EGU General Assembly 2024
Thursday 18/04/2024

Spatial downscaling of CAMS surface pollutants using Machine Learning

Athanasios Tsikerdekis[1], Michael Tsikerdekis[2] and Henk Eskes[1]

1. Royal Netherlands Meteorological Institute (KNMI), De Bilt, the Netherlands
2. Western Washington University, Bellingham, WA, USA

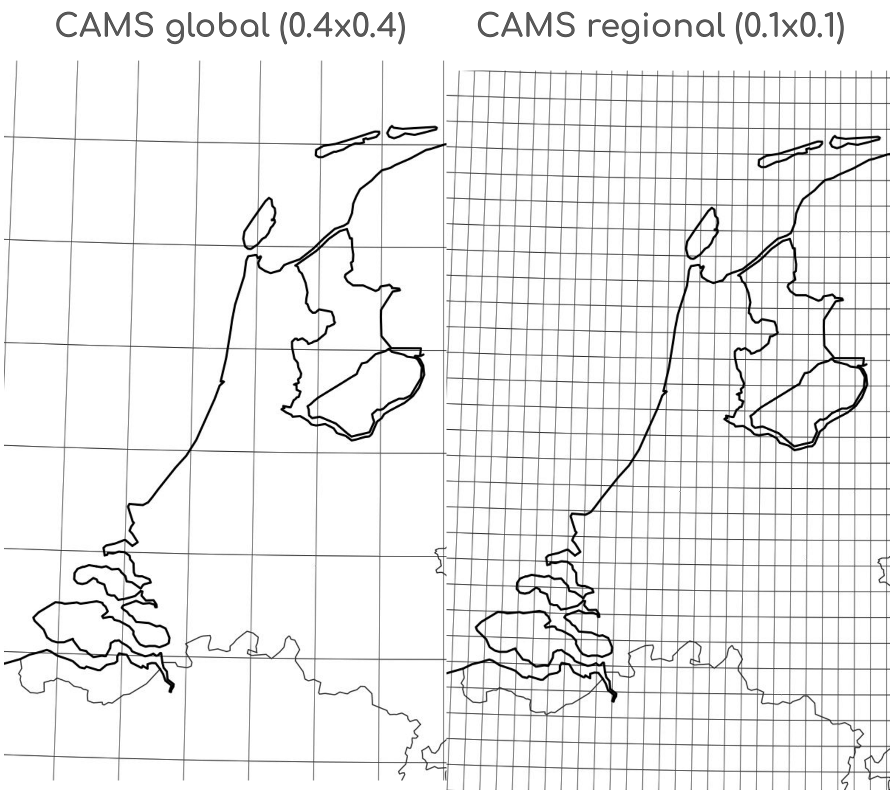
Motivation

Can we go from global (~40km) to regional (10km) resolution with ML?

Assumption:
Surface concentration of a pollutant correlates with various meteorological, emission and surface characteristics

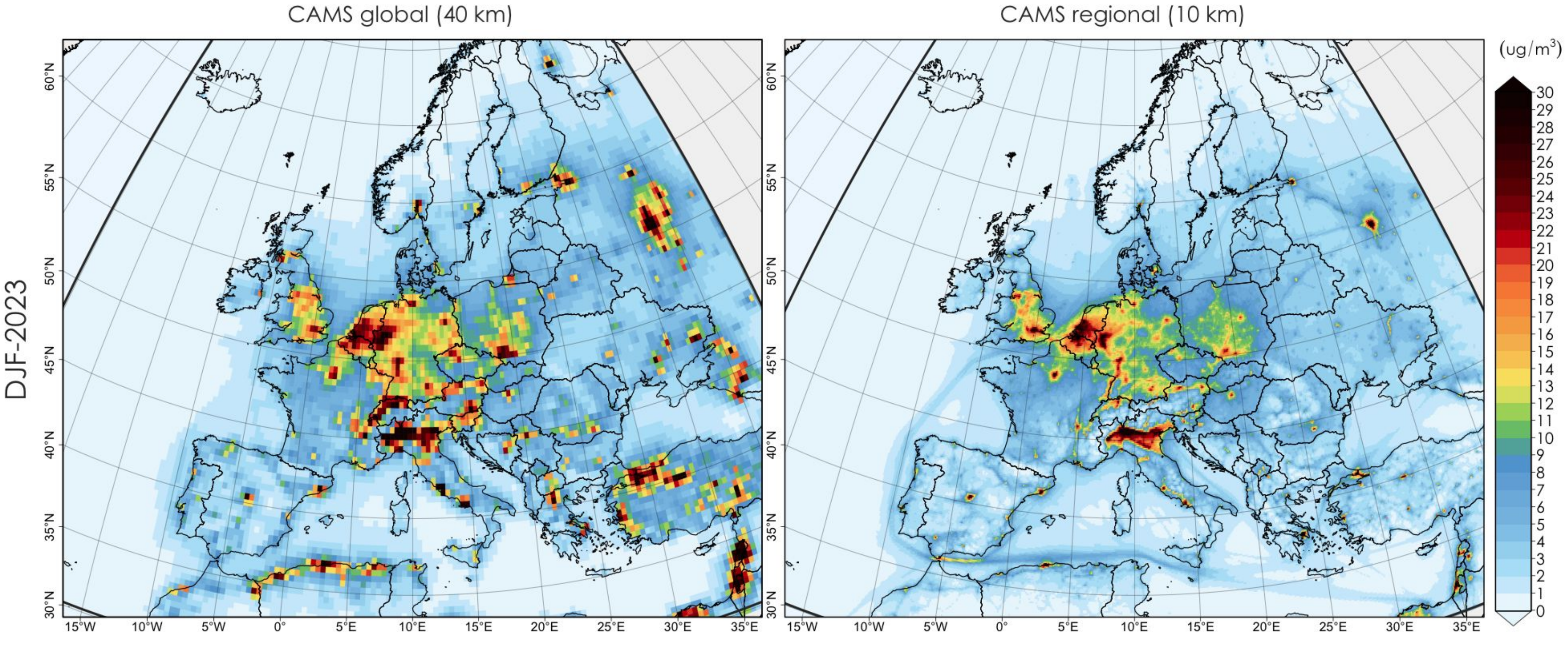
Data availability:
Since the final goal will be to use the algorithm globally for surface pollutants, the <u>inputs</u> need to be:

- Available globally in 10km.
- Operationally created
- Provided as a forecast in NRT



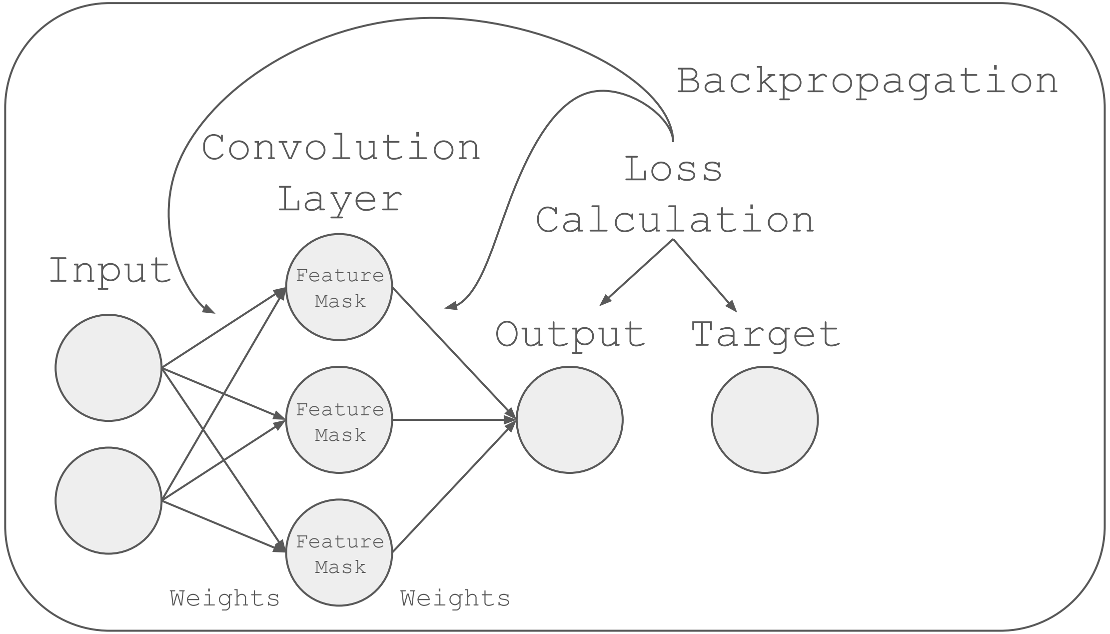
Motivation

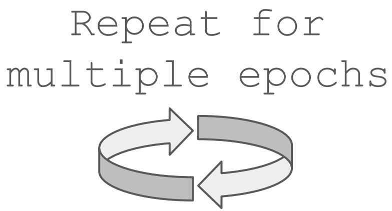
Can we go from global (~40km) to regional (10km) resolution globally for "free" (computationally) ?



A simplified Convolutional Neural Network (CNN) structure

A CNN is a deep learning architecture designed specifically for processing structured grid-like data, such as images or spatial data. It combines several Input in order to predict (Output) a Target. The Input, Output and Target are pictures or model domains of N x N pixels or grid cells respectively.



