

# Correcting Keck/NIRC2's Systematic Errors Using Linear SVMs

Justin Kang

December 8, 2017

## Abstract

Keck/NIRC2 uses AO to produce the highest-resolution ground-based images and spectroscopy in the 1-5 micrometer range. It is one of the best instruments for planet discovery and characterization, with 44 papers published just this year that have used it. However, there are several systematic errors that reduce its effectiveness and usability. Examples of these are read errors mirrored across the image quadrants, and the noisier lower-left quadrant. The current method of dealing with this noise is using a three-point dither pattern to reduce the effects of the bad quadrant, and there is no real method of dealing with the mirrored errors as of yet.

We propose to instead create a model of these errors. By doing so, we can then attempt to eliminate the bad quadrant, allowing for more usable image space, and also improve the deepness of the images by removing instrumental noise in all quadrants. One possibility is training a linear support vector machine and using it to classify sources as either real or noise. Linear classifiers are compact, fast to train, fast to execute, and can be trained on large amounts of data, making them an ideal tool for this application. Furthermore the KOA database makes readily available large amounts of both training and testing data. Thus by successfully modeling and eliminating this noise, we can improve one of the best ground-based instruments for future use.

# Scientific Justification

## Direct Imaging

Direct imaging for planet detection, as the name suggests, involves directly taking images of a planetary system (or debris disk). From a series of images, we can estimate the planet's orbit, size, temperature, atmosphere, and other properties from its photometry, colors, and spectra. This method favors planets that can be resolved from the host star(s) - thus young and hot planets that are still radiating away energy from their creation (primordial thermal emission), widely separated from their host, and large (several times that of Jupiter). Unlike most other detection methods, direct imaging works best with face-on orbits rather than edge-on, as it can then accurately measure the entirety of the planet's orbit around the star. This gives direct imaging a unique parameter space in relation to the other detection methods - it finds large planets with large orbital radii, occupying the space beyond what current radial velocity methods detect. Thus the method complements well the other detection methods, such as radial velocity and transits.

## Keck/NIRC2

The second generation Near-Infrared Camera (NIRC2) at Keck Observatory uses the Keck Adaptive Optics system to produce the current highest-resolution ground-based images and spectroscopy in the 1-5 micrometer range. Its first major planet discovery was in Marois et al. (2008), where it was used to observe the first multiplanet system in HR 8799. Since then it has been widely used by the direct imaging community, with 44 papers published in just 2017 that used the instrument.

**However, there are several systematic errors that reduce NIRC2's effectiveness and usability.** The first example of these is the lower-left quadrant of the array; as noted by many that have used it (Bowler et al. (2014), Crossfield et al. (2017), Rodriguez et al. (2017)), this corner suffers from elevated noise levels. This noise presents itself in the form of vertical stripes - here we see groups of 8 vertical pixels with increased intensity, repeated in a stripe pattern across the entire quadrant. The current popular method of dealing with this quadrant is adopting a 3-point dither pattern (as suggested in the Keck observer's manual), which omits the lower left and central positions of the detector. Another method is to place the coronagraph far from the region, keeping the objects of interest outside the region. Both of these "solutions" avoid the use of the error-prone region, losing an entire quadrant of the detector. Thus, these aren't really useful or efficient methods - we need a better way of dealing with this error. There is another similar systematic error - across images we can see groups of eight horizontal pixels with increased intensity, mirrored across each of the quadrants. A consequence of detection in the near-infrared, this error is also commonly handled in a similar way - by not actually systematically dealing with the error itself, but raising the criteria for detections of real objects such that this error is below a threshold and ignored. Thus these solutions for NIRC2's systematic errors leave a lot to be desired, and call for better techniques to deal with them.

## Linear SVMs and the Sliding Window Detection

We now introduce a technique commonly used in the field of computer vision - the sliding window classification from Dalal and Triggs (2005), used commonly in pedestrian detection. This algorithm employs a sliding window to independently classify all image windows as being an object or non-object. Given heterogeneous training and testing data, we transform images to a histogram-of-gradients representation, train a linear classifier, and use this classifier to classify up to millions of sliding windows. Here the training data is initially a combination of positive training examples and a negative set free of the sample. The classifier is then trained, and the negative examples are searched for false positives. The classifier is then retrained using the initial set and the false positives to improve accuracy.

The benefits of using linear classifiers are that they are compact, fast to train, and fast to execute, allowing for quick verification of the results. The one we use here is a support vector machine - given a set of training examples, which are annotated as belonging to one of two categories (e.g. object or non-object), SVMs build a model that assigns new examples to one of the categories. The model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by the widest gap possible. New examples are then mapped to that same space, and are assigned a category based on which side of the gap they're on.

