Glacier Movement Prediction with Deep Learning Models and Satellite Data

Jonas Müller

April 18, 2023

Abstract

Contents

1 Introduction

Glaciers and their inherent dynamics are a well documented indicator for global climate change impact (Abram et al., 2019; Avian et al., 2020; Berthier et al., 2023; Dyurgerov & Meier, 2000; Mazhar et al., 2021; Scambos et al., 2017; Shrestha et al., 2015; Stoffel & Huggel, 2012; Yu et al., 2023; Zemp, 2008). While in addition 10 % of the Earth's land area is covered by glaciers or ice sheets, which in total hold about 69% of Earth's freshwater (Gleick, 1996), it is of crucial importance to study and monitor glacial changes over time. In adition to global sea level changes due to cryospheric changes, glacier melting can have local impacts on close regions like landslides, debris flows, rock slope failures, or ice avalanches (Stoffel & Huggel, 2012). In order to study the impacts of glacial change advanced remote sensing approaches have been developed cosisting of e.g. terrestial laserscanning, radar satellite applications, unmanned aerial vehicels, automatic camera imaging and *multispectral satellite imaging* (Avian et al., 2020). In multispectral satellite imaging the spectral reflectance of glacier surfaces is captured by sensors on the satllite. These sensors are sensitive to different bands in the light reflectance spectrum and are then used to estimate the composition of the underlying terrain, e.g. snow, ice, rocks or water (Kääb et al., 2014). From the green ($0,53-0,59\mu m$, resolution: 30 m) and the shortwave infrared band ($1.57-1.65\mu m$, resolution 30m) the *Normalized Snow Difference Index* [NDSI] can be calculated for example as a measure indicating snow/ice covered land masses.

Many models of glacier dynamics explicitly focus on the physical processes inside glaciers (Berthier et al., 2023; Colgan et al., 2016). This aproach comes with the disadvantage of varying estimates of diffferent models, varying uncertainties of different data sources, varying estimates of the same models by different research groups and the need of required explicit knowledge of the underlying processes for parameter estimation and data prediction. Therefore deep learning methods have been deployed for glacier movement prediction from satellite data (Min, Mukkavilli, & Bengio, 2019), where glacier dynamics are modeled indirectly through input-output relations. Due to the ability to learn complex recurrent patterns and the emergence and improvement of advanced neural network architectures in the theoretical deep learning literature, e.g. long short-term memory networks (Hochreiter & Schmidhuber, 1997), convolutional autoencoders (Ronneberger, Fischer, & Brox, 2015a) or transformer models (Vaswani et al., 2017), the use of higher computational resources in model training (Anthony, Kanding, & Selvan, 2020) and larger datasets, the deep learning research domain has made large progresses, especially in the field of computer vision (Chai, Zeng, Li, & Ngai, 2021). Based on

