objects is a fundamental goal of SDA

To address these challenges, the study proposes a novel approach combining deep learning with synthetic tracking techniques. Synthetic tracking not only reduces noise in short-exposure sidereal images but also allows for the correction of camera imperfections, refining the astrometric fit for deep learning. This refinement enhances the accuracy of satellite localization and subsequent tracking. Most importantly, synthetic tracking allows for the detection of objects of unknown velocities [8].

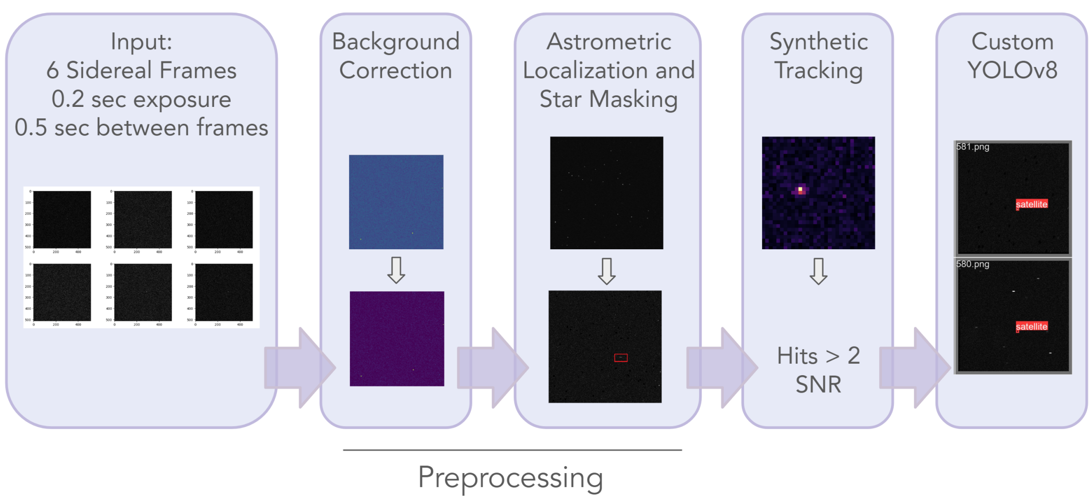

Fig. 1: Integrated Synthetic Tracking + CNN Pipeline

The potential for false positives and star confusion drives classic synthetic tracking algorithms to adopt high signal to noise requirements. In this work, we demonstrate that convolutional neural networks can significantly relax the SNR constraint while maintaining performance on dim signals. The research demonstrates the efficacy of the proposed methodology in localizing and detecting unknown objects with high precision. By leveraging CNNs alongside synthetic tracking, the study advances the state-of-the-art in satellite detection and identification, offering a promising framework for applications requiring accurate and robust object localization in satellite imagery.

Studying NEOs, such as satellites and asteroids, is crucial for planetary protection and space domain awareness (SDA). This study hopes to improve the detection rate of especially dim satellites and NEOs moving at unknown velocities. More specifically, we utilize sidereal data to confirm and localize objects with precise astrometric measurements.

2. RELATED WORK

2.1 Astrometric Localization

Astrometric Localization–as stated prior–is the ability to take an image of stars and match it to a known set of real-world stars. This process allows for us to track satellites based off of the set of stars which is critical in space domain awareness (SDA) and plays a vital part in future protection against threats from space.

2.2 SatSim

SatSim [2] is an open-source software that can simulate moving objects in the sky projected on real world starfields. This software is critical for our approach as it allows for generating synthetic data which is still usable on a massive scale whilst providing annotations. SatSim played a significant role in allowing us to experiment and train our architecture whilst retaining integrity with real-world cases. We selected the parameters and configurations to mimic ground based sensors and telescopes to best ensure our results could be used in the field. In this study, we will be using 6 sequential frames taken 0.5 seconds apart with an exposure time of 0.2 seconds each. This results in a cumulative exposure time of 1.2 seconds.

