# How To Normalize Satellite Images for Deep Learning

Tackling the long-tailed satellite imagery data in deep learning applications

Written by *Nika Oman Kadunc*. Work performed by *Devis Peressutti*, *Nejc Vesel*, *Matej Batič*, *Sara Verbič*, *Žiga Lukšič*, *Matej Aleksandrov* and *Nika Oman Kadunc*.

*Normalization of input data for deep learning (DL) applications is an important step that impacts network convergence and final results. In case of long-tailed satellite signals, proper normalization can be quite a challenge — we were tired of trying to understand why the models we trained on one location didn't always translate to another location as well as we*

*thought they should — so we set out to explore what kind of normalization schemes are most suited for the task.*

## Introduction

Deep-learning-based automatic field delineation from satellite images is becoming an important tool in large-scale evaluations and monitoring of land cover and crop production. One of the steps in the workflow is normalization of the band values, which impacts network performance and quality of the results.

The aim of this study is to investigate and quantify the effects of several normalization methods on the performance of our existing field delineation algorithm. In addition, we want to assess the feasibility of using a single set of normalization factors for large scale applications, performing well under various types of variability in band distributions. For this purpose, it is necessary that the training dataset includes imagery from a larger geographical region which captures the reflectance

variability and a large time period (whole year) to capture the seasonal variability.

Proper normalization of the images is a step often underestimated, although it is essential to the DL algorithms. Typically, the input images are normalized so that the mean is centered at 0 and the standard deviation at 1 [1][2]. This assumption holds when the distributions are close to normal but is less suited to the case of reflectances or digital numbers (DN), as they are 0-bounded and long-tailed. In addition, saturated DN values represent outliers which can greatly affect the statistics computed. In this study, we aim to investigate different normalization methods that would be more suited to the properties of the satellite imagery data and would allow to center the distributions and reduce the impact of outliers.

We present our investigations of satellite image histograms, different normalization methods and their effect on the results of

field delineation across the whole region of Europe. First, we present the dataset of satellite imagery obtained for the purpose of this study. Next, we investigate the effects of image histogram variability according to land type, geographical location and time period. As we are interested in field delineation on agricultural land, we focused on the variability of cropland according to geographical location. We then present three methods of histogram normalization and compare the results for automatic field delineation.

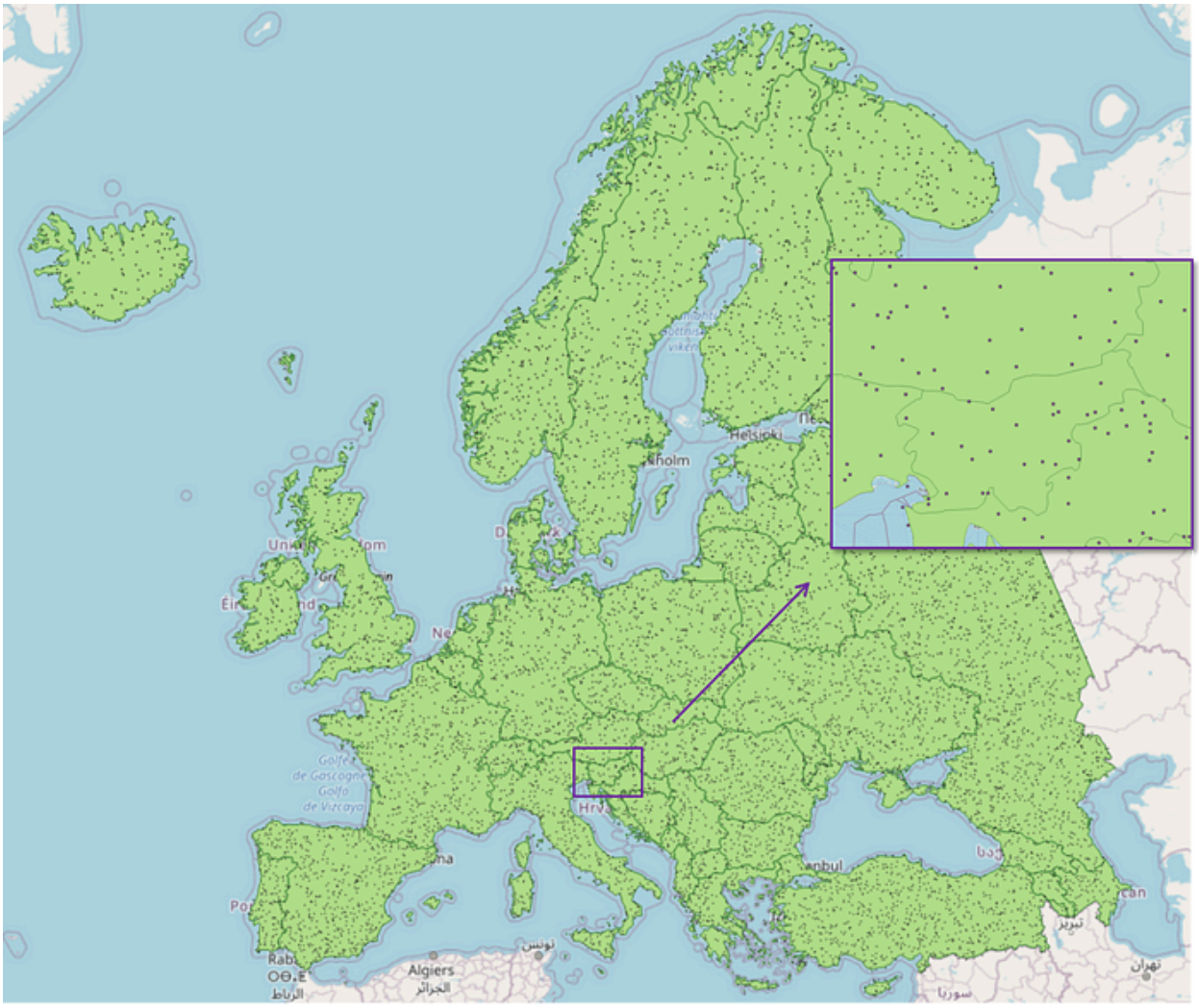
Although we focus on our algorithm for field delineation, the findings presented here could be applicable to different large-scale applications based on machine learning, such as land cover, crop classification and super-resolution.

## The dataset

The dataset was designed to capture the variability of different geographical locations across Europe and of different time

periods. Small patches were selected to give a better spatial sampling for a given total sampled area. Sentinel 2 L1C bands of 10 000 randomly distributed patches of size 256 × 256 pix at a 10 m resolution across Europe and some neighboring regions for a whole year were obtained. The patches correspond to 0.66 % of the given European AOI and are shown in Fig. 1 (top). Bands B2, B3, B4 and B8 were selected for analysis. The images were filtered to remove snowy and cloudy acquisitions using s2cloudless and the snow masking available in eo-learn. After filtering, a total number of 180M pixels constituted the dataset. ESA World Cover data was also downloaded for each of the patches to obtain information on the land type (i.e. tree cover, cropland, water, etc.). An example of a patch and the corresponding land cover data is shown in Fig. 1 (bottom).

The entire workflow was implemented using the functionalities of Sentinel Hub, eo-learn and eo-grow.




Patch image (RGB)

World Cover data

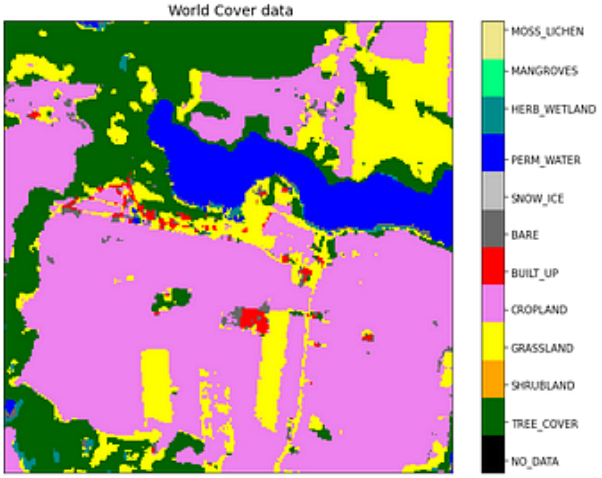


Figure 1: Randomly distributed patches over Europe and neighboring regions that constitute the dataset (top) ; an example of a RGB image of a patch together with the corresponding World Cover data (bottom).

## A look into the dataset

We explored the properties and the variability of the histograms of considered bands in terms of different parameters: land cover, geographical location and time period.

### Land cover exploration

Firstly, the distribution of sampled pixels in terms of land cover was investigated and is presented in Fig. 2. The ESA Land Cover does not provide intra-year temporal data, so we have taken single time frame of the Sentinel-2 data into account for this exploration.

We see in Fig. 2 that tree cover is the most represented class of land cover in the sample dataset, followed by grassland and cropland. Bear in mind that these distributions are subjected to

the classification error of World Cover, so the actual values might slightly differ.

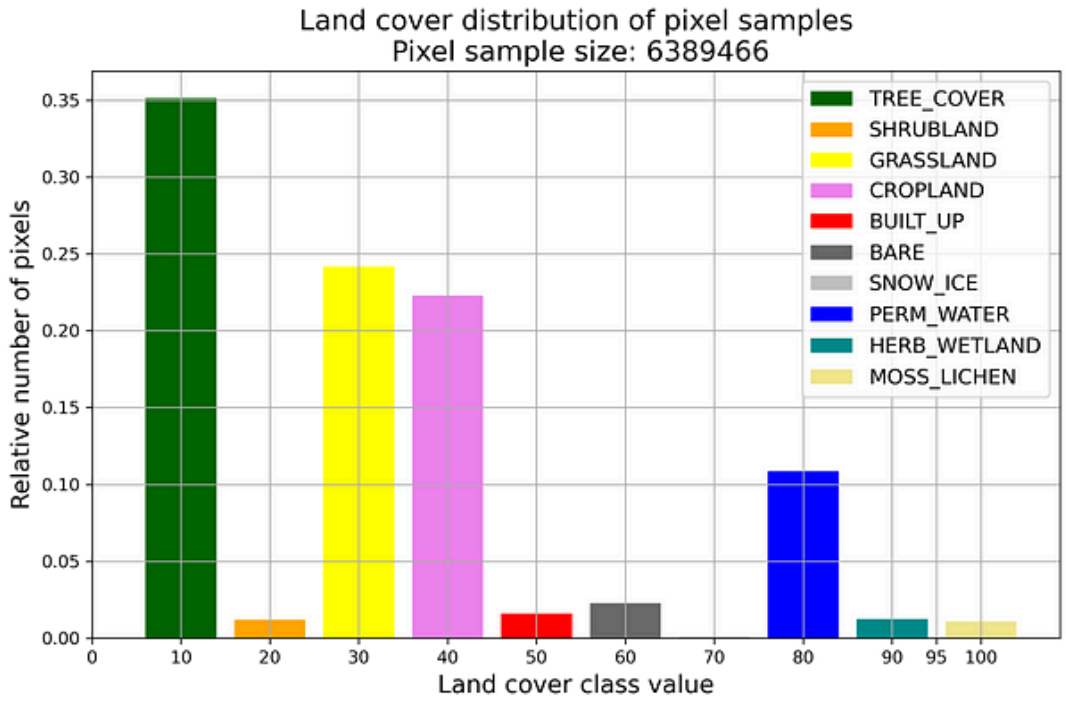



Figure 2: Land cover distribution of the sample spatial pixels of the dataset.

Next, we explore the contributions of different land cover classes to the whole band DN values histograms of pixels (Fig. 3). Remember that for Sentinel-2, reflectances are obtained from DN values by dividing them by 10 000. The logarithmic scale of the data is added for better visibility and easier comparison. We see from Fig. 3 that water dominates the left part of the

histograms (bands B3, B4, B8). Cropland class, which is of most interest in this analysis, lies in the mid part of the histogram, strongly overlapping with grassland in bands B2, B3 and B4. The information about relative position of specific land cover classes within the whole histogram can be important when choosing the appropriate normalization method that can affect different parts of the histograms.

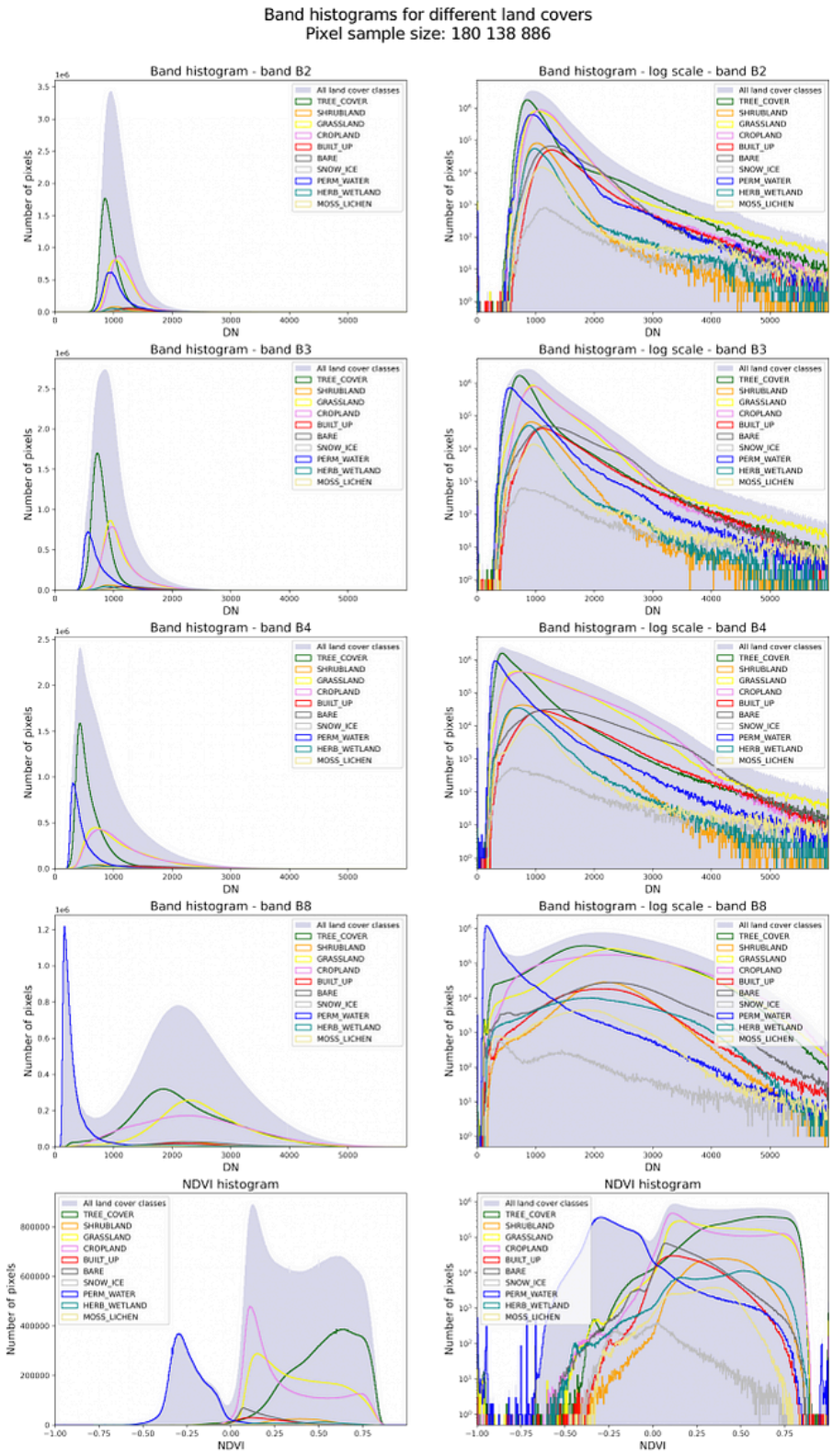



Figure 3: Band histograms of different land covers in linear scale (left) and logarithmic scale (right).

## Geographical exploration

For the purpose of geographical variability analysis of the band DN values, a partition of the Europe AOI was made into different regions (depicted in Fig. 4). Five distinct regions as divided by the partition grid were selected for comparison of variation of the histograms, shown in Fig 5.

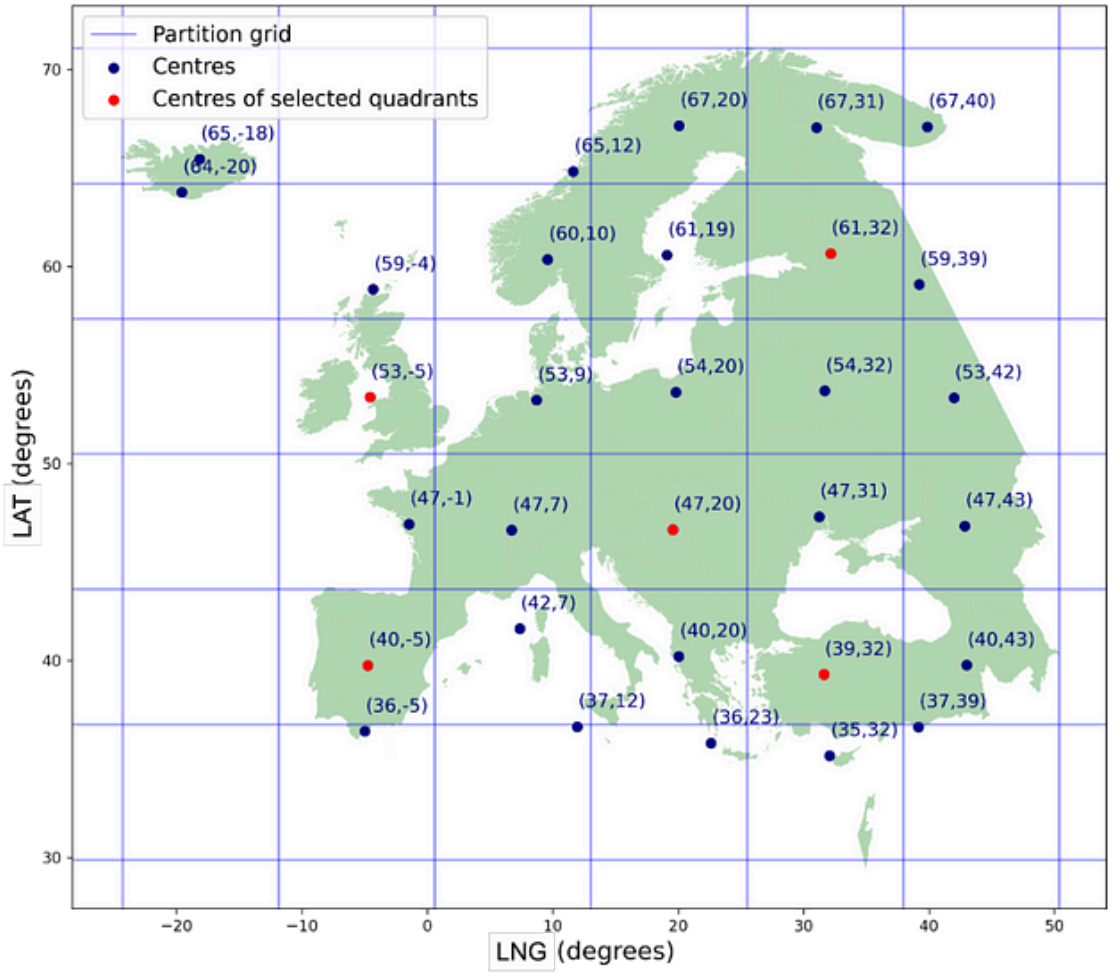


Figure 4: Locational partition of the Europe AOI with the mean latitude and longitude of the sample patches in each quadrant. The centers of selected quadrants used for further comparisons are colored red.

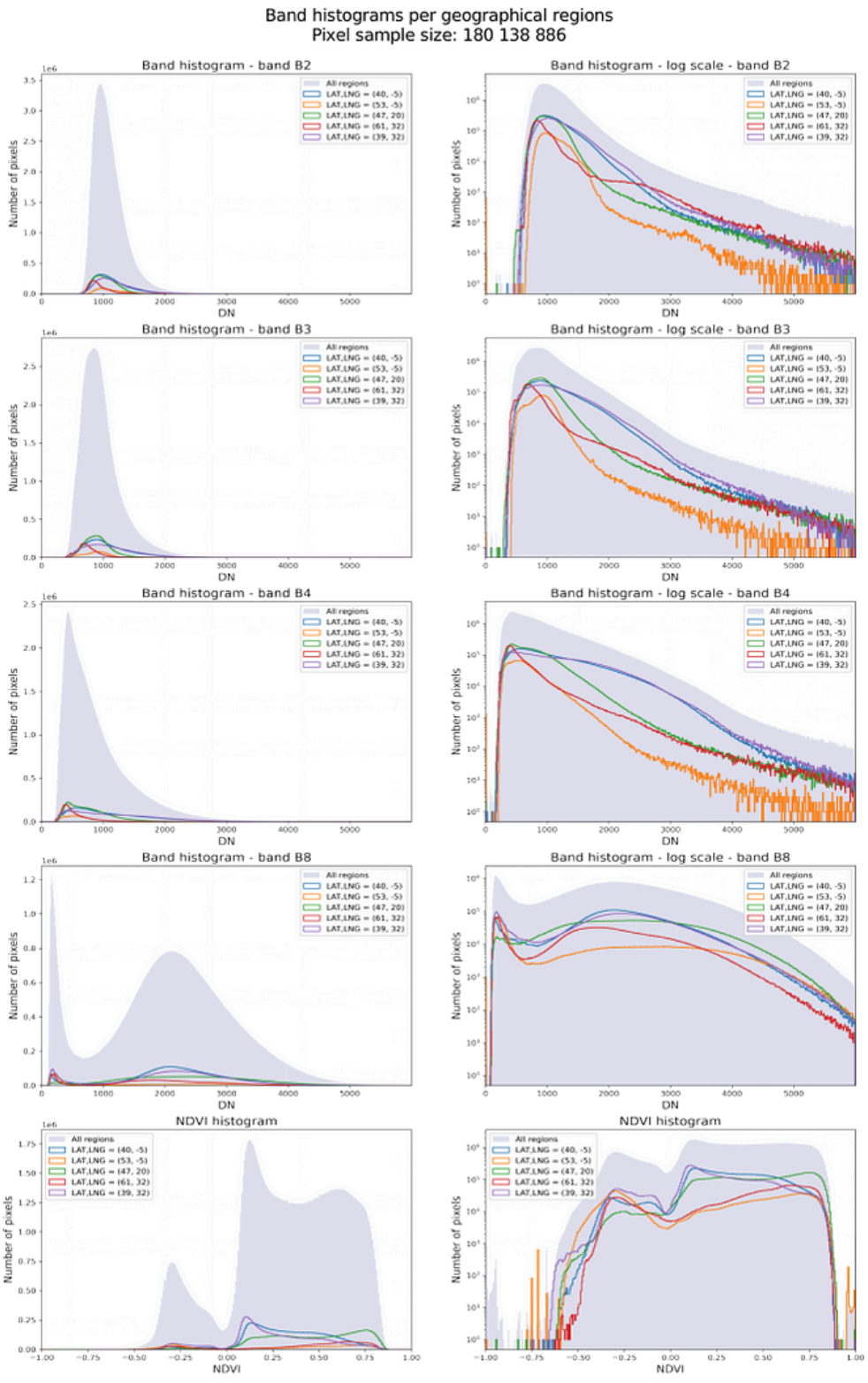



Figure 5: Locational variability of the band DN values histograms.

We see in Fig 5 that the variability in the DN values is much more correlated to the latitude of the region than to the longitude as the histograms from regions with similar latitudes show the most similarity. The difference is most apparent in the band B4 values.

To analyze the geographical (locational) variability of the agricultural land across the Europe region, a comparison of only the cropland land cover class was made and is shown in Fig. 6. Again, differences according to region latitude can be observed in the cropland histograms. For instance, the blue (Spain) and purple (Turkey) histograms are very similar, and greatly differ from the red (Baltic countries) and orange (United Kingdom) histograms, which are however similar between themselves. The green histogram (Hungary) lies between the two pairs mentioned above. These differences mean that the normalization factors computed for one region might not be equally suitable for normalization of a region with a different latitude.