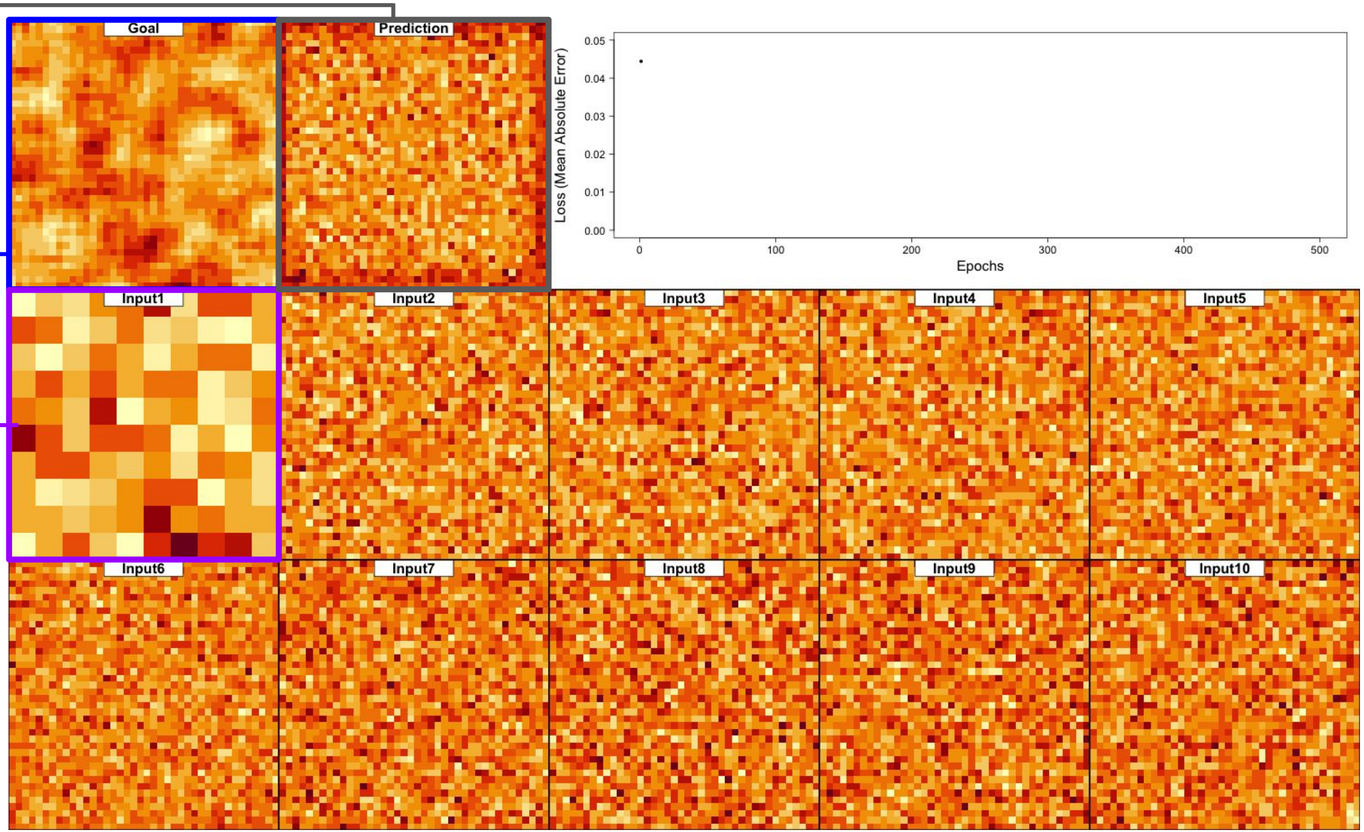


Toy model for spatial downscaling

Prediction
Our recreated Goal using a trained ML algorithm

Target (Goal)
What we want to recreate

Input1 = Coarse Goal
Using as one of the inputs the coarse version of goal



Input Data

Meteorology **(10 km)**
ECWMF
operational forecast

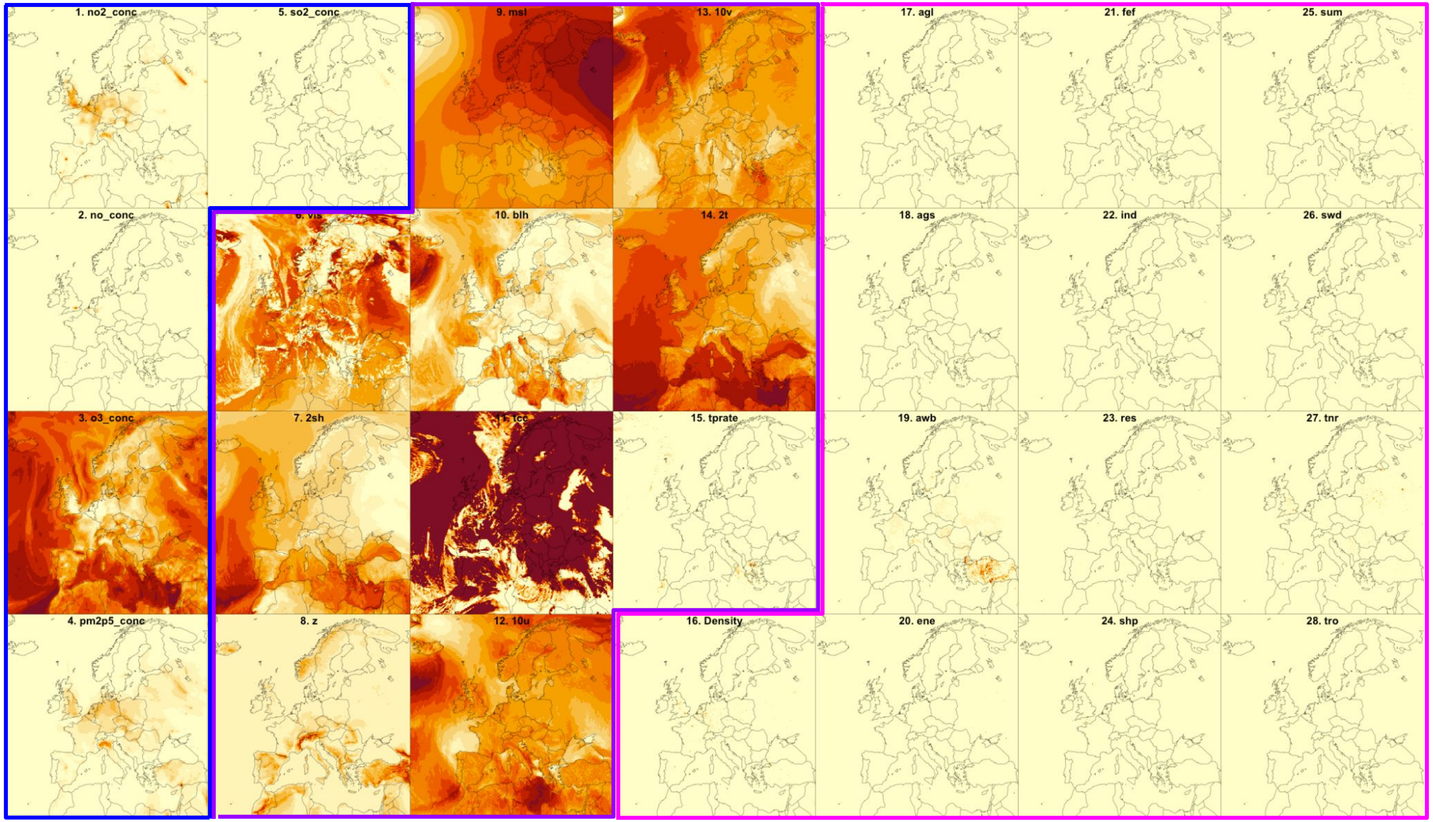
Emissions **(10 km)**
CAMS global

Future plans
to use
landuse as
input

Surface
Concentration
(40km)

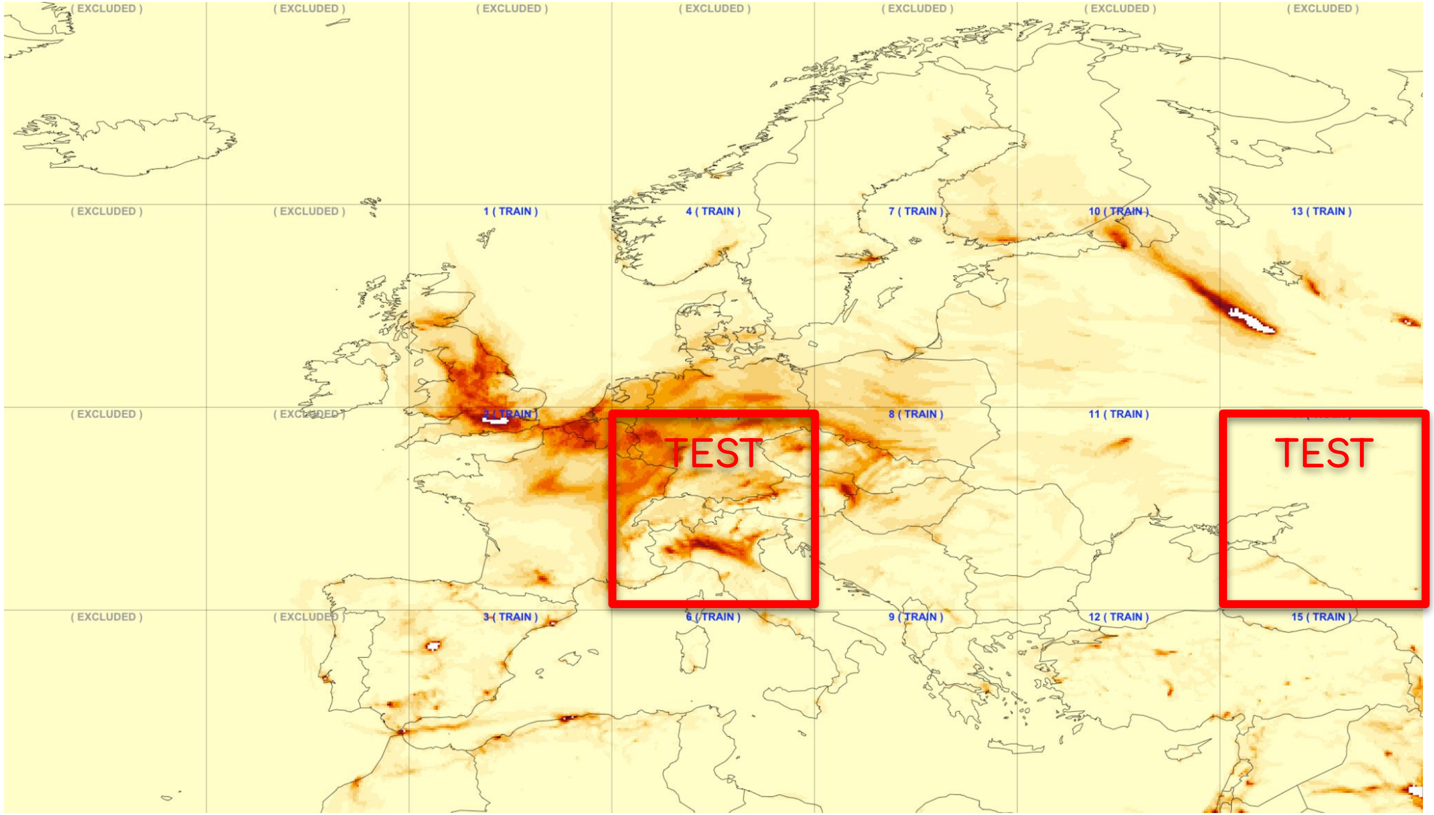
ENSEMBLE
of CAMS regional

Interpolated from
10km → 40km → 10km
(fine scale details are lost)



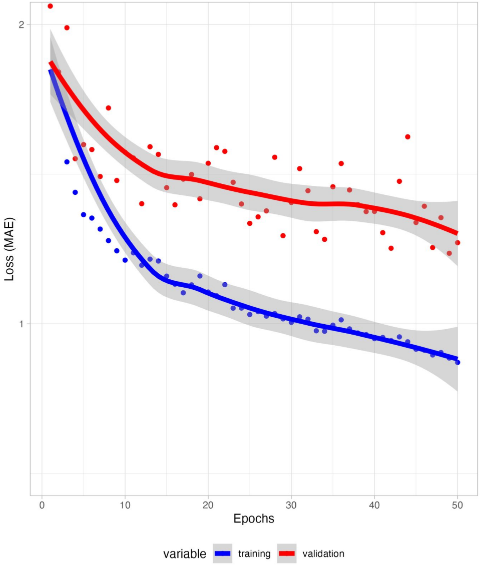
Data

Splitting data into training and testing



Results: Prediction for a specific timestep (09:00 UTC)

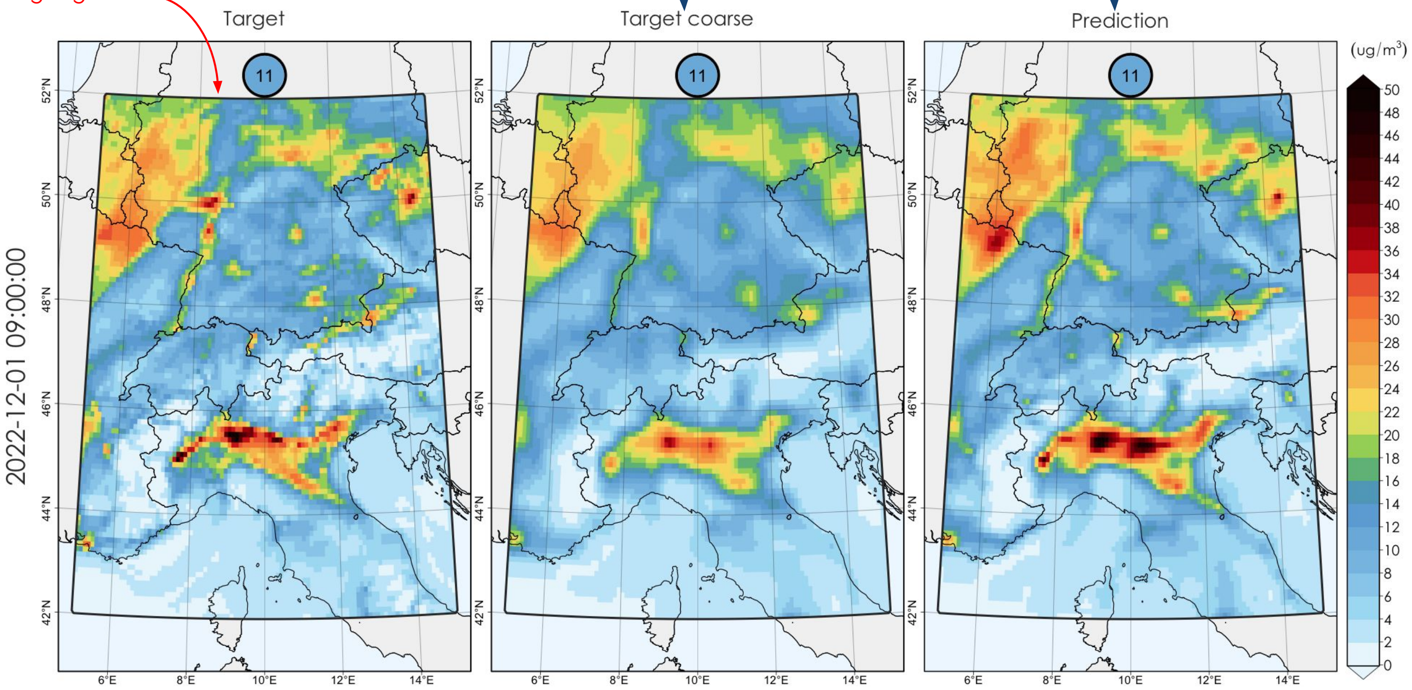
CNN
S(L64-L128-L256-L128-L64-D0.5-L1)
E(50)
LF(MAE)
LR(1e-04)
D(20221201)
T(1to24)
Total Time: 00:53:29, Epoch Time: 00:01:04
Min Train: 0.87, Min Test: 1.24



Testing region!