Parameter	Value
Spatial Oversampling Factor	15
Temporal Oversampling Factor	100
FOV (x, y)	0.308, -0.308
Dark Current	Random Log-normal
Gain	1
Bias	0
Zero Point	20.666
A2D Response	Linear
A2D Bias	1500
Read Noise	9
Electronic Noise	0
PSF Mode	Gaussian
Exposure Time	0.2 sec
Exposure Gap	0.5 sec
Number of Frames	6
Objects per Image	1
Object Apparent Magnitude	Variable
Star Mode	SSTRC7

Table 1: SatSim Configurations

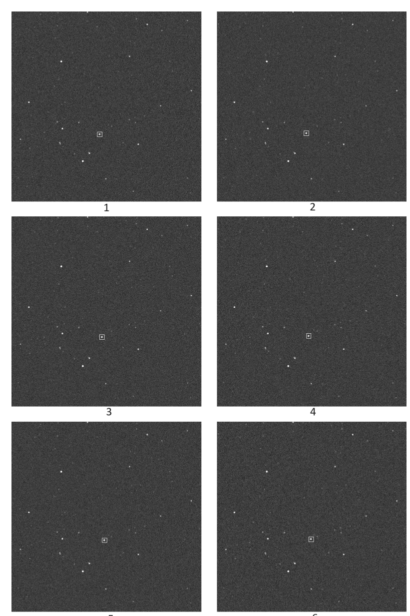

Fig. 2: 6 sidereal frames generated by SatSim

2.3 Synthetic Tracking

Synthetic tracking involves taking several short exposure frames and stacking the images based on potential velocities [8]. After stacking the images, based on the signal-to-noise ratio (SNR), we can then see which velocity shifts were successful, thus identifying the object's speed and location. There are more complicated methods of calculating SNR, but for our study we define SNR as:

$$SNR = \frac{\text{Brightest Pixel}}{\text{Median Pixel}}$$

where we define the brightest pixel as the signal, and the median pixel as the noise.

2.4 Convolutional Neural Networks

CNNs are a commonly used architecture used to help find overarching patterns in images. CNNs provide not only the backbone for YOLOv8, but are also used in our custom task-specific head mentioned later on. Past literature indicates that CNNs succeed at identifying NEOs in rate tracked images, but have yet to be proven in synthetically tracked images [3].

3. FORMALIZATIONS

3.1 Detection

We will be measuring detection by comparing the reported object coordinate by the detection algorithm with the annotation provided by SatSim. For this, we will be allowing 3 pixel allowed error for correct classifications to prevent over-saturated signals from being reported incorrectly. The definition of the pixel coordinate of detection is defined separately for the bounding box learned approach and the synthetic tracking algorithm.

In this study, $(x_{\text{True}}, y_{\text{True}})$ is the ground truth (x, y) pixel coordinate of an object provided by SatSim annotations.

For all reported objects, a detection is marked as a true positive (TP) if $\sqrt{(x_{\text{True}} - x_{\text{Reported}})^2 - (y_{\text{True}} - y_{\text{Reported}})^2} \leq 3$, and false positive (FP) otherwise. Any remaining objects that were not reported are marked as false negatives (FN). True negatives (TN) are not applicable to object detection tasks.

3.1.1 Synthetic Tracking Detection

For the synthetic tracking portion of the pipeline, the algorithm reports the highest SNR pixel as the detected object's location on the image. Intuitively, by applying a starmask, the brightest object left on the image should be an object of interest. We use objects with SNR greater than 7.5 as a detection and report the coordinate for comparison.

3.1.2 YOLOv8 Detection

For our custom YOLOv8 model, the detection coordinate is defined as the center of the bounding box of a reported detection. YOLOv8 provides each detection with a bounding box. These bounding boxes have a center coordinate, height, and width. We use the center coordinate as the detection coordinate.

3.2 Performance Metrics

For this study, we will be measuring object detection performance using the following metrics: precision, recall, and F1 score.

Precision (also called Positive Predictive Value) is the ratio of true positives to the sum of true positives and false positives.

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}}$$

Recall (also called Sensitivity or True Positive Rate) is the ratio of true positives to the sum of true positives and false negatives.

$$Recall = \frac{N_{TP}}{N_{TP} + N_{FN}}$$

The F1 score is the harmonic mean of precision and recall.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

In this study, we place high emphasis on minimizing N_{FP} which correlates to enhanced precision. We hope to improve detection rates of dim objects without sacrificing low false positive rates. For this study, we set the threshold for an acceptable precision to be greater than 0.8.