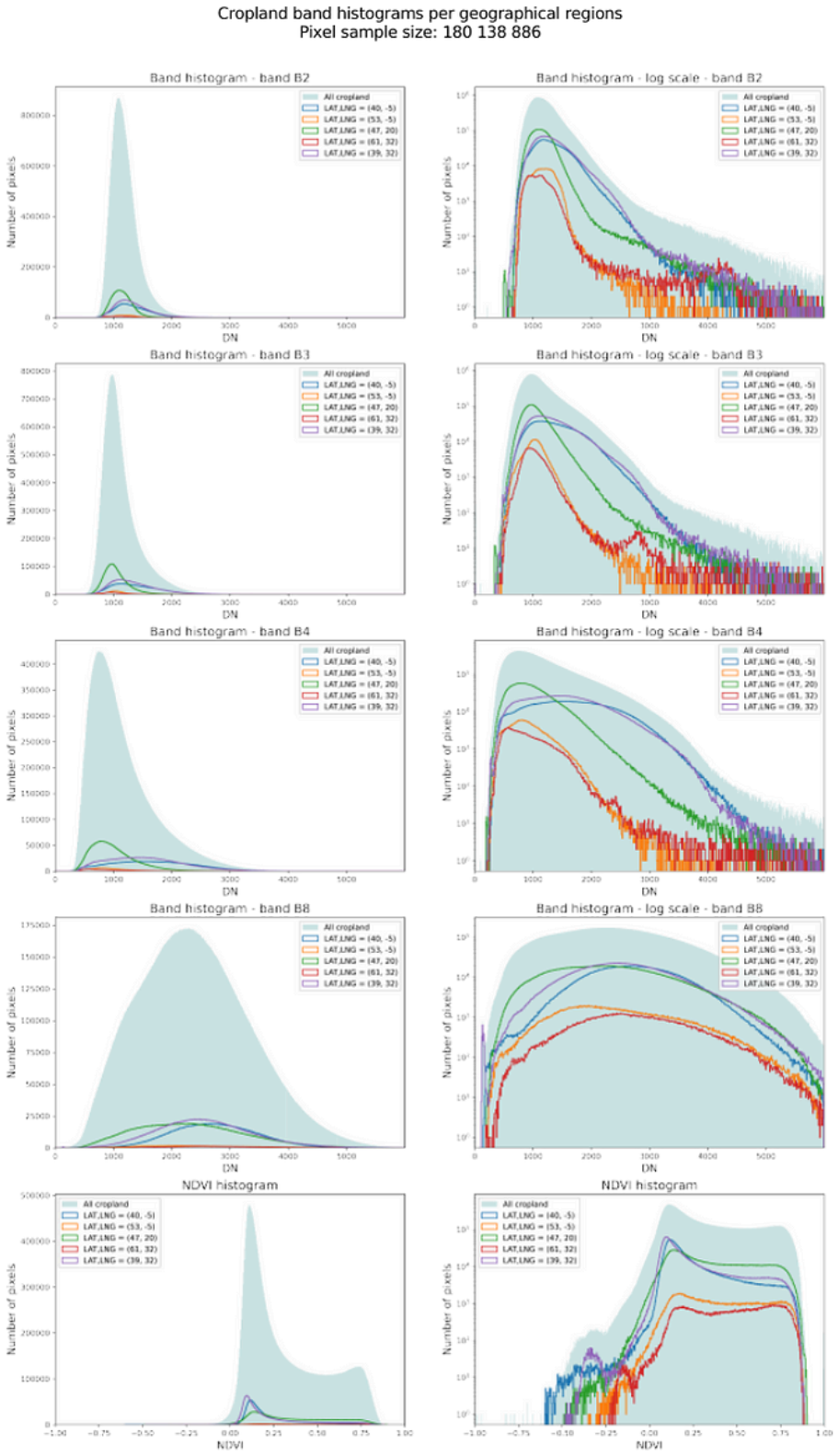



Figure 6: Locational variability of cropland histograms.

### Temporal exploration

Finally, an analysis of temporal differences in band values was made. The dataset was divided into monthly time periods and the distribution of samples with regards to months is shown in Fig. 7. We can see that when filtering the region of Europe with the snow and cloud mask, acquisitions in winter months get filtered out more, resulting in a distribution with a peak in July.

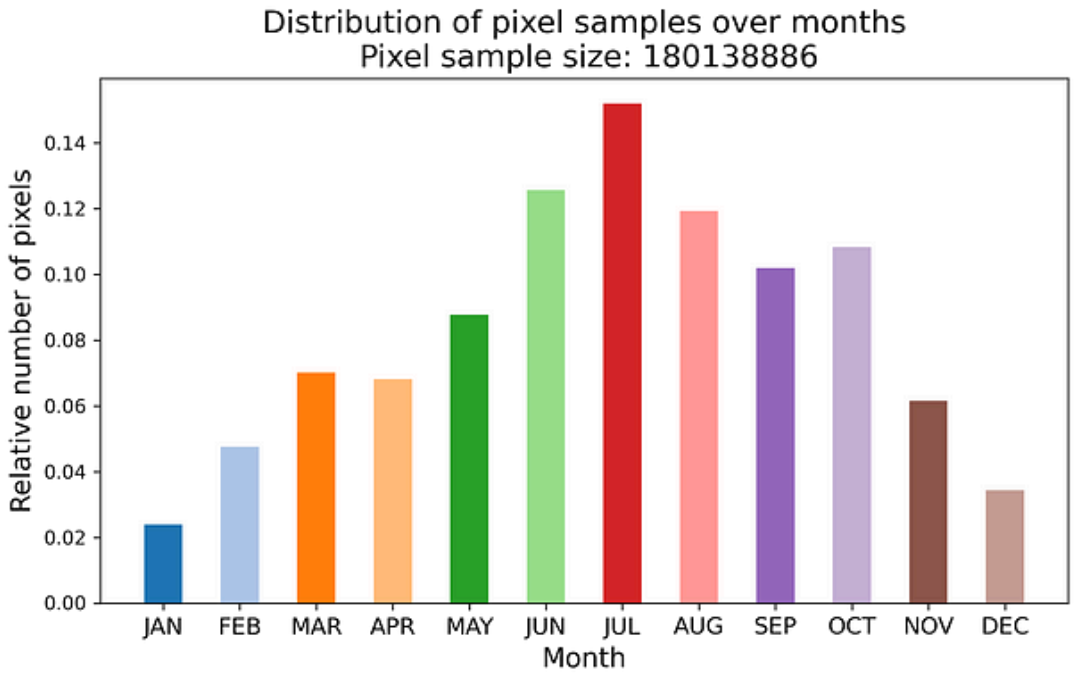



Figure 7: Distribution of dataset samples according to monthly time periods.

Fig. 8 shows the temporal variability of the band DN histograms. In all bands, the discrepancy is the most apparent between the

months of January and July, due to the difference in the presence of vegetation. These differences are the most discernible in the B4 and B8 bands, which strongly reflect changes in vegetation.

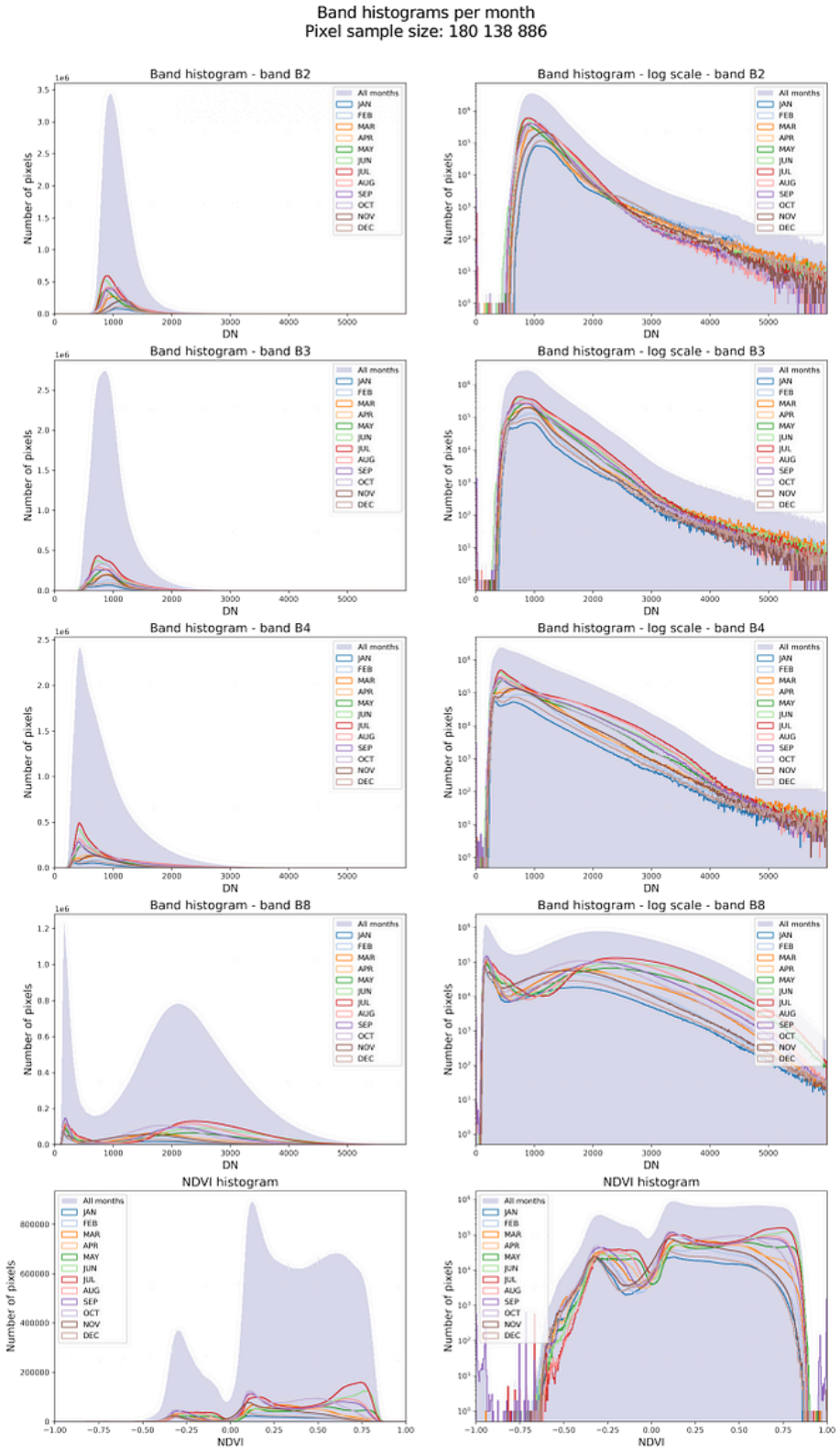



Figure 8: Temporal variability of band DN values histograms.

The temporal variability of satellite imagery is very important when choosing the time period for DL purposes, especially for field delineation. The fields change significantly with the seasons and are subject not only to natural changes but also cultivation activities.

## Histogram normalization

Normalization of the input data is very important because the network training convergence depends on the input values; it converges faster if its inputs are transformed to have zero means and unit variances [1]. This is called input histogram normalization or standardization. It allows the network to operate in a good range, since it is usually initialized with random weights with 0 mean. Normalizing images with regards to standard deviation prevents the gradients from exploding, which could happen if values of the computed features are too large, making the convergence of the network more difficult.

While histogram standardization may be the best choice for many cases where the input data follows a nearly normal distribution, in the case of DNs, where the band distributions are long tailed and 0-bounded, applying standardization does not give the desired properties of the data for the network operation. This issue has already been identified and addressed by applying a different normalization function to the band data [3]. In this study, we will test three different methods of normalization that aim to give a better-balanced data for the network operation in field delineation.

### Linear normalization

The linear normalization yields a re-mapping of the given range of input values to span across a different range, more appropriate for the task at hand. It is performed with the application of a linear scaling function to the band DN values:

$$val_{out} = (val_{in} - c)\left(\frac{b-a}{d-c}\right) + a$$

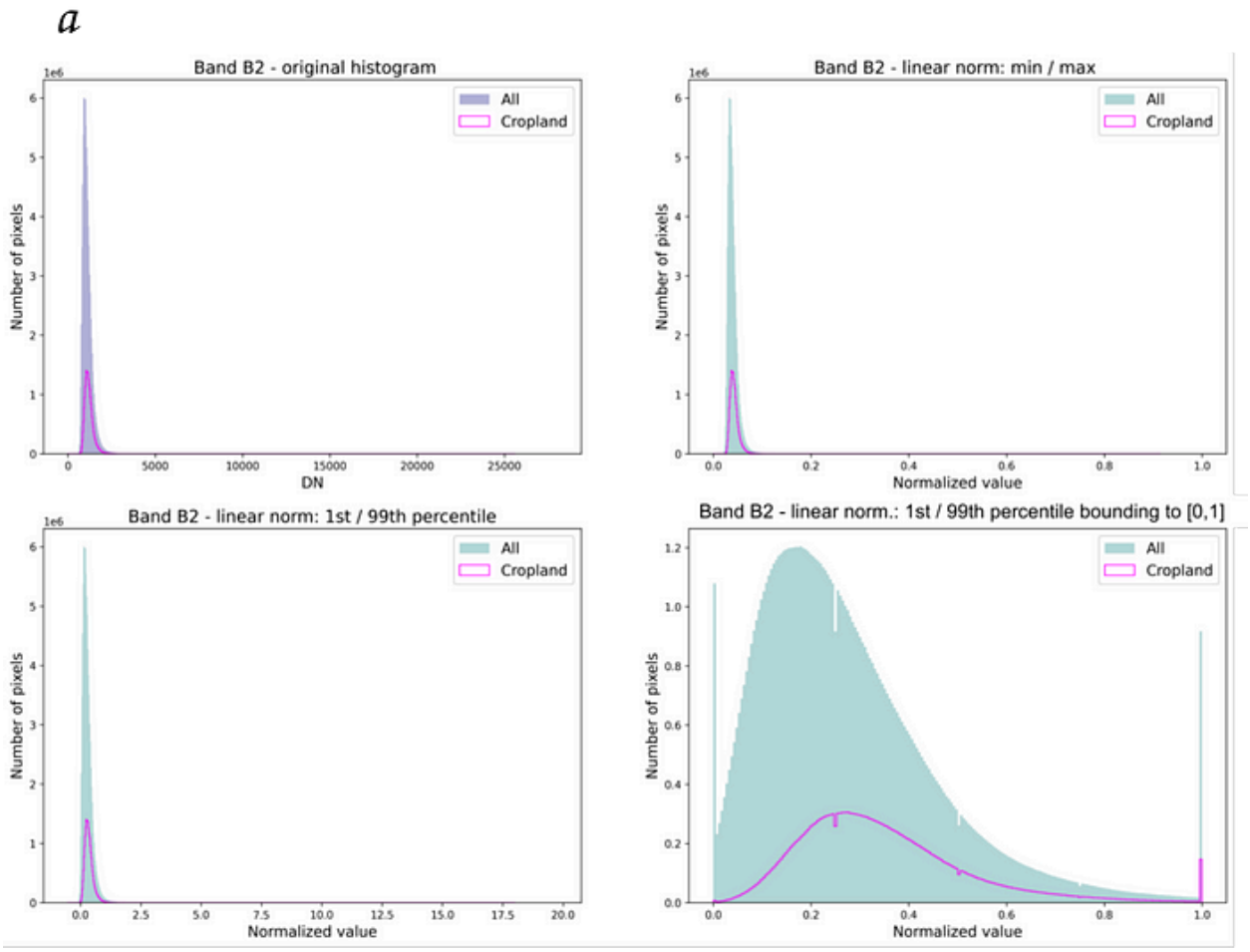
where $a$ and $b$ are the lower and upper limit of the resulting range and $c$ and $d$ are the lower and upper values of the input range. The resulting range of values was chosen to be between 0 and 1 and the scaling can be performed choosing different values for $c$ and $d$. One option is to simply take the *min* and *max* values of the input data. The problem, especially with long-tailed signals, is that a single outlier value can greatly affect the value of $c$ or $d$ which can result in a very unrepresentative scaling. A more robust approach is to select $c$ and $d$ at 1st and 99th percentile of the value histogram — this reduces the impact a few outliers can have on the scaling.

We tested the linear normalization with three different sets of parameters:

- $c$, $d$ as *min*, *max*
- $c$, $d$ as 1st and 99th percentile

- *c*, *d* as 1st and 99th percentile with the range bounded between 0 and 1

Some examples of band histogram transformations with the linear normalization are presented in Fig. 9.



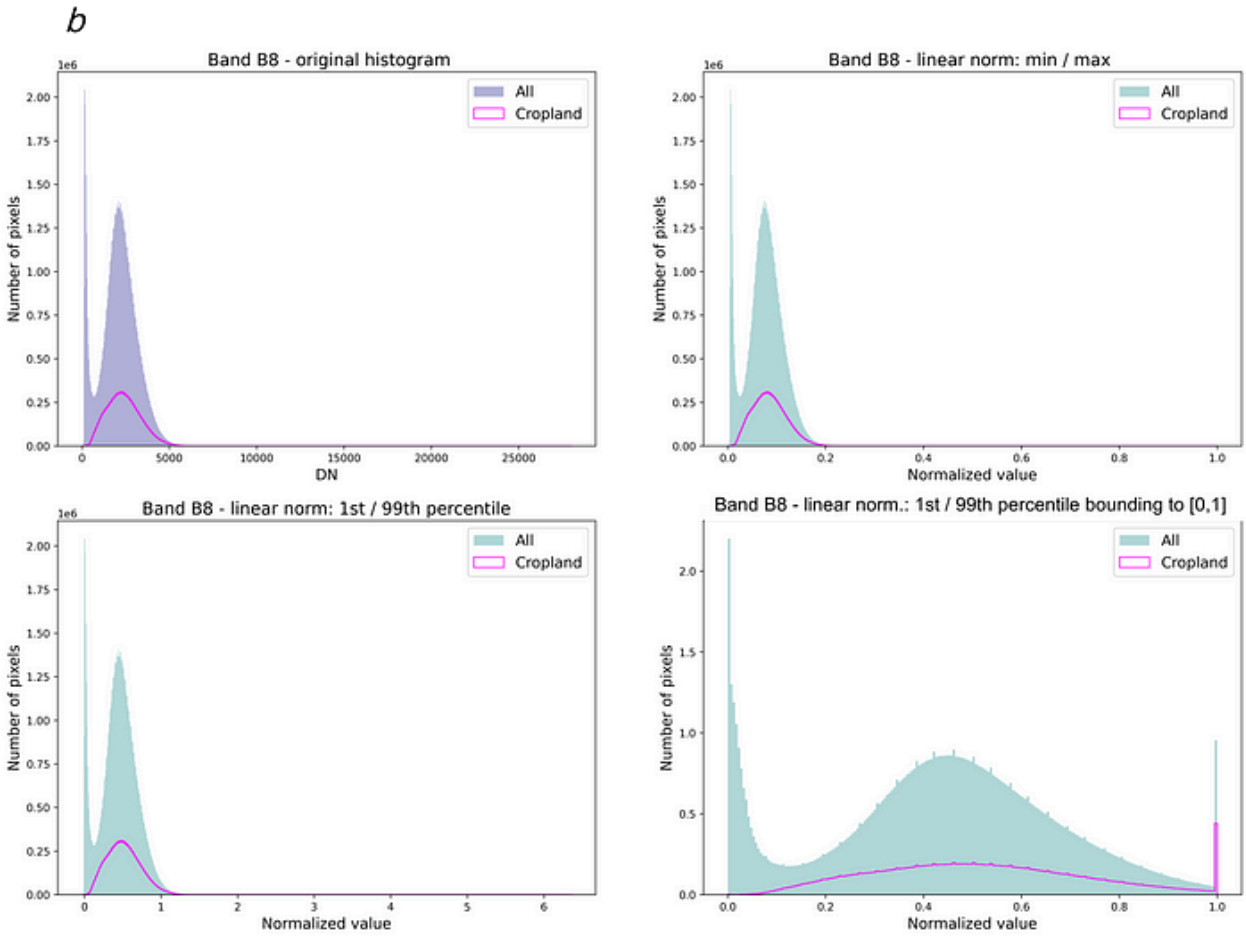



Figure 9: Examples of linear normalization with different sets of parameters for bands B2 (a) and B8 (b).

We see in Fig. 9 that with the *c* and *d* as *min*, *max* (top right), the range of the histogram values are transformed to the interval [0,1], but the long-tail shape of the histogram stays the same. This means that the effective range is reduced to a smaller interval, for instance, for band B2, between 0.02 and 0.08. On the other hand, with the *c, d* as 1st and 99th percentile with no bounding, the mid-part of the histogram is centered to [0,1] and

the lower and the upper 1 % of values are extended beyond this interval, retaining the long-tail shape of the distribution. With the use of bounds, the whole range of the histogram values lies within the interval [0, 1], where the lower and the upper percent of values are squeezed (condensed) in the extreme (first or last) histogram bins. Bounding can introduce some information loss. Since these transformations are linear, the shape of the original distribution is maintained.

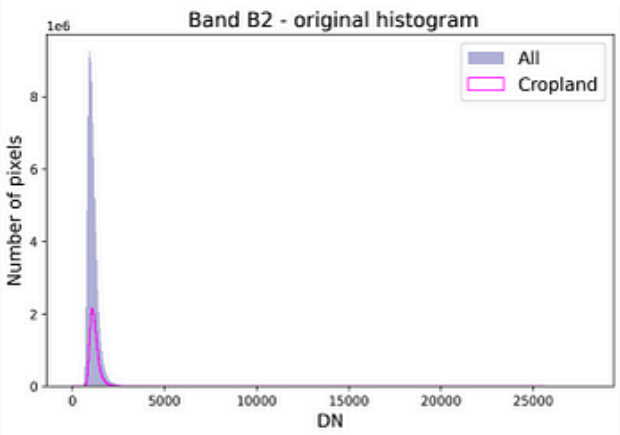
### Dynamic World normalization scheme

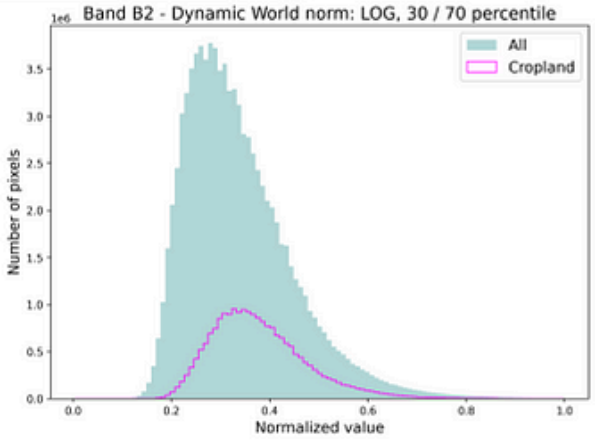
We tested the normalization scheme introduced in [3]. First, the log-transform is used on the original signal in order to deal with the imbalanced long-tailed values. Next, the 30th and 70th percentiles of the log-transformed signals are remapped to points on a sigmoid function. This bounds the resulting histogram range to the interval [0,1] and squeezes (condenses) the extreme values to a smaller range [3]. To experiment with the effect of the log-transform on the normalization, we additionally

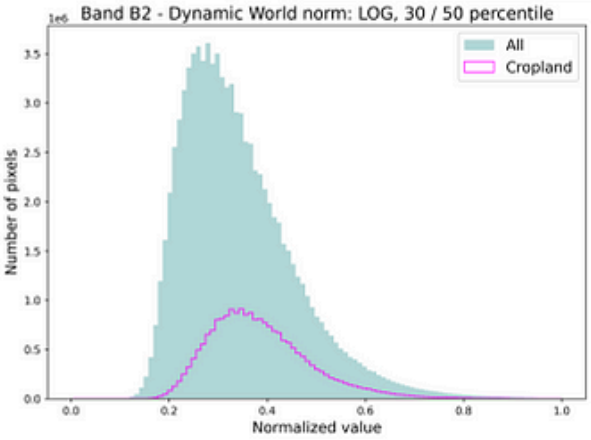
tested the scheme without the log-transform and also with different percentile values for the remapping. The four parameter sets used were:

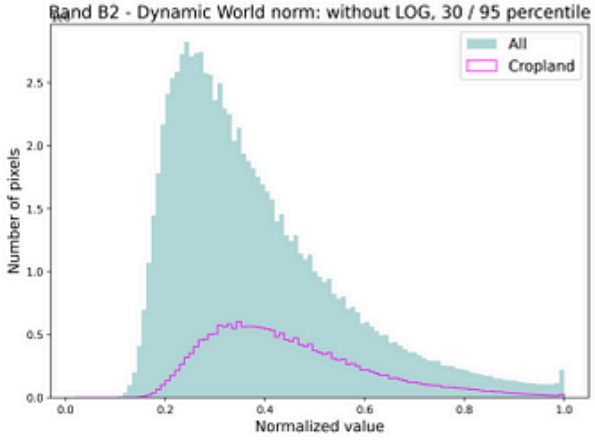
- 30th / 70th percentile with log transformation
- 30th / 50th percentile with log transformation
- 30th / 95th percentile without log transformation
- 20th / 95th percentile without log transformation

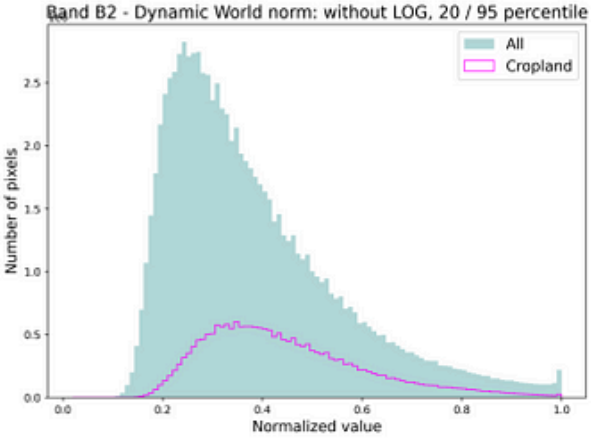
Examples of transformed histograms are presented in Fig. 10.


