

Model Optimization for Deep Space Exploration via Simulators and Deep Learning

James Bird^{1,*}, Kellan Colburn^{2,*}, Linda Petzold¹, Philip Lubin²

Abstract

Machine learning, and eventually true artificial intelligence techniques, are extremely important advancements in astrophysics and astronomy. We explore the application of deep learning using neural networks in order to automate the detection of astronomical bodies for future exploration missions, such as missions to search for signatures or suitability of life. The ability to acquire images, analyze them, and send back those that are important, as determined by the deep learning algorithm, is critical in bandwidth-limited applications. Our previous foundational work solidified the concept of using simulator images and deep learning in order to detect planets. Optimization of this process is of vital importance, as even a small loss in accuracy might be the difference between capturing and completely missing a possibly-habitable nearby planet. Through computer vision, deep learning, and simulators we introduce methods that optimize the detection of exoplanets. We show that maximum achieved accuracy can hit above 98% for multiple model architectures, even with a relatively small training set.

Keywords: computer vision, simulator, deep learning, space, universe, exoplanet, object detection, novelty detection, neural network, machine learning, transfer learning, interstellar travel, planet detection

1. Introduction

This paper will address some of the challenges and possibilities of exoplanet detection and classification for future exosolar system missions. Future missions may allow for travel far outside of our solar system, as well as deep into our own solar system, where return bandwidth will be severely limited; thus, choices of which data (images in particular) are

*Corresponding author

Email addresses: james_bird@ucsb.edu (James Bird), kellancolburn@ucsb.edu (Kellan Colburn), petzold@ucsb.edu (Linda Petzold), lubin@ucsb.edu (Philip Lubin)

¹Department of Computer Science, University of California Santa Barbara, 2104 Harold Frank Hall, Santa Barbara, CA 93106-9530

²Department of Physics, Broida Hall, University of California Santa Barbara, Santa Barbara, CA 93106-9530

important to "send back" (Lubin, P. (2016), Lubin & Hettel (2020), Sheerin et al. (2020)). The basis for exoplanetary detection via fast interstellar travel is a combination of The Starlight Program (Kulkarni et al., 2017) and recent results that show how exoplanets can be detected, and distinguished from other objects, via AI-based modeling that utilizes simulated data (Bird et al., 2020). The groundwork has been laid for an AI-based small spacecraft that can travel long distances in a short amount of time, gather information on its surroundings with minimal energy requirements, and detect exoplanets and other targets of interest with excellent accuracy. The same core technology we discuss here can be applied to a wide range of astrophysics and cosmology where subtle and often transient phenomenon are critical to retrieve in low SNR situations. In future papers we will discuss using our techniques in these other application spaces.

The major points that we will discuss and examine here are related to the accuracy of exoplanetary detection. In our foundational paper (Bird et al., 2020), we used a robust model and detection score for proof of concept. Going forward, this paper will compare a wide array of models using accuracy as our main metric to determine model strength and reliability.

2. Previous Work

The basis for much of our work lies in deep learning via TensorFlow (Abadi et al., 2016), as well as the expected additions, such as cuDNN (Chetlur et al., 2014) and CUDA, which allows for faster deep neural network processing via a graphics processing unit (GPU). Although the idea of direct exoplanetary detection and imaging via interstellar travel is new, astronomy has been attempting the general feat via light curves for years, and even more recently with deep learning (Shallue & Vanderburg (2017), Zucker et al. (2018), Carrasco-Davis et al. (2018)).

For direct imaging purposes, we test a variety of robust models, including variants of each model, and analyze factors such as accuracy and computational complexity. Since deep space is uncharted territory, an extremely large training data set is not possible. Therefore, we include some simpler models to offset the possibility of having models that are too advanced for the data. The overall goal of these models is to be able to identify when a planet is present in an image, while also being capable of not mistaking other astronomical objects for planets.

For the simpler models, we will compare MobileNet (Howard et al., 2017), MobileNet V2 (Sandler et al., 2018), DenseNet 121, 169, and 201 (Huang et al., 2018), and NASNet-Mobile (Zoph et al., 2017). These provide solid baseline accuracy and low computational complexity, which may prove to be beneficial for our specific needs. For the intricate models, we will compare NASNet-Large (Zoph et al., 2017), Xception (Chollet, F., 2017), VGG 16 and VGG19 (Simonyan & Zisserman, 2015), Inception V3 (Szegedy et al., 2015), Inception-ResNet V2 (Szegedy et al., 2016), and ResNet 50, 50 v2, 101, 101 v2, 152, and 152 v2 (He et

al., 2015). In contrast to the simpler models listed above, the training time and complexity will increase with these. However, that process is done beforehand while the wafer satellite (wafersat) is still on Earth, so these concerns are negligible when compared to the possible gains in accuracy from the more robust models.

