



Article

End-to-End Prediction of Lightning Events from Geostationary Satellite Images

Sebastian Brodehl* ¹, Richard Müller ², Elmar Schömer ¹, Peter Spichtinger ³ and Michael Wand ¹¹ Johannes Gutenberg University Mainz, Institute of Computer Science, Mainz, Germany² German Weather Service, Offenbach, Germany³ Johannes Gutenberg University Mainz, Institute for Atmospheric Physics, Mainz, Germany

* Correspondence: (brodehl@uni-mainz.de)

Abstract: While thunderstorms can pose severe risks to property and life, forecasting remains challenging, even at short lead times, as these often arise in meta-stable atmospheric conditions. In this paper, we examine the question of how well we could perform short-term (up to 180min) forecasts using exclusively multi-spectral satellite images and past lightning events as data. We employ representation learning based on deep convolutional neural networks in an “end-to-end” fashion. Here, a crucial problem is handling the imbalance of the positive and negative classes appropriately in order to be able to obtain predictive results (which is not addressed by many previous machine-learning-based approaches). The resulting network outperforms previous methods based on physically-based features and optical flow methods (similar to operational prediction models) and generalizes across different years. A closer examination of the classifier performance over time and under masking of input data indicates that the learned model actually draws most information from structures in the visible spectrum, with infrared imaging sustaining some classification performance during the night.

Keywords: neural networks; satellite images; class imbalance; feature attribution; lightning prediction; nowcasting; short-term forecasts; machine learning; meteorology

1. Introduction

Thunder and lightning are violent atmospheric events that must have impressed humans since prehistoric times. Lightning itself as well as strong winds and precipitation, hail, or even down bursts or tornadoes that might accompany thunderstorms are not only impressive appearances in satellite images, but can pose significant risk to life and property. Even in modern times, fatalities and severe damages from thunderstorms are still occurring at unfortunate rates, and many commercial operations, such as airports or public outdoor events (sports, music, gatherings), rely on risks assessments and prediction of lightning in order to operate safely. In this context, accurate forecasts for the next few hours are particularly important. Thunderstorms are by definition associated with lightning, which easily can be *detected* by triangulation using multiple radio wave antennas.

Unfortunately, the *prediction* of thunderstorms and lightning (Cb) is a very difficult problem: The development of atmosphere’s electric field with locally strong charges of different signs is crucially dependent on cloud processes. Charge generation is based on the collision of ice particles in the presence of super-cooled water droplets. Many different processes of this kind are proposed for charge generation, see, e.g. [1]; however, at the moment it is not clear which mechanism might be dominant. Nevertheless, the collisions are enhanced in strong, localized atmospheric updrafts that are typically formed by frontal movements or spontaneous convective events induced by heat in situations of unstable atmospheric layering. In particular the latter effects are highly sensitive to small scale perturbations and thus hard to predict.

Although significant improvements in Cb forecasting have been achieved with numerical weather prediction (NWP) [2,3], accurate forecasts of Cb location and strength are still a major challenge. Hence, nowcasting methods are typically used to issue warnings with short lead times (in the range of a up to a few hours) [4]. An early overview of nowcasting



Citation: Brodehl, S.; Müller, R.; Schömer, E.; Spichtinger, P.; Wand, M. End-to-End Prediction of Lightning Events from Geostationary Satellite Images. *Preprints* **2022**, *1*, 0. <https://doi.org/>

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

of (severe) weather phenomena is given in [5]. Nowcasting can in general be viewed as a special type of a video prediction problem. Thus, methods proposed for e.g. video frame prediction in natural videos are applicable to nowcasting and are related, see [6].

Radar-based methods are often used for regional nowcasting, in detail the (Doppler-) radar-based remote sensing of (heavy) precipitation events and their movement, depending on the method in combination with lightning data [7]. The forecast is typically based on two subsequent images and extrapolation of the movement of the observed storm cells, e.g by optical flow methods [8] or tracking methods [9]. This limits the ability to predict newly forming or decaying of cells and limits long term precision. For example, operational systems such as NowCastMIX [9] struggle with predictions for lead times beyond two hours. That is why for longer lead times NWP models are used.

An alternative source of information are images recorded by geostationary satellites. They cover larger regions than methods based on weather radar. For example, the SEVIRI instrument onboard Meteosat's second generation (MSG) platform provides data in twelve spectral bands with a temporal resolution of 15 minutes and a spatial resolution with 3×3 sub satellite point, which corresponds to 0.05×0.03 in Central Europe. This data can be used like radar data to determine atmospheric motion vectors (AMVs) using optical flow and to provide information about the brightness temperature of clouds (BT), which can be used to indicate thunderstorms. For example, the physically motivated usage of the "sandwich" method, which is based on the BT difference between the 6.3 water vapour (WV) channel and the IR window channel [10], or alternatively the 7.3 WV channel [11]. Yet, Cbs are defined by lightning. Hence, in order to improve the accuracy of the detection of Cbs, which is the basis for the nowcasting, it is recommended to use lightning data in addition. For example, Cintineo *et al.* [12] use the Advanced Baseline Imager (ABI) and flash-extent density (FED) from the Geostationary Lightning Mapper (GLM) on board GOES-16, as well as satellite- and solar zenith angles and geo-coordinates to predict *intense convection*. In Müller *et al.* [13] satellite based information is combined with lightning data from the Vaisala Global Lightning Detection Network (GLD360) and information from NWP. Also combination of satellite and radar information are applied, for example, Lee *et al.* [14] make use of five temporal images from two ABI channels (02, 14) to predict convective regions, derived from Multi-Radar Multi-Sensor (MRMS) precipitation types.

However, it remains unclear whether the physical nowcasting methods are capable of making the best possible use of the available information in the satellite image and lightning data. As discussed before, current physical approaches, e.g. [13,15,16], do not take into account the decay or development of (new) cells. Corresponding information may be hidden in the input data and unused. The visual analysis of thunderstorms based on lightning data and satellite imagery suggests that there may be more information in the data that can be extracted. Thus, an obvious alternative is to resort to machine learning [17,18] and in particular deep representation learning techniques [12,19], which have, in recent years, become able to automatically build highly predictive statistical models even from data with extremely complex statistical dependency structure, such as differentiating different breeds of dogs in general photography [20–22].

Our study follows this approach: We predict lightning events from image sequences containing satellite imagery and a map of recent lightning events through an image-to-image translation performed by (a variant of) a convolutional U-Net [23], which could be considered a "canonical" network approach to image-segmentation and translation tasks. The use of machine learning methods, and specifically deep neural networks, have been studied previously in the literature [24–28]. Specifically, U-Nets have been used in related studies for precipitation forecasting [29,30]. Regarding convection, Agrawal *et al.* [31] published a study, which uses radar images (2min, 1km grid) to predict precipitation with a lead time up to 6h using a (residual) U-Net. Similarly, Shi *et al.* [32] train a Trajectory GRU model to predict precipitation based on radar images (6min, ca. 1km grid). Han *et al.* [33] propose a machine learning method to detect convective initiation (CI) from geostationary satellite data training a random forest classifier on hand-crafted features. CI is derived

from radar data. Mostajabi *et al.* [34] developed a model based on four commonly available surface weather variables (air pressure at station level (QFE), air temperature, relative humidity, and wind speed) to predict lightning events for lead times up to 30 training a decision tree.

However, lightning prediction comes with the problem of heavy *class imbalance* which is a general challenge for classification with statistical machine learning: In many relevant scenarios, lightning events are rather rare; thus, a classifier that just predicts “no (nearby) lightning” for all outputs can easily reach accuracies close to 100%. Previous work has mitigated this issue by resampling in evenly balanced examples [17] or considering conditions with high prevalence [19], which then also assumes balanced base-rates when making predictions. In our paper, we develop a method to automatically balance class weights to optimize a deep learning classifier for high predictive power (such as high critical success index, CSI). This permits a simple “end-to-end” training with operationally meaningful predictions (i.e., using recent lightning and satellite observations, the classifier predicts future events with above-chance accuracy). To obtain an indicator of the quality achieved, our results are compared with those of an optical flow-based nowcasting method of the German Weather Service (DWD).

A second, and possibly even more interesting question is understanding which information actually contributes to making better predictions. Having a strong classifier that outperforms hand-engineered operational models by simple, automatic statistical learning opens up the opportunity to study this question: By withholding data at the learning stage and tracking the reduction in performance, we can attribute how much information the classifier was able to draw from these sources (formally as lower bound of how much statistical information about the event is contained in portions of the data). We conduct an experiment where we measure the variability of the prediction performance over different times of the day, and retrain classifiers for visible (including a $1.6\mu\text{m}$ near-infrared channel), infrared, and two water vapor bands.

Overall, our study makes two key contributions: (i) a simple method for end-to-end training of lightning events from image data that is both practical (applicable to data with real-world, skewed prior class frequencies) and accurate (outperforming state-of-the-art optical flow-based systems), and (ii) we obtain some novel insights of which sources of information are useful for making predictions by examining the performance characteristics of the learned classifier.

2. Materials and Methods

Our method performs statistical learning to learn a mapping that takes satellite images and measurements of the current lightning activity as input and predicts future lightning activity as output, with a specific lead time.

