Feature-Based Satellite Detection using Convolutional Neural Networks

Justin Fletcher
Air Force Space Command
Ian McQuaid, Peter Thomas
Air Force Research Laboratory
Jeremiah Sanders
MD Anderson Cancer Center
Greg Martin
Centuari, Inc.

ABSTRACT

This work introduces a convolutional neural network that detects geosynchronous Earth orbit resident space objects in ground-based electro-optical telescope imagery. Model performance on a variety of object detection tasks is analyzed, and an extension of the general object detection algorithm assessment framework relevant to RSOs is described. Additionally, this work introduces two new datasets for the development and evaluation of detection algorithms. We report a maximum F1 point of 0.971, corresponding to 0.973 precision and 0.969 recall at a localization threshold of 8 pixels, which is the highest reported performance on the SatNet dataset known to the authors. Measured performance exceeds that of a classical detection algorithm, SExtractor, evaluated on the same task. We demonstrate improved sensitivity to objects clusters with smaller apparent separation by simulating low-frequency occurrences and augmenting natural training data. By incorporating temporal information using a recurrent extension of a detection model, we further improve sensitivity to dim and closely spaced objects.

1. INTRODUCTION

This work explores the application of convolutional neural networks (CNNs) to the detection of geosynchronous Earth orbit (GEO) resident space objects (RSOs) in ground-based electro-optical (EO) telescope imagery. The task of estimating RSO centroids in a frame is mapped to the estimation of object bounding boxes, and standard object detection CNNs are trained to perform the estimation task. Detection model training is complicated by a lack of diversity among naturally-collected data, and the absence of relevant phenomena in simulated data. We introduce two datasets to enable detailed analysis of model performance across relevant scenarios.

A brief overview of related works in both the astronomical and deep learning literature is given in Section 2. Section 3 provides a formalization of the detection problem, and maps the deep convolutional neural network metaheuristic approach to that problem. Section 4 describes the datasets developed in support of this work. In Section 5, four experimental protocols, each quantifying a separate aspect of object detection performance, are described. Section 6 describes the development software and hardware used for this study. Finally, Section 7 summarizes the findings of this study, and suggests several topics for future work.

2. RELATED WORKS

The detection of objects in ground-based electro-optical imagery is known as source extraction and has been studied primarily for astronomical applications. Source extraction consists of the localization of objects by

determining which pixels correspond to an object (a "source" of signal), and which are background. Automated approaches to source extraction generally involve the computation of a frame background estimation followed by either a peak-finding or intensity-thresholding step [18]. Of note, a foundational work in source extraction [11], upon which several modern applications are built essentially uses a hand-tuned convolutional filter to extract contiguous object regions of an image. Several modern works extend this approach via data-driven techniques including matched, scale-adaptive, and wavelet filtering [10, 17, 12, 13]. Several works have explored Bayesian methods for source extraction [6, 16], placing source extraction in astronomical surveys on rigorous statistical footing. Additionally, [2] used a single hidden layer multilayer perceptron to classify each detected object as either a star or galaxy using derived photometric information.

Object detection is a widely studied task in computer vision [3, 8], in which a system simultaneously localizes and classifies objects in an image. Localization is estimated as a bounding box, and class is estimated using a class score normalized over several classes. In this work we consider the application of a benchmark object detection CNN, You Only Look Once version 3 (YOLOv3) [14] to the problem of detection of RSOs in EO imagery.

3. FORMALIZATION

3.1 Detection

Let a detector of C object classes, D, be a function mapping the set of images, $\mathcal{X}$, to a set of inferences, $\hat{\mathcal{Y}}$, such that

$$D : \mathcal{X} \rightarrow \hat{\mathcal{Y}},$$

where $\mathcal{X} = \{x_i \in \mathbb{R}^{\mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}} \mid p \in \mathbb{N} \; \forall \; p \in x_i\}$ and $\hat{\mathcal{Y}} = \{\hat{y_i} \in \mathbb{Z} \times \mathbb{R}^{5+C}\}$. This formalism constrains a detector to produce from a two dimensional array of natural numbers, x_i, an integer-length list of object location inferences, $\hat{y_i}$. Each element of $\hat{y_i}$ is interpreted as a single object location inference of D, and comprises $5+C$ real numbers. The first 4 real numbers in an inference list element correspond to the minimum horizontal, minimum vertical, maximum horizontal, and maximum vertical location of the bounding box, respectively. The fifth real number is interpreted as the score prediction of the background class. The final C real numbers are interpreted as the inferred class score of C classes.

3.2 Deep Learning

Deep learning is a metaheuristic optimization technique by which a composition of parameterized nonlinear functions are conditioned to a dataset via back-propagation of errors [9, 15]. The functions most commonly optimized via deep learning are known as deep neural networks (DNNs). DNNs map a given input vector $\boldsymbol{x}$ to an output vector $\hat{\boldsymbol{y}}$, with the intent that $\hat{\boldsymbol{y}}$ is as close as possible to a target vector $\boldsymbol{y}$. We treat $\boldsymbol{x}$ and $\boldsymbol{y}$ as vector-valued random variables drawn jointly from a distribution, p_{data}. We denote the DNN mapping as $\hat{\boldsymbol{y}} = f(\boldsymbol{x}; \boldsymbol{\Theta})$, where $\hat{\boldsymbol{y}}$ is the computed estimate of $\boldsymbol{y}$ and $\boldsymbol{\Theta}$ parameterizes f, which is a function mapping $\boldsymbol{x}$ to $\hat{\boldsymbol{y}}$. We may further specify f as a composition of L layers, such that

