Reconstructing incomplete historical Arctic sea-ice concentration data from 1950 (1901)

Applied Machine Learning 2022 – Final Project

Nils Bochow, Anna Poltronieri

Introduction

- Sea ice is frozen seawater that floats on the surface of the ocean
- The rapid loss of Arctic sea ice (ASI) in the last decades is one of the most evident manifestations of anthropogenic climate change
- An ice-free Arctic would impact climate and ecosystems, both regionally and globally

Introduction

- Sea ice is frozen seawater that floats on the surface of the ocean
- The rapid loss of Arctic sea ice (ASI) in the last decades is one of the most evident manifestations of anthropogenic climate change
- An ice-free Arctic would impact climate and ecosystems, both regionally and globally

↓

Knowing the history of the ASI is crucial to understanding its future evolution

Problem

- Satellite observations for the ASI start from 1979
- We have an incomplete spatiotemporal dataset of ASI concentration (1901-2013)
- We want to reconstruct the missing data

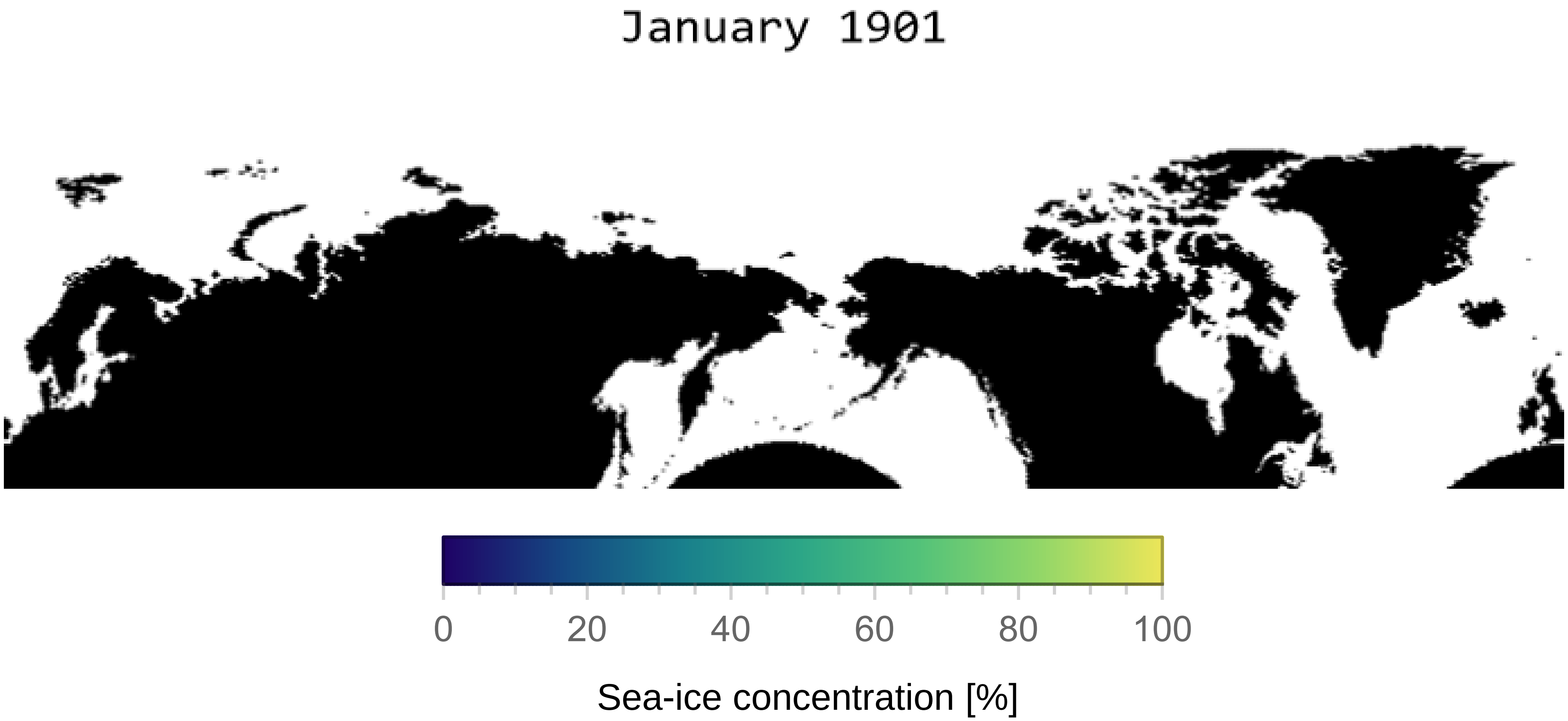
Problem

- Satellite observations for the ASI start from 1979
- We have an incomplete spatiotemporal dataset of ASI concentration (1901-2013)
- We want to reconstruct the missing data