Band B2 - original histogram
1e6
All
Cropland
Number of pixels
DN


Band B2 - Dynamic World norm: LOG, 30 / 70 percentile
1e6
All
Cropland
Number of pixels
Normalized value


Band B2 - Dynamic World norm: LOG, 30 / 50 percentile
1e6
All
Cropland
Number of pixels
Normalized value


Band B2 - Dynamic World norm: without LOG, 30 / 95 percentile
All
Cropland
Number of pixels
Normalized value


Band B2 - Dynamic World norm: without LOG, 20 / 95 percentile
All
Cropland
Number of pixels
Normalized value

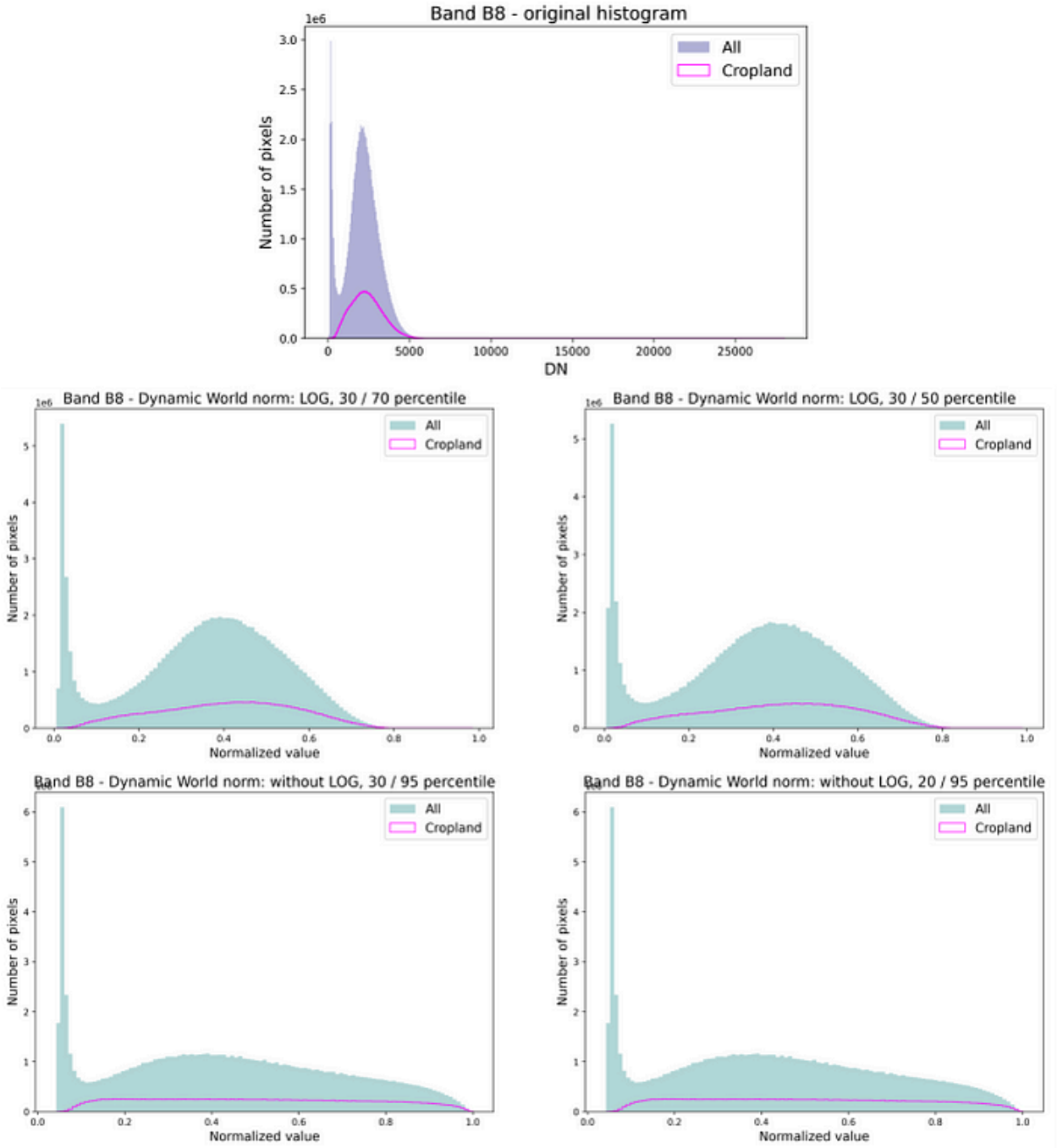



Figure 10: Effect of the Dynamic World normalization scheme with different parameters on the DN histograms of bands B2 and B8.

Comparing the log and no-log normalizations in Fig. 10, we see that log has the effect of retaining the flat tails of the histogram

and the no-log normalization squeezes (condenses) the values of the long histogram tail, similar to the bounding in linear normalization. Using different values for the mapped percentiles gives slightly different shapes of the resulting normalized histograms. In addition, these non-linear transformations change the original distribution of the band values.

### Histogram equalization

Unlike linear normalization, histogram equalization is a type of histogram modeling technique that applies a non-linear mapping between the input and the resulting signal and offers a way to obtain any desired histogram shape. Histogram equalization defines a mapping based on the cumulative histogram and re-maps the input (in our case long-tailed DN histogram) to a uniform distribution. It increases the contrast by spreading out band values to the entire output range. We used 40 000 bins for each band to construct the cumulative distribution of the Europe dataset to obtain the mapping function. Fig. 11 shows the

resulting uniform distributions obtained for each of the bands after histogram equalization of the dataset histograms.

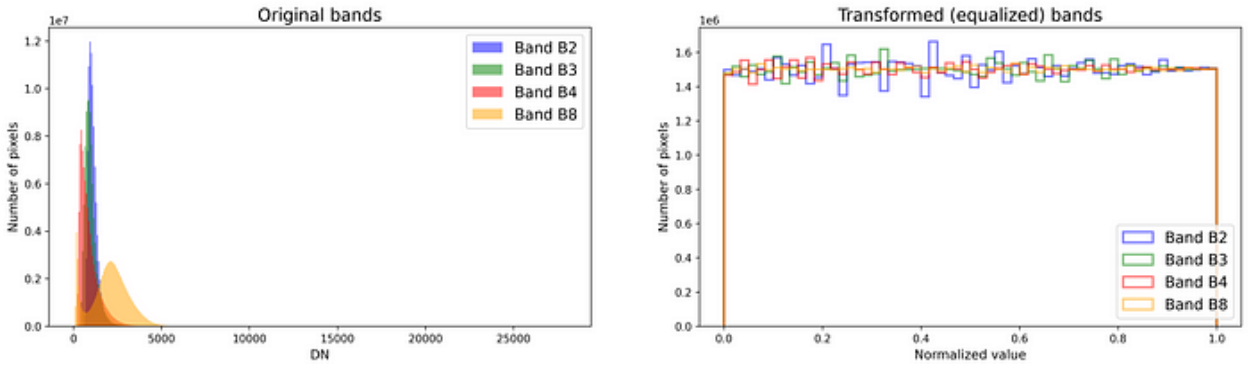
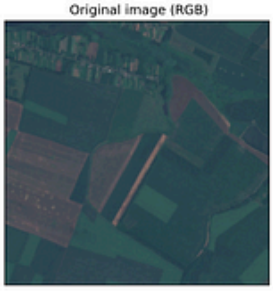



Figure 11: Uniform distributions obtained for each of the bands with histogram equalization.

## Visualization of the normalization methods

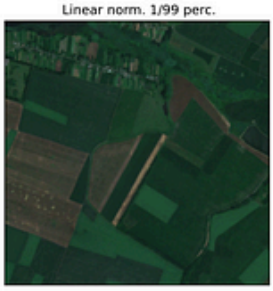
For comparison, we can visually present the effect of the three normalization methods. Fig. 12 shows some examples of Sentinel L1C RGB images (bands B4, B3, B2) and false color images (bands B8, B4, B3) under different normalization transformations. An advantage of using an output range between 0 and 1 is that we can visually assess and interpret the effect of normalization, which would be more challenging if we used, for instance, a range of [−1,1].

Original image (RGB)

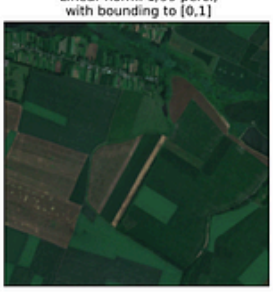


Linear norm. min/max

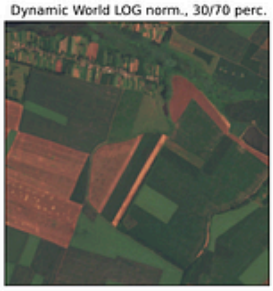
Linear norm. 1/99 perc.



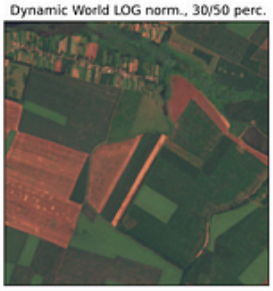
Linear norm. 1/99 perc.,
with bounding to [0,1]



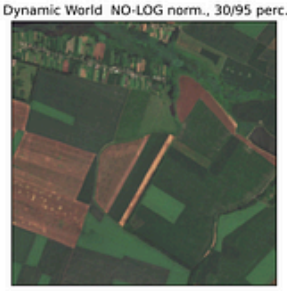
Dynamic World LOG norm., 30/70 perc.



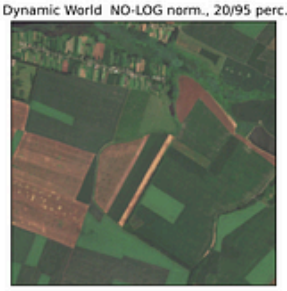
Dynamic World LOG norm., 30/50 perc.



Dynamic World NO-LOG norm., 30/95 perc.

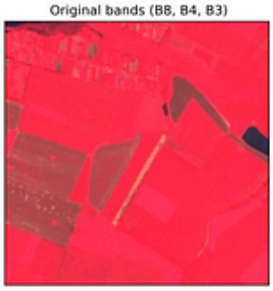


Dynamic World NO-LOG norm., 20/95 perc.

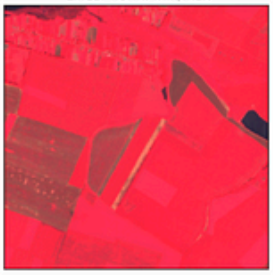


Histogram equalization

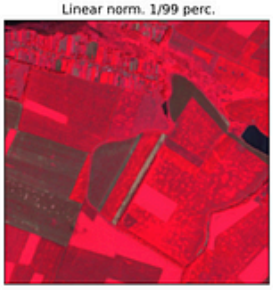
Original bands (B8, B4, B3)



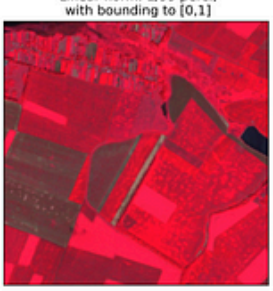
Linear norm. min/max



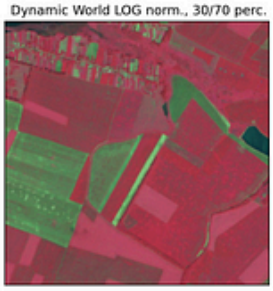
Linear norm. 1/99 perc.



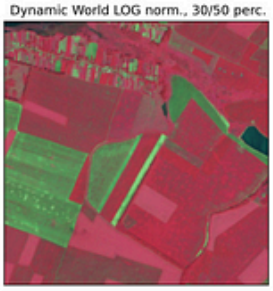
Linear norm. 1/99 perc.,
with bounding to [0,1]



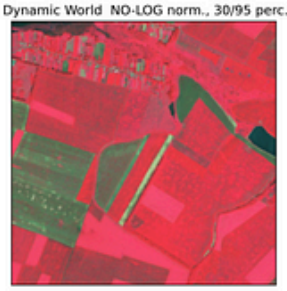
Dynamic World LOG norm., 30/70 perc.



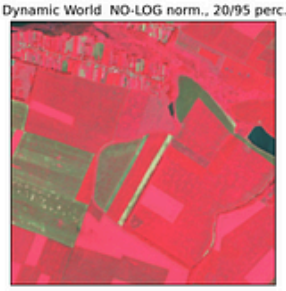
Dynamic World LOG norm., 30/50 perc.



Dynamic World NO-LOG norm., 30/95 perc.



Dynamic World NO-LOG norm., 20/95 perc.



Histogram equalization

Original image (RGB)

Linear norm. min/max

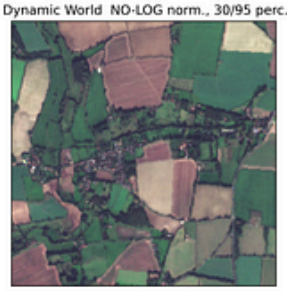
Linear norm. 1/99 perc.

Linear norm. 1/99 perc., with bounding to [0,1]

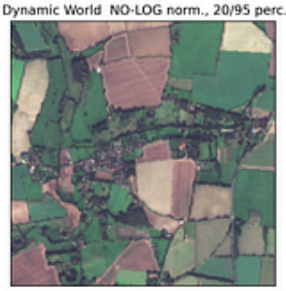
Dynamic World LOG norm., 30/70 perc.

Dynamic World LOG norm., 30/50 perc.

Dynamic World NO-LOG norm., 30/95 perc.



Dynamic World NO-LOG norm., 20/95 perc.



Histogram equalization

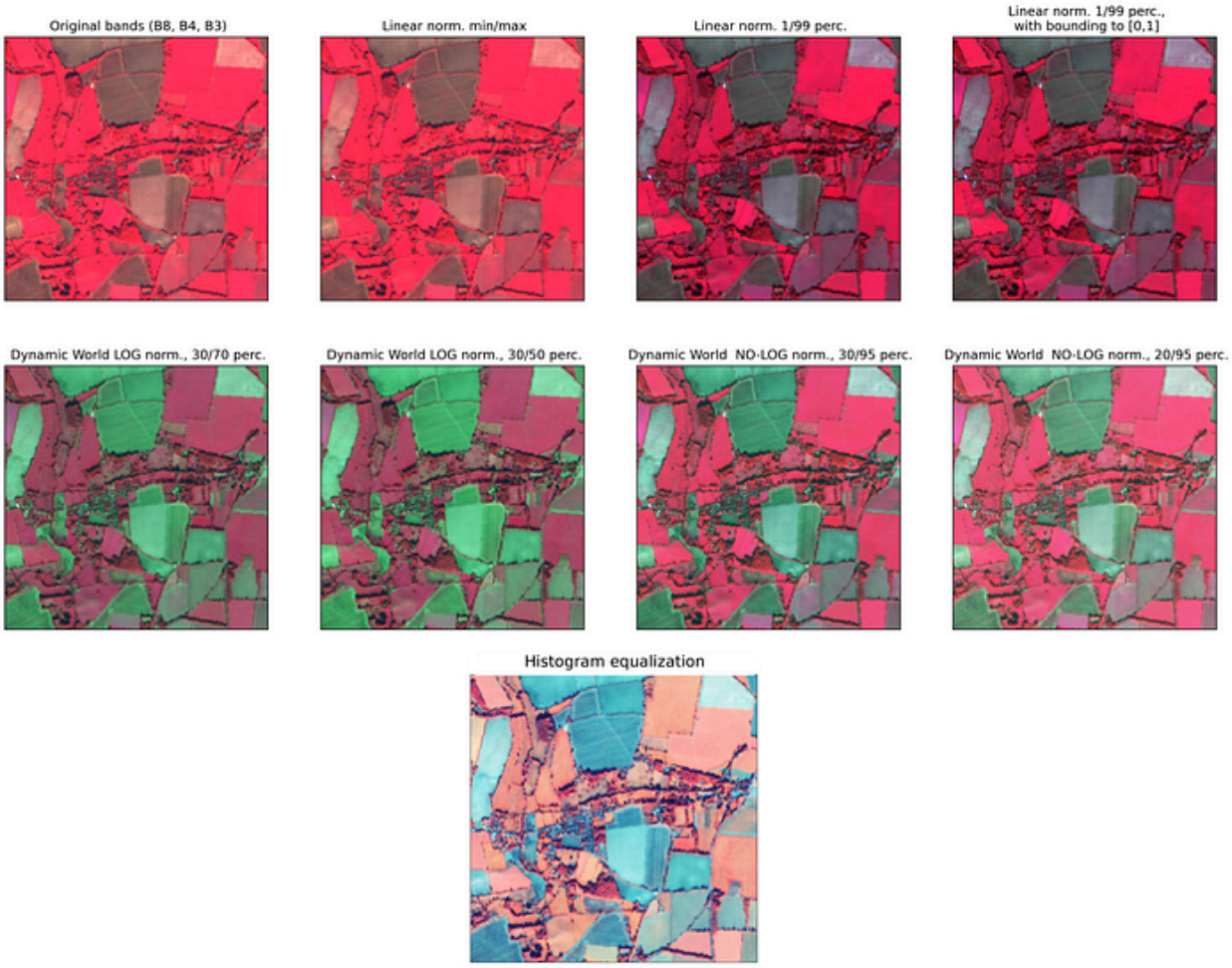



Figure 12: Examples of Sentinel L1C RGB images (bands B4, B3, B2) and false color images (bands B8, B4, B3) under different normalization transformations.

As we can see in Fig. 12, the linear normalization with *min* / *max* as *c* and *d* has no effect on the image appearance, as the values get shifted, but retain all the properties of the original band histograms. With the use of 1st and 99th percentiles as *c* and *d*, the contrast of the image is improved, both with and without bounding. The Dynamic World log and no-log transformations

have a more dramatic effect on images, since they change the shape of the band histograms. We see that the differences between vegetated and non-vegetated land are additionally enhanced, which is most apparent in the false color images showing the B8 band in red. This effect could be beneficial for many image processing or DL applications aiming at distinguishing vegetated from non-vegetated regions. Histogram equalization has the most dramatic visual effect on the images, which is not surprising as the histograms get re-shaped from a narrow-peaked and long-tailed histogram to a uniform distribution.

Although these transformations are interesting to a human eye, the effect on the network performance is not always predictable and straightforward. So we further investigated which of these transformations is the most appropriate for our field delineation application.

## Field delineation experiments

To explore the effect of different normalization methods on the training of the DL architecture, we set up a set of experiments with our existing field delineation algorithm. We used a subset of the ai4boundaries dataset for training of the model and a UNET with randomly initialized weights as a base model. The training of the model was performed over 4 epochs for each of the normalization methods.

The results of the experiments are first compared in terms of convergence of the network through evaluation of the losses for the training and the validation set. These are presented in Fig. 13.

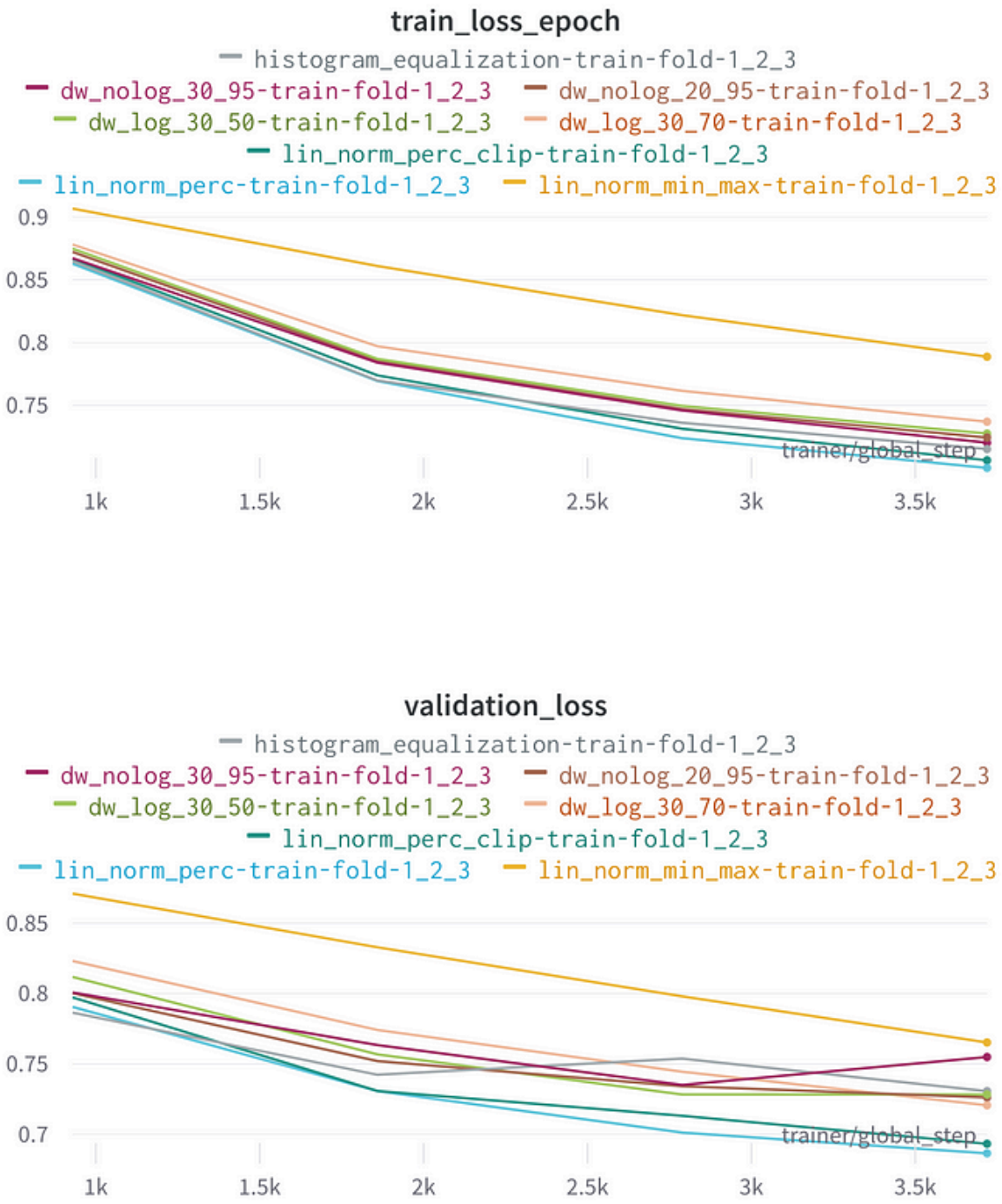



Fig. 13: Training and validation losses over 4 epochs for each of the normalization methods.