Satellite images: Formally, we define the geostationary satellite images as functions

$$S : \mathbb{R} \times [-79^\circ, 79^\circ] \times [-81^\circ, 81^\circ] \times \mathbb{R} \rightarrow \mathbb{R}, \quad (1)$$

denoted as $S_t(\theta, \phi, \lambda)$, where t is the time at which the image capturing has begun, θ and ϕ are the longitude and latitude, respectively, and λ refers to the wave-length band measured. The specific satellite imagery used is provided via the *Spinning Enhanced Visible and InfraRed Imager* (SEVIRI) of the *Meteosat Second Generation* (MSG) system, and obtained from EUMETSAT [35]. Images are available in 12 discrete frequency bands with a finite temporal resolution of $t_s = 15$. Eight bands represent the thermal infrared (IR) range, providing radiance resulting from the emission of the atmosphere and the Earth surface. They can be used to estimate the brightness temperature of the atmosphere and the surface. Three channels in the visible (VIS) spectrum measure the reflection of solar light at clouds or the Earth’s surface. This information can be used to retrieve the albedo of clouds and the surface. Lastly, the High-Resolution Visible (HRV) channel contains multiple broadband detection elements to scan the Earth with a lower sampling distance. The spatial resolution of the satellite images in Central Europe is roughly 0.05 latitude and 0.03

longitude (except the HRV band, which we do not consider). All satellite data is projected to an equirectangular projection with an equal spatial resolution of 0.05. We represent the satellite data as collections of 2D images $S_{\lambda,t}$ as data with values quantized to 16 bit integer. Figures 1b, 1c, 1d show examples of processed satellite imagery of the VIS006 ($\lambda = 0.6\mu\text{m}$), WV062 ($\lambda = 6.2\mu\text{m}$), and IR120 ($\lambda = 12\mu\text{m}$) channels at June 04 2016 at 12:00h UTC, where a lot of lightning occurred across Central Europe. The images already indicate that the optical band might be more informative, as the infrared channels only see the cloud tops. However, the patterns in the visible spectrum are complex and not easy to capture in a hand-designed classifier.

Lightning images: Measurements of lightning activity are given as point sets $\{l_1, \dots, l_{n_l}\} \in \mathbb{R}^4$, with two spatial geo-coordinates, a time coordinate, and the electrical current I_i . The lightning data used in this study was obtained from the LINET lightning detection network [7]. LINET is a low-frequency long-range lightning detection network (VLF/LF) using the time-of-arrival (TOA) method, consisting of several ground-based lightning sensors. For further processing, we convert the point set into images, with the same spatial and temporal resolution of the satellite images. To this end, we bin all lightning events in an *aggregation time span* of $t_a = 15$, during which the satellite data has been measured, or, during experiments with varying *lead-time* Δt , offset by the corresponding Δt , and binarize them by setting the closest pixel to 1 (with background set to 0). To filter out potential noise, only lightning events with an electric current of at least 1 were considered. We denote the resulting lightning images by L_t . Figure 1a shows an example of a processed lightning image for the same example date (04 June 2016, 12h UTC).

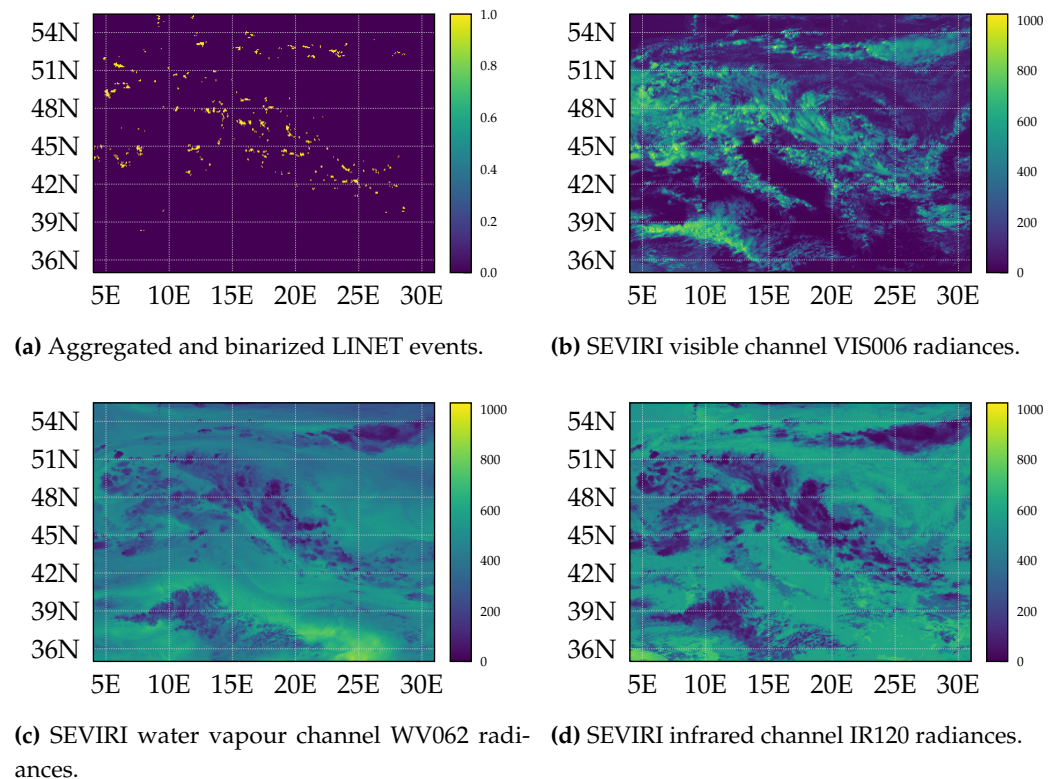


Figure 1. Satellite images and LINET events for Central Europe on June 04 2016 at 12:00 UTC. In the IR (WV) only the cloud top contributes to the emission and hence to the signal measured at the satellite, with exception of semi-transparent clouds. However, in the VIS the complete cloud contributes to the signal. The texture, thickness and shape of clouds are therefore much more pronounced in the VIS.

The region of interest (ROI) in our work are the mid-latitudes in Central Europe, between 2.0 W and 21.5 E, and 44.5 N and 57.5 N. The data is cropped to the defined ROI, and split up in smaller patches, according to Section A.3. We focus our work on the months

May to August in the years of 2016 and 2017. As additional test data set, we use data from August and September of 2021. Detailed information about the data and its use can be seen in Table A1, A2 for 2016 / 2017 and 2021, respectively. The processing of the data is done using multiple tools, such as *pyPublicDecompWT* [36], *SatPy* [37], *Pyresample* [38], and *Cartopy* [39].

The learning task can now be posed as a probabilistic prediction of a future lightning image: Let t_0 denote the *current time*, Δt the *lead time* for the prediction, t_s the *temporal sample spacing*, t_a the *aggregation time span* (in our case: $t_s = t_a = 15$) and k the *number of past satellite images* used for the prediction. With this information, we want to determine a probabilistic classifier f_θ that computes a probability map that specifies for each pixel the likelihood of a lightning event being marked in the future lightning image:

$$f_\theta(L_{t_0}, S_{t_0}, L_{t_0-1 \cdot t_s}, S_{t_0-1 \cdot t_s}, \dots, L_{t_0-k \cdot t_s}, S_{t_0-k \cdot t_s}) \approx L_{t_0+\Delta t}. \quad (2)$$

This problem is solved by representing the function f_θ by a deep neural network with parameters $\theta \in \mathbb{R}^d$. Learning is performed using maximum-likelihood: We assume that we are given a set of N training images with pixel-dimensions (W, H) and determine a good-fitting $\hat{\theta}$ as a (local) maximum of the likelihood function with L1 regularization:

$$\hat{\theta} = \arg \max_{\theta \in \mathbb{R}^d} -\frac{\lambda}{d} \sum_{j=1}^d |\theta_j| + \sum_{\substack{i \in \{1, \dots, N\} \\ x \in \{1, \dots, W\} \\ y \in \{1, \dots, H\}}} w(f_\theta^{(i)}, L_{t_0+\Delta t}^{(i)}, r_s)[x, y] \cdot \ell(f_\theta^{(i)}, L_{t_0+\Delta t}^{(i)})(x, y), \quad (3)$$

with

$$\ell(f_\theta^{(i)}, L_{t_0+\Delta t}^{(i)}) = \underbrace{\log f_\theta^{(i)} \cdot L_{t_0+\Delta t}^{(i)}}_{\text{positives}} + \underbrace{\log(1 - f_\theta^{(i)}) \cdot (1 - L_{t_0+\Delta t}^{(i)})}_{\text{negatives}}, \quad (4)$$

$$f_\theta^{(i)} = f_\theta(L_{t_0}^{(i)}, S_{t_0}^{(i)}, L_{t_0-1 \cdot t_s}^{(i)}, S_{t_0-1 \cdot t_s}^{(i)}, \dots, L_{t_0-k \cdot t_s}^{(i)}, S_{t_0-k \cdot t_s}^{(i)}). \quad (5)$$

Here, the operator $[x, y]$ refers to the pixel at position x, y in the respective images, w denotes a per-pixel class weight, based on the prediction skill of the sample. Hyperparameter settings are depicted in Tab. 4.

