Smart starting guesses from machine learning for phase retrieval

Scott W. Paine, James R. Fienup

Scott W. Paine, James R. Fienup, "Smart starting guesses from machine learning for phase retrieval," Proc. SPIE 10698, Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter Wave, 106985W (6 July 2018); doi: 10.1117/12.2307858

SPIE.

Event: SPIE Astronomical Telescopes + Instrumentation, 2018, Austin, Texas, United States

Smart starting guesses from machine learning for phase retrieval

Scott W. Paine and James R. Fienup

University of Rochester, Rochester, NY, USA

ABSTRACT

Image-based wavefront sensing uses a physical model to simulate a point-spread function (PSF) that attempts to match measured data. Nonlinear optimization is used to update parameters corresponding to the wavefront. If the starting guess for the wavefront is too far from the true solution, these nonlinear optimization techniques are unlikely to converge. We trained a convolutional neural network (CNN) based on Google's Inception v3 architecture[1] to predict Zernike coefficients from simulated images of PSFs with simulated noise added. These coefficients were used as starting guesses for nonlinear optimization techniques. We performed Monte Carlo analysis to compare these predicted coefficients to 30 random starting guesses for total root-mean-square (RMS) wavefront errors (WFE) ranging from 0.25 waves to 4.0 waves. We found that our CNN's predictions were more likely to converge than the random starting guesses for RMS WFE larger than 0.5 waves.

Keywords: Phase Retrieval, Wavefront Sensing, Machine Learning

1. INTRODUCTION

Image-based wavefront sensing is a method to retrieve optical wavefront given only the point-spread function (PSF), the image of a point source. Often, this method uses a physical model to simulate an image which attempts to match a measured PSF. This can be accomplished by gradient-based nonlinear optimization, which will update the wavefront to minimize the difference between simulated and measured images via reduction of a cost function, such as a bias-and-gain invariant normalized mean square error (NMSE) metric.[2] Often, it is useful to parameterize the wavefront by a basis set such as the Zernike polynomials to reduce the dimensionality of the search space and enforce physically appropriate smoothness in the wavefront. To update the wavefront, the gradient-based nonlinear optimization algorithm uses information about the gradient of the polynomial coefficients with respect to the cost function to search through parameter space. The gradient can be efficiently computed using algorithmic differentiation methods.[3] The optimizer will iterate using these gradient terms until either the change in the error metric or the norm of the gradient is sufficiently small.

These termination criteria can cause a gradient-based nonlinear optimization algorithm to stagnate in a local minimum, where the gradient is sufficiently small. In such local minimum, the simulated image is inconsistent with the measured PSF, with the exception of unlikely ambiguous solutions.[4] In order to prevent this stagnation, it is necessary to have additional information about the system such as defocus planes, or to have a good starting guess for the wavefront error. We refer to the distance of a starting guess that is likely to converge as the "capture range", and the failure to converge when outside of this distance as the "capture range problem". The capture range of phase retrieval problems can be extended by using phase diversity[5,6] or series of measurements from a translated subaperture.[7] It is also possible to use a series of random starting guesses in an attempt to randomly fall within the capture range.[8]

In the case of retrieving polynomial coefficients, random starting guesses becomes infeasible as both the number of coefficients and root-mean-square wavefront error (RMS WFE) increase. If it is assumed that one can select random starting guesses which have the exact same RMS WFE as the true wavefront, then the process of selecting random starting guesses is equivalent to choosing random points on the surface of a hypersphere whose dimensionality is given by the number of coefficients. In order to be within the capture range, it is necessary to choose a point within the hyperspherical cap such that all points within that cap are in the capture range. The ratio of the surface area of the hyperspherical cap (the set of points that are within the capture range) to the total surface area of the hypersphere (the set of all points we are randomly sampling) is proportional to $1/R^{n+1}$,

Space Telescopes and Instrumentation 2018: Optical, Infrared, and Millimeter Wave, edited by Makenzie Lystrup, Howard A. MacEwen, Giovanni G. Fazio, Proc. of SPIE Vol. 10698, 106985W
 · doi: 10.1117/12.2307858

where n is the dimensionality and R is the radius of our hypersphere.[9] The large values of R correspond to large RMS WFE, meaning it is unlikely to determine the wavefront for highly-aberrated PSFs from random starting guesses alone.