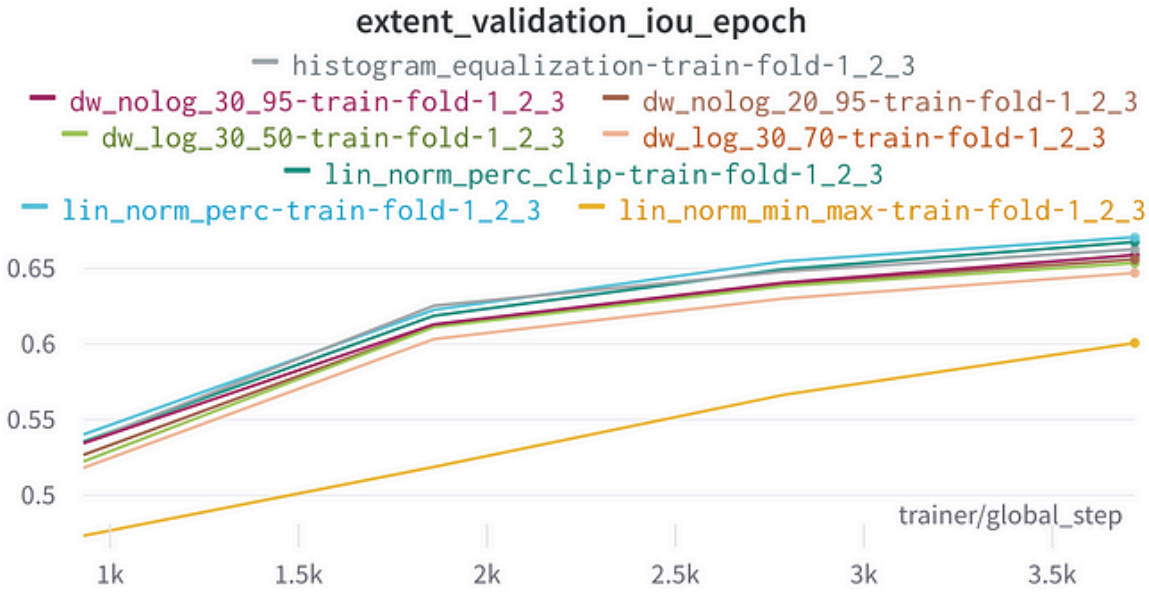
We see in Fig. 13 that the linear normalization with $c$, $d$ as 1st and 99th percentile gives the best results for training and

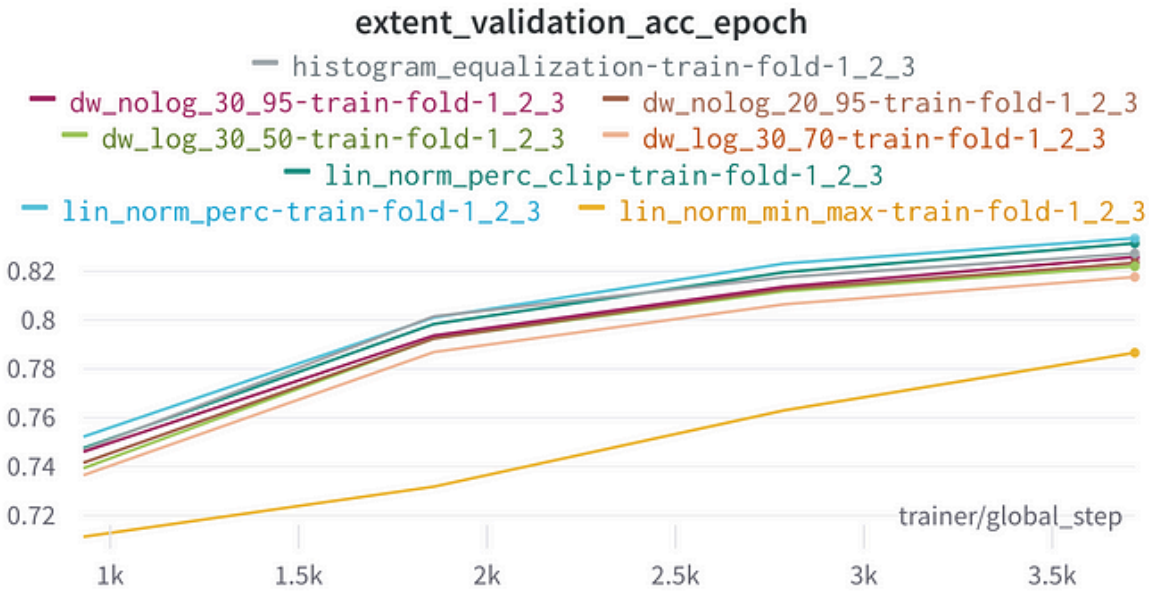
validation loss and is closely followed by its bounded version. The worst result is obtained using the linear normalization with *c, d* as *min* / *max*, although faster convergence can be observed compared to other methods. Histogram equalization and the non-linear Dynamic World normalization scheme in all its tested forms yield comparable results in this test run.

While the loss for the three linear normalization methods is comparable between training and validation datasets, this is not the case for the non-linear methods. For these, the validation loss is not monotonically decreasing, which might indicate less stable convergence.

The performance metrics in terms of intersection over union (IoU), accuracy and Matthews correlation coefficient (MCC) were also computed and are presented in Fig. 14. They show similar behavior and ranking of the normalization methods, with the linear normalization with *c, d* as 1st and 99th percentile giving

the best results and the linear normalization with *c, d* as *min / max* performing the worst.



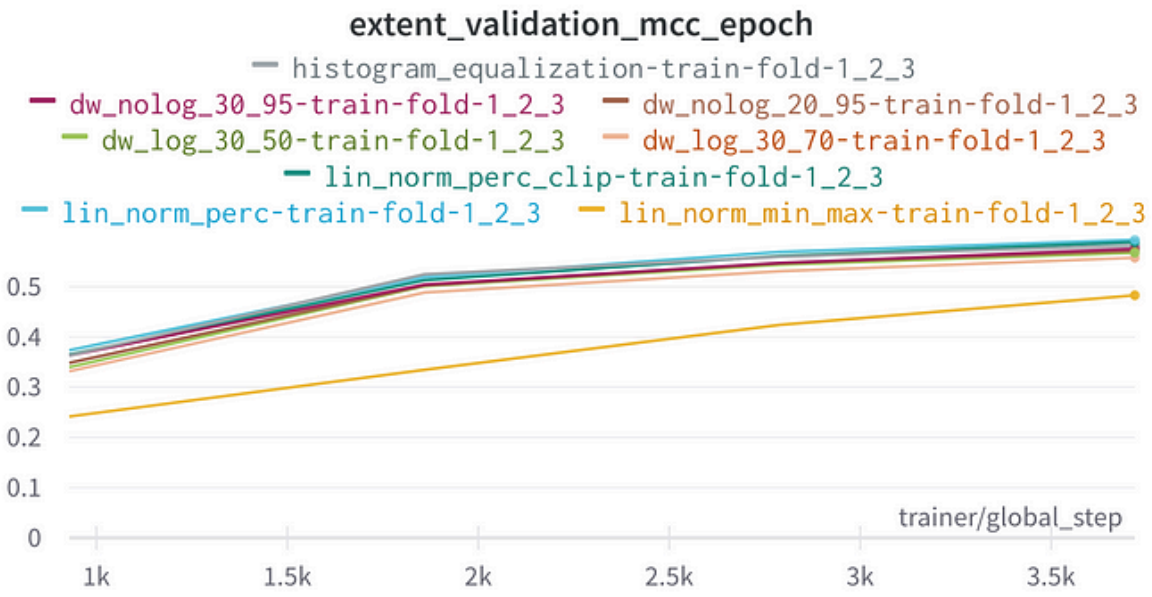



Fig. 14: Performance metrics for each of the normalization methods.

It's worth noting that 4 epochs may not be enough to draw conclusions on the final convergence state and possibly all methods might reach the same performance given enough time. However, our analysis already shows that some methods lead to a sharper and more stable convergence rate than others.

We see that the choice of the appropriate normalization method can affect both the convergence in the training and validation phase as well as the final results. The choice is not straightforward, though, as similar methods can give quite

different results as we see in the case of different types of linear normalization and vice versa; substantially different normalization can produce rather similar results in terms of network convergence and the final performance. The observed results of our experiments indicate that mapping the main part of histogram data into the interval [0, 1], but moving outlier values out of this interval (by the use of 1st and 99th percentile in the linear normalization) has a large positive effect on the network convergence and performance.

## Conclusions

We explored locational and temporal variability of satellite imagery band data and found that latitude is the most important locational parameter affecting the DN band histograms in our study area, probably because of its effect on climate and vegetation. Also, vegetation changes throughout the seasons contribute the most to the temporal variability. Both effects are

also reflected in the cropland histogram, which is especially important for field delineation purposes.

We chose three different types of histogram normalization with different parameters and investigated the impact on the resulting DN band histograms. Despite the fact that the visual effects can be quite dramatic in the case of non-linear band histogram transformations, convergence and performance of the network are not affected by these changes. Rather, the results of our field delineation experiments showed that even small modifications to the same method (e.g. how we transform outlier values) can have a much larger impact on the convergence and the performance of the model.

## References

[1] Ioffe, Sergey, and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal

covariate shift." *International conference on machine learning*. PMLR, 2015.

[2] Wiesler, Simon, and Hermann Ney. "A convergence analysis of log-linear training." *Advances in Neural Information Processing Systems* 24 (2011).

[3] Brown, Christopher F., et al. "Dynamic World, Near real-time global 10 m land use land cover mapping." *Scientific Data* 9.1 (2022): 1–17. https://doi.org/10.1038/s41597-022-01307-4

*remote sensing*

MDPI

*Article*

# Combining Satellite Imagery and a Deep Learning Algorithm to Retrieve the Water Levels of Small Reservoirs

**Jiarui Wu [1,2], Xiao Huang [1,*], Nan Xu [3], Qishuai Zhu [4], Conrad Zorn [5], Wenzhou Guo [2], Jiangnan Wang [1], Beibei Wang [2], Shuaibo Shao [6] and Chaoqing Yu [1,7]**

1 Key Laboratory of Agro-Forestry Environmental Processes and Ecological Regulation, School of Ecology and Environment, Hainan University, Haikou 570228, China; wujr143@hhu.edu.cn (J.W.)
2 Key Laboratory of Integrated Regulation and Resource Development on Shallow Lakes, Ministry of Education, College of Environment, Hohai University, Nanjing 210098, China
3 College of Geography and Remote Sensing, Hohai University, Nanjing 210098, China; 20220134@hhu.edu.cn
4 College of Computer and Information, Hohai University, Nanjing 210098, China
5 Department of Civil and Environmental Engineering, University of Auckland, Auckland 1010, New Zealand
6 School of Breeding and Multiplication (Sanya Institute of Breeding and Multiplication), Hainan University, Sanya 572025, China
7 Ministry of Education Key Laboratory for Earth System Modeling, Department of Earth System Science, Tsinghua University, Beijing 100190, China
* Correspondence: xiao.huang@hainanu.edu.cn

**Abstract:** There are an estimated 800,000 small reservoirs globally with a range of uses. Given the collective importance of these reservoirs to water resource management and wider society, it is essential that we can monitor and understand the hydrological dynamics of ungauged reservoirs, particularly in a changing climate. However, unlike large reservoirs, continuous and systematic hydrological observations of small reservoirs are often unavailable. In response, this study has developed a retrieval framework for water levels of small reservoirs using a deep learning algorithm and remotely sensed satellite data. Demonstrated at four reservoirs in California, satellite imagery from both Sentinel-1 and Sentinel-2 along with corresponding water level field measurements was collected. Post-processed images were fed into a water level inversion convolutional neural network model for water level inversion, while different combinations of these satellite images, sampling approaches for training/testing data, and attention modules were used to train the model and evaluated for accuracy. The results show that random sampling of training data coupled with Sentinel-2 satellite imagery was generally the most accurate initially. Performance is improved by incorporating a channel attention mechanism, with the average $R^2$ increasing by 8.6% and the average RMSE and MAE decreasing by 15.5% and 36.4%, respectively. The proposed framework was further validated on three additional reservoirs in different regions. In conclusion, the retrieval framework proposed in this study provides a stable and accurate methodology for water level estimation of small reservoirs and can be a powerful tool for small reservoir monitoring over large spatial scales.

**Keywords:** small reservoirs; water level; satellite imagery; convolutional neural network; attention mechanism

check for updates

**Citation:** Wu, J.; Huang, X.; Xu, N.; Zhu, Q.; Zorn, C.; Guo, W.; Wang, J.; Wang, B.; Shao, S.; Yu, C. Combining Satellite Imagery and a Deep Learning Algorithm to Retrieve the Water Levels of Small Reservoirs. *Remote Sens.* **2023**, *15*, 5740. https://doi.org/10.3390/rs15245740

Academic Editor: Prasad S. Thenkabail

Received: 4 October 2023
Revised: 2 December 2023
Accepted: 12 December 2023
Published: 15 December 2023

**Copyright:** © 2023 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/).

## 1. Introduction

Small reservoirs play an important role in the management and utilization of water resources, through water storage [1], power generation [2], and flood control [3], amongst others. Estimates suggest that globally there are currently over 800,000 small reservoirs (capacity < 10 million $m^3$), accounting for over 95% of all reservoirs worldwide [4]. With climate change and population growth [5], the number of small reservoirs is expected to continue to increase for reasons such as agricultural irrigation and to further help mitigate the impact of floods [6]. However, these reservoirs may also affect wider catchments by altering river connectivity and flow [7]. This in turn can impact sediment and nutrient

transport [8], ultimately leading to the degradation of downstream environments and ecosystems [9].

To better manage and operate these small reservoirs both efficiently and safely, access to timely and accurate knowledge of their hydrological dynamics is essential. However, there remains a lack of in situ observations collated over large geographic scales, such as in most global datasets (e.g., GeoDAR [10]) or satellite-based data products (e.g., ICESat-2). This can be partly attributed to confusion over reservoir ownership, insufficient funding to collect data, or various technical difficulties [11]. Such information gaps not only prevent obtaining reservoir operation conditions and determining optimal water management strategies [12], but also limit our wider scientific understanding of material transport, energy balance, and the resource environment within small reservoirs [13,14]. Therefore, there is an urgent need to improve our monitoring of hydrological parameters in small reservoirs, among which the water level is an important parameter in many contexts. In arid regions, reservoir levels directly correlate with the available water volume [15], whereas in flood-prone areas, monitoring reservoir levels becomes essential for appropriately adjusting various functions such as power generation, water supply, and flood relief [16].

Compared with costly and difficult-to-implement water level monitoring systems across large scales, satellite-based remote sensing technology can provide accurate and high-frequency information at much lower costs [17–19]. Currently, the use of satellite altimetry to obtain bathymetric data for inland water bodies is popular and reliable [20–23]. For example, Ryan, et al. [24] used ICESat-2 data to investigate water level changes in 3712 reservoirs worldwide, identifying different regional patterns of reservoir level changes based on water availability and management strategies. An issue raised here though is where existing altimetry satellites (e.g., ICESat-2) take measurements with large 3.3 km horizontal spacings [25]. Such wide spacings ensure that most of what we would consider to be small reservoirs may not be scanned or captured [26]. This ensures that the widespread application of satellite altimetry and associated methods for retrieval to monitor water depth in small reservoirs remains a challenge [27]. At the same time, the long repeat period of altimetry satellites (91 days for ICESat-2 [28]) prevented the establishment of a more complete water level monitoring sequence for small reservoirs. As a function of water depth, it is also a common and useful practice to establish area–storage–depth relationships for individual reservoirs [29,30]. Yigzaw, et al. [31], for example, iteratively selected the best geometry for a given reservoir from five possible regular geometries to establish area–storage–depth relationships to form a new global reservoir bathymetry dataset. A potential issue here being that some small reservoir geometries are more cylindrical in reality. For example, unlike the Lyons reservoir (LYS) shown in Figure 1, the surface area of the reservoir does not vary with storage volume or water level in the Gerle Lake reservoir (GLL). It is not feasible for these reservoirs to determine water level based on changes in water surface area.

In recent years, deep learning-based algorithms have been widely used to obtain information about water bodies [32–34] by automatically extracting features from satellite images or relevant input variables. This involves continuously adjusting its internal parameters to create intricate, non-linear functional relationships between these input variables and observed water level information. For example, Yang, et al. [35] implemented water level inversion of lakes on the Qinghai-Tibet Plateau based on common machine learning models (backpropagation, support vector machines, and random forest). Many studies leveraging deep learning models for water level inversion incorporate additional variables like precipitation, air temperature, and reservoir inlet and outlet flows, in addition to reservoir levels. The inclusion of these supplementary variables poses challenges, particularly for small reservoirs with limited monitoring capabilities. While the fusion of satellite imagery and deep learning has predominantly been employed for water body detection and classification [36,37], there is a recent trend among researchers to extend this combination to regression problems, specifically the estimation of water body bathymetry. Lumban-Gaol, et al. [38] used Sentinel-2 Level 2A images to provide reflectance values, establishing their

relationship to water depth. Najar, et al. [39] used Sentinel-2 imagery with a deep learning model to retrieve coastal bathymetry. Changes in water level impact the penetration and reflection of optical satellites in the water body. SAR imagery's water body detection mechanism relies on comparing backscattered signal intensities between the liquid surface, land, and vegetation. Deep learning could effectively capture the intricate relationship between these reflection/backscattering values and water level changes, thus enabling the direct inversion of the water level from satellite images. To the best of our knowledge, this approach has not been applied to the study of water levels in small reservoirs yet, which shows promise for the inversion of water levels in numerous small reservoirs.

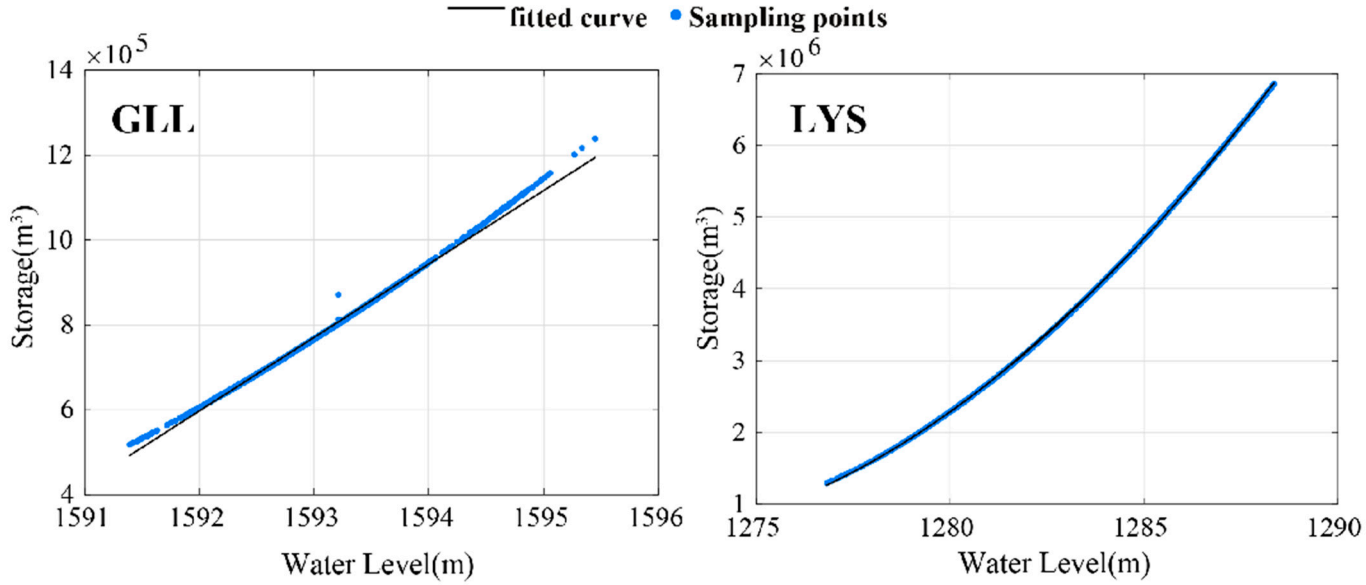



**Figure 1.** Water Storage–depth relationships for the Gerle Lake Reservoir (GLL) and Lyons Reservoir (LYS).

In response to this, this study aims to develop a practical method for retrieving water levels in small reservoirs by combining accurate deep learning inversion algorithms and high-resolution remote sensing images to retrieve and study the water level dynamics in small reservoirs. Specifically, in this paper, we will: (1) establish spatiotemporal relationships between remote sensing images and water levels using deep learning algorithms, (2) assess the impact of different data sources and sampling methods on a water level inverse model, and (3) apply coupling attention mechanisms to the model to test for improvements to our inversions.

## 2. Materials

Four small reservoirs located in California, United States, were selected for this study as shown in Figure 2. Due to its proximity to the ocean, the study area experiences higher precipitation during the winter and predominant dry conditions during the summer [40]. In recent years, California has experienced frequent droughts, leading to enormous water stress [41]. This has resulted in more frequent reservoir scheduling and an increased demand for water level monitoring compared to reservoirs in other regions. Similar to California, global small reservoirs are often situated in the mid to low latitudes. The selected reservoirs are distributed across different sub-catchments, with storage capacities ranging from $1.3 \times 10^6$ $m^3$ to $7.6 \times 10^6$ $m^3$ (Table 1). These are typically regarded as small reservoirs but with a range of uses including water storage, irrigation, and power generation.

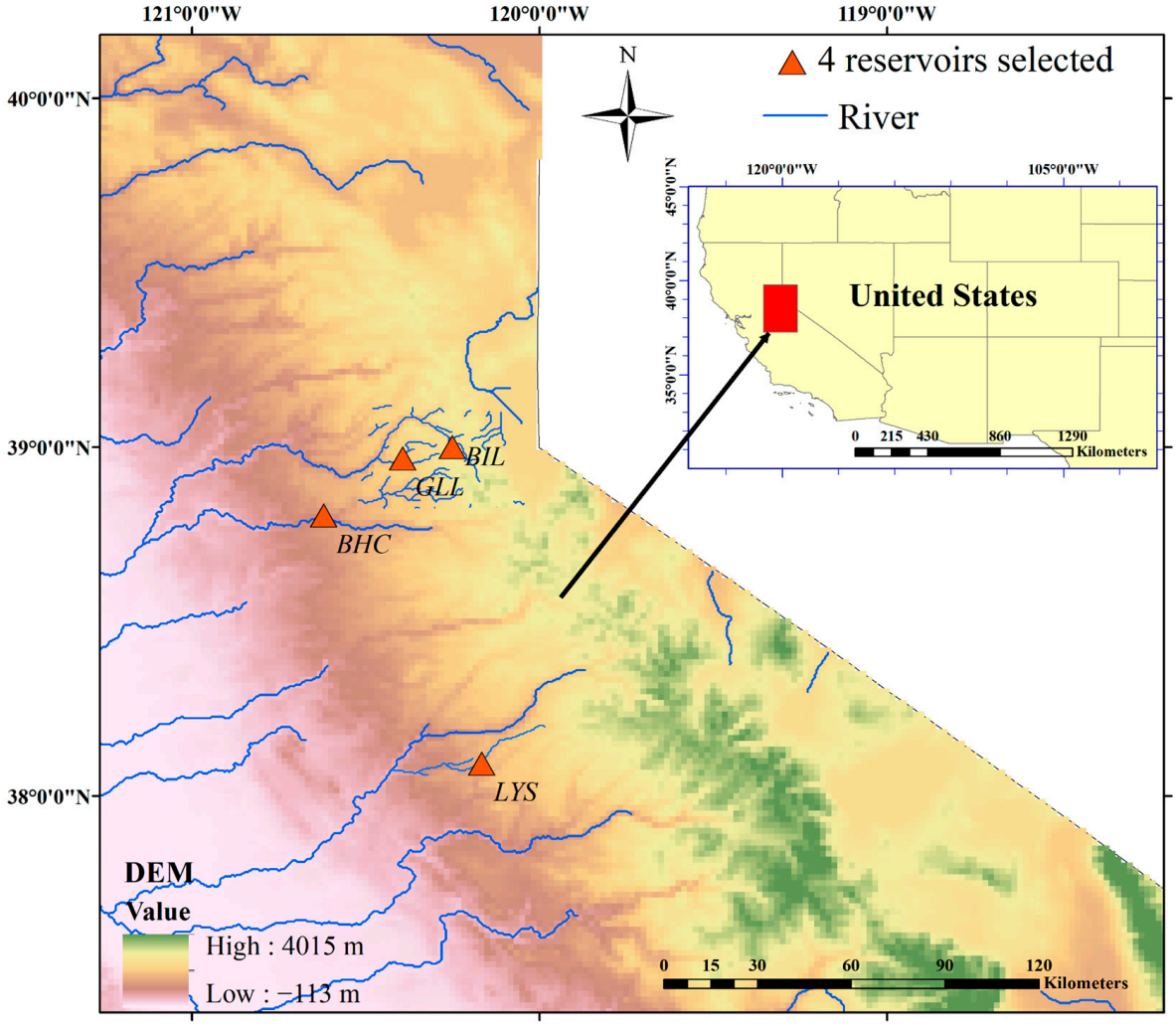



**Figure 2.** Locations of the studied reservoirs in California (US): Brush Creek (BHC), Gerle Lake (GLL), Buck Island (BIL), and Lyons Reservoir (LYS).

**Table 1.** Attributes of the California reservoirs.

| Name of Reservoir | Longitude | Latitude | Hydrologic Area | Capacity ($10^3$ $m^3$) |
|---|---|---|---|---|
| Brush Creek (BHC) | −120.62° | 38.80° | Sacramento River | 1887 |
| Buck Island (BIL) | −120.25° | 39.00° | Sacramento River | 1319 |
| Gerle Lake (GLL) | −120.39° | 38.97° | Sacramento River | 1480 |
| Lyons (LYS) | −120.16° | 38.09° | San Joaquin River | 7682 |

To further validate the effectiveness of our proposed framework for water level inversion in small reservoirs across diverse regions, we selected three additional reservoirs with climatic and geographic characteristics that differ significantly from those in California. These reservoirs include Mackay Creek Reservoir (MRR), characterized by a cold climate leading to freezing of the water surface in winter; Arrow Reservoir (ARR), situated in a region with a humid climate and abundant precipitation; and Costilla Reservoir (CTR), positioned at a higher elevation with a typical alpine climate. Refer to Table 2 for details about the reservoirs.

