

Speckle Cn2 Profiler: Improving Satellite Communications with Machine Learning

Simone Ciarella¹✉, Luisa Orozco¹, Victor Azizi¹, Marguerite Arvis², and Rudolf Saathof²

¹ Netherlands eScience Center, Amsterdam, The Netherlands ² Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands ✉ Corresponding author

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Open Journals](#) ↗

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Optical satellite communications is a growing research field in which -using lasers- signals can be sent from the ground to satellites, from satellites to satellites and then back to the ground. The main advantage of using laser communication over radio waves is increased bandwidth that enables the transfer of more data in less time. However, one of the challenges for this protocol is the turbulence in the atmosphere that perturbs such transmission. The reduction of the quality of signal communication can be calculated and then compensated, but this requires a knowledge of the turbulence strength. A common way to model the turbulence is to use the refractive index structure constant, C_n^2 , which is a measure of the strength of the turbulence. Its profile can be used to estimate the effect of the turbulence on the signal and then apply a correction. To measure C_n^2 , there are several alternative instruments, each one with its own limitations. One possibility is to use speckle-based observation, which consists on looking at the twinkling of the stars and use their pattern to infer the turbulence profile. This is a non-intrusive method that can be used in real time, but it requires a deep understanding of the turbulence and the observed speckle patterns, which are highly influenced by the turbulence profile. The connection between speckle observation, and turbulence (C_n^2) is not clearly understood, so an analytical theory does not exist. Here we present speckcn2, a Python package that uses machine learning to provide a numerical reconstruction of the turbulence profile from a speckle pattern (Ciarella, 2024).

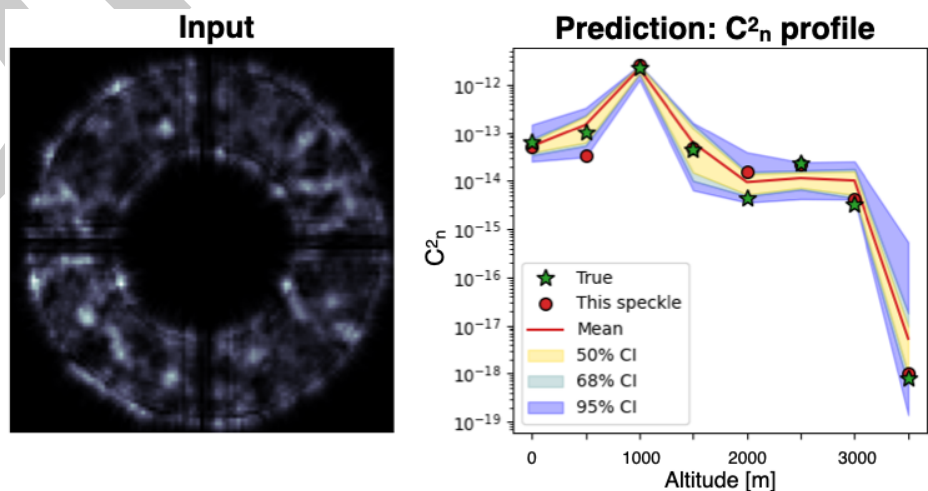


Figure 1: Example of speckcn2 pipeline: speckle pattern as input to output a prediction of the turbulence profile (J).

Statement of need

The turbulence in the atmosphere is a well-known phenomenon that affects the quality of optical communication. One way to compensate for this effect is to estimate the turbulence strength and then apply a correction to the signal. However, there are not many instruments that can provide this information. speckcn2 is a Python package that enables the use of machine learning to estimate the turbulence and reconstruct its profile. Using deep learning to compensate for atmospheric turbulence is not a new idea, and it has been already explored in the context of temporal mitigation using videos (Zhang et al., 2024) and to compensate static image degradation (Yasarla & Patel, 2021). However, these tools mainly provide a way to mitigate the effect of turbulence, while speckcn2 aims to provide a numerical reconstruction of the turbulence profile that can be used to: a) understand the turbulence and b) improve the communication system by inserting the turbulence profile in the communication model.

The approach of speckcn2 is based on the observation of speckle patterns, which are the result of the interference of light waves that have been perturbed by the atmosphere. Using PyTorch framework (Ansel et al., 2024), it is possible to build, train and deploy deep learning models that predict the turbulence. speckcn2 was created mainly for research in aerospace engineering in which one would aim at inserting a real-time estimation of the turbulence to a model, but it can also be useful in other fields. The package is designed to be simple to use and flexible enough to handle different image regression ML task. It works equally well with synthetic data from simulations and real data from experiments, making it versatile for different research needs. speckcn2 is extendable to other research fields. By combining techniques like equivariance and ensemble learning, it offers a strong and reliable tool for turning images into regression models, opening doors for many innovative applications.

