Determining Multi-frame Blind Deconvolution Resolvability using Deep Learning

Trent Kyono, Jacob Lucas, Michael Werth
The Boeing Company
Justin Fletcher
Odyssey Systems Consulting
Ian McQuaid
AFRL

Abstract

Astronomical images collected by ground-based telescopes suffer from degradation and perturbations attributed to atmospheric turbulence. The Multi-Frame Blind Deconvolution (MFBD) algorithms that can extract well-resolved images from these degraded data frames are computationally expensive, requiring supercomputing infrastructure for relatively fast performance (on the order of hours to resolve a LEO pass) and currently can't be done in real-time. Because of this, optimal collection parameters cannot be adjusted for maximizing the likelihood of producing a resolved image with MFBD. In this paper we present a neural network that predicts whether an MFBD algorithm will be able to resolve a degraded image (or sequence of images) in real-time, and present experiments conducted on actual data collected at the Maui Space Surveillance Site.

1. Introduction

Multi-Frame Blind Deconvolution (MFBD) algorithms are a class of maximum-likelihood estimation techniques for the problem of forming object estimates from turbulence-degraded images when the point-spread functions are unknown [19]. They are computationally expensive, and typically require supercomputing infrastructure for expedited performance. Even with such computational power this task is not possible in real-time, and it takes on the order of hours to resolve a single low-earth orbit (LEO) pass. Due to this computational latency, real-time feedback (regarding MFBD likelihood) is not provided to telescope operators who could use this to optimize their collection parameters.

Due to the recent superiority of convolutional neural networks (CNNs) on a majority of computer vision tasks, we investigate the application of CNNs for predicting whether or not an atmospherically perturbed image will be resolvable by an MFBD algorithm. We refer to this binary classification task as *resolvability* throughout this paper. We first select a CNN from a pool of ImageNet pre-trained models by its performance in terms of area under the receiver operating characteristic curve (AUROC) for the task of *resolvability* given a single image frame. We then train a secondary network to ingest the predictions of sequential frames to provide a time-series prediction of *resolvability*.

We present experiments on actual sensor data collected at the Maui Space Surveillance Site (MSSS) using the Space Observation Control Kit (SpOCK) [15]. The primary contributions of our paper provides a proof-of-concept investigation into providing an approximation of a real-time metric for *resolvability*. This provides several immediate benefits. The first is the provision of real-time feedback for telescope operators with which they can use to tune collection parameters (camera gains, etc.) to maximize the likelihood of

collecting a *resolvable* pass. The second is our provided prediction can be used as a score to prioritize collections for a downstream MFBD algorithm, such that the most likely to be *resolved* passes are processed first. Lastly, our method can be used as a filter to screen out collections that will likely never *resolve*, and therefore can be used as a "garbage" cleaner for long term space savings.

We provide a brief discussion of related works in Section 2. Then, we formalize our problem and approach in Section 3. We provide a brief discussion of our dataset used, as well as our training architecture, hyperparameters, and regime used along with our experimental results in Section 4. Lastly, we conclude with a few brief remarks in Section 5.

2. Related Works

Convolutional neural networks (CNNs) are one of the most widely accepted methods in computer vision and are status quo for image classification, segmentation, and detection tasks. With the recent success of these algorithms in imaging competitions, such as Kaggle, ImageNet, etc., coupled with the advancements in deep learning libraries, GPU hardware acceleration and data availability, these methods have received heightened interest in the astronomical and space situational awareness (SSA) domain. These recent advancements in image processing with CNNs motivate our application of a customized neural network to classify MFBD *resolvability*.

Several related works have explored the applications of deep learning to astronomy. For noise reduction, [14] recently presented a proof-of-concept neural network for denoising the Bispectrum for astronomical image recovery on synthetic data. For classification, [10, 13, 4] investigated the application of object classification using neural networks on photometric light curves and showed promising results. Using Generative Adversarial Networks (GANs), [18] recovered features from artificially degraded images with worse seeing and higher noise than the original with a performance that far exceeded simple deconvolution. Additionally, [5] used a GAN to generate more realistic images of galaxies than existing state of the art. [8] used machine learning to automatically segment and label galaxies in astronomical images. [16] showed promising results using an autoencoder for real-time MFBD of solar images.