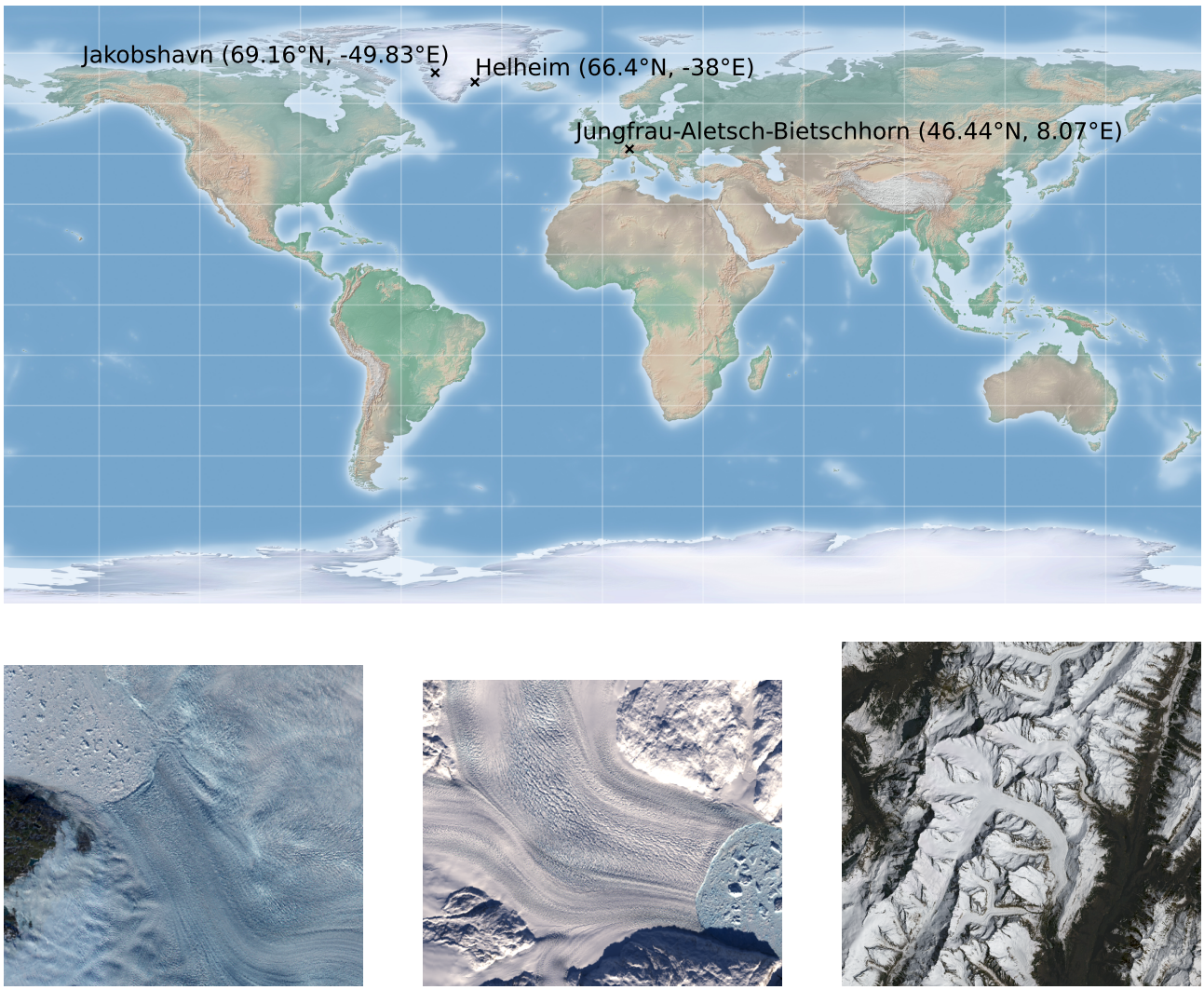

Figure 1. Overview of the studied glaciers. The Images are extracted from Landsat-8 spectral bands and plotted in RGB coordinates.

these promising findings the present study therefore focuses on the application of recurrent deep learning models to time series satellite image data of different glaciers. For this task landsat-8 satellite image data (figure 1), is used, with which deep learning models are trained and tested. The used models are adapted from the theoretical literature in computer vision deep learning research, representing state of the art approaches in image processing.

1.1 Related Work

While many studies are focused on satellite image classification with machine learning approaches [e.g. Chu et al. (2022); Marochov, Carbonneau, and Stokes (2020); Prieur, Rabatel, Thomas, Farup, and Chanussot (2022)], few studies are exploring the prediction of future glacier states with recurrent neural networks [RNN]. Many studies on the other hand focus on the pre-

diction of sea ice motion with RNN [e.g. Mu, Luo, Yuan, and Liang (2023); Petrou and Tian (2017, 2019); Zhai and Bitz (2021)]

Min et al. (2019) for example are stressing the importance of machine learning applications in the realm of climate science problems. In their study the authors developed a method to predict ice flow movements of the byrd glacier in antarctica with satellite data from the landsat-8 imaging system. The authors used correlations of image patches in a given scene to patches from another scene to find a matching target patch as ground truth for ice movement, thus tracking ice flow in a dynamic region of interest between scenes. The authors used 2 long term-short term memory [LSTM] layers in order predict scene patches with stochastic video generation, thereby conditioning a prior network on previous subscenes.

Petrou and Tian (2017) used a LSTM encoder-decoder architecture (Srivastava, Mansimov, & Salakhudinov, 2015) in order to predict future patches of artic sea ice motion satellite imaging scenes. They used the Advanced Microwave Scanning Radiometer - Earth Observing System [AMSR-E] from NASA's aqua satellite in order to calculate optical flow patches for each pair of consecutive days. The authors improved this network by using convolutional LSTM cells, which switches multiplication operations in the LSTM cell with convolution operations and works on the images directly instead of vector representations (Petrou & Tian, 2019)

Ali, Huang, Huang, and Wang (2021) used an attention based ensemble LSTM approach in order to predict sea ice motion from a mix of ERA-5 global reanalysis product data and Nimbus-7 satellite data. The authors then used both monthly and daily data in two sepearte LSTM layer branches, where each branch has an attention mechanism that calculates attention weights for the different hidden states, therefore evading the bottleneck problem of accumulating context information in LSTM models in the hidden state representation and instead using full self attention in order to not loose any important context information.

Mu et al. (2023) furthermore developed a LSTM-transformer hybrid model, the *ice temporal fusion transformer* for sea ice motion prediction. The network in addition uses physical variables like cloud-microphysics, thermodynamics or radiation as input to the network in order to predict sea ice changes over time. Up to this timepoint no research papers are known that use transformer neural networks for glacier movement prediction from satellite image data. Therefore this paper presents an unique opportunity to provide a contribution in the understanding of a large climate change detection indicator.

2 Methodology

2.1 Dataset

The generated dataset consists of landsat-8 level 2 satellite images of multiple glaciers. For the acquisition of the image data the SpatioTemporal Asset Catalog [STAC] was used. The

catalog can be accessesd with an API using Microsofts Planetary Computer, a cloud consisting of environment data for scientific investigations. The satellite images (*Scenes*) were filtered for cloud coverage and amount of missing values. Scenes were excluded if they had more than 20 % of cloud coverage and more than 50 % of missing pixel values. The scenes come with 19 bands consisting of 11 different spectral bands with a spatial resolution of $30m$ per pixel, measuring light reflectance spectra, and different bands supplying additional information, e.g. a band giving spatial information of cloud coverage for pixels. From the spectral bands 2 bands were used to estimate glacier movements: the green band ($0,53 - 0,59\mu m$, resolution: 30 m) and the shortwave infrared band ($1.57 - 1.65\mu m$, resolution 30m). These bands are used to create the NDSI. This index quntifies a continuous measure of snow/ice. In addition He, Zhang, Ma, and Wu (2020) already used the NDSI in order to classify glaciers with a U-net architecture. In order to further quantify snow/ice in pixels a threshold of 0.3 is used (Vonica, Ancuta, & Frincu, 2021). With this threshold masks are generated, indicating snow, where the value for each pixel is generated as:

$$p_{i,z} = \begin{cases} NDSI(p_{i,z}) & NDSI(p_{i,z}) \geq 0.3 \\ 0 & else, \end{cases}$$