In order to generate starting guesses that are likely to fall within the capture range, we turn to machine learning. Machine learning has been used previously with some success for phase retrieval on the Hubble Space Telescope.[10] However, that attempt was limited by computational power to use a machine learning model known as a perceptron. Perceptrons like this flatten the input image array to a vector, and then transform this vector using a matrix multiplication to a "hidden" vector, which is transformed by matrix multiplication again to a final output vector corresponding to coefficient predictions. In the case of image data, single-pixel information is often not very informative. This has given rise to the convolutional neural network (CNN), a machine learning model that considers groups of pixels through learned convolutional kernels, which is often better than perceptrons for image-based tasks.[11] CNNs have been used recently in an attempt to solve the inverse problem for computational imaging tasks.[12] We intend to use a CNN to examine an input PSF and give an estimate of Zernike coefficients. These coefficients will then be used as a starting point in our gradient-based nonlinear optimization methods.

2. MODEL SELECTION AND TRAINING

To select a model, we looked at pre-existing implementations of CNNs, and chose Google's Inception v3 architecture, which has performed well in classification tasks.[1] We modified the architecture to perform regression analysis rather than classification analysis, which consisted of changing the final few layers to fully-connected layers like those of a perceptron. Our model[13] is seen in Figure 1, with different-colored blocks corresponding to different arithmetical operations. The branches between concatenation blocks in Figure 1 have convolutional kernels of different sizes, which in turn allows the model to consider larger and smaller features in the PSF.

In order to create the dataset to train our model on, we used our existing physical model to generate PSFs from randomly drawn Zernike coefficients. We assumed that the wavefront was the only unknown in our system, which means we had knowledge of sampling requirements and pupil apodization. In our simulations, our PSFs were Nyquist-sampled, and we used a uniformly-illuminated James Webb Space Telescope (JWST) aperture.[14] Our wavefront aberrations consisted of second- through fifth-order Zernike coefficients which span the entire aperture. We ignored global piston and did not predict tip or tilt terms, since these can be rapidly estimated using centroiding algorithms.[15,16]

To perform the training, we used minibatch training, which updates the learned values in a machine learning model based on a group of data, rather than the whole dataset or a single data point. This has been shown to be a computationally efficient way to train machine learning models without using an entire dataset.[17] During training of a machine learning model, the step size of these updates is controlled in part by a parameter known as the "learning rate".[18] A larger learning rate indicates more drastic updates to learned values, but at the cost of potentially moving over desirable values. A smaller learning rate is valuable for "fine-tuning" the model, but can result in the model getting stuck in an undesirable minimum for the learned values. The updates for learned values such as the convolutional kernels were done using the Adam gradient-based stochastic optimization algorithm, which has an adaptive learning rate that is intialized to a user-defined value, but changes during optimization according to gradient values.[19] The loss function we attempted to lower during training was the residual RMS difference between the prediction coefficients and the true coefficients used to generate the data.

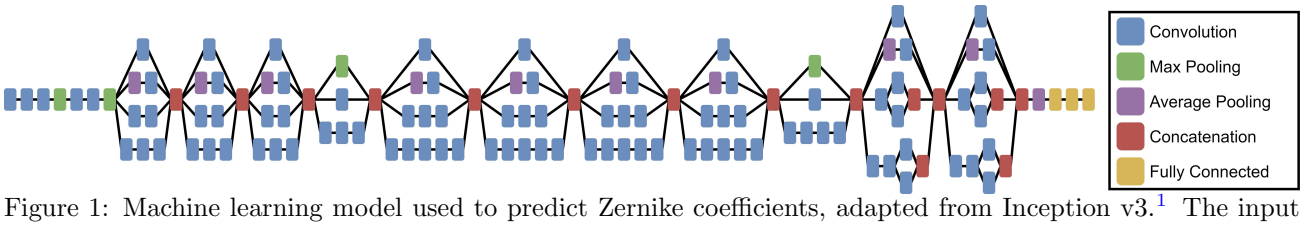


Figure 1: Machine learning model used to predict Zernike coefficients, adapted from Inception v3.[1] The input is fed to the furthest left convolutional block, and the predicted coefficients come from the furthest right fully connected block. Reproduced with permission.[13]