$$f(\boldsymbol{x}) = f^{(L)}(f^{(L-1)}(...f^{(2)}(f^{(1)}(\boldsymbol{x}))))$$

where $f^{(i)}$ is said to be the i-th layer of the DNN. Each layer, i, computes a feature representation, $\boldsymbol{h}^{(i)}$, of the preceding layer given by

$$\boldsymbol{h}^{(i)} = g^{(i)}(\boldsymbol{W}^{(i)\top}\boldsymbol{h}^{(i-1)} + \boldsymbol{b}^{(i)}),$$

where $\boldsymbol{W}^{(i)}$ and $\boldsymbol{b}^{(i)}$ are trainable parameters comprising $\boldsymbol{\Theta}^{(i)}$, $g^{(i)}$ is an activation function, and $\boldsymbol{h}^{(0)} = \boldsymbol{x}$. If at least one $g^{(i)}$ is a squashing nonlinearity and $g^{(L)}$ is linear, this structure is known to be a universal function approximator [5].

In an L-layer network $\boldsymbol{h}^{(L)} \equiv \hat{\boldsymbol{y}}$. The loss between the inferred $\hat{\boldsymbol{y}}$ and the target $\boldsymbol{y}$ is quantified by a scalar-valued conditional function $J(\boldsymbol{\Theta}$. In this work, adaptive momentum [7] is utilized to compute the gradients of $\boldsymbol{\Theta}$ with respect to $J(\boldsymbol{\Theta})$, and to apply those gradients to update $\boldsymbol{\Theta}$. Through the iterative application of this technique, $J(\boldsymbol{\Theta}$ is minimized with respect to p_{data}.

The choice of g for each layer, the structure of $\boldsymbol{\Theta}$, the design of interconnections between each $\boldsymbol{h}^{(i)}$, and $J(\boldsymbol{\Theta})$ are problem-specific architecture decisions made by a practitioner during DNN development. Often, these choices are said to comprise a "model architecture." In this work, we utilize the SSD and YOLOv3 model architectures.

Both SSD and YOLOv3 are CNNs, and therefore require $\boldsymbol{x}$ to have the form $\mathbb{R}^{\mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}}$. Additionally, both SSD and YOLOv3 produce output feature maps ($\boldsymbol{h}^{(L)}$) which are interpreted by their specified loss functions ($J(\boldsymbol{\Theta})$) as bounding boxes for object detection, conforming to $\mathbb{Z}\times\mathbb{R}^{5+C}$. These architecture constraints admit a one-to-one correspondence between the deep learning metaheuristic, as realized by SSD and YOLOv3, to the problem of object detection.

3.3 Performance Quantification

Object detection performance is quantified using precision and recall, which are in turn computed from the counts of true positives (N_{TP}), false positives (N_{FP}), and false negatives (N_{FN}). Recall, or sensitivity, measures the ratio of true positives to the total count of detection targets, and is given by

$$\texttt{recall} = \frac{N_{TP}}{N_{TP} + N_{FN}}.$$

Similarly, precision measures the ratio of true positives to the total count of inferred objects and is given by

$$\texttt{precision} = \frac{N_{TP}}{N_{TP} + N_{FP}}.$$

N_{TP}, N_{FP}, and N_{FN} are computed by matching the detections inferred by a model to the ground truth annotations for an image.

To deterministically match ground truth boxes and probabilistic object inferences, a matching algorthim must be used. This algorithm is parameterized by two thresholds: confidence (T_c) and fit (T_f). T_c establishes the class confidence required to count a box as valid for the purposes of analysis; inference with confidences less than T_c will be ignored. Increasing T_c reduces N_{FP} and N_{TP} by excluding low-confidence estimates. T_f sets the spatial measure value required for an inferred and ground truth box to match. Large values of T_f increase N_{TP} at the expense of admitting poorly-localized inferences. Thus, T_f quantifies the trade-off between localization performance, and precision and recall. The intersect-over-union (IoU) between inferred box and ground truth box is the most common measure used in the object detection literature. In this work, all objects are less than one pixel in extent, and therefore do not admit IoU as a consistent localization measure. Instead, the L_2 norm between box centroids, measured in pixels, is used for localization in this work.

For each value of T_c and T_f, we filter the inferred boxes with confidence lower than T_c, then match the remaining inferred boxes to ground truth boxes. Box pairs with the lowest L_2 norm are matched first, if the L_2 norm is at least T_f. After matching is complete the number of matched boxes is N_{TP}, the number of unmatched inferences is N_{FP}, and the number of unmatched ground truth boxes is N_{FN}.

Each choice of T_c and T_f constitutes a trade-off between recall, precision, and localization. As such, it is customary to present each value as a point on a curve in precision and recall, and to present one curve, known as a precision-recall curve, for each choice of T_f. Each point on these curves has an F_1 score, given by

$$F_1 = 2 \cdot \frac{\texttt{precision} \cdot \texttt{recall}}{\texttt{precision} + \texttt{recall}},$$

which is the harmonic mean of precision and recall. The maximum F_1 point, F_1^*, for a given precision-recall curve provides a scalar quantification of the model performance equally weighting false positives and false negatives. Precision, recall, and F_1^* will be used to quantify detector performance throughout this work.

4. DATASETS

This work utilizes two datasets, SatNet and SatSim. Both datasets comprise examples containing an image, or collection of images, captured by ground-based electro-optical telescopes and corresponding annotations

for each image. An annotation contains a bounding box record for each object in the corresponding image, as well as metadata about the image and objects. In both datasets, object extents are less than the instantaneous field of view (IFOV) of a single pixel. This case is not handled in the typical bounding box annotation schema for object detection models. In order to remain coherent with existing literature, we pad object annotations to a 20×20-pixel bounding box centered at the object centroid.

