2021 IEEE Visualization Conference (VIS) | 978-1-6654-3335-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/VIS49827.2021.9623327

CloudFindr: A Deep Learning Cloud Artifact Masker for Satellite DEM Data

Kalina Borkiewicz* Viraj Shah† J.P. Naiman‡ Chuanyue Shen§ Stuart Levy¶ Jeff Carpenter‖

University of Illinois at Urbana-Champaign

Abstract

Artifact removal is an integral component of cinematic scientific visualization, and is especially challenging with big datasets in which artifacts are difficult to define. In this paper, we describe a method for creating cloud artifact masks which can be used to remove artifacts from satellite imagery using a combination of traditional image processing together with deep learning based on U-Net. Compared to previous methods, our approach does not require multi-channel spectral imagery but performs successfully on single-channel Digital Elevation Models (DEMs). DEMs are a representation of the topography of the Earth and have a variety applications including planetary science, geology, flood modeling, and city planning.

1 Introduction

Cloud detection in satellite imagery is a problem that has plagued scientists for decades (e.g. [7, 8, 19, 20, 23]). Whether a scientist's area of research is the clouds themselves or the land beneath them, it is useful to separate the two classes of objects, though a universal method for doing so remains elusive. Various methods have been proposed depending on the type of data (e.g. spectral [10, 18], time-evolving [5, 17]) and the specific research objective.

However, as of this writing no current cloud detection methods exist for when the objective is not scientific data analysis, but rather cinematic scientific visualization which aims to create aesthetically pleasing imagery for a general audience. A visualization created for outreach purposes requires a different focus in the underlying data processing in order to create a result that is not only understandable, but also visually appealing to the general public. Aesthetically-pleasing visualizations are both more educational [3] and are perceived to be more credible than those which are not [16].

Our work differs from other cloud detection methods primarily in two ways – the underlying data, which is limited to a 1-dimensional elevation model rather than 3+-dimensional spectral imagery; and the general purpose. The aim of our work is not data cleaning for purposes of data analysis, but rather to create a cinematic scientific visualization which enables effective science communication to broad audiences. Great care must be applied in visualizations of complex data for lay audiences, and additional data processing, camera choreography, and different methods of rendering are required to achieve a goal of clear communication [15].

The CloudFindr method described here can be used to algorithmically mask the majority of cloud artifacts in satellite-collected DEM data by visualizers who want to create content for documentaries, museums, or other broad-reaching science communication mediums, or by animators and visual effects specialists who want to use such DEM data to create realistic landscapes and backdrops in otherwise fictional computer-generated movie scenes.

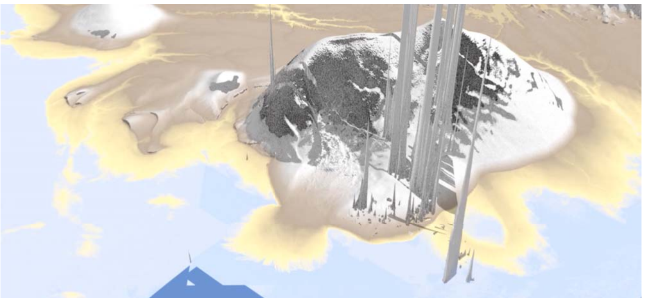

Figure 1: A 3D visualization of a DEM region without cloud artifact removal, showing large spikes where the height of the land is incorrectly labelled with the height of a cloud.

1.1 Cinematic Scientific Visualization in *Atlas of a Changing Earth*

When creating a public-facing outreach visualization for broad public distribution via films shown in giant immersive theaters (e.g. planetarium domes, IMAX screens), it is critical that data must be artifact-free. If the dataset in question is a digital elevation model (DEM) of land, clouds are considered to be artifacts and must be removed. A single cloudy DEM pixel, reprojected into 3D, would result in an unacceptable massive spike in the landscape that is sure to draw audience attention away from the immersive experience of the story (see Figure 1), especially on a 75+ foot screen.

