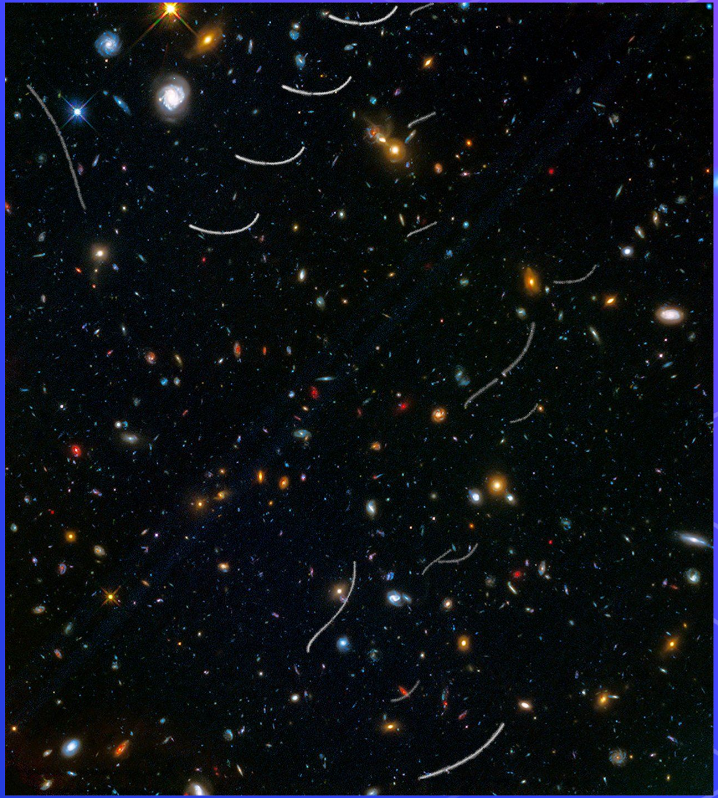
Modeling the Impact of Gaussian Noise on Machine Learning Detections of Near Earth Objects in Astrophotography

Rahsaan Simon

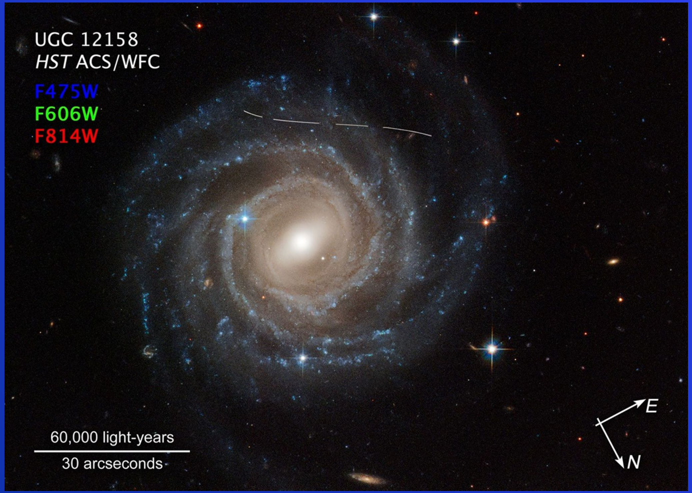
Abstract

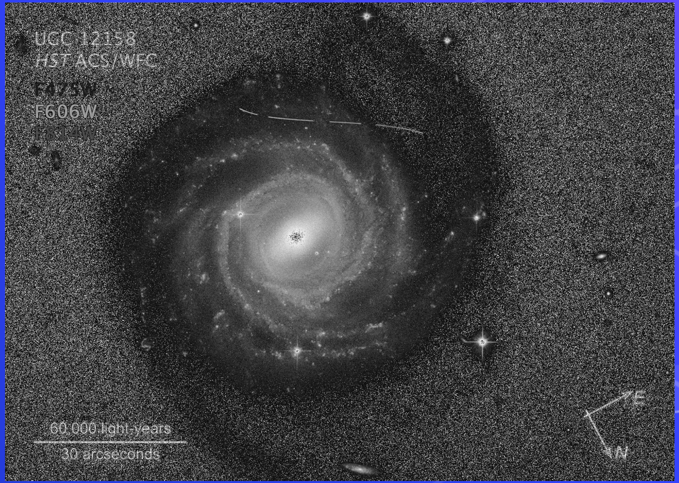
I tested artificial intelligence models' performance on detecting asteroids in astrophotography of increasing gaussian noise intensity to simulate space telescope identification of near earth objects. Results show that model performance deteriorates logarithmically against intensity, and can thus be modeled using a logarithmic function. Mathematical modeling may help engineers determine the maximum viable magnitude for which they can accurately detect NEO's quickly and without processing or immediately upon receiving space telescope images.