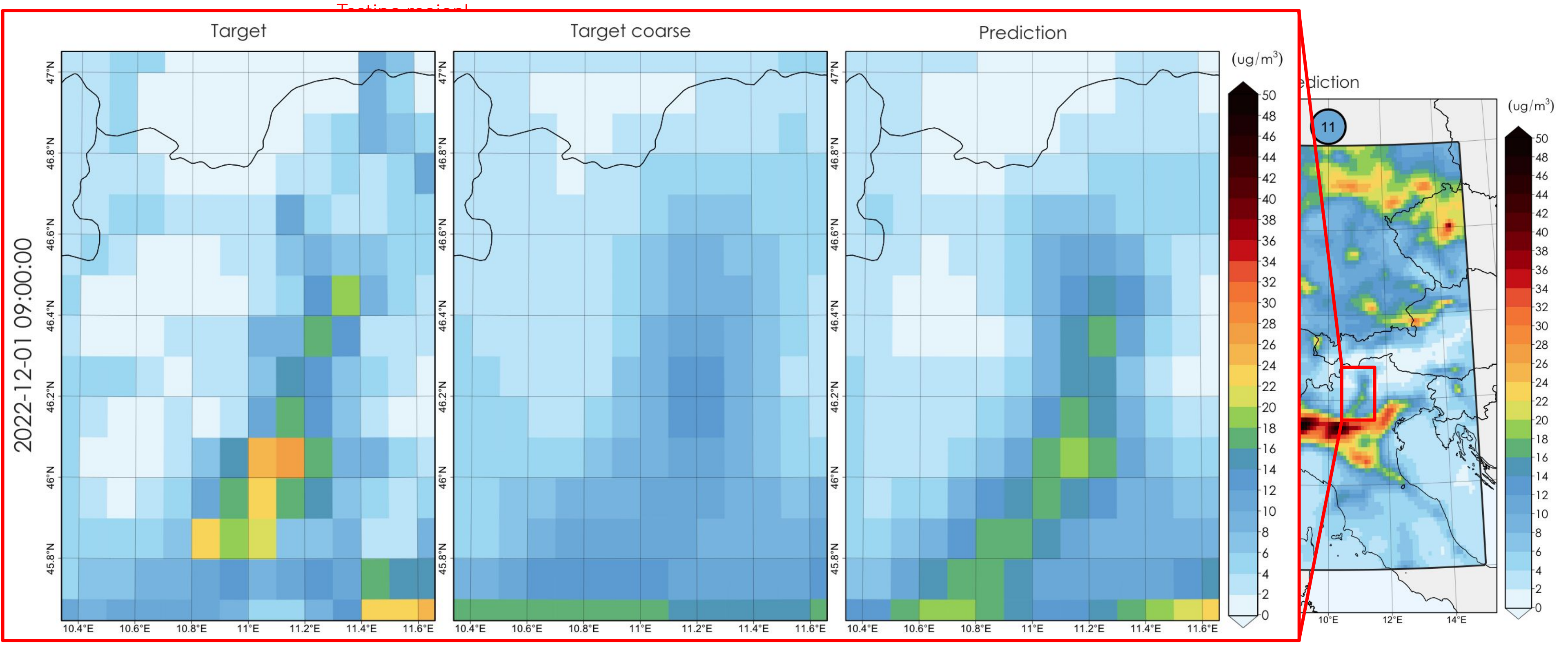
This is used as input

ML result



Results: Prediction for a specific timestep (09:00 UTC)

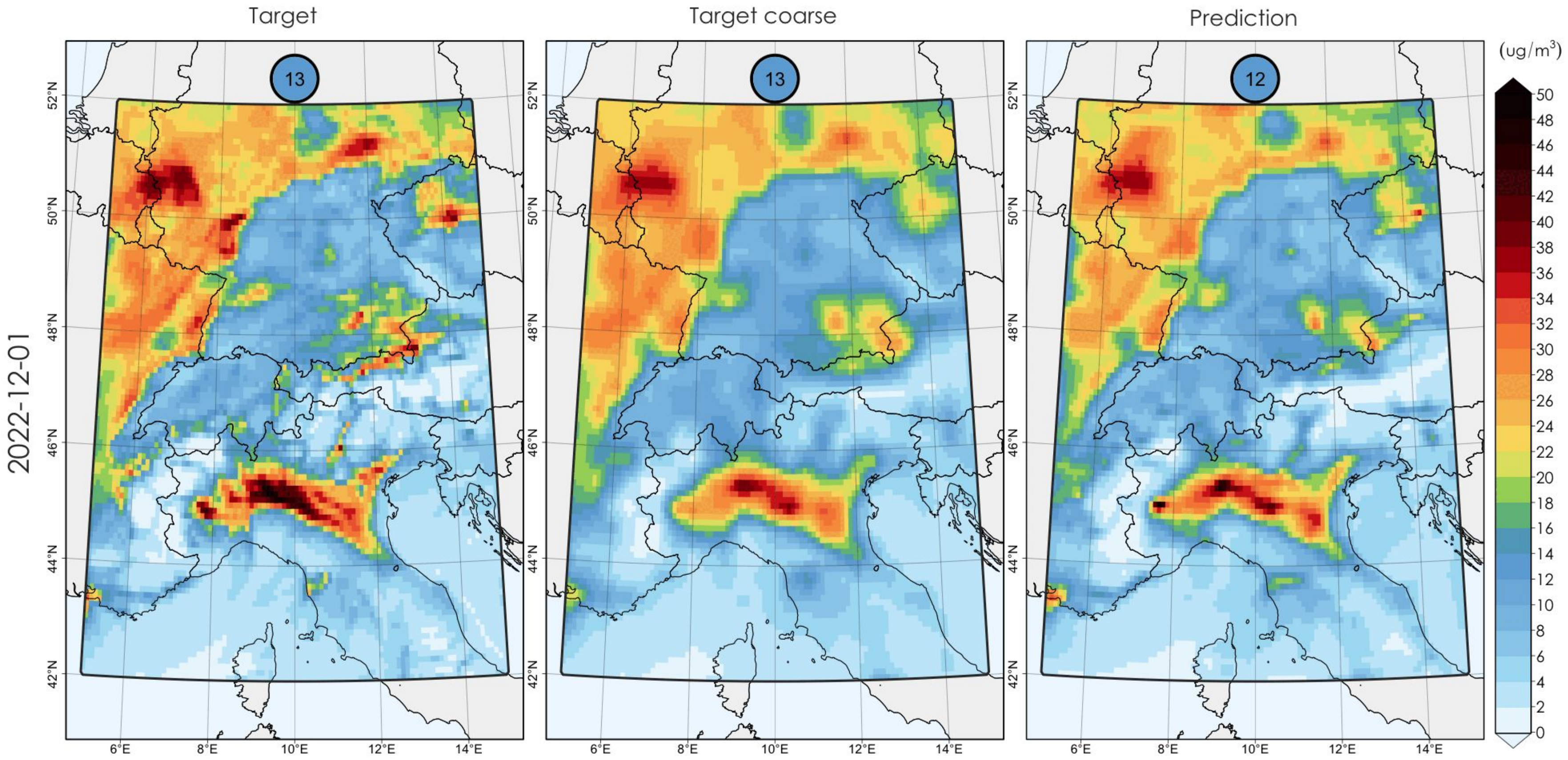
Results: Prediction for a specific timestep (09:00 UTC)



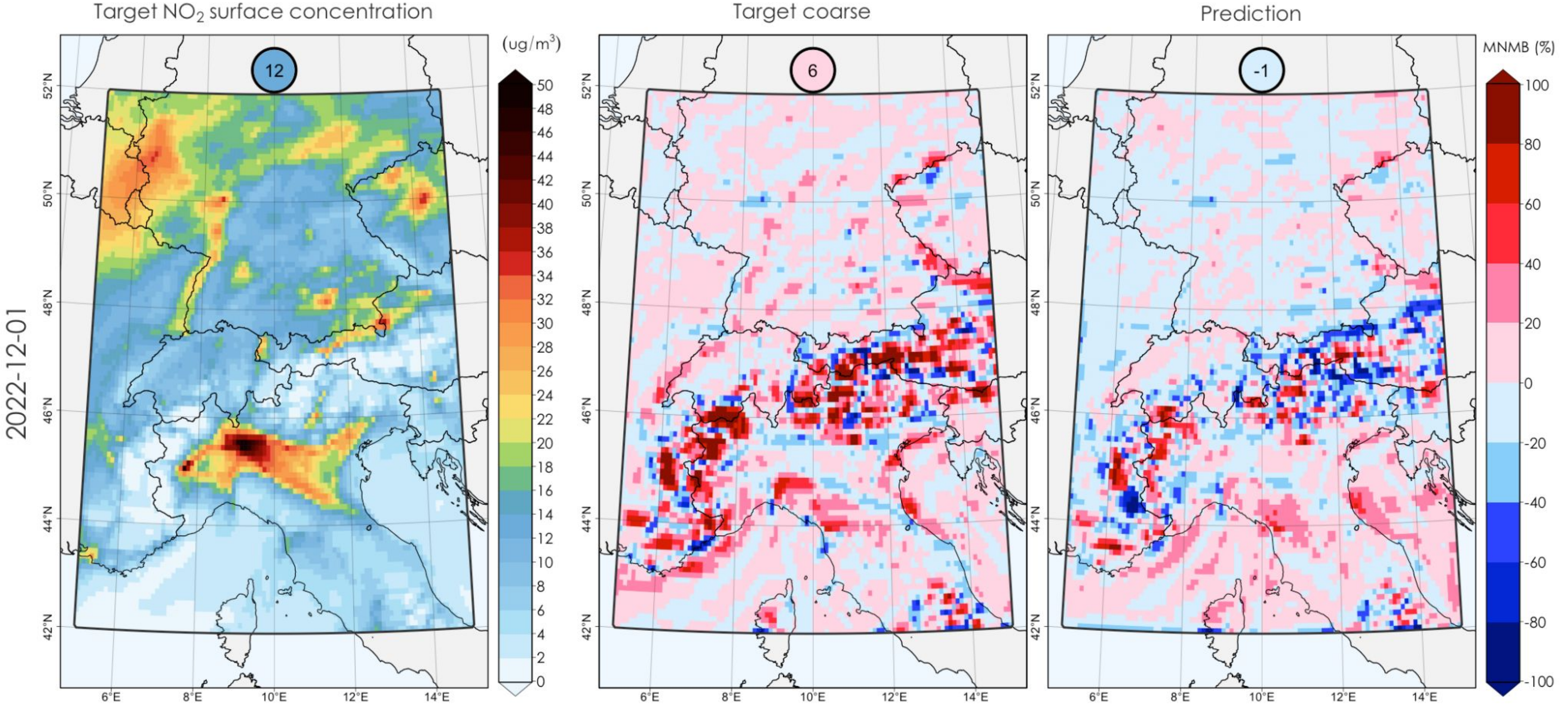
Results: Daily cycle is preserved

Spatial Similarity Index Structure (Wang et al. 2014)