4.1 SatNet Dataset

SatNet (version 1.1.0.0) comprises 104,100 annotated images. SatNet images were captured in rate-track mode against GEO targets and were collected over a two-year period by four sensors at three geographic locations. All SatNet images are 512×512 pixels in extent with a single 16-bit channel. SatNet images were annotated by a trained analyst, using annotation assistance software developed for this application. An automatic labeling approach leveraging space object catalog verification was attempted but introduced unacceptable variation in training annotations.

Sensors are composed of multiple subsystems (e.g., CCD, telescope, optics, etc.), each of which contributes distortion and noise to the collected data. The environment in which collection occurs also introduces noise, in the form of stray light, atmospheric distortion, and occlusions. The SatNet dataset contains collections from multiple small telescopes at diverse geographic locations. As such, many of these noise sources are present in the SatNet dataset. Several examples of observed noise are shown in the Fig. 1.

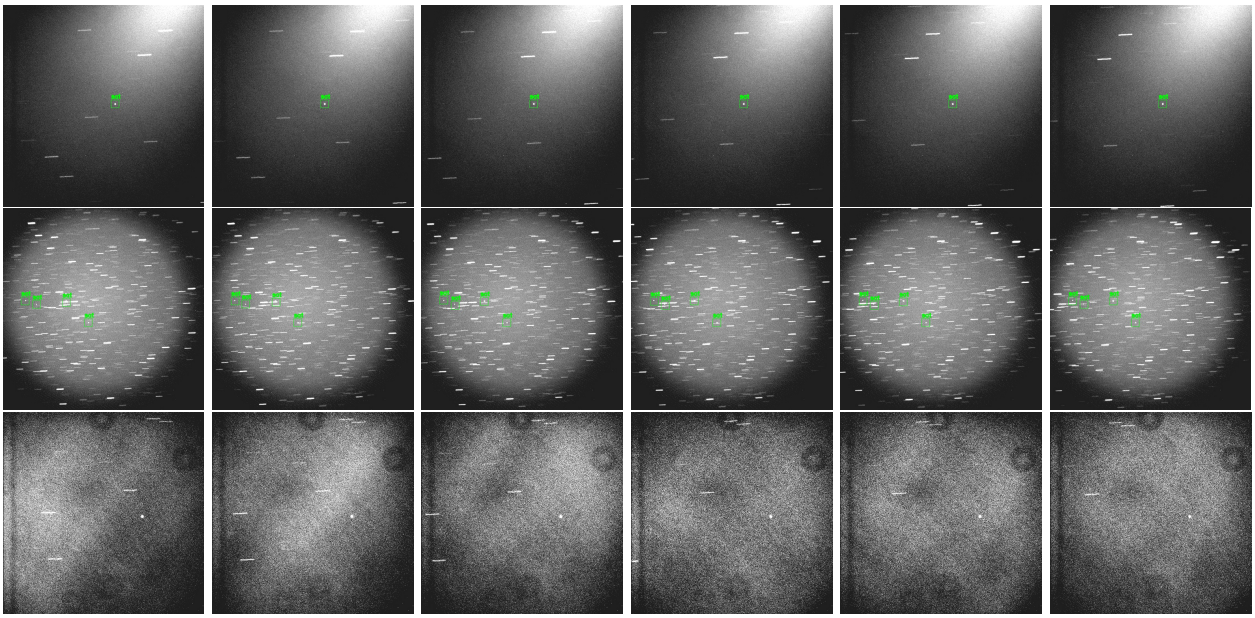


Fig. 1: Several example images from the SatNet Dataset, including annotations. Each row is a pass.

Images were collected in a six-frame "pass;" this collection regime results in images captured periodically at a fixed exposure time and collection cadence. Thus, images within a pass tend to have high spatial correlation. As such, the pass association of each image is maintained in the dataset, to ensure that images from a given pass will not be distributed across dataset partitions. This image segregation approach ensures that partitions remain independent and identically distributed from the underlying natural data distribution.

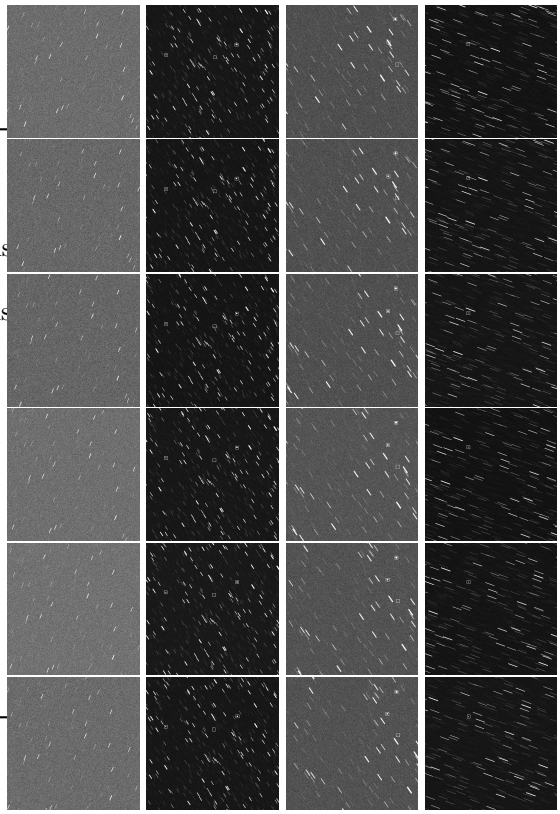
4.2 SatSim Dataset

SatNet inherits several human biases from the image annotation approach used, and several system biases from the relatively small number of sensors from which data was collected. For example, the dataset includes a selection bias for brighter objects and objects in low-background regions of an image. SatNet also lacks observations of scenarios that, while relevant to situational awareness in space, occur infrequently. Such scenarios include clusters of objects with dissimilar apparent magnitudes and small apparent separations.