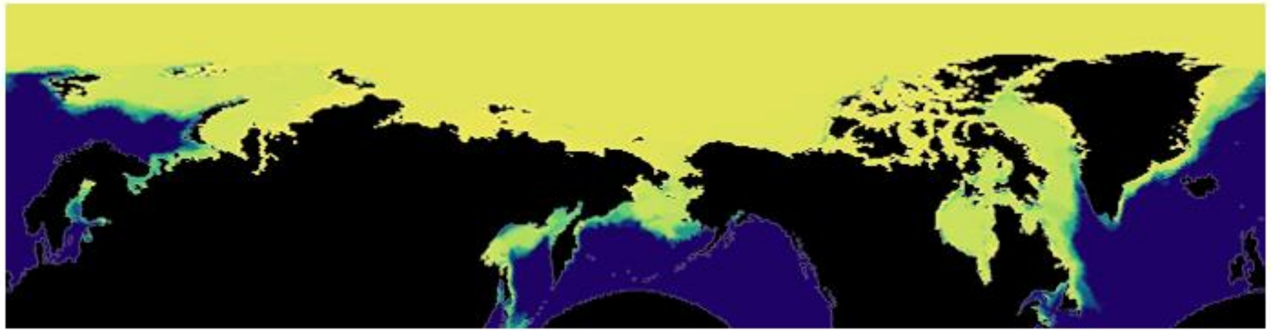
↓

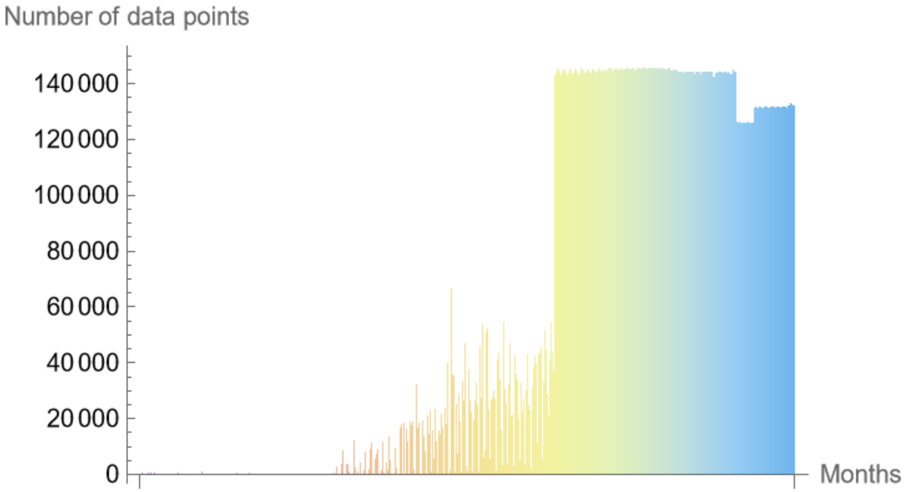
How can we fill the (massive) gaps in the dataset?

