Effects of Neural Network Architecture on Topography Estimation From Satellite Imagery for Multi-Terrain Autonomous Vehicle Path Planning and Control

Ryan Lynch[1]
Depertment of Mechanical and Aerospace Engineering
North Carolina State Univeristy
Raleigh, NC, USA
rwlynch@ncsu.edu

Sumedh Beknalkar[2]
Depertment of Mechanical and Aerospace Engineering
North Carolina State Univeristy
Raleigh, NC, USA
sbeknal@ncsu.ede

Jack Lynch[3]
Lynch Brothers Consulting
208 Cobble Place
Durham, NC, USA
jmlynch3@ncsu.edu

Dr. Andre Mazzoleni[4]
Depertment of Mechanical and Aerospace Engineering
North Carolina State Univeristy
Raleigh, NC, USA
apmazzol@ncsu.edu

Dr. Matthew Bryant[5]
Depertment of Mechanical and Aerospace Engineering
North Carolina State Univeristy
Raleigh, NC, USA
mbryant@ncsu.edu

***Abstract*— Global warming is one of the world's most pressing issues. The study of its effects on the polar ice caps and other arctic environments, however, can be hindered by the often dangerous and difficult to navigate terrain found there. Multi-terrain autonomous vehicles can assist researchers by providing a mobile platform on which to collect data in these harsh environments while avoiding any risk to human life and speeding up the research process. The mechanical design and ultimate efficacy of these autonomous robotic vehicles depends largely on the specific missions they are deployed for, but terrain conditions can vary wildly geographically as well as seasonally, making mission planning for these unmanned vehicles more difficult. This paper proposes the use of various UNet-based neural network architectures to generate digital elevation maps from satellite images, and explores and compares their efficacy on a single set of training and validation datasets generated from satellite imagery. These digital elevation maps generated by the model could be used by researchers not only to track the change in arctic topography over time, but to quickly provide autonomous exploratory research rovers with the topographical information necessary to decide on optimal paths during the mission. This paper analyzes different model architectures and training schemes: a traditional UNet, a traditional UNet with data augmentation, a UNet with a single active skip-layer vision transformer (ViT), and a UNet with multiple active skip-layer ViT. Each model was trained on a dataset of satellite images and corresponding digital elevation maps of Ellesmere Island, Canada. Utilizing ViTs did not demonstrate a significant improvement in UNet performance, though this could change with longer training. This paper proposes opportunities to improve performance for these neural networks, as well as next steps for further research, including improving the diversity of images in the dataset, generating a testing dataset from a completely different geographic location, and allowing the models more time to train.**

Keywords—Machine learning, UNet, vision transformer neural network, autonomous robotics, arctic exploration, terrain identification.

I. INTRODUCTION

In recent years, environmental scientists have watched vast tracts of Arctic Sea ice - including Greenland's ice sheet - melt away [1]. There is a growing demand for survey missions focusing on climate change research in the Arctic using rovers [2]. Arctic missions, however, can be very dangerous for researchers. A combination of the harsh arctic weather, treacherous terrain conditions, and remoteness of important research locations slows manned research efforts. To aid research efforts and eliminate the danger faced by researchers, some have proposed autonomous rover systems for these exploratory missions. Autonomous rover systems have been used in the past to explore the polar regions of the earth [3, 4], and while autonomous rovers have been deployed successfully as data collection tools in the Antarctic, they have been limited to areas of flat and mostly uniform terrain. The rapidly changing Arctic climate presents unique and heterogeneous combinations of terrain, ranging from snow and ice to firm, frozen permafrost, ice-covered lakes, and even flowing mixtures of sea ice and open ocean. In addition to the rapid changes in the terrain caused by global climate change, the terrain also varies seasonally, making the mission planning process more difficult, limiting the use of autonomous vehicles. Exciting rover technologies, including those extensively developed for Lunar and Martian exploration, can be applied to multi-terrain rovers designed for Arctic missions, but the design, control, and capabilities of these rovers would greatly depend on the mission specifications determined during the mission planning phase. Specifically, the terrain the rover will encounter is important to consider beforehand, and as previously mentioned, is difficult to predict quickly and accurately for Arctic missions.