Question

Mathematically, what is the form of the general equation for deteriorating accuracy in artificial intelligence models when predicting near earth objects (NEOs, moving asteroids) from images, given the magnitude of noise?





Hypothesis

I hypothesize that the model's loss (which is inversely proportional to its accuracy), will have a linear dependence on the magnitude of noise, such that the resulting equation would be of the form

$$loss = nM + C$$

where n is the magnitude of dataset noise and M and c are variables dependent on the AI model.

I hypothesize this because an increase in noise itself has a linear effect on pixel values, and since AI models depend on these values they should scale linearly with the noise [1, 2, 3].

Background

Space telescopes present a good opportunity for detecting moving asteroids (or NEOs), which could potentially be threatening to the earth. Asteroids can be identified in many space telescopes and photographs, like Hubble, as being white streaks across otherwise clear images [5], due to long exposure time and fast movement. In general, when using a constant exposure time, a long streak indicates a faster object which, when being photographed near earth, suggests that the object could be a NEO. For this purpose, it would be useful to know what magnitude of noise is permissible so that the telescope can reserve resources when quickly transmitting images for the timely evaluation of potential asteroids and the assessment of their danger.

Noise that occurs during image transmission could be roughly approximated as gaussian. [3]

Materials

1. Computer with a Graphics Processing Unit
2. Python with MatPlotLib for data processing and Pytorch for model training
3. Model Architectures: resnet [7] and mobilenet [8]

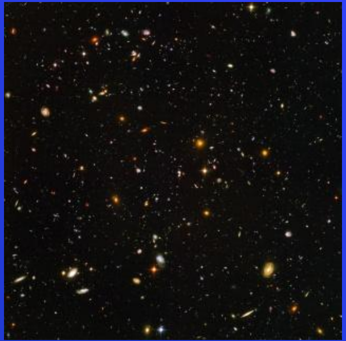
Procedure

1. Three types of astronomical photos of high quality, one that was slightly crowded, one that was moderately crowded and one that was highly crowded, were processed and resized
2. These three photos were then seeded with a singular fake asteroid in a random position (represented by a line that ranged in brightness) 1000 times each and split into testing and evaluating categories (500 images for test or train per type of crowdedness)
3. Two AI models,resnet and mobilenet, were finetuned/trained on a singular type of image and saved, making 6 total models.
4. These AI models were repeatedly evaluated on the testing data for their respective image type. With each iteration, the magnitude of noise increased a constant amount (0.01) and the resulting loss was recorded and graphed

Pictures

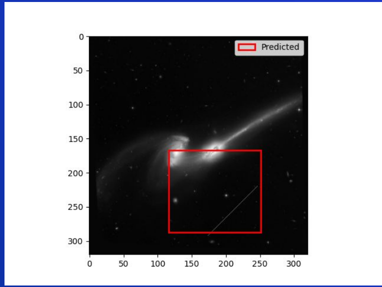
Images from Hubble Space Telescope [4]

Slightly Crowded

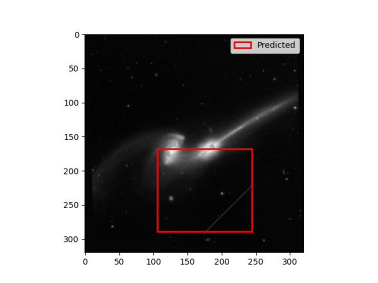


Moderately Crowded

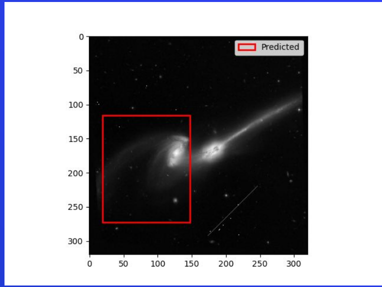
Highly Crowded



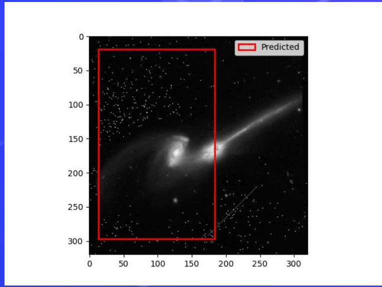
Rough Prediction of Asteroid (streak) location by resnet for n = 0