Dataset

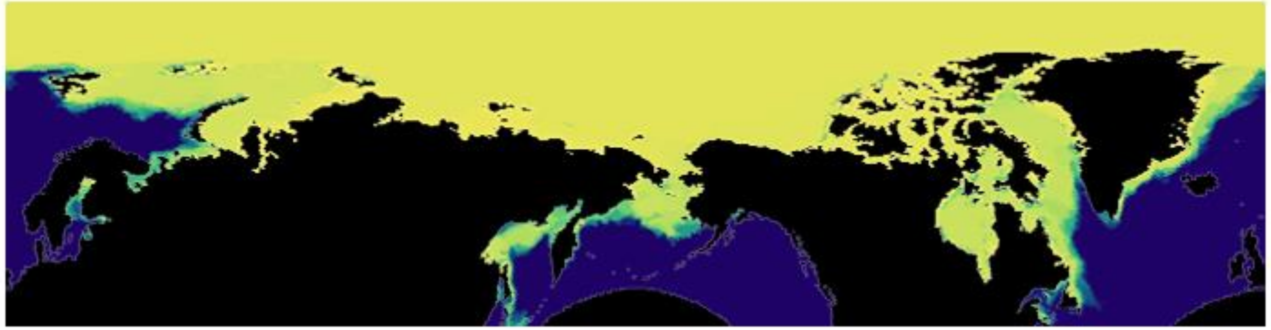


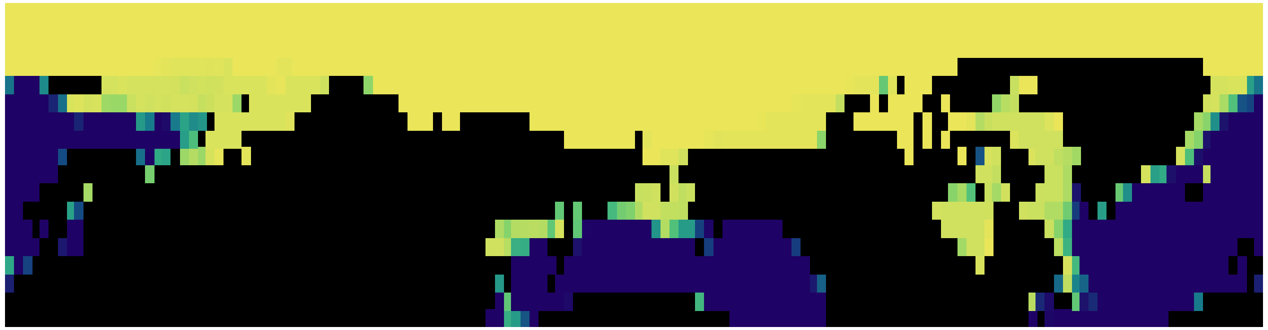
Preprocessing

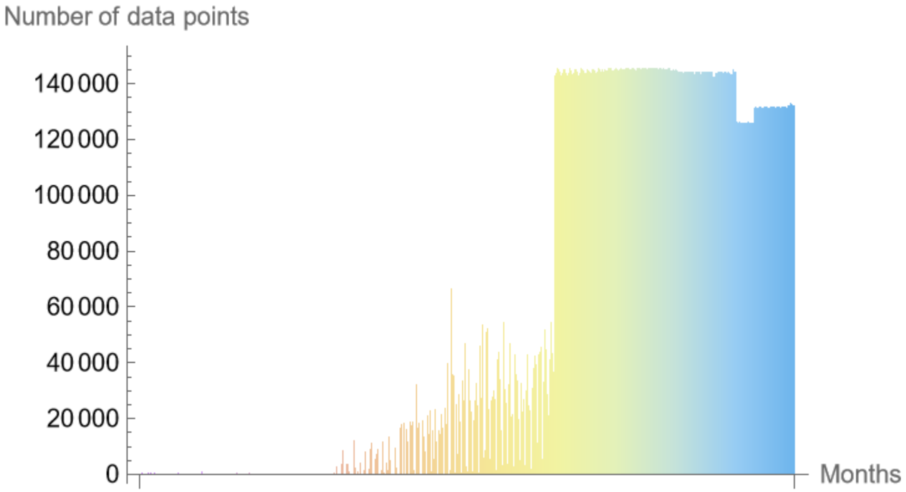


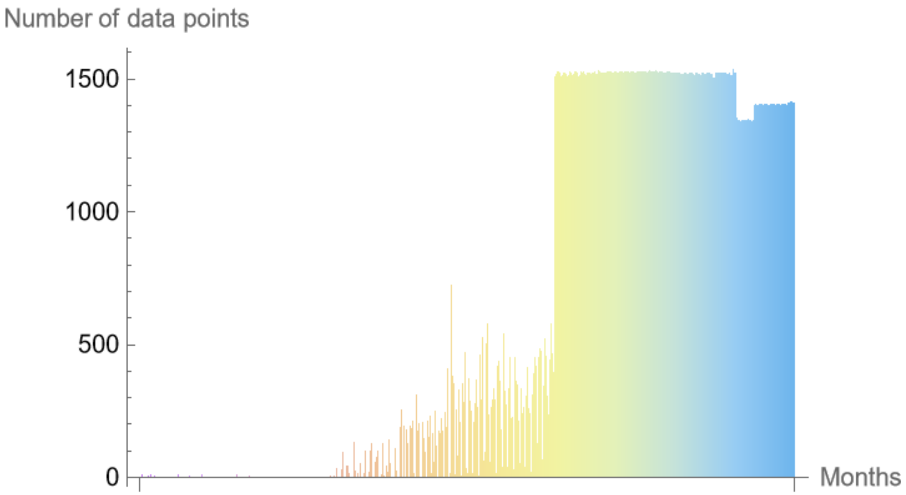


Preprocessing









Methods

- Regression at every site (90-10% split training-validation)

Methods

- Regression at every site (90-10% split training-validation)
  - Linear regression

Methods

- Regression at every site (90-10% split training-validation)
  - Linear regression
  - Ridge regression

Methods

- Regression at every site (90-10% split training-validation)
  - Linear regression
  - Ridge regression
  - Single Layer Perceptron

```
Layer (type)                 Output Shape              Param #
=================================================================
input_layer (Dense)          (None, 4)                 20

hidden_layer (Dense)         (None, 100)               500

output (Dense)               (None, 1)                 101

=================================================================
Total params: 621
Trainable params: 621
Non-trainable params: 0
```

Methods

- Regression at every site (90-10% split training-validation)
  - Linear regression
  - Ridge regression
  - Single Layer Perceptron
  - Multi Layer Perceptron

Layer (type)	Output Shape	Param #
input_layer (Dense)	(None, 4)	20
hidden_layer (Dense)	(None, 100)	500
output (Dense)	(None, 1)	101

Total params: 621
Trainable params: 621
Non-trainable params: 0

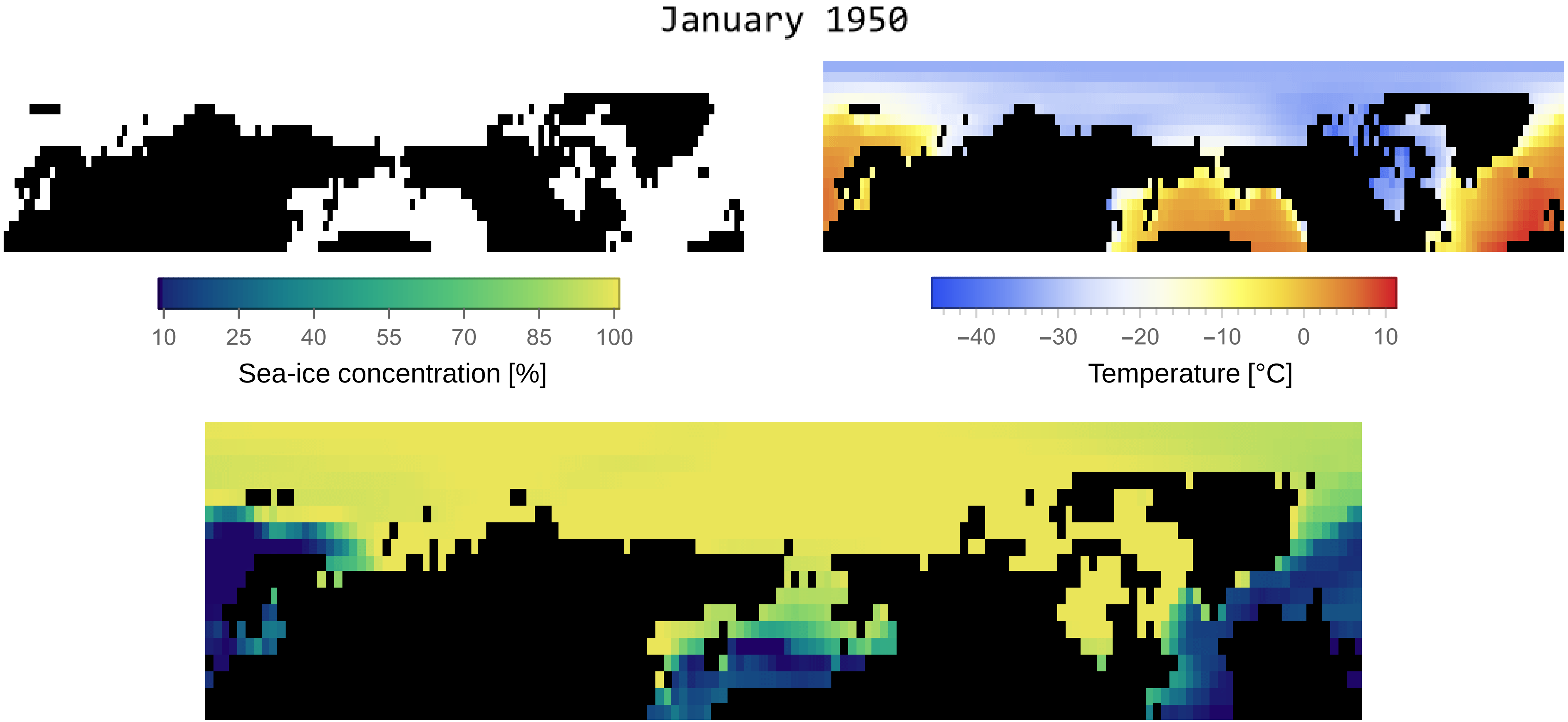
Layer (type)	Output Shape	Param #
input_layer (Dense)	(None, 4)	20
hidden_layer1 (Dense)	(None, 50)	250
hidden_layer2 (Dense)	(None, 50)	2550
hidden_layer3 (Dense)	(None, 100)	5100
hidden_layer4 (Dense)	(None, 100)	10100
hidden_layer5 (Dense)	(None, 100)	10100
hidden_layer6 (Dense)	(None, 12)	1212
output (Dense)	(None, 1)	13