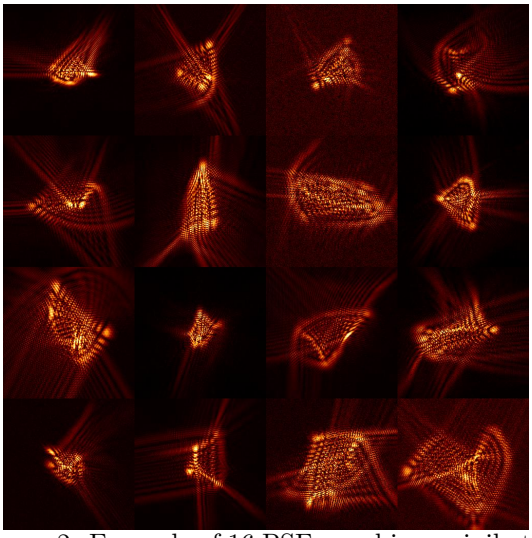


Figure 2: Example of 16 PSFs used in a minibatch to train our CNN. All PSFs have been square-rooted to show dim features and noise.

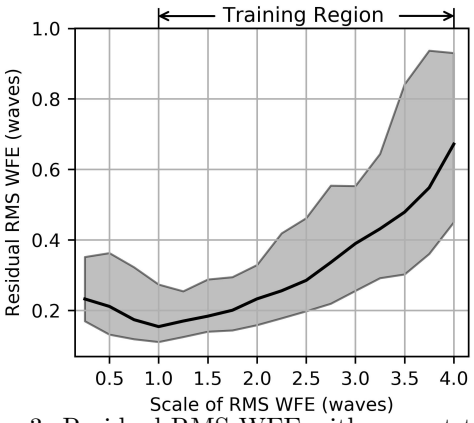


Figure 3: Residual RMS WFE with respect to total RMS WFE for wavefronts made with CNN-predicted Zernike coefficients compared to true wavefronts. The central black line represents the median values of the residual RMS WFE from 100 trials, and the shaded area represents the bounds of the 10^{th} and 90^{th} percentiles. Reproduced with permission.[13]

For our training, we would perform 40 updates, with each update based on a mini-batch of 16 PSFs. After these updates, we would perform a validation step, in which 160 new PSFs would be fed to the model and the average loss would be computed amongst this group. This validation stage inform us of overfitting in our CNN.[20] We refer to a group of 40 updates and the validation step as an "epoch" of training. We started training the model on PSFs with 2.3 RMS waves of total aberration for 5000 epochs, with an initial learning rate of 2×10^{-2}, which we halved after every 1000 epochs. We then expanded the possibilities of total RMS WFE to an amount between 1.0 and 4.0 RMS waves of aberration. We trained with these PSFs for 20,000 epochs, starting with a learning rate of 1×10^{-3} for the first 10,000 epochs and lowering the rate to 0.5×10^{-3} for the second 10,000 epochs. Finally, we included simulated noise in our PSFs that included Poisson noise and optionally included detector noise, background noise, and dead pixels. An example of one minibatch of these noisy input PSFs can be seen in Figure 2. The peak photons and any additional noise parameters for each PSF were chosen from a uniform random distribution, with low and high values given in Table 1. The intent of having many noise options was to make our CNN robust to a wide variety of noise. We trained on noisy PSFs for 50,000 epochs, starting with a learning rate of 2.5×10^{-3} and lowering every 10,000 epochs to 1.0×10^{-3}, 0.75×10^{-3}, 0.5×10^{-3}, and 0.3×10^{-3}, respectively. At the conclusion of this training, our model's average coefficient predictions over 160 PSFs had 0.373 waves of residual RMS WFE. We noted that the residual RMS WFE grew with the total RMS WFE inside the training region, as shown in Figure 3.