## Proposed Program

We propose to create a model of these errors by training a linear support vector machine and use it to classify windows of Keck/NIRC2 images as either real or noisy data. Using this classifier, our goal is to thus eliminate the elevated noise in the lower-left quadrant by subtracting the effects of the noise. The direct benefit of this program would be to both improve the image quality and reliability of NIRC2 images, allowing for better data for anyone that uses the instrument.

In terms of benefits for direct imaging, this could improve both the deepness and scale of the images. For deepness, we would more reliably be able to tell whether an object in that quadrant is real or just noise, reducing the magnitude and effects of noise on the data. For scale, we could then expand to fully using the lower-left quadrant of the instrument rather than just using three quadrants like the 3-point dither method recommends, allowing for more objects to be captured in the image. Overall, this would lead to an effective increase in the resolution of the instrument, improving its accuracy even more.

Depending on the results, this could have impacts for other sub-fields of astronomy. Depending on how easy it is to generate training data for this noise, SVMs could be used to classify error in any other sets of imaging data. Due to the portability of this technique and its origins as a pure object detector, this method could easily be used in any application where such an image manipulation technique could be useful. SVMs are also starting to become an "outdated" technique in the field of computer vision. Assuming that this method works reliably, we can then attempt to introduce newer methods from computer vision and machine learning, and apply these to the field of astronomy.



Figure 1: From Kraus et al. (2006), a very low mass star in the substellar boundary in Taurus. Here we see a 4-point dither pattern, which literature suggests has an improvement in the signal-to-noise over simpler dither patterns.