4. METHODOLOGY

4.1 Preprocessing

The preprocessing stage is crucial for preparing the raw astronomical images for analysis. This stage includes background correction to enhance the visibility of celestial objects and astrometric localization coupled with star masking to accurately identify and mask stars. These steps are foundational for effective synthetic tracking and deep learning processes that follow.

4.1.1 Background Correction

The objective of background correction is to mitigate the effects of ambient light and other artifacts that may obscure the detection of NEOs. This process begins with a median estimator to calculate the background light level across the image. The median is chosen due to its robustness against outliers, such as anomalously bright pixels that could represent stars or cosmic rays, providing a stable baseline for background light levels.

After determining the median background, it is subtracted from the entire image using the photutils library [1], a Python package specifically designed for astronomical image processing. This subtraction effectively removes the uniform background noise, enhancing the contrast between celestial objects and the now flattened background. This step is critical for isolating the signals of NEOs and other celestial bodies from the diffuse light and noise inherent in raw astronomical images.

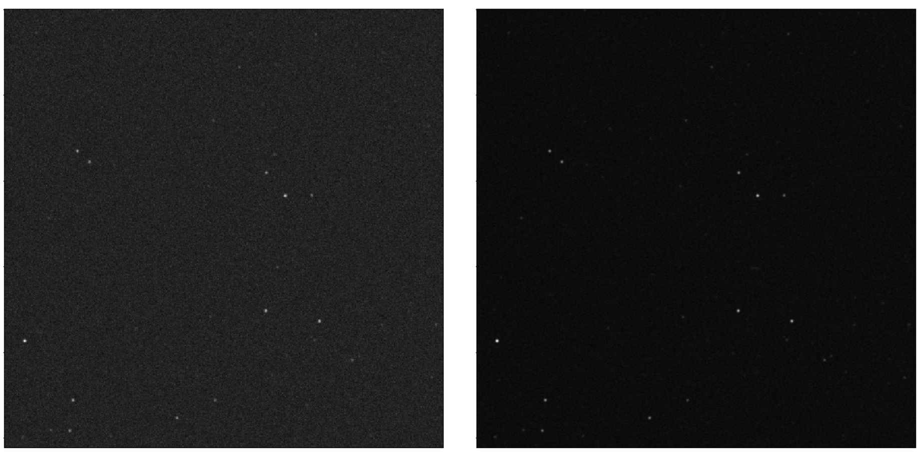

Fig. 3: Raw image (left), background corrected image (right).

4.1.2 Astrometric Localization and Star Masking

Astrometric localization and star masking use an integrated Gaussian point spread function (PSF) to model the stars, accounting for the blurring effects of the atmosphere and telescope optics [4]. Each star is identified based on its Gaussian profile, a method that enables the accurate determination of its position.

Once stars are detected, they are matched against the GAIA star catalogue using a triangular matching algorithm. This approach matches detected stars with those in the catalogue by identifying unique patterns formed by triangles of three stars, facilitating precise astrometric alignment with known celestial coordinates. With this alignment, we can localize objects, if detected, down to the arcsecond [6].

Misalignments between the image and the star catalogue are corrected using an inverse transformation matrix, implemented through skimage's transform module. This correction ensures that the positions of detected stars are accurately aligned with their known coordinates in the sky, essential for reliable subsequent analysis.

Furthermore, a star mask is applied to suppress the signals from stars in the image. This is achieved by creating masks over the positions where stars have been localized, effectively removing their light from the image. The purpose of this star mask is to prevent the bright signals of stars from interfering with the detection of NEOs, allowing for a clearer analysis of objects of interest without the distraction of stellar light.