The Advanced Visualization Lab (AVL) at the National Center for Supercomputing Applications encountered this problem when working on a documentary, *Atlas of a Changing Earth*, which features three locations visualized from the ArcticDEM dataset [13]. The motivation for the work described in this paper was the time-consuming manual cloud removal that was required in order to create a seamless, smooth, artifact-free cinematic visualization of the DEM data. Though some basic automatic cloud-removal techniques were used during the making of the documentary, they were not satisfactory, and the process still required weeks of manual review.

1.2 Cloudy DEMs

Digital elevation model data is represented as imagery, where each pixel holds a value for the elevation at that location. DEM data is often gathered by satellite, and in these cases, clouds become an issue. The annual global mean cloud cover is approximately 66% [21], so Earth-facing satellites which aim to study the land collect noisy or incorrect data when clouds obstruct their view. Making this data more usable requires significant data cleaning. For many scientific purposes, it is useful to retain as much of the collected data as possible during data cleaning, even if there is a small number of artifacts. In other words, there is higher value in the precision of the data than in the recall. For purposes of cinematic visualization, the opposite is true - recall is more important than precision. It is preferable to lose some correct data in the process of data cleaning

*e-mail: kalina@illinois.edu
†e-mail: vjshah3@illinois.edu
‡e-mail: jnaiman@illinois.edu
§e-mail: cs11@illinois.edu
¶e-mail: salevy@illinois.edu
‖e-mail: jdcarpen@illinois.edu

DOI 10.1109/VIS49827.2021.00008

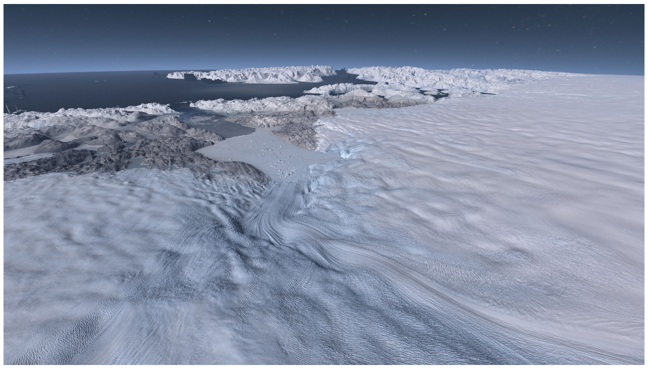

Figure 2: Final cloud-free cinematic rendering of the Jakobshavn glacier used in the *Atlas of a Changing Earth* documentary.

to ensure the removal of the artifacts, which are not only visually unappealing but also inaccurate representations of the data.

The ArcticDEM dataset covers all land north of 60°, and it redefined the arctic from the worst to one of the best-mapped regions of the Earth [12]. The data is collected by the WorldView 1-3 satellites and GeoEye-1 satellite, and is processed to remove clouds and other errors, with an absolute error of <0.5 meters [13]. The ArcticDEM project makes their derived DEMs readily available, but intellectual property considerations prevent release of the original optical imagery, hence the need for the innovations in this paper. The ArcticDEM data is collected in "strips", with each strip being a snapshot of a particular area. This can be thought of as a puzzle piece. Over time, the various puzzle pieces are gathered, and eventually there are enough to put together a complete puzzle. Because the strips are gathered at different points in time, putting them together does not create one single, seamless, final mosaic, but rather, puzzle pieces are periodically updated and replaced. This is where the puzzle analogy starts to break apart - a strip may cover some of the same area as a previous strip, but does not exactly "replace" a previous puzzle piece, as it may not have the same shape and coverage.

To build a cloud-free, complete mosaic for the *Atlas of a Changing Earth* documentary, cloud masks were manually created for each strip (described in Section 3.1) and multiplied against the data to remove the artifacts. The cloud-free strips were then accumulated to build up the mosaic, at which point the visualization video begins. The strips continue to update throughout the visualization.