**Table 2.** Attributes of the validation reservoirs.

| Name of Reservoir | Longitude | Latitude | Location | Capacity ($10^3$ $m^3$) |
|---|---|---|---|---|
| Mackay Creek (MCR) | −110.19° | 49.71° | Alberta, Canada | 797 |
| Arrow (ARR) | −102.57° | 49.70° | British Columbia, Canada | 822 |
| Costilla (CTR) | −105.28° | 36.87° | New Mexico, USA | 4734 |

We obtained daily water level heights for the studied reservoirs in California between 2015 and 2022 from the California Data Exchange Center (https://cdec.water.ca.gov/dynamicapp/selectQuery (accessed on 1 August 2022)), operated by the California Department of Water Resources. The water level data between 2015 and 2022 for three additional reservoirs were sourced from the United States Geological Survey's National Water Information System (https://waterdata.usgs.gov/nwis (accessed on 1 December 2023)) and the Canada Water Agency (https://wateroffice.ec.gc.ca/ (accessed on 1 December 2023)).

The remote sensing image data were from the European Space Agency's (ESA) Sentinel-1 and Sentinel-2 satellites (https://scihub.copernicus.eu/ (accessed on 1 August 2022)). Launched in April 2014, Sentinel-1 is an active microwave remote sensing satellite that can provide imagery regardless of weather conditions such as clouds and rain, making it a valuable complement to optical remote sensing satellite. It offers a raw resolution of 20 × 22 m [42], via images in dual-polarisation mode (VV and VH bands). Sentinel-2 is a multispectral satellite launched in June 2015. It is a passive optical remote sensing satellite providing 13 spectra in the green (B3), red (B4), and near-infrared (B8) bands with a high resolution of 10 m × 10 m.

Reservoir information was obtained from the Global Surface Water database (https://global-surface-water.appspot.com/map (accessed on 1 August 2022)) with the Joint Research Center (JRC) Global Surface Water Mapping Layers used to identify the scope of the small reservoirs selected. These mapping layers provide a clear delineation of the extent and changes in water surface area and were further checked for alignment with satellite images from the Google Earth Engine platform (https://earthengine.google.com/platform/ (accessed on 1 August 2022)), as shown in Figure 3 for the GLL reservoir.

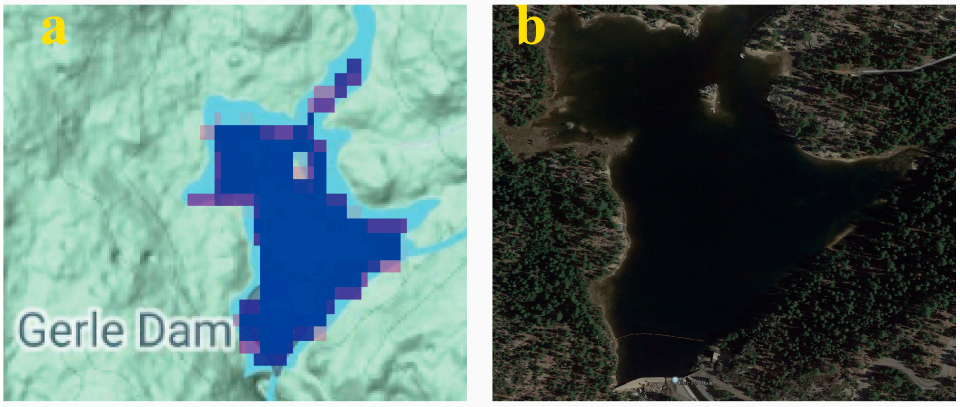



**Figure 3.** The Gerle Lake (GLL) reservoir as represented in (**a**) the Global Surface Water database and (**b**) the Google Earth Engine platform.

## 3. Methodology

Our methodology for processing raw data for use in the models is presented in Figure 4. Each step is further described in the following sections.

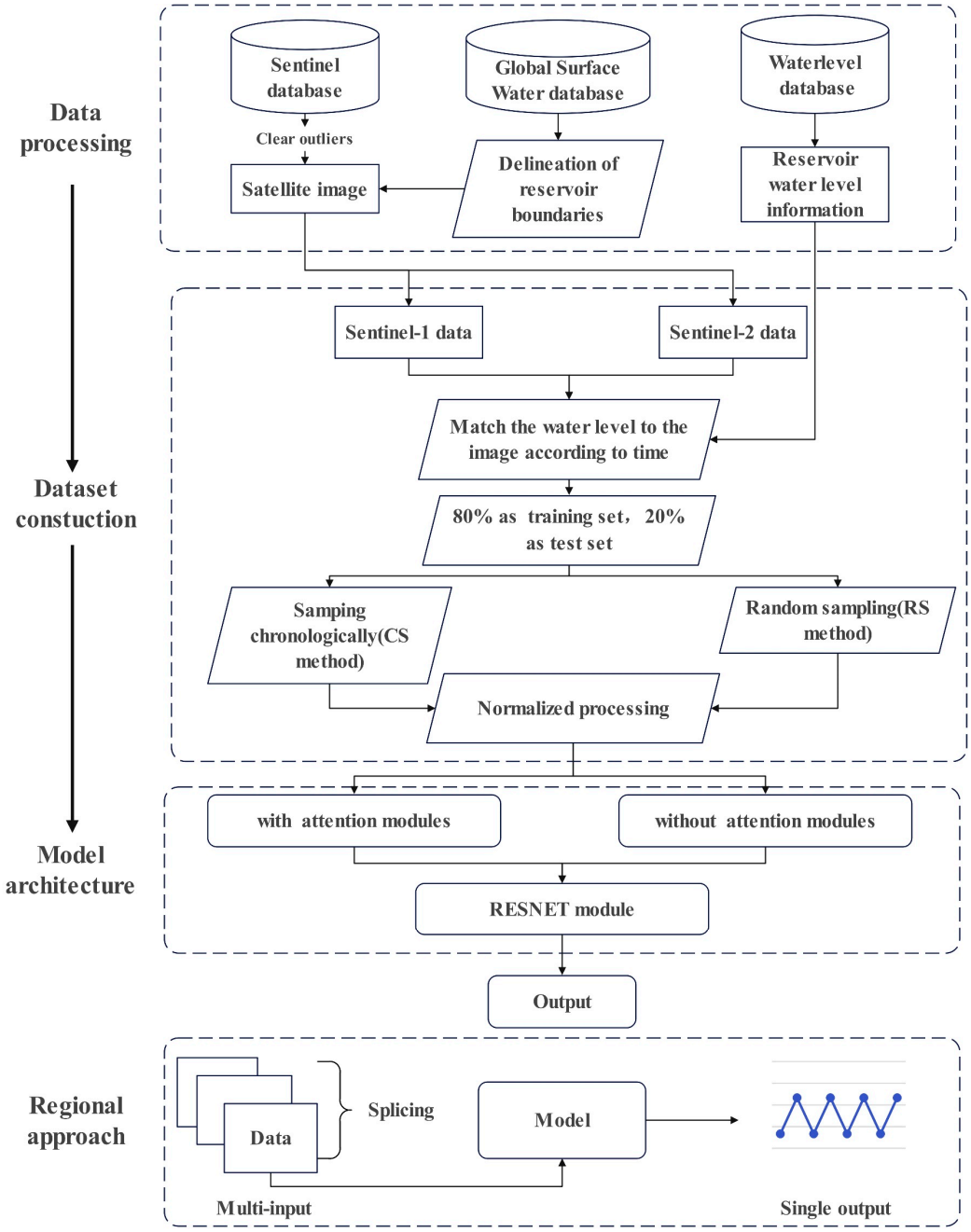



**Figure 4.** Flowchart of a deep learning model for predicting water levels in small reservoirs.

### *3.1. Data Processing*

The Sentinel series satellite data were acquired and processed using the Google Earth Engine platform. Sentinel-1 GRD data were utilized, and preprocessing on the GEE platform included updating orbital metadata, eliminating GRD boundary noise, removing thermal noise, radiometric calibration, and terrain correction. The Level-1 images were filtered based on polarisation mode, resulting in VV and VH. Sentinel-2 satellite imagery employed Level-2A data, pre-processed by ESA for radiometric calibration, atmospheric correction, etc. Using the QA60 band to mark and remove clouds, anomalous images were subsequently deleted, retaining only the B3, B4, and B8 bands. Finally, the remaining images were organized corresponding to collected water level data to produce an aligned time series.

### *3.2. Dataset Construction (Sampling)*

The observed water level data were split into subsets, with 80% for the training set and 20% for the test set. To explore the impact of both temporal correlation and peak values in the data on the model, two sampling approaches were employed to do this split. Chronological Sampling (CS) involved setting the first 80% of the time series and corresponding remote sensing images in chronological order as the training set, with the remaining 20% of the data as the testing set. Random Sampling (RS) involved sorting the reservoir water levels in descending order, dividing them into four quartile intervals, then randomly selecting 80% of the data from each interval as the training set and the remaining data as the test set. Both datasets from the two processing methods were normalized and the raw water level was linearly transformed so that the resultant values were mapped between 0 and 1. To help reduce the uncertainty of the results, the evaluation metrics obtained after running the model 20 times were averaged and used to evaluate the model.

### *3.3. Model Architecture*

After preprocessing the collected remote sensing and water level data, our constructed convolutional neural network (CNN) model could be applied to the water level inversion of the selected reservoirs. Here we select the RESNET34 model [43] as a deep learning model for water level inversion in reservoirs, with the discussed Sentinel satellite images used as inputs and the inversed reservoir water level as the output. RESNET34 includes cross-layer connections that reduce the dimensionality of the grid and minimize network parameters and computational complexity. The unique design of residual blocks in RESNET enables training deeper networks while achieving higher accuracy.

The structure of the convolutional neural network is shown in Figure 5. The benchmark model (Figure 5a) comprises a multi-band remote sensing image of a single reservoir being fed into a $7 \times 7$ convolutional layer with 64 output channels and a step size of 2 (Conv-1D). The second convolutional group consists of a $3 \times 3$ maximal pooling layer spanning 2 with a down-sampling padding of 1 (Max pool), and three residual modules (Layer1) connected to the previous convolutional group (Layer2–4) for down-sampling. These consist of four, six, and three residual modules (the backbone of each residual module is composed of two $3 \times 3$ convolutional layers), respectively. Finally, the water level was output through the average pooling layer and fully connected layer. The additional models of Figure 5b,c show the processed images are fed into the subsequent neural network after passing through the different attention modules. These are the Channel Attention Module (CAM; Figure 5b) and the Channel and Spatial Attention Module (CSAM; Figure 5c).

Some previous studies on remote sensing images using CNN have demonstrated differences in the weights of different bands of an image across various application scenarios. The utilization of attentional mechanisms allows for improved focus on these weights [44,45]. Given that various bands or polarization modes in remote sensing data may respond differently to changes in water level, the incorporation of these attention mechanisms into the models could be a valuable addition. The channel attention mechanism and spatial attention mechanism are introduced as additional layers preceding the RESNET34 network in the network structure. For input images from different satellite bands, global max pooling and global average pooling, based on width and height, are applied to generate the final feature map in the shared multilayer perceptron (MLP) network. The channel attention mechanism enhances the significance of channels influencing water level inversion (specifically, different bands of the same satellite) in the RESNET34 network while diminishing the importance of channels with a lesser impact on water level inversion [46]. Simultaneously, the spatial attention mechanism compresses the feature map corresponding to each individual band through global maximum pooling and global average pooling. Weight coefficients, obtained through a convolutional layer with an activation function, are multiplied with the input feature map to generate a new feature map, ensuring the model focuses on regions of the image that exert a greater impact on

water level inversion. A schematic illustrating both the channel attention mechanism and the spatial attention mechanism is presented in Figure 6.

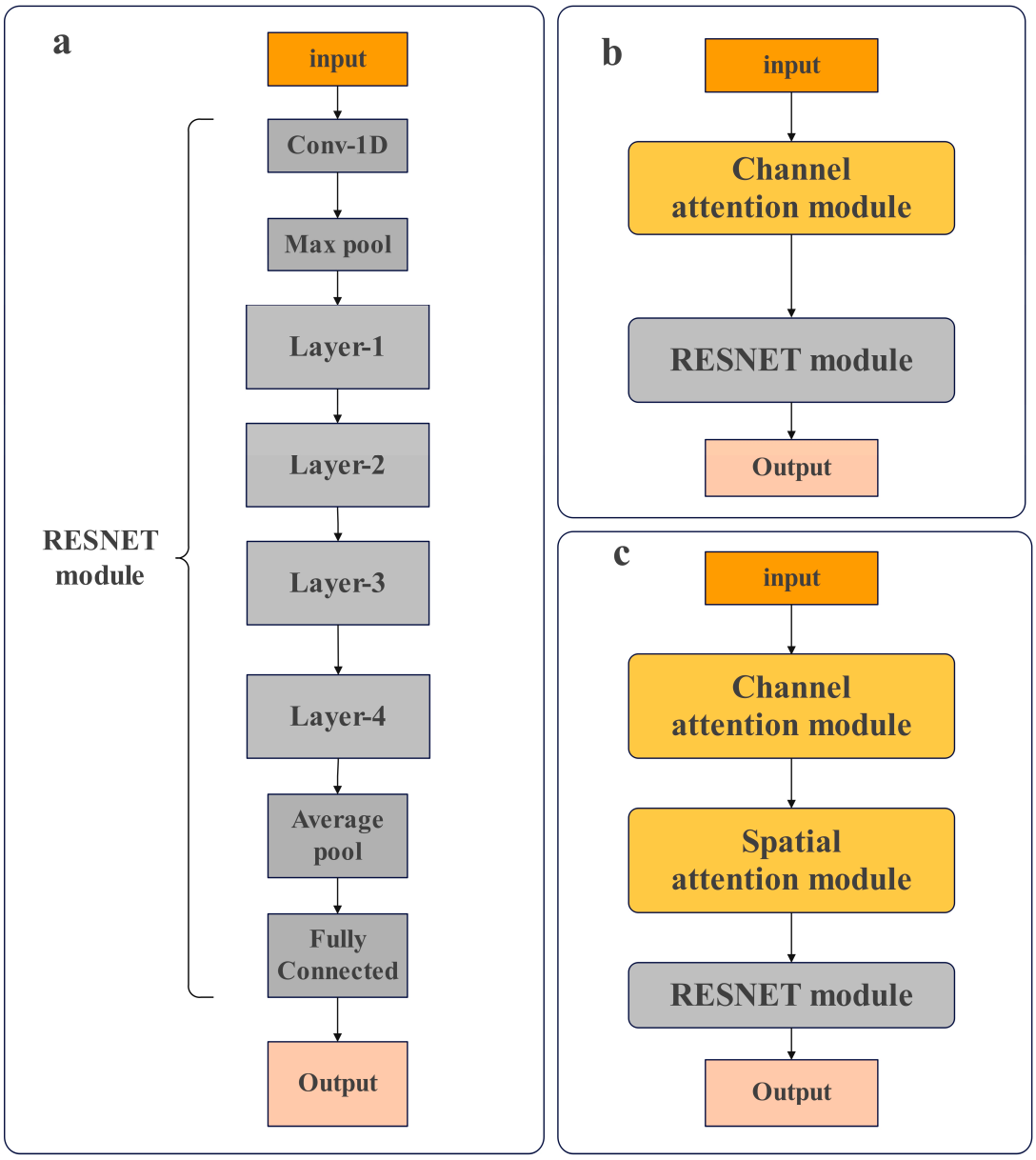



**Figure 5.** Convolutional neural network architecture. (**a**) CNN benchmark model, (**b**) CNN model with channel attention modules, and (**c**) CNN model with channel and spatial attention modules.

The model framework was developed using PyTorch lib [47] and trained on an NVIDIA GeForce RTX 3090 with the hyperparameter settings shown in Table 3.

**Table 3.** CNN model hyperparameter settings.

| Hyperparameter | Specified Values |
|---|---|
| Batch Size | 4 |
| Learning Rate | 0.0001 |
| Number of Epochs | 1500 |

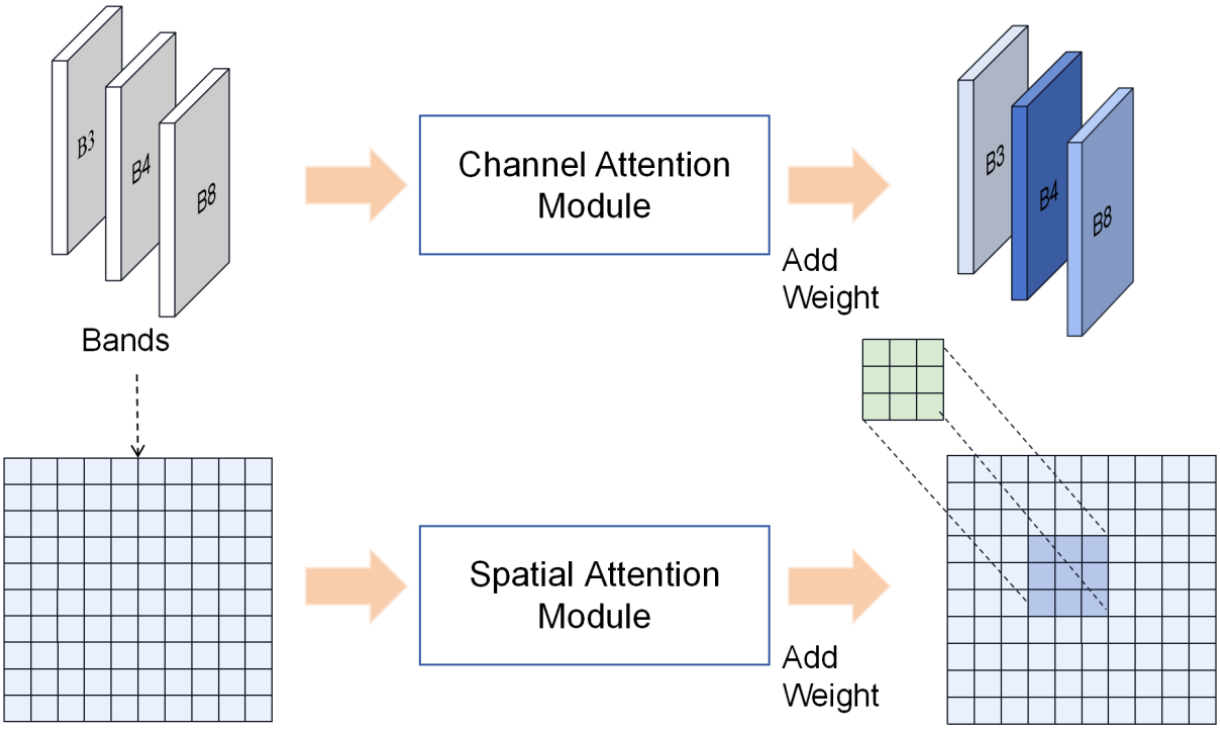



**Figure 6.** Schematic diagram of channel attention mechanism and spatial attention mechanism.

### *3.4. Model Regionalization Approach*

Efforts were undertaken to regionalize the model, which involved incorporating additional data from other reservoirs sharing similar characteristics within the region to estimate water levels in reservoirs with limited training data. For reservoirs with relatively short training datasets, data from one or more other similar reservoirs were integrated to augment the training dataset. To minimize the impact of variations in datum height among different reservoirs, the original water level data were converted into water depth data before being input into the model.

### *3.5. Model Performance Indicators*

In this study, we used three evaluation metrics commonly used for regression inversion to quantify model performance: R-square ($R^2$) given as Equation (1), root-mean-squared error (RMSE) as Equation (2), and mean absolute error (MAE) given as Equation (3). $R^2$ is employed to evaluate the model's capacity to capture the variability in the observed data. Higher values denote a better fit, with $R^2$ = 1 indicating a perfect fit. RMSE is similar to MAE in that it provides a measure of the overall accuracy of the model, with lower values indicating better model accuracy, when a value of 0 indicates an exact match between observed and simulated values. However, RMSE gives greater weight to larger errors and penalizes larger discrepancies.

$$R^2 = 1 - \frac{\sum_{t=1}^{NT}(H_{obs}(t) - H_{sim}(t))^2}{\sum_{t=1}^{NT}\left(H_{obs}(t) - \overline{H_{obs}}\right)^2} \quad (1)$$

$$RMSE = \sqrt{\frac{1}{NT}\sum_{t=1}^{NT}(H_{obs}(t) - H_{sim}(t))^2} \quad (2)$$

$$MAE = \frac{1}{NT}\sum_{t=1}^{NT}|H_{obs}(t) - H_{sim}(t)| \quad (3)$$

where $H_{obs}(t)$ is the observed variable; $H_{sim}(t)$ is the computed variable; $NT$ is the total number of observations; and $\overline{H_{obs}}$ is the overall mean observed variable.

## 4. Results

Our results demonstrating model performance are structured in six contexts: a comparison of sampling approaches (Section 4.1), a comparison of imagery data sources (Section 4.2), the quantified optimal combinations of sampling and imagery for each reservoir (Section 4.3), the model performance with attention mechanisms (Section 4.4), the model performance of regionalized approach (Section 4.5), and finally, the validation of the model's application in different regions (Section 4.6).

### *4.1. Comparative Performance of Sampling Approaches*

Here we consider the two sampling approaches: CS (chronological) and RS (random) as defined in Section 3.2. The RS approach outperformed CS across all three average $R^2$, average RMSE, and average MAE metrics. Additional analyses were conducted on the information in Table 4; the average $R^2$ values of the models using the RS method and the models using the CS method are 0.73 and 0.26, the average RMSE values are 0.65 and 0.84, and the average MAE values are 0.41 and 0.59. Figure 7 shows that the models trained using the RS method consistently exhibited better metric values compared to those trained using the CS method, varying depending on the combined data sources. The performance of models trained with both sampling methods was close for the LYS, BIL, and GLL reservoirs. However, the CS method was not applicable to the BHC reservoir, resulting in negative metric values. This could be related to the fact that the BHC historical water level sequences were not distributed consistently in the training and test sets. Overall, the use of the RS approach in determining training/testing data ensures better model robustness compared to the CS method, mainly because the RS method improves the representativeness of the training data and allows the model to learn from the peak water level data.

**Table 4.** Model performance across reservoirs (BHC, BIL, GLL, LYS) and different combinations of satellite data (Sentinel-1 and Sentinel-2) and sampling approaches (Random and Chronological).

| Reservoir and Sampling Approach | $R^2$ | RMSE | MAE |
|---|---|---|---|
| BHC-1-R | 0.08 | 1.40 m | 1.07 m |
| BHC-2-R | 0.54 | 0.86 m | 0.63 m |
| BHC-1-C | −1.1 | 1.22 m | 0.97 m |
| BHC-2-C | −0.24 | 1.00 m | 0.84 m |
| BIL-1-R | 0.72 | 0.43 m | 0.29 m |
| BIL-2-R | 0.80 | 0.37 m | 0.24 m |
| BIL-1-C | 0.81 | 0.32 m | 0.24 m |
| BIL-2-C | −0.05 | 0.29 m | 0.22 m |
| GLL-1-R | 0.13 | 0.57 m | 0.43 m |
| GLL-2-R | 0.75 | 0.28 m | 0.21 m |
| GLL-1-C | −0.00 | 0.53 m | 0.42 m |
| GLL-2-C | 0.77 | 0.26 m | 0.20 m |
| LYS-1-R | 0.60 | 1.65 m | 1.20 m |
| LYS-2-R | 0.84 | 1.10 m | 0.59 m |
| LYS-1-C | 0.72 | 1.64 m | 1.11 m |
| LYS-2-C | 0.59 | 1.66 m | 1.02 m |

For example, BHC-1-R is the Brush Creek Reservoir using Sentinel-1 imagery and Random Sampling.

