RPC 2020

Virtual Research Presentation Conference

Enabling astrometric exoplanet detection using Diffractive Pupil and ML-based distortion correction

Principal Investigator: Eduardo Bendek (383A)
Co-Is: Umaa Rebbapragada (398), Mark Wronkiewicz (398), Aram Hamidi (393)
Program: Spontaneous Concept

Assigned Presentation # RPC-023

Jet Propulsion Laboratory
California Institute of Technology

Tutorial Introduction

Abstract

Detecting exoplanets and measuring their masses is a priority for the astrophysics. Astrometry is the only exoplanet detection technique that can unequivocally measure exoplanet masses. The signal of an earth-like planet around nearby stars ~ 1 micro arcsecond (µas), while current instrumentation reaches (Hubble, GAIA) reaches 25 µas at best. The main limiting factor, after photon noise, to measure the astrometry signal (vector that indicates the target's motion) are optical distortions that arise from small deformation of the optical system. A novel technique called Diffractive Pupil (DP) allows obtaining distortion-calibrated astrometry vectors from the image. Currently, we use analytical methods to extract the distortion map from the diffractive features, however different sources of noise limit the accuracy of the algorithm. We hypothesize that ML could directly estimate the astrometry vector from data containing the diffractive pupil calibration fiducials with higher accuracy than analytical methods

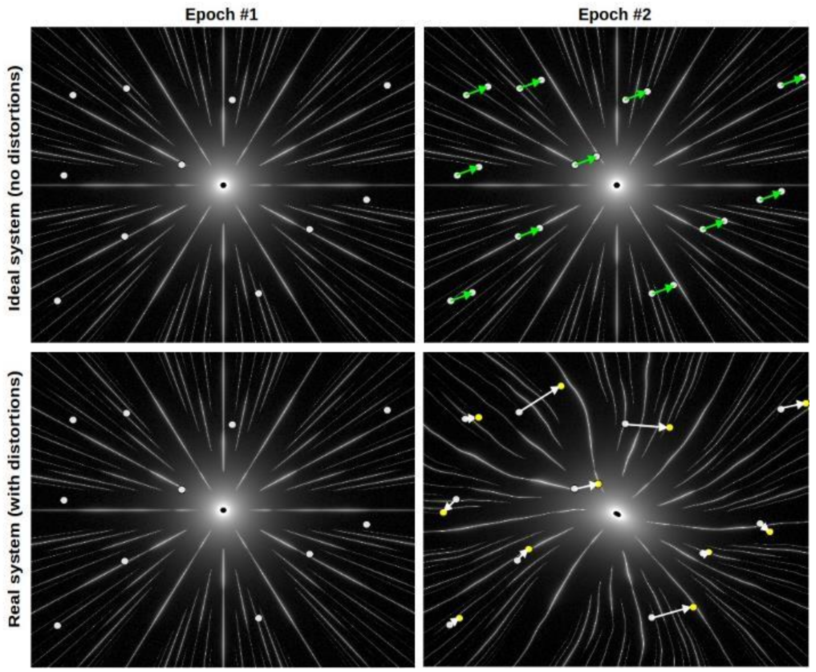


Fig.1. Optical distortions limits astrometric measurement of a star w/r to background. The Hubble distortion field amplified 1e6 times is represented on the lower right corner

Problem Description

a) Context: Analytical methods to extract the distortion map from the diffractive features are limited by noise in the images and the complexity of the signal.

b) SOA: Currently the astrometric accuracy state-of-the-art is in the order of 25uas for Hubble using scanning techniques and about 30uas for the Gaia missions

We need and accuracy of ~ 1 micro arcsecond (µas) to detect earth-analogs around sun-like stars

c) Relevance to NASA and JPL (Impact on current or future programs)