3. MONTE CARLO ANALYSIS

In order to determine the effectiveness of our machine learning model compared to random starting guesses, we performed a Monte Carlo analysis. We started by simulating PSFs from wavefronts consisting only of the

Table 1: Table of the bounds on peak photons and noise parameters added to PSFs for training and Monte Carlo simulation.

	Low Value	**High Value**
Peak photons (photons)	4000	15000
Read noise (e^-)	10	100
Background noise (photons)	0.0	4.0
Fraction of bad pixels (%)	0.1	1.0

Zernike polynomials whose coefficients were learned. We added noise in the same fashion as done in section 2. We simulated Nyquist-sampled PSFs with total RMS WFE varying from 0.25 waves to 4.0 waves, in steps of 0.25 waves. For each step of total RMS WFE, we simulated 250 PSFs. Initially, the size of our pupil plane and image plane was 256×256 pixels. However, we found that the high amounts of RMS WFE caused energy to fall outside of our image plane window. To fix this, we increased the size of our image plane to 512×512 pixels, which is reasonable given that the smallest detectors in the JWST are 1024×1024 pixels.[21,22] We also doubled the sampling in our pupil plane, which prevented aliasing in our simulated data. We fed a 256×256 pixel crop of the PSF to our CNN, so that we did not need to retrain. We then performed nonlinear optimization with a limited-memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm[23] using the 512×512 pixel PSFs. For each PSF, we performed nonlinear optimization with 30 different random starting guesses, keeping track of the solutions with the lowest residual RMS WFE and the lowest gain and bias invariant NMSE metric values. We then performed nonlinear optimization with the predicted coefficients from our CNN as a starting guess.

The results of our Monte Carlo analysis are summarized in Figures 4 and 5. In Figure 4, the dashed line represents the median values of the residual RMS WFE for the random starting guess of the 30 with the lowest NMSE, while the dotted line indicates the median values of the residual RMS WFE for the random starting guess with the lowest residual RMS WFE. In a real-world situation, we would not have access to the true wavefront, so we picked the random starting guess that gives the best NMSE value. However, even if we were able to pick the random starting guess with the best residual RMS WFE, we see that the residual RMS WFE for wavefronts with the CNN-predicted coefficient starting guesses outperform random starting guesses by several orders of magnitude for all total RMS WFE values greater than 0.5 waves. We also see that the median value of the residual RMS WFE for wavefronts with the predicted coefficient starting guesses is below $1/140$ waves (or $1/10$ of the Marechal criterion), indicated by the horizontal red dashed line in Figure 4. This indicates that although our machine learning model was not trained on PSFs generated from wavefronts below 1 waves RMS, it can still perform accurate predictions at these scales. By comparison, the median RMS WFE for wavefronts with random starting guesses was below this cutoff only for total RMS WFE below 0.5 waves. Figure 5 shows the total

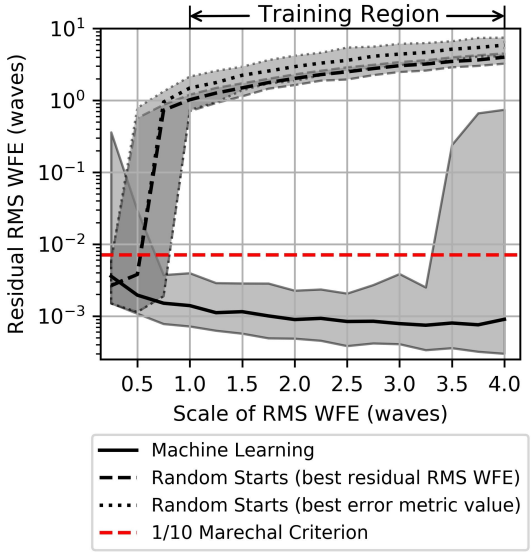


Figure 4: Residual RMS WFE values for optimizations based on random starting guesses and the CNN's predictions. The shaded area represents the bounds of the 10^{th} and 90^{th} percentiles of the residual RMS WFE, with the central black line representing the median values. The dashed red line indicates $1/140$ waves, or $1/10$ of the Marechal criterion. Reproduced with permission.[13]