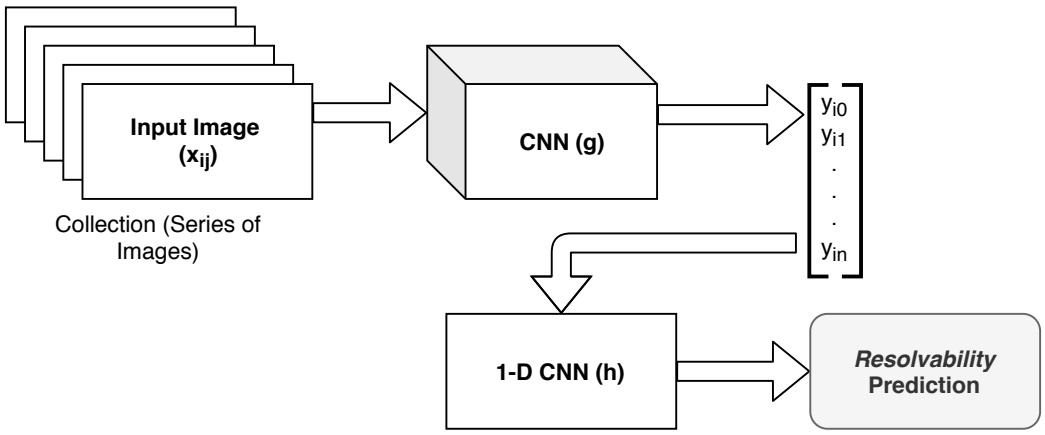

Figure 1: Schematic of our implementation. A series of input images are fed through a CNN to extract an overall prediction of *resolvability*. Each prediction is then fed into a secondary network to make a time-series prediction to provide an overall *resolvability* prediction.

3. Formalization

Let X be a set of perturbed (raw) astronomical images corresponding to a collection, and let Y be the collection's expert annotated/rated *resolvability*, i.e., $Y = \{0, 1\}$, where 0 corresponds to *not resolvable*, and 1 corresponds to *resolvable*. We refer to each collection $x_i \in X$ as a pass containing n sequential images. Given a pass $x_i \in X$, our primary design goal is to train a classifier $f : X \rightarrow Y$, which takes as input a collection $x_i \in X$ and provides a *resolvability* prediction, $y_i \in Y$. This can be further decomposed into a two networks. In the first network, we train a single image classifier $g : X_j \rightarrow Y_j$ that takes as input a single image, $x_{ij} \in X_j$, for the j^{th} image belonging to a pass x_i and makes a prediction $y_{ij} \in Y_j$ for image x_{ij}. In the second network, we train a time-series classifier $h : X \rightarrow Y$ that takes as input the predictions of g on $x_i \in X$, such that $g(x_i) \in X$, and provides a prediction for *resolvability*. All networks (f, g, and h) output either 1 or 0 when the image (or series) is either *resolvable* or *not-resolvable*, respectively. A summary schematic for this is shown in Fig. 1.

4. Experiments

This section briefly covers our dataset used in this work, our experimental settings (training architecture and regimes), and our experimental results.

4.1. Datasets

We perform experiments on two privately collected datasets, which we will refer to as Sensor A and Sensor B. For anonymity of our data, we will not specify the original image dimensions, but for image training and inference the images were resized to 128 x 128 pixels (for Sensor A) and 256 x 256 pixels (for Sensor B). Our two datasets contained images that were previously labeled by a trained analyst as either *resolved* or not. For each sensor, we examined satellite passes collected over the past 5 years. A majority of passes contained greater than a thousand frames each, from which we randomly selected a contiguous 100 frame chunk for our dataset. Each 100 frame chunk was selected such that the positive and negative samples were sampled from the frames annotated by the human analyst as positive or negative for *resolvability*, respectively. We partitioned our dataset into 3 partitions, i.e., 60% for CNN training, 10% for CNN validation, and the remaining 30% for time-series 10-fold cross validation.

Table 1: Performance in terms of AUROC of various ImageNet models on our dataset with or without CLAHE augmentation. Bold denotes model best performance.