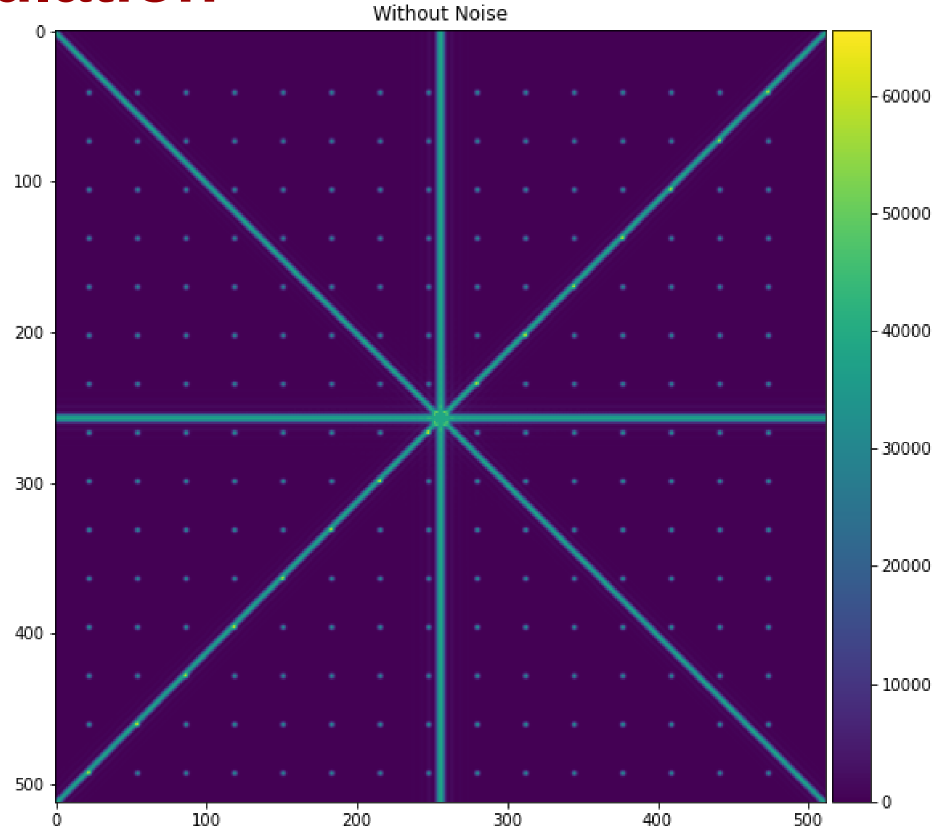
Enabling high-precision astrometry on future space mission would enable measuring exoplanet masses, thus constraining their habitability.

Methodology – Image Simulation

Goal: Explore the ability of Machine Learning to learn the shift in astrometric signal from images that have DP calibration features under both noiseless and noisy conditions

Image Simulation: We simulated images containing a source with diffraction spike (represented by the lines) along with a consistent grid of background sources (the grid of dots). Each image is than arbitrarily shifted in the x and y directions up to 1 pixel. We generated 10K sets of the following:

- One set of noiseless images (example on right)
- Two sets of Noisy images with readout, photon and flat field noise
- Simulated images were 512x512 and scaled to be 16 bit
- We also generated FFTs of the images



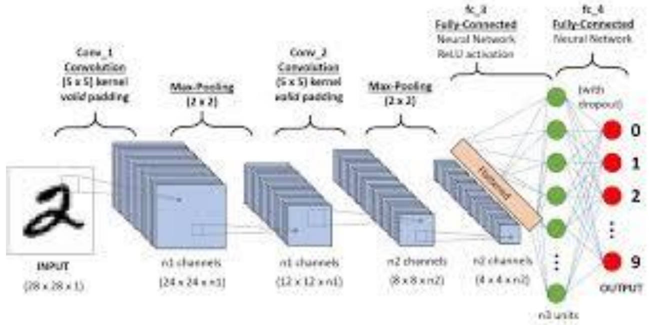
Methodology – Learning Algorithms

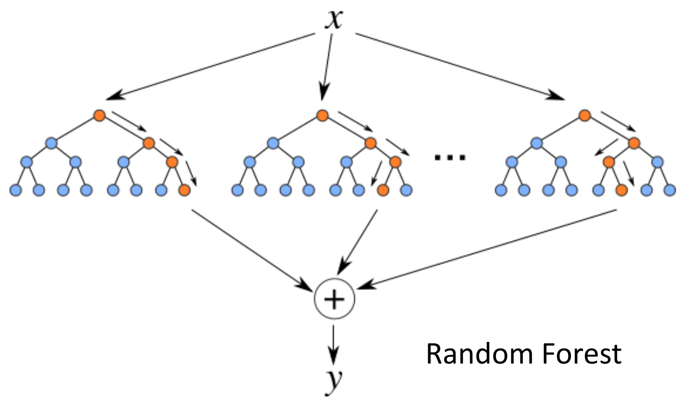
We selected the following ML regression methods for experimentation:

- Random Forest (RF)
- Multi-layer Perceptron (MLP)
- Deep Learning Convolutional Neural Network (DL)

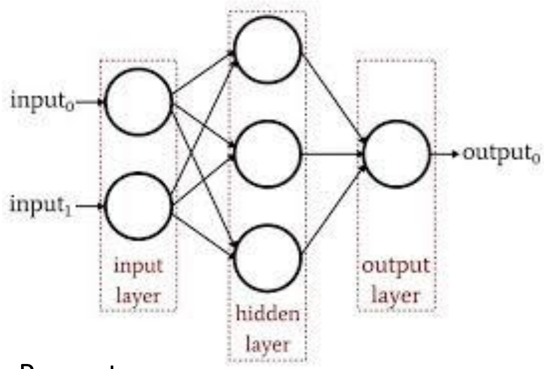
This represents a spectrum of well-known, robust ML regression methods.

Deep Learning





Random Forest



Multi-layer Perceptron

Methodology – Random Forest Regression

The random forest (RF) is a widely-used regression method, known for its robustness for noise and low variance models. The ”forest” is comprised of individual decision tree regressors, which are binary trees that recursively partition the training examples from root to leaf into a prediction is made that closely matches the target. The forest reports an average of individual tree predictions. Each tree is built using a sampling of the training data in order to ensure diversity among the trees.

We use the scikit-learn’ sRandom Forest regressor in Python, setting the number of trees in the forest to 100.

Most ML classifiers and regressors typically predict a single value. Our problem requires the ML algorithm to learn both x- and y-shifts. The RF regressor natively allows for this, but we utilized the MultiOutput wrapper provided by scikit-learn that learned each coordinate individually and combines the result. We tried both the native and Multi-Output wrapper in some preliminary results and noted similar performance, and outputs to continue to use the Multi-Output wrapper. Please note the deep learning methods described next use a similar type of wrapper.

We experimented with the following feature sets:

- row-wise flattened image
- col-wise flattened image
- row and column sums of the images
- row and column sums of FFT of the image (both real and imaginary parts)

Methodology – Deep Learning

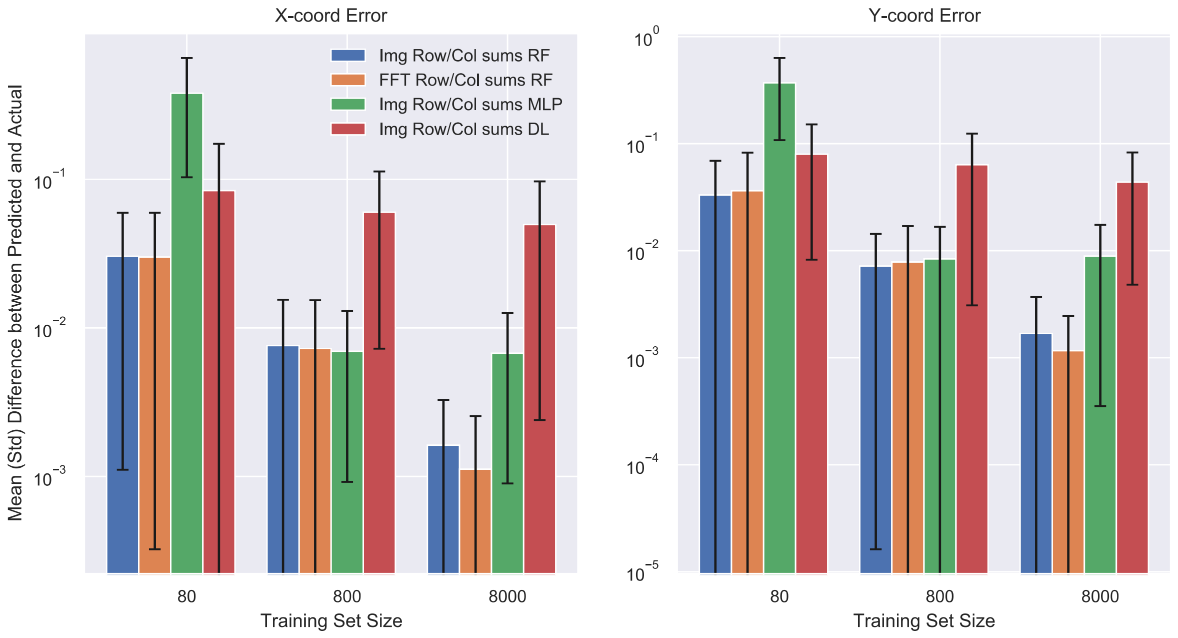
Deep learning models represent a class of tools that excel at capturing complex and potentially non-linear relationships in data. They are considered “deep” because many consecutive layers of artificial neurons are stacked on top of each other. Convolutional neural networks are a large class of deep models that are a good example of this concept: In convolutional neural networks, early layers in the network may capture lines and edges, middle layers may use combinations of lines to capture shapes, and later layers can use combinations shapes to identify objects.