In order to capture each glacier fully, large regions are used, e.g. 800x800 pixels for the parvati glacier. Because of that, the scenes are sampled in patches of size (50, 50) to decrease the computational load of the used models. In order to generate the final model input, patches of the same coordinates in different scenes are used over time, therefore sampling the same region at different timepoints. The models get 4 patches from different consecutive scenes as an input $\{\mathbf{a}_{k,t} \in A, \mathbf{b}_{k,t+1} \in B, \mathbf{c}_{k,t+2} \in C, \mathbf{d}_{k,t+3} \in D,\ k \in K, t \in T\}$ and predict the next 4 patches in the same coordinates $\{\mathbf{f}_{k,t+4} \in F, \mathbf{g}_{k,t+5} \in G, \mathbf{h}_{k,t+6} \in H, \mathbf{i}_{k,t+7} \in I,\ k \in K, t \in T\}$, where k denotes the patch number and t the time index. Because the intervals between scenes are not evenly spaced (e.g. exclusions because of clouds or missing data) the scenes are averaged over months (3 months in 1. experiment, 6 months in second experiment). Thus the model input and target dimensions, ignoring the batch dimension, in summary are 4x50x50 tensors.

The different bands of the satellite contain some amount of missing pixel values which is why a kernel is used to average out missing values. The procedure is adapted from Vonica et al. (2021). The Kernel selects missing pixels, places a 5x5 kernel on top of the missing value, replaces all missing pixels in the respective image space the kernel is targeted at with 0, and calculates a weighted average for the missing pixel in question.

Because landsat-8 imaging is not completely accurate (Vonica et al., 2021), an allignment procedure is used to compensate for pixel shifts in images to produce more robust model predictions. For this the enhanced correlation coefficient maximization algorithm is used (Evangelidis & Psarakis, 2008). The algorithm iteratively refines an estimate of alignment paramters, e.g. image translation transformations by maximiziing the *enhanced correlation coefficient* between

two images. The algorithm used $\min(10000, x)$ iterations and a stopping criterion of .0001. First a mean image (median-filtered) is created with all images of the scene and then all scenes are aligned to this mean image with e.g. rotation, translation, scaling and shearing transformations through a homography matrix.