Model	**Sensor A**		**Sensor B**	
	Without CLAHE	CLAHE	Without CLAHE	With CLAHE
Densenet 121	0.818	0.836	0.884	0.895
Densenet 169	0.832	0.836	0.884	0.889
Densenet 201	0.830	0.832	0.877	0.893
InceptionResnetV2	0.838	0.844	0.890	**0.896**
InceptionV3	0.829	0.841	0.888	0.892
Mobile Net	0.829	0.843	0.890	0.893
NasNet Large	0.821	0.824	0.886	0.891
NasNet Mobile	0.839	0.842	0.891	0.881
ResNet 50	0.838	**0.855**	0.877	0.891
VGG 16	0.835	0.847	0.878	0.888
VGG 19	0.833	0.818	0.885	0.888
Xception	0.843	0.853	0.880	0.894

4.2. Selecting the best CNN

In this work, we investigated the following networks: Densenet 121/169/201 [9], InceptionResnetV2 [21], InceptionV3 [22], Mobile Net [17], NasNet Large and Mobile [23], ResNet 50 [7], VGG 16/19 [20], and Xception [3]. For each of the pre-trained models we replaced the top dense layers with the following sequence of top layers (non-convolutional layers): a global average pooling layers, a 1024 dense layer with *ReLU* activation, a dropout layer with a rate of 0.2, a 256 neuron dense layer with *ReLU* activation, a dropout layer with a rate of 0.2, and finally a 2 neuron dense layer with *softmax* activation. Each dense layer was initialized with Glorot normal distribution [6].

For input image augmentation, we applied random augmentation to each training image with the following specification: rotation within ± 30 degrees, horizontal flips, vertical flips, and zoom within $\pm 10\%$. Image augmentation was not used during testing inference. We used an iterative deepening training regime, where we first train the top dense layers with the *Adam* optimizer (learning rate $1e^{-3}$) by freezing all lower layers (CNN layers) for 5 epochs. We then iteratively "unfreeze" CNN layers at 25% increments until the entire network is trainable for 20 epochs each. We use the *Adam* optimizer with a learning rate $1e^{-4}$ and $1e^{-5}$ for the first 50% and the remaining 50% iterations, respectively. We used a binary cross-entropy loss function, and saved models at the point with which the validation loss did not improve.

The predictions on our test set are presented in Table 1. In addition to AUROC, we also rate our models in terms of Area Under the Precision Recall Curve (AUPRC), which is a more stable performance metric under class imbalances and for comparison between disparate datasets. We show our results using two methods, i.e. that is with or without CLAHE augmentation. CLAHE augmentation is a method shown in [11] to improve classification performance, particularly when issuing predictions on grayscale images on pre-trained ImageNet models that are optimized for RGB inputs. Formally, the CLAHE grid size and nominal clip limit was augmented according to the following:

$$a \in \mathcal{U}(-log_2(k), log_2(k)) \mid g(k) = k + a, \tag{1}$$

where k is the nominal grid size or the nominal clip limit [11]. In Table 1, we see that CLAHE augmentation significantly improved our predictive performance in terms of AUROC and AUPRC for almost all of the models. The overall most performant model for Sensor A was ResNet50 with CLAHE augmentation, which we use as the CNN for the remainder of the experiments for Sensor A. Similarly, the overall most performant model for Sensor B was InceptionResNetV2 with CLAHE, which we use as the CNN for the remainder of the experiments for Sensor B.

4.3. Time-series

In this section, we investigate how sequential frames (time-series) predictions can improve our performance. Particularly we investigate three methods of doing so: 1) fully connected or dense hidden layers, 2) recurrent neural networks (RNNs) via long-short term memory (LSTM), and 3) 1-D CNNs. We perform experiments on sequences of 10, 20, 30, and 60 frames. We used the same input augmentations presented in the previous section, and fed our predictions from our selected CNN (ResNet 50 and InceptionResNetV2) into each time-series model.

For the dense networks we used three hidden layers with input neurons equal to the number of input frames. We used a *ReLU* activation function, and a dropout layer with a rate of 0.2 between each dense layer. For our 1-D CNN layers, we used three convolutional layers containing 32 kernels of size 3 in each layer. The final output layer in each network was a pair of output neurons with softmax activation.

Our results are presented for Sensor A and B in Table 2 and 3, respectively, for each type of network using 10-fold cross validation. The 1-D CNN performed the best for both sensors. Additionally, the 60 frame

ensemble had the highest AUROC for both sensors, and a general increase in performance was observed as the ensemble sizes increased (particularly true for the 1-D CNNs).