Any data-driven model of object detection is likely to perform poorly in scenarios not represented in the training dataset. These limitations of the collected data in SatNet motivate the use of simulation to augment training data. Existing space scene simulators lack the interface and simulation throughput needed to enable practical generation of the data needed for this study. As such, a new scene simulator, SatSim, was developed. SatSim is implemented entirely in TensorFlow, is GPU accelerated, and has a modular interface enabling the rapid generation of large volumes of annotated imagery data. Examples of SatSim imagery are shown in Fig. 2b

Parameter	Value or Distribution
Spatial Oversampling Factor	15
Temporal Oversampling Factor	100
FOV	$U(0.1, 1.5)$ degrees
Dark Current	$U(1.0, 2.0)$ photoelectrons
Gain	$U(1.0, 2.0)$
Bias	$U(90, 110)$ photoelectrons
Zero Point	$U(21.0, 26.0)$
A2D Response	Linear
A2D Gain	$U(1.0, 2.0)$
A2D Bias	$U(9, 11)$ photoelectrons
Read Noise	$U(5, 20)$ photoelectrons
Electronic Noise	$U(5, 10)$ photoelectrons
PSF Mode	Gaussian
PSF Energy on Detector	$U(0.1, 1)$
Exposure Time	$U(1.0, 2.0)$ Seconds
Exposure Gap	$U(0.1, 1.0)$ Minutes
Objects per Image	$U([0, 1, 2, 3, 4, 5])$
Object Apparent Magnitude	$U(5.0, 20.0)$

(a) Parameters used in the procedural generation of sensors and scenes via SatSim. To create a sensor, a sample is drawn from the listed parameter distribution.



(b) Several examples from the SatSim dataset. Each column is a pass generated from a different simulated sensor.

Fig. 2: The parameter distributions used to generation SatSim dataset sensors, and several randomly selected scenes.

For this study, SatSim was used to produce a dataset comprising 50,000 images from 20 procedurally-generated sensors, for a total of 1,000,000 images. To enable coherent comparison with SatNet, these images were simulated in a six-frame pass collection regime. The relevant simulation parameters were drawn from several distributions, which are described in Table 2a.

5. EXPERIMENTS

This work comprises four independent experiments, each of which evaluates a different aspect of model performance on either the SatNet or SatSim dataset.

5.1 Resident Space Object Detection Performance Analysis

Data-driven methods require rigorous performance quantification. The quantification methods used in this work are described in Section 3.3. In order to establish a baseline of performance, we train a benchmark model and compare it to SExtractor. We train YOLOv3 using the training partition of the SatNet dataset,

and evaluate the performance of YOLOv3 and SExtractor against the a held-out test partition of the SatNet dataset. For YOLOv3 a DarkNet-53 backbone is used for feature extraction, and a single YOLOv3 predictor head is used to infer bounding boxes.

Performance is quantified as specified by Section 3.3 for YOLOv3. Performance quantification for SExtractor is also conducted as described in Section 3.3, with one exception. Unlike YOLOv3, SExtractor produces no confidence estimate. Instead, SExtractor parameterizes the trade-off between precision and recall by limiting the signal-to-noise ratio (SNR) required to declare a detection. Thus, SNR threshold is used for T_c, rather than a class confidence estimate. Fig. 3 displays the observed precision-recall curves for each implementation, and Table 1 summarizes the performance analysis of these curves.