Cloud detection was a manual, time-consuming process during the documentary production, however, it produced a valuable output in addition to the visualization itself - a large collection of labelled data. Detecting clouds in DEM data has a unique set of challenges: clouds may be be a small cluster of pixels or may cover the whole strip and beyond; strips that have hard edges may cut through features, so there is no guarantee that even a cumulus cloud is complete and has an organic, recognizable outline; haze and clouds that are low to the ground may be difficult to distinguish from land and may create noise which is not otherwise identifiable as "clouds"; and there is only a single channel of data per pixel, unlike in multispectral imagery, which is most commonly used for cloud detection [11].

2 Related Work

Cloud detection is a specific application of the broader field of anomaly detection with methods spanning different techniques and applications. Techniques range from information theoretic to classification-based to statistical; applications span cyber-intrusion detection to image processing to sensor networks [6]. Deep learning methods can be applied to anomaly detection using algorithms that are supervised, unsupervised, hybrid, or one-class neural networks [4]. An issue when attempting anomaly detection with spatiotemporal data is that there is often a lack of a clear boundary between normal and abnormal cases [2] – in the case of cloud detection, it can be difficult to determine if a pixel contains a cloud, or a snow-peaked mountain.

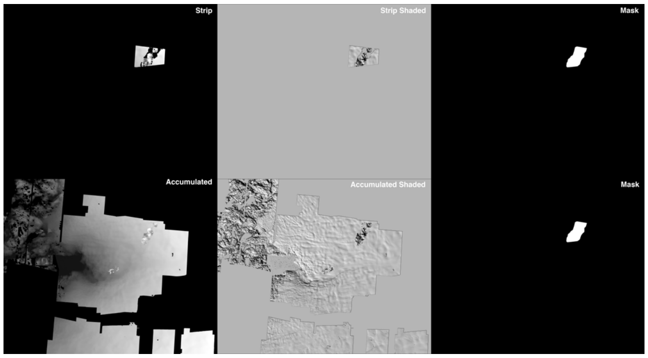

Figure 3: Example showing the inputs (left, middle) used to output a hand-drawn mask (right) for one sample timestep. Top row shows individual strips, bottom row shows accumulated buildup of strips. Left column shows DEM data, middle column shows artificially shaded preview, right column shows resulting mask (repeated in both rows).

Much research on cloud detection in particular focuses on spectral imagery as input data, rather than DEM input. Cloud detection methods for these data are based on cloud optical properties and may detect cloud/no-cloud, cloud/snow, and/or thin/thick cloud regions of an image [11]. Fmask [23] is a popular algorithm for detecting both clouds and cloud shadows in spectral imagery. A recent paper by Wu, et al [18] uses DEM data, but for validation of their spectral cloud-finding results, rather than for the detection directly.

The method described in this paper uses deep learning image segmentation to detect and mask out cloud regions. This is based on the popular U-Net algorithm [14], initially developed for medical image segmentation but which has since been adopted for use in other fields that require classifying image pixels. The RS-Net [10] and MC-Net [20] methods also use U-Net for cloud detection, but once again on spectral imagery rather than DEM data. Other notable recent machine learning image segmentation papers based on U-Net include a method for identifying vortex boundaries in scientific visualizations [1] and a method for removing clouds in 3-channel RGB spectral imagery with generative adversarial networks [22].

3 Method

3.1 Ground Truth Mask Creation

The labelled dataset used as the ground truth in training was created as a byproduct of the work toward the documentary *Atlas of a Changing Earth*, co-produced by Thomas Lucas Productions and the Advanced Visualization Lab at the National Center for Supercomputing Applications. The artifacts were masked and removed manually in order to fit the timeline of the film production, and these resulting masks served a secondary purpose as the inputs to our machine learning model.