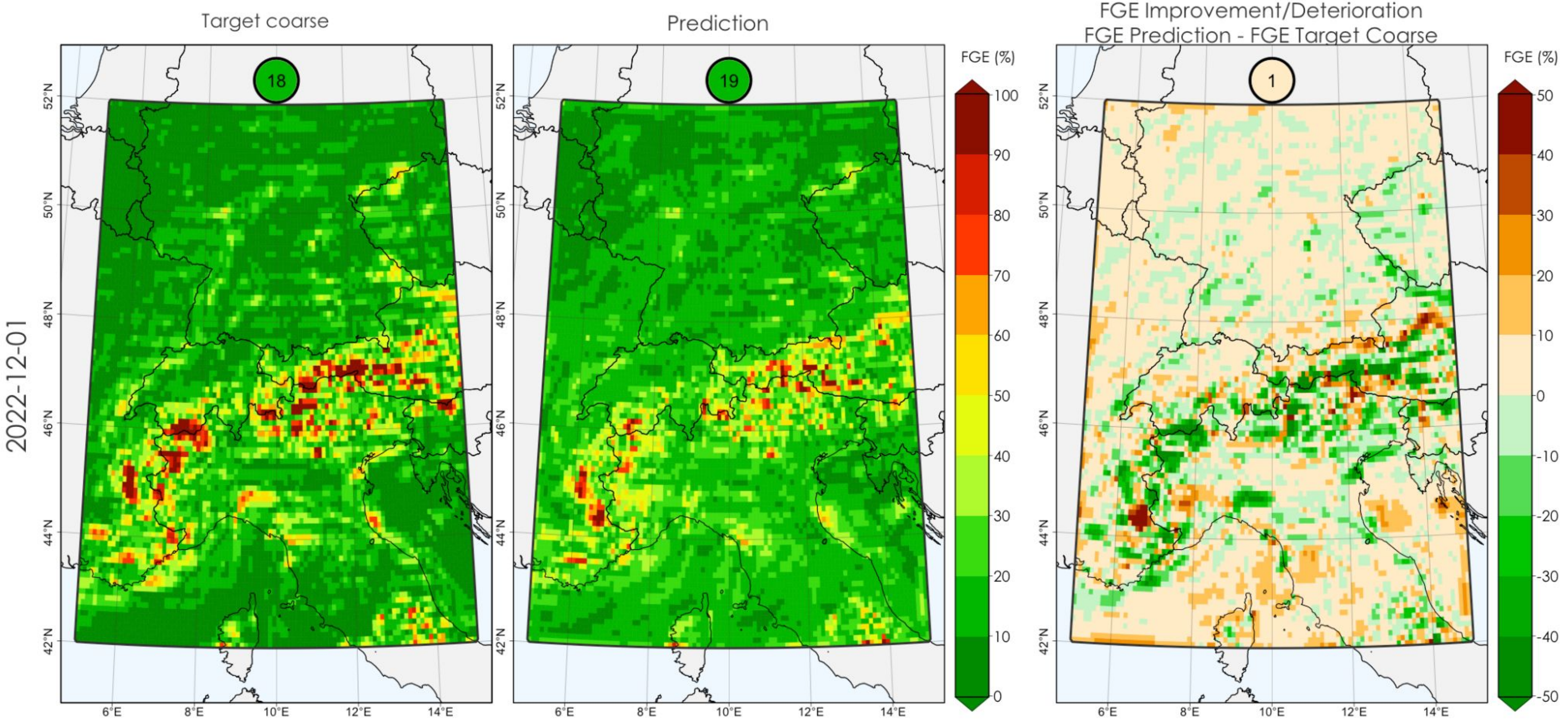
```
Target Coarse = 0.966
Prediction    = 0.968
```



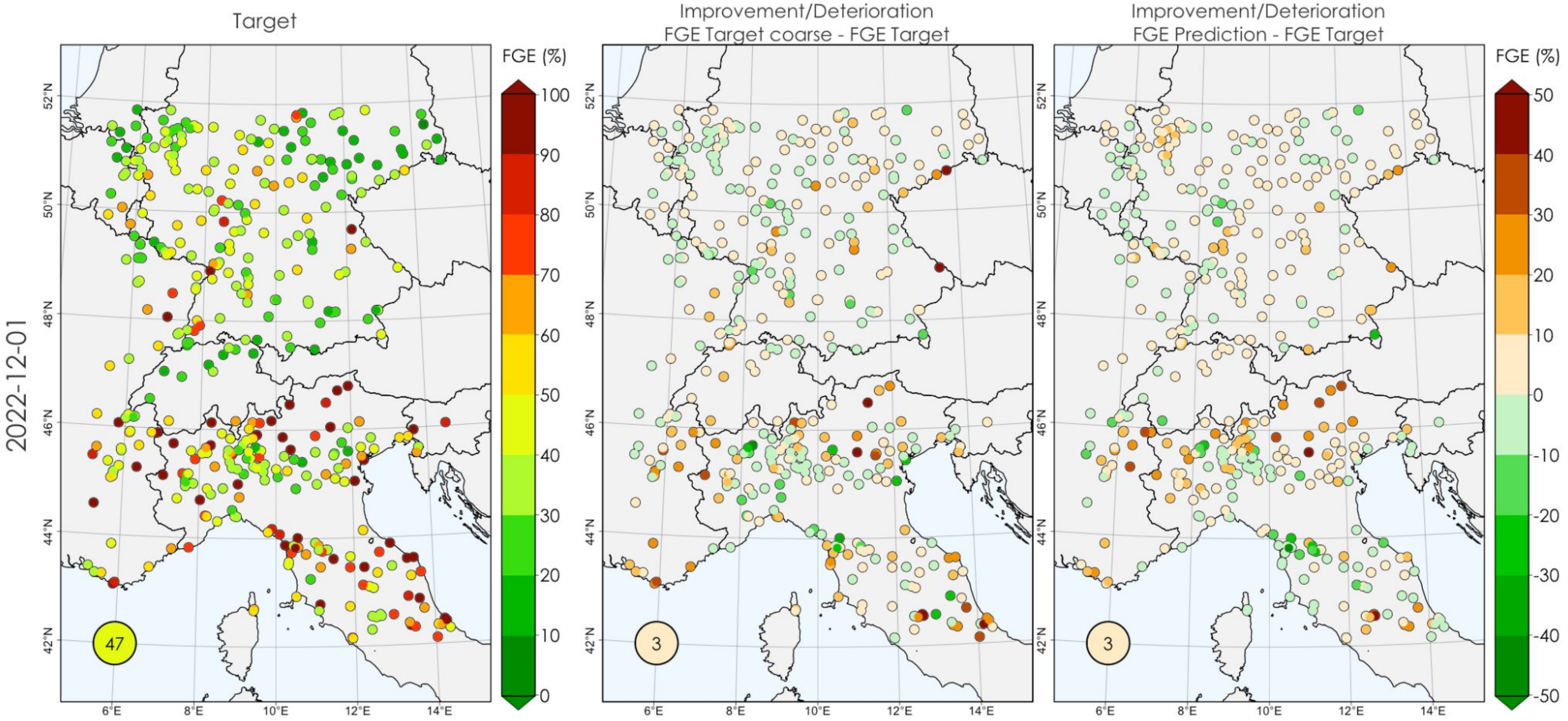
Results: Evaluation statistics (MNMB) → $\frac{1}{N}\sum_{i=1}^{N}\frac{M_i - O_i}{O_i + M_i}$



Results: Evaluation statistics (FGE) → $\frac{1}{N}\sum_{i=1}^{N}\frac{|M_i - O_i|}{O_i + M_i}$



Results: Evaluation using EEA observations (FGE)



Conclusions & Future plans

Trained a CNN to spatially downscale CAMS NO_2 surface concentration from 40 km to 10 km.

Lessons Learned

- Successfully gained some high resolution features, through not everywhere.
- No improvement compared to EEA stations. Alternative set up to consider surface observations (?)

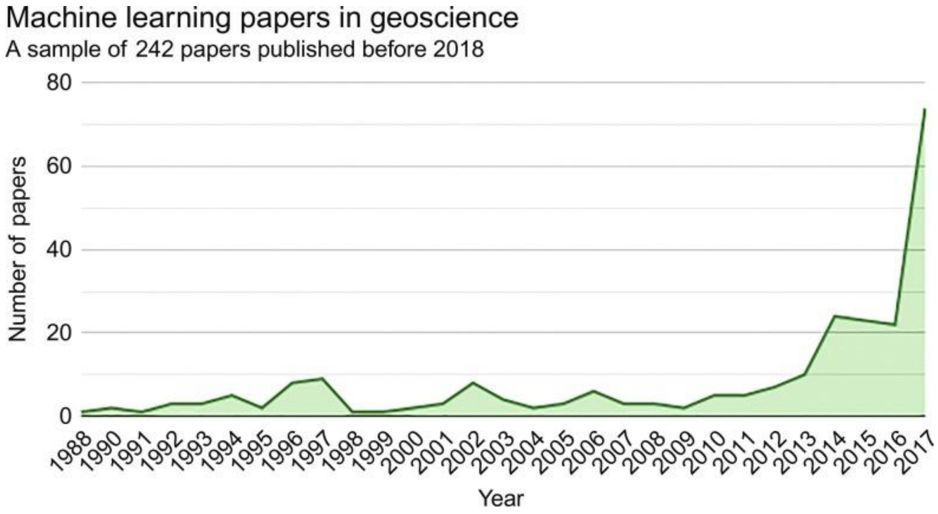
Future plans

- Transition to a GPU framework (train on more days, on other CAMSr model, cross-validation).
- Input importance analysis (e.g. Shapley Additive exPlanations).
- Use surface (EEA) or satellite observations (e.g. TROPOMI NO2 tropospheric column).
- Test the prediction of other species (e.g. PM2.5).
- Explore other scoring options that prioritize more the gain of fine scale details.
- Apply the trained model using CAMS global to a region outside of Europe.