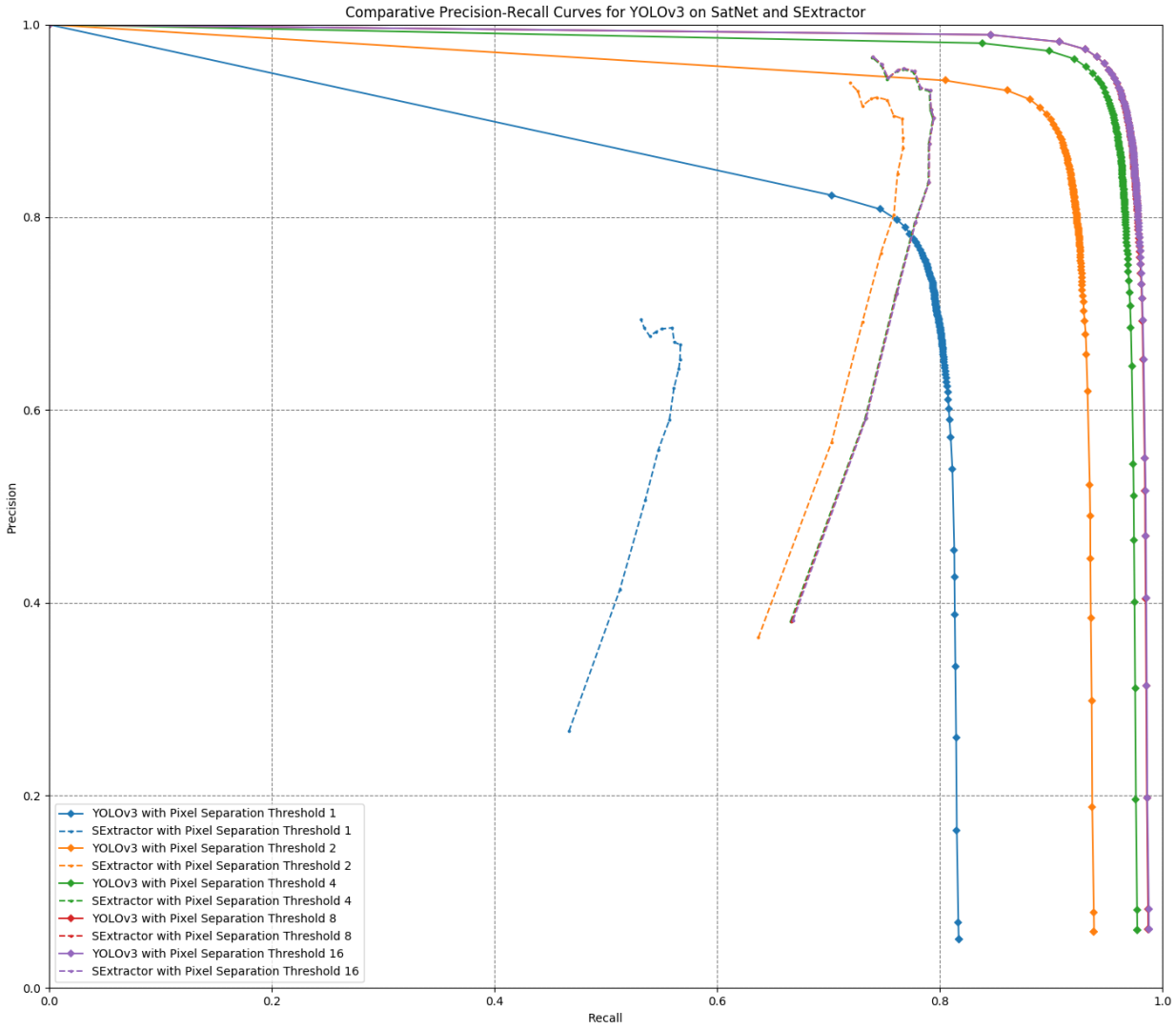


Fig. 3: Precision-recall curve comparison for YOLOv3 trained on the SatNet dataset and SExtractor at various localization accuracy thresholds. YOLOv3 precision-recall pairs are generated by varying class confidence threshold. SExtractor precision-recall pairs are generated by varying the estimated SNR required to declare a detection. All precision and recall values are assessed against a held-out test partition of the SatNet dataset. SExtractor was provided relevant calibration data for images in the test partition; YOLOv3 was provided no information from the test partition.

Performance is assessed at several fit threshold values, T_f, to characterize the localization performance

Table 1: Object detection performance measurements for SExtractor and SatNet YOLOv3.

Detector	T_f (L_2, Pixels)	F_1^*	Precision at F_1^*	Recall at F_1^*
SExtractor	1	0.599	0.673	0.5401
	2	0.818	0.901	0.750
	4	0.843	0.924	0.776
	8	0.843	0.924	0.776
	16	0.843	0.924	0.776
SatNet YOLOv3 (Ours)	1	0.780	0.773	0.788
	2	0.906	0.911	0.901
	4	0.956	0.960	0.953
	8	0.971	0.973	0.969
	16	**0.971**	**0.973**	**0.969**

of both approaches. There is minimal performance degradation for localization accuracy requirements of four pixels or greater. However, localizations accuracy requirements of two or less pixels result in substantial precision and recall degradation for both detectors. Noise in SatNet object localization annotation is difficult to characterize, but subjectively appears to be on the order of approximately two pixels. Thus, it is possible that the observed degradation in localization performance is caused, for both detectors, by annotation noise.

For each fit threshold and confidence or SNR threshold considered, SatNet YOLOv3 outperforms SExtractor in precision and recall. There is no choice of SNR threshold of SExtractor, for which a better-performing confidence threshold value of SatNet YOLOv3 cannot be found. When comparing performance relative to localization accuracy, we find that SatNet YOLOv3 F_1^* decreases by 0.191, whereas SExtractor F_1^* decreases by 0.244. Thus, SExtractor detection performance decreases more in absolute terms for detection applications requiring more precise localization. This result establishes SatNet YOLOv3 as the state of that art for GEO object detection in real EO imagery.

5.2 Cross-Sensor Generalization

In this work, trained models are not conditioned by sensor calibration data. Thus, in order to generalize well, the model must either learn sensor-specific noise patterns from training data or learn a detection function which is robust to sensor-specific noise. Robustness to sensor-specific noise patterns is quantified by evaluating model precision and recall on a test partition of the dataset on which the model was trained, as shown in Section 5.1. To estimate model cross-sensor generalization performance, the model must be cross-validated on sensor data from sensors unseen during training.

We estimate cross-sensor generalization error by cross-validating against unseen SatSim sensor data. Model cross-sensor generalization improvement is measured by training separate models on separate, variably-sized datasets. Each dataset in this experiment comprises between one and sixteen simulated sensors; for each sensor, $30,000$ simulated images from that sensor are added to the dataset. The models resulting from this training regime are then cross-validated against unseen SatSim sensor data. Additionally, we cross-validate against the SatNet test partition to assess model generalization to unseen noise cases (vignetting, clouds, stray light, etc.). The results of this experimental procedure are shown in Table 2. Comparing cross-validation performance between sensor counts, we find monotonic but diminishing improvement in performance as the number of training partition sensors is increased. From these results, we conclude that CNNs are capable of learning features that are robust to the details of a particular sensor or set of sensors. Of particular interest is the precipitous inscrease in SatNet cross-validation precision as more simulated sensors were included in training.