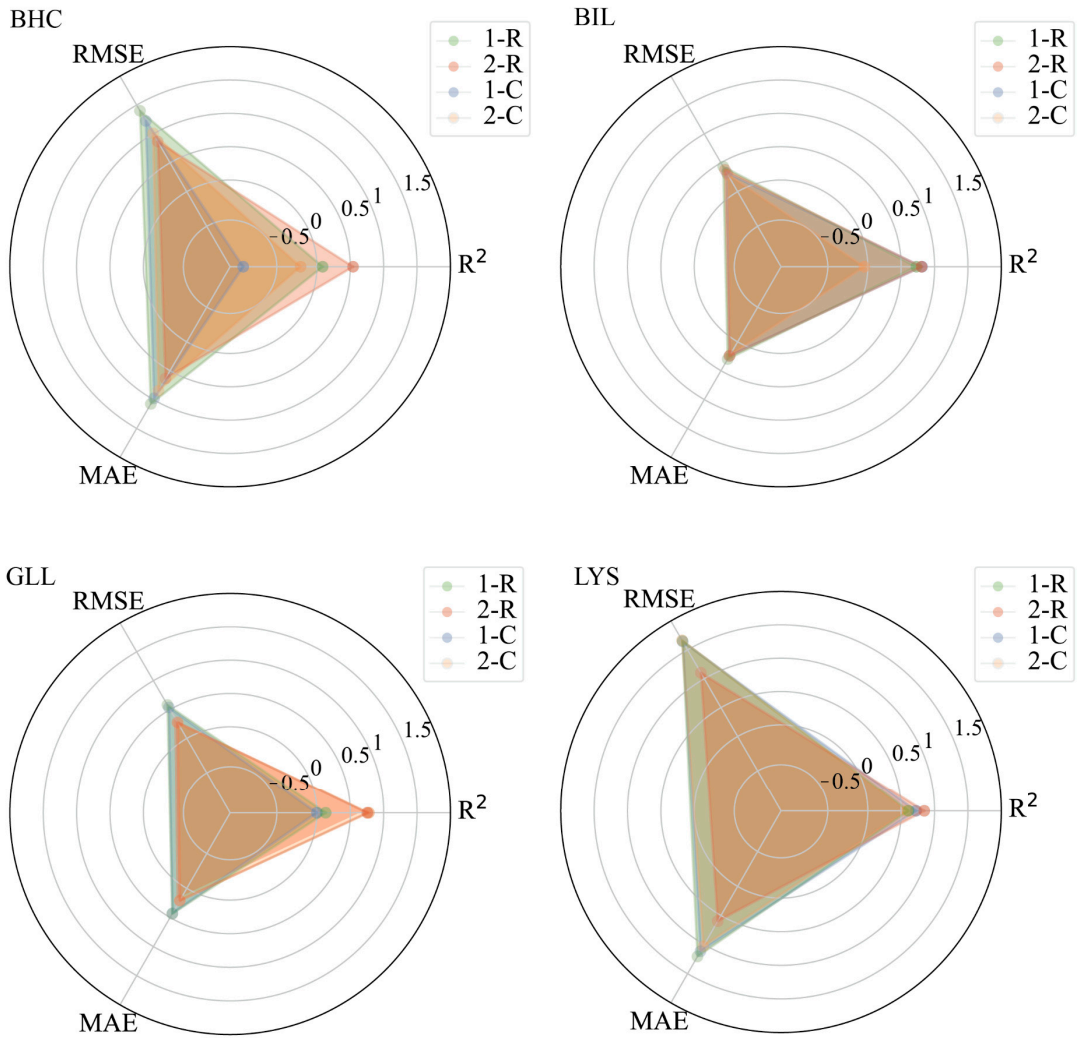



**Figure 7.** Radar plots showing model performance across three evaluation metrics for the studied reservoirs under the following input data combinations: (1-R) Sentinel-1 satellite data with random sampling, (2-R) Sentinel-2 satellite data with random sampling, (1-C) Sentinel-1 satellite data with chronological sampling, and (2-C) Sentinel-2 satellite data with chronological sampling.

### *4.2. Comparative Performance of Imaging Data Sources*

The use of Sentinel-1 data was not suitable for the GLL and BHC reservoirs where the maximum $R^2$ value of the models obtained from either sampling approach did not exceed 0.2 (Table 4). This may be attributed to the fact that dense vegetation around these two reservoirs affects the penetration and reflection of readings from the Sentinel-1 satellite. We have excluded these in our comparison between data sources. For the remaining LYS and BIL reservoirs, when sampled using the RS method, the model using Sentinel-2 data achieved better results at both LYS and BIL. When using the CS method, the opposite is observed and the model using Sentinel-1 data achieves better results than the model using Sentinel-2 data in the inversion. One of the explanations for the two opposite conclusions is that Sentinel-1 uses active remote sensing using synthetic aperture radar (SAR), which is unaffected by cloud cover, and thus increases the image availability. This enables the model to better learn the mapping relationship between remote sensing imagery and water level data observed over a range of changes seen in a longer time series.

### *4.3. Optimal Combination of Data Source and Sampling Approach*

We have further examined the best combination of satellite data sources and sampling approaches for each reservoir. Specifically, LYS and BHC reservoirs achieved the highest

inversion accuracy using a combination of Sentinel-2 data and the RS method. At the GLL reservoir, the best results were achieved using a combination of Sentinel-2 data and the CS method, whereas the BIL reservoir predictions were the best when using the Sentinel-1 data combined with the CS approach.

Focusing on the performance of the models on time-series inversion under the optimal combinations, Figure 8 compares the observed and predicted water levels and their associated relative errors. We see the relative errors of LYS and BHC inversions are larger, with the same random sampling approach and larger ranges of observed water levels throughout the time series. LYS had 9 days in the test set where the inversion water level error exceeded 1 m, with an average inversion error value of 0.5 m, while BHC had 13 days where the inversion error exceeded 1 m, with an average inversion error value of 0.62 m. The remaining reservoirs, GLL and BIL, demonstrated the effectiveness of using temporal sequential data to capture smaller fluctuations in water level, with both reservoirs having only 1 day in which the inversion water level error exceeded 1 m, with average inversion error values of 0.2 and 0.23 m, respectively. Models using the CS method exhibit an advantage in inversing water level time series. This is due to the fact that the CS method improves the model's understanding of the link between satellite imagery and water level data, as mentioned above.

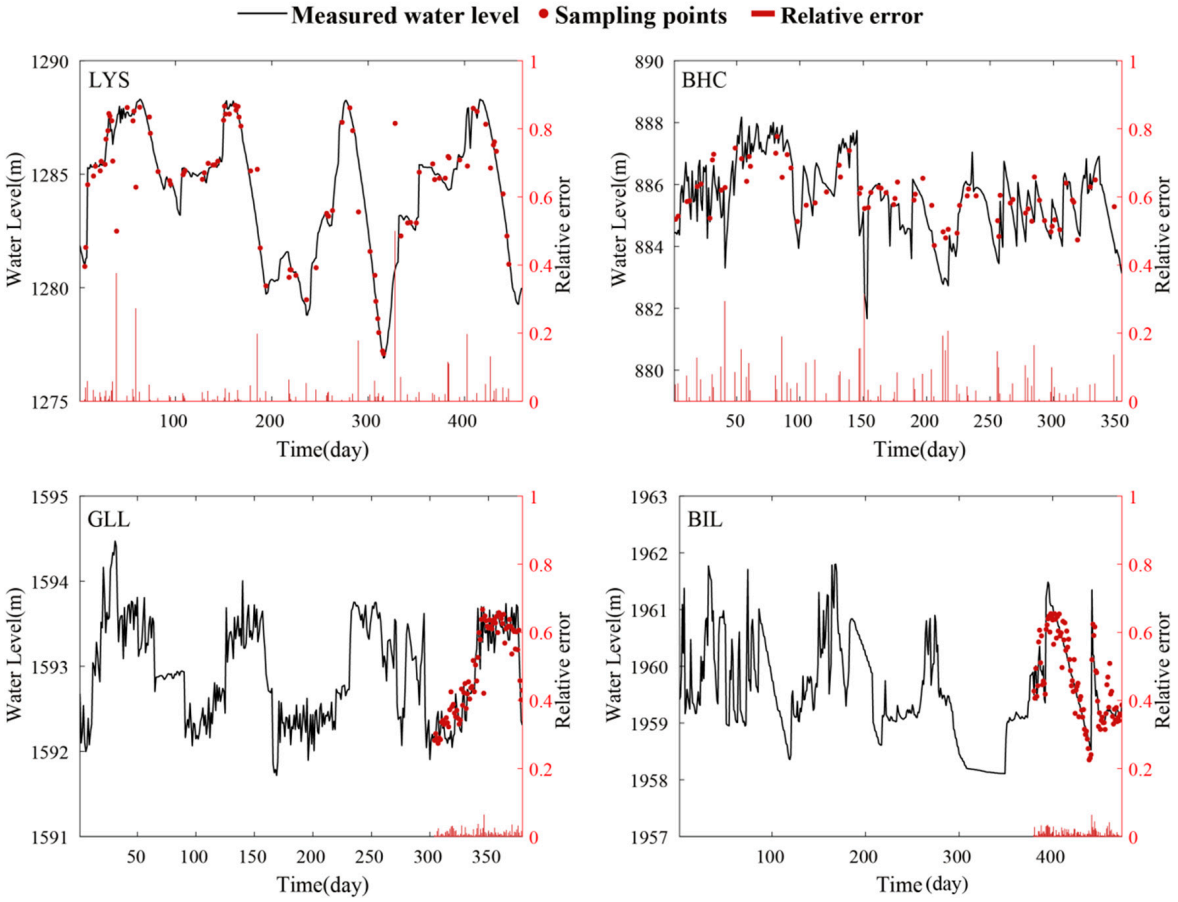



**Figure 8.** Comparison of (median) inversion results with measured water levels.

### *4.4. Model Performance with the Addition of Different Attention Mechanisms*

Figure 9 provides a comparison of results from the three models without the addition of attention module (Figure 9a), with the addition of CAM (Figure 9b), and with the addition of CSAM (Figure 9c). Both visually and across our given performance metrics ($R^2$, RMSE, MAE) we see that with the addition of the attention mechanism, the model's inversion results can be improved. The model with only the CAM added (Figure 9b) has

the highest inversion accuracy of the four reservoirs, with an average improvement of 8.6% in the $R^2$ value and an average reduction in both the RMSE (21.8%) and MAE (23.8%) when compared to the inversions before any attention mechanism was added. However, the performance at the BHC reservoir is still poor, as indicated by its comparatively low maximum $R^2$ value of 0.62 and corresponding RMSE and MAE values of 0.73 m and 0.53 m, respectively. This could be mainly attributed to the fact that the training data at the BHC reservoir were the smallest among the four reservoirs ($n$ = 275). Meanwhile, the significant height differences in the terrain surrounding BHC may give rise to occlusion effects, resulting in images with pronounced shadows. This poses a challenge in effectively extracting feature values through the model.

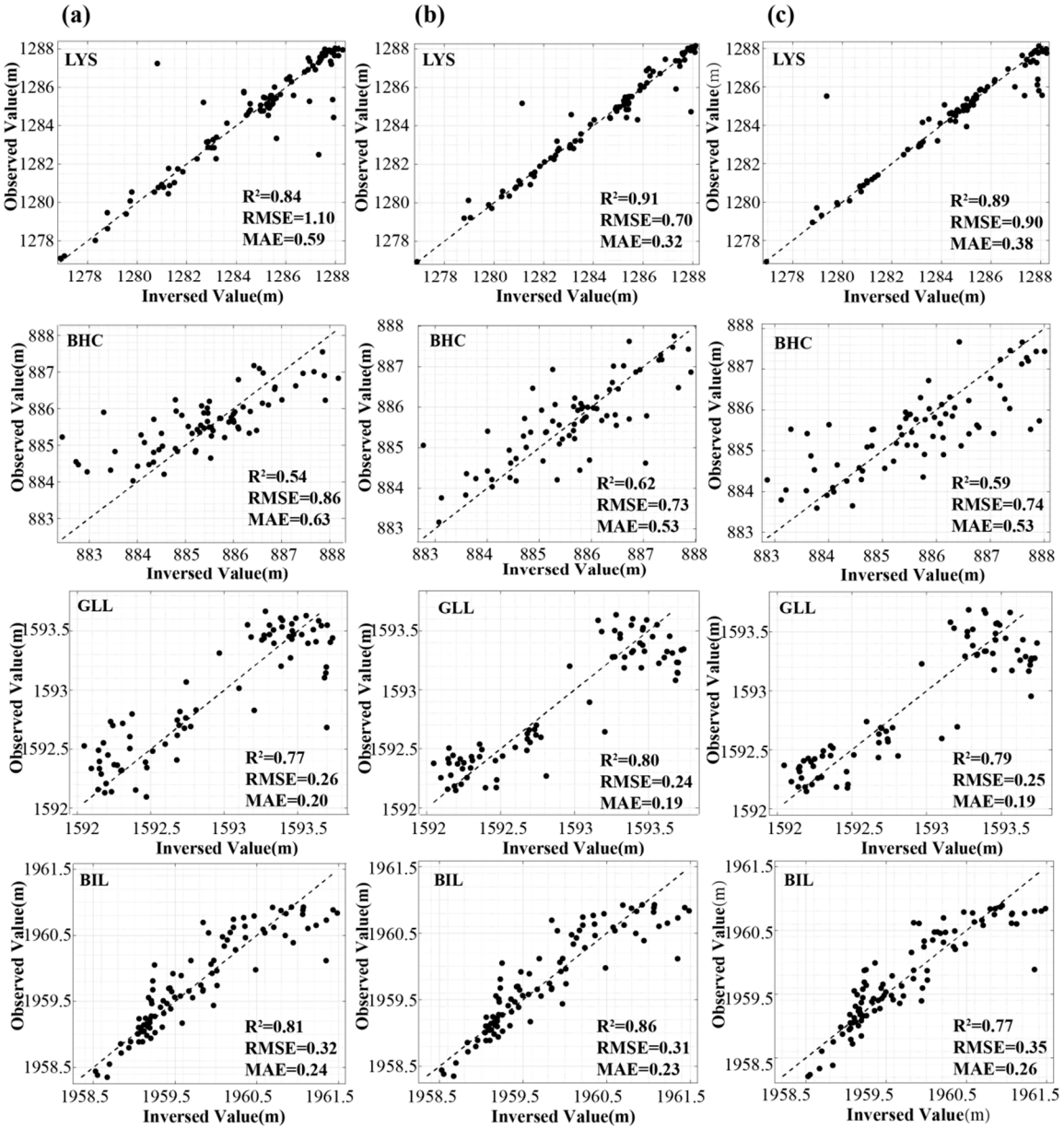



**Figure 9.** Comparisons of observed and measured data across different models (columns) and reservoirs (rows). Column (**a**) represents the simulation results of the baseline model without any added attention mechanism, column (**b**) is the model with only the channel attention mechanism added, and column (**c**) is the model with both the channel and the spatial attention mechanisms added.

The addition of CSAM improved the inversion accuracy on three of the reservoirs compared to the model without this attention mechanism. These show an average increase

of 5.9% in the $R^2$ value, an average reduction of 12.4% in the RMSE value, and an average reduction of 19% in the MAE value, with only one reservoir (BIL) showing a slight decrease in inversion accuracy.

Overall, the model with CSAM did not significantly improve the inversion accuracy compared to the model with only CAM. This suggests that the variation in band sensitivities to reservoir level changes indeed exists, and the channel attention mechanism can enhance model performance by assigning higher weights to the more sensitive bands. The addition of the spatial attention mechanism had a negative impact on model performance, which could be attributed to the spatial attention mechanism excessively emphasizing less important features or neglecting critical ones. Alternatively, such an increase in model complexity might require a larger dataset to enhance the model's generalization capability.

Focusing on the LYS and BHC reservoirs, where we have greater water level fluctuations and saw poorer performance in Figure 8, we have further examined whether there is an improvement with the addition of the attention mechanism. As depicted in Figure 10, this is promising with the relative error in the inversion of at LYS decreasing. Model-added CAM and Model-added CSAM had 6 and 7 days in the test set where the inversion water level error exceeded 1 m, with average inversion error values of 0.3 m and 0.36 m, respectively. However, for the BHC reservoir, the overall inversion accuracy of the model remains poor compared to the other three reservoirs with no pronounced improvement in results under these conditions with large changes in water levels. Model-added CAM and Model-added CSAM had 13 and 15 days in the test set where the inversion water level error exceeded 1 m, with average inversion error values of 0.5 m and 0.6 m, respectively.

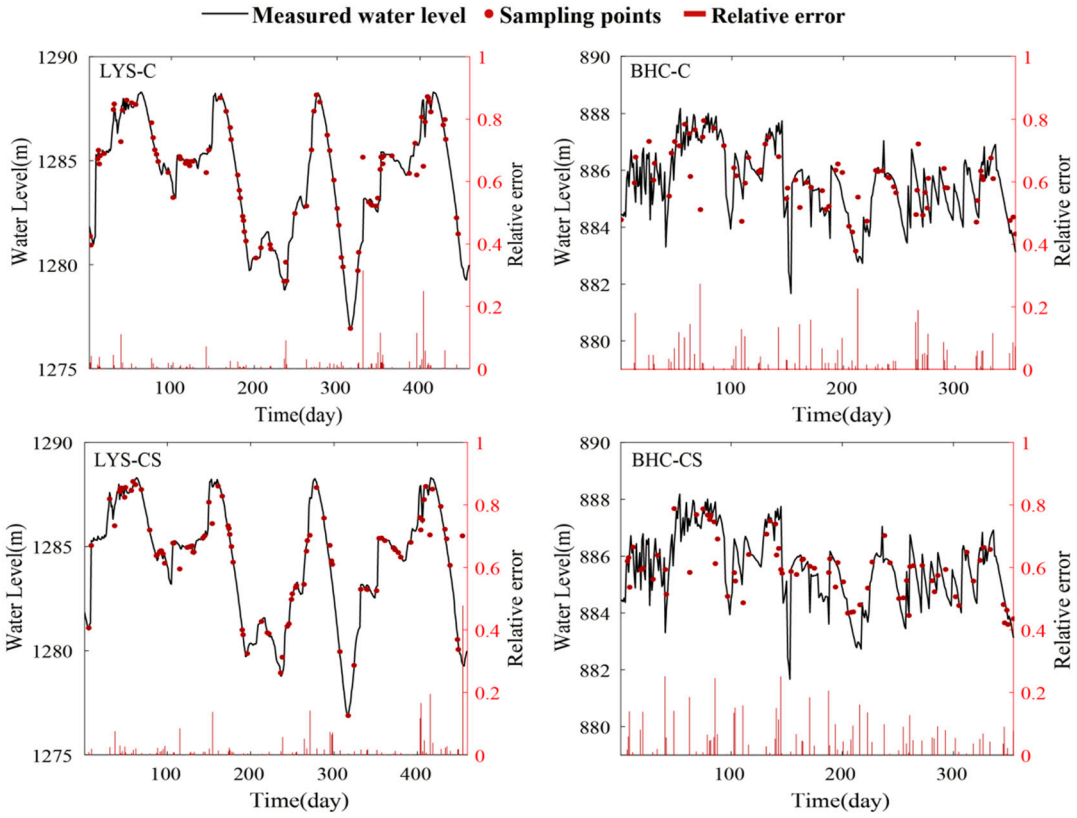



**Figure 10.** Comparison of inversion results with measured water levels after adding different attention mechanisms. C: add only the channel attention mechanism, CS: add channel and spatial attention mechanisms.

### *4.5. Model Performance of the Regionalized Approach Applied on Selected Reservoirs*

To address the issue of a potentially small training dataset for the BHC reservoirs, the regionalization approach was employed. The BIL reservoirs, given their proximity to the

BHC reservoirs, similar reservoir depths, and geographical features, were chosen for data splicing with the BHC training dataset. As indicated in Table 5, this strategy led to a slight improvement in the accuracy of BHC water level inversion.

**Table 5.** Model performance before and after using the regionalization approach in the BHC reservoirs.

| | Before | After |
|---|---|---|
| $R^2$ | 0.63 | 0.67 |
| RMSE | 0.73 m | 0.69 m |
| MAE | 0.53 m | 0.50 m |

Expanding on this, data from additional reservoirs were similarly integrated, and the water level inversion for the BHC reservoir was conducted after incorporating data from 2 to 4 reservoirs. Notably, when merging data from three reservoirs into a single training set, only BHC, BIL, and GLL, which are geographically proximate, were considered. Figure 11 illustrates the impact of using data from different numbers of reservoirs on the inversion accuracy of BHC. It is evident that the accuracy of water level inversion decreases when the model is trained with data from 3 or 4 reservoirs. One plausible explanation for this trend is that augmenting the training dataset, while providing more data for learning, also introduces disparities in other feature information between the two reservoirs that cannot be adequately accounted for using the existing variables. As the dataset length increases, these disparities become more pronounced, resulting in a significant reduction in accuracy.

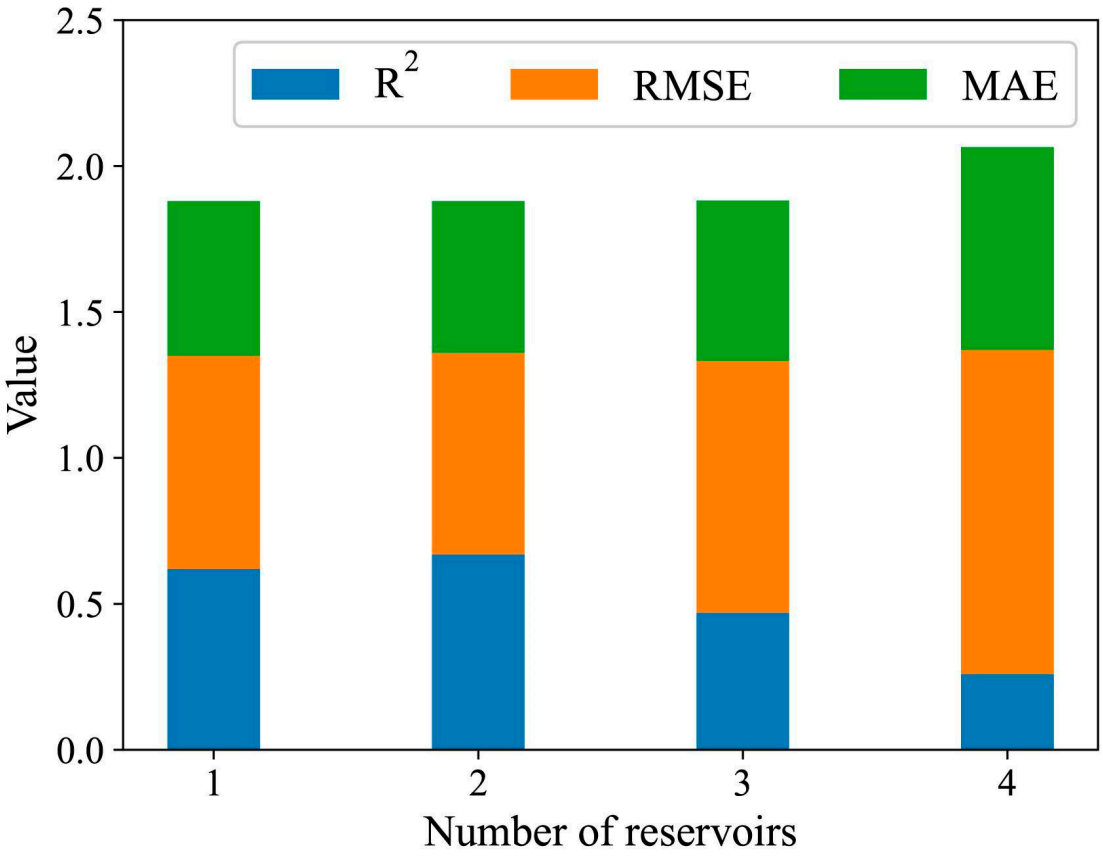



**Figure 11.** Comparison of model performance with varied training set sizes in BHC.

### *4.6. Further Validation of the Model for Application in Diverse Regions*

To explore the applicability of our proposed framework in different regions, we conducted validation on three additional randomly selected reservoirs. Initially, the CNN benchmark model was employed, and the results, as shown in Table 6, revealed $R^2$ values

exceeding 0.8 when using the optimal combination of data sources and sampling methods. However, the RMSE and MAE for CTR and ARR were relatively large, indicating noticeable differences compared to the four reservoirs in California. This suggests that the model can well explain the variability of the data and robustly invert water levels. In certain instances, a notable discrepancy exists between the inversed and observed values. This is mainly due to the smaller size of the available training sets for CTR and ARR ($n$ = 92, $n$ = 245). Additionally, constructing the training set using the RS method ensures that the model maintains good performance when the water level sequence is incomplete (e.g., CTR water level observation sequence partially missing during 2018–2020, and MCR lacking observed water levels due to winter freeze).

**Table 6.** Benchmark model performance on three additional reservoirs.

| Reservoir and Sampling Approach | $R^2$ | RMSE | MAE |
|---|---|---|---|
| MCR-2-R | 0.85 | 0.41 m | 0.23 m |
| CTR-1-R | 0.91 | 1.27 m | 0.77 m |
| ARR-2-R | 0.88 | 1.00 m | 0.72 m |

Building upon this, an assessment was conducted for the model with the addition of attention mechanisms. The outcomes remained consistent with those presented in Section 4.4, detailed in Table 7. For MCR, the model with the incorporation of CSAM displayed a decrease in performance, while other cases exhibited varying degrees of improvement. Models featuring CAM consistently outperformed those incorporating CSAM.