The main issue with the cross entropy (CE) based loss function shown in Eq. 4 is the naive (unweighted) summation over all pixels, regardless of their classification. Even when including all pixels in a *search radius* r_s , the amount of pixels “with lightning” is extremely sparse. Therefore, we add the event-based, per-pixel weight w in Eq. 3, which addresses the issue of heavy class imbalance in a sample. This differs from per-sample weighting strategies based on class balance, such as [40]. Per-sample weights work well with known class imbalances, e.g. when data from one class is underrepresented in the data set, by scaling the importance of single samples. This is only partially useful in our case: Thunderstorms are comparatively rare, but when they occur, lightning activity is reasonable high, spanning over multiple pixels in a sample.

In the following, we will construct a per-pixel weight map w , based on the samples class labels $L_{t_0+\Delta t}^{(i)}$, the predictions $f_\theta^{(i)}$, and a (fixed) “search radius” (r_s) as follows:

$$w(f_\theta^{(i)}, L_{t_0+\Delta t}^{(i)}, r_s)[x, y] = \begin{cases} \omega_{\text{lightning}}, & \text{if } L_{t_0+\Delta t}^{(i)}[x, y] = 1 \text{ or } ((x, y) \in \text{FP}_{r_s}(f_\theta^{(i)}, L_{t_0+\Delta t}^{(i)})) \\ \omega_{\text{no-lightning}}, & \text{otherwise,} \end{cases} \quad (6)$$

where $\text{FP}_{r_s}(p, l)$ is the set of pixels classified as false positive given the search radius r_s , the predictions p , and ground truth labels l of the sample, and $\omega_{\text{lightning}}$ is the modified

class-weight of the “lightning” class in the data set, and $\omega_{\text{no-lightning}}$ its corresponding counterpart. The class-weight ω_{cw} of a class “cw” is defined as

$$\omega_{cw} = \frac{1}{2} \times \frac{1}{|cw|}, \quad (7)$$

where $|cw|$ is the amount of instances of the class. Scaling by $\frac{1}{2}$ keeps the overall loss at a similar magnitude, so that the sum of the weights of all examples roughly stays constant. The values of $\omega_{\text{lightning}}$ and $\omega_{\text{no-lightning}}$ are pre-computed for the whole training split of the data set, and are shown in Table 2. Taking false positives into account, w will adapt to the performance of the classifier f .

Table 1. Measured class-weight of the training split for the 2016/2017 data set.

Class	Class-Weight
lightning	0.15
no-lightning	99.85

Table 2. Computed pixel-weights of the training split for the 2016/2017 data set, based on class-weights shown in Table 1 and Eq. 7.

Class	Pixel-Weight
$\omega_{\text{lightning}}$	3.33
$\omega_{\text{no-lightning}}$	5e-5

The *search radius* r_s is used to model the label uncertainty introduced by various factors, such as the dislocation between lightning and the center of the Cb, the geolocation error of the satellite, and the movement of the cloud [11]. During the computation of w , it is used to determine false positive predictions. The weight map w can efficiently be computed at every training step in parallel. During training, the loss values are averaged over all pixels and samples in a batch.

The network f used in our study is based on the U-Net [23] architecture, combined with ResNet-v2 [41] residual blocks, adapted to work with three dimensional input.

The input to our model is of the form (B, H, W, T, C) , where B is the batch size, H and W the height and width of an image, T the amount of time-frames and C the amount of channels. We fixed the height and width of our model to be $256\text{px} \times 256\text{px}$, which equals 12.8×12.8 , or roughly $1425\text{km} \times 1425\text{km}$. Larger input areas are split up according to the domain decomposition scheme described in Section A.3. Further, each input possesses a boundary region, overlapping with neighbouring inputs, which allows for a more precise prediction of the non-overlapping region, but which is excluded from the optimization process and evaluation.

Fig. 2 shows an overall view of the used network architecture. The U-Net structure, including skip-connections, can be seen. Varying from the original architecture, we replace the stacked convolution layers and the pooling layer at each down/up sampling step with residual blocks (with stride where necessary). Instead of cropping the feature map of the contracting path, we use the feature map after the down-sampling operator. We use convolutions for down-sampling, but deterministic trilinear up-sampling operations. Deviating from the original architecture, we replace batch normalization layers with instance normalization [42], which normalizes each element of the batch independently, i.e., only across the spatial and time dimension.

The residual blocks consist of full pre-activation units, optionally paired with pre-activated convolution shortcuts, as described in [22]. Fig. 3 illustrates the used residual (3c), down-sampling (3a) and up-sampling blocks (3b). Convolution layers use 3×3 filters for up- and down-sampling, and 1×1 otherwise. Non-linear layers consist of ReLU [43] activation functions.

As usual, we increase the amount of channels per block (representing coarser spatial scales). Empirically, we found good validation results for a stronger increase (only) for larger lead-times — details are shown in Table 3. When growing, it follows an exponential curve countering the spatial down- and up-sampling per block in the U-Net, which restrains memory use for weights and activations. The resulting network can be trained

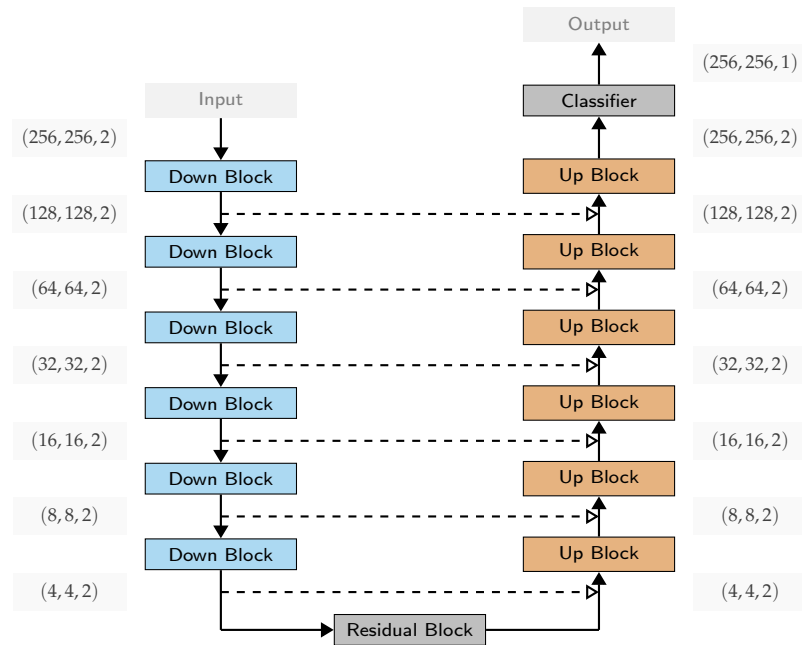


Figure 2. Illustration of the used network architecture, based on a U-Net with residual blocks. Depicted in light gray is the tensor size (H, W, T) between each block, the batch size and the number of channels have been omitted. Blue boxes correspond to down-sampling residual blocks, green boxes correspond to up-sampling residual blocks, and dark gray boxes correspond to residual blocks. Dashed lines indicate skip connections between the encoding and decoding phases of the U-Net. Solid lines indicate connections between layers.

on consumer graphics cards at a reasonable batch sizes. Together with the training settings, the expressivity of the neural network is adapted to “match” the complexity of the modeled problem.

Table 3. Detailed amount of channels in the network, depending on the lead time.

Lead time	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Bottleneck
0	13	13	14	14	15	16	16
30	14	16	20	28	46	81	81
60	14	16	21	31	52	97	97
90	14	16	21	32	56	106	106
120	14	16	22	33	59	114	114
180	14	17	23	38	70	145	145

Table 4. Detailed training settings, depending on lead time and network size.

Lead time	Bottleneck Channels (#)	Base LR	WD	Regularization Factor	Epochs
0	16	0.5	5.0d-6	0.1	14
30	81	0.5	7.0d-6	0.15	14
60	98	0.5	7.0d-6	0.2	14
90	107	0.5	7.0d-6	0.2	14
120	114	0.5	1.0d-5	0.3	14
180	124	0.5	1.0d-5	0.3	14

Our model is trained using stochastic gradient descent with momentum using decoupled weight decay (SGDW) [44] to optimize the loss function, described in Eq. 3. The base