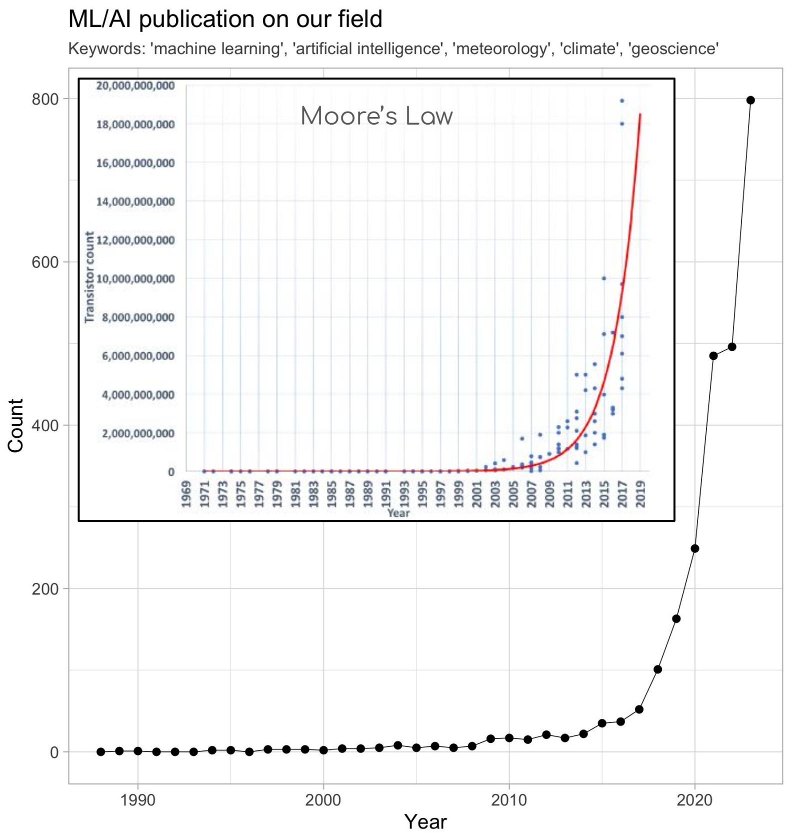
Backup slides

Machine Learning in Geoscience

An exponential increasing trend



Jesper Sören Dramsch 2020, Advances in Geophysics
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7500415/



Source: Google Scholar
thanos.tsikerdekis@knmi.nl

Data

Input, output and pre-processing of data

- **TARGET:** NO_2 Surface concentration in 10km
- **OUTPUT:** Predicting NO_2 Surface concentration in 10km
- **INPUT :**
  - ECMWF HRES Forecast — Meteorology (**10km**): Visibility, Specific Humidity at 2m, Temperature at 2m, Mean Sea Pressure, Boundary Layer Height, Total Cloud Cover, U and V components of the wind at 10m, Total Precipitation Rate
  - CAMS global — Emissions (**10km**): For NO_2 from 12 different sectors
  - Land (**10km**): Topography and Road Density
  - CAMS regional Interpolated — Concentration(**40km**): NO_2, NO, O_3, PM2.5, PM10 and SO_2

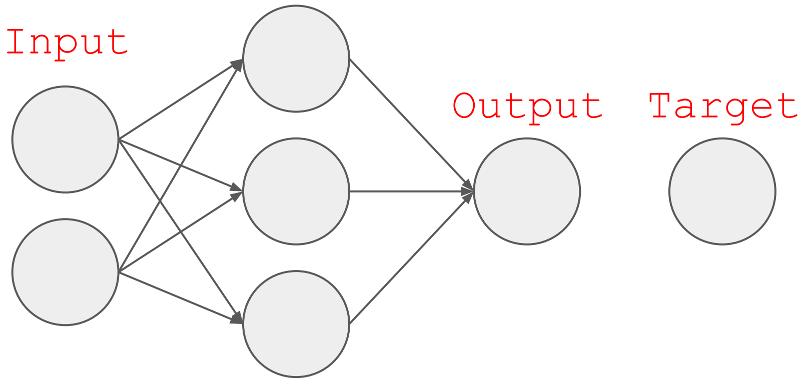
*** Temporal frequency of each variable differs from hourly to constant in time.

*** Normalize input before training

A simplified CNN structure

CNN (Convolutional Neural Network)

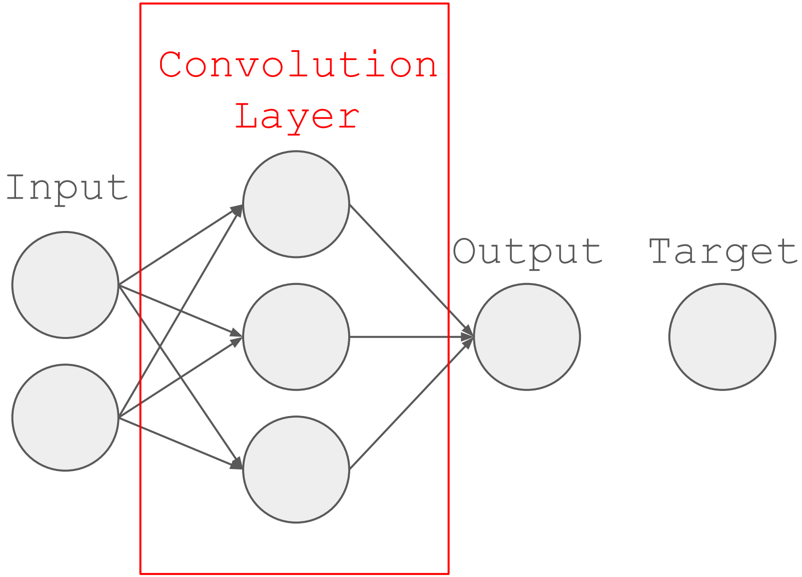
A CNN is a deep learning architecture designed specifically for processing structured grid-like data, such as images or spatial data. It combines several Input in order to predict (Output) a Target. The Input, Output and Target are pictures or model domains of N x N pixels or grid cells respectively.



A simplified CNN structure

Convolution Layer

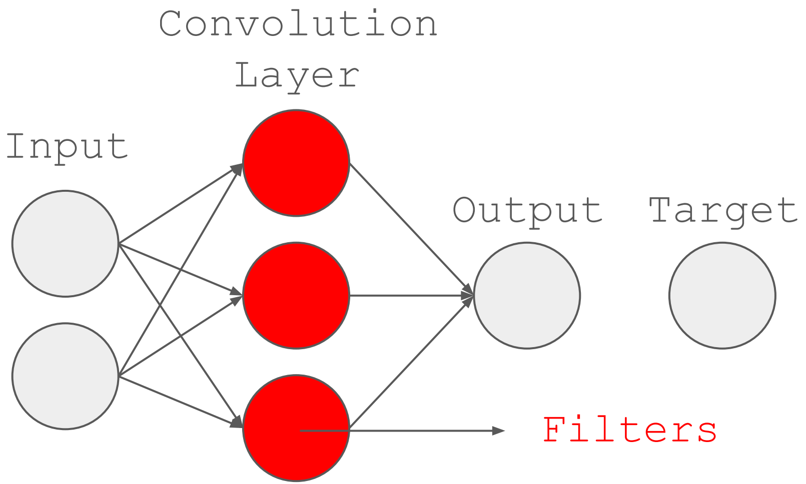
Convolutions extract features from the input data, each using filters to detect patterns.



A simplified CNN structure

Filter

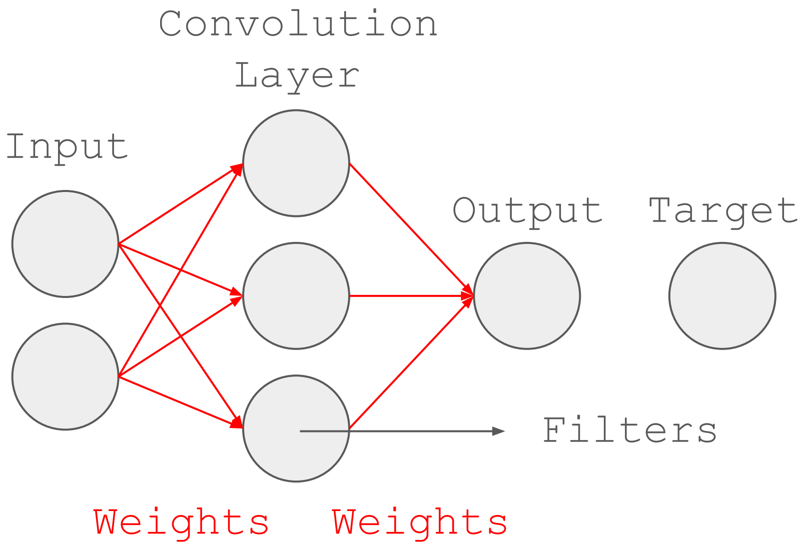
Each filter is a small-sized matrix of composed by combining different inputs and learnable weights.



A simplified CNN structure

Weights

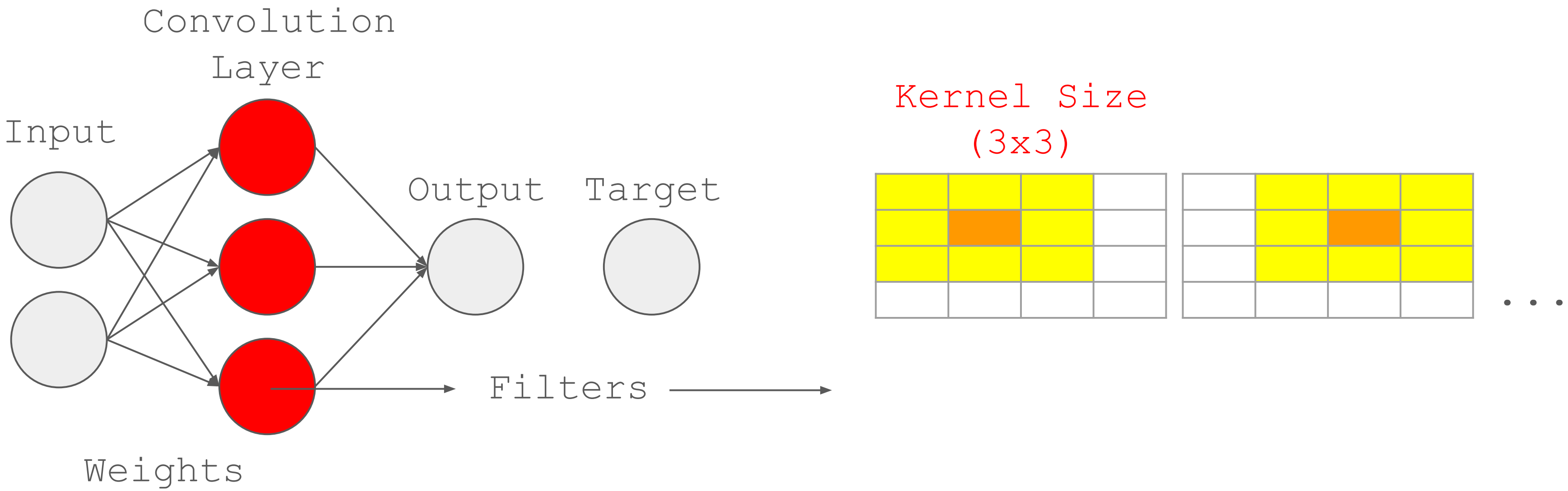
Parameters that are adjusted during training to minimize the difference between predicted and actual values, typically through techniques like backpropagation or gradient descent.



A simplified CNN structure

Kernel size

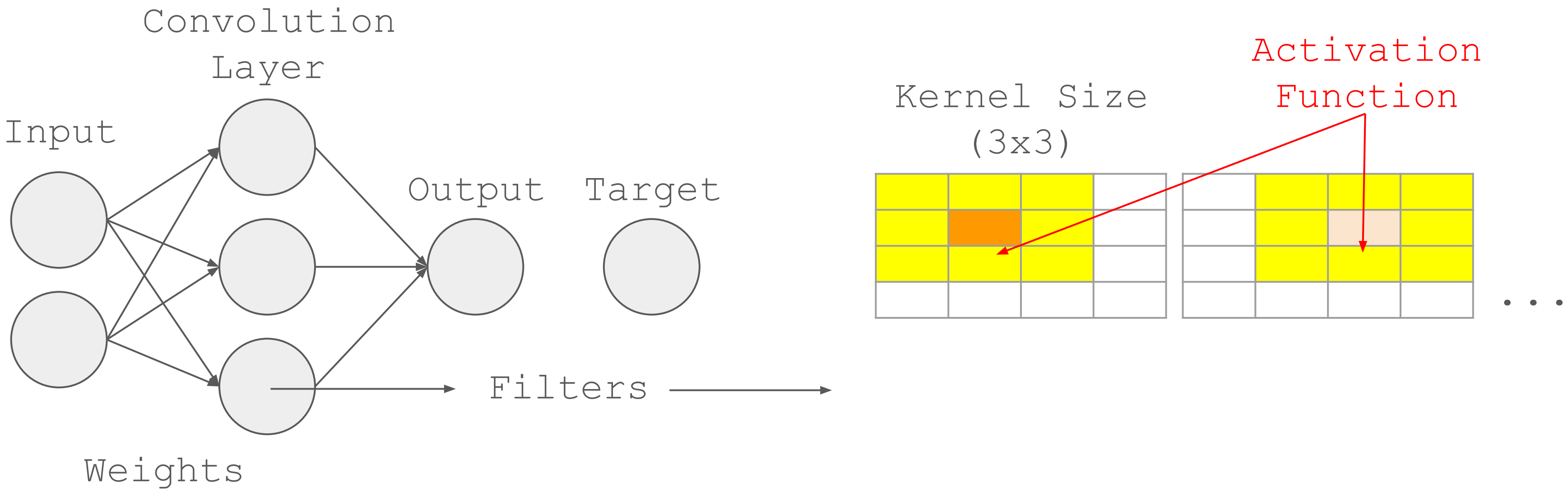
Determines the spatial extent of the features to be detected. Larger kernel sizes (e.g. 9x9) capture broader features, while smaller ones (3x3) capture finer details, influencing the scale of patterns the model can detect.