5.3 Model Sensitivity to Apparent Magnitude

Section 5.1 summarizes the model performance across the SatNet test partition, which contains representative examples of myriad obejcts and scenarios. There exist, however, subsets of the partition that comprising examples likely to present a challenge to detectors. These include examples containing objects with high apparent visual magnitude and in which RSOs have small apparent separation. Summarizing model

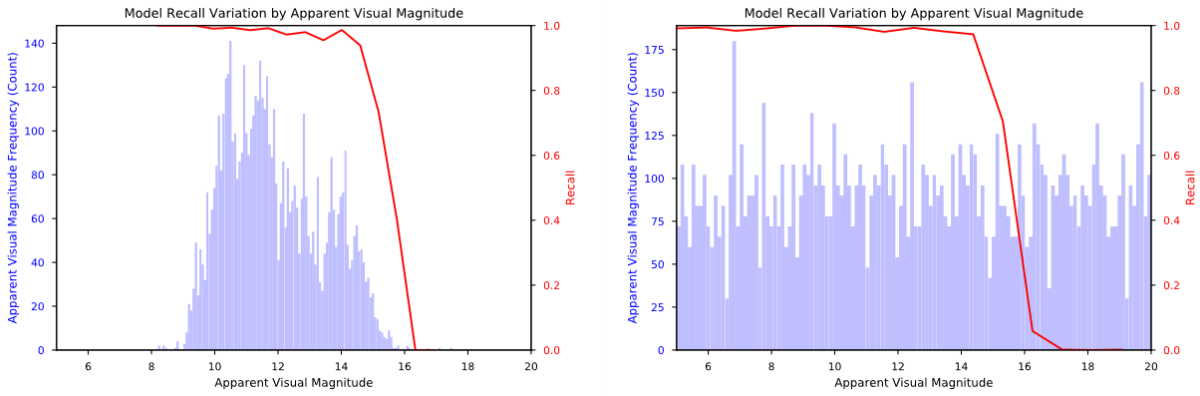
Table 2: YOLOv3 object detection performance for training datasets comprising 1, 2, 4, 8, and 16 simulated sensors, as measured by F_1^*, precision at F_1^*, and recall at F_1^*, when cross-validated against held-out test partitions of SatNet v1.1.0.0 and SatSim.

SatSim Sensors Count	SatSim Cross-Validation			SatNet Cross-Validation		
	F_1^*	Precision at F_1^*	Recall at F_1^*	F_1^*	Precision at F_1^*	Recall at F_1^*
1	0.747	0.968	0.608	0.131	0.073	0.661
2	0.726	0.976	0.578	0.117	0.064	0.662
4	0.807	0.979	0.687	0.342	0.219	0.776
8	0.830	0.979	**0.720**	0.342	0.263	**0.490**
16	**0.830**	**0.981**	0.716	**0.459**	**0.516**	0.413

performance across all examples obscures variance due to these challenging cases. In this section, model performance against dim targets is measured; Model performance against closely spaced objects is described in Section 5.4. For both experiments, we include training and evaluation for both SatNet and SatSim. SatSim is included so as to provide a larger population of dim and closely-spaced objects than occurs in SatNet.

To assess model performance across objects of varying apparent visual magnitude, magnitude values must be included in object annotation as metadata. SatSim uses apparent magnitude during scene simulation, and annotates them automatically. SatNet images are processed using AstroGraph, and each object is tagged with apparent visual magnitude estimated for that object. Objects annotated in SatNet but not detected by AstroGraph are omitted from this analysis. Model recall is assessed, as described in Section 3.3, for each example with magnitude metadata for all objects. Because the true positive and false negative counts are the only statistics that varies with respect to properties of ground truth object, recall is sufficient to characterize any changes in model performance. For consistency with other performance assessments, all assessments in this section will be performed using the F_1^* confidence threshold.

To assess CNN performance on this task we train two separate YOLOv3 models. One model, YOLOv3 SatNet, is trained on SatNet v1.1.0.0. The other model, YOLOv3 SatSim, is trained on the SatSim dataset described in Section 4.2. Each model is trained on the training partition of the corresponding dataset as described in 3.2 for approximately 100 epochs. Model recall is then measured against subsets of test partition of the corresponding dataset. Each subset comprises objects in a shared bin apparent magnitude values. Thus, model recall is assessed as a function of apparent magnitude of the target object. Fig. 4 displays (in red) the evaluated model recall for 16 bins of apparent visual magnitude. Additionally, the natural distribution of magnitude for both datasets is shown (in blue) as a histogram.