The first step in acquiring the data was identifying an area of interest and downloading a subset of the data at a suitable resolution. A 3473x2840 pixel region was initially selected around the Jakobshavn glacier, a 110,000-square km glacier in Greenland, and serves as our dataset. GEOTIFF images were downloaded from the ArcticDEM website and aligned using the georeferenced imagery, so that each new data strip would be in the correct pixel location within our selected region of interest. Several derivative versions of the data were created: (1) images that show one strip at a time and leave the rest of the frame blank; (2) images that are an accumulation of strips up until the current timestep; (3) images where each pixel corresponds to the time that an accumulated pixel was added; and (4) images that are artificially-shaded using gdaldem's "hillshade" mode[1] for easier visual inspection; among others.

A multimedia specialist on the team used the software Nuke[2] to visually inspect the individual DEM strips, comparing them with strips gathered immediately before and after to identify and manually mask out areas that appeared to be artifact-ridden. Using a visual effects technique called rotoscoping, in which a vector mask is created in one image frame and filled in with imagery from another, the expert drew the masks for each new data strip by comparing the various images described above over time, interactively making adjustments to image intensity as needed for better visual acuity. Figure 3 shows a sample of types of inputs into this manual process as well as the output mask for a single timestep.

The hand-drawn masks were not pixel-precise, but were overdrawn for reasons of convenience - e.g. if 90% of a strip was cloud-covered, it was more time-efficient to mask out the whole strip rather than finding the individual pixels that were valid. This was satisfactory for purposes of the documentary, but would not be suitable for a machine learning task. We therefore created a second set of "motion masks" where each pixel contained a 1 only if the pixel had been updated (moved) in that current timestep, and 0 otherwise, based on derivative data version (3) described above. Multiplying these two masks together clipped the expert-created overdrawn masks to only pixels that were present in the strip at that timestep. The resulting masks are both expert-driven and pixel-precise.

3.2 Data Pre-Processing

Data must be processed prior to being used for training in order to optimize training time and results. First, each image and its corresponding ground-truth mask is subdivided into patches of size 224x224 pixels. This size was chosen in order to divide cleanly into whole numbers when downsampled with the U-Net algorithm. Other patch sizes were tested parameter tuning, ranging from roughly 100x200 - 600x600, and this size was chosen for having a good ratio of processing speed to manageable number of output images. Patches were set to overlap one another by 50 pixels to account for artifacts around the borders of the image, which are known to occur with many Convolutional Neural Network-based image processing algorithms [10]. This also had the result of creating more training data with different patch croppings. The value of 50 pixels was selected by visually inspecting a sampling of predicted output masks and determining the region of consistently-inaccurate predictions around the borders. Because clouds are more rare than non-clouds in the data and they are the subject of interest, only the patches that had at least one pixel of cloud (as determined by the ground-truth mask) were saved. There were originally 978 images of size 3473x2840, which were converted into 4399 patches of size 224x224. Scripts were developed for splitting the full-sized image into patches and for reassembling the patches into a full-size image.

Our initial machine learning model used these images as training data, but produced poor results where many discontinuous, individual pixels were identified as clouds rather than broad, connected areas. To resolve this issue, an additional second order textural analysis pre-processing step was added to create derivative data that considers the spatial relationship among the image pixels. A Gray Level Co-occurrence Matrix (GLCM) [9] is an image representation which keeps track of different combinations of pixel values (gray levels) as they occur in an image, identifying various image texture features such as contrast, dissimilarity, homogeneity, and entropy. Figure 4 shows three of these features over different types of land covers. Calculating the GLCM requires specifying two parameters - the window size to use around each pixel, and the relationship direction, which is the distance vector between the reference pixel and the neighborhood pixel (often taken as a single unit distance in each of the 4 directions left, right, up, and down). In order to consider both small-scale and large-scale texture features, 3-, 5-, and 15-pixel window sizes were used to create three derivative datasets, to be used in an ensemble method of cloud mask prediction. Each of these datasets consisted of 4399 52-channel textural "images". After the GLCM calculations, the images were normalized to be between 0-1, as a best practice for machine learning.