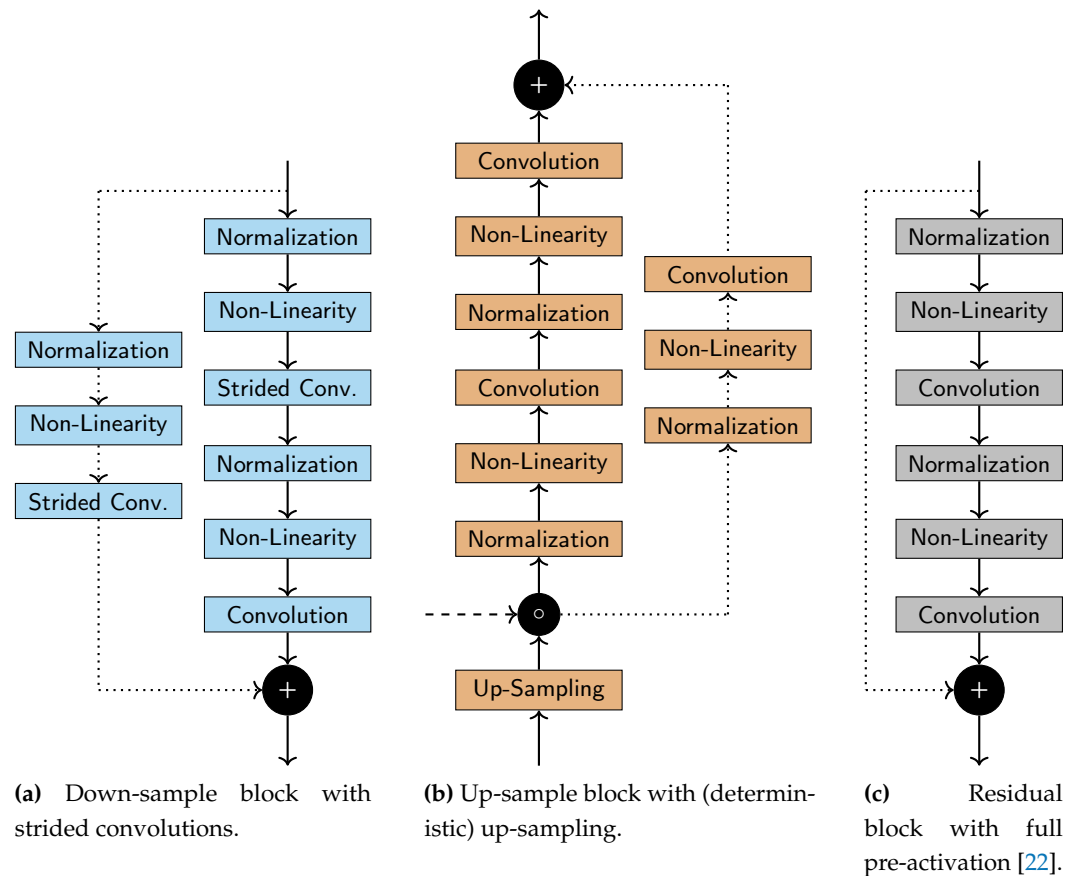


Figure 3. Detailed illustration of the used architecture blocks of our network, used for down- and up-sampling, as well as a residual block. Concatenation of tensors in the channel dimension is denoted as \circ . Addition of tensors of the same size is denoted as $+$.

learning rate (LR) is set as $\lambda = 0.5$, the weight decay (WD) and L1 regularization is set based on the amount of channels of the network, as shown in Tab. 4. We use a batch size of 64 and train the network for 14 epochs. In each epoch the network trains on a random permutation of the complete training data set. At the end of each epoch, we validate the network's performance on the validation data set and save its parameters to disk. Every 8th batch we adapt the decision threshold based on the current best performance. During training, the LR is scheduled with a 1-cycle LR scheduler [45] (two-phase, cosine decay). The highest LR is reached at the beginning of epoch 3, the initial learning rate is set to be $\frac{1}{10}$ of the base LR, after reaching the high point, the LR is decaying to $\frac{1}{1000}$ of the base LR. Additional to the use of our custom weighted loss function, samples with no lightning activity are discarded during training.

All parameters (of all experiments) were tuned based on data given in Table A1. Our method uses the following input features: SEVIRI channel VIS 0.6, VIS 0.8, nIR 1.6, IR 3.9, WV 6.2, WV 7.3, IR 8.7, IR 9.7, IR 10.8, IR 12.0, IR 13.4 and the most recent lightning events from LINET.

We use PyTorch [46] for all of our experiments. A reference implementation is provided under a free license.

3. Results

We evaluate our method on testing data set aside from the original 2016/17 data set (Table A1 in the appendix), as well as testing data from the month of August and September of 2021, for which comparison data with a nowcasting method [13] at DWD was available. In line with good experimental practice, all hyperparameter tuning has

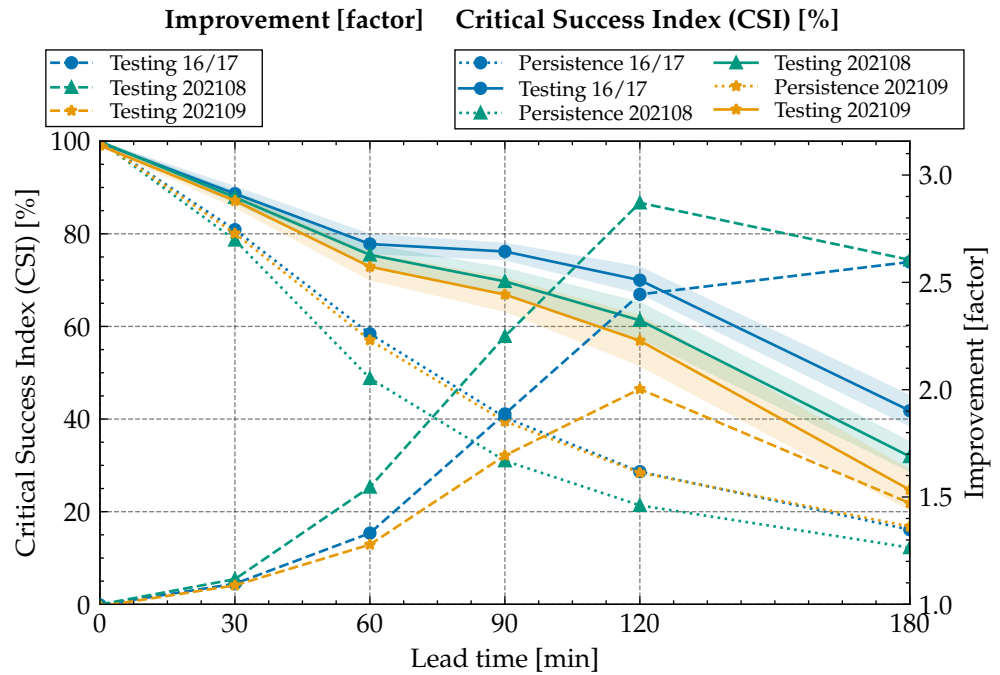


Figure 4. Impact factor (right y-axis) and CSI (left y-axis) over lead time: Performance measures using CSI for search radius 30.61 km of our method for various data sets. All lines showing the improvement factor start at the bottom-left, all lines showing the CSI start at the top-left.

been concluded using validation data before running any inference on any test data has been performed (with frozen, final hyperparameter settings). In particular in our case, where substantial hyperparameter tuning was required, this protocol minimizes the risk of reporting a coincidental success. Further, training and testing was performed 5 times with independent random initialization and training of the network, which can always lead to (a bit of) spread in the results, and mean and standard deviation are reported.

We report the critical success index (CSI) (see Appendix A.2 for details) of the deep network and a base-line “persistence” method, which just assumes that the most recent lightning events at the start of the prediction period persist at the same spot indefinitely. We also compute an “improvement factor” that measures the factor by which the deep network outperforms the base-line persistence model in terms of CSI. As discussed in more detail in Appendix A.4, this makes it easier to compare to other methods while avoiding misjudgement due to small deviations in the evaluation protocols for the corresponding CSIs.

The findings are shown in Figure 4 and Table 5.

Table 5. CSI of our ML model for various data sets, SR 30.61 and varying lead time (LT).

Data Split \ LT	0	30	60	90	120	180
Validation 16/17	99.9 ± 0.0	89.2 ± 0.8	78.5 ± 1.6	74.4 ± 1.6	66.8 ± 2.9	38.2 ± 2.3
Testing 16/17	99.7 ± 0.0	88.6 ± 1.4	77.8 ± 1.9	76.2 ± 1.6	70.0 ± 2.7	41.8 ± 3.1
Testing Aug. 21	99.9 ± 0.0	87.9 ± 1.1	75.4 ± 2.0	69.7 ± 2.7	61.3 ± 3.7	31.9 ± 3.0
Testing Sep. 21	99.1 ± 0.0	87.2 ± 1.3	72.9 ± 2.7	66.9 ± 3.5	56.9 ± 5.2	24.7 ± 4.2

3.1. Classifier Performance and Base-Line Comparison

As a first sanity check, for zero lead time, the network reaches very close to 100% success (i.e., matching the base-line when tasked with just predicting the last known events), which means that the network has been able to successfully learn to base its decisions solely on the lightning data in this setting.

With growing lead-time the results diverge from base-line, with substantial advantages for the learning-based model. We obtain a CSI of about 57 – 80% at 120 minutes and 25 – 42% at 180 minutes lead time, corresponding to an factor of improvement of ≥ 1.5 at 3 hours and ≥ 2.0 at 2 hours lead time.

This can be interpreted as a good outcome: Usually a CSI value of 50% is used to decide if the forecast provides a useful prediction. The CSI value is well above 50% for all testing periods up to a lead time of 120 minutes, demonstrating the quality and potential practical relevance of the classifier obtained from statistical machine learning.

Especially, our model is able to transfer to unseen data from September 2021, as neither the year, nor the month has been in the training or validation data set. As expected, testing results for this period show a decline in performance, but are still above 50% CSI for a 120 minute lead time. Data from 2021 in general performs a bit worse than testing data from the corresponding months of the 2016/17 data. As shown in Table 5, the performance on validation data is not systematically better than testing performance, showing that the hyperparameter choice has most likely not been overly specific to training and validation data used for method development.