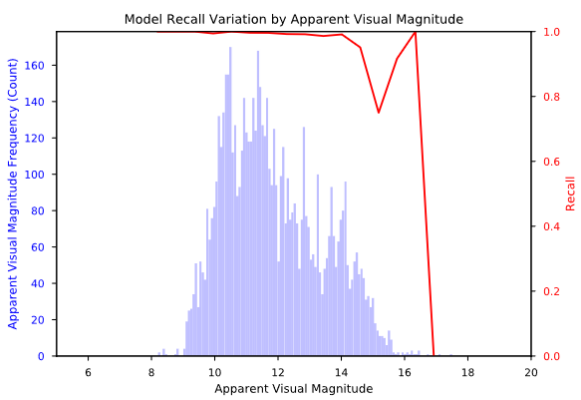


(a) Single-Frame YOLOv3 on SatNet

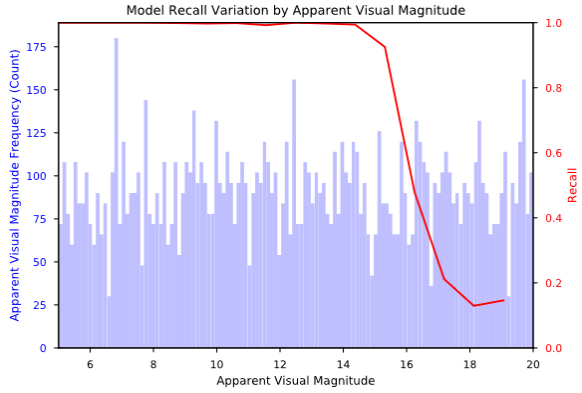
(b) Single-Frame YOLOv3 on SatSim

Fig. 4: Natural data distributions of apparent visual magnitude (blue), and trained model recall assessments by target apparent visual magnitude (red).

Reviewing Fig. 4a, we find that recall declines rapidly after approximately 14th visual magnitude, and reaches zero at approximately 16th visual magnitude. The decline in recall approximately follows the decline in data population, which is to be expected for trained models. Comparing Fig. 4a and Fig. 4b, we observe a small improvement in recall for very dim objects.



(a) Recurrent YOLOv3 on SatNet



(b) Recurrent YOLOv3 on SatSim

Fig. 5: Natural data distributions of apparent visual magnitude (blue), and trained recurrent model recall assessments by target apparent visual magnitude (red).

Human annotators report that it is easier to identify comparatively dim object when review a sequence of images, rather than a single image. This is possible due to variable realizations of noise between frames, and variations in object brightness. To leverage this additional information, we train a recurrent variant of YOLOv3. The recurrent YOLOv3 is realized as an LSTM block predicting object locations from the final feature map of the DarkNet-53 backbone. The experimental procedures to assess performance across object brightness are repeated with the recurrent YOLOv3 model. Results for this procedure are shown in Fig. 5.

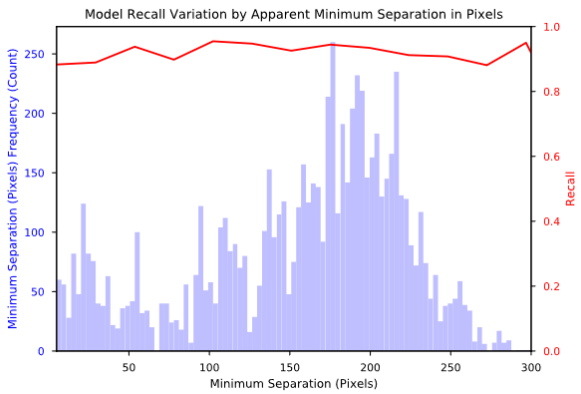
Comparing 4 and 5, we find substantial improvement in trained model recall on both SatNet and SatSim. SatNet recall improves for all data present in the dataset, but does not improve for the comparatively few objects above 17th apparent magnitude. Comparing 4b to 5b we find a considerable increase across all apparent magnitude subsets. Whereas the non-recurrent SatSim YOLOv3 reached 0.0 recall at approximately 17th magnitude, recurrent SatSim YOLOv3 achieved non-zero recall to 20th visual magnitude. This result is indicative of the ability of data-driven models to adapt to the data distributions upon which they are trained.

5.4 Model Sensitivity to Apparent Separation

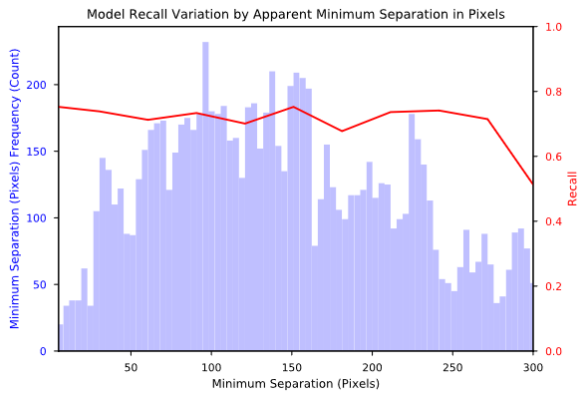
Like dim objects, closely spaced objects present a challenge for detectors. As two non-resolved RSO come into increasingly close proximity, they come to resemble one object, thereby causing false negative detections. The experimental procedure described in Section 5.3 may be applied to apparent separation, if object separation metadata is available. We compute apparent separation for each RSO by computing the L_2 norm, in pixels, to each other RSO and annotating distance to the object with the minimum apparent separation.

Model recall performance for several values of minimum apparent separation are reported in Fig. 6. We find that SatNet YOLOv3 and SatSim YOLOv3 are functionally invariant to minimum apparent separation. This result is consistent with Table 1, which indicates that model localization performance is invariant T_f as small 4. Comparing Fig. 6a and Fig. 6b reveals lower average recall performance for SatSim. This is a refection of the fact that a much larger portion of the SatSim population is very dim, relative to SatNet.

As in Section 5.3, we evaluate a recurrent model for model performance estimation a various apparent RSO separations. Results are shown in Fig. 7. We observe that, while recall remains invariant to minimum apparent separation, all assessed recall scores are improved. This may indicate that the model has learned to extract features which are more effective generally by leveraging the available temporal information.