These models have been tested against each other in the past to some degree. ResNet has been shown to out-perform VGG (He et al. (2015), Canziani et al. (2016)) and even advanced Inception models (Le et al., 2020), while other results show all of these models being out-performed by the DenseNet and InceptionResNet architectures (Zhen et al., 2018).

The structure of these models and their performance is dependent on the data that is being processed. In this case, we are training on simulated images of planets and testing on real images of planets. This concept was shown to be viable in Bird et al. (2020); however, optimizing this process will require an in-depth look at advanced deep learning techniques and models.

3. The Process

3.1. Deep Neural Network Architecture

Deep neural networks, including those used for object detection, begin by deconstructing images into pixel-based groupings that constitute an input layer. This layer, along with the hidden layer(s) and output layer, is comprised of smaller entities called neurons. Each layer of neurons is connected to the next via weights, which are learned through a training process. Gradient descent is a powerful and widely used method that allows us to minimize the cost function in order to get the most effective learning process. By taking the negative gradient, we minimize the cost function. After all is done, we are left with a network that can take an input image and output something of interest based on the training and model parameters. A more illuminating analogy would be to treat the initial inputs (pixels, or in the case of a convolutional neural network, groupings of pixels) as an input tensor. This input tensor is then essentially acted upon by a function (the neural network), which outputs a tensor corresponding to the the categorization of the input (in our case it is a binary categorization). This function initializes with random values, and is then optimized via the methods described above such that the output tensor has the highest accuracy when identifying inputted data.

3.2. The Setup

As discussed in detail in Bird et al. (2020), the simulator (SpaceEngine.org) provides us with easy access to 4K, 3-D rendered images of exoplanets. Although they are randomly generated, one could create a specific planet, or filter planets by a set of conditions in order to achieve a subset of planets that have certain traits. All models were pre-trained on ImageNet (Deng et al., 2009) and fine-tuned on simulated images of exoplanets. This allowed for a robust learning experience for features, and a more specific learning experience for our data

set. All models were evaluated using an AMD Ryzen Threadripper 3970X 32-Core Processor @ 3.70 GHz, 128 GB of RAM and an NVIDIA Titan RTX graphics card.

In most deep learning applications for image analysis, both training sets and testing sets contain images of the same object. In our deep learning application, the training set is taken from a universe simulator (SpaceEngine.org), and the testing images are real images of planets. Without a simulator, we would not have enough images of planets, and those planets would not constitute a large enough sub-sample of possible exoplanets. By using a physics-based simulator, we can produce an abundance of realistic novel exoplanets to train on. Then, we use real planets to test the model’s accuracy. This translates directly to the wafersats process during an actual interstellar journey. Image counts for all three sets is shown below in Table 1.

Training	Validation	Testing
915	200	284

Table 1: Image count for the training, validation, and testing sets.

The process being performed here is unique for two major reasons. First, the entire training set is simulated images, while the entire testing set is real images. This presents a particular challenge for neural networks, as they learn in a template space and are then tested in a real space. Second, deep space provides an enormously large variety of objects. For example, gas giants vary wildly in many ways, such as size, feature differences, surface gas formations, colors, temperature, and more. In our solar system alone, we witness two quite unique gas giants: Saturn with its rings and famous hexagon, and Jupiter with its eye and dolphin formations. Training on extremely unique objects can cause neural networks to lose their generality and under-perform. The images below in Figure 1 compare real images taken by NASA against simulated images.