Table 2: Sensor A: time-series performance over various frame ensembles. The baseline single-frame AUROC for this split is 0.855 from Table 1

Model	**AUROC**			
	10 Frames	20 Frames	30 Frames	60 Frames
Dense	0.860 ± 0.021	0.871 ± 0.023	0.865 ± 0.024	0.854 ± 0.022
LSTM	0.853 ± 0.022	0.852 ± 0.027	0.843 ± 0.016	0.853 ± 0.025
1-D CNN	0.863 ± 0.022	0.862 ± 0.028	0.868 ± 0.022	$\mathbf{0.869 \pm 0.020}$

Table 3: Sensor B: time-series performance over various frame ensembles. The baseline single-frame AUROC for this split is 0.896 from Table 1

Model	**AUROC**			
	10 Frames	20 Frames	30 Frames	60 Frames
Dense	0.883 ± 0.027	0.891 ± 0.023	0.902 ± 0.025	0.899 ± 0.026
LSTM	0.884 ± 0.024	0.897 ± 0.019	0.894 ± 0.021	0.902 ± 0.021
1-D CNN	0.896 ± 0.023	0.899 ± 0.017	0.902 ± 0.023	$\mathbf{0.903 \pm 0.021}$

4.4. Reproducibility

For reproduction, all neural networks were trained using Python 3 and *Keras* in conjunction with TensorFlow [2, 1]. Operating system and hardware specifications include RedHat Linux 7 on a NVidia DGX Workstation with four Tesla v100 GPUs with 32 GB of memory on each card. Because the goal of this work was a feasibility investigation, we did not tune or search for optimal hyperparameters.

5. Conclusion

We have shown the ability for CNNs to predict *resolvability* with great accuracy. Particularly, given a single frame from our dataset we showed that ResNet 50 (Sensor A) and InceptionResNetV2 (Sensor B) performed the best, and that CLAHE augmentation significantly improved classification performance. Additionally, we showed that a 1-D CNN performed the best for time-series predictions, especially when the increasing the number of input frames. For future work, it may be beneficial to explore a customized network, as well as perform an exhaustive search for optimal hyperparameters. Additionally, we would like to apply active learning to improve to sharpen our overall *resolvability* performance as well as implement a filtering or triage system as done in [12]. We hope that this work will inspire others in the SSA community to investigate neural networks for ground-based imaging.

References

[1] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] François Chollet et al. Keras, 2015.

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[4] Robert Furfaro, Richard Linares, and Vishnu Reddy. Space objects classification via light-curve measurements: Deep convolutional neural networks and model-based transfer learning. 09 2018.

[5] Levi Fussell and Ben Moews. Forging new worlds: high-resolution synthetic galaxies with chained generative adversarial networks. *Monthly Notices of the Royal Astronomical Society*, 485(3):3203–3214, 03 2019.

[6] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[8] Alex Hocking, James E. Geach, Yi Sun, and Neil Davey. An automatic taxonomy of galaxy morphology using unsupervised machine learning. *Monthly Notices of the Royal Astronomical Society*, 473(1):1108–1129, 09 2017.

[9] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[10] B. Jia, K. D. Pham, E. Blasch, Z. Wang, D. Shen, and G. Chen. Space object classification using deep neural networks. In *2018 IEEE Aerospace Conference*, pages 1–8, March 2018.

[11] Trent Kyono, Fiona J. Gilbert, and Mihaela van der Schaar. MAMMO: A deep learning solution for facilitating radiologist-machine collaboration in breast cancer diagnosis. *CoRR*, abs/1811.02661, 2018.

[12] Trent Kyono, Fiona J. Gilbert, and Mihaela van der Schaar. Improving workflow efficiency for mammography using machine learning. *Journal of the American College of Radiology*, 2019.

[13] Richard Linares. Space object classification using deep convolutional neural networks. 07 2016.

[14] Jacob Lucas, Brandoch Calef, and Trent Kyono. Recovering astronomical images with deep neural network supported bispectrum processing. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*. 2018.

[15] John Mooney, Richard Cleis, Trent Kyono, and Matthew Edwards. Modular mount control system for telescopes. In *Advanced Maui Optical and Space Surveillance (AMOS) Technologies Conference*. 2017.

[16] A. Asensio Ramos, J. de la Cruz Rodriguez, and A Pastor Yabar. Real-time multiframe blind deconvolution of solar images. 06 2018.

[17] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018.

[18] Kevin Schawinski, Ce Zhang, Hantian Zhang, Lucas Fowler, and Gokula Krishnan Santhanam. Generative adversarial networks recover features in astrophysical images of galaxies beyond the deconvolution limit. *Monthly Notices of the Royal Astronomical Society: Letters*, 467(1):L110–L114, 01 2017.

[19] Timothy J. Schulz. Multiframe blind deconvolution of astronomical images. *J. Opt. Soc. Am. A*, 10(5):1064–1073, May 1993.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[21] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[22] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[23] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017.