3.2. Prediction Structure

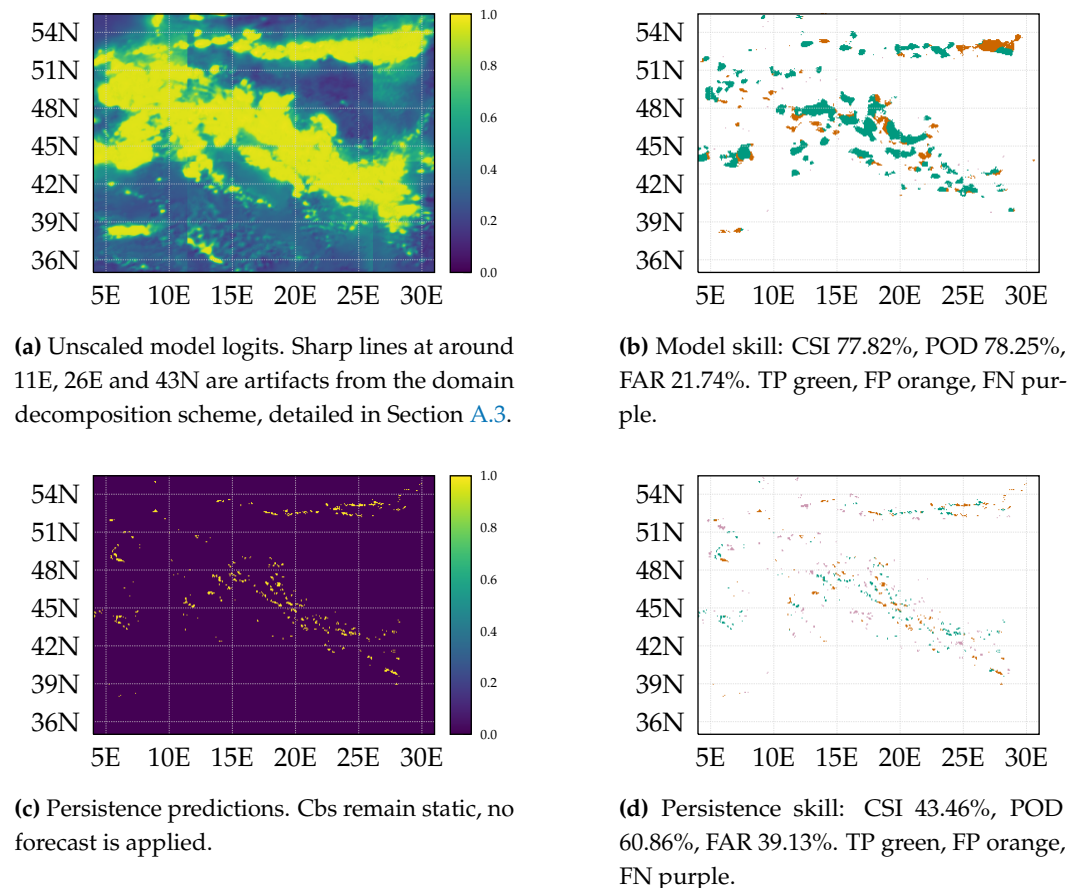


Figure 5. Predictions and skill of the compared methods for a 120 min forecast during a severe weather event, 4. June 2016 12:00 UTC [47].

To provide an impression of the outputs produced by the network, Figure 5 shows the model logits (inputs to the softmax-layer that yields normalized probabilities) and skill (True/False positives and negatives per pixels) for the previously used example of a severe weather event on June 04 2016 at 12:00h UTC [47]. It shows how the predicted lightning map significantly reduces false negatives and positives over base-line persistence.

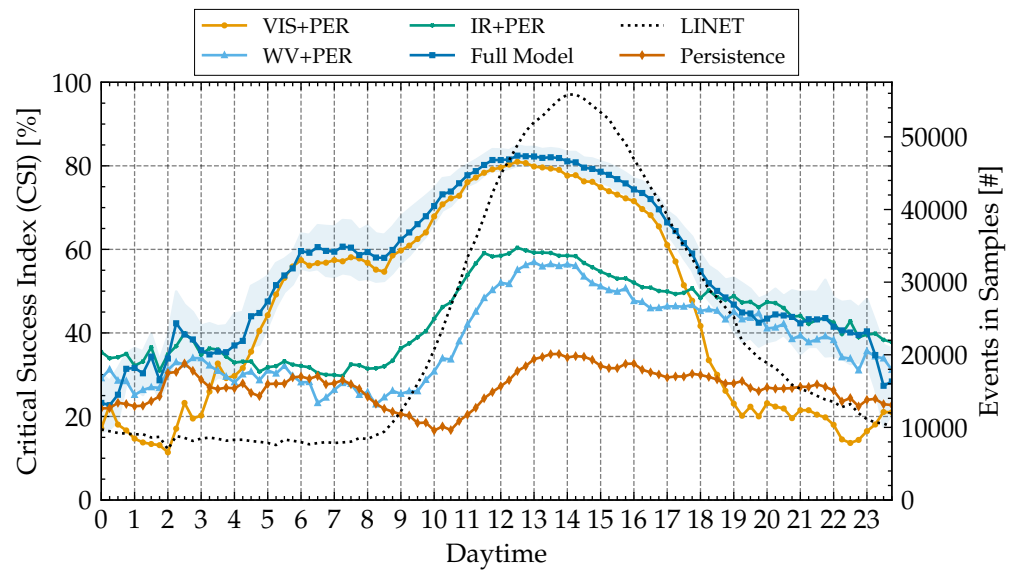


Figure 6. CSI over daytime (left y-axis): Performance measures for 120 min forecasts using CSI for search radius 30.61 of our method for the 2016/2017 test data sets, using only VIS+PER (VIS 0.6, VIS 0.8, nIR 1.6, Persistence), WV+PER (WV 6.2, WV 7.3, Persistence) and IR+PER (IR 3.9, IR 8.7, IR 9.7, IR 10.8, IR 12.0, IR 13.4, Persistence) channels. Average lightning events over daytime (right y-axis).

Visually, one can see that the predictions are automatically enlarged by the network to match the search radius, but our formulation of the adaptively weighted loss (Equation 6) does not lead to generally unsharp predictions, thereby reducing false positives (a naive dilation operation on all input and output data to model the search radius would lead to “overblown” predictions at the border regions and can seriously harm false positive rates).

Further details of the results are presented and discussed in the following subsections.

3.3. Comparison Against Physical Nowcasting Method

The DWD currently works on an improvement of the nowcasting applied within the 24/7 nowcasting method, discussed in detail in [13]. In order to get a first hint of the possible improvements of a machine learning-based method, the factor of improvement between persistence and nowcasting was calculated for the 120 minute prediction for both approaches. Due to a deviation in the lead time reference point, a 120 minute lead time in our work transfers to a 120 + 10 minute lead time in the physical nowcasting method. The validation method has been adapted as far as possible for this comparison. Nevertheless, it is not a direct comparison and hence does only provide an indicator for possible improvements using the method presented in this manuscript. The authors of [13] state a CSI for the persistence algorithm of 26%, which is consistent with the values reported in our work, see Table A3. For the physical nowcasting method, they state a CSI of 38%, thus, the improvement factor is of about 1.4 and hence significantly below the values achieved by our learned classifier. This is not a proof, but a strong indicator, that the developed method is able to gain hidden information and to improve the prediction of Cbs in the time frame of 0-3h.

3.4. Feature Attribution

After observing that our classifier is able to make statistically strong predictions (in comparison to alternative methods), we ask the question which “features” (input channels) are most informative in making the decision.

Figure 6 shows the result of training the classifier *again*, with some inputs omitted. The graph shows the CSI over the course of the day, with an overlay of the number of lightning

events recorded by LINET (dotted black curve) at these times of day on average (clearly peaking in the afternoon UTC time, which is late afternoon local time in the observed region). The time on the x-axis denotes to the prediction time, and corresponds to the number of lightning events shown (the actual data used for the prediction is taken 2h prior).

The full classifier model (dark blue curve) shows a significantly lower CSI at night-time than during the day: With the sunrise starting, performance improves, and drops again later in the evening: The increase of performance during 4-6am UTC coincides with the beginning dawn in Central Europe (local daylight-saving time being offset by +2h) in the relevant time frame of May to August. As more daylight becomes available closer to the start of summer, more visible information becomes available earlier in the day, which could explain the transition in accuracy. Similarly, accuracy declines towards the end of the day.

A second phenomenon, apparently overlaid with the hypothesized daylight effect, is a further increase in CSI between 9-16h UTC, peaking around noon (notably, at about 80% CSI). This corresponds to the time where most LINET events have been recorded. Apparently, prediction becomes easier if events are more common (which is a statistically plausible finding and would be expected).

The hypothesis that the network bases its decision mostly on information in the visible spectrum of light is solidified by looking at the CSI-curves when using only images from the visible spectrum (VIS/light orange) versus various infrared channels only (light blue/green): During daylight, the model using only visible features drops only slightly in performance over the full model (often within the 1σ -margin of error of the full model), while the infrared models perform significantly worse. Only at night, the prediction benefits from IR-information (with visible-only falling behind), although at a overall poorer level of performance. Surprisingly, the water-vapor bands alone yield an even slightly worse performance than the other IR-channels.

The comparison against base-line persistence shows an occasional drop below base-line for the restricted models, which might be explained by training noise (with variance increasing during low-event-count night times); this also suggest to not over-interpret smaller differences (WV vs. IR), but the gap to visible light appears very large and correspondingly very unlikely coincidental.

In summary, we observe that visible light has by far the largest contribution to classification performance, indicating the image recognition in the visible spectrum plays a major role. When visible light is available, the model performs very well at levels of CSI of 60 – 80%, which might have some operational relevance for applications to daytime activities. Infrared imagery appears to be much less informative to our classifier, which can only sustain a generally lower prediction performance consistently at day and night times. The water vapor channels alone, which are a target for physically-motivated, “hand-crafted” models [11], lead to overall worse prediction results than visible or other infrared bands.