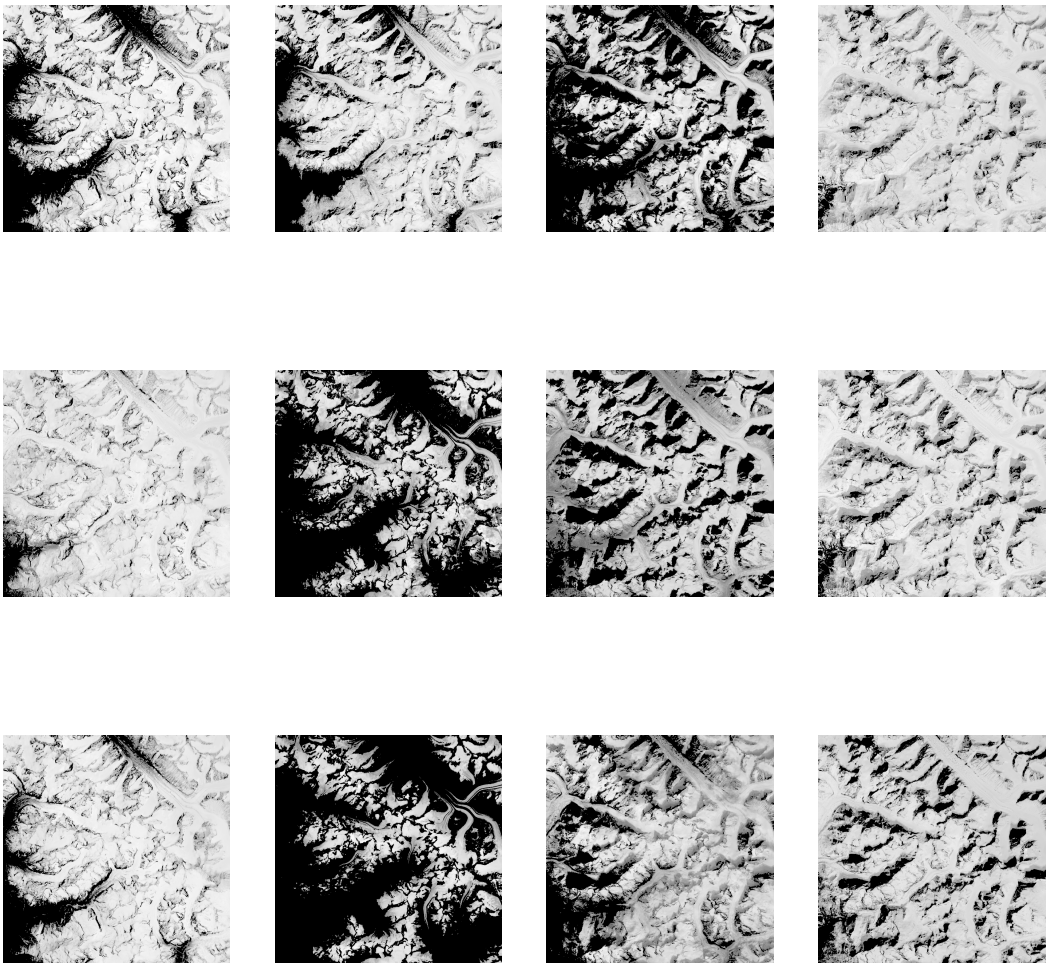

Figure 2. Extracted and aligned scene masks of the parvati glacier from April 2013 until March 2015.

Glacier movements (figure 2) are characterized by seasonal and annual changes Sam, Bhard-

waj, Kumar, Buchroithner, and Martín-Torres (2018). Figure 3 in addition shows the estimated optical flow between two consecutive scenes. For the optical flow estimation the dense optical flow algorithm by Farnebäck (2003) is used. In comparison to sparse optical flow algorithms which use only a small subset of pixels for motion estimation, dense optical flow algorithms estimate the movement of every pixel in the image. The algorithm uses a image pyramid which smoothes and resamples the image at multiple levels (resampling factor: 0.5, levels of pyramid: 6, thus decreasing the size of the image by a factor of 0.5). At each level of the pyramid the algorithm uses quadratic polynomials in order to estimate the local local brightness changes between the two images. The algorithm uses 10 iterations at each level of the pyramid. All other parameters are adapted to their standard values (Vonica et al., 2021). Figure 3 shows two aligned consecutive scenes of the parvati glacier and estimated optical flow vectors. In order not to populate the whole scene with motion vectors the vectors are only drawn every 40 pixels (1200 meter).

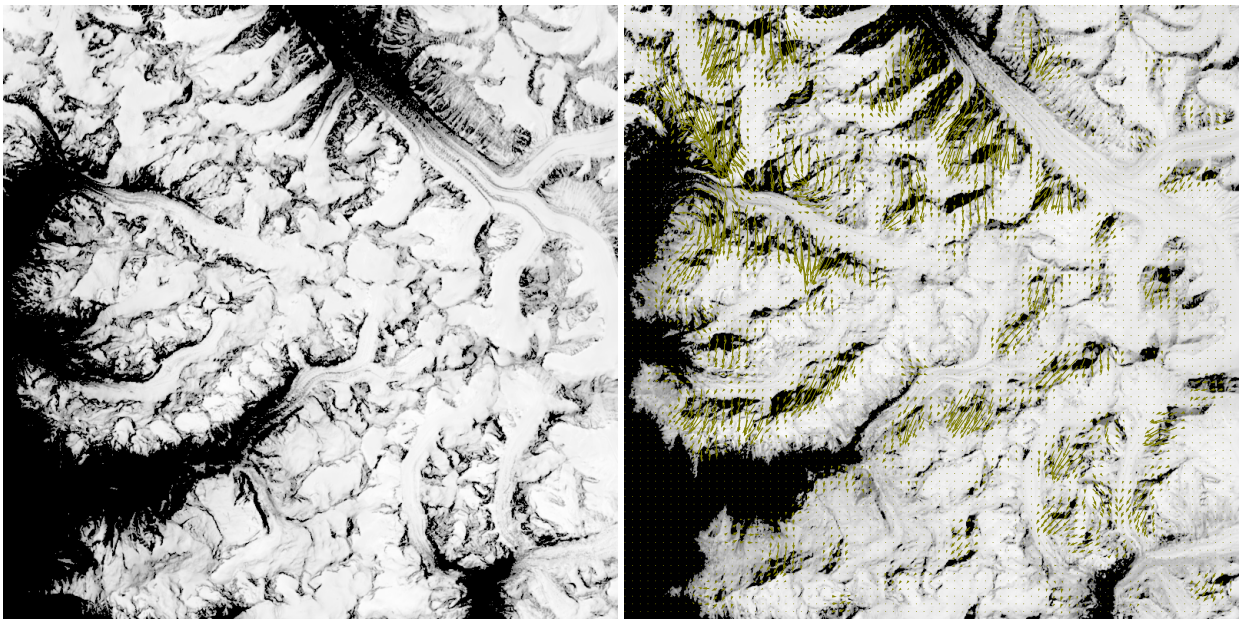

Figure 3. Dense optical flow estimated from two consecutive scenes with $\Delta_t = 3$ months (left: average over April, Mai and June 2013; right: Juli, August and September 2013) of the parvati glacier.