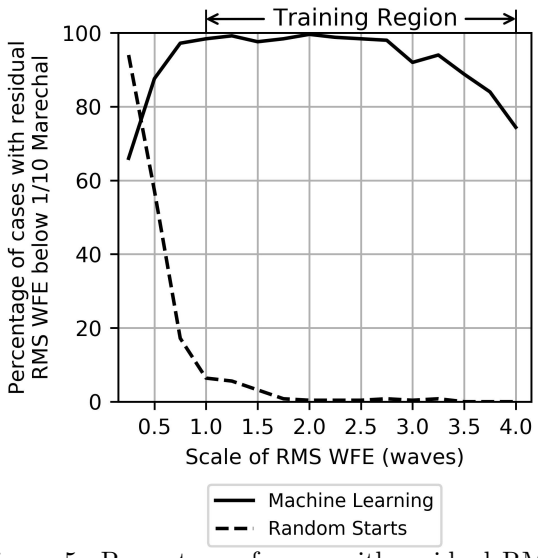
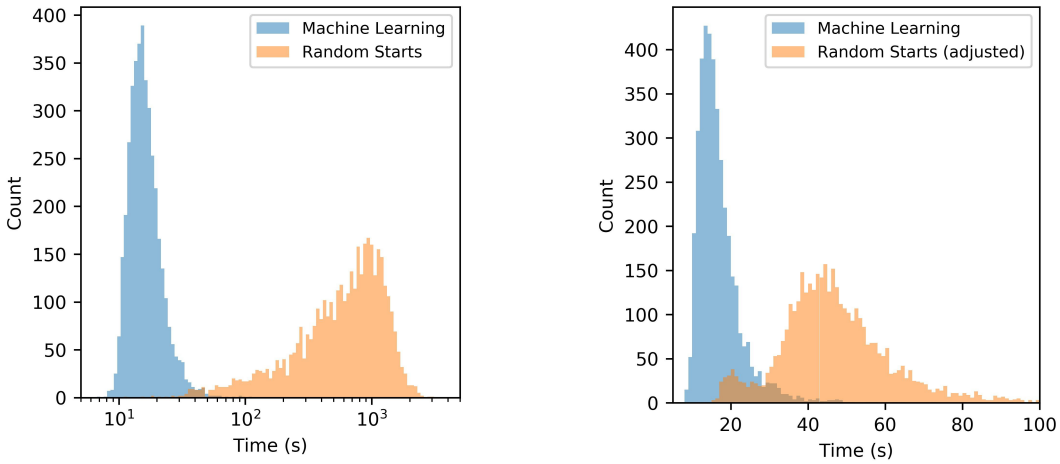


Figure 5: Percentage of cases with residual RMS WFE below $1/10$ of the Marechal criterion when using random starting points and the CNN's predictions. For the random cases, both the best residual RMS WFE solutions and the best error metric value solutions had the same percentage of cases below the threshold. Reproduced with permission.[13]



(a) (b)

Figure 6: Histogram of total time to best solution for CNN-predicted coefficients and random starting guesses with best error metric. In (a), the time of all previous random starting guesses are included in the total time and are plotted on a logarithmic scale, while in (b) the total time to solution was measured only for that random starting guess and is plotted on a linear scale.

percentage of cases whose final residual RMS WFE fell below $1/140$ waves for both all random starting guesses and CNN-predicted coefficient starting guesses. We can see that the random starting guesses outperformed the CNN predictions only for wavefronts with 0.25 waves of RMS WFE, for which the CNN was not trained. Additionally, the likelihood of convergence below $1/140$ waves for CNN-predicted starting guesses never fell below 60%.

