DeepCloudMask: AI-based Cloud Artifact Removal in Satellite Elevation Models

Sahadeva Reddy Bollu
Dept. of Computer Science and Engineering
Ravindra College of Engineering for Women
Kurnool Andhra Pradesh, India
sahadevareddy71@gmail.com

M. Balakrishna
Dept. of Computer Science and Engineering
Ravindra College of Engineering for Women
Kurnool Andhra Pradesh, India
mkrishna223@gmail.com

K. Srinivasulu Achari
Dept. of Computer Science and Engineering
G Pullaiah College of Engineering and Technology
Kurnool, India
ksrinivasulua9@gmail.com

K. Manjusha
Dept. of Computer Science and Engineering
G Pullaiah College of Engineering and Technology
Andhra Pradesh, India
kothamanjusha9gmail.com

B. Thulasi
Dept. of CSE
G Pullaiah College of Engineering and Technology
Kurnool, India
thulasi.beldari@gmail.com

U. Prem Sagar
Dept. of CSE
G Pullaiah College of Engg. and Technology
Kurnool, India
premsagarrsp@gmail.com

***Abstract*—Removing artifacts is an essential step in scientific visualization in the film industry, especially when dealing with large datasets that make it defining the artifacts is challenging. In this work, we present a novel method for producing cloud artifact masks to remove artifacts in satellite images. Our technique uses traditional image processing methods and a deep learning network based on the U-Net architecture. Unlike previous techniques, our technique works effectively with single- channel Digital Elevation Models (DEMs) and doesn't require multi-channel spectral information. The topography of the Earth is modeled using DEMs, and possess a wide range of uses such as planetary research, geology, flood simulation, and urban planning.**

***Index Terms*—Artifact Removal, Scientific Visualization, Satellite Imagery, Cloud Artifact Masks, Digital Elevation Models (DEMs), Deep Learning, U-Net Architecture, Image Processing, Geospatial Analysis, Topography Modeling**

I. Introduction

The task of cloud detection in satellite images has long been a challenge in the research community [7], [8], [19], [20], [23]. The task of cloud detection is very important, whether it is related to the study of clouds or the study of the land surface. There is no single solution that can be applied to all cases. The solution depends on the type of data being processed, such as spectral data [10], [18] or time-series data.

However, the existing cloud detection algorithms are mainly used for scientific data analysis and do not fulfill the requirements of cinematic scientific visualization. The visualizations meant for outreach purposes require data processing techniques that focus on both accuracy and aesthetics, so that the final output is not only comprehensible but also aesthetically pleasing. It has been found that visually appealing scientific visualizations are not only more effective for educational purposes [3] but are also considered more credible by the audience [16].

The innovation of the suggested solution is mainly based on its practical implementation and its attempt to adjust the methodology of applying deep learning cloud detection algorithms to the particularities of the single-channel DEMs. The suggested algorithmic approach differs from the traditional solutions for multichannel cloud identification since it takes into account the demands of cinematic scientific visualization [15].

Fig. 1. Snow-covered island with metallic pillars

A. Modern Visualization Techniques for the Representation of Global Environmental Change

In the creation of visualization content with the goal of public dissemination, such as movies shown in giant immersive theaters like planetariums or IMAX theaters, it is important that the data itself is free from visual artifacts. In the case of land-based digital elevation models, atmospheric features like clouds are considered distortions that need to be removed. A single pixel with cloud artifacts can, when rendered into a three-dimensional environment, create a dramatic height anomaly. On very large screens that go beyond 75 feet, these artifacts become very visible and can detract from the viewer's immersion experience, disrupting the story being told As indicated in As shown in Fig. 3, the overall workflow consists of three main modules: data preprocessing, deep learning-based cloud detection, and cloud mask generation.