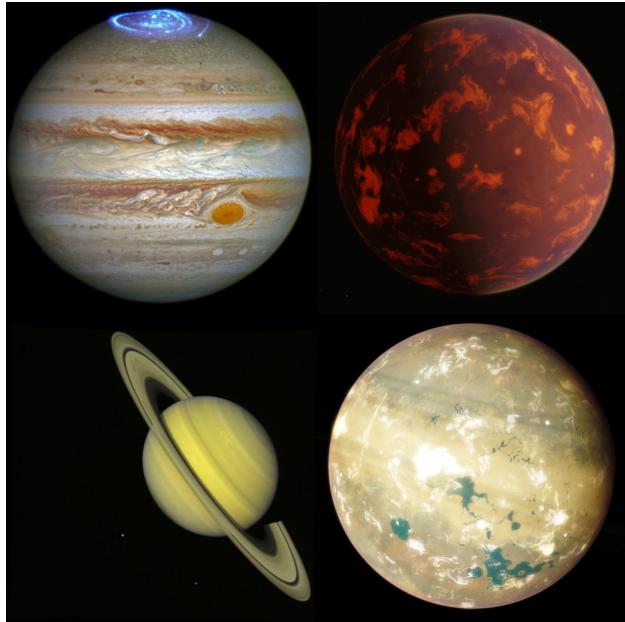


Figure 1: Two examples of real testing images are Jupiter and Saturn, seen on the left, while the right side shows examples of simulated training images.

4. The Results

For each model previously mentioned, we trained, validated, and tested the neural network in batches of five epochs each. An epoch is when the entire data set is run through the neural network once.

From Table 2 above, as well as Figures 2 and 3 below, it can be seen that VGG19 reached the highest maximum accuracy at five epochs, while VGG16 reached the second-highest maximum accuracy at both five and 25 epochs. This result is extremely interesting, as VGG variants are typically under-performing models when compared to Inception or ResNet variants. MobileNet performed extremely well, not only in maximum accuracy achieved, but also in terms of consistency. The remaining models were simply out-performed and provided no concrete reason why they should be chosen as a viable model for this specific task.

Model	Maximum Accuracy Achieved	Respective Epoch
VGG19	0.9964788556	5
VGG16	0.9929577708	5 & 25
ResNet50	0.9894366264	85 & 120
ResNet101	0.9894366264	115
ResNet152	0.9894366264	10 **
MobileNet	0.985915482	5
MobileNet v2	0.9788732529	5**
DenseNet121	0.9788732529	5
DenseNet169	0.9788732529	5 - 15
ResNet152 v2	0.9753521085	25 *
DenseNet201	0.9753521085	5-15
Inception v3	0.9683098793	5 & 15
ResNet101 v2	0.9647887349	5
ResNet50 v2	0.9612675905	10 - 20
NasNet-Mobile	0.9542253613	10
Inception-ResNet v2	0.950704217	5 & 10
Xception	0.950704217	10
NasNet-Large	0.9154929519	5

Table 2: Maximum epoch-based accuracy achieved for each model, ordered from highest to lowest. * denotes continued accuracy for all remaining epoch counts. ** denotes that maximum accuracy was sporadically achieved again after first occurrence.

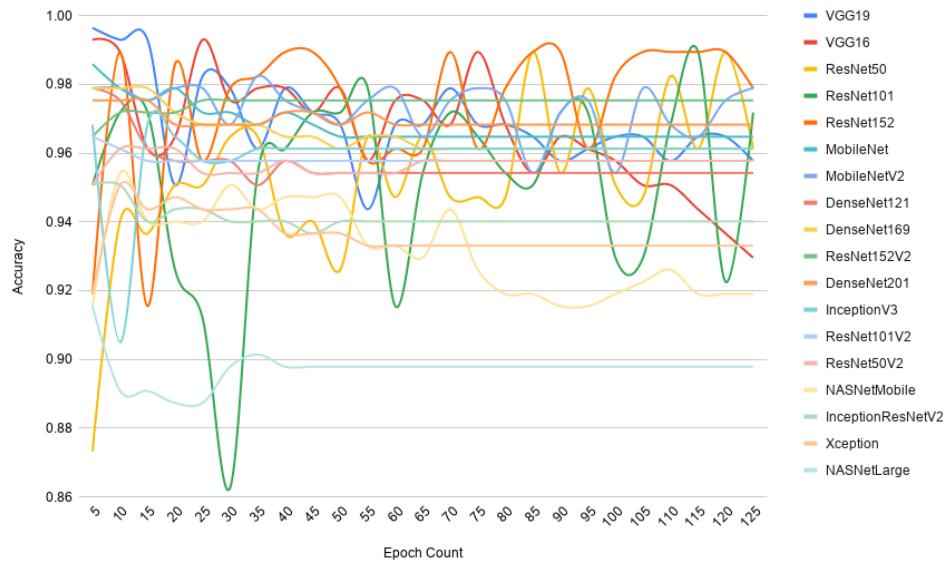


Figure 2: Accuracy of all models based on epoch count.