We also found an improvement in total time to solution using the CNN-predicted coefficients. Figure 6a shows that, on average, the time to convergence for the CNN-predicted coefficient starting guesses were almost two orders of magnitude faster than the random starting guesses. This is due in part to the random starting guesses being run in serial, and only recording the time to the best solution. This means if the best answer from the random starting guesses occured after several other starting guesses, the time for all previous guesses was included in the time to solution. In order to examine the time to solution in a parallel situation, we recorded the time to solution for the single best starting guess. We see this adjusted histogram in Figure 6b, which shows that the CNN-predicted coefficients still converge to a solution faster than the best random starting guess. It is worth noting that even with this speed improvement, using CNN-predicted coefficients and optimization was not a real-time process, and takes on average around 16 seconds on a desktop CPU to both predict coefficients and optimize to convergence. We found that it only took approximately 0.20 seconds for the CNN to predict coefficients, so the majority of the time to solution is in the nonlinear optimization step.

4. CONCLUSIONS

We have demonstrated the capability to train a CNN to predict Zernike coefficients based on a single, noisy, centered PSF. Normally, defocus diversity is used to provide robustness in phase retrieval, but we present results here for a single PSF. We have shown that the predictions from our trained CNN differ from the true wavefronts on average by 0.37 waves RMS, but this value increases monotonically with the total RMS WFE of the true wavefront, as shown in Figure 3. These estimated coefficients were shown to be useful as smart starting guesses for nonlinear optimization, and were likely to converge within $1/140$ waves RMS of the true wavefront. We found during this analysis that optimization requires the majority of the energy to be present, but the CNN can be trained on a cropped subset the omits some of the energy in the PSF. We found that these predicted coefficients were more likely to converge than 30 random starting guesses if the total RMS WFE of the true wavefront was greater than 0.25 waves, which will be useful during deployment and alignment of large space telescopes such as the JWST. Such a system may also be useful for attempting to correct atmospheric phase errors of earth-based

telescopes. However, the system is not real-time using a desktop CPU. We found that it took, on average, 0.2 seconds for the CNN to predict coefficients, and 16 seconds for nonlinear optimization to converge. Even if we were to improve the performance of the CNN, the current bottleneck is our nonlinear optimization procedure.

We note that our current CNN has been trained to predict second- through fifth-order Zernike coefficients, and does not attempt to predict tip or tilt. It would be useful to include small amounts of tip or tilt to insure that the CNN can predict, even in situations where centroiding algorithms may not have a perfect answer. Additionally, all of our analysis was model-matched, so future examination should consider wavefronts which include higher-order contributions that cannot be described by the predicted coefficients, such as mid-spatial frequencies. The simulated noise used here is a useful first step, but it would be ideal to collect a true dataset on real detectors and train the CNN on this data, which will have noise terms that were not simulated in our simple noise model.

It would also be beneficial to train the CNN to detect subaperture wavefront errors, which are common in segmented systems like the JWST. To improve predictions on PSFs generated from larger amounts of RMS WFE, we could consider a loss function that is weighted by the total RMS WFE. This would give increased "importance" to reducing the residual RMS WFE of predictions with large amounts of initial RMS WFE. We also could improve the CNN by predicting terms unrelated to wavefront, such as the width of a Gaussian blur kernel as an estimate to vibration, or amplitude apodization.

Our training method was very simple, so future work to improve training could include smarter initialization of weights in our CNN. We used random numbers, but alternate initialization schemes using the data itself could lead to faster convergence and shorter training time.[24] Additionally, we could determine if it is possible to achieve similar performance with a smaller CNN. The Inception v3 CNN has over 23 million learned values, which increases training time and the time to perform predictions.

ACKNOWLEDGMENTS

Funding provided by NASA Goddard Space Flight Center (GSFC) (grant NNX17AH93A).

REFERENCES

[1] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z., "Rethinking the inception architecture for computer vision," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* , 2818–2826 (2016).

[2] Thurman, S. T. and Fienup, J. R., "Phase retrieval with signal bias," *J. Opt. Soc. Am. A* **26**, 1008–1014 (Apr. 2009).