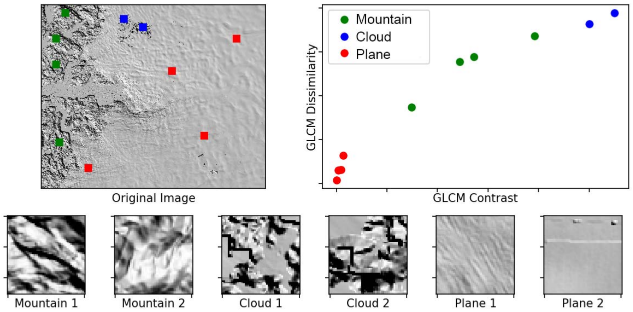

Figure 4: GLCM features for three main types of land covers.

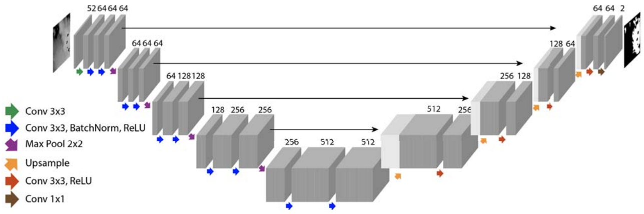

Figure 5: The CloudFindr architecture, based on U-Net [14].

3.3 Deep Learning for Cloud Prediction

U-Net was selected as the basis for CloudFindr. Other architectures were considered - notably RS-Net [10] and MC-Net [20] - which are specialized use cases of the more basic underlying U-Net algorithm and are optimized for different use cases: RS-Net for spectral and MC-Net for multi-channel satellite imagery. U-Net was chosen as it is more generalized and allows for customization at a lower level. The CloudFindr architecture is outlined in Figure 5. The downstream branch consists of four convolutional blocks, each being a combination of two convolution and ReLU operations, followed by a maxpool to reduce the dimensions of the image by a factor of two (with stride 2 and kernel size 2). At the end of the downstream branch, the input is reduced to a size of width/16 by height/16 by 512 features. The upstream branch consists of four upsampling convolutional blocks. Each block first upsamples the input by a factor of two using up-convolution followed by a ReLU operation, increasing the size of the input again by a factor of 16. A final convolutional layer is applied to convert the resulting 16 channels into 2, followed by a softmax to obtain a probability for each class, "cloud" versus "non-cloud". The resulting image contains a pixel-wise confidence between 0-1 for whether that pixel contains a cloud or not. This image is thresholded to produce discrete 0 or 1 values in the final output mask to give a prediction of "cloud" or "no cloud".

The dataset has a 60-20-20 split between training-validation-testing. The hyperparameters of loss function, optimizer, learning rate, regulation, and number of epochs were tuned via control experiments. A combined evaluation of IoUs and segmentation results was performed after each experiment to determine if current variable value would be retained for next experiments. The optimal combination of parameters is found as: loss function weights = [0.3,0.7] to account for the imbalance between number of instances for each

[1] https://gdal.org/programs/gdaldem.html

[2] https://www.foundry.com/products/nuke

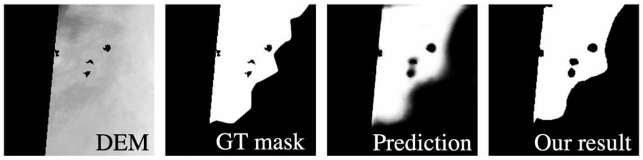

Figure 6: One example patch where it would be difficult for a casual observer to identify clouds, but the expert and machine learning prediction have closely-aligned results. From left to right: Input DEM patch, ground truth mask hand-drawn by an expert, confidence of prediction after ensemble voting, final thresholded predicted mask.