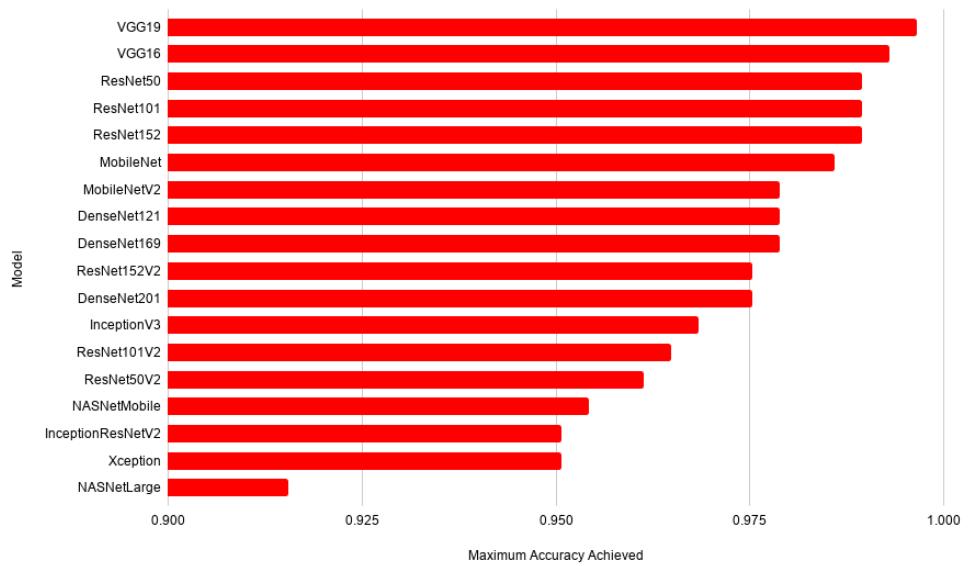


Figure 3: Accuracy of all models based on epoch count.

Based on Figure 4 below, ResNet50 and ResNet101 both dipped below 88% accuracy, and often times bounced from low to high accuracy, showing clear signs of inconsistency. Despite overall good performance, ResNet152 has large dips in the 5-20 epoch count, the main area where most models performed at their best. For these reasons, the ResNet variants were ultimately rejected as reasonable choices, as their epoch-based accuracy fluctuated too wildly.

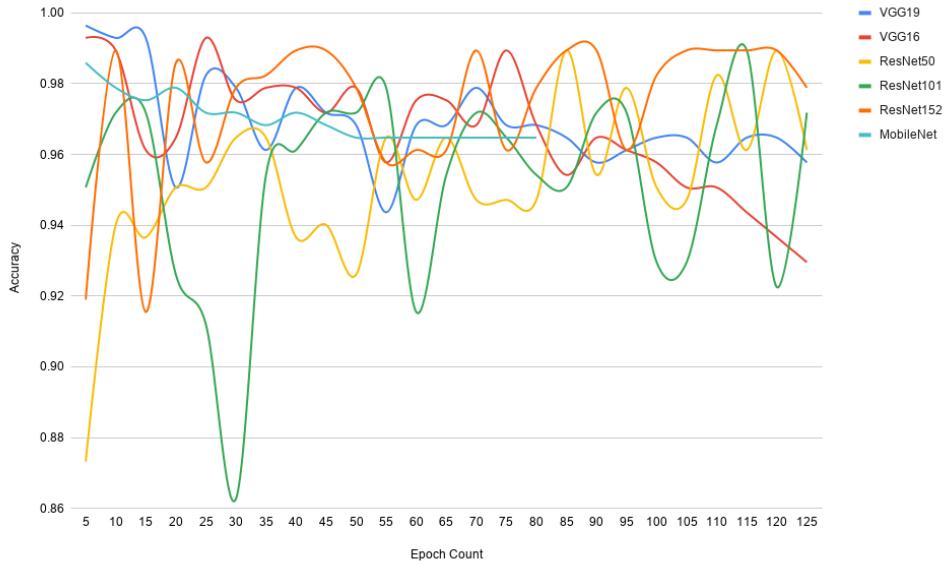


Figure 4: Accuracy of models that reached at least 98% maximum accuracy based on epoch count.