2.2 Models

While some of the used models are adapted from the theoretical literature and are inspired by previous work on sea-ice motion prediction (e.g. Petrou and Tian (2017, 2019)), therefore using LSTM and convolutional LSTM cells, other models are used because of their successes in other vision tasks, e.g. the U-net architecture (Ronneberger, Fischer, & Brox, 2015b) in biomedical image segmentation. Interestingly Holzmann et al. (2021) used an U-net with an attention mechanism for glacier calving front segmentation, showing the applicability of U-net to satellite glacier data. The third class of used models consists of transformer models. These are selected because of their recent success in language processing, image classification and video prediction (Khan et al., 2022). In addition transformer models have the added benefit of no vanishing gradient problem, no fuzzy context accumulation problems because of information accumulation, e.g. in LSTM cells, and the capacity to generalize to arbitrary long sequences by design of the architecture (Vaswani et al., 2017).

As described above, all models get 4 consecutive scene patches as input and predict the next 4 consecutive patches in the same coordinates. The models are trained with the mean squared error loss function [MSE] over all predicted future patches resulting in:

$$MSE = \frac{\sum_i^N \sum_t^T \sum_x^X \sum_y^Y (\tilde{y}_{i,t,x,y} - y_{i,t,x,y})^2}{N \times T \times X \times Y}, \tag{1}$$

where $\tilde{y}$ represents the snow mask prediction of the model and y represents the snow mask ground truth. In addition $i \in N$ represents the index of the training example, $t \in T$ the time index, $x \in X$ the image width coordinate and $y \in Y$ the image height coordinate.

2.2.1 LSTM model

The first model uses the LSTM cell developed by Hochreiter and Schmidhuber (1997). The LSTM cell has four different gates that are used recursively in order to predict sequential data (equation 2). The LSTM cell gets a hidden state representation, a cell state representation and a flattened input patch vector as an input. The central idea is that context information is accumulated and integrated at each timestep in the hidden and cell states. In the beginning the hidden and cell states are initialized as tensors containing 0 values. The different gate activations are then calculated with the following equations (note that vectors and matrices are bold):

$$
\begin{aligned}
\mathbf{i} &= \sigma(\mathbf{W}_{ii}\mathbf{x} + \mathbf{b}_{ii} + \mathbf{W}_{hi}\mathbf{h} + \mathbf{b}_{hi}),\\
\mathbf{f} &= \sigma(\mathbf{W}_{if}\mathbf{x} + \mathbf{b}_{if} + \mathbf{W}_{hf}\mathbf{h} + \mathbf{b}_{hf}),\\
\mathbf{g} &= \tanh(\mathbf{W}_{ig}\mathbf{x} + \mathbf{b}_{ig} + \mathbf{W}_{hg}\mathbf{h} + \mathbf{b}_{hg}),\\
\mathbf{o} &= \sigma(\mathbf{W}_{io}\mathbf{x} + \mathbf{b}_{io} + \mathbf{W}_{ho}\mathbf{h} + \mathbf{b}_{ho}),\\
\mathbf{c}' &= \mathbf{f} \odot \mathbf{c} + \mathbf{i} \odot \mathbf{g},\\
\mathbf{h}' &= \mathbf{o} \odot \tanh(\mathbf{c}'),
\end{aligned}
\tag{2}
$$

where $\mathbf{W}$ consitutute weight matrices, $\mathbf{b}$ additive biases, $\mathbf{h}$ hidden state representations, *tanh* the tangent hyperbolic activation function, σ the sigmoid activation function, $\mathbf{c}'$ the new cell state, $\mathbf{h}'$ the updated hidden state, $\mathbf{x}$ the new incoming input and $\odot$ signifies the Hadamard product. The indices of the weight matrices and bias vectors represent firstly, if the weight matrix is multiplied by the input or the hidden state, and secondly, to which gate the matrix belongs to. In addition the different gates are connected through *peephole connections* (Gers, Schraudolph, & Schmidhuber, 2002), therefore allowing information flow between gates in each cell state update.

The differnt gates were explicitly designed for different functionalities. The forget gate ($\mathbf{f}$) controls how much of the previous hidden state is remembered by integrating the last hidden state with the input $\mathbf{x}$ by outputting values between 0 and 1 on a continuous scale qunatifying remembering (towards 1) and forgetting (towards 0). The input gate ($\mathbf{i}$) integrates the input with the last hidden state in order to quantify how much of the new information is integrated into the cell state. The output gate ($\mathbf{o}$) quantifies how much of the calculated cell state is integrated in the new hidden state (which is also the output of the LSTM cell for each timestep). The candidate gate ($\mathbf{g}$) quantifies how much of the input gate is added to the cell state.

In the optimization process the backpropagation algorithm is then used across timesteps, thus gradients are accumulated across timesteps backwards through time, so called *backpropagation through time*. The LSTM cell does not suffer from vanishing gradients by design as the gating mechanism in the forget gate controls the gradient flow in the internal recurrence of the cell. This can be seen in the derivative of the cell state:

$$
\frac{\partial \mathbf{c}'}{\partial \mathbf{c}} \mathbf{f} \odot \mathbf{c} + \mathbf{i} \odot \mathbf{g} = \mathbf{f}, \tag{3}
$$

as the derivative of the cell state with respect to itself is always a vector of 1 (the so called *constant error carousel* (Hochreiter & Schmidhuber, 1997)) multiplied by the forget gate output. If the biases in the forget gate are initialized with high values, the gradients never vanish or explode as both bounds are regulated by the forget gate, which learns its activation across training.