Key features

Instrument specialization

When estimating the turbulence features, it is of fundamental importance to not mix the instrumental noise with the real effects that are being measured. A fundamental aspect of speckcn2 is the possibility to train models with different noise profiles, representing the noise of different instruments and modeling different detectors, whether real or simulated. By adapting the `apply_noise` function, users can model any type of effect related to their research and instruments. The current API provides a series of parameters that can be tuned to simulate the noise of different instruments, such as the signal-to-noise ratio, the detector gain, and the obscuration.

Equivariant model

To take advantage of the symmetry in the input data, speckcn2 uses a concept called equivariance (Cohen & Welling, 2016). This means that the model can learn the same features independently of the input data orientation. This is especially helpful for turbulence reconstruction, where the direction of the speckle pattern is not relevant.

speckcn2 supports two types of equivariance: weak and strong. Weak equivariance is achieved by randomly rotating the input data, which can then be used with any model from torchvision (TorchVision maintainers and contributors, 2016), including fine-tuning ResNets (He et al., 2015).

Strong equivariance is achieved by using the equivariant sparse convolutional neural network (escnn) (Cesa et al., 2021; Weiler & Cesa, 2021). These networks are more powerful for this type of problem but are harder to train.

Ensemble learning

speckcn2 can also use ensemble learning by averaging the predictions from multiple input images. This means that the prediction of each model requires a set of multiple input images. This is only useful if the input images change faster than the output. Since this is not the case for laser communications, this feature is optional and can be turned off.

Software implementation

speckcn2 is implemented in Python and uses PyTorch (Ansel et al., 2024) for its machine learning tasks. This allows the user to take advantage of GPU acceleration, making computations faster and more efficient.

The package is published to PyPI and is easy to install via pip, and it works on both Linux and MacOS. It has a simple and user-friendly API that lets users quickly build, train, and evaluate models. Whether you are a beginner or an experienced user, speckcn2 is designed to be accessible and flexible.

For new users, the documentation and examples provide a great starting point, helping them get up to speed quickly. Experienced users will appreciate the flexibility of the package, which allows for customization to meet specific research needs.

Acknowledgements

The authors would like to acknowledge the Netherlands eScience Center for the funding provided under grant number NLESC.SSIML.2022c.021.

References

- Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., ... Chintala, S. (2024). PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. <https://doi.org/10.1145/3620665.3640366>
- Cesa, G., Lang, L., & Weiler, M. (2021, October 6). *A Program to Build $E(N)$ -Equivariant Steerable CNNs*. International Conference on Learning Representations. <https://openreview.net/forum?id=WE4qe9xlnQw>
- Ciarella, S. (2024). *Speckle to Cn2* (Version 1.0.0). Zenodo. <https://doi.org/10.5281/zenodo.14178688>
- Cohen, T., & Welling, M. (2016). Group Equivariant Convolutional Networks. In M. F. Balcan & K. Q. Weinberger (Eds.), *Proceedings of The 33rd International Conference on Machine Learning* (Vol. 48, pp. 2990–2999). PMLR. <https://proceedings.mlr.press/v48/cohen16.html>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep residual learning for image recognition*. <https://doi.org/10.48550/arXiv.1512.03385>
- TorchVision maintainers and contributors. (2016). *TorchVision: PyTorch's Computer Vision library*. <https://github.com/pytorch/vision>
- Weiler, M., & Cesa, G. (2021, April 6). *General $E(2)$ -Equivariant Steerable CNNs*. <https://doi.org/10.48550/arXiv.1911.08251>

- 113 Yasarla, R., & Patel, V. M. (2021). Learning to restore images degraded by atmospheric
114 turbulence using uncertainty. *2021 IEEE International Conference on Image Processing*
115 *(ICIP)*, 1694–1698. <https://doi.org/10.1109/ICIP42928.2021.9506614>
- 116 Zhang, X., Chimitt, N., Chi, Y., Mao, Z., & Chan, S. H. (2024). Spatio-temporal turbulence
117 mitigation: A translational perspective. *Proceedings of the IEEE/CVF Conference on*
118 *Computer Vision and Pattern Recognition (CVPR)*. [https://doi.org/10.1109/CVPR52733.](https://doi.org/10.1109/CVPR52733.2024.00279)
119 [2024.00279](https://doi.org/10.1109/CVPR52733.2024.00279)

DRAFT