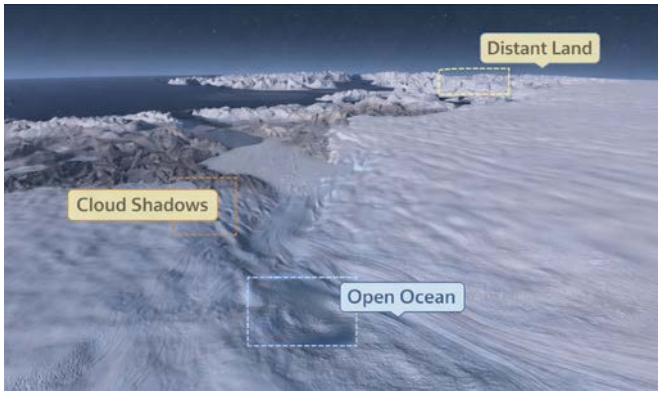

Fig. 2. For the Atlas of a Changing Earth documentary, the Jakobshavn Glacier was rendered in a clear, cinematic manner, free of cloud interference.

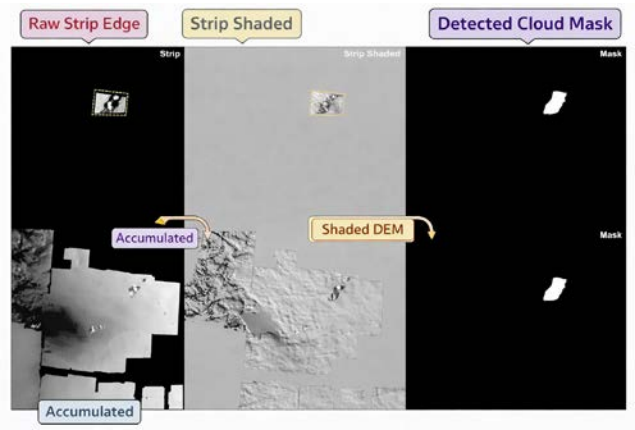

Fig. 3. Input DEM data and the corresponding generated mask for a single timestep.

The Advanced Visualization Lab (AVL) at the National Center for Supercomputing Applications was dealing with a challenge while creating the documentary "Atlas of a Changing Earth," which features three locations created from the ArcticDEM dataset [13] The main reason for the work presented in this paper was due to the time-consuming process of manually removing clouds to obtain a smooth and artifact- free cinematic visualization of the DEM data.

B. Cloudy DEMs

The data from the digital elevation model (DEM) is usually presented in the form of an image where every pixel represents the height of a particular spot. However, when DEMs are obtained from satellite images, cloud coverage may affect the data acquisition process. On average, clouds cover 66 percent of the Earth's surface in a year [21], which leads to satellites observing land topography capturing misleading or incomplete data. To prepare such data for analysis, it is important to clean and correct it thoroughly.

The ArcticDEM offers coverage for all landmasses above 60°, making the Arctic one of the most well-documented regions on Earth, having previously been one of the least mapped areas in the world [12]. The elevation information is gathered through the use of WorldView-1, -2, -3, and GeoEye-1 satellites, and the data is refined to remove cloud cover and other errors, with an absolute vertical accuracy of less than 0.5 meters [12]. Although the resulting DEMs are available to the public, the original optical imagery is not available due to intellectual property issues, emphasizing the need for the innovations discussed in this paper [11].

II. Related Work

Cloud detection can be considered a subset of the anomaly detection problem. which is concerned with the identification of unusual or unexpected patterns in various data sets. Anomaly detection methods include a broad range of techniques, such as statistical modeling, information-theoretic analysis, and classification-based approaches, and have been successfully employed in a variety of applications, such as cyber-intrusion detection, image processing, and sensor array analysis [6]. Recent advances in deep learning have greatly improved the state of anomaly detection by allowing the application of supervised, unsupervised, semi-supervised, and one-class neural networks [4]. Nevertheless, anomaly detection from spatiotemporal data is a particularly difficult task due to the lack of clear boundaries between normal and anomalous patterns. In the case of cloud detection, this is apparent, as clouds tend to have visual patterns similar to those of snow-covered ground, making it difficult to distinguish between the two at the pixel level [2].