MLP model
The Multilayer Perceptron (MLP) model we used was a fairly simplistic deep learning model that we trained using the image row and column sums as input. The model consisted of several hidden layers with dense connections and exponential linear unit (elu) as the activation function. Specifically, we used five hidden layers of size 512, 256, 128, 64, and 32. The output layer was simply 2 neurons that gave the predicted X and Y offset. During model training, we used the stochastic gradient descent optimizer, decreased learning rate as training progressed, and used mean average error as the loss function. All work was carried out using the TensorFlow framework

Convolutional model
The convolutional model we tested is a relatively heavy deep learning model and was trained using the raw images as input. Here, we used the EfficientNetB0 model as our backbone with one 64 neuron hidden layer appended to this backbone. Again, the output layer consisted of 2 neurons that predicted the X and Y offset. During training, we used the stochastic gradient descent optimizer, decreased learning rate as training progressed, and used mean average error as the loss function. Because this is a much more sophisticated model, we trained on an Nvidia Tesla P4 GPU. All work was carried out using the TensorFlow framework.

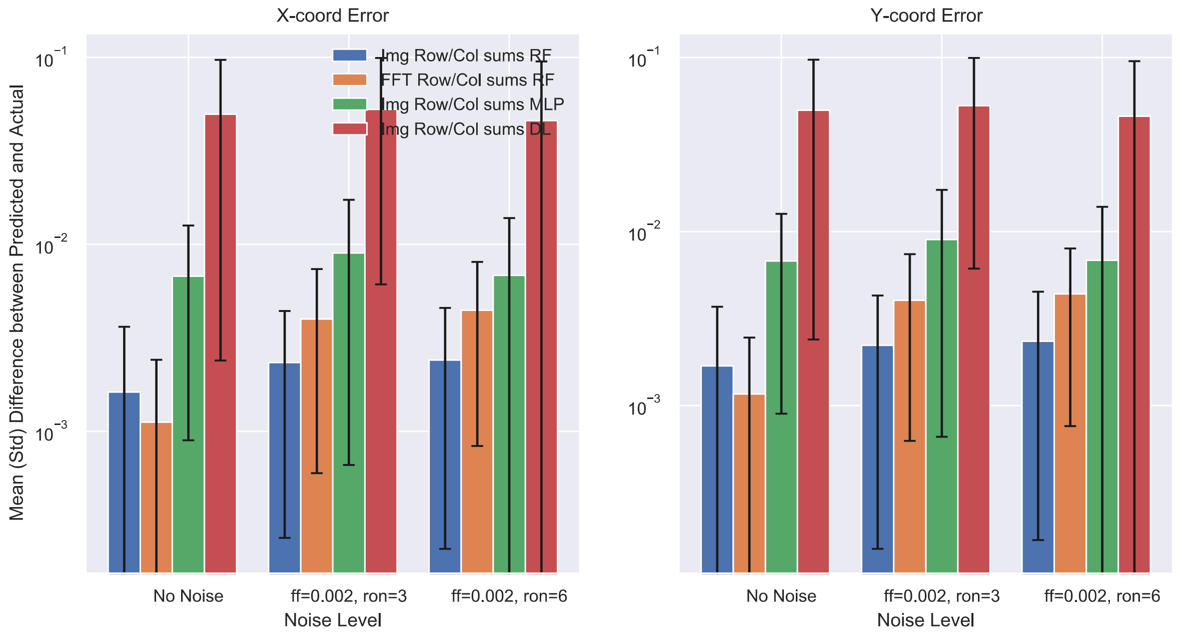
Results on Noise-free Images

- We compare two RF models (with row-col sums on the science image, and row-col sums on the FFT image), and two deep models (MLP, DL), where MLP was trained on image row-col sums and DL was input the science image
- We experimented with increasing training set sizes
- As expected, model performance improves as training data increases