Total params: 29,345
Trainable params: 29,345
Non-trainable params: 0

Methods

- Regression at every site (90-10% split training-validation)
  - Linear regression
  - Ridge regression
  - Single Layer Perceptron
  - Multi Layer Perceptron
- Partial CNN-based image inpainting (90-10% split training-validation on ERA5 daily ASI concentration dataset)

Prediction (Multi Layer)

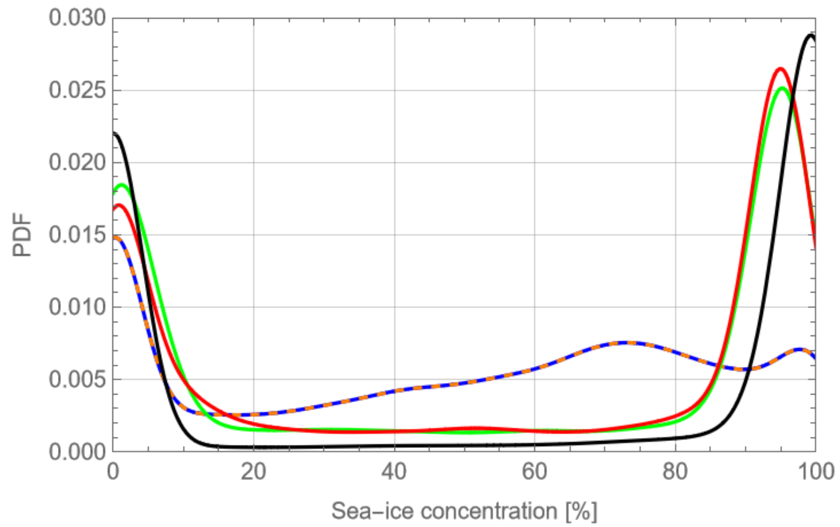


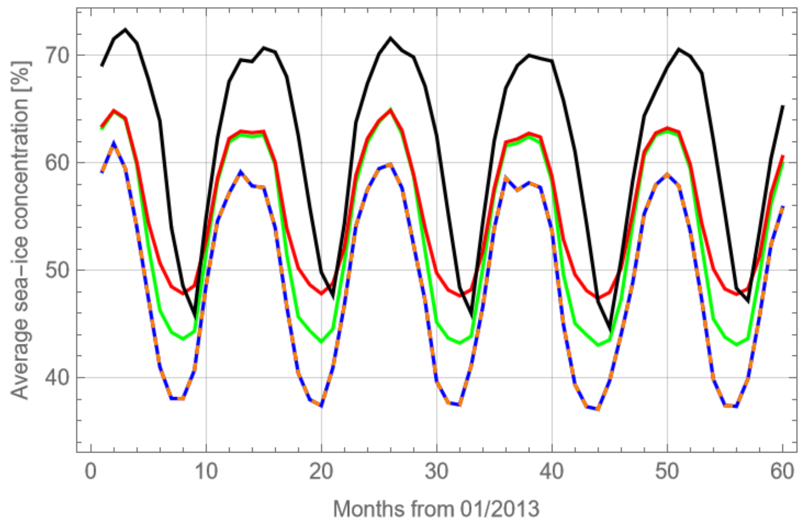
Results

Method	MSE on validation set
Linear Regression	643.8
Ridge Regression	643.8
Single Layer Perceptron	314
Multi Layer Perceptron	242

Results

Method	MSE on validation set
Linear Regression	643.8
Ridge Regression	643.8
Single Layer Perceptron	314
Multi Layer Perceptron	242





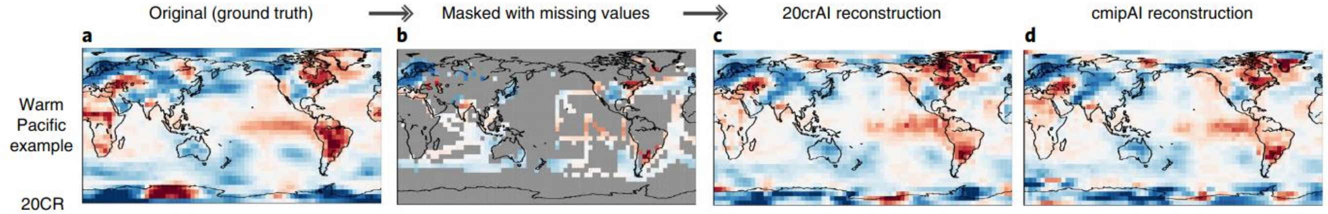
Attempt with partial convolution

Based on

ARTICLES
https://doi.org/10.1038/s41561-020-0582-5
nature geoscience
Check for updates