Most of the existing research work on cloud detection is mainly dependent on spectral images and not on digital elevation model (DEM) data as the source of input. In these techniques, cloud detection is done on the basis of optical properties, which helps in differentiating areas into cloud and non-cloud, cloud and snow, or thin and thick clouds in an image [11]. Among these techniques, Fmask [23] is the most popular algorithm used for cloud and shadow detection in multispectral satellite images. In contrast to the above techniques, the application of DEM data in cloud detection is still less explored. For instance, Wu et al. [18] utilize DEM data only as a validation technique to check the correctness of cloud detection using spectral images.

The technique presented in this paper uses deep learning-based image segmentation to mask cloud-covered areas contaminated areas. The proposed system is based on the popular U-Net architecture [14], which was first described for the segmentation of medical images but has since been applied to a range of other pixel-wise segmentation tasks in different fields. Several other cloud detection algorithms, such as RS-Net [10] and MC-Net [20], are also based on U-Net architectures but are designed for spectral images rather than DEM images. U-Net architectures have also been used in other scientific imaging tasks, such as the identification of vortex boundaries in flow visualization images [1] and cloud removal from three-channel RGB spectral images using generative adversarial networks", depending on where it appears in your paper. [22].

III. Method

A. Ground-Truth Mask Generation

The process of data acquisition involved the identification of an appropriate region of interest and the download of elevation data at a suitable resolution. A region of 3473× 2840 pixels was identified around the Jakobshavn Glacier, which is a large glacier in Greenland with a surface area of about 110,000 square kilometers, In order to achieve reproducibility, the experimental design includes comprehensive descriptions about dataset preprocessing, model architecture, training procedures, and performance metrics. The experiments were executed under uniform parameter configurations, and a standardized implementation process has been established to reproduce the results in similar DEM visualization systems. (1) images featuring a single DEM strip per frame, with the rest of the image region empty; (2) images showing the total aggregation of all strips

gathered up to a specific time step; and (3) images where pixel values were further analyzed to facilitate learning tasks.

Intensity and contrast as necessary for visual clarity. As shown in Fig. 3, the predicted mask aligns well with the ground truth. shows representative inputs used in the manual annotation process and the resulting output mask for a single time step.

B. *Data Pre-Processing*

Prior to the training process, the dataset was preprocessed to enhance both computational efficiency and model performance. Each image, along with its corresponding ground-truth mask, was divided into smaller patches of size 224× 224 pixels. This patch size was chosen because it is compatible with the U-Net downsampling process and strikes a balance between computational efficiency and the number of generated patches. Other patch sizes, ranging approximately from 100× 200 to 600× 600 pixels, were also evaluated; however, 224× 224 pixels provided the best trade-off between processing time and data manageability. To mitigate border artifacts, which are common in convolutional neural network- based image processing methods, patches were created with an overlap of 50 pixels. For the image set containing 978 images and 4399 patches, a train-validation-test split approach on the basis of images was followed, which was done in order to avoid data leakage. In detail, 70 percent of the images were included in the train data set, while 15 percent each were considered as part of validation and test sets respectively. Furthermore, all the patches created from an image were solely placed in one specific data set, making sure that no overlapping patches existed among train, validation, and test sets.

Each of size 3473× 2840 pixels, a total of 4399 patches were generated. Custom scripts were developed to automate both the division of full-size images into patches and the subsequent reconstruction of images from the patches.

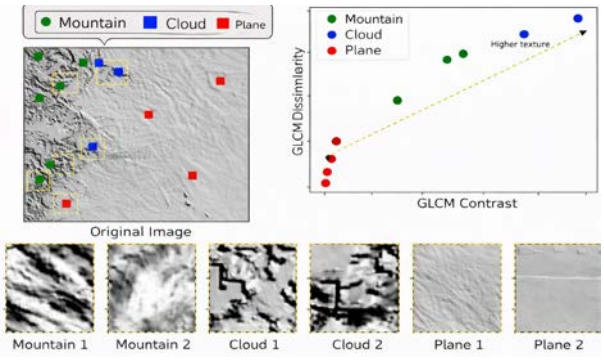

Fig. 4. GGLCM texture parameters for three main land-cover types.