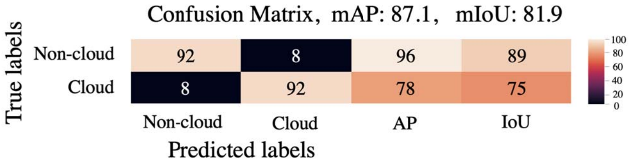

Figure 7: Confusion matrix showing the success of the predictions after all processing.

class, Adam optimizer with learning rate of 0.005, no dropout regulation, and 200 epochs. Both Adam and SGD optimizers were tested with learning rates between 0.005 and 0.001. The best results came from the use of Adam with a learning rate of 0.005.

Initially, the model was run on derivative datasets with GLCM window sizes of 3, 5, and 15 with the aim of finding a single optimal window size. As designed, all resulting predictions skewed toward higher recall rather than higher precision and tended to over-label areas as “clouds” rather than under-labelling them. However by visually analyzing the output masks, it became clear that the three methods tended to agree with one another about the areas *correctly* identified as clouds, but disagreed about the areas labelled *incorrectly*. This inspired the use of an ensemble method for gathering the final result. The final prediction combines results from all three runs by multiplying the outputs together. The effect of this is that the overall confidence value is significantly reduced, but if any one of the runs predicts a 0 value (predicting that there are no clouds present), this overrides any other predictions and a 0 value is placed in the final output mask. The multiplied confidence is thresholded with a value of 0.1 to create the final binary cloud/non-cloud prediction. Figure 6 shows one example patch prediction.

When a cloud is mislabelled as a non-cloud, this most often appears around the perimeter of a correctly-labelled cloudy area. To account for this, a final post-processing step is applied to dilate the image masks with a kernel of size (5,5). This reduces the error around the edges of cloud regions, and creates masks that are slightly “overdrawn” similarly to how the human expert performed manual rotoscope labelling.

4 Results

The neural network was trained on a GM200GL Quadro M6000 NVIDIA GPU for approximately 12 hours. In the final result, the model was able to correctly identify cloudy DEM pixels 92% of the time. The mean average precision of the optimal model described above is 87.1% and the mean IoU is 81.9%, with a further breakdown for each class shown in Figure 7.

The output of the described algorithm is 4399 patches of size 224x224 with values of 1 where there are likely clouds present, and 0 where there are not. These patches are stitched back together to create 978 masks of size 3473x2840 which can be multiplied against the 978 DEMs of size 3473x2840 around the Jakobshavn area. The DEM strips and masks are then accumulated to create the final DEMs to be used in the 3D cinematic rendering. Figure 8 shows how our result compares to the ground truth in final 3D rendered imagery, as well as what the render looks like without cloud removal. These renderings are created with the software Houdini[3], where the DEM values are used to drive both the height and the color of the land. In this figure, the vast majority of the cloud artifacts have been removed, and the ones that have been missed are not as visually disturbing as the more prominent spikes.