One very interesting result was concerning the remaining 98% or better maximum accuracy models, namely VGG19, VGG16, and MobileNet. Pertaining to Figure 5 below, we can see that VGG19 and VGG16 fluctuate to some degree, while MobileNet acts as a hedge. While the dependability of VGG19 and VGG16 in certain epoch ranges is vastly superior, MobileNet grants you a consistently strong choice across all epoch ranges.

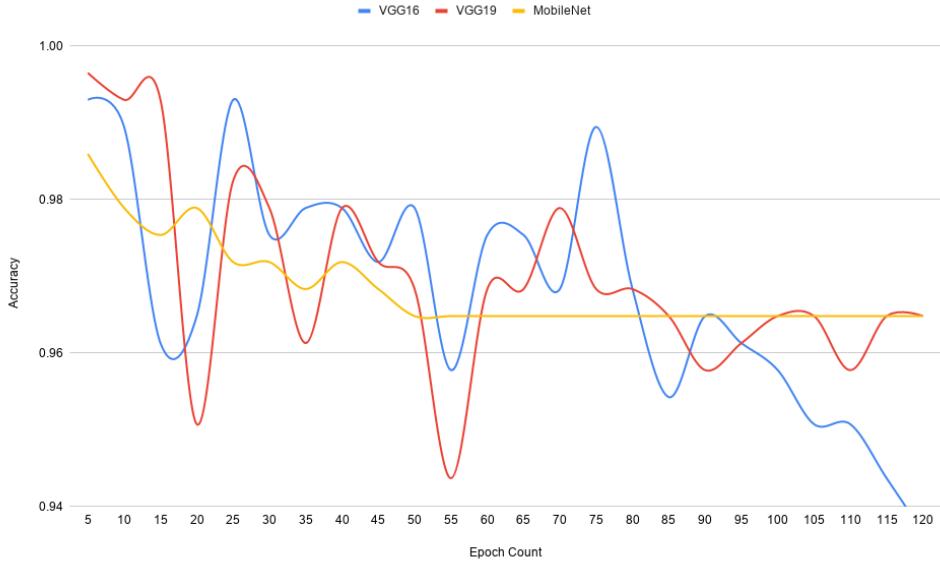


Figure 5: Accuracy of dependable models that reached at least 98% maximum accuracy based on epoch count. These do not include the ResNet variants.

We note that once ResNet variants are removed for their instability, every model reaches its peak accuracy at the 5-25 epoch range. Conditional on this range, the strongest models remain VGG19 and VGG16, but we can clearly see that their strength can fluctuate with small changes to epoch. Yet, MobileNet, MobileNet v2, DenseNet169, and DenseNet201 respectively perform the most consistently, while preserving most of the accuracy that we see in VGG19 and VGG16.

Some insight can be obtained by breaking down the accuracy into false negatives, which occur when a planet is present but the model does not identify it, and false positives, which occur when a planet is not present but the model identifies one. For our objective, false negatives are a much more severe error, as finding a planet and missing it is the worst possible situation. Alternatively, false positives would simply send back a picture of empty space to Earth, which would result in something mildly interesting, but nothing lost. Linking this information to our previous findings, ResNet variants continue to show instability, with some models having as many as 13 false negatives. Both VGG variants had no false negatives, meaning that they exhibit both extreme accuracy and reliability in detecting planets when they are actually present in the image. Lastly, we noted that DenseNet variants were very reliable in the 5-25 epoch range. In terms of false negatives, all DenseNet variants continue this stability with no false negatives.