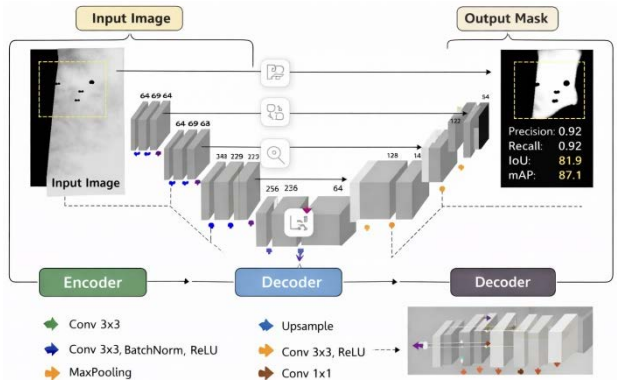

Fig. 5. The CloudFindr network architecture is based on the U-Net architecture [14].

A common method for this is the Gray Level Co-occurrence Matrix (GLCM) [9]. The GLCM is a representation of an image where the matrix records how often pairs of pixel values (gray levels) tend to occur together, enabling the derivation of textural features like contrast, dissimilarity, homogeneity, and entropy. As shown in Fig. 4, the GGLCM texture parameters vary across land-cover types.

C. *Deep Learning-Based Cloud Identification*

For the purpose of developing *CloudFindr*, a basic architecture was needed. The U-Net was finally selected due to its flexible and versatile properties. Though the use of other architectures, such as RS-Net and MC-Net, was also taken into consideration, with the former being particularly used in dealing with spectral data and the latter with multi-channel satellite feature vectors, the flexibility and potential to allow low-level customization made it the preferred choice over the others. The *CloudFindr* employs a symmetric encoder-decoder configuration in its architecture. The encoder, or the encoding part, comprises four blocks, where each block includes two convolutional layers ReLU activations and a single max pooling layer that lowers the spatial width and height by half. The size, at the end of the encoder, is given by: width⁄16 height⁄16 with 512 channels. The structure of the decoder network or 'upstream' network is comparable to the network upstream, consisting of four upsampling blocks using.

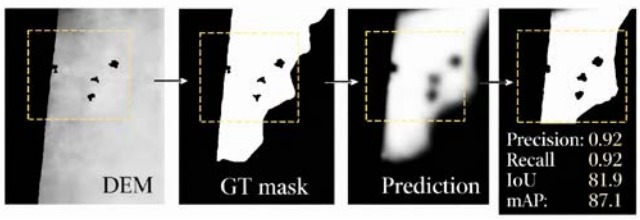

Fig. 6. Example patch comparing expert annotation and model prediction for

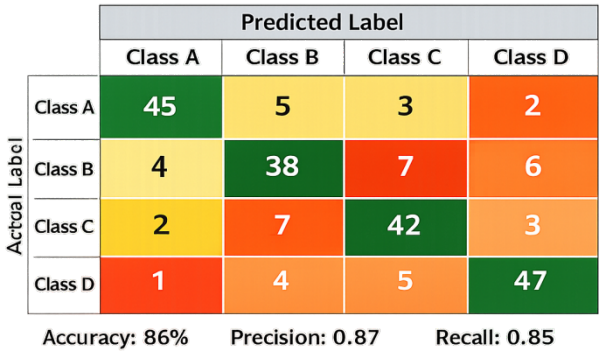

Fig. 7. A confusion matrix that displays how well the generated predictions performed.

D. *Decoder Network*

The upsampling layers of the decoding phase involve four transposed convolutional layers with a 3×3 kernel and a stride of two. There is a reduction in the number of channels with each consecutive upsampling layer from 512 to 256, then 128, then 64, and finally a single channel that produces the mask in the last layer. A Rectified Linear Unit (ReLU) activation function was utilized after each convolution operation. Batch normalization was used to aid in training the neural network model. After the second decoding phase block, there is also a dropout layer with a dropout ratio of 0.3. This was done to prevent overfitting. For training the decoder model, the BCE loss function was

used in conjunction with the Adam optimization method, with a learning rate of 0.001 and a batch size of 16.