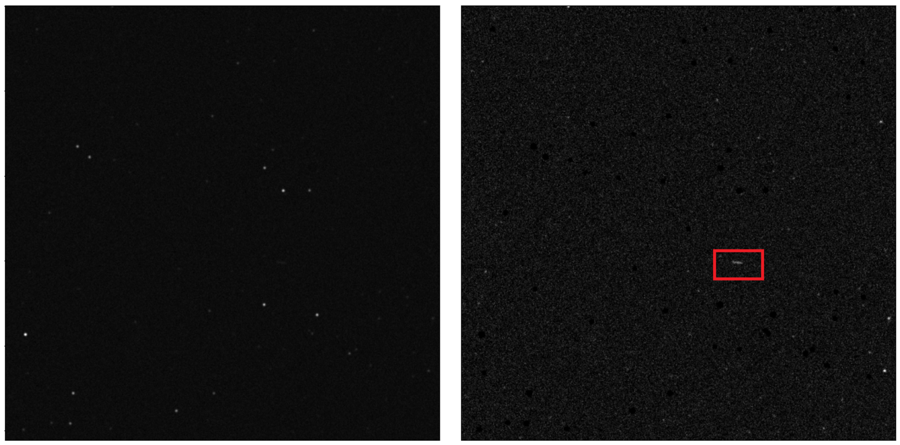

Fig. 4: Image before star suppression (left), image after star suppression (right). In the rightmost image, without stellar noise, a NEO streak across the aperture becomes visible (red box)

These preprocessing steps lay the groundwork for successfully applying synthetic tracking and deep learning techniques in the detection and analysis of NEOs, by ensuring the input images are optimally prepared for the subsequent complex analysis.

5. SYNTHETIC TRACKING

The challenge in detecting unknown NEOs is that their velocities are unknown. To address this, we use a synthetic tracking approach with a wide array of potential velocities. Specifically, we initialize a velocity grid designed to explore approximately 1000 distinct potential velocities, establishing a comprehensive search space.

We developed a stacking algorithm that uses the ndimage module from SciPy, which is useful for manipulating multidimensional arrays. This algorithm shifts each frame according to a set of predetermined velocities and stacks the frames on top of each other.

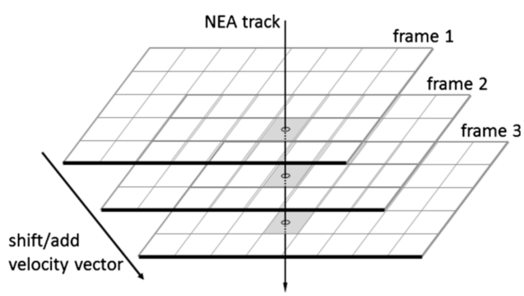

Fig. 5: Visual Representation of Synthetic Tracking. Figure from Shao et al. (2018) [8]

The core idea behind synthetic tracking is accumulating the signal from the moving object across multiple frames. When the applied velocity vector aligns with the actual motion of the object, the stacking process overlays the object's signal across the frames. As a result, the object's brightness and detectability is enhanced. To efficiently identify the optimal velocity vectors that maximize the object's signal, we generate a heatmap of the maximal signal for each velocity vector.

Afterward, we apply maximal filters from SciPy to the heatmap to isolate peaks, which represent the velocities that align the best with the object's motion. This allows us to identify the velocities that maximize signal enhancement, thus determining both the speed and the precise location of dim, previously undetectable objects. Combined with results from astrometric localization, we have all the information we need to document the NEO.

Because it is important to detect these objects in a timely manner, we employ CuPy, a library that allows common functions from the NumPy and SciPy libraries to access and utilize CUDA subroutines. On CUDA compatible GPUs, we significantly improve runtime where 6 frames can undergo background correction, astrometric localization, and 1000 experimental velocities with synthetic tracking analysis in around 5 seconds [7].

This synthetic tracking method not only broadens the scope of detectable NEOs by incorporating those with dim signals but also enhances the precision of location and speed estimation.

5.1 Synthetic Tracking Performance Metrics

Apparent Object Magnitude	Precision	Recall	F1 Score
6-7	1.000	1.000	1.000
7-8	1.000	1.000	1.000
8-9	1.000	1.000	1.000
9-10	1.000	1.000	1.000
10-11	1.000	0.826	0.905
11-12	1.000	0.569	0.733
12-13	1.000	0.154	0.266
13-14	-	0.000	0.000
14-15	-	0.000	0.000
15-16	-	0.000	0.000

Table 2: Synthetic Tracking performance with SNR threshold 7.5

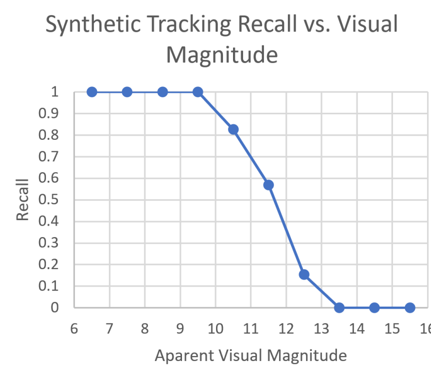

Fig. 6: Recall vs. Visual Magnitude

Note that larger apparent object magnitude correspond to dimmer objects. As depicted by our results in table 2, we observe that standalone synthetic tracking does not detect objects dimmer than SNR 7.5, as expected due to the threshold setting. For objects brighter than 7.5, synthetic tracking is extremely robust with little to no error and maintains high precision for brighter objects.