The used model is an encoder decoder architecture with LSTM cells (3 encoder, 3 decoder

cells), which uses the input in the encoder, copys cell and hidden states to the decoder and then recurrently predicts into the future by using the model predictions from previous timesteps.

2.2.2 Convolutional LSTM Network

The next model has a similar structure to the LSTM network described above, with the difference, that now instead of vector representations of image patches, the patches itself are fed into the network. Therefore all the matrix multiplication operations now become convolution operations:

$$\begin{aligned}
\mathbf{i} &= \sigma(\mathbf{W}_{ii} * \mathbf{x} + \mathbf{W}hi * \mathbf{h} + \mathbf{b}_{hi}), \\
\mathbf{f} &= \sigma(\mathbf{W}_{if} * \mathbf{x} + \mathbf{W}_{hf} * \mathbf{h} + \mathbf{b}_{hf}), \\
\mathbf{g} &= \tanh(\mathbf{W}_{ig} * \mathbf{x} + \mathbf{W}_{hg} * \mathbf{h} + \mathbf{b}_{hg}), \\
\mathbf{o} &= \sigma(\mathbf{W}_{io} * \mathbf{x} + \mathbf{W}_{ho} * \mathbf{h} + \mathbf{b}_{ho}), \\
\mathbf{c}' &= \mathbf{f} \odot \mathbf{c} + \mathbf{i} \odot \mathbf{g}, \\
\mathbf{h}' &= \mathbf{o} \odot \tanh(\mathbf{c}'),
\end{aligned} \tag{4}$$

The used network also uses a encoder-decoder structure with a recurrent future prediction module.

2.2.3 Convolutional U-net Autoencoder

The network structure of the U-net follows a contracting and an expanding path in a fully convolutional architecture with 23 layers (Ronneberger et al., 2015b). The main idea of the network is to increase the channel dimensions in the contracting path in such a way that information of the image is abstracted into a large number of feature maps, while descending into the latent space with decreasing image size. The used skip connections project intermediate feature maps from the same part of the contracting to the expanding path horizontally, therefore protect the network from information loss over long distances in the network and increase stability of predictions.

The contracting path uses a typical convolutional network architecture consisting of repeated 3x3 convolutions, ReLU activations, and 2x2 max pooling operations with downsampling. At each downsampling step, the number of feature channels is doubled. In the expansive path the feature maps are upsampled followed by a 2x2 convolution that halves the number of feature maps again, then the cropped feature map (necessary due to the loss of border pixels in every convolution) from the corresponding part of the contracting path is concatenated and two 3x3 convolutions are performed, which are followed by ReLu activation functions. The final layer consists of a 1x1 convolutional layer in order to map the channel dimension back to 1 for prediction.

The used model is adapted in order to predict patches over time with a future prediction module. Therefore the model always predicts the next patch from the patch of the previous timestep recurrently.

2.2.4 Self Attention LSTM Hybrid Model

The next model stems from a recently developed class of models, the so called *transformer* models, that show promising results in e.g. natural language processing (Vaswani et al., 2017), computer vision (Khan et al., 2022) or audio processing tasks (Gong, Chung, & Glass, 2021). At the core of the architectures attention mechanisms are used, which side step the bottleneck of accumulating context information and exploding or vanishing gradients and instead use a mechanism that attends to all elements in a sequence in parallel. One prominent example of an atttention mechanisms is the so called *multi-head self attention* [MHSA]. In analogy to database systems the attention mechanism uses three matrices, the query ($\mathbf{Q}$), the key ($\mathbf{K}$) and the value ($\mathbf{V}$) matrix. These three matrices are multiplied by three copies of the input sequence. After that, the query and the key matrices are combined to form *attention weights*. The matrices are here multiplied, scaled by the square root of the dimensionality of the key matrix and fed through a softmax activation function. These calculated weights then control the attention of the system to different parts of the value matrix:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{5}$$

In order for the attention mechanism to become multi headed this process is repeated multiple times, where the outputs are concatenated and then fed through linear layers to decrease dimensionality for further processing. Therefore the mechanism can abstract hierarchical patterns from the input sequences.

In the search process of finding a suitable arhcitecture for the task at hand, multiple architectures have been tried, e.g. a language transformer model (as descirbed by (Vaswani et al., 2017)) without tokenizer and flattened scene masks as input, a language transformer model with multilayer perceptron or convolutional tokenizer, a vision transformer model (Dosovitskiy et al., 2020) with future prediction module (predicting future frames from last frames) and a self attention LSTM hybrid [SA-LSTM-H] Model. The model architecture search for the attention based model class was based on the data of the first experiment descirbed below. The only model producing visually sound results, not prone to overfitting because of high model capacity and good validation set generalisation was the SA-LSTM-H.

The SA-LSTM-H, inspired by Mu et al. (2023), processes the flattened input patches in the LSTM layer (3 LSTM cells) in order to generate hidden state representations (last LSTM cell). These are then fed into a MHSA layer which produces a matrix of size (ignoring batch dimension) 4x2500 (timesteps t_{-3} to t). This matrix is then flattened to a 1x10000 vector which

is then fed through a linear layer and projected to a size of 1x2500, producing the output for timestep t_{+1}. This vector is then concatenated to the output of the hidden states of the LSTM layer. The LSTM layer then recursively predicts the next hidden state outputs using timesteps t_{-2} until t_{+1}. This process is repeated until all future timesteps are predicted.

