Exo-Hunt: Discovering Planets with Deep Convolutional Neural Networks

Simon Camacho, Moritz Stephan, and Griffin Miller

CS 231n

Stanford University

Background/Introduction

- Propose a novel method for detecting exoplanets from H-Alpha (real planets) and Continuum (injected planets) images using deep convolutional networks (36.1% overlap accuracy and 24.7% box-count accuracy)
- Build on existing methods for identifying planets in images using starlight removal
- Explore different network architectures, building upon Faster R-CNN
- Experiment with CNN backbones and channel downsampling

Problem Statement

- Input: 451x451x9 H-Alpha or Continuum image
- Use the Faster R-CNN object detection model to classify and predict different planet bounding boxes on the input image
- Use two evaluation metrics:
  - Count the number of images for which our model at least predicted one bounding box correctly containing a planet (overlap accuracy)
  - Divide the number of planets over all train, validation, and test images for which we predicted a bounding box correctly containing the planet by the total number of planets (box-count accuracy)

Dataset

- Professor Kate Follette (Amherst College) gave us her proprietary data, where each image in the input data is a 451x451x9 pixel image
- Classification dataset contains 6700 real images (800 train and 200 test chosen at random)
- Obj. detection dataset contains 4033 real/fake planetary images
- At random, 360 images chosen for training set, 48 for validation set and 72 for test set
- For classification task, each image is divided into four separate images, with each image representing one quadrant of the original image
- For obj. detection task, each image is randomly rotated between 0 and 360 degrees for robustness

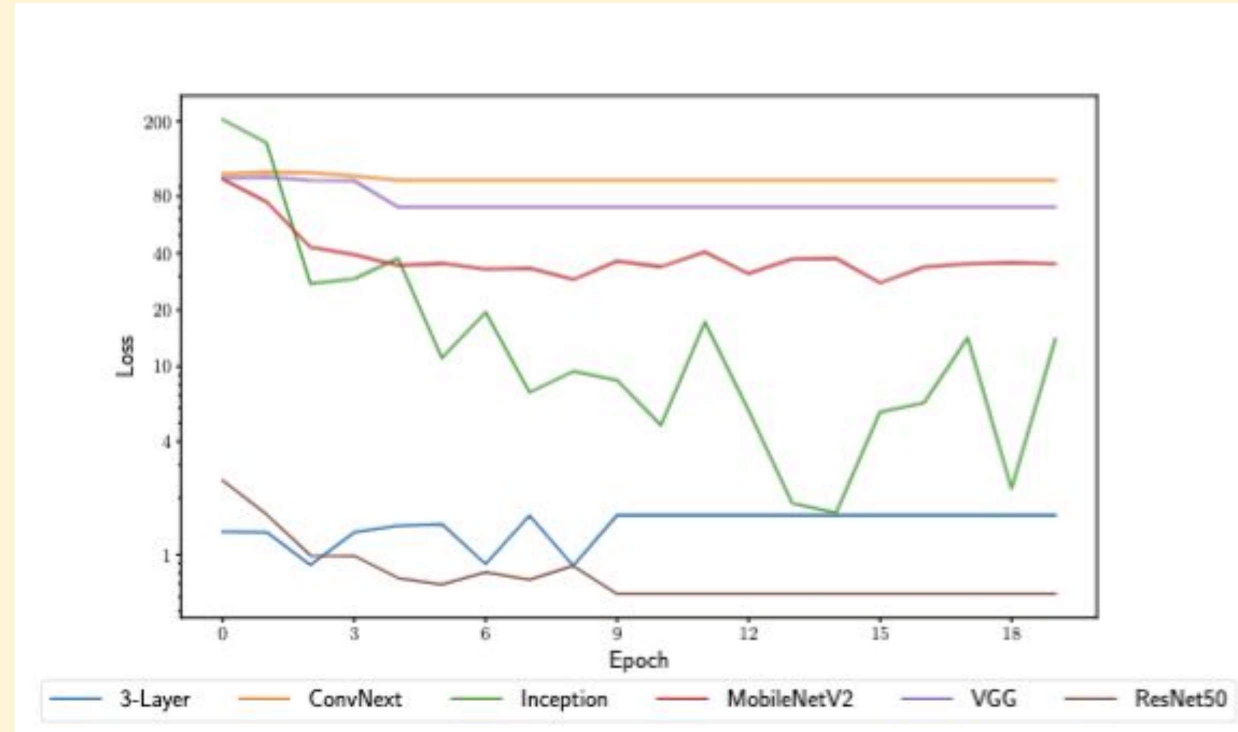

Figure 1: Loss graphs for overfitting on different model backbones. Epoch size varies due to model throughput speed differences and early stopping.