4. Discussion

Our paper provides an end-to-end training technique for Cb nowcasting. As it works directly on unbalanced data, it can easily be used for predictions, as no pre-filtering of data is required. The experimental results provide clear indications that deep learning can be used to improve the Cb nowcasting even in this general scenario of naturally unbalanced classes. It should be noted, however, that the data used is not maximally unbalanced: All training and testing has so far been taken from the “warm” months of May to September, where lightning is much more prevalent in the observed region of Central Europe; including winter data and/or a more global excerpt of the planet might likely lead to a decrease in performance.

Comparison to operational nowcasting approaches: Most physical models are based on the extrapolation of the detected Cbs with atmospheric motion vectors (AMV; see [13] and references therein). This approach works well as long as the cells do not decay during the prediction period. Unfortunately, cells usually do decay after a certain lifetime, so

that this effect occurs regularly. Further, newly developed cells can not be captured by extrapolation of detected cells. These are serious drawbacks of the AMV approach. The quite good results achieved with deep learning indicate that the training process might enable the network to gain information of the life cycles of cells (decay, newly developed cells). The network seems to be able to learn to a certain extent whether Cbs decay or newly develop within the prediction period or not.

Importance of the visible-light spectrum: The results diagrammed in Figure 6 clearly show that the learning process benefits tremendously from the visible channels. In the IR only the cloud top contributes to the emission and hence to the signal measured at the satellite, with exception of semi-transparent clouds. However, in the VIS the complete cloud contributes to the signal, the reflections holds the information of the cloud optical thickness and effective cloud droplet radii [48]. Thus, the VIS provide much more information about the cloud textures, shapes and micro physics. It is likely that this information is used to learn information about the life cycle of Cbs. Otherwise, the better results compared to physical methods are hard to explain. Although the prediction of Cbs with NWP has been significantly improved, nowcasting is still assumed to outperform NWP in the first 0-3 hours. Thus, a lot of scientists aim to develop a seamless transition between nowcasting and NWP. Currently, mainly physical methods are used to achieve this goal, but deep learning might be a powerful alternative.

Author Contributions: Conceptualization, Sebastian Brodehl and Michael Wand; Data curation, Sebastian Brodehl and Richard Müller; Formal analysis, Sebastian Brodehl; Funding acquisition, Peter Spichtinger; Investigation, Sebastian Brodehl; Methodology, Sebastian Brodehl; Project administration, Elmar Schömer, Peter Spichtinger and Michael Wand; Resources, Elmar Schömer, Peter Spichtinger and Michael Wand; Software, Sebastian Brodehl; Supervision, Elmar Schömer and Michael Wand; Validation, Sebastian Brodehl and Richard Müller; Visualization, Sebastian Brodehl; Writing – original draft, Sebastian Brodehl, Richard Müller and Michael Wand; Writing – review & editing, Sebastian Brodehl, Richard Müller, Elmar Schömer, Peter Spichtinger and Michael Wand.

Funding: This research has been partially funded by the Carl-Zeiss-Stiftung (grant no. P2018-02-003, “Big Data in Atmospheric Physics (BINARY)”).

Data Availability Statement: Restrictions apply to the availability of the used data. Meteosat SEVIRI image data was obtained from EUMETSAT and is available at <https://navigator.eumetsat.int/product/EO:EUM:DAT:MSG:HRSEVIRI>. LINET lightning data was obtained from DWD.

Acknowledgments: The study is supported by the project “Big Data in Atmospheric Physics (BINARY)”, funded by the Carl-Zeiss-Stiftung (grant P2018-02-003). We acknowledge the DWD for providing access to the LINET data. We acknowledge EUMETSAT for providing access to the Meteosat SEVIRI image data. We acknowledge the ZDV of the Johannes Gutenberg University and the Mogon II Super Cluster for providing the necessary hardware, computing time and storage to prepare our experiments. We thank David Hartmann for fruitful discussions.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABI	Advanced baseline imager
AI	Artificial intelligence
ANN	Artificial neural network
BT	Brightness temperature
Cb	Cumulonimbus cloud (incl. thunderstorms & lightning)
CE	Cross entropy (loss function)
CI	Convective initiation
CRS	Coordinate reference system
CSI	Critical success index
DWD	Deutscher Wetterdienst (German Weather Service)
FAR	False alarm ratio
FED	Flash-extent density
FN	False negative
FP	False positive
GLD360	Vaisala global lightning detection network
GLM	Geostationary lightning mapper
HRIT	High rate image transmission
IF	Improvement factor
IR	Infrared
LR	Learning rate
MRMS	Multi-radar multi-sensor
MSG	Meteosat second generation
NWP	Numerical weather prediction
POD	Probability of detection
ROI	Region of interest
SEVIRI	Spinning enhanced visible and infrared imager
SGDW	Stochastic gradient descent with momentum using decoupled weight decay
SSP	Sub-satellite point
TN	True negative
TOA	Time-of-arrival
TP	True positive
WD	Weight decay
WV	Water vapour

Appendix A

Appendix A.1 Data Set Splits

Detailed information about the data and its use can be seen in Table [A1](#), [A2](#) for 2016 / 2017 and 2021, respectively. We focus our work on the summer months May to August in the years of 2016 and 2017. As additional test data set, we use data from August and September of 2021. We split the data based on weeks, and assign each to a different data set (shown in column "Data Set"). To uniformly cover the complete time span of the 2017/17 data with every data set, we assign successive weeks to alternating, repetitive sets of "train", "test", "train", "validation", and so on. This culminates in 19 train, 10 test, and 9 validation "weeks" for the 2016/17 period, and 10 weeks of testing data for the 2021 period (which is solely used for testing purposes).

Table A1. Train, test and validation splits for 2016/2017 data set. Events are shown in millions ($\times 1e6$) in the area (2.0 W – 21.5 E, 44.5 N – 57.5 N).

Year	Month	Week	Events	Data Set
2016	May	17	0.03	Train
2016	May	18	0.10	Test
2016	May	19	0.24	Train
2016	May	20	0.13	Validation
2016	May	21	0.70	Train
2016	May/June	22	0.94	Test
2016	June	23	0.80	Train
2016	June	24	0.56	Validation
2016	June	25	1.81	Train
2016	June/July	26	0.58	Test
2016	July	27	0.36	Train
2016	July	28	1.08	Validation
2016	July	29	0.70	Train
2016	July	30	1.18	Test
2016	August	31	0.95	Train
2016	August	32	0.28	Validation
2016	August	33	0.67	Train
2016	August	34	0.50	Test
2016	August	35	0.56	Train
2017	May	17	0.01	Validation
2017	May	18	0.15	Train
2017	May	19	0.26	Test
2017	May	20	0.28	Train
2017	May	21	0.33	Validation
2017	May/June	22	1.10	Train
2017	June	23	0.37	Test
2017	June	24	0.61	Train
2017	June	25	1.88	Validation
2017	June/July	26	1.05	Train
2017	July	27	0.81	Test
2017	July	28	0.77	Train
2017	July	29	1.29	Validation
2017	July	30	1.13	Train
2017	July/August	31	0.89	Test
2017	August	32	1.25	Train
2017	August	33	0.45	Validation
2017	August	34	0.43	Train
2017	August/September	35	0.95	Test

Table A2. Additional test split for 2021 data set. Events are shown in millions ($\times 1e6$) in the area (2.0 W – 21.5 E, 44.5 N – 57.5 N).

Year	Month	Week	Events	Data Set
2021	July	30	0.23	Test
2021	August	31	1.09	Test
2021	August	32	1.06	Test
2021	August	33	0.89	Test
2021	August	34	1.33	Test
2021	August/September	35	0.82	Test
2021	September	36	0.97	Test
2021	September	37	0.56	Test
2021	September	38	0.88	Test
2021	September/October	39	0.29	Test

Appendix A.2 Critical Success Index Calculation

We evaluate the performance of our method and the baseline method by calculating the critical success index (CSI) [49,50], similar to previous works, e.g. [11,51]. The CSI is the amount of correct classified events out of all events classified and undetected events. A natural question that arises is how to combine multiple CSI values of different samples to form a single performance measure.

We define the (combined) Critical Success Index (CSI) of all samples in a data set Ω as

$$\text{CSI}(\Omega) = \frac{\sum_{s \in \Omega} \text{TP}(s)}{\sum_{s \in \Omega} \text{TP}(s) + \sum_{s \in \Omega} \text{FN}(s) + \sum_{s \in \Omega} \text{FP}(s)}, \quad (\text{A1})$$

where TP, FN, FP are the true positives, false negatives and false positive of a sample, respectively. This way, we always measure the performance for an entire data set, by combining all events from all samples. This differs from classical machine learning tasks, e.g. image classification, where a more fine-grained performance measure that combines each individual measure of each sample is used.

Appendix A.3 Domain-Decomposition

Domain decomposition has become an essential tool in large-scale computation over the past decades because of its use to solve problems on parallel machines in the context of physical simulations [52]. In our case, a trade-off arises between the precision of the representation (or the level of detail) of the phenomenon and the memory and computer performance required to compute the model. We aim at fully capturing the spatial context in the input patch, using the highest level of detail. To maintain viable memory and compute requirements, we apply a domain decomposition scheme to split the input in smaller subdomains. This allows us to split the computational workload and process the smaller subdomains, without excluding regions.