Artificial intelligence reconstructs missing climate information

Christopher Kadow[1,2], David Matthew Hall[3] and Uwe Ulbrich[2]



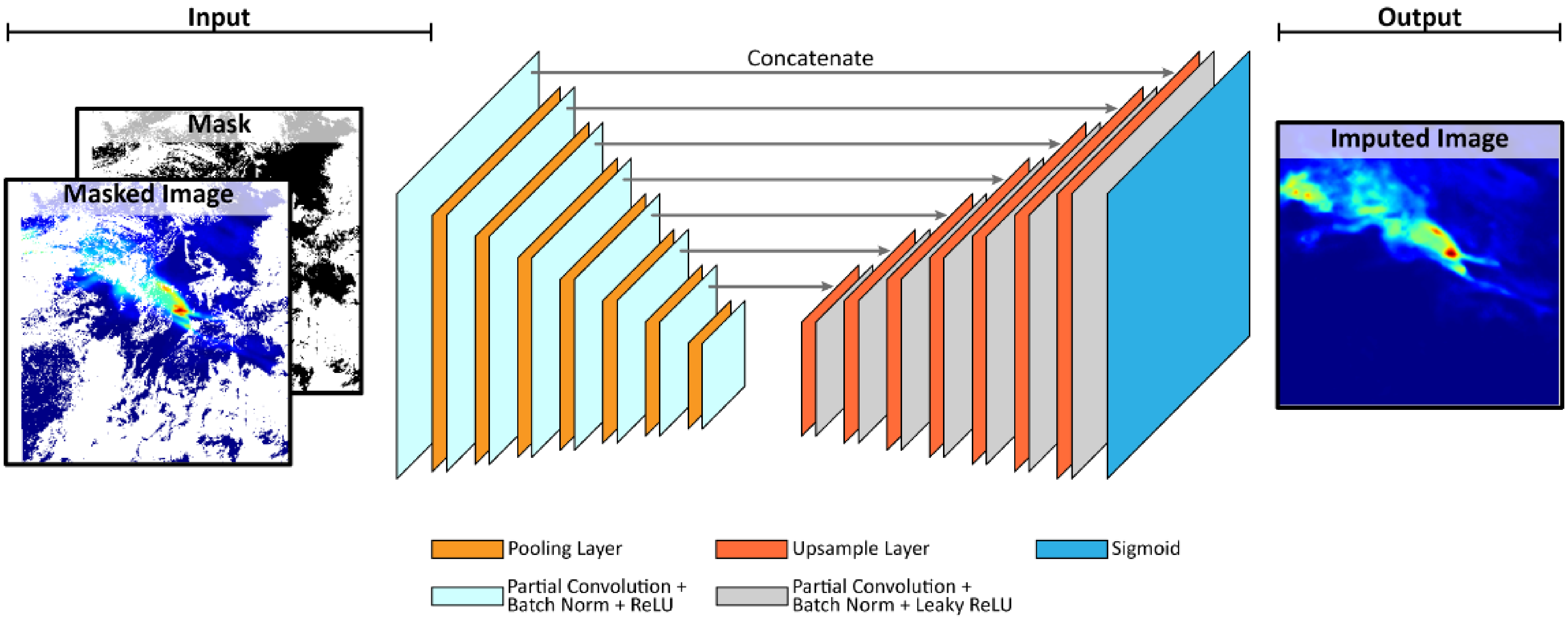
which is based on

Image Inpainting for Irregular Holes Using Partial Convolutions

Guilin Liu Fitsum A. Reda Kevin J. Shih Ting-Chun Wang
Andrew Tao Bryan Catanzaro