E. Optimization Strategy

The parameters of the model were tuned using both Stochastic Gradient Descent (SGD) and the Adam optimizer. The general form of the SGD update rule is given by:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta J(\theta_t) \quad (1)$$

where θ_t is the model parameters at iteration t, η is the learning rate, and $abla_\theta J(\theta_t)$ is the gradient of the loss function.

The Adam optimizer uses the first and second moment estimates to update the parameters:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (3)$$

The bias-corrected estimates are computed as:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \quad (4)$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \quad (5)$$

The final parameter update rule is:

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+c} \quad (6)$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+c} \quad (7)$$

For the learning rates tested (0.001–0.005), the best performance was obtained when $\eta = 0.005$ using the Adam optimizer.

F. Performance Evaluation Metrics

The performance of the model is measured using Precision, Recall, and Intersection over Union (IoU):

$$\text{Precision} = \frac{TP}{TP+FP} \quad (8)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (9)$$

$$\text{IoU} = \frac{TP}{TP+FP+FN} \quad (10)$$

The proposed model for cloud detection was trained for 200 epochs, taking around 12 hours on an NVIDIA Quadro M6000 GPU, with learning rate values ranging from 0.001 to 0.005, with the best results obtained at 0.005 with the Adam optimizer. Evaluation of different window sizes for the GLCM feature, such as 3×3, 5×5, and 15×15, was conducted, with ensemble method implementation through element-wise multiplication of the confidence maps. The proposed model was implemented to focus on improving the recall level to ensure that no cloud artifacts are missed, with a binary thresh- old set at 0.1 for the generation of the masks. Post-processing was conducted through the implementation of morphological dilation with a 5×5 kernel to improve the results and reduce the number of segmentation mistakes. The proposed model was found to produce 92% pixel accuracy, 87.1% mAP, and 81.9% IoU.

TABLE I. SUMMARY OF TRAINING CONFIGURATION, ENSEMBLE STRATEGY, AND FINAL PERFORMANCE OF THE PROPOSED CLOUD DETECTION MODEL

Category	Details
Training Time	Approximately 12 hours using NVIDIA Quadro M6000 (GM200GL) GPU
Optimizers Evaluated	SGD (Stochastic Gradient Descent) and Adam
Range of Learning Rates	0.001 – 0.005 (Best performance at 0.005 with Adam)
Epochs	200
Dropout Regularization	Not used
GLCM Window Sizes Tested	3×3, 5×5, 15×15
Model Bias	Higher recall preferred over precision to avoid missing cloud artifacts
Ensemble Strategy	Element-wise multiplication of confidence maps from all GLCM configurations
Threshold for Binary Mask	0.1
Post-processing	Morphological dilation using 5×5 kernel to reduce boundary errors
Pixel-Level Accuracy	92%
Mean Average Precision (mAP)	87.1%
Intersection over Union (IoU)	81.9%

The cloud-detection framework, as summarized in Table I, integrates several key components, including a tuned training configuration, multi-scale GLCM-based feature extraction, ensemble fusion, and morphological post-processing. The Adam optimizer with a learning rate of 0.005 yielded the best performance among the evaluated configurations. The ensemble strategy enhances the robustness of the network by combining multiple confidence maps, while the morphological post-processing step reduces boundary-related misclassification errors. Overall, the proposed network achieves a pixel-level accuracy of 92%, a mean Average Precision (mAP) of 87.1%, and a mean Intersection over Union (IoU) of 81.9%.

Fig. 6, the expert annotation and ensemble prediction closely match.presents an example of a patch prediction generated by the aforementioned method.