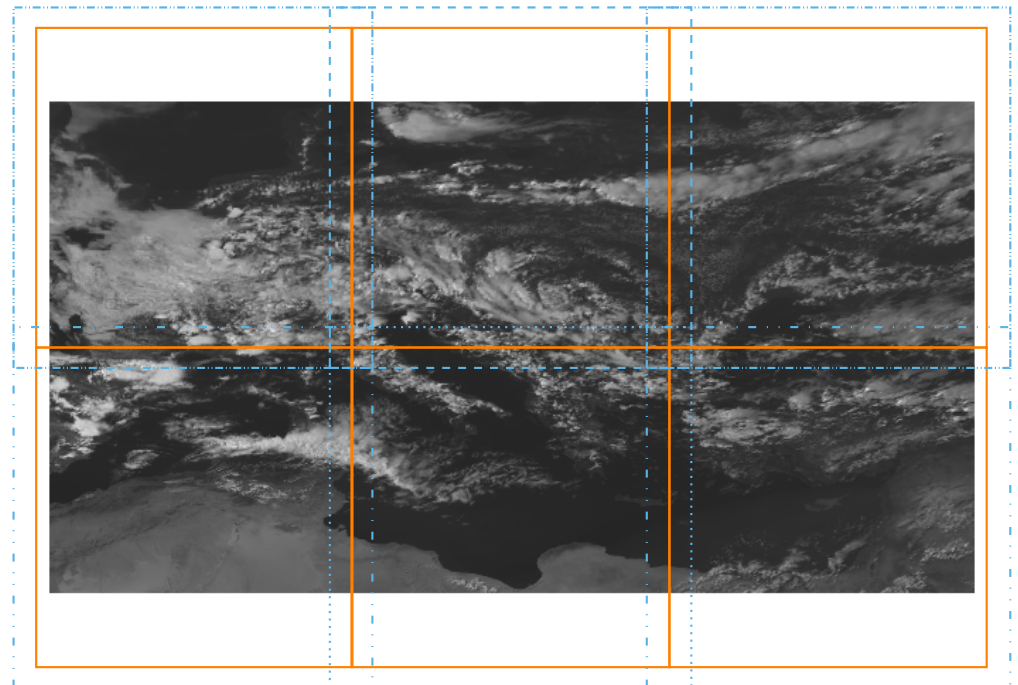


Figure A1. Example of the applied domain decomposition scheme into six slightly overlapping subdomains, shown for VIS006 (grayscale), 20160604 12:00 UTC. The non-overlapping regions are shown in orange, and their boundary (overlapping) regions are shown in blue.

Fig. A1 shows an example of the used domain decomposition. Here, the original domain (347px \times 654px; 17.35 \times 32.7; ca. 1929km \times 3636km) (grayscale image) is split

into six subdomains ($256\text{px} \times 256\text{px}$; 12.8×12.8 ; ca. $1423\text{km} \times 1423\text{km}$). The boundary (overlapping) region is marked in blue, consisting of a padding of 16px (0.8 ; ca. 88km) around the non-overlapping subdomain, which is shown in orange. The boundary region allows a more precise prediction at regions at the edge of the orange subdomain, because the cause of occurring lightning might lay outside of it. With a boundary width of roughly 88 and a time discretization of 15 of the satellite imagery, clouds with a horizontal speed of up to 100 are visible during their transition through the boundary region.

To ensure all subdomains have the same size the original domain is padded with zeros to the needed size, which is shown as white "background" in Fig. A1. Pixels in these padded regions are excluded from the optimization process and evaluation of our method.

Appendix A.4 Improvement Factor

During our research, we found it very difficult to (directly) compare our work with similar methods. The reasons for this were the under-specification of the metrics used to report the results, the lack of access to the used data and the lack of availability of the code necessary for evaluation. Thus, we were unable to reproduce the reported results and evaluate our method in the same way - which seems to be a general problem in machine learning research [53,54].

Therefore, we compare our method against a baseline method using a naive forecast with the PERSISTENCE algorithm and compute the improvement factor (IF) of our method over the given baseline. The PERSISTENCE algorithm is a model in which each lightning is assumed to be lightning in the same location and the same intensity it is at the last known time step. Thus, the Cbs remain at their original position throughout the prediction time, hence no forecast is applied and the Cbs are assumed to be static, which can be seen in Fig. 5c. This deterministic method requires no training, is fast to implement with no significant additional compute, and is often already available in the literature.

The improvement factor (IF) of a method α over method β is defined as

$$\text{IF}(\alpha, \beta) = \frac{\text{CSI}_\alpha}{\text{CSI}_\beta}, \quad (\text{A2})$$

in particular, the IF over the persistence model is given as $\text{IF}(\alpha, \text{PERSISTENCE})$.

By computing the IF over the PERSISTENCE model we are able to compare our work with others, where otherwise no direct comparison would be possible, e.g. due to a different implementation of the CSI, or a deviation in ROI. In this way, we can improve the comparison between studies.

Table A3. CSI of the PERSISTENCE model for various data sets, SR 30.61 and varying lead time (LT).

Data Split \ LT	0	30	60	90	120	180
Validation 16/17	100.00	81.38	56.11	38.08	27.73	17.10
Testing 16/17	100.00	80.89	58.42	40.37	28.64	16.11
Testing Aug. 21	100.00	78.66	48.79	31.01	21.36	12.25
Testing Sep. 21	100.00	80.08	57.03	39.50	28.42	16.79

Table A3 shows the CSI of the persistence model used for the calculation of the improvement factor. Figure 5d shows the skill for a severe weather event (4. June 2016 [47]).

References

1. Pruppacher, H.R.; Klett, J.D. *Microphysics of Clouds and Precipitation*; Vol. 18, *Atmospheric and Oceanographic Sciences Library*, Kluwer Academic Publishers: Dordrecht, 2010.
2. Zängl, G.; Reinert, D.; Rípodas, P.; Baldauf, M. The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core. *Quarterly Journal of the Royal Meteorological Society* **2015**, *141*, 563–579, [<https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1002/qj.2378>]. doi:<https://doi.org/10.1002/qj.2378>.

3. D. Reinert, D.; Prill, F.; Frank, H.; Denhard, M.; Baldauf, M.; Schraff, C.; Gebhardt, C.; Marsigli, C.; Zängl, G. DWD Database Reference for the Global and Regional ICON and ICON-EPS Forecasting System. Technical Report Version 2.1.7, DWD, 2021.
4. Schmid, F.; Wang, Y.; Harou, A. Nowcasting Guidelines—A Summary. *Bulletin n°* **2019**, *68*, 2.
5. Mass, C. Nowcasting: The Promise of New Technologies of Communication, Modeling, and Observation. *Bulletin of the American Meteorological Society* **2012**, *93*, 797 – 809. doi:10.1175/BAMS-D-11-00153.1.
6. Oprea, S.; Martinez-Gonzalez, P.; Garcia-Garcia, A.; Castro-Vargas, J.A.; Orts-Escolano, S.; Rodríguez, J.G.; Argyros, A.A. A Review on Deep Learning Techniques for Video Prediction. *CoRR* **2020**, *abs/2004.05214*, [2004.05214].
7. Betz, H.; Schmidt, K.; Oettinger, W.; Montag, B. Cell-tracking with lightning data from LINET. *Advances in Geoscience* **2008**, *17*, 55–61.
8. Fortun, D.; Bouthemy, P.; Kervrann, C. Optical flow modeling and computation: A survey. *Computer Vision and Image Understanding* **2015**, *134*, 1–21.
9. James, P.; Reichert, B.; Heizenreder, D. NowCastMIX: Automatic Integrated Warnings for Severe Convection on Nowcasting Time Scales at the German Weather Service, Weather and Forecasting. *Weather and Forecasting* **2018**, *33*, 1413–1433.
10. Schmetz, J.; Tjemkes, A.; Gube, M.; van der Berg, L. Monitoring deep convection and convective overshooting with Meteosat. *Advances in Space Research* **1997**, *19*, 433–441.
11. Müller, R.; Haussler, S.; Jerg, M.; Heizenreder, D. A Novel Approach for the Detection of Developing Thunderstorm Cells. *Remote Sens.* **2019**, *11*, 443. doi:10.3390/rs11040443.
12. Cintineo, J.L.; Pavlonis, M.J.; Sieglaff, J.M.; Wimmers, A.; Brunner, J.; Bellon, W. A Deep-Learning Model for Automated Detection of Intense Midlatitude Convection Using Geostationary Satellite Images. *Weather and Forecasting* **2020**, *35*, 2567 – 2588. doi:10.1175/WAF-D-20-0028.1.
13. Müller, R.; Barleben, A.; Haussler, S.; Jerg, M. A Novel Approach for the Global Detection and Nowcasting of Deep Convection and Thunderstorms. *Preprints* **2022**. doi:10.20944/preprints202206.0008.v1.
14. Lee, Y.; Kummerow, C.D.; Ebert-Uphoff, I. Applying machine learning methods to detect convection using Geostationary Operational Environmental Satellite-16 (GOES-16) advanced baseline imager (ABI) data. *Atmospheric Measurement Techniques* **2021**, *14*, 2699–2716. doi:10.5194/amt-14-2699-2021.
15. Autones, F. Algorithm Theoretical Basis Document for Convection Products. Technical report, NWC-SAF, 2016.
16. Gijben, M.; Coning, C. Using Satellite and Lightning Data to Track Rapidly Developing Thunderstorms in Data Sparse Regions. *Atmosphere* **2017**, *8*.
17. Schön, C.; Dittrich, J.; Müller, R. The Error is the Feature: How to Forecast Lightning using a Model Prediction Error. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019; Teredesai, A.; Kumar, V.; Li, Y.; Rosales, R.; Terzi, E.; Karypis, G., Eds. ACM, 2019, pp. 2979–2988. doi:10.1145/3292500.3330682.
18. Mecikalski, J.R.; Williams, J.K.; Jewett, C.P.; Ahijevych, D.; LeRoy, A.; Walker, J.R. Probabilistic 0–1-h convective initiation nowcasts that combine geostationary satellite observations and numerical weather prediction model data. *Journal of Applied Meteorology and Climatology* **2015**, *54*, 1039–1059.
19. Zhou, K.; Zheng, Y.; Dong, W.; Wang, T. A Deep Learning Network for Cloud-to-Ground Lightning Nowcasting with Multisource Data. *Journal of Atmospheric and Oceanic Technology* **2020**, *37*, 927 – 942. doi:10.1175/JTECH-D-19-0146.1.
20. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. 2009 IEEE conference on computer vision and pattern recognition. Ieee, 2009, pp. 248–255.
21. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*; Pereira, F.; Burges, C.J.C.; Bottou, L.; Weinberger, K.Q., Eds. Curran Associates, Inc., 2012, Vol. 25.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.
23. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III*; Navab, N.; Hornegger, J.; III, W.M.W.; Frangi, A.F., Eds. Springer, 2015, Vol. 9351, *Lecture Notes in Computer Science*, pp. 234–241. doi:10.1007/978-3-319-24574-4_28.
24. Dewitte, S.; Cornelis, J.; Müller, R.; Munteanu, A. Artificial Intelligence Revolutionises Weather Forecast, Climate Monitoring and Decadal Prediction. *Remote Sens.* **2021**, *13*, 3209. doi:10.3390/rs13163209.
25. Hoeser, T.; Künzer, C. Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends. *Remote Sens.* **2020**, *12*, 1667. doi:10.3390/rs12101667.
26. Hoeser, T.; Bachofer, F.; Kuenzer, C. Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review - Part II: Applications. *Remote Sens.* **2020**, *12*, 3053. doi:10.3390/rs12183053.
27. Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing* **2019**, *152*, 166–177. doi:https://doi.org/10.1016/j.isprsjprs.2019.04.015.
28. Boukabara, S.A.; Krasnopolsky, V.; Stewart, J.Q.; Maddy, E.S.; Shahroudi, N.; Hoffman, R.N. Leveraging Modern Artificial Intelligence for Remote Sensing and NWP: Benefits and Challenges. *Bulletin of the American Meteorological Society* **2019**, *100*, ES473 – ES491. doi:10.1175/BAMS-D-18-0324.1.