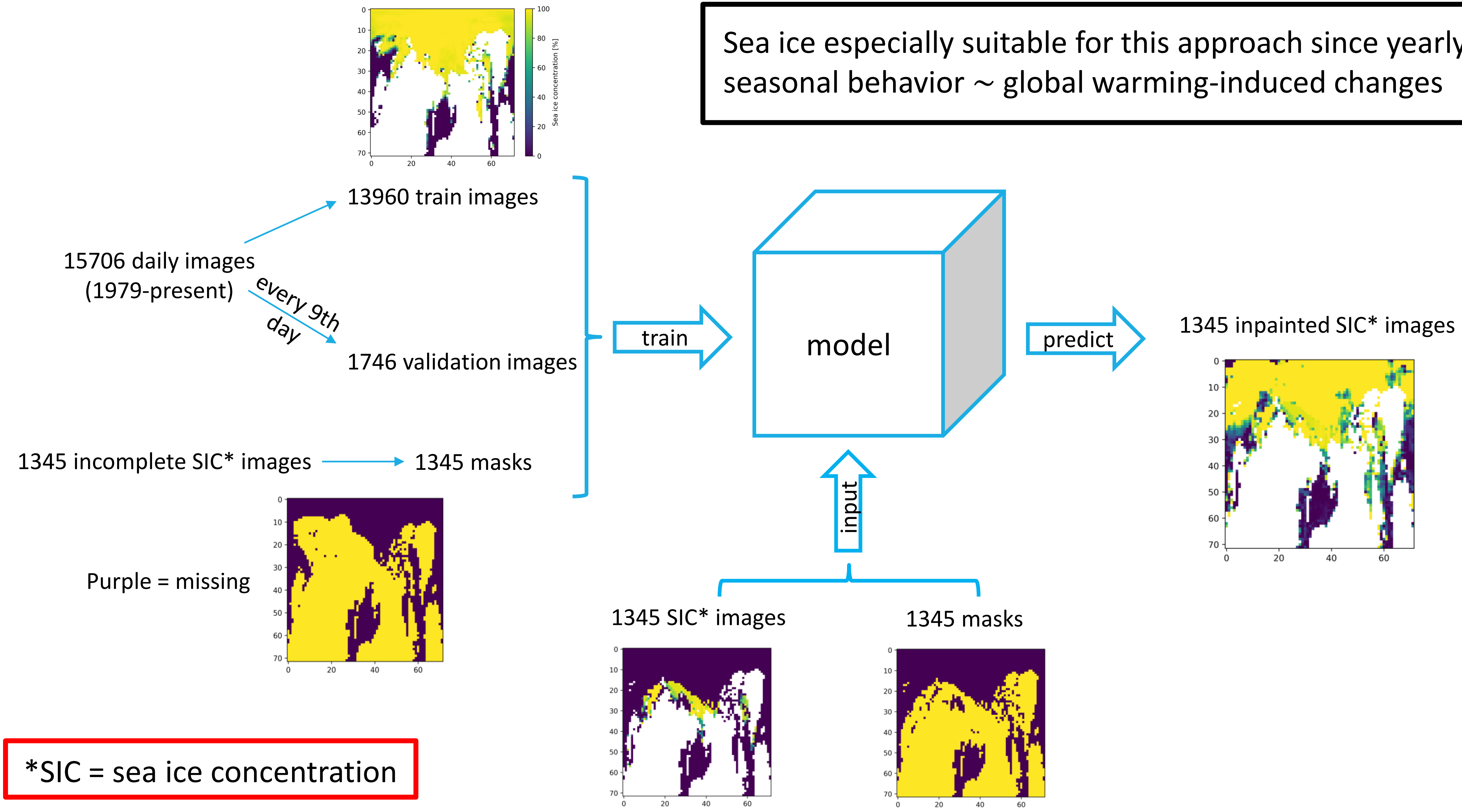
NVIDIA Corporation

Attempt with partial convolution



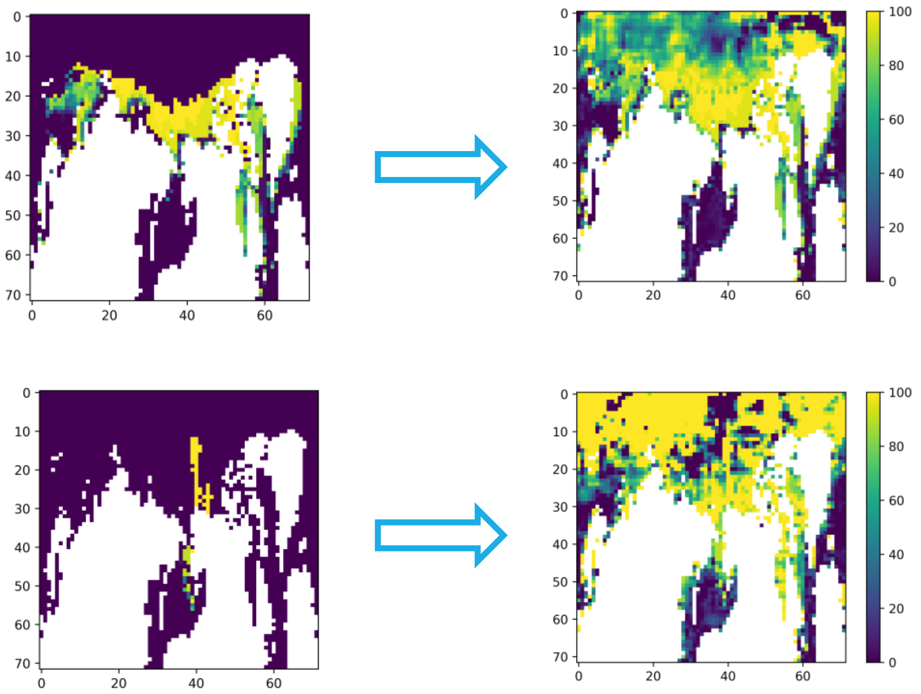
Application of a Partial Convolutional Neural Network for Estimating Geostationary Aerosol Optical Depth Data, Lops *et al.* (2021)

Sea ice especially suitable for this approach since yearly seasonal behavior ~ global warming-induced changes



*SIC = sea ice concentration

But...

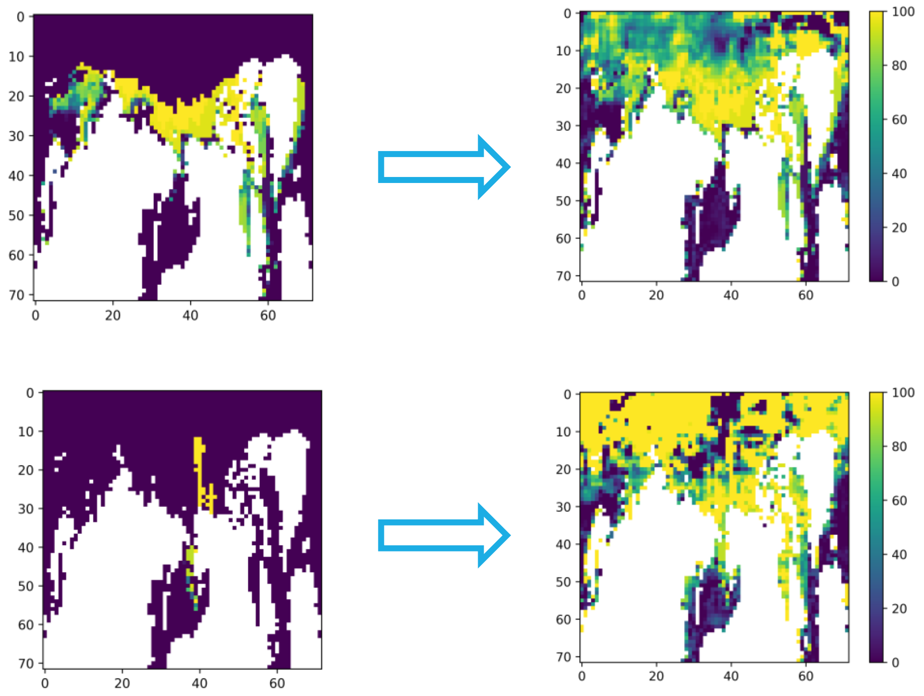


SIC* < 0 and > 100?