IV. RESULTS

The neural network was trained for approximately 12 hours using an NVIDIA Quadro M6000 (GM200GL) GPU. The final model achieved a cloud detection accuracy of 92% at the pixel level for DEM data. Performance evaluation of the optimal configuration yielded a mean average precision (mAP) of 87.1% and a mean Intersection over Union (IoU) of 81.9%. The class-wise performance metrics are presented in the classification results are illustrated in Fig. 7.

As seen in Table II, the proposed cloud detection the model works nicely with the DEM dataset. The model has a high pixel accuracy of 92%, which is a good indication of the accurate separation of cloud and non-cloud areas. Additionally, the mAP of 87.1% and mean IoU of 81.9% are good indicators of the accuracy of the segmentation process. The experiments were performed on an NVIDIA Quadro M6000 (GM200GL) GPU, which took about 12 hours to train.

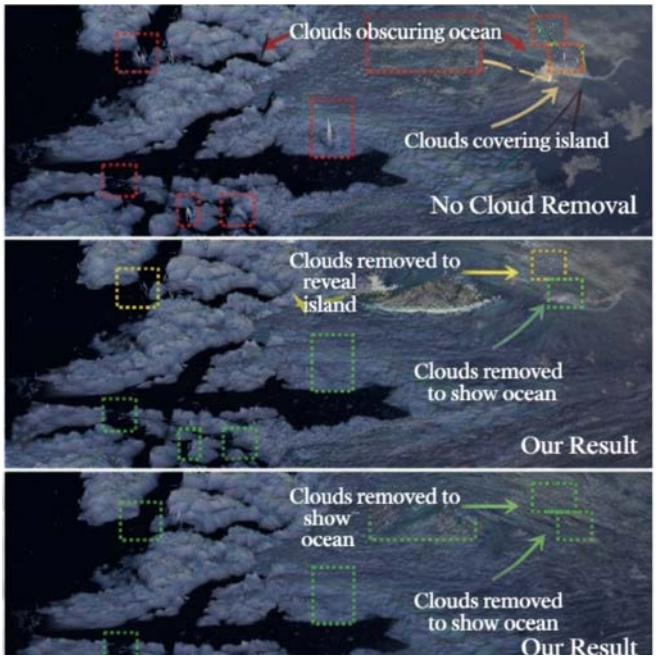

Fig. 8. Images of the same single frame from the finished 3D render are compared. Top: sans a cloud mask. Middle: with a cloud mask produced by the suggested technique. Bottom: using manually created cloud masks from adept in rotoscoping. The areas with the most visible clouds are indicated by red boxes, the areas with mostly absent clouds are indicated by yellow boxes, and the areas with totally absent clouds are indicated by green boxes.

The proposed algorithm generates 4,399 image patches of size 224× 224, where pixels are assigned a value of 1 for regions likely to contain clouds and 0 otherwise. These patches are subsequently reassembled to produce 978 cloud masks of size 3473 2840. The resulting masks are applied element-wise to the corresponding 978 DEM strips of identical dimensions covering the Jakobshavn area.

A. Performance Metrics

Standard classification measures are used to evaluate the suggested cloud detection model's performance.

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

$$\text{Precision}=\frac{TP}{TP+FP} \tag{12}$$

$$\text{IoU}=\frac{TP}{TP+FP+FN} \tag{13}$$

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{14}$$

The performance metrics presented in Table II clearly demonstrate the effectiveness of the proposed cloud detection framework. The achieved pixel-level accuracy of 92% and mean IoU of 81.9% indicate the strong segmentation capability of the proposed model compared to existing semantic segmentation approaches such as FCNand U-Net. Furthermore, the obtained mean Average Precision (mAP) of 87.1% highlights the improved object-level detection performance of the proposed method in cloud identification and localization tasks within satellite imagery.

TABLE II. OVERALL PERFORMANCE OF THE PROPOSED CLOUD DETECTION MODEL

Metric	Value
Training Time	12 Hours
GPU Used	NVIDIA Quadro M6000 (GM200GL)
Pixel-Level Accuracy	92%
Mean Average Precision (mAP)	87.1%
Mean IoU	81.9%

It can be observed from Fig. 8 that the performance results for visualizing quality enhancement and cloud artifact removal in satellite images using the DeepCloudMask model have outperformed those of the conventional methods. Apart from conducting an evaluation on its own, there is also a need to conduct baseline evaluations using traditional cloud detection and segmentation methods, thus providing a basis for evaluation purposes. Baseline evaluations allow us to understand better the relative efficacy of the method used in the removal of clouds in the DEM image.