**Table 7.** Model performance with attention mechanism on three additional reservoirs.

| Reservoir and Different Attention Mechanisms | $R^2$ | RMSE | MAE |
|---|---|---|---|
| MCR-CAM | 0.96 | 0.20 m | 0.09 m |
| MCR-CSAM | 0.82 | 0.54 m | 0.33 m |
| CTR-CAM | 0.97 | 0.68 m | 0.38 m |
| CTR-CSAM | 0.95 | 1.03 m | 0.39 m |
| ARR-CAM | 0.92 | 0.77 m | 0.49 m |
| ARR-CSAM | 0.91 | 0.92 m | 0.52 m |

The validation on three additional reservoirs illustrates the validity of our proposed water level inversion framework for small reservoirs across different regions and climatic conditions, it could be further expanded to cover more small reservoirs globally.

## 5. Discussion

In this study, a deep learning framework based on convolutional neural networks is proposed for inversing reservoir water levels. The results show that the deep learning model with the addition of CAM exhibits the best performance followed by the model incorporating CSAM and, finally, the model without any attention mechanism.

The accuracy of our water level inversion model was compared with some large-scale studies. Chen, et al. [48] utilized ICESat-2 to detect global reservoir dynamics, achieving $R^2$ values ranging from 0.60 to 0.99 and RMSE values from 0.37 m to 1.01 m for 40 random reservoirs' water levels (essentially large reservoirs). Donchyts, et al. [49] compared water level measurements from 768 small and medium-sized reservoirs in Spain, India, South Africa, and the United States, establishing a relationship between reservoir surface area and water level. It was shown that 67% of the reservoirs achieved $R^2$ values higher than 0.7. In our study, the best-performing model inversed water levels for four California reservoirs and three reservoirs in other regions, achieving $R^2$ values ranging from 0.62 to 0.97, RMSE values from 0.19 m to 0.77 m, and a mean $R^2$ value of 0.86. This underscores the remarkable accuracy with our proposed framework, offering a practical solution to address the limitations of water level monitoring in small reservoirs.

In general, the use of Sentinel-2 data outperforms the use of Sentinel-1 data. Although Sentinel-1 offers the advantage of active radar sensing, which can overcome cloud cover limitations, it is susceptible to interference from terrain and vegetation. Variations in terrain relief and vegetation coverage can alter signal reflection and scattering properties, as noted in previous studies [50,51]. The inherent scattering effect in Sentinel-1 images introduces noise, further affecting the accuracy of water level monitoring. Contrastingly, the higher spatial resolution of Sentinel-2, although susceptible to shadowing effects in mountainous areas, along with its multispectral bands (both near-infrared and visible), exhibits greater sensitivity to water transparency and bottom features. This allows effective capture of water reflectance and spectral characteristics.

Models trained using the RS approach demonstrate better performance compared to those trained using the CS approach. A possible reason is that following the preprocessing of the remote sensing data, there no longer exhibits a strong temporal correlation due to the dynamic nature of the reservoir operational rules. Consequently, the overall temporal characteristics of the data become less evident, making it challenging for deep learning models to capture them. By randomly splitting the training and validation sets, the model effectively learns the image features associated with high and low water levels, improving its generalization capability and mitigating the occurrence of overfitting.

This study selected four typical small reservoirs within California to construct water level inversion models and further verified the portability in new and diverse geographical locations. The impact of the training set size on the model's accuracy is a critical consideration. Therefore, different sizes of training sets should also be tested to quantify the model inversion accuracy as input data availability changes.

Furthermore, the model frameworks presented currently only learn to inverse water levels at the specific individual reservoirs and cannot be easily transferred to simultaneously inverse the water levels of a large number of unknown small reservoirs. The basic regionalization method employed yielded mediocre results. A more sophisticated regional model could improve the accuracy of water level inversion and has the potential to be useful in inversing water levels in small reservoirs by assimilating more general rules from satellite images where observed water level data are limited. This may require further changes in the architecture of the model to accommodate regionalized inversions and any additional considerations of more generic feature information about reservoir water levels, such as spatial characteristics and morphological differences of reservoirs.

## 6. Conclusions

In contrast to conventional CNN applications primarily focused on water body detection and range change identification, in this study, a deep learning framework based on satellite data and CNN is proposed for retrieving the water level of small reservoirs. The findings suggest that combining remote sensing imagery with a CNN-based inversion model provides a stable and accurate approach for remotely estimating water levels in small reservoirs. The addition of CAM allows the model to better capture the pattern of response of different bands or polarisation modes to changes in water level. However, an increase in model complexity does not necessarily lead to an increase in model performance. Future work will need to focus on developing a more mechanistic regionalized framework for water level inversion and utilizing additional generic variables to contribute to water level inversion for the growing number of small reservoirs globally that lack observed measurement data.

**Author Contributions:** J.W. (Jiarui Wu): Software, Methodology, Investigation, Writing—Original draft. X.H.: Conceptualization, Supervision. N.X.: Resources. Q.Z.: Investigation. C.Z.: Writing—Reviewing and Editing. W.G.: Data curation. J.W. (Jiangnan Wang): Visualization. B.W.: Data curation. S.S.: Visualization. C.Y.: Funding acquisition. All authors have read and agreed to the published version of the manuscript.

**Funding:** This study was supported by the Hainan Province Science and Technology Special Fund (ZDYF2023XDNY181), the Hainan Province Science and Technology Special Fund (ZDYF2023SHFZ172) the College Students' Innovation and Entrepreneurship Training Program (202310589040) and the Hainan University Research Start-up Fund (RZ2300002833).

**Data Availability Statement:** All the data used in the manuscript are publicly available and can be found at the corresponding URLs mentioned in the text.

**Acknowledgments:** The authors wish to express their gratitude to Remote Sensing, as well as to the anonymous reviewers who helped to improve this paper through their thorough review.

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

## References

1. Cooley, S.W.; Ryan, J.C.; Smith, L.C. Human alteration of global surface water storage variability. *Nature* **2021**, *591*, 78–81. [CrossRef] [PubMed]
2. Niu, W.J.; Feng, Z.K.; Li, Y.R.; Liu, S. Cooperation Search Algorithm for Power Generation Production Operation Optimization of Cascade Hydropower Reservoirs. *Water Resour. Manag.* **2021**, *35*, 2465–2485. [CrossRef]
3. Liu, J.J.; Yuan, X.; Zeng, J.H.; Jiao, Y.; Li, Y.; Zhong, L.H.; Yao, L. Ensemble streamflow forecasting over a cascade reservoir catchment with integrated hydrometeorological modeling and machine learning. *Hydrol. Earth Syst. Sci.* **2022**, *26*, 265–278. [CrossRef]
4. Chen, W.J.; Nover, D.; He, B.; Yuan, H.L.; Ding, K.M.; Yang, J.; Chen, S.Z. Analyzing inundation extent in small reservoirs: A combined use of topography, bathymetry and a 3D dam model. *Measurement* **2018**, *118*, 202–213. [CrossRef]
5. Pachauri, R.K.; Reisinger, A. *Climate Change 2007: Synthesis Report. Contribution of Working Groups I, II and III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*; IPCC: Geneva, Switzerland, 2007.
6. Thomas, B.; Steidl, J.; Dietrich, O.; Lischeid, G. Measures to sustain seasonal minimum runoff in small catchments in the mid-latitudes: A review. *J. Hydrol.* **2011**, *408*, 296–307. [CrossRef]
7. Grill, G.; Lehner, B.; Thieme, M.; Geenen, B.; Tickner, D.; Antonelli, F.; Babu, S.; Borrelli, P.; Cheng, L.; Crochetiere, H.; et al. Mapping the world's free-flowing rivers. *Nature* **2019**, *569*, 215–221. [CrossRef]
8. Ignatius, A.R.; Rasmussen, T.C. Small reservoir effects on headwater water quality in the rural-urban fringe, Georgia Piedmont, USA. *J. Hydrol. Reg. Stud.* **2016**, *8*, 145–161. [CrossRef]
9. Zhang, X.; Fang, C.; Wang, Y.; Lou, X.; Su, Y.; Huang, D. Review of Effects of Dam Construction on the Ecosystems of River Estuary and Nearby Marine Areas. *Sustainability* **2022**, *14*, 5974. [CrossRef]
10. Wang, J.; Walter, B.A.; Yao, F.; Song, C.; Ding, M.; Maroof, A.S.; Zhu, J.; Fan, C.; McAlister, J.M.; Sikder, S.; et al. GeoDAR: Georeferenced global dams and reservoirs dataset for bridging attributes and geolocations. *Earth Syst. Sci. Data* **2022**, *14*, 1869–1899. [CrossRef]
11. Liebe, J.; van de Giesen, N.; Andreini, M. Estimation of small reservoir storage capacities in a semi-arid environment: A case study in the Upper East Region of Ghana. *Phys. Chem. Earth Parts A/B/C* **2005**, *30*, 448–454. [CrossRef]
12. Krol, M.S.; de Vries, M.J.; van Oel, P.R.; de Araujo, J.C. Sustainability of Small Reservoirs and Large Scale Water Availability Under Current Conditions and Climate Change. *Water Resour. Manag.* **2011**, *25*, 3017–3026. [CrossRef]
13. Habets, F.; Molenat, J.; Carluer, N.; Douez, O.; Leenhardt, D. The cumulative impacts of small reservoirs on hydrology: A review. *Sci. Total Environ.* **2018**, *643*, 850–867. [CrossRef] [PubMed]
14. Deemer, B.R.; Harrison, J.A.; Li, S.Y.; Beaulieu, J.J.; Delsontro, T.; Barros, N.; Bezerra-Neto, J.F.; Powers, S.M.; dos Santos, M.A.; Vonk, J.A. Greenhouse Gas Emissions from Reservoir Water Surfaces: A New Global Synthesis. *Bioscience* **2016**, *66*, 949–964. [CrossRef] [PubMed]
15. Song, J.-H.; Her, Y.; Kang, M.-S. Estimating Reservoir Inflow and Outflow from Water Level Observations Using Expert Knowledge: Dealing with an Ill-Posed Water Balance Equation in Reservoir Management. *Int. J. Elect. Power Energy Syst.* **2022**, *58*, e2020WR028183. [CrossRef]
16. Li, J.Z.; Sun, H.F.; Feng, P. How to update design floods after the construction of small reservoirs and check dams: A case study from the Daqinghe river basin, China. *J. Earth Syst. Sci.* **2016**, *125*, 795–808. [CrossRef]
17. Mandlburger, G.; Kölle, M.; Nübel, H.; Soergel, U. BathyNet: A Deep Neural Network for Water Depth Mapping from Multispectral Aerial Images. *PFG J. Photogramm. Remote Sens. Geoinf. Sci.* **2021**, *89*, 71–89. [CrossRef]
18. Ma, Y.; Xu, N.; Sun, J.; Wang, X.H.; Yang, F.; Li, S. Estimating water levels and volumes of lakes dated back to the 1980s using Landsat imagery and photon-counting lidar datasets. *Remote Sens. Environ.* **2019**, *232*, 111287. [CrossRef]
19. Xu, N.; Gong, P. Significant coastline changes in China during 1991–2015 tracked by Landsat data. *Sci. Bull.* **2018**, *63*, 883–886. [CrossRef]
20. Xu, N.; Zheng, H.Y.; Ma, Y.; Yang, J.; Liu, X.Y.; Wang, X.H. Global Estimation and Assessment of Monthly Lake/Reservoir Water Level Changes Using ICESat-2 ATL13 Products. *Remote Sens.* **2021**, *13*, 2744. [CrossRef]
21. Shen, Y.J.; Liu, D.D.; Jiang, L.G.; Nielsen, K.; Yin, J.B.; Liu, J.; Bauer-Gottwein, P. High-resolution water level and storage variation datasets for 338reservoirs in China during 2010–2021. *Earth Syst. Sci. Data* **2022**, *14*, 5671–5694. [CrossRef]

22. Da Silva, J.S.; Seyler, F.; Calmant, S.; Rotunno, O.C.; Roux, E.; Araujo, A.A.M.; Guyot, J.L. Water level dynamics of Amazon wetlands at the watershed scale by satellite altimetry. *Int. J. Remote Sens.* **2012**, *33*, 3323–3353. [CrossRef]
23. Duan, Z.; Bastiaanssen, W.G.M. Estimating water volume variations in lakes and reservoirs from four operational satellite altimetry databases and satellite imagery data. *Remote Sens. Environ.* **2013**, *134*, 403–416. [CrossRef]
24. Ryan, J.C.; Smith, L.C.; Cooley, S.W.; Pitcher, L.H.; Pavelsky, T.M. Global Characterization of Inland Water Reservoirs Using ICESat-2 Altimetry and Climate Reanalysis. *Geophys. Res. Lett.* **2020**, *47*, e2020GL088543. [CrossRef]
25. Kwok, R.; Kacimi, S.; Markus, T.; Kurtz, N.T.; Studinger, M.; Sonntag, J.G.; Manizade, S.S.; Boisvert, L.N.; Harbeck, J.P. ICESat-2 Surface Height and Sea Ice Freeboard Assessed with ATM Lidar Acquisitions from Operation IceBridge. *Geophys. Res. Lett.* **2019**, *46*, 11228–11236. [CrossRef]
26. Dettmering, D.; Ellenbeck, L.; Scherer, D.; Schwatke, C.; Niemann, C. Potential and Limitations of Satellite Altimetry Constellations for Monitoring Surface Water Storage Changes—A Case Study in the Mississippi Basin. *Int. J. Elect. Power Energy Syst.* **2020**, *12*, 3320. [CrossRef]
27. Wang, Y.; Long, D.; Li, X. High-temporal-resolution monitoring of reservoir water storage of the Lancang-Mekong River. *Remote Sens. Environ.* **2023**, *292*, 113575. [CrossRef]
28. Zhang, C.; Lv, A.; Zhu, W.; Yao, G.; Qi, S. Using Multisource Satellite Data to Investigate Lake Area, Water Level, and Water Storage Changes of Terminal Lakes in Ungauged Regions. *Int. J. Elect. Power Energy Syst.* **2021**, *13*, 3221. [CrossRef]
29. Zhang, S.; Gao, H.L.; Naz, B.S. Monitoring reservoir storage in South Asia from multisatellite remote sensing. *Water Resour. Res.* **2014**, *50*, 8927–8943. [CrossRef]
30. Gao, H.; Birkett, C.; Lettenmaier, D.P. Global monitoring of large reservoir storage from satellite remote sensing. *Water Resour. Res.* **2012**, *48*. [CrossRef]
31. Yigzaw, W.; Li, H.Y.; Demissie, Y.; Hejazi, M.I.; Leung, L.R.; Voisin, N.; Payn, R. A New Global Storage-Area-Depth Data Set for Modeling Reservoirs in Land Surface and Earth System Models. *Water Resour. Res.* **2018**, *54*, 10372–10386. [CrossRef]
32. Kim, J.; Kim, H.; Jeon, H.; Jeong, S.-H.; Song, J.; Vadivel, S.K.P.; Kim, D.-j. Synergistic Use of Geospatial Data for Water Body Extraction from Sentinel-1 Images for Operational Flood Monitoring across Southeast Asia Using Deep Neural Networks. *Remote Sens.* **2021**, *13*, 4759. [CrossRef]
33. Available online: https://cdec.water.ca.gov/dynamicapp/selectQuery (accessed on 1 August 2022).
34. Miao, Z.M.; Fu, K.; Sun, H.; Sun, X.; Yan, M.L. Automatic Water-Body Segmentation from High-Resolution Satellite Images via Deep Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 602–606. [CrossRef]
35. Yang, H.; Guo, H.L.; Dai, W.H.; Nie, B.K.; Qiao, B.J.; Zhu, L.P. Bathymetric mapping and estimation of water storage in a shallow lake using a remote sensing inversion method based on machine learning. *Int. J. Digit. Earth* **2022**, *15*, 789–812. [CrossRef]
36. Audebert, N.; Le Saux, B.; Lefèvre, S. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 20–32. [CrossRef]
37. Chen, Y.; Tang, L.; Kan, Z.; Bilal, M.; Li, Q. A novel water body extraction neural network (WBE-NN) for optical high-resolution multispectral imagery. *J. Hydrol.* **2020**, *588*, 125092. [CrossRef]
38. Lumban-Gaol, Y.A.; Ohori, K.A.; Peters, R.Y. Satellite-derived bathymetry using convolutional neural networks and multispectral sentinel-2 images. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *43*, 201–207. [CrossRef]
39. Najar, M.A.; Benshila, R.; Bennioui, Y.E.; Thoumyre, G.; Almar, R.; Bergsma, E.W.J.; Delvit, J.-M.; Wilson, D.G. Coastal Bathymetry Estimation from Sentinel-2 Satellite Imagery: Comparing Deep Learning and Physics-Based Approaches. *Remote Sens.* **2022**, *14*, 1196. [CrossRef]
40. Hu, F.; Zhang, L.Y.; Liu, Q.; Chyi, D. Environmental Factors Controlling the Precipitation in California. *Atmosphere* **2021**, *12*, 997. [CrossRef]
41. Wan, W.; Zhao, J.; Li, H.-Y.; Mishra, A.; Leung, L.R.; Hejazi, M.; Wang, W.; Lu, H.; Deng, Z.; Demissisie, Y.; et al. Hydrological Drought in the Anthropocene: Impacts of Local Water Extraction and Reservoir Regulation in the US. *J. Geophys. Res. -Atmos.* **2017**, *122*, 11313–11328. [CrossRef]
42. Peña-Luque, S.; Ferrant, S.; Cordeiro, M.C.R.; Ledauphin, T.; Maxant, J.; Martinez, J.-M. Sentinel-1&2 Multitemporal Water Surface Detection Accuracies, Evaluated at Regional and Reservoirs Level. *Remote. Sens.* **2021**, *13*, 3279.
43. He, K.M.; Zhang, X.Y.; Ren, S.Q.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 27–30 June 2016; pp. 770–778.
44. Haut, J.M.; Fernandez-Beltran, R.; Paoletti, M.E.; Plaza, J.; Plaza, A.J.I.T.o.G.; Sensing, R. Remote sensing image superresolution using deep residual channel attention. *IEEE Trans. Geosci. Remote. Sens.* **2019**, *57*, 9277–9289. [CrossRef]
45. Dou, H.-X.; Pan, X.-M.; Wang, C.; Shen, H.-Z.; Deng, L.-J.J.R.S. Spatial and spectral-channel attention network for denoising on hyperspectral remote sensing image. *IEEE Trans. Geosci. Remote. Sens.* **2022**, *14*, 3338. [CrossRef]
46. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E.H. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 2011–2023. [CrossRef]
47. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.M.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 8–14 December 2019.
48. Chen, T.; Song, C.; Luo, S.; Ke, L.; Liu, K.; Zhu, J. Monitoring global reservoirs using ICESat-2: Assessment on spatial coverage and application potential. *J. Hydrol.* **2022**, *604*, 127257. [CrossRef]

49. Donchyts, G.; Winsemius, H.; Baart, F.; Dahm, R.; Schellekens, J.; Gorelick, N.; Iceland, C.; Schmeier, S. High-resolution surface water dynamics in Earth's small and medium-sized reservoirs. *Sci. Rep.* **2022**, *12*, 13776. [CrossRef]
50. Chen, Z.; Zhao, S. Automatic monitoring of surface water dynamics using Sentinel-1 and Sentinel-2 data with Google Earth Engine. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *113*, 103010. [CrossRef]
51. Wang, X.; Li, C.; Wu, R. River boundaries extraction in mountain areas for SAR images with fusing GIS information. In Proceedings of the 2011 IEEE CIE International Conference on Radar, Chengdu, China, 24–27 October 2011; pp. 1586–1588.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

# Image Classification using a Hybrid LSTM-CNN Deep Neural Network

**Aditi, Mayank Kumar Nagda, Poovammal E**

***Abstract:*** ***This work elaborates on the integration of the rudimentary Convolutional Neural Network (CNN) with Long Short-Term Memory (LSTM), resulting in a new paradigm in the well-explored field of image classification. LSTM is one kind of Recurrent Neural Network (RNN) which has the potential to memorize long-term dependencies. It was observed that LSTMs are able to complement the feature extraction ability of CNN when used in a layered order. LSTMs have the capacity to selectively remember patterns for a long duration of time and CNNs are able to extract the important features out of it. This LSTM-CNN layered structure, when used for image classification, has an edge over conventional CNN classifier. The model which has been proposed is based on the sets of Artificial Neural Network like Recurrent and Convolutional neural network; hence this model is robust and suitable to a wide spectrum of classification tasks. To validate these results, we have tested our model on two standard datasets. The results have been compared with other classifiers to establish the significance of our proposed model.***

***Keywords*****: *Artificial Intelligence, Computer Vision, Deep Learning, Neural Networks.***

## I. INTRODUCTION

Computer Vision is a topic which has observed wide attention of researchers in the past few decades. It is an interdisciplinary field that aims at gaining a high-level understanding from digital images and videos. It aims at developing methods that can reproduce the capability of human vision. Computer Vision aims at developing different methods to understand the content of digital images. In order to understand the images better, computer vision automatically extracts information from input images and videos. Some of the major applications of computer vision are navigation, assisting humans in identification tasks, event detection and organizing information [26][27]. Image recognition and classification continues to be a predominant area in the field of computer vision. It has recently gained a lot of fame due to its widespread applications. Image recognition is used to perform a large number of visual tasks which are based on machines. A few of these tasks include the labelling of the images with meta-tags, performance of image content search, guidance to autonomous robots and self-driving cars [1][2][3][4]. Image classification has an integral role to play in the field of computer-aided-diagnosis as well. Medical image classification which is a part of computer-aided-diagnosis aims at achieving a high accuracy along with the identification of the parts of the human body which are infected by the disease [7].

These wide range applications of image classification and recognition necessitate the need for good learning algorithms and models which can perform these tasks with high accuracy. In the last few decades rapid advancement has been done in this field using different machine learning algorithms and models. Machine learning is an important application of artificial intelligence (AI). Systems trained with the help of machine learning do not require explicit programming. They can learn automatically based on the previous experience [9]. In recent years researchers have discovered another breakthrough technology in this field which is popularly known as Deep Learning [28]. Unlike machine learning which employs shallow architectures, deep learning resembles the pattern of our brain which is quite similar to a deep architecture. Due to these deep architectures, the information undergoes through multiple transformations before it is finally represented. It passes the input through various layers of simulated neural connection to achieve improved accuracy.

A novel hybrid deep neural network consisting of LSTM and CNN layer has been proposed in this paper. In order to perform a comprehensive evaluation of the proposed model of deep convolutional neural networks, we have applied it on two of the classic image classification datasets i.e. MNIST [10] and IDC Breast Cancer [25]. To establish the efficiency of the proposed model, a comparison has been made with the other state-of-the-art classifiers.

## II. RELATED WORK

Image classification task has recently gained fame amongst researchers due to its colossal contribution in the computer vision field. It finds its application in a variety of automation tasks such as self-driving cars, scene detection and computer-aided diagnosis [4][5][6].

Due to a large number of applications of image classification, different methods have been adopted to achieve higher accuracy in this field. Out of all these methods, Deep learning models are the most preferred ones since they possess deep architectures.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
**Aditi***, Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur 603203, India.
**Mayank Kumar Nagda**, Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur 603203, India.
**E. Poovammal***, Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur 603203, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

They have been employed to achieve reliable results on simple image classification tasks such as handwritten digits, face recognition, textures and objects [11]. It has been observed that deep architectures perform better than shallow ones like SVM, and hence this justifies their increased usage in this field [12].

One of the most common types of deep neural networks applied in the domain of visual imagery is Convolutional Neural Network (CNN) [13]. LeCun et al. applied the supervised deep back-propagation convolutional network for recognition of digits [10]. A state-of-the-art layered HCCR-GoogLeNet has been proposed for the Chinese handwritten characters recognition by Zhuoyao Zhong et al. They found out that such architectures perform better even with a lesser number of parameters [16]. Apart from this, CNNs have proved to be efficient in a variety of other tasks. Hokuto Kagaya applied CNN to recognize food images. They concluded that the results obtained through CNN are better than those of conventional methods [14]. These Deep Convolutional Networks (DNN) have been found to perform remarkably good in the task of face recognition as well [15]. CNN has also been employed to visualize the performance on several scenic datasets [17]. Activation functions play a vital role in the functioning of Neural Networks. They make the task of backpropagation possible by supplying gradients along with the errors to update weight and bias. Kaiming He [18] investigated the rectifier properties of Neural Networks. They proposed a novel Parametric Rectified Linear Unit (PReLU) as a generalized version of the traditional rectified unit which gave remarkable results on ImageNet 2012 dataset [19]. Rectified units help to alleviate the problem of vanishing gradient. However, this problem can be avoided by using special gated recurrent units having tanh as an activation function. These special units are known as Long Short-Term Memory [20]. Wonmin Byeon et al. performed the pixel-level segmentation and classification of scene images using LSTM which outperformed the state-of-the-art methods in the same field [21]. Soo Hyun Bae et al. proposed a parallel combination of CNN and LSTM layer to efficiently improve the classification accuracy of acoustic scenes [5]. The datasets used in our work have been extensively used to discover and unveil new breakthrough technologies in this field. Ciresan et al. attained an accuracy of 99.65% on MNIST dataset with the help of 6-layer Neural Network [8]. Dan Cires et al. achieved the near-human accuracy on the same dataset with an error rate of only 0.23% [29]. Cruz-Roa, on the other hand applied a deep learning approach on the IDC breast cancer dataset. They attained F-measure and balanced accuracy of 71.80% and 84.23% respectively [25].