A simplified CNN structure

Activation function

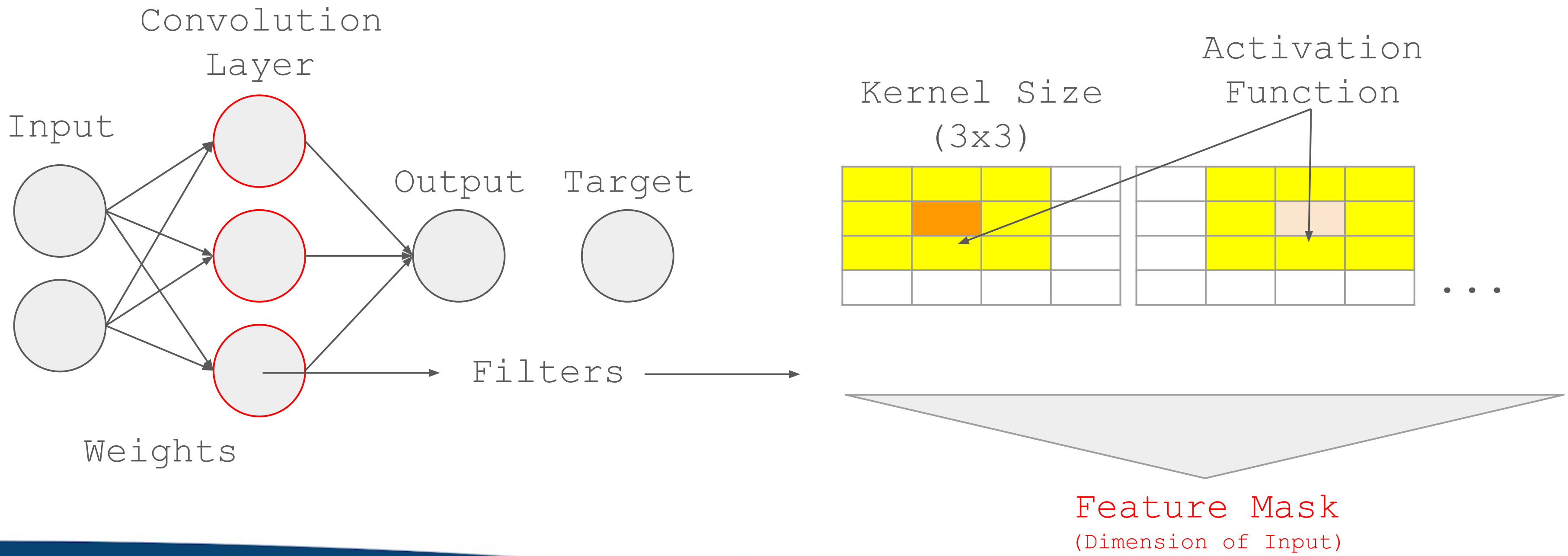
Apply in each filter (element-wise) a ReLU function that sets negative values to zero and leaves positive values unchanged. This introduces non-linearity into the network.



A simplified CNN structure

Feature Mask

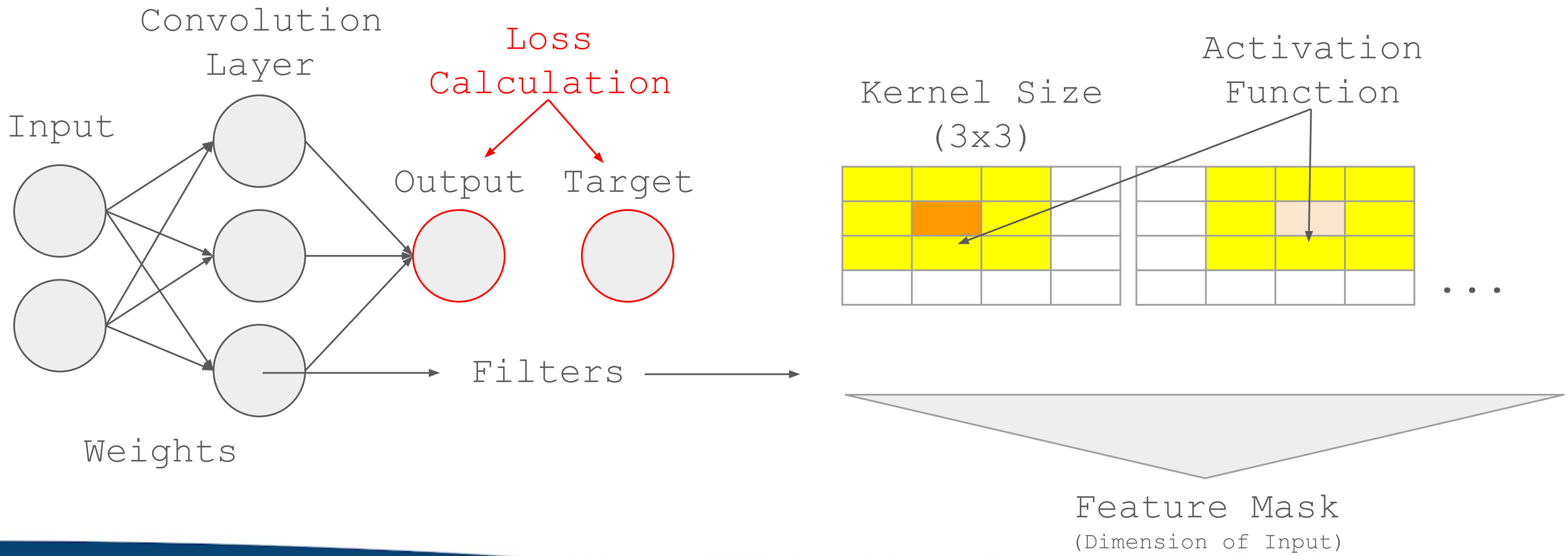
The values in the feature map represent the outputs of the filter across the entire input data



A simplified CNN structure

Loss Calculation

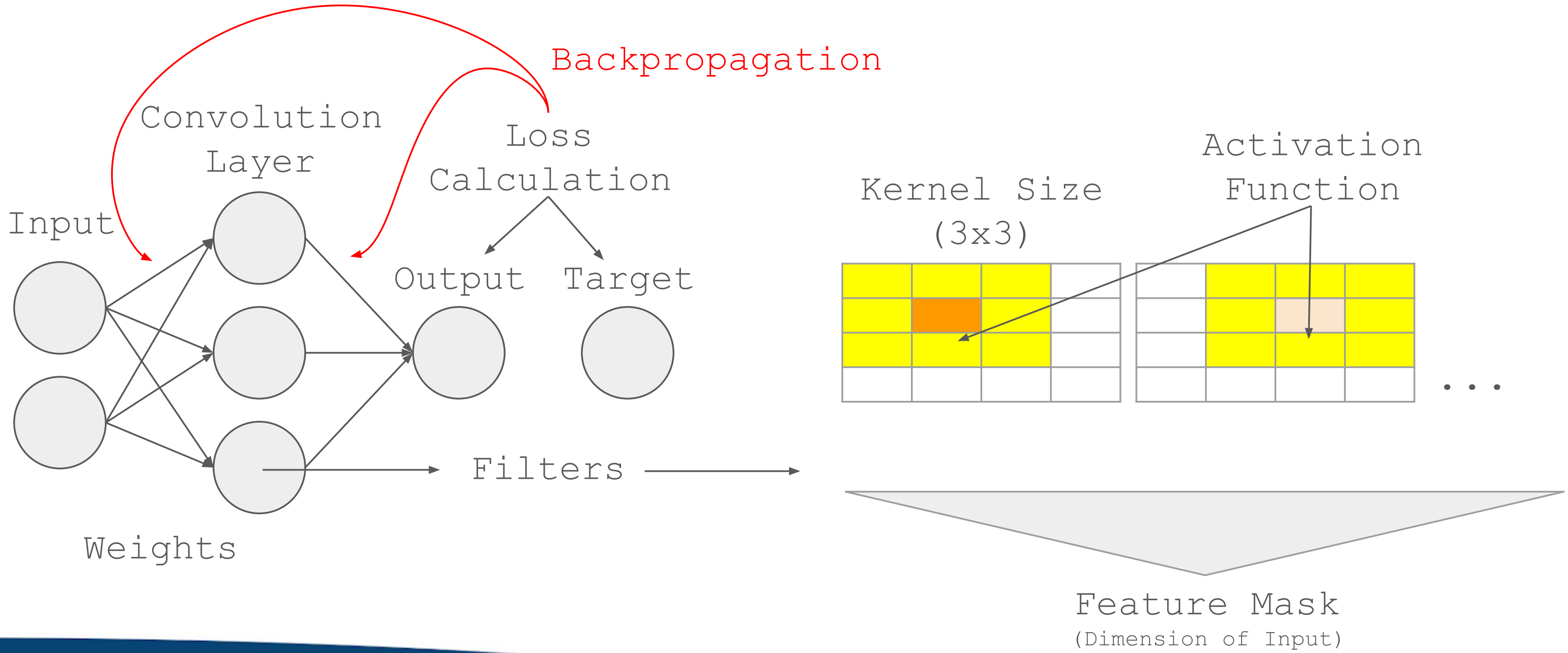
Loss between the Output (Prediction) and the Target based on a Loss Function (e.g. MSE, MAE etc).



A simplified CNN structure

Backpropagation

Computes the gradient of the loss with respect to the model's parameters backwards layer by layer.