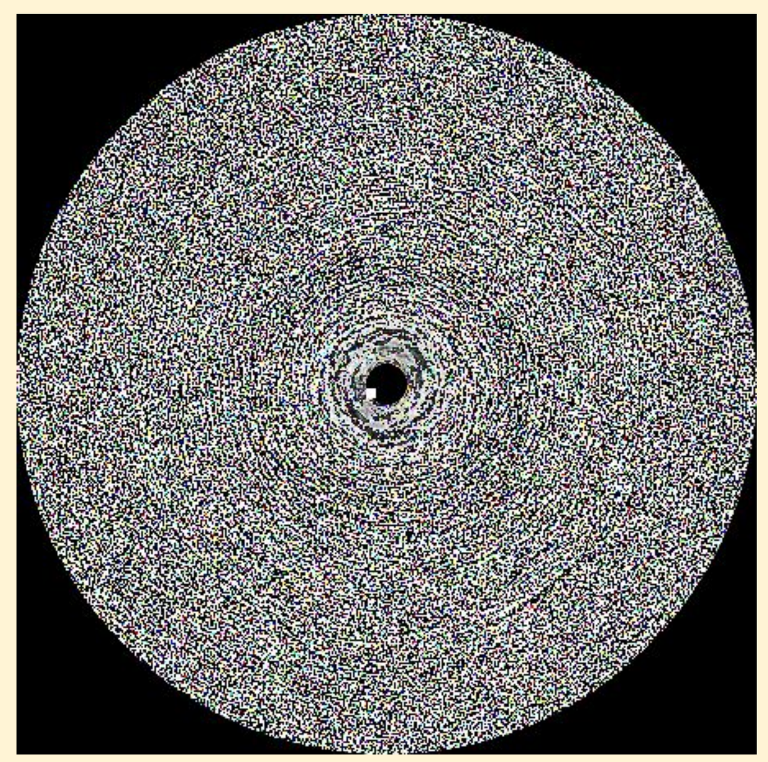

Figure 2: RGB visualization of image containing real exoplanet downsampled to 3 channels using mean. Box at star location.

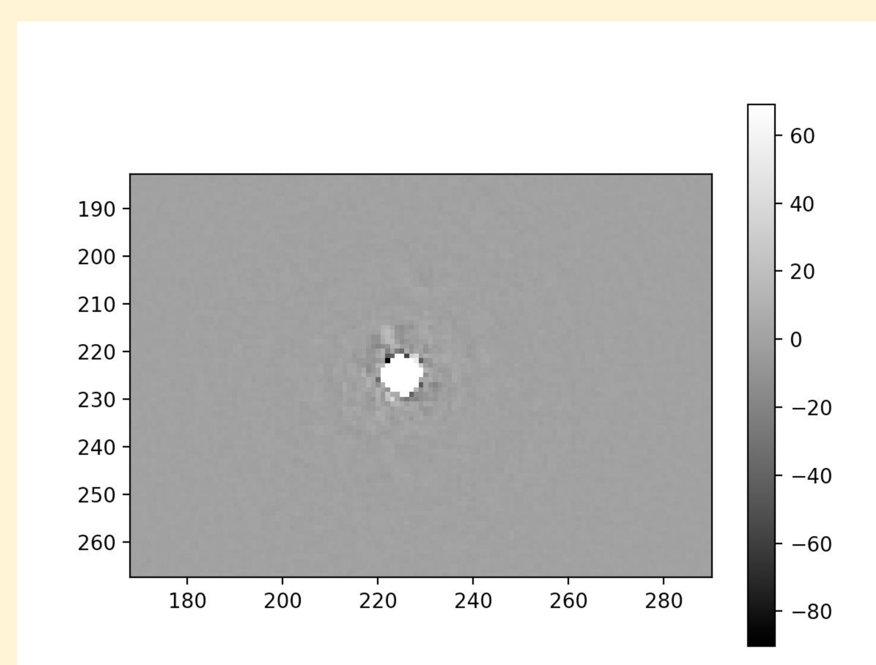

Figure 3: Single channel representation of a star (center) with planet just above it (the blurred shape).

Methods

- Implemented novel data augmentation method for classification task
  - Divide each “real” image into 4 quadrants, one containing the planet and the others empty (4x dataset size)
- Explored using many different CNN backbones for the Faster R-CNN architecture
  - Custom 3 Layer Convolutional network
    - [Convolution-BatchNorm-ReLU-MaxPool] x2 - [Convolution-BatchNorm-ReLU].
  - ResNet-50, VGG16, MobileNetV2, InceptionV3, ConvNext-Tiny
- Implement different downsampling schemes for 9-channel inputs
  - Mean, max, min, and sum
  - 1x1 convolution with 9 channel input and 3 channel output

Experiments & Analysis

- Conv and Mean channel downsampling outperform with 80%-100% and 20% overfitting train accuracy
- Custom 3-layer backbone and InceptionV3 outperform other backbone models with 80% and 20% overfitting accuracy respectively
- Achieved 98.53% accuracy on planet image classification baseline task. Proves that planets are detectable with computer vision.
- Achieved 36.1% overlap and 24.7% box-count accuracy on the detection task using a custom 3-layer backbone and mean channel downsampling, but there are many false positives. Images contain lots of noise and planets are often just 2 pixels wide, so difficult

Conclusion & Future Work

- Experimented with data augmentation, hyperparameters, data invariances, and different CNN designs
- Found that using simpler backbone CNNs can on average improve overlap accuracy by around 60% based on overfitting tests
- We would be interested in exploring the influence of CNNs on other methods of exoplanet discovery, such as as radial velocity, transit, and gravitational microlensing
- Could also measure precision, recall, and F-1 in order to collect more task-relevant performance metrics
- Improve our method and use it on dataset of candidate images to actually find novel exoplanets
- Implement non-deep learning detection baselines and compare performance