## III. PROPOSED MODEL

The proposed image classification model is a layered Deep Neural Network consisting of Long short-term memory (LSTM) and a Convolutional Neural Network (CNN). LSTM is a kind of RNN which unlike the traditional feedforward neural networks has feedback connections. This feature of possessing feedback connections make LSTM a type of "general purpose computer" enabling it to compute everything a Turing machine can. Section III (A) and III (B) gives a detailed explanation of LSTMs and CNNs.

### A. *Long short-term memory (LSTM)*

As represented in Fig.1, we can define a unit of LSTM at each time step $t$ as a collection of vectors in $R^d$ consisting of forget gate $f_t$, input gate $i_t$, memory cell $c_t$, output gate $o_t$ and a hidden state $h_t$, where $d$ is the magnitude of the memory dimension [30][31]. The LSTM equations are numbered from (1) through (6).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (1)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \quad (2)$$

$$\tilde{C}_t = tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \quad (5)$$

$$h_t = o_t * \tanh\left(C_t\right) \quad (6)$$

In the equations, b and W denote the bias vector and weight matrices for the input gate, output gate, forget gate, memory cell, tanh layer and the hidden layer. While $\sigma$ denotes logistic sigmoid function.

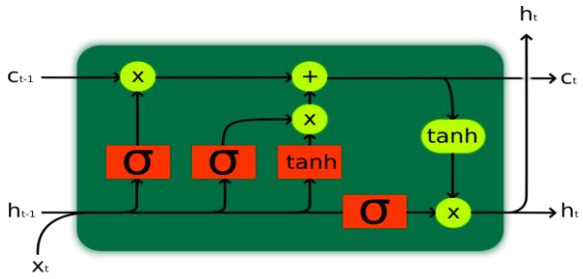



**Figure 1 A LSTM Cell**

In an LSTM unit, the input gate is fed with a new stream of data at every time step t and is responsible for making the decision on remembering the information it processes. Forget gate, on the other hand, is responsible for regulating the amount of information that should be removed from the memory cell.

### B. *Convolutional Neural Network*

CNNs are influenced by the biological neural networks and are defined as a regularized version of multilayer perceptron [22][23][24]. They are structured as a fully connected layer, which means that each neuron present in one layer is connected to other neurons in the succeeding layer. CNNs are known to use little bit of pre-processing in comparison to other traditional image classification methods. A CNN is made up of different layers, one of them is input layer, one is output layer and the others include multiple hidden layers. The hidden layers are composed of multiple convolutional layers that convolve with multiplication or dot product. Fig. 2 represents a simple architecture of a CNN being used to predict handwritten digits.

CNNs can easily capture important features from an image by taking advantage of local spatial coherence with which they give good results on image classification problems. Also, possessing the quality to extract important features makes CNNs a very good option for a completely new task.

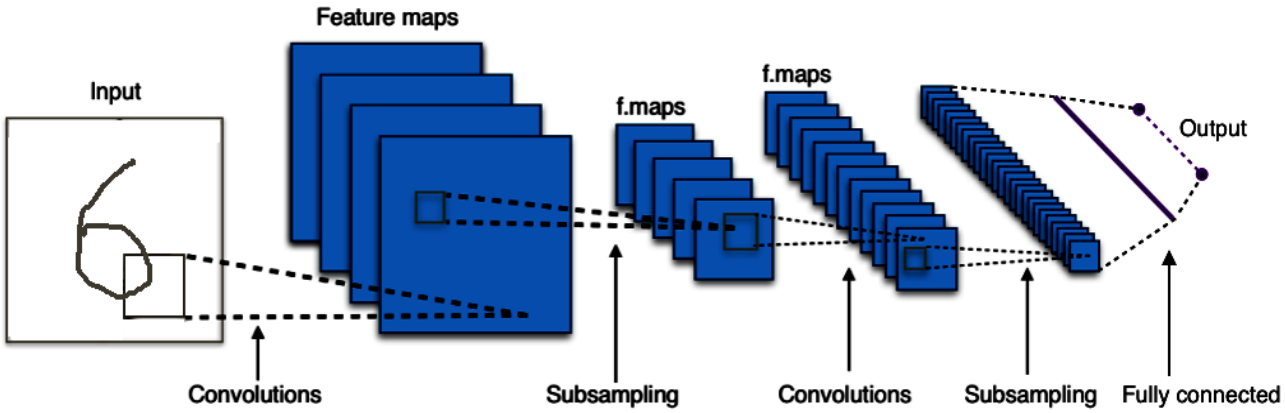



**Figure 2 Convolutional Neural Network Architecture**

### C. *The LSTM-CNN Neural Network*

Figure 3 represents the architecture of our proposed LSTM-CNN model. The images are passed through the input layer. This input layer is then fed into the Batch Normalization layer. Batch Normalization layer applies transformation to the preceding layer which maintains the standard deviation of activation function close to 1 and mean activation close to 0, thus normalizing it. We apply normalization to each feature such that every feature map of the input is separately normalized. "Axis" argument specifies the axis on which the normalization has to be performed. we have applied statistics to every batch in order to normalize the data during training, and during testing, we use running averages computed during the training phase.

The output shape from Batch Normalization layer is the same as that of the input shape, which makes it unusable for LSTM cell. To change the shape to the desired dimension, a reshape layer can be used before the LSTM layer. After the dimensions of the input layer are reshaped it is passed through the LSTM cell. Tanh i.e. the Hyperbolic tangent is used as an activation function of the LSTM cell. The LSTM cell also has a dropout rate to help prevent overfitting of data.

Because of these characteristics of LSTM, it will particularly remember the long-term dependencies and shape of the input image in a particular pattern. The output from the LSTM layer is directly provided to the convolutional layer. A convolution kernel is created by the convolutional layer which produces a tensor of outputs by convolving with the layer input over a single spatial (temporal) dimension. The convolutional layer will extract the local important features. Rectified Linear Unit (ReLU) has been used as an activation function in this convolutional layer. A dropout layer can be applied after the convolutional layer to prevent the overfitting present due to "fully-connectedness" of the neurons in the CNN. For complex classification problems, a committee of LSTM-CNN networks can be used.

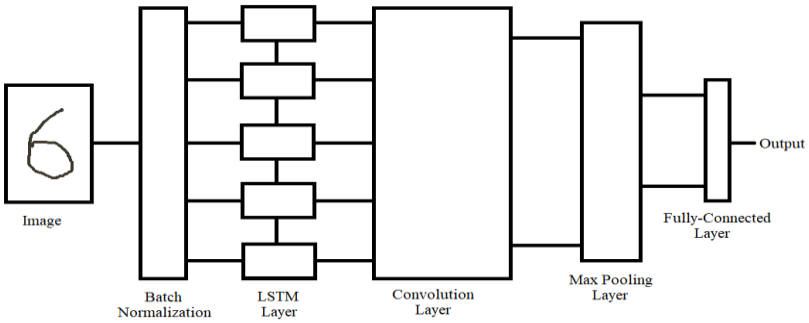



**Figure 3 LSTM-CNN Neural Network Architecture**

## IV. EXPERIMENT AND RESULTS

The hybrid model proposed in Section III (C) is theoretically supposed to have a better performance than its conventional counterparts in the task of image classification. In theory, this model has an edge over other models in tasks such as handwriting recognition, sentimental analysis, and other such problems which can benefit from the LSTMs capability to remember long term dependencies.

To validate the proposed model, we have used the benchmark datasets and compared the final results. We have chosen two different datasets to benchmark our results. The highly competitive MNIST and Breast Cancer IDC datasets are chosen for testing the proposed model.

### A. *MNIST Dataset*

The available MNIST dataset is composed of a large number of images of handwritten digits. This MNIST dataset is a subset of the large NIST dataset [10]. By default, to get a fixed-size image the digits have been size-normalized and centered already.

To evaluate our proposed model, we compare it against CNN, LSTM, and CNN-LSTM as these models have attained excellent and near perfect accuracy on the same dataset. The parameters of these models are kept identical to perform a fair comparison.

The average training accuracy and validation accuracy are used to perform comparisons between different models. As represented in Fig. 4 and Fig. 5, it was observed that the proposed LSTM-CNN model performed significantly better than the other classifiers. The overall training, as well as validation accuracy of the LSTM-CNN model, was found to be greater than those of other models it was compared against.

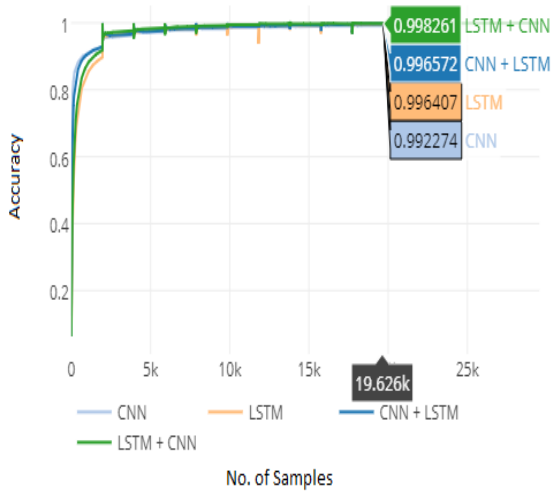



**Figure 4 Training Accuracies for MNIST dataset**

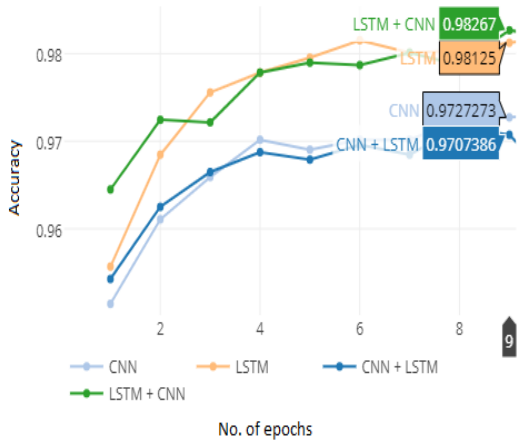



**Figure 5 Validation Accuracy for MNIST dataset**

Table I displays the accuracy achieved by different classifiers. On analyzing the results, it can easily be comprehended that the LSTM-CNN model outperforms the other models as a significant difference in percentage accuracies can be observed.

**Table I Accuracies comparison over MNIST dataset.**

| Model | Training Accuracy (%) | Validation Accuracy (%) |
|---|---|---|
| **LSTM + CNN** | **99.8261** | **98.267** |
| CNN + LSTM | 99.6572 | 97.074 |
| LSTM | 99.6407 | 98.125 |
| CNN | 99.2274 | 97.273 |

The proposed LSTM-CNN model was given a full run with an addition of 2 more LSTM-CNN layers as mentioned in the model. This particular model gives an accuracy of more than 99% on both validation and test set. It gives near-perfect accuracy of 99.7% on training set and 99.29% on the validation set as represented in Fig. 6 and Fig. 7 respectively.

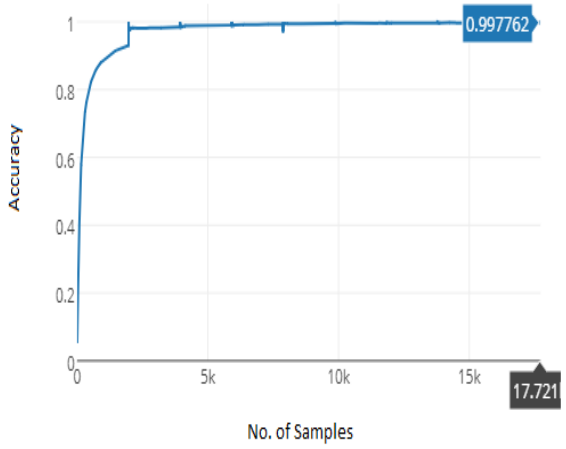



**Figure 6 Training accuracy of LSTM-CNN**

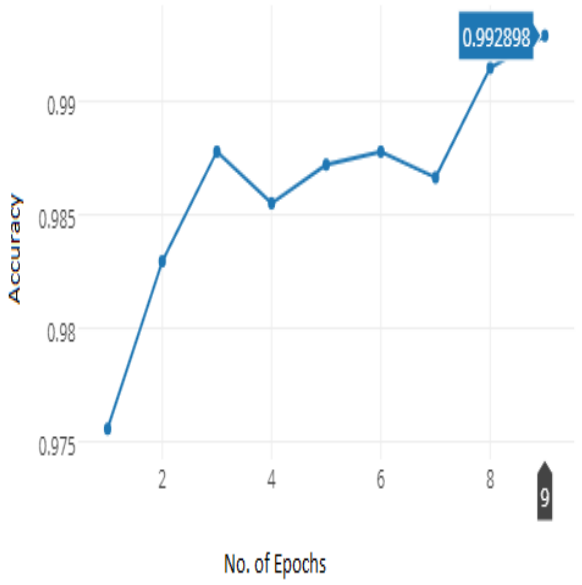



**Figure 7 Validation Accuracy of LSTM-CNN**

### B. *Breast Cancer IDC Dataset*

Breast cancer is one amongst the common forms of cancer in females. The manual evaluation of the presence of invasive ductal carcinoma (IDC) tissue regions present in the whole slide images (WSI) is a critical task, which can be assisted with the help of computerized evaluation. Since Invasive Ductal Carcinoma (IDC) is one of the most common type of a breast cancers, we use the IDC dataset produced by Cruz-Roa A et al. for evaluation of our proposed model [25]. This dataset consists of digital image patches that were derived from 162 patients. These images are small patches that were extracted from digital images of breast tissue samples. We utilize these images to detect the presence of IDC tissue regions in WSI.

Fig. 8 and Fig. 9 represents the training and validation accuracies of different models on the IDC dataset.

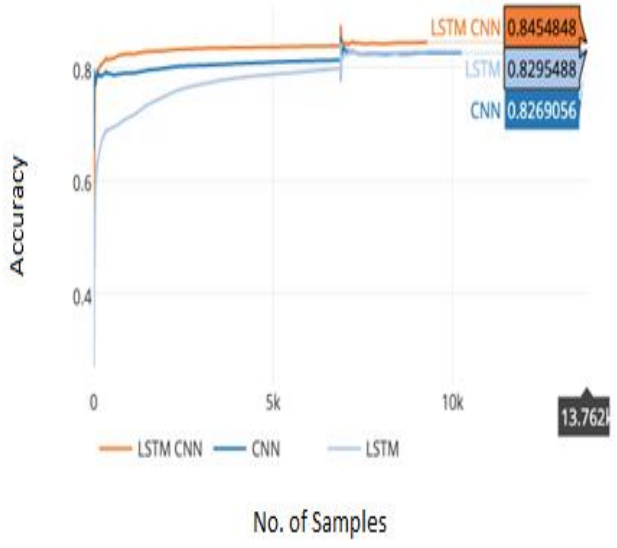



**Figure 8 Cancer IDC dataset training accuracies**

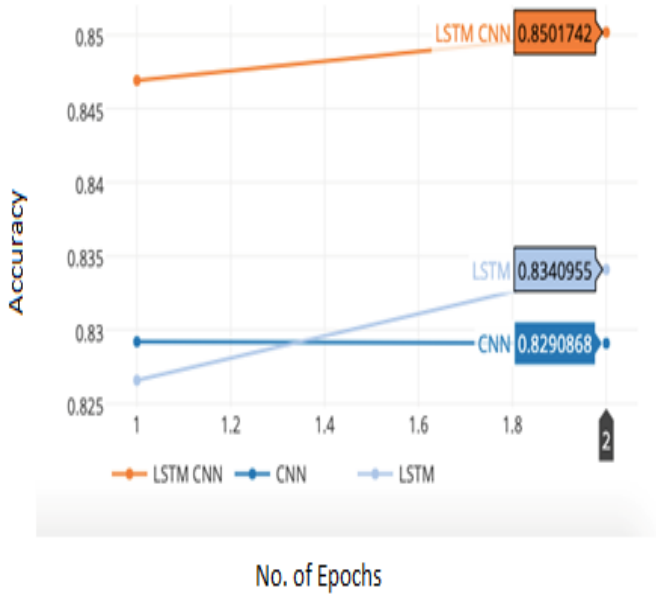



**Figure 9 Validation accuracies of IDC dataset**

Table II shows a comparison of our model with other classifiers. Our hybrid model achieves a training accuracy of 84.5% and a validation accuracy of 85% which is significantly better than the two other classifiers that it was compared against. These results hence set a new benchmark in this field.

**Table II. Accuracy comparison over IDC Breast Cancer Dataset**

| Model | Training Accuracy (%) | Validation Accuracy (%) |
|---|---|---|
| **LSTM + CNN** | **84.548** | **85.017** |
| LSTM | 82.955 | 83.410 |
| CNN | 82.691 | 82.909 |

## V. CONCLUSION

In this work, we have proposed a novel LSTM-CNN hybrid model for improving the accuracy of the image classification task. In comparison with other state-of-the-art classifiers like CNN, LSTM and hybrid CNN-LSTM, we found out that our proposed model significantly outperforms them. To establish the significance of our model, we tested it against two benchmark datasets i.e.

MNIST handwritten digit dataset and IDC Breast Cancer dataset. On both the datasets, our model gave remarkable accuracy. On MNIST dataset, the proposed LSTM-CNN hybrid model attained a training accuracy of 99.8% and a validation accuracy of 98.2%. On using the multiple LSTM-CNN layers it further gave us an improved validation accuracy of 99.29%. Similarly, benchmark results were obtained on

IDC Breast Cancer dataset with the attained training and validation accuracies of 84.5% and 85% respectively. Having attained a high accuracy with a single layer of LSTM-CNN, the model lays the foundation for further improvements by utilizing its multiple layers and controlling the overfitting in the presence of powerful GPUs and optimized drop out layers.

## REFERENCES


1. Sharma, A., Hua, G., Liu, Z. and Zhang, Z., 2008, June. Meta-tag propagation by co-training an ensemble classifier for improving image search relevance. In 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (pp. 1-6). IEEE.
2. Wan, J., Wang, D., Hoi, S.C.H., Wu, P., Zhu, J., Zhang, Y. and Li, J., 2014, November. Deep learning for content-based image retrieval: A comprehensive study. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 157-166). ACM.
3. Diallo, A.D., Gobee, S. and Durairajah, V., 2015. Autonomous tour guide robot using embedded system control. Procedia Computer Science, 76, pp.126-133.
4. Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., Andriluka, M., Rajpurkar, P., Migimatsu, T., Cheng-Yue, R. and Mujica, F., 2015. An empirical evaluation of deep learning on highway driving. arXiv preprint arXiv:1504.01716.
5. Bae, S.H., Choi, I. and Kim, N.S., 2016, September. Acoustic scene classification using parallel combination of LSTM and CNN. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016) (pp. 11-15).
6. Hua, K.L., Hsu, C.H., Hidayati, S.C., Cheng, W.H. and Chen, Y.J., 2015. Computer-aided classification of lung nodules on computed tomography images via deep learning technique. OncoTargets and therapy, 8.
7. Miranda, E., Aryuni, M. and Irwansyah, E., 2016, November. A survey of medical image classification techniques. In 2016 International Conference on Information Management and Technology (ICIMTech) (pp. 56-61). IEEE.
8. Claudiu Ciresan, D., Meier, U., Gambardella, L.M. and Schmidhuber, J., 2010. Deep big simple neural nets excel on handwritten digit recognition. arXiv preprint arXiv:1003.0358.
9. Michie, D., Spiegelhalter, D.J. and Taylor, C.C., 1994. Machine learning. Neural and Statistical Classification, 13.
10. LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), pp.2278-2324.
11. Chan, T.H., Jia, K., Gao, S., Lu, J., Zeng, Z. and Ma, Y., 2015. PCANet: A simple deep learning baseline for image classification?. IEEE transactions on image processing, 24(12), pp.5017-5032.
12. Bengio, Y., Lamblin, P., Popovici, D. and Larochelle, H., 2007. Greedy layer-wise training of deep networks. In Advances in neural information processing systems (pp. 153-160).
13. Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
14. Hassannejad, H., Matrella, G., Ciampolini, P., De Munari, I., Mordonini, M. and Cagnoni, S., 2016, October. Food image recognition using very deep convolutional networks. In Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management (pp. 41-49). ACM.
15. Chen, J.C., Patel, V.M. and Chellappa, R., 2016, March. Unconstrained face verification using deep cnn features. In 2016 IEEE winter conference on applications of computer vision (WACV) (pp. 1-9). IEEE.
16. Zhong, Z., Jin, L. and Xie, Z., 2015, August. High performance offline handwritten chinese character recognition using googlenet and directional feature maps. In 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (pp. 846-850). IEEE.
17. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A. and Oliva, A., 2014. Learning deep features for scene recognition using places database. In Advances in neural information processing systems (pp. 487-495).
18. He, K., Zhang, X., Ren, S. and Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE international conference on computer vision (pp. 1026-1034).
19. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), pp.211-252.
20. Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.
21. Byeon, W., Breuel, T.M., Raue, F. and Liwicki, M., 2015. Scene labeling with lstm recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3547-3555).
22. Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics, 36(4), pp.193-202.
23. Hubel, D.H. and Wiesel, T.N., 1968. Receptive fields and functional architecture of monkey striate cortex. The Journal of physiology, 195(1), pp.215-243.
24. Matsugu, M., Mori, K., Mitari, Y. and Kaneda, Y., 2003. Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Networks, 16(5-6), pp.555-559.
25. Cruz-Roa, A., Basavanhally, A., González, F., Gilmore, H., Feldman, M., Ganesan, S., Shih, N., Tomaszewski, J. and Madabhushi, A., 2014, March. Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. In Medical Imaging 2014: Digital Pathology(Vol. 9041, p. 904103). International Society for Optics and Photonics.
26. Sinha, P., Balas, B., Ostrovsky, Y. and Russell, R., 2006. Face recognition by humans: Nineteen results all computer vision researchers should know about. Proceedings of the IEEE, 94(11), pp.1948-1962.
27. Vu, A., Ramanandan, A., Chen, A., Farrell, J.A. and Barth, M., 2012. Real-time computer vision/DGPS-aided inertial navigation system for lane-level vehicle navigation. IEEE Transactions on Intelligent Transportation Systems, 13(2), pp.899-913.
28. LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. nature, 521(7553), p.436.
29. Cireşan, D., Meier, U. and Schmidhuber, J., 2012. Multi-column deep neural networks for image classification. arXiv preprint arXiv:1202.2745.
30. Tai, K.S., Socher, R. and Manning, C.D., 2015. Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075.
31. Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.


## AUTHORS PROFILE

**Aditi Arora** is an undergrad student at SRM Institute of Science and Technology where she is currently pursuing her bachelor's degree in Computer Science and Engineering. Her research interests include big data, Machine Learning, Data Analytics and Natural Language Processing. Aditi has been actively involved in various research endeavors with an aim of holistic development and advancement of society.

**Mayank Kumar Nagda** is an alumnus of SRM Institute of Science and Technology where he earned a bachelor's degree in the field of Computer Science and Engineering. Artificial Intelligence, Natural Language Processing, and Data Analytics are some of his areas of interests and expertise. Mayank is also fascinated by the idea of introducing automation in the regular lives of human beings so that it can assist and can create a new firm base for the next human evolution.

**Dr. E. Poovammal** is a Professor in the Department of Computer Science and Engineering at SRM Institute of Science and Technology. She joined in SRM in the year 1996. Before joining SRM, she worked in industry for five years. She obtained her B.E. Degree in Electrical and Electronics Engineering from Madurai Kamaraj University in the year 1990, M.E degree in Computer Science and Engineering from Madras University in the year 2002 and Ph.D. degree in Computer Science and Engineering from SRM University in 2011. Her research interests include data Big Data Analytics and machine learning. She is certified as Adjunct Faculty by Institute of software Research, Carnegie Mellon University, Pittsburgh, USA. She has published more than 40 referred journals and presented various international and national conferences.