- Without any noise, it appears both RF models outperform the DL models



Results on Noisy Images

- Using the largest training set size (8000 images), we compare performance on noise-less and two sets of noisy images (with flat field (ff) and read-noise constants (ron) shown below. Note that the right side results have 2x the noise of the center results.
- The deep methods (MLP and DL) seem to not be affected by noise, while the RF methods are affected.
- RF methods perform better, but under noise, the image features perform better than FFT features.



Best Performing Result under Highest Noise

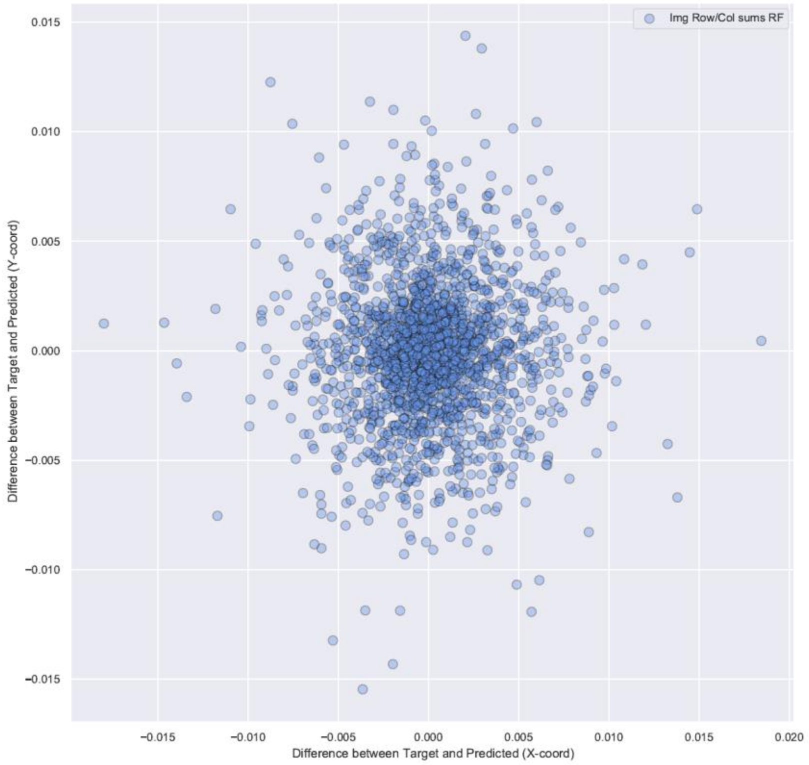
- Here are the x- and y-coordinate errors plotted for the best performing result (RF using row-col sums on the science image) under the highest noise.
- The RMS error of the best performing algorithm is

RMS x = 0.00329px

RMS y = 0.00319px

Calculated over 2000 images.

- Typical analytical centroiding for similar noise levels is in the order of 0.001px
- ML was able to get to the same order of magnitude than analytical algorithms.



Results

a) Accomplishments versus goals

Goal:

Measure astrometry vectors to better accuracy than using geometrical centroiding on the simulated images.

Results:

None of the machine learning algorithms could estimate the image shifts better than the analytical approach

BUT, Deep Learning (DL) showed robustness to noise.

a) Significance

We could not train DL to optimal levels because the time needed. However, there is indication that DL can improve from its current status and it can be resilient to noise.

a) Next steps

Continue this research with an Intern that is lined up with Div 39.

Publications and References

No publications have resulted from this work yet.