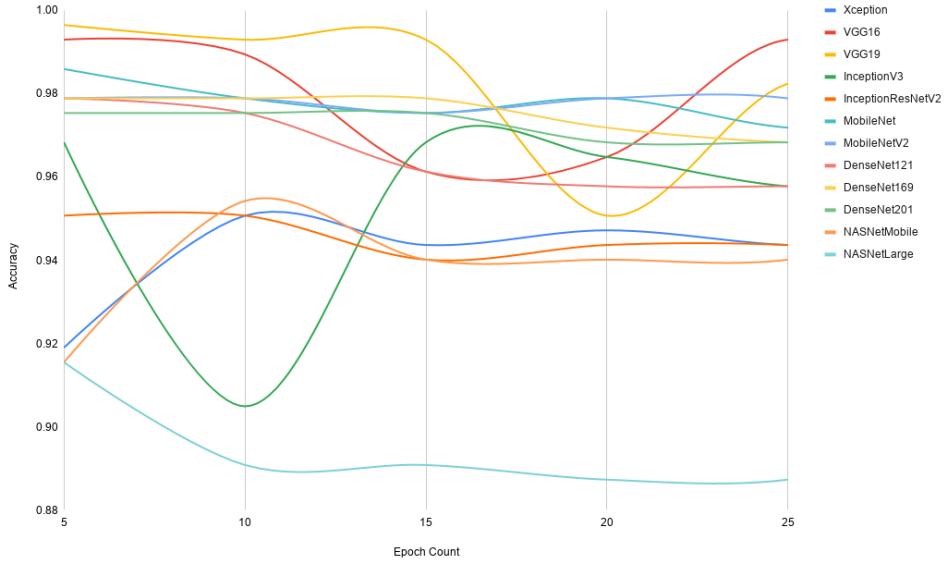


Figure 6: Accuracy of dependable models based on the 5-25 epoch range.

These results connect well to the previous section, where we outlined the unique circumstances that surround this particular problem. We note that we are training models on extremely specific objects in space, with unique features, patterns, colors, etc. Advanced models such as ResNet and Inception learn too well and proceed to analyze the extremely minute details in the training set. Then, when asked about other images that are unique in different ways, they struggle to find the connection. Meanwhile, less advanced models, such as VGG and MobileNet variants, do not lose that generality while learning.

4.1. Implications for future simulator training

Results show quite a few viable models, depending on whether we want extreme accuracy at the cost of variance (VGG variants), or dependability at the slight cost of accuracy (MobileNet variants).

Moreover, we note that this approach yields high accuracy with relatively few training images. With only 900 training images, we have achieved 98% accuracy with multiple models, giving us a wide variety of options depending on the situation. Thus, for future work, even small samples of simulated images can successfully train a neural network to detect real objects in space with extremely high accuracy.

5. Foundations for Future Work

We have expanded upon Bird et al. (2020) and have shown that a small subset of simulated images can produce extremely accurate predictions of real-world planets during an

interstellar journey and beyond. This paper solidifies the overall outlook on optimization methods for exoplanet detection, while introducing many ideas that will open new and exciting problems in deep space exploration. The same methodology can be used in a wide variety of astrophysical (and other) applications, where subtle issues in both the temporal and spatial domain are critical to access, and make decisions for, low bandwidth return applications.

An upcoming paper will address categorization, which will expand the ideas of simulator-based detection to objects beyond exoplanets. In particular, we will further explore whether simulators can help train neural networks to distinguish between specific types of planets. What about specific types of stars, comets, asteroids, and even more interestingly, signs of life?

6. Conclusion

In our previous work, we showed how simulator images could be used to successfully train a neural network to identify real images of planets. In this paper, we delve into specific model optimization and obtain some fascinating results. First, multiple models are proven to have above 99% accuracy when trained only on simulator images and tested on real images of planets. This result completely supports a simulator-based training model for deep space journeys, allowing us to train large neural networks pre-flight on Earth. Second, we have shown that extremely high accuracy does not depend on large data sets in this niche problem. With under 1,000 training images, we have achieved over 98% maximum accuracy with six different models. Finally, we demonstrate that there exists both high accuracy and high stability models that can perform well with no false negatives.

7. Acknowledgements

7.1. NASA

PML gratefully acknowledges funding from NASA NIAC NNX15AL91G and NASA NIAC NNX16AL32G for the NASA Starlight program and the NASA California Space Grant NASA NNX10AT93H, a generous gift from the Emmett and Gladys W. Technology Fund, as well as support from the Breakthrough Foundation for its Breakthrough StarShot program. More details on the NASA Starlight program can be found at www.deepspace.ucsb.edu/Starlight.

7.2. NSF

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562. Models were run through initial testing phases using the Comet GPU cluster, allocation ID: TG-CCR180013.