(-5189 ≤ SIC* ≤ 16418)

*SIC = sea ice concentration

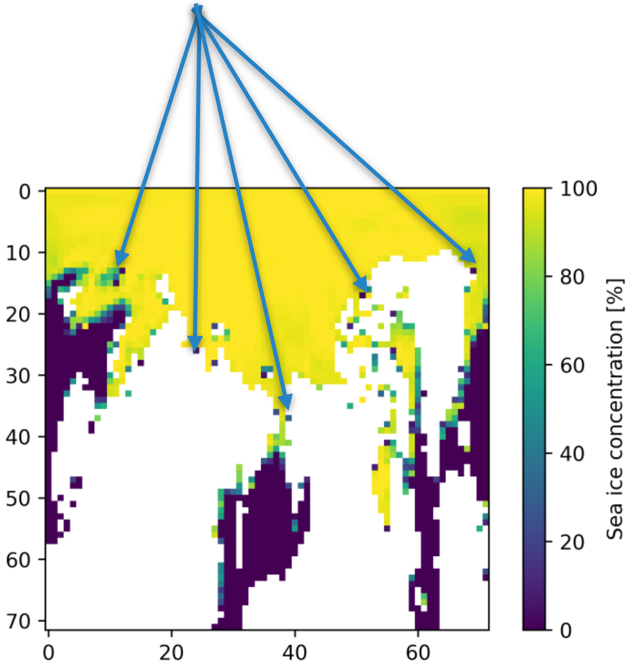
But…



Several problems:

- Continent distribution different between train and test data
- How to deal with continents?
- We didn't use anomalies (normalize images)
- Some missing data in training data?

SIC* < 0 and > 100?
(-5189 ≤ SIC* ≤ 16418)



*SIC = sea ice concentration

Conclusions

- All the regression methods underestimated the observed sea ice concentration
- The multi layer perceptron showed the best performance in terms of MSE compared to the other regression models
- The partial convolution might be a better approach, but it requires additional work (especially in the data preprocessing phase)

Future work

- **Goal:** complete dataset of monthly sea ice concentration on a 0.25°×0.25° grid from the year 1900

- Reconstruction using PConv Network where enough data are available

- “Forecast” of the past using ConvLSTM or similar method

- Explore more recent methods for image inpainting

Appendix

Datasets

- ERA5 monthly temperature data (1950 - 2021): DOI: 10.24381/cds.f17050d7
- Sea ice concentration data by Vasily Smolyanitsky, Arctic and Antarctic Research Institute (unpublished)
- Observational sea ice concentration data: https://doi.org/10.7265/jj4s-tq79
- ERA5 hourly sea ice concentration data (1979 - 2021): DOI: 10.24381/cds.adbb2d47

Preprocessing

- The data used as input for the regression methods were regridded to a 2.5°×2.5° resolution
- The data used as input for the partial convolution method were regridded to a squared 72×72 cells grid (5°×0.625°)
- The regridding assigned values between 100 (maximum for SIC) and 122 (default value for land cells) to some cells, that we manually set to land
- The daily data used in the partial convolution was downloaded as hourly data (24 files per day) and then averaged to daily data

Partial convolutional layer

The partial convolutional layer is defined as:

$$x'_{(i,j)} = \begin{cases} \mathbf{W}^T(\mathbf{X}_{(i,j)} \odot \mathbf{M}_{(i,j)}) r_{(i,j)}, & ||\mathbf{M}_{(i,j)}||_1 > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where

$$r_{(i,j)} = \frac{||\mathbf{1}_{(i,j)}||_1}{||\mathbf{M}_{(i,j)}||_1}, \tag{2}$$

with *X* as input, *M* as mask, and *W* as filter weight matrix.

The mask is updated after each partial convolution as:

$$m'_{(i,j)} = \begin{cases} 1, & \text{if } ||\mathbf{M}_{(i,j)}||_1 > 0 \\ 0, & \text{otherwise} \end{cases}$$

Source: Liu *et al.* (2018), *Partial Convolution based Padding*, DOI: 1811.11718.pdf (arxiv.org)

Partial convolutional layer

Example calculation of partial convolution:

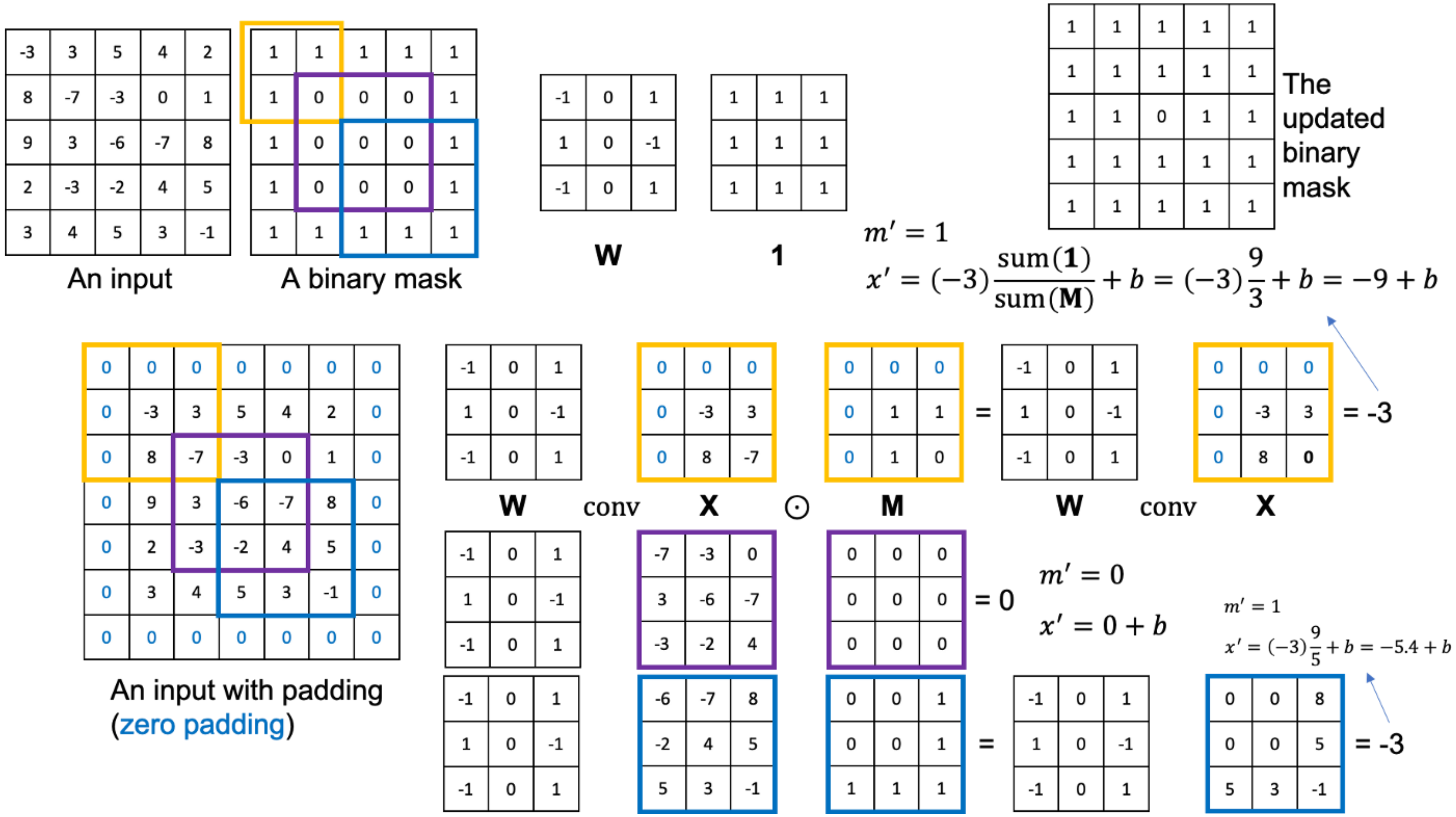


Image by Chu-Tak Li ([Pushing the Limits of Deep Image Inpainting Using Partial Convolutions | by Chu-Tak Li | Towards Data Science])

Network architecture

Module Name	Filter size	# Filters/Channels	Batch Norm	Nonlinearity
PConv1	7x7	18	No	ReLU
PConv2	5x5	36	Yes	ReLU
PConv3	5x5	72	Yes	ReLU
NearestUpSample1 Concat1(w/PConv2) PConv4	3x3	72 72+36 36	Yes	LeakyRelu(0.2)
NearestUpSample2 Concat2(w/ PConv1) PConv5	3x3	36 36+18 18	Yes	LeakyRelu(0.2)
NearestUpSample3 Concat3(w/ Input) PConv6	3x3	18 18 +3 3	No	-

Partial convolutional network sources

- The code we use for the partial convolutional U-net is based on the paper *Artificial intelligence reconstructs missing climate information* by Kadow *et al.* (2021) (https://www.nature.com/articles/s41561-020-0582-5 & https://github.com/FREVA-CLINT/climatereconstructionAI)

- which is a modified version of https://github.com/naoto0804/pytorch-inpainting-with-partial-conv, a "ready-to-go" implementation of the paper *Image Inpainting for Irregular Holes Using Partial Convolutions* by Liu *et al.* (2018) (https://arxiv.org/pdf/1804.07723.pdf & https://github.com/NVIDIA/partialconv)

PConv loss function

The total loss function is the sum of several different loss functions, including a per-pixel loss, perceptual loss, style loss, and a total variation loss:

$$\mathcal{L}_{total} = \mathcal{L}_{valid} + 6\mathcal{L}_{hole} + 0.05\mathcal{L}_{perceptual} + 120(\mathcal{L}_{style_{out}} + \mathcal{L}_{style_{comp}}) + 0.1\mathcal{L}_{tv}$$

The perceptual loss is the L^1 distance between the ground truth and the raw image/computed image after projecting these images into higher-level feature spaces using an ImageNet-pre-trained VGG-16.

The style loss is similar to the perceptual loss but with applied autocorrelation (Gram matrix) on each feature map before calculating the L^1 distance.

The total variation loss is a smoothing penalty on the region of 1-pixel dilation of the holes.

Source: Liu *et al.* (2018), *Image Inpainting for Irregular Holes Using Partial Convolutions* (https://arxiv.org/abs/1804.07723)

Running the PConv UNet

- We run the model on an NVidia Tesla V100 GPU on the HPC at the Potsdam Institute for Climate Impact Research (Germany) with an average of 27 iterations/second

- We train the model for 500000 iterations and run 500000 more iterations of fine-tuning using a batch size of 18 (total duration ca. 10 – 12 hours)

- Our implementation is available at [GitHub - spin0r/image_reconstruction: Image reconstruction for climate using AI]