6. CUSTOM YOLOV8

Given the limitations of traditional synthetic tracking methods, particularly in detecting objects with low SNRs, we integrated a custom version of YOLOv8 (specifically, the YOLOv8 medium model), a state-of-the-art deep learning model known for its efficiency and accuracy in object detection tasks [5]. This integration aims to enhance the detection capabilities for dim objects that synthetic tracking alone may fail to identify confidently.

6.1 Model Customization and Training

To adapt YOLOv8 for the unique challenges posed by astronomical data, we augmented the model with a task-specific head designed to distinguish between NEOs and the vast array of celestial and atmospheric phenomena captured in astronomic observations.

Our training dataset consisted of 2500 images synthesized using SatSim. This dataset was instrumental in providing a controlled environment to train our model, ensuring a diverse range of examples including objects with SNRs lower than 7.5, the threshold below which traditional synthetic tracking methods struggle. More specifically, we used a uniform distribution of visual magnitudes from 10 to 20. The rest of of the configurations were identical to those mentioned in table 2.

We formatted the data to include annotations of the NEOs, specifying their bounding boxes and velocities. This preprocessing step was critical for training our model to detect the presence of an object a sequence of sidereal images.

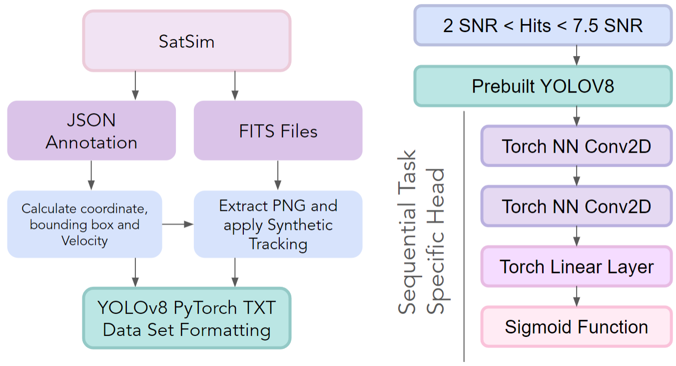

Fig. 7: Data retrieval pipeline (left), pipeline for low SNR objects through YOLOv8 and task-specific head (right)

Training parameters were chosen to optimize the model's performance:

- **Epochs**: 25, to balance between underfitting and overfitting.
- **Optimizer**: AdamW, selected for its effectiveness in handling sparse gradients and adaptive learning rates.
- **Learning Rate**: Initiated at 0.01, with adjustments based on performance metrics.
- **Weight Decay**: Set to 0.0005, to regularize and prevent overfitting.
- **Batch Size**: 16, to ensure a manageable memory footprint while maintaining sufficient granularity for gradient updates.
- **Dropout**: Not applied, to maximize the model's learning capacity from the limited dataset.

Fig. 8: box area loss (left), classification loss (middle), box anchor loss (right)

6.2 YOLOv8 Performance Metrics

Initial experiments with synthetic tracking alone, decreasing the SNR threshold resulted in high false positive rates and low precision. The limitations of synthetic tracking in isolating dim objects were evident, underscoring the need for a more robust detection method.

For visual magnitudes below 11, our custom model struggles with false positives, resulting in inadequate precision measurements. This is expected, as the model was trained to localize dimmer objects with visual magnitudes 10-20. For visual magnitudes 11-14, this learned method demonstrates acceptable precision levels while enhancing recall beyond what synthetic tracking with SNR threshold 7.5 is able to provide. For enhancing recall levels beyond visual magnitude 14, we likely need to increase our cumulative exposure time. Mathematically, the signal emitted from NEOs is undifferentiatable from background noise without longer exposure times.