29. Ayzel, G.; Scheffer, T.; Heistermann, M. RainNet v1.0: a convolutional neural network for radar-based precipitation nowcasting. *Geoscientific Model Development* **2020**, *13*, 2631–2644. doi:10.5194/gmd-13-2631-2020.
30. Lebedev, V.; Ivashkin, V.; Rudenko, I.; Ganshin, A.; Molchanov, A.; Ovcharenko, S.; Grokhovetskiy, R.; Bushmarinov, I.; Solomentsev, D. Precipitation Nowcasting with Satellite Imagery. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019; Teredesai, A.; Kumar, V.; Li, Y.; Rosales, R.; Terzi, E.; Karypis, G., Eds. ACM, 2019, pp. 2680–2688. doi:10.1145/3292500.3330762.
31. Agrawal, S.; Barrington, L.; Bromberg, C.; Burge, J.; Gazen, C.; Hickey, J. Machine Learning for Precipitation Nowcasting from Radar Images. *CoRR* **2019**, *abs/1912.12132*, [1912.12132].
32. Shi, X.; Gao, Z.; Lausen, L.; Wang, H.; Yeung, D.; Wong, W.; Woo, W. Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model. Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA; Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H.M.; Fergus, R.; Vishwanathan, S.V.N.; Garnett, R., Eds., 2017, pp. 5617–5627.
33. Han, D.; Lee, J.; Im, J.; Sim, S.; Lee, S.; Han, H. A Novel Framework of Detecting Convective Initiation Combining Automated Sampling, Machine Learning, and Repeated Model Tuning from Geostationary Satellite Data. *Remote. Sens.* **2019**, *11*, 1454. doi:10.3390/rs11121454.
34. Mostajabi, A.; Finney, D.L.; Rubinstein, M.; Rachidi, F. Nowcasting lightning occurrence from commonly available meteorological parameters using machine learning techniques. *Npj Climate and Atmospheric Science* **2019**, *2*, 1–15.
35. EUMETSAT. High Rate SEVIRI Level 1.5 Image Data - MSG - 0 degree. <https://navigator.eumetsat.int/product/EO:EUM:DAT:MSG:HRSEVIRI>, 2022.
36. Brodehl, S. pyPublicDecompWT: Python bindings for EUMETSAT's PublicDecompWT (v2.8.1.3). <https://pypi.org/project/pyPublicDecompWT>, 2022.
37. Raspaud, M.; Hoese, D.; Lahtinen, P.; Finkensieper, S.; Holl, G.; Proud, S.; Dybbroe, A.; Meraner, A.; Feltz, J.; Zhang, X.; et al. pytroll/satpy: Version 0.36.0 (2022/04/14), 2022. doi:10.5281/zenodo.6460193.
38. Hoese, D.; Lahtinen, P.; Raspaud, M.; Roberts, W.; Lavergne,.; Bot, S.; Finkensieper, S.; Dybbroe, A.; Holl, G.; Itkin, M.; et al. pytroll/pyresample: Version 1.23.0, 2022. doi:10.5281/zenodo.6375741.
39. Met Office. Cartopy: a cartographic python library with a Matplotlib interface. <https://scitools.org.uk/cartopy>, 2010 - 2015.
40. Cui, Y.; Jia, M.; Lin, T.; Song, Y.; Belongie, S.J. Class-Balanced Loss Based on Effective Number of Samples. IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE, 2019, pp. 9268–9277. doi:10.1109/CVPR.2019.00949.
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV; Leibe, B.; Matas, J.; Sebe, N.; Welling, M., Eds. Springer, 2016, Vol. 9908, *Lecture Notes in Computer Science*, pp. 630–645. doi:10.1007/978-3-319-46493-0_38.
42. Ulyanov, D.; Vedaldi, A.; Lempitsky, V.S. Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR* **2016**, *abs/1607.08022*, [1607.08022].
43. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel; Fürnkranz, J.; Joachims, T., Eds. Omnipress, 2010, pp. 807–814.
44. Loshchilov, I.; Hutter, F. Fixing Weight Decay Regularization in Adam. *CoRR* **2017**, *abs/1711.05101*, [1711.05101].
45. Smith, L.N.; Topin, N. Super-Convergence: Very Fast Training of Residual Networks Using Large Learning Rates. *CoRR* **2017**, *abs/1708.07120*, [1708.07120].
46. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *CoRR* **2019**, *abs/1912.01703*, [1912.01703].
47. Klimaveränderung und Wasserwirtschaft (KLIWA). Klimawandel in Süddeutschland - Veränderung von meteorologischen und hydrologischen Kenngrößen. Monitoringbericht 2016., 2016.
48. Nakajima, T.; King, M.D. Determination of the Optical Thickness and Effective Particle Radius of Clouds from Reflected Solar Radiation Measurements. Part I: Theory. *Journal of Atmospheric Sciences* **1990**, *47*, 1878 – 1893. doi:10.1175/1520-0469(1990)047<1878:DOTOTA>2.0.CO;2.
49. Schaefer, J.T. The Critical Success Index as an Indicator of Warning Skill. *Weather and Forecasting* **1990**, *5*, 570 – 575. doi:10.1175/1520-0434(1990)005<0570:TCSIAA>2.0.CO;2.
50. Gerapetritis, H.; Pelissier, J.M. On the behavior of the critical success index, 2004.
51. Ravuri, S.; Lenc, K.; Willson, M.; Kangin, D.; Lam, R.; Mirowski, P.; Fitzsimons, M.; Athanassiadou, M.; Kashem, S.; Madge, S.; et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature* **2021**, *597*, 672–677.
52. Dolean, V.; Jolivet, P.; Nataf, F. *An introduction to domain decomposition methods - algorithms, theory, and parallel implementation*; SIAM, 2015.
53. Pineau, J.; Vincent-Lamarre, P.; Sinha, K.; Larivière, V.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.B.; Larochelle, H. Improving Reproducibility in Machine Learning Research(A Report from the NeurIPS 2019 Reproducibility Program). *J. Mach. Learn. Res.* **2021**, *22*, 164:1–164:20.
54. Garg, S.; Rasp, S.; Thuerey, N. WeatherBench Probability: A benchmark dataset for probabilistic medium-range weather forecasting along with deep learning baseline models. *CoRR* **2022**, *abs/2205.00865*, [2205.00865]. doi:10.48550/arXiv.2205.00865.