3 Experiment 1: Parvati Glacier

3.1 Experimental Design

The Parvati glacier is well suited for the prediction of glacier movements, because first the landsat-8 image catalog provides a high temporal density of scenes that are almost evenly distributed over the months in a year, and second because of the high glacier density in the region and the surroundings. Therefore a dataset was created in which scenes are averaged over a Δ_t of three months of each year from January 01, 2013 until January 01, 2021. If for a given month no scenes were available the missing months were interpolated linearly in order to prevent temporal bias in the data (one month in sequence). The patches were extraced from the provided coordinates (1) in regions of interest of 800x800 pixels with a stride of 10, resulting in 161 728 sequences of 4 scenes and 4 corresponding targets. the data was then splitted into 80% training and 20% test data, while 10% of the training data was used for validation. The dataset was firstly used in order to test the different model architectures and get an understanding of the best performing architectures with the validation set performance. Afterwards generalization performance was tested on the test set. As data augmentation techniques teacher forcing, dropout and weight decay were used (see table 1 for hyperparameter settings). All models except the convolutional LSTM [convLSTM] network model were trained on nvidia RTX 2080ti graphics processing units, while the convLSTM was trained on a nvidia v100 gpu bcaue of higher model capacity.

Table 1

Model Parameters

Model	Learning Rate	Weight Decay	Epochs	Batch Size	Optimizer
LSTM	.0001	.001	40	100	adamW
U- net	.001	.01	20	100	adamW
SA-LSTM-H	.0001	.001	40	100	adamW
ConvLSTM	.0001	.001	40	100	adamW

3.2 Results

Except for the U-net architecture, all other models seem to capture the underlying data distribution.

Table 2

Testset Performance

	MSE	*MAE*
U-net	0.083	0.192
MA-LSTM-H	0.047	0.136
LSTM Encoder-Decoder	0.054	0.139
ConvLSTM	0.084	0.209

3.3 Discussion

4 Experiment 2: Jungfrau-Aletsch-Bietschhorn Glacier

5 Experiment 3: Generalisation to Full Scenes

5.1 Full Scene Learning

In a next step the models are retrained and fine-tuned in order to predict full scenes across time. In order to achieve this the models should learn smooth borders between patches of each scene in order to create a full consistent image. Thus a new dataset is used containing only non overlapping patches from consecutive scenes.

Two approchaes are tested to achieve this. In the first approach the model now takes a sequence of scenes, e.g. $\{A, B, C, D, E\}$ and predicts the next n patches for the next scenes, e.g. $\{F, G, H, I, J, ...\}$ in the future. Therefore the model input is now $\{a_k \in A, b_k \in B, c_k \in C, d_k \in D, e_k \in E\}\ \forall k \in K$, where K is the number of respective patches in each scene, while the model output is now $\{f_k \in F, g_k \in G, h_k \in H, i_k \in I, j_k \in J\}\ \forall k \in K$ (all patches with the same indices stem from the same coordinates in the scene). After the model predicted all patches for the next secenes they are put together again from the patches and a final MSE loss is calculated on the full scenes and propagated through the model. With this procedure the model then learns to smooth the edges of the patches in the full scene image.

5.2 U-net Denoising Autoencoder

5.3 Results

5.4 Generalisation Test

6 Discussion

dimensionality reduction techniques for image vector embeddings

other method to combine scenes

prediction forecasting warning system pakistan etc...

7 Limitations

carbon footprint of model training

combine with alignment on ground

8 Summary

References

Abram, N., Gattuso, J.-P., Prakash, A., Cheng, L., Chidichimo, M. P., Crate, S., ... others (2019). Framing and context of the report. *IPCC special report on the ocean and cryosphere in a changing climate*, 73–129.

Ali, S., Huang, Y., Huang, X., & Wang, J. (2021). Sea ice forecasting using attention-based ensemble lstm. *arXiv preprint arXiv:2108.00853*.

Anthony, L. F. W., Kanding, B., & Selvan, R. (2020). Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. *arXiv preprint arXiv:2007.03051*.

Avian, M., Bauer, C., Schlögl, M., Widhalm, B., Gutjahr, K.-H., Paster, M., ... others (2020). The status of earth observation techniques in monitoring high mountain environments at the example of pasterze glacier, austria: Data, methods, accuracies, processes, and scales. *Remote Sensing*, *12*(8), 1251.

Berthier, E., Floricioiu, D., Gardner, A. S., Gourmelen, N., Jakob, L., Paul, F., ... others (2023). Measuring glacier mass changes from space-a review. *Reports on Progress in Physics*.

Chai, J., Zeng, H., Li, A., & Ngai, E. W. (2021). Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, *6*, 100134.

Chu, X., Yao, X., Duan, H., Chen, C., Li, J., & Pang, W. (2022). Glacier extraction based on high-spatial-resolution remote-sensing images using a deep-learning approach with attention mechanism. *The Cryosphere*, *16*(10), 4273–4289.