Apparent Object Magnitude	Precision	Recall	F1 Score
6-7	0.594	0.388	0.459
7-8	0.623	0.360	0.462
8-9	0.520	0.591	0.552
9-10	0.550	0.647	0.595
10-11	0.667	0.833	0.741
11-12	0.833	0.833	0.833
12-13	0.833	0.714	0.769
13-14	0.933	0.466	0.622
14-15	-	0.000	0.000
15-16	-	0.000	0.000

Table 3: YOLOv8 Performance

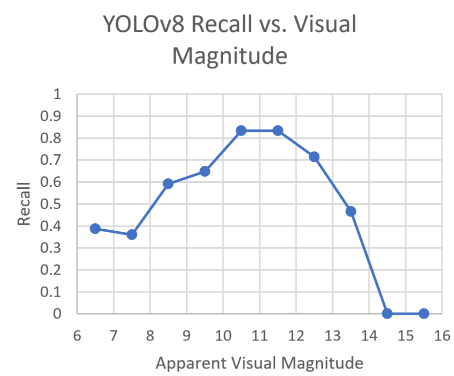

Fig. 9: Recall vs. Visual Magnitude

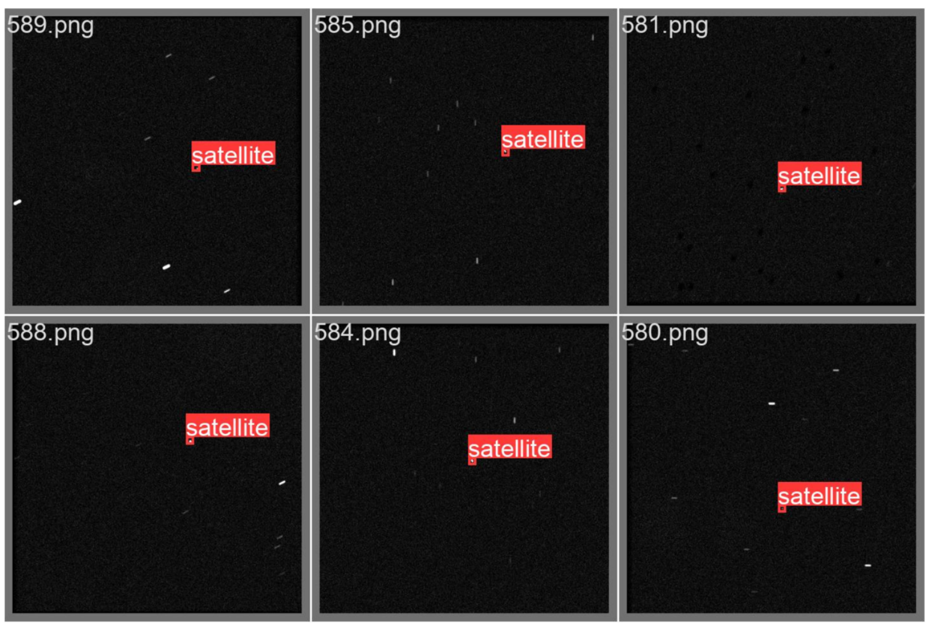

Fig. 10: Example Outputs from our Custom YOLOv8 model

7. SYNTHETIC TRACKING + YOLOV8 COMBINED PIPELINE

As demonstrated in the prior results, both synthetic tracking and YOLOv8 succeed at different SNR levels. In order to combine the success of these two detection algorithms, we assemble a pipeline for preprocessing data with synthetic tracking for YOLOv8.

For a set of sidereal frames, we explore 1000 possible velocities, resulting in 1000 images to analyze. For velocities that result in an SNR higher than 7.5 after stacking, the brightest pixel's location is reported. For every other velocity, it is analyzed by the custom model. When the object within a reported bounding box has an SNR higher than 2, it is reported as a detection. This threshold is designed is to prevent excessive false positives.