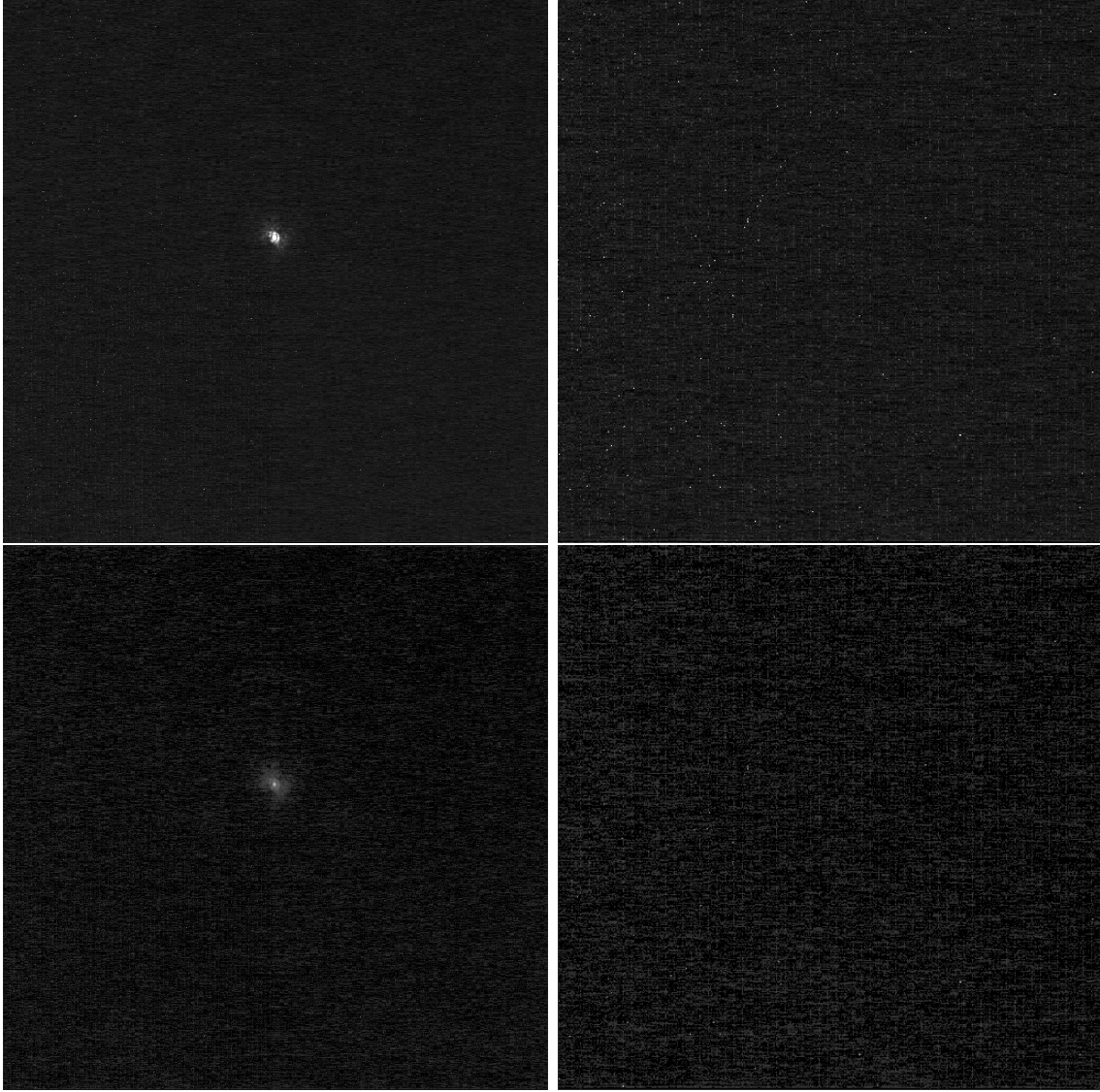


Figure 2: From Bryan et al. (2016), ROXs 12 and its planetary companion. The left column shows the full image from Keck/NIRC2, while the right column shows a zoom-in of the lower-left quadrant. The images on the bottom row show the fourth root of the intensities of the image, to better demonstrate the enhanced noise seen in that quadrant.

# Analysis Plan

The KOA database makes readily available large amounts of both training and testing data. We first must obtain training data of the errors in question. In computer vision this data is large, easily available, and well-annotated; unfortunately the field of astronomy is not so lucky. In order to generate this training data, we must look for examples where we see these errors, as demonstrated in Figure 2. Because these errors are not perfectly regular, one option of generating the training data - windows each containing one instance of an error, for example one vertical stripe error - is to manually go through a small subset of images and collecting each individual error. Attempts at automating this process can be made depending on the regularity of this error within images.

Once this training data is obtained, we first go through a training phase, training the classifier on these images annotated as errors, and samples of images known to not be error. Thus our SVM will be able to differentiate between these two windows through supervised learning. We then run this classifier on a small verification set. By doing so we can get an initial idea of the accuracy of our training, and gather false positives that result from the classifier. We then retrain the SVM based on these hard negatives to further improve it.

We then move onto our testing phase. We run our classifier on a set of completely new testing data - images that were not used for either training or verification. We can then determine accuracy by looking through each of the resulting images, seeing which windows were classified as errors, and noting both the true and false positive rates of the classifier. In order to do so, without a more regular error, we must manually go through each of the images and see whether errors exist, whether the classifier correctly identified them, and what else the classifier identified.

After identifying these errors with the classifier, we must remove them from the image. For now we propose a simple subtraction - once the windows in question are identified from the images, we can simply subtract the pixels that are relevant. In the case that there are objects in that area that amplify the intensity of these pixels, one method is to analyze the intensity properties of the noise itself. If it can be shown to be regular or match a certain pattern, then we can subtract this noise based on this assumed regularity. This would then better preserve any object that was hidden due to this noise in the quadrant.

The first image set this will be trained on is Bryan et al. (2016). The relative small size of this set allows for quick training and testing, allowing for easy predictions of the performance of this method. Depending on the success of this set, the second image set is that of Kraus et al. (2006). The advantage of this is the easily demonstratable 4-point dither method - after subtracting the errors on each of the images, we can check for recovery of each of the "real" objects that we expect in the lower-left quadrant based on what we know from other perspectives of the same image. These image sets can both be obtained immediately digitally through the KOA. Thus we can analyze the effectiveness of both the SVM in detecting errors, and the subtraction method in eliminating errors.

## Management Plan

The PI will carry out all aspects of this program, from beginning to completion. The relevant data can almost immediately be obtained from the KOA. Once obtained, around a day will be needed to go through all of the images, finding the relevant ones with the errors, and generating the testing data. Several days will be needed to code the general algorithm, tune it for the purposes of this program, and debug it if necessary. Several hours will be needed to train the actual SVM, depending on the number of actual training samples generated. A day or two will then be needed to analyze the training results, make adjustments to the program if necessary, then retrain the classifier. Finally, several days will be needed to both produce and analyze the results of the program. Once these results are obtained, the code can be made openly available on Github for anyone interested in using the software, or exploring the application.

## References

- Bowler, B. et al. 2015, ApJS, 216, 53  
Bryan, M et al. 2016, ApJ, 827, 16  
Crossfield, I. et al. 2017, ApJ, 153, 255  
Dalal, N & Triggs, B. 2005, CVPR, 1063-6919, 05  
Kraus, A et al. 2006, ApJ, 649, 306  
Marois, C. et al. 2008, Science, 322, 1348  
Rodriguez, J. et al. 2017, arXiv:1709.01957