Colgan, W., Rajaram, H., Abdalati, W., McCutchan, C., Mottram, R., Moussavi, M. S., & Grigsby, S. (2016). Glacier crevasses: Observations, models, and mass balance implications. *Reviews of Geophysics*, *54*(1), 119–161.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... others (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dyurgerov, M. B., & Meier, M. F. (2000). Twentieth century climate change: evidence from small glaciers. *Proceedings of the National Academy of Sciences*, *97*(4), 1406–1411.

Evangelidis, G. D., & Psarakis, E. Z. (2008). Parametric image alignment using enhanced correlation coefficient maximization. *IEEE transactions on pattern analysis and machine intelligence*, *30*(10), 1858–1865.

Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Image analysis: 13th scandinavian conference, scia 2003 halmstad, sweden, june 29–july 2, 2003 proceedings 13* (pp. 363–370).

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, *3*(Aug), 115–143.

Gleick, P. H. (1996). Water resources. *Encyclopedia of climate, weather*, 817–823.

Gong, Y., Chung, Y.-A., & Glass, J. (2021). Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*.

He, Q., Zhang, Z., Ma, G., & Wu, J. (2020). Glacier identification from landsat8 oli imagery using deep u-net. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *3*, 381–386.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Holzmann, M., Davari, A., Seehaus, T., Braun, M., Maier, A., & Christlein, V. (2021). Glacier calving front segmentation using attention u-net. In *2021 ieee international geoscience and remote sensing symposium igarss* (pp. 3483–3486).

Kääb, A., Bolch, T., Casey, K., Heid, T., Kargel, J. S., Leonard, G. J., ... Raup, B. H. (2014). Glacier mapping and monitoring using multispectral data. *Global land ice measurements from space*, 75–112.

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan, F. S., & Shah, M. (2022). Transformers in vision: A survey. *ACM computing surveys (CSUR)*, *54*(10s), 1–41.

Marochov, M., Carbonneau, P., & Stokes, C. (2020). Automated image classification of greenlandic outlet glaciers using deep learning: A case study on helheim glacier.

Mazhar, N., Mirza, A. I., Abbas, S., Akram, M. A. N., Ali, M., & Javid, K. (2021). Effects of climatic factors on the sedimentation trends of tarbela reservoir, pakistan. *SN Applied Sciences*, *3*, 1–9.

Min, Y., Mukkavilli, S. K., & Bengio, Y. (2019). Predicting ice flow using machine learning. *arXiv preprint arXiv:1910.08922*.

Mu, B., Luo, X., Yuan, S., & Liang, X. (2023). Icetft v 1.0. 0: Interpretable long-term prediction of arctic sea ice extent with deep learning. *Geoscientific Model Development Discussions*, 1–28.

Petrou, Z. I., & Tian, Y. (2017). Prediction of sea ice motion with recurrent neural networks. In *2017 ieee international geoscience and remote sensing symposium (igarss)* (pp. 5422–5425).

Petrou, Z. I., & Tian, Y. (2019). Prediction of sea ice motion with convolutional long short-term memory networks. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(9), 6865–6876.

Prieur, C., Rabatel, A., Thomas, J.-B., Farup, I., & Chanussot, J. (2022). Machine learning approaches to automatically detect glacier snow lines on multi-spectral satellite images. *Remote Sensing*, *14*(16), 3868.

Ronneberger, O., Fischer, P., & Brox, T. (2015a). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, *abs/1505.04597*. Retrieved from http://arxiv.org/abs/1505.04597

Ronneberger, O., Fischer, P., & Brox, T. (2015b). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241).

Sam, L., Bhardwaj, A., Kumar, R., Buchroithner, M. F., & Martín-Torres, F. J. (2018). Heterogeneity in topographic control on velocities of western himalayan glaciers. *Scientific reports*, *8*(1), 12843.

Scambos, T. A., Bell, R. E., Alley, R. B., Anandakrishnan, S., Bromwich, D., Brunt, K., ... others (2017). How much, how fast?: A science review and outlook for research on the instability of antarctica's thwaites glacier in the 21st century. *Global and Planetary Change*, *153*, 16–34.

Shrestha, M., Koike, T., Hirabayashi, Y., Xue, Y., Wang, L., Rasul, G., & Ahmad, B. (2015). Integrated simulation of snow and glacier melt in water and energy balance-based, distributed hydrological modeling framework at hunza river basin of pakistan karakoram region. *Journal of Geophysical Research: Atmospheres*, *120*(10), 4889–4919.

Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015). Unsupervised learning of video representations using lstms. In *International conference on machine learning* (pp. 843–852).

Stoffel, M., & Huggel, C. (2012). Effects of climate change on mass movements in mountain environments. *Progress in physical geography*, *36*(3), 421–439.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Vonica, M.-M., Ancuta, A., & Frincu, M. (2021). Glacier movement prediction through computer vision and satellite imagery. In *2021 23rd international symposium on symbolic and numeric algorithms for scientific computing (synasc)* (pp. 113–120).

Yu, A., Shi, H., Wang, Y., Yang, J., Gao, C., & Lu, Y. (2023). A bibliometric and visualized analysis of remote sensing methods for glacier mass balance research. *Remote Sensing*, *15*(5), 1425.

Zemp, M. (2008). *Global glacier changes: facts and figures*. UNEP/Earthprint.

Zhai, J., & Bitz, C. M. (2021). A machine learning model of arctic sea ice motions. *arXiv preprint arXiv:2108.10925*.