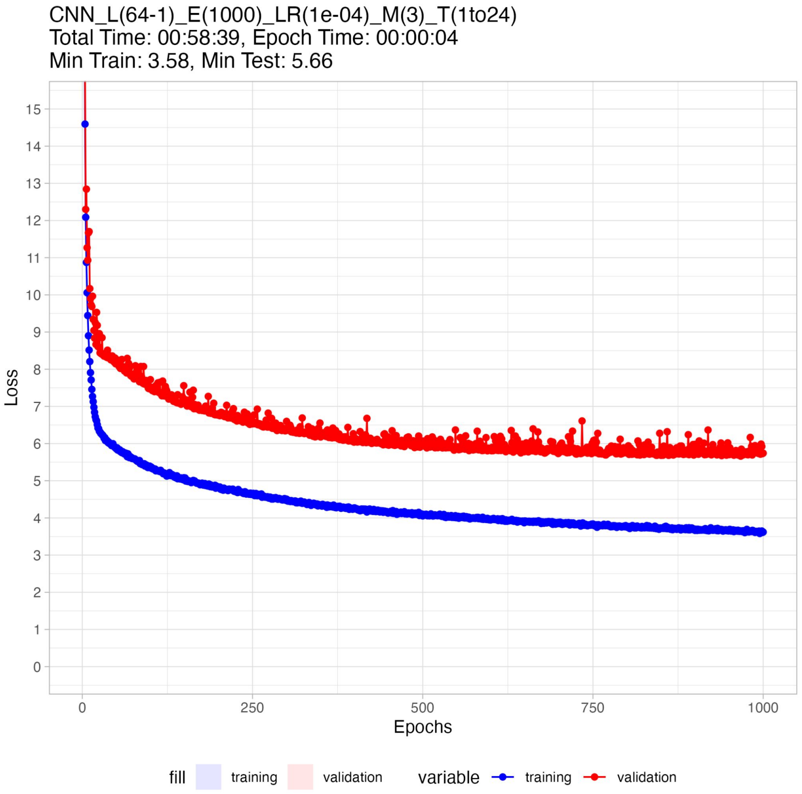


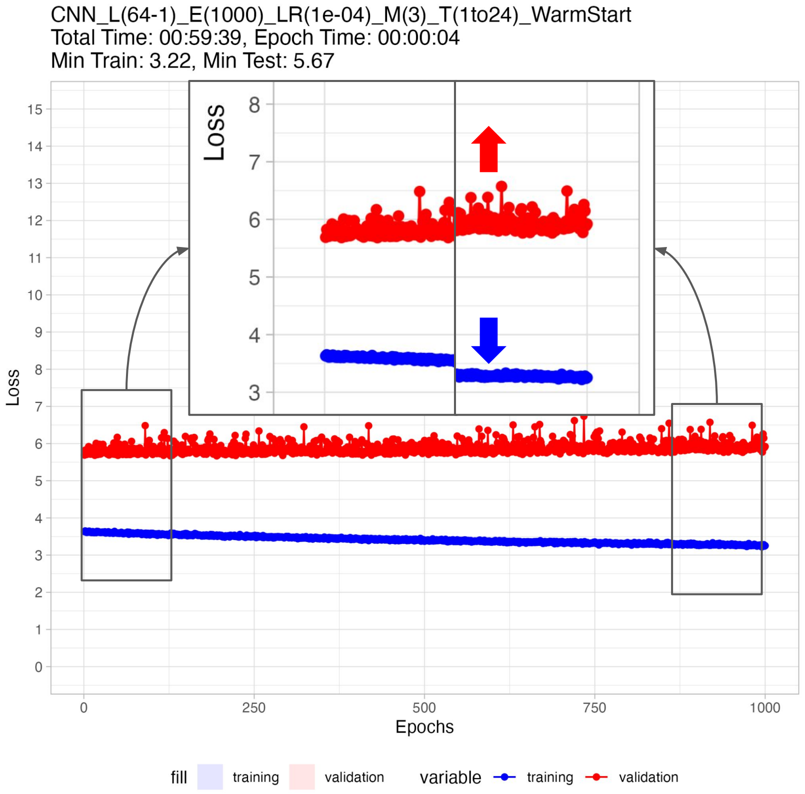
Experiments

Exp	CNN Structure	Epochs	Learning Rate	Train/Test Size	Loss Function	Train Min Loss	Test Min Loss
1	L64-L1	1000 + 1000	0.001	312 / 48	MSE	3.22	5.67
2	L64-L1	30	0.001	312 / 48	MSE	6.23	8.39
3	L64-L128-L64-L1	30	0.001	312 / 48	MSE	4.64	6.64
4	L64-L128-L256-L128-L64-L1	30	0.001	312 / 48	MSE	3.78	6.06
5	L64-L128-L256-L128-L64-D0.5-L1	30	0.001	312 / 48	MSE	4.08	5.97
6	L64-L128-L256-L128-L64-D0.5-L1	30	0.001	312 / 48	HL	0.68	0.96
7	L64-L128-L256-L128-L64-D0.5-L1	30	0.001	312 / 48	MAE	1.00	1.30
8	L256K5-L32K1-L32K3-L256K1-L1K9	30	0.001	312 / 48	MAE	1.04	1.36
9	L64-L128-L256-L128-L64-D0.5-L1	100	0.001	312 / 48	MAE	0.70	1.25
10	L64-L128-L256-L128-L64-D0.5-L1	100	0.001	936 / 144	MAE	0.56	1.29
11	L64-L128-L256-L128-L64-D0.5-L1_OBS	30	0.001	312 / 48	MAE	1.01	1.33

Should I train a CNN forever = many many epochs?

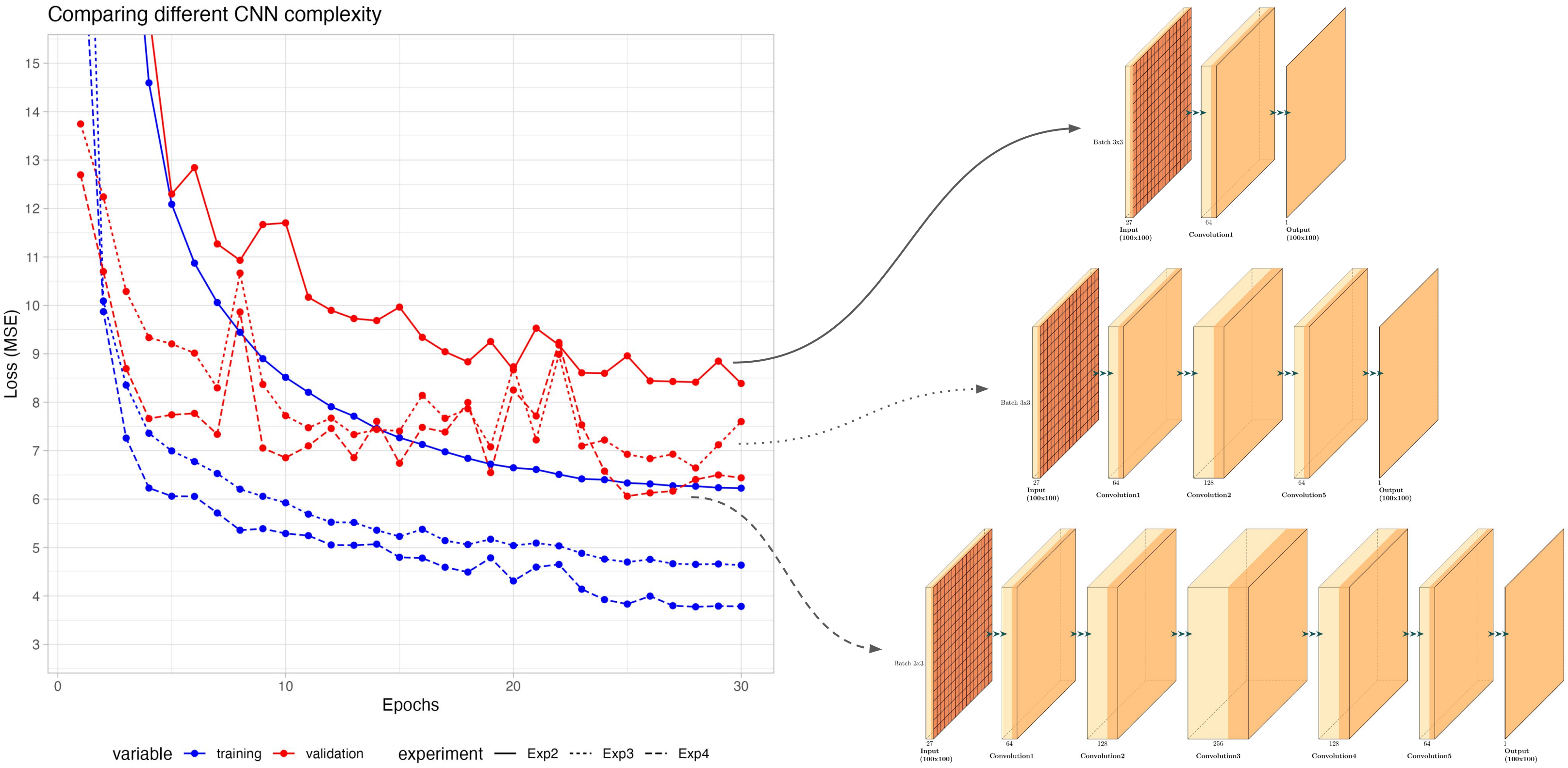
Does not make a difference after a point, especially for the testing data + you are overfitting…





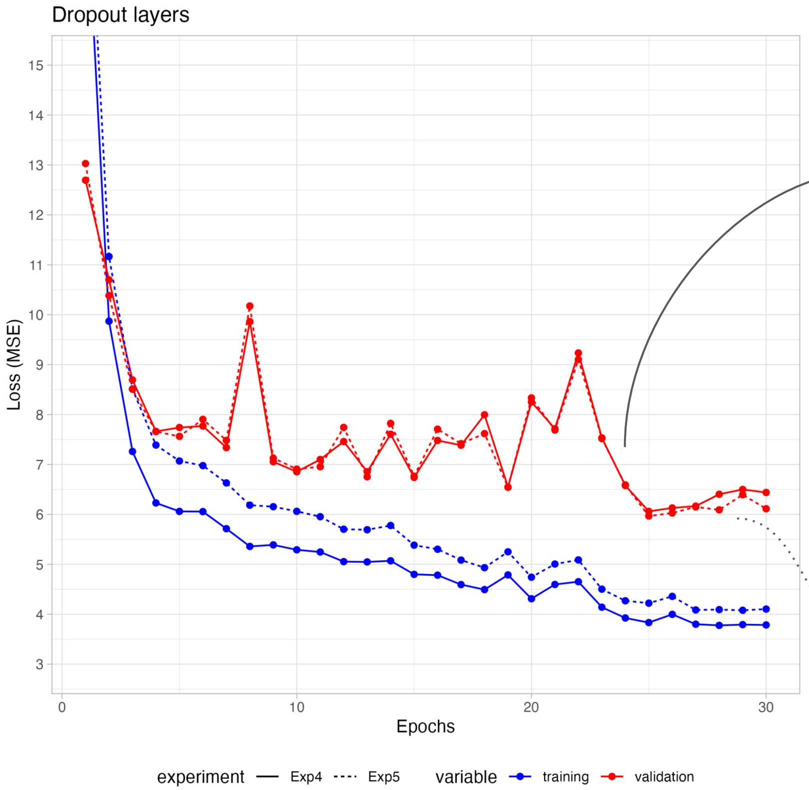
Should I use a more complex CNN?

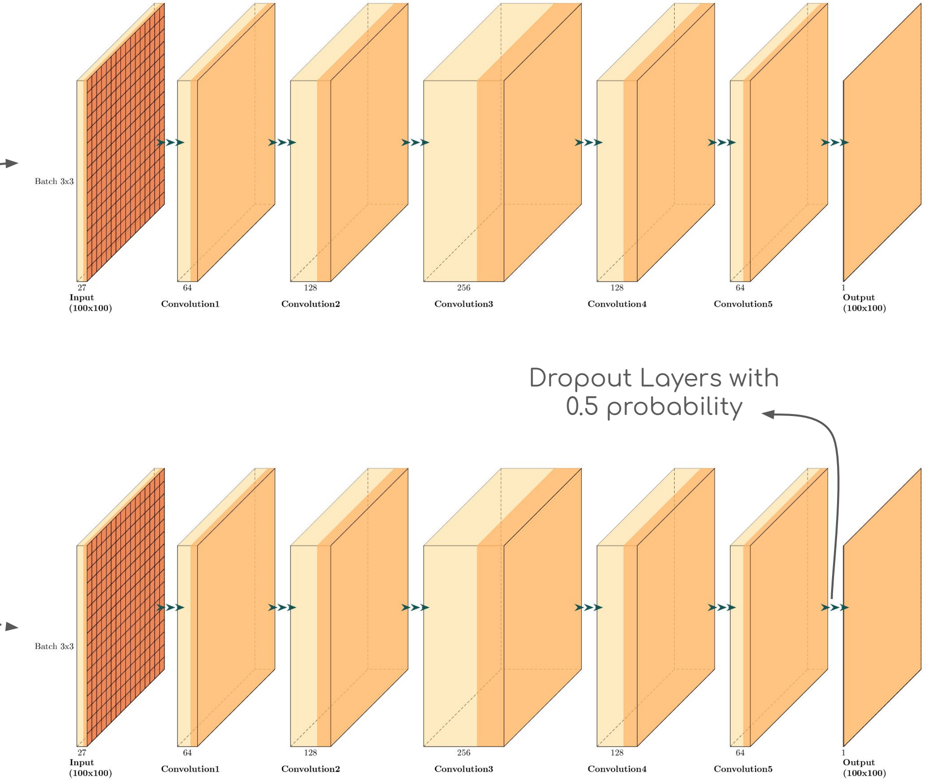
Yes, since it can learn more complex connection between the input



Can I somehow restrain overfitting?