References

- Abadi, M., Agarwal, A., Barham, P., et al. 2016, TensorFlow:large-scale machine learning on heterogeneous distributed systems, arXiv:1603.04467v2 [cs.DC].
- Bird, J., Petzold, L., Lubin, P., Deacon, J. 2020, Advances in Deep Space Exploration via Simulators & Deep Learning, *New Astronomy*, Vol. 84.
- Canziani, A., Paszke, A., & Culurciello, E. 2016, An analysis of deep neural network models for practical applications, arXiv:1605.07678v4 [cs.CV].
- Carrasco-Davis, R., Cabrera-Vives, G., Förster, F., et al. 2018, Deep learning for image sequence classification of astronomical events, arXiv:1807.03869v3 [astro-ph.IM].
- Chetlur, S., Woolley, C., Vandermersch, P., et al. 2014, cuDNN: efficient primitives for deep learning, arXiv:1410.0759v3 [cs.NE].
- Chollet, F. 2017, Xception: Deep Learning with Depthwise Separable Convolutions, arXiv:1610.02357 [cs.CV].
- Deng, J., Dong, W., Socher, L., Li, K.L. & Fei-Fei, L. 2009, ImageNet: A large-scale hierarchical image database, 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
- Greaves, J.S., Richards, A.M.S., Bains, W. et al. 2020, Phosphine gas in the cloud decks of Venus, *Nat Astron* (2020), <https://doi.org/10.1038/s41550-020-1174-4>
- He, K., Zhang, X., Ren, S., & Sun, J. 2015, Deep Residual Learning for Image Recognition, arXiv:1512.03385 [cs.CV].
- Honniball, C.I., Lucey, P.G., Li, S. et al. 2020 Molecular water detected on the sunlit Moon by SOFIA, *Nat Astron* (2020), <https://doi.org/10.1038/s41550-020-01222-x>
- Howard, A., Zhu, M., Chen, B., Kalenichenko, D., et al. 2017, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, arXiv:1704.04861 [cs.CV].
- Huang, G., Zhuang, L., vander Maaten, L., Weinberger, K.Q. 2018, Densely Connected Convolutional Networks, arXiv:1608.06993 [cs.CV].
- Kulkarni, N., Lubin, P., & Zhang, Q. 2017, Relativistic spacecraft propelled by directed energy, arXiv:1710.10732 [astro-ph.IM].
- Le, H., Gupta, R., Hou, L., Abousamra, S., et al. 2020, Utilizing Automated Breast Cancer Detection to Identify Spatial Distributions of Tumor Infiltrating Lymphocytes in Invasive Breast Cancer, arXiv:1905.10841[eees.IV].
- Lubin, P. 2016, The Roadmap to Interstellar Flight, *Journal of the British Interplanetary Society - JBIS*, vol. 69, pp.40-72.
- Lubin, P. & Hettel, W. 2020, The Path to Interstellar Flight, *Acta Futura* 12, 9-45.
- Sandler, M., Howard, A., Zhu, M., Chmoginov, A., Chen, L-C. 2018, MobileNetV2: Inverted Residuals and Linear Bottlenecks, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp.4510-4520
- Shallue, C.J., & Vanderburg, A. 2017, Identifying exoplanets with deep learning: a five planet resonant chain around Kepler-80 and an eight planet around Kepler-90, *AJ*, 155, 94.
- Sheerin, T., Petro, E., Winters, K., Lozano, P., & Lubin, P. 2020, Fast Solar System Transportation with Electric Propulsion Powered by Directed Energy, *Acta Astronautica* 179, 78, Feb 2021.
- Simonyan, K., & Zisserman, A. 2015, Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv:1409.1556 [cs.CV].
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. 2016 Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, arXiv:1602.07261 [cs.CV].
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2015, Rethinking the Inception Architecture for Computer Vision, arXiv:1512.00567 [cs.CV].
- Towns, J., Cockerill, T., Dahan, M., Foster, I., et al. 2014, XSEDE: Accelerating Scientific Discovery, Computing in Science & Engineering, Vol. 16, No. 5, pp. 62-74, Sept.-Oct. 2014, doi:10.1109/MCSE.2014.80.
- Zhen, Y., Wang, L., Lui, H., Zhang, J., & Pu, J. 2018, Performance assessment of the deep learning

- technologies in grading glaucoma severity, arXiv:1810.13376 [cs.CV].
- Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V. 2017 Learning Transferable Architectures for Scalable Image Recognition, arXiv:1707.07012 [cs.CV].
- Zucker, S., Giryes, R. 2018, Shallow transits - deep learning I: feasibility study of deep learning to detect periodic transits of exoplanets, AJ, 155, 147.