Slightly deviated prediction of asteroid location by resnet for n = 0.01

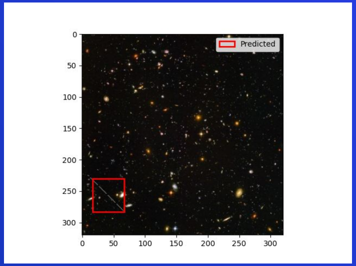


Noticeably deviated prediction of asteroid location by resnet for n = 0.05

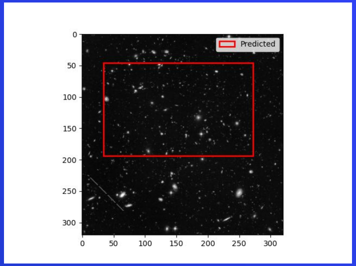


Very deviated prediction of asteroid location by resnet for n = 0.1

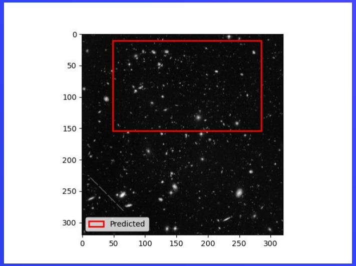
Pictures (cont'd)



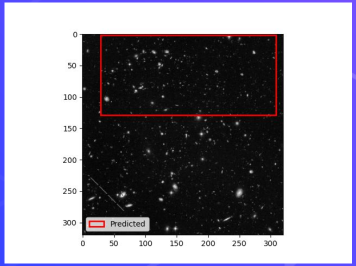
Precise prediction of Asteroid (streak) location by mobilenet for n = 0



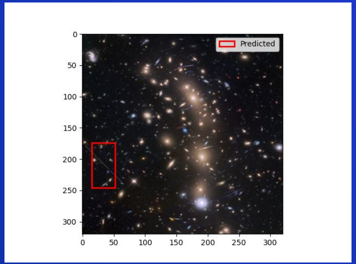
Noticeably deviated prediction of asteroid location by mobilenet for n = 0.01



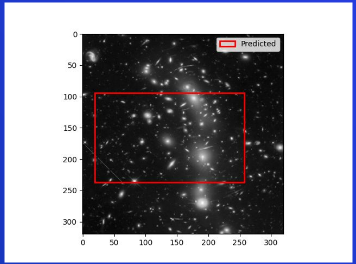
Very deviated prediction of asteroid location by mobilenet for n = 0.05



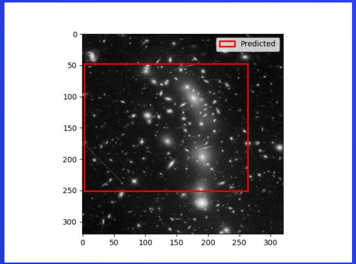
Very deviated prediction of asteroid location by mobilenet for n = 0.1



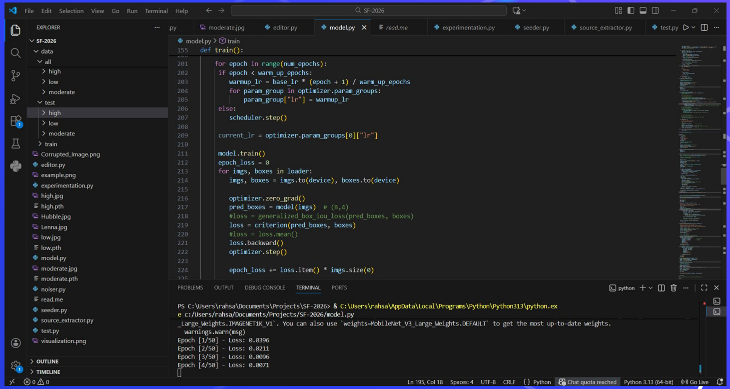
Precise prediction of Asteroid (streak) location by mobilenet for n = 0



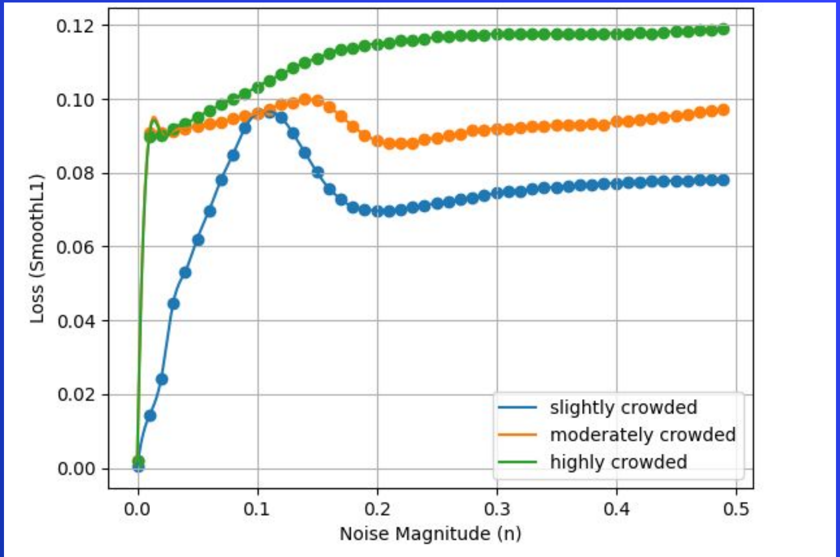
Noticeably deviated prediction of asteroid location by mobilenet for n = 0.01