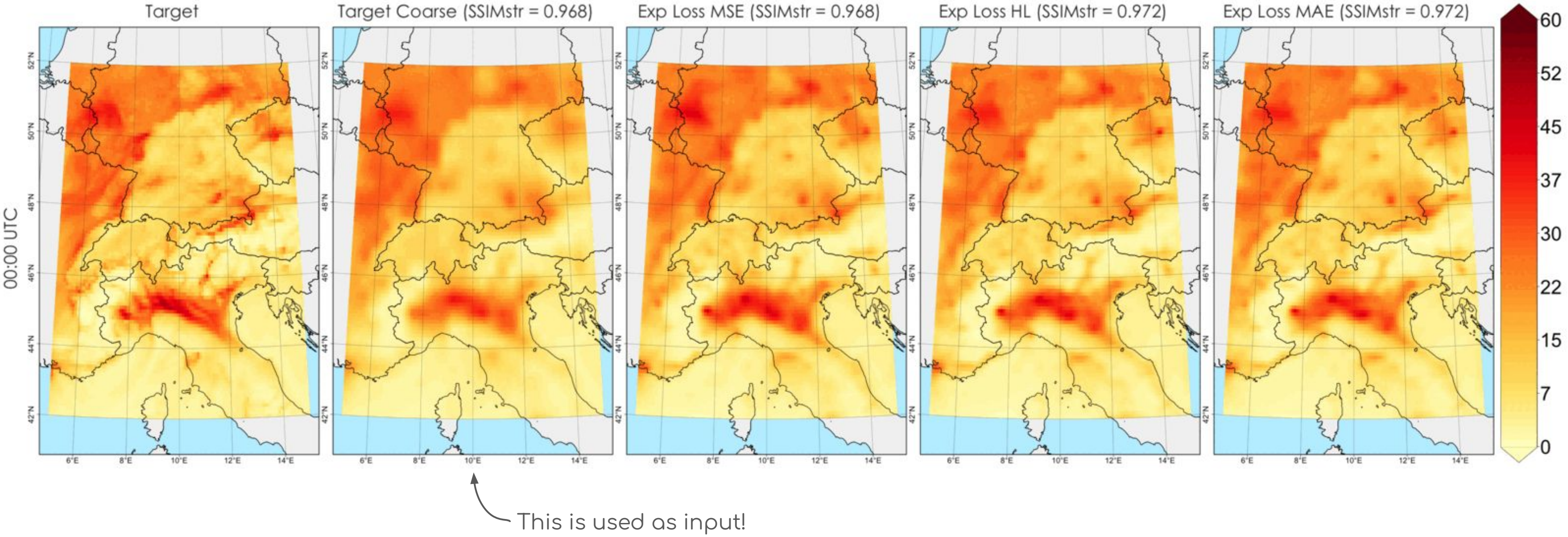
Dropout layer: Small changes but clearly worse training and better testing loss





What about other timesteps? Is the diurnal cycle correct?

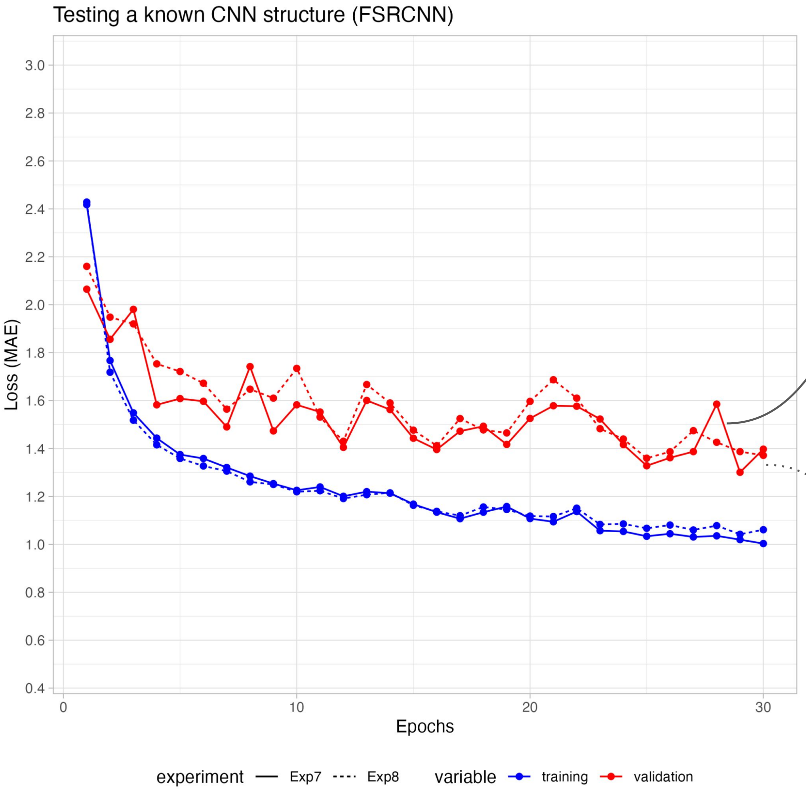
Since target coarse is used as input, the diurnal cycle is retained.

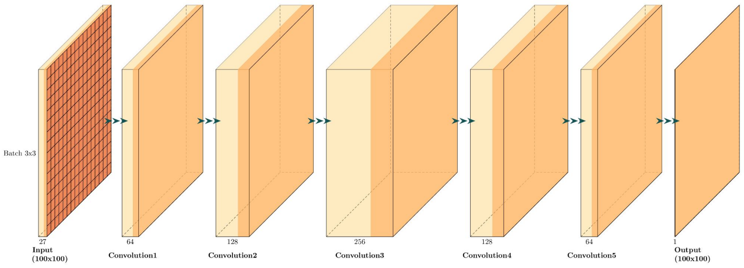


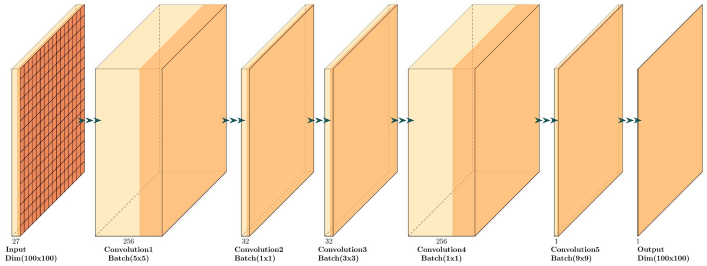
Testing other CNN structures: FSRCNN structure

At the moment does not provide much better results

Fast Super-Resolution Convolutional Neural Network

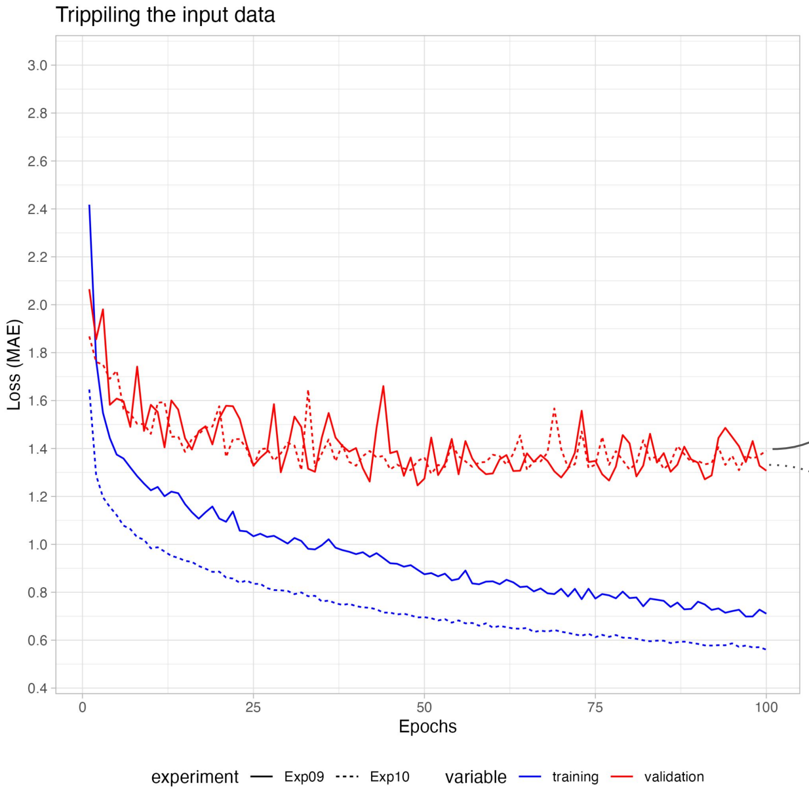






Should I use more data for training?

At this state (current inputs), tripling inputs provide the same testing loss and lower training loss

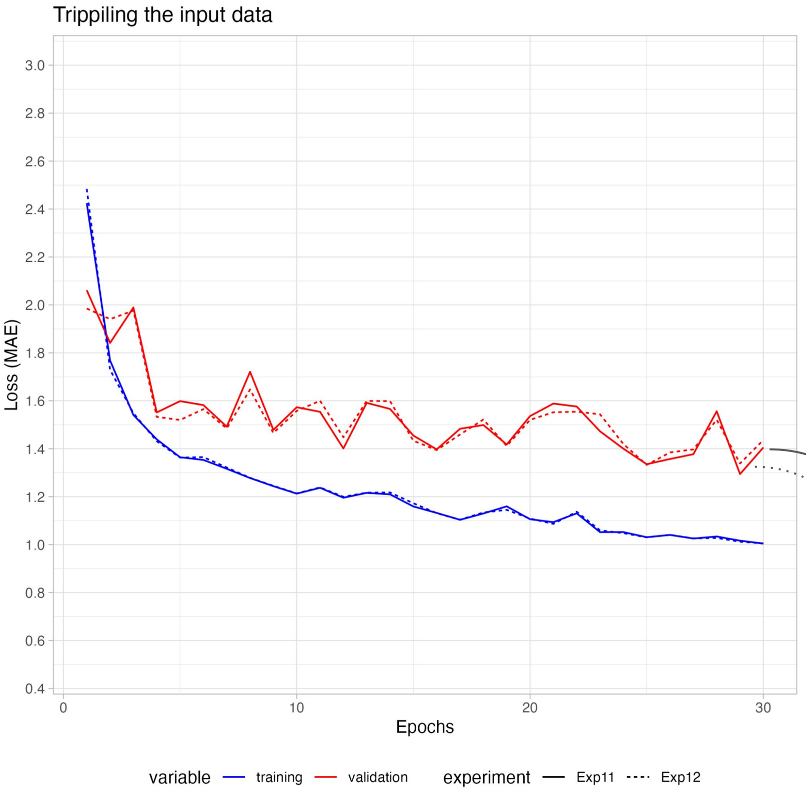


1 day * 24 hours * 13 regions = 312 cases

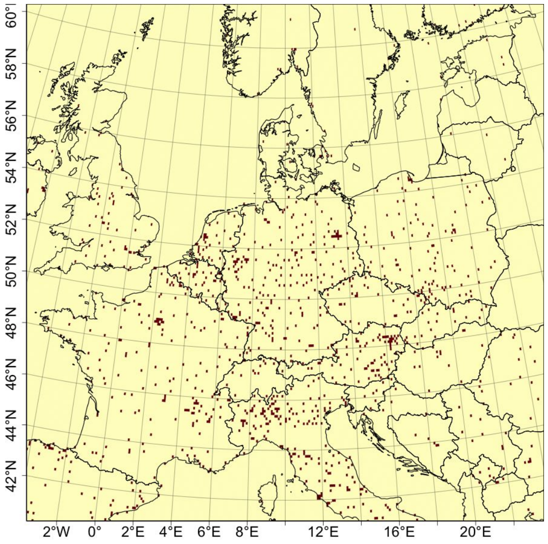
3 days * 24 hours * 13 regions = 936 cases

Adding "observations" as input?

Using the dense EEA network of stations over Europe as input is not improving results



Model grid cells (10km) that contain an EEA station



Not using "station" observations as input

Using "station" observations as input

Results: Evaluation using EEA observations (MNMB)