29T　機械学習による大気の揺らぎの補正技術の開発

成蹊高等学校天文気象部：赤川 陽大、 岡部 慎、松澤 幸希、森田 仁意（高2）、
福井 千紘、根岸 陽向、五島 紘（高1）【成蹊高等学校】

1. 要旨

天文画像処理のソフトウェアは広く使われている。手法もまた様々であり補償光学的処理や、ステライメージなどの画像処理ソフトで、明瞭な画像を選択的に抽出し重ね合わせる手法などがある。そんな中私たちは、機械学習を取り入れることで以前より画像を明瞭にすることを目指した。

2. 研究背景

今回、天体の模様の細部を色彩良く引き出すために惑星を素材とし、大きさなどの観点から扱いやすい木星を学習モデルとして使用した。したがって私たちが作ったプログラムは木星の画像処理にのみ用いることが可能なものとなっている。プログラムにはU-Netを利用したNoise2Noise、また楕円を真円にするためのアフィン変換を用いた。

3. Noise2NoiseとU-Net

Noise2Noise[1]とは、ノイズを除去する機械学習の手法で、複数のノイズ画像を用意することでクリーンな画像を予測するアルゴリズムである。今回はこれをU-Net[2]によって実装した。U-net (図1のようなニューラルネットワーク)とは、下層で細部の特徴を引き出し、画像を簡略化・圧縮し上層でそれを復元することで、細部の色彩・区別を正確に出す仕組みのことを指す。処理層一つには画像の特徴の抽出に使われる技術であるCNN（畳み込みニューラルネットワーク）[3]の技術が使われている。特徴抽出を行い圧縮されたマップを復元することでノイズ除去後の画像を作成した。

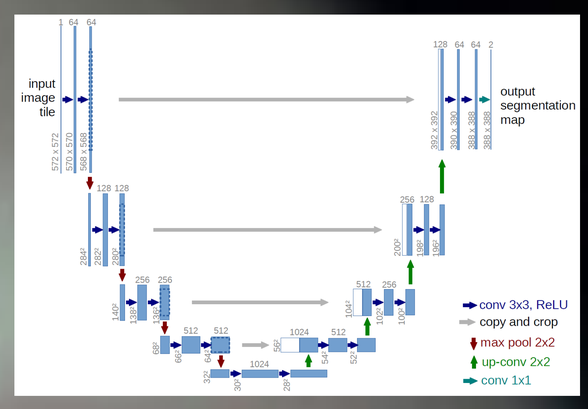

図1　U-netで組まれたニューラルネットワーク

4. 画像処理

木星面の動画撮影をタカハシ製望遠鏡FS-152、Explore Scientific製5倍バローレンズ、Player One製CMOSカメラUranus-Cで行った。撮影した動画をffmpegを用いて連番画像に変換した。連番画像一枚ずつにアフィン変換[4]を施し楕円形の木星面を真円に変えた。図2(左)が連番画像で図2(右)が変換後である。U-netの処理層に連番画像を全て通しノイズを除去する。図2(中)から図2(右)でノイズ除去。最後にOpenCVですべての画像を連結して動画にした。

図2(左)
撮影時そのままの生データ

図2(中)
アフィン変換を加えた画像

図2(右)
アフィン変換＋ノイズ除去

5. 結果と考察

右図はNoise2Noiseでのディープラーニングにおける学習過程で重要な指標となる損失関数[5]の値を、学習回数ごとにプロットしたものだ。AIは、パラメータを調整しながら計算を行い、その際、この損失関数の値を最小化するようにパラメータを最適化する。右図には、学習に使用したデータから計算される損失（Train Loss）と、学習に使用していない検証用データから計算される損失（Valid Loss）の2種類が表示されている。右図から学習が進むにつれて、両方の損失が減少していることが確認できる。
減少量が途中で緩やかになっているが、現状では明確な原因は明らかになっていないが、今後改善していきたい。

なお、アフィン変換の機械学習を使用した代替として、強化学習（AIが自立して、報酬が最大となるように試行錯誤する）を用いてできないか試作してみたが、時々変換に失敗するため、アフィン変換を使用した。

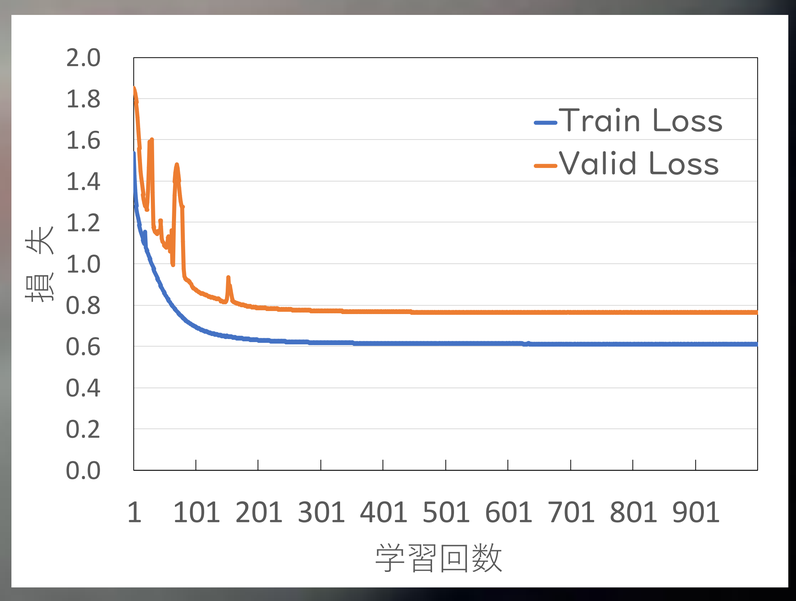

図3　学習回数(横軸)と損失(縦軸)の推移

6. まとめ

このようにして私たち成蹊高校天文気象部は、機械学習による大気の揺らぎの補正技術の開発を行った。今回の知見を活かして今後、リアルタイムによる処理や、学習量を増やしてさらなる精度の向上を目指す。私たちは機械学習による画像処理が更なる天文学の発展に寄与できることを願い、天文学に関する分野研究でAIが革新をもたらすことを期待したい。

参考文献：
[1] 雛星 φυβλας、ノイズいっぱいの画像だけで学習しても綺麗な画像が復元できる『noise2noise』をpytorchで実装 - Qiita、https://qiita.com/phyblas/items/d4884a5804leaa01ef5e
[2] @gensal【Pytorch】UNetを実装する #Python - Qiita、https://qiita.com/gensal/items/03e9a6d0f7081e77ba37e
[3] もっと MNIST で高い精度を出す | PyTorch で学ぶ！やさしい Deep Learning、https://zenn.dev/seelog/books/easy_deep_learning/viewer/more_mnist
[4] 隼平 山本　斜めから撮ったダーツボードを「円」にする方法 – 理屈編 #Python - Qiitaa、https://qiita.com/mikemike_kus/items/e5e2f1928b7b7b66ca20
[5] 原 辰徳　ディープラーニングの実装と過学習対策 #Python - Qiita、https://qiita.com/hara_tatsu/items/b7423e90574cf7730978