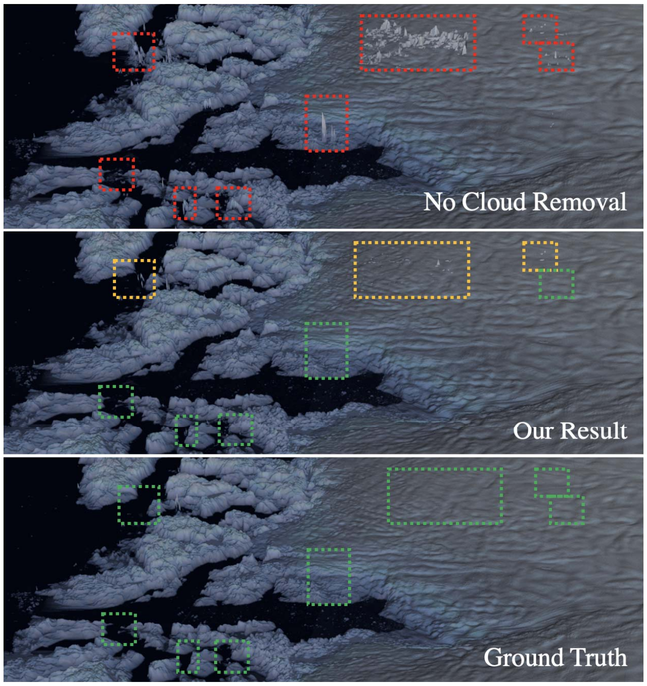

Figure 8: Images showing the same single frame of a final 3D render. Top: using no cloud mask. Middle: using cloud mask created via the method described here. Bottom: using masks created manually by a rotoscoping expert. Red boxes draw attention to areas with especially visible clouds; yellow boxes show that the clouds have been mostly removed; green boxes show that they have been entirely removed.

5 Conclusion and Future Work

In this paper, we describe CloudFindr, a method of labelling pixels as “cloud” or “non-cloud” from a single-channel DEM image. We first extract textural features from the image with varying window sizes. We feed this derived data into a U-Net based model, trained on labelled data created by an expert, to create image segmentation predictions. The results have high accuracy as demonstrated both by metrics and by a 3D rendering created from the data.

In the future, we will plan a large hyperparameter tuning study including features at different sizes, learning rate, momentum, and batch size to optimize our results. Additionally, we would like to apply this method to other DEM datasets outside the Jakobshavn region of the ArcticDEM dataset, and also incorporate the time dimension into the training to differentiate between strips that are updating a previously-seen area from strips covering a new region.

Acknowledgments

Thank you to Donna Cox, Bob Patterson, AJ Christensen, Saurabh Gupta, Sebastian Frith, and the reviewers. This work was supported by the Blue Waters Project, National Science Foundation, National Geospatial-Intelligence Agency, and Fiddler Endowment.

[3] https://www.sidefx.com/products/houdini/

References

[1] M. Berenjkoub, G. Chen, and T. Günther. Vortex boundary identification using convolutional neural network. 2020.

[2] N. Cao, C. Lin, Q. Zhu, Y. Lin, X. Teng, and X. Wen. Voila: Visual anomaly detection and monitoring with streaming spatiotemporal data. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):23–33, 2018.

[3] N. Cawthon and A. V. Moere. The effect of aesthetic on the usability of data visualization. In *2007 11th International Conference Information Visualization (IV '07)*, pp. 637–648, 2007. doi: 10.1109/IV.2007.147

[4] R. Chalapathy and S. Chawla. Deep learning for anomaly detection: A survey. *CoRR*, abs/1901.03407, 2019.

[5] N. Champion. Automatic Cloud Detection from Multi-Temporal Satellite Images: Towards the Use of PLÉIADES Time Series. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39B3:559–564, Aug. 2012. doi: 10.5194/isprsarchives-XXXIX-B3-559-2012

[6] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009. doi: 10.1145/1541880.1541882

[7] M. Derrien, B. Farki, L. Harang, H. LeGléau, A. Noyalet, D. Pochic, and A. Sairouni. Automatic cloud detection applied to noaa-11 /avhrr imagery. *Remote Sensing of Environment*, 46(3):246 – 267, 1993. doi: 10.1016/0034-4257(93)90046-Z

[8] A. Goodman and A. Henderson-Sellers. Cloud detection and analysis: A review of recent progress. *Atmospheric Research*, 21(3):203 – 228, 1988. Clouds and Climate. doi: 10.1016/0169-8095(88)90027-0

[9] R. M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3(6):610–621, 1973. doi: 10.1109/TSMC.1973.4309314