7.1 Synthetic Tracking + YOLOv8 Combined Pipeline Performance Metrics

Apparent Object Magnitude	Precision	Recall	F1 Score
6-7	1.000	1.000	1.000
7-8	1.000	1.000	1.000
8-9	1.000	1.000	1.000
9-10	1.000	1.000	1.000
10-11	1.000	0.826	0.905
11-12	0.833	0.833	0.833
12-13	0.833	0.714	0.769
13-14	0.933	0.466	0.622
14-15	-	0.000	0.000
15-16	-	0.000	0.000

Table 4: Synthetic Tracking + YOLOv8 Performance

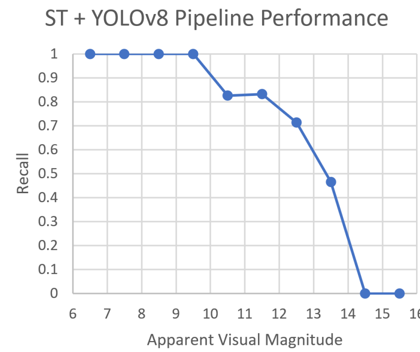

Fig. 11: Recall vs. Visual Magnitude

These results underscore the efficacy of combining synthetic tracking with deep learning, particularly in overcoming challenges associated with low SNR NEO detections.

Integrating synthetic tracking with the Custom YOLOv8 model significantly enhanced our ability to detect unknown NEOs with low SNRs. Through testing and validation, our methodology demonstrated substantial improvements in detecting and accurately localizing NEOs in the lower SNR ranges. Introducing the Custom YOLOv8 model corrected star confusion that occurred for in cases where the star mask was only partially complete. Overall, this pipeline demonstrates high precision while detecting more objects than synthetic tracking or deep learning would alone.

8. REPRODUCIBILITY

Synthetic tracking experiments were performed locally on Python 3.9.18, PyTorch 2.1.2, CUDA 11.2. Windows Subsystem for Linux 2 was utilized for astrometry compatibility with 5200 LITE indices.

Data was generated with SatSim 0.18.0 with the SSTRC7 for accurate real-world starfield generation.

Models were trained on NVIDIA T4 GPU on Google Cloud Platform.

9. CONCLUSION AND FUTURE WORK

This research illustrates a significant advancement in the field of astronomical observation, specifically in the detection and cataloging of NEOs. By integrating synthetic tracking with a Custom YOLOv8 deep learning model, we have not only expanded the scope of detectable NEOs but also significantly enhanced the accuracy and reliability of these detections.

This approach addresses a critical gap in traditional astronomical detection methodologies, which often struggle to identify objects with low SNRs. The combination of synthetic tracking to concentrate the signal of moving objects and deep learning to accurately identify and classify these signals has proven to be a powerful tool for detecting dim NEOs.

Future work will focus on further refining the deep learning model, expanding the dataset with additional simulated and real-world observations, and exploring integrating this methodology with other astronomical detection and tracking systems. The ultimate goal of this research is to provide a robust, reliable framework for detecting NEOs, thereby contributing to planetary defense efforts and space domain awareness.

In conclusion, the successful application of this integrated approach represents a significant step forward in our capability to detect, track, and study NEOs of unknown velocities, offering promising implications for both planetary defense and the advancement of astronomical research.

10. REFERENCES

[1] Larry Bradley, Brigitta Sipőcz, Thomas Robitaille, Erik Tollerud, Zé Vinícius, Christoph Deil, Kyle Barbary, Tom J Wilson, Ivo Busko, Axel Donath, et al. Astropy/photutils. *Zenodo*, 2016.

[2] Alexander Cabello and Justin Fletcher. Satsim: a synthetic data generation engine for electro-optical imagery of resident space objects. In *Sensors and Systems for Space Applications XV*, volume 12121, pages 53–74. SPIE, 2022.

[3] Justin Fletcher, Ian McQuaid, Peter Thomas, Jeremiah Sanders, and Greg Martin. Feature-based satellite detection using convolutional neural networks. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, page 11, 2019.

[4] Chris Harvey. *New algorithms for automated astrometry*. Citeseer, 2004.

[5] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics YOLO, January 2023.

[6] Dustin Lang, David W. Hogg, Keir Mierle, Michael Blanton, and Sam Roweis. Astrometry.net: Blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal*, 139(5):1782–1800, March 2010.

[7] Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. Cupy: A numpy-compatible library for nvidia gpu calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017.

[8] Michael Shao, Russell Trahan, Chengxing Zhai, Navtej Saini, and Slava G Turyshev. Synthetic tracking on a small telescope. 2018.