[3] Jurling, A. S. and Fienup, J. R., "Applications of algorithmic differentiation to phase retrieval algorithms," *J. Opt. Soc. Am. A* **31**, 1348–1359 (July 2014).

[4] Bruck, Y. M. and Sodin, L. G., "On the ambiguity of the image reconstruction problem," *Opt. Commun.* **30**(3), 304–308 (1979).

[5] Gonsalves, R. A., "Phase retrieval and diversity in adaptive optics," *Opt. Eng.* **21**(5), 829–832 (1982).

[6] Paxman, R. G., Schulz, T. J., and Fienup, J. R., "Joint estimation of object and aberrations by using phase diversity," *J. Opt. Soc. Am. A* **9**, 1072–1085 (July 1992).

[7] Brady, G. R., Guizar-Sicairos, M., and Fienup, J. R., "Optical wavefront measurement using phase retrieval with transverse translation diversity," *Opt. Express* **17**, 624–639 (Jan. 2009).

[8] Moore, D. and Fienup, J. R., "Extending the capture range of phase retrieval through random starting parameters," *Frontiers in Optics 2014* , FTu2C.2, Optical Society of America (2014).

[9] Li, S., "Concise formulas for the area and volume of a hyperspherical cap," *Asian J. Math. Stat.* **4**(1), 66–70 (2011).

[10] Barrett, T. K. and Sandler, D. G., "Artificial neural network for the determination of Hubble Space Telescope aberration from stellar images," *Appl. Opt.* **32**, 1720–1727 (Apr. 1993).

[11] Driss, S. B., Soua, M., Kachouri, R., and Akil, M., "A comparison study between MLP and convolutional neural network models for character recognition," *Proc. SPIE* **10223**, 10223–10223–11 (2017).

[12] Sinha, A., Lee, J., Li, S., and Barbastathis, G., "Lensless computational imaging through deep learning," *Optica* **4**, 1117–1125 (Sept. 2017).

[13] Paine, S. W. and Fienup, J. R., "Machine learning for improved image-based wavefront sensing," *Opt. Lett.* **43**, 1235–1238 (Mar 2018).

[14] Cox, C. and Hodge, P., "Point-spread function modeling for the James Webb Space Telescope," *Proc. SPIE* **6265**, 6265–6265–6 (2006).

[15] Delabie, T., Schutter, J. D., and Vandenbussche, B., "An accurate and efficient gaussian fit centroiding algorithm for star trackers," *J. of Astronaut. Sci.* **61**, 60–84 (Mar. 2014).

[16] Thomas, S., "Optimized centroid computing in a Shack-Hartmann sensor," *Proc. SPIE* **5490**, 5490–5490–9 (2004).

[17] Li, M., Zhang, T., Chen, Y., and Smola, A. J., "Efficient mini-batch training for stochastic optimization," *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 661–670, ACM, New York, NY, USA (2014).

[18] Jacobs, R. A., "Increased rates of convergence through learning rate adaptation," *Neural Networks* **1**(4), 295–307 (1988).

[19] Kingma, D. and Ba, J., "Adam: A method for stochastic optimization," *International Conference for Learning Representations* (2014).

[20] Kuhn, M. and Johnson, K., [*Applied Predictive Modeling*], Springer, New York, 1 ed. (2013).

[21] Rieke, G. H., "Infrared detector arrays for astronomy," *Annu. Rev. Astron. Astrophys.* **45**(1), 77–115 (2007).

[22] Rieke, G. H., Ressler, M. E., Morrison, J. E., Bergeron, L., Bouchet, P., García-Marín, M., Greene, T. P., Regan, M. W., Sukhatme, K. G., and Walker, H., "The mid-infrared instrument for the James Webb Space Telescope, VII: The MIRI detectors," *Publ. Astron. Soc. Pac.* **127**, 665 (July 2015).

[23] Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C., "A limited memory algorithm for bound constrained optimization," *SIAM J. Sci. Comput.* **16**, 1190–1208 (Sept. 1995).

[24] Koturwar, S. and Merchant, S., "Weight initialization of deep neural networks(dnns) using data statistics," *CoRR* **abs/1710.10570** (2017).