[10] J. H. Jeppesen, R. H. Jacobsen, F. Inceoglu, and T. S. Toftegaard. A cloud detection algorithm for satellite imagery based on deep learning. *Remote Sensing of Environment*, 229:247 – 259, 2019. doi: 10.1016/j.rse.2019.03.039

[11] S. Mahajan and B. Fataniya. Cloud detection methodologies: variants and development—a review. *Complex & Intelligent Systems*, 6:251,261, 2020. doi: 10.1007/s40747-019-00128-0

[12] P. Morin, C. Porter, M. Cloutier, I. Howat, M.-J. Noh, M. Willis, B. Bates, C. Willamson, and K. Peterman. ArcticDEM; A Publically Available, High Resolution Elevation Model of the Arctic. In *EGU General Assembly Conference Abstracts*, EGU General Assembly Conference Abstracts, pp. EPSC2016–8396, Apr. 2016.

[13] C. Porter, P. Morin, I. Howat, M.-J. Noh, B. Bates, K. Peterman, S. Keesey, M. Schlenk, J. Gardiner, K. Tomko, M. Willis, C. Kelleher, M. Cloutier, E. Husby, S. Foga, H. Nakamura, M. Platson, J. Wethington, Michael, C. Williamson, G. Bauer, J. Enos, G. Arnold, W. Kramer, P. Becker, A. Doshi, C. D'Souza, P. Cummens, F. Laurier, and M. Bojesen. Arcticdem, 2018. doi: 10.7910/DVN/OHHUKH

[14] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, eds., *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241. Springer International Publishing, Cham, 2015.

[15] M. Sener, S. Levy, J. E. Stone, A. Christensen, B. Isralewitz, R. Patterson, K. Borkiewicz, J. Carpenter, C. N. Hunter, Z. Luthey-Schulten, and D. Cox. Multiscale modeling and cinematic visualization of photosynthetic energy conversion processes from electronic to cell scales. *Parallel Computing*, 102:102698, 2021. doi: 10.1016/j.parco.2020.102698

[16] D. Southwell, E. Thorson, and L. Sheble. *Misinformation and Mass Audiences*. University of Texas Press, 2018.

[17] H. Tang, K. Yu, O. Hagolle, K. Jiang, X. Geng, and Y. Zhao. A cloud detection method based on a time series of modis surface reflectance images. *International Journal of Digital Earth*, 6(sup1):157–171, 2013. doi: 10.1080/17538947.2013.833313

[18] T. Wu, X. Hu, Y. Zhang, L. Zhang, P. Tao, and L. Lu. Automatic cloud detection for high resolution satellite stereo images and its application in terrain extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 121:143 – 156, 2016. doi: 10.1016/j.isprsjprs.2016.09.006

[19] Xiaoning Song, Zhenhua Liu, and Yingshi Zhao. Cloud detection and analysis of modis image. In *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, vol. 4, pp. 2764–2767 vol.4, 2004. doi: 10.1109/IGARSS.2004.1369875

[20] Z. Yao, J. Jia, and Y. Qian. Mcnet: Multi-scale feature extraction and content-aware reassembly cloud detection model for remote sensing images. *Symmetry*, 13(1), 2021. doi: 10.3390/sym13010028

[21] Y.-C. Zhang, W. B. Rossow, A. A. Lacis, V. Oinas, and M. I. Mishchenko. Calculation of radiative fluxes from the surface to top of atmosphere based on isccp and other global data sets: Refinements of the radiative transfer model and the input data. *J. Geophys. Res.*, 109:D19105, 2004. doi: 10.1029/2003JD004457

[22] J. Zheng, X.-Y. Liu, and X. Wang. Single image cloud removal using u-net and generative adversarial networks. *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–15, 2020. doi: 10.1109/TGRS.2020.3027819

[23] Z. Zhu and C. E. Woodcock. Object-based cloud and cloud shadow detection in landsat imagery. *Remote Sensing of Environment*, 118:83 – 94, 2012. doi: 10.1016/j.rse.2011.10.028