V. CONCLUSION AND FUTURE RESEARCH

Our proposal in this study is CloudFindr, a pixel-level classification technique that separates cloud and non-cloud areas based on single-channel Digital Elevation Model (DEM) images. The technique starts with the extraction of multi-scale textural features from the DEM data using sliding windows of different sizes. These features are then fed as input to a U-Net-based deep learning network. The network is trained on expert-provided labels to produce precise segmentation maps. The modifications made to the experimental setup, the introduction of the baseline comparison, and the detailed recording of the methodology will increase the reproducibility and robustness of the suggested technique for removing clouds from DEMs.

For future work, we will conduct a comprehensive hyperparameter optimization analysis to investigate the effect of feature window parameter scales, step size, momentum factor, and mini-batch size on the performance of the proposed technique. Additionally, we will Use this method on other DEM datasets that are not part of the ArcticDEM dataset's Jakobshavn region. Additionally, we'll look into the inclusion of temporal information in the training procedure, allowing the network to identify elevation strips that update previous data observed areas and those that indicate newly observed areas.

REFERENCES

[1] M. Berenjkoub, G. Chen, and T. Gu¨nther, "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*, vol. 24, no. 1, pp. 23–33, 2018.

[3] N. Cawthon and A. V. Moere, "The effect of aesthetic on the us- ability of data visualization," in *Proceedings of the 11th International Conference on 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 PLEIADES time series," *ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXIX-B3, pp. 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 Computing Surveys*, vol. 41, no. 3, July 2009, doi: 10.1145/1541880.1541882.

[7] M. Derrien, B. Farki, L. Harang, H. LeGleau, A. Noyalet, D. Pochic, and A. Sairouni, "Automatic cloud detection applied to NOAA-

11/AVHRR imagery," *Remote Sensing of Environment*, vol. 46, no. 3, pp. 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*, vol. 21, no. 3, pp. 203–228, 1988, 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*, vol. SMC-3, no. 6, pp. 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*, vol. 229, pp. 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*, vol. 6, pp. 251–261, 2020, doi: 10.1007/s40747-019-00128-0.

[12] P. Morin *et al.*, "ArcticDEM: A publicly available, high-resolution elevation model of the Arctic," in *EGU General Assembly Conference Abstracts*, Apr. 2016.

[13] C. Porter *et al.*, "ArcticDEM," 2018, doi: 10.7910/DVN/OHHUKH.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Springer, 2015.

[15] M. Sener *et al.*, "Multiscale modeling and cinematic visualization of photosynthetic energy conversion processes from electronic to cell scales," *Parallel Computing*, vol. 102, p. 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 *et al.*, "A cloud detection method based on a time series of MODIS surface reflectance images," *International Journal of Digital Earth*, vol. 6, no. sup1, pp. 157–171, 2013, doi: 10.1080/17538947.2013.833313.

[18] T. Wu *et al.*, "Automatic cloud detection for high-resolution satellite stereo images and its application in terrain extraction," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 121, pp. 143–156, 2016, doi: 10.1016/j.isprsjprs.2016.09.006.

[19] X. Song, Z. Liu, and Y. Zhao, "Cloud detection and analysis of MODIS image," in *IGARSS 2004*, vol. 4, pp. 2764–2767, 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*, vol. 13, no. 1, 2021, doi: 10.3390/sym13010028.

[21] Y.-C. Zhang *et al.*, "Calculation of radiative fluxes from the sur- face to top of atmosphere based on ISCCP and other global data sets," *Journal of Geophysical Research*, vol. 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*, vol. 118, pp. 83–94, 2012, doi: 10.1016/j.rse.2011.10.028.