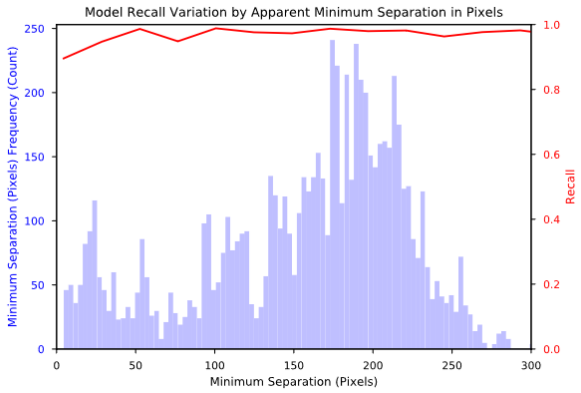


(a) Single-Frame YOLOv3 on SatNet

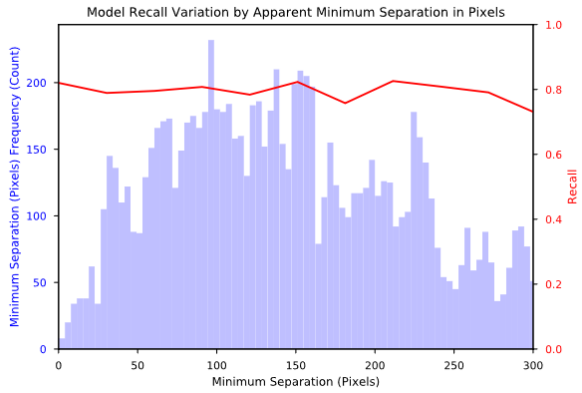


(b) Single-Frame YOLOv3 on SatSim

Fig. 6: Natural data distributions of apparent minimum separation (blue), and trained model recall assessments by target apparent minimum separation (red).



(a) Recurrent YOLOv3 on SatNet



(b) Recurrent YOLOv3 on SatSim

Fig. 7: Natural data distributions of apparent minimum separation (blue), and trained recurrent model recall assessments by target apparent minimum separation (red).

6. REPRODUCIBILITY

This work was conducted using TensorFlow version 1.14 [1] and Keras [4] on NVIDIA DGX Stations. A batch size of 28 examples per NVIDIA V100 was used for all single-frame experiments; recurrent model experiments used a batch size of 4 examples per V100. The SatNet v1.1.0.0 and SatSim datasets will be made available via the Unified Data Library, along with representative examples, working model demonstrations, and evaluation scripts.

7. CONCLUSION AND FUTURE WORK

This work establishes a new state of the art for GEO object detection in EO imagery, as measured by precision and recall on the SatNet v1.1.0.0 and SatSim datasets. We have quantified model robustness to dim and closely-spaced objects, and demonstrated that recurrence improves model recall for very dim objects. Future work should include detailed study of transfer learning between the simulated and real EO imagery, longer time-series detection performance on low SNR objects, and object tracking.

Acknowledgments

The authors wish to thank the Maui High Performance Computing Center for the use of computational resources necessary for the completion of this paper. Additionally, the authors thank Brandoch Calef for assistance in the production of SExtractor detections, and for useful feedback during the development of this work.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, Savannah, GA, November 2016. USENIX Association.

[2] E. Bertin and S. Arnouts. SExtractor: Source Extractor. Astrophysics Source Code Library, October 2010.

[3] Gong Cheng and Junwei Han. A survey on object detection in optical remote sensing images. *CoRR*, abs/1603.06201, 2016.

[4] F. Chollet and others. Keras: The Python Deep Learning library. Astrophysics Source Code Library, June 2018.

[5] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[6] M. P. Hobson and C. McLachlan. A Bayesian approach to discrete object detection in astronomical data sets. *Monthly Notices of the Royal Astronomical Society*, 338(3):765–784, 01 2003.

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[8] X. Wang L. Liu, W. Ouyang, P. W. Fieguth, J. Chen, X. Liu, and M. Pietikäinen. Deep learning for generic object detection: A survey. *CoRR*, abs/1809.02165, 2018.

[9] Yann Lecun, Leon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient backprop, 1998.

[10] M. López-Caniego, D. Herranz, R. B. Barreiro, and J. L. Sanz. Filter design for the detection of compact sources based on the neyman–pearson detector. *Monthly Notices of the Royal Astronomical Society*, 359(3):993–1006, 05 2005.

[11] R. K. Lutz. An Algorithm for the Real Time Analysis of Digitised Images. *The Computer Journal*, 23(3):262–269, 08 1980.

[12] J. L. Sanz D. Herranz-E. Martinez-Gonzalez R. B. Barrei, ro. Comparing filters for the detection of point sources. *Monthly Notices of the Royal Astronomical Society*, 342:119–133, 2003.

[13] L. Tenorio R. Vio and W. Wamsteker. On optimal detection of point sources in cmb maps. *Astronomy and Astrophysics*, 391:789–794, 2002.

[14] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[16] Richard S. Savage and Seb Oliver. Bayesian methods of astronomical source extraction. *The Astrophysical Journal*, 661(2):1339–1346, jun 2007.

[17] D. P. I Pierce-Price A. W. Blain . B. Barreiro J. S. Richter C. Qualtrough V. E. Barnard, P. Vielva. The very bright scuba galaxy count: Looking for scuba galaxies with the mexican hat wavelet. *Monthly Notices of the Royal Astronomical Society*, 352:961–974, 2004.

[18] H. K. C. Yee. A faint-galaxy photometry and image-analysis system. *Publications of the Astronomical Society of the Pacific*, 103:396–411, 1991.