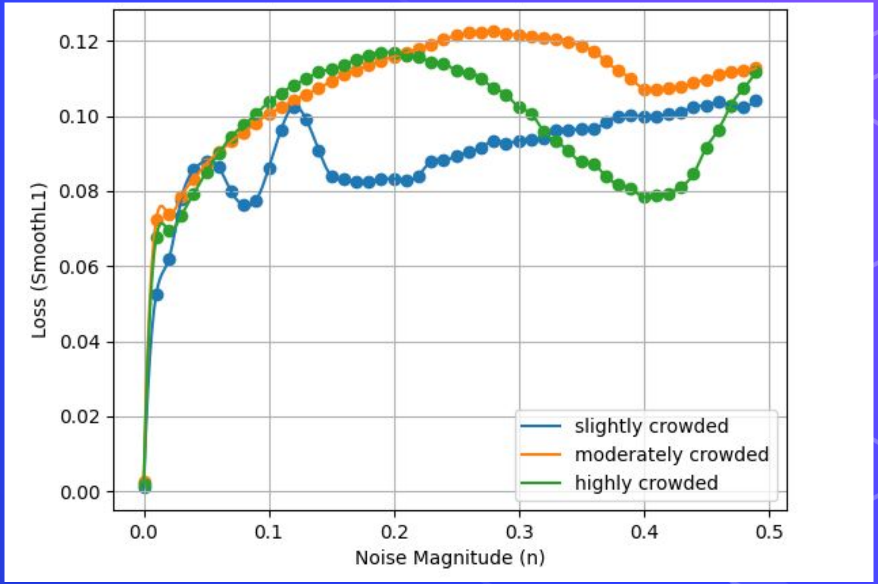
Very deviated prediction of asteroid location by mobilenet for n = 0.05



Results (Graph)



Resnet18 Average Loss vs Noise for Asteroid Location Predictions



MobileNetV3 (Large) Average Loss vs Noise for Asteroid Location Predictions

Discussion

The results show that for both models the loss (which is inversely proportional to the accuracy) increases nonlinearly with the noise magnitude (n). This implies that, across AI models, there is a function which is independent of the model that can be used to represent the loss as logarithmic. Therefore, the general equation for modeling loss as a function of noise magnitude should be of the form

$$loss = Mlog(n) + C$$

Conclusion

In conclusion, the results suggest that there is a nonlinear correlation between model loss and noise, contrary to my hypothesis. Furthermore, the results also suggest that this relationship can be mathematically approximated using a logarithmic function. They are suggestive of epistemic deep learning model uncertainty [6] for distorted images, as they may quickly become unaware of how to handle patterns when they are slightly interrupted. Since noise can be random and is not discrete, it would be impractical to attempt to train models on many different forms of noise, thus there would have to be some compromise between training the model more effectively and determining via math what level of noise would be the most acceptable when trying to detect quickly NEO objects in realistic scenarios.

Although the results suggest that there is a logarithmic relationship between loss and noise, there is room for improvement in terms of better training the model and gathering realistic telescope imagery data for training instead of seeding template images.

Bibliography

[1] IBM. (2021, July 27). What is Computer Vision? Ibm.com. https://www.ibm.com/think/topics/computer-vision

[2] Ayyadevara, V. K., & Reddy, Y. (2020). Modern Computer Vision with PyTorch: Explore deep learning concepts and implement over 50 real-world image applications. Packt Publishing Ltd.

[3] Image Synthesis - Noise Generation. (n.d.). Homepages.inf.ed.ac.uk. https://homepages.inf.ed.ac.uk/rbf/HIPR2/noise.htm

[4] Hubble Images - NASA Science. (n.d.). Science.nasa.gov. https://science.nasa.gov/mission/hubble/multimedia/hubble-images/

[5] Kruk, S., Martín, P. G., Popescu, M., Merín, B., Mahlke, M., Carry, B., Thomson, R., Karadag, S., Durán, J., Racero, E., Giordano, F., Baines, D., Marchi, de, & Laureijs, R. (2022). Hubble Asteroid Hunter: I. Identifying asteroid trails in Hubble Space Telescope images. ArXiv (Cornell University). https://doi.org/10.48550/arxiv.2202.00246

[6] Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. Machine Learning, 110(3), 457–506. https://doi.org/10.1007/s10994-021-05946-3

[7] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. ArXiv. https://doi.org/10.48550/arxiv.1512.03385

[8] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.