Having topographical data such as the elevation and slope of the terrain, as well as the location of any bodies of water, is crucial not only for mission planning, but for designing the rovers themselves. Depending on the dynamics of the rover,

certain regions may be deemed too steep or inaccessible for the rover. Formulating mission targets in terms of total area of exploitation, duration of mission, research activities, and a step-by-step plan for achieving said mission targets requires the knowledge of rover locomotion dynamics and information about the environments of operation- i.e., topographical data. Additionally, the elevation of a particular mission path will also determine the rover's capabilities of communicating with either humans or other potential autonomous agents, and thereby its decision-making capabilities, or more specifically its controller. Thus, the topographical data will be essential in designing a controller that can operate with intermittent feedback. While advanced field surveying techniques exist, most methods require extensive fieldwork in, as previously mentioned, dangerous and inaccessible areas [5]. This paper proposes the use of deep neural networks with novel model architectures to quickly generate digital elevation maps for autonomous rovers to utilize during their research expeditions into the Arctic.

Neural networks (NNs) have a wide variety of uses, and can be adapted to many different problems by changing their architectures and input and output data types. Transformers, for example, are a kind of neural network that was originally developed for natural language processing [12]. The transformer architecture helps a model to identify the relationship between individual input datapoints. After their successful implementation on one-dimensional data analysis problems, they were adapted for use on two-dimensional data as well [11]. The UNet is another example of neural network architecture developed to solve a completely different problem. The UNet is built of multiple convolutional encoding-decoding pair layers linked with a skip connection designed originally to perform medical image semantic segmentation [7]. The UNet model is capable of generating new images equal in size and shape to the original input image it is fed, but able to alter them to match a desired output. This could be outlining an important feature in found in the input image, or changing the image all together. Often times, different neural network architectures are combined to enhance performance on a given problem. UNet and Transformer neural networks specifically have been combined in the past with great success [9]. This paper proposes combining transformer neural networks with UNets in a novel way, by implementing the transformer neural network within the skip connections found within the UNet. Using satellite imagery from Sentinel Hub [6] along with this novel neural network model, a digital elevation map of several regions in the Arctic can be derived.

II. Generating the Dataset

A. Generating Data With Sentinel Hub

The dataset for this paper was generated using Sentinel Hub. Sentinel hub is a big-data satellite imagery service. Data from Sentinel Hub (in this case, satellite imagery of the earth's surface captured by the Sentinel-2 satellite mission and the corresponding DEMs available on Sentinel Hub) was pulled and processed to build our data set.

An area in the Arctic that is relevant to climate change research – Ellesmere Island, was selected for this project. The original dataset was built by creating a large bounding box over Ellesmere Island that specified the longitudes and latitudes of the bottom left and top right coordinates of the box that contained the island. Twenty-thousand satellite images and their matching DEMs were then randomly sampled via smaller boxes from within this larger bounding box. In order to maximize the number of images with useful data, the bounding box was chosen in such a way as to minimize the amount of area showing bodies of water, and any images with more than 5% cloud coverage were rejected, resulting in a dataset of 11,100 images and their corresponding DEMs. These image pairs were then split into training, testing, and validation, respectively. A test dataset was not created. Figure 1 shows examples of the satellite and corresponding DEM images created in the datasets.

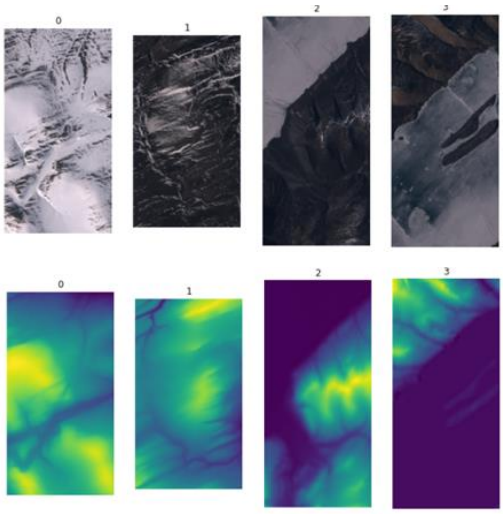

Fig. 1. The first four satellite image (top row) and DEM image (bottom row) pairs from the dataset before resizing using the transform function.

B. Possible issues with the dataset generated

1) Imbalanced instances of topographical features: Ellesmere Island has a wide variety of topographical and geographic features including mountains, plateaus, fjords, lakes, and flat plains near the coast. However, the distribution and instances of each type of feature is not uniform. Sampling satellite images of Ellesmere Isalnd evenly across the bounding box could have resulted in data imbalances that would cause the model to be more likely to identify some types of topographical features correctly and confuse the others. Similarly, since the dataset consists of randomly spliced images, the dataset could be additionally imbalanced if a large portion of the dataset was by chance disproportionately sampled from one subsection of the island's bounding box.

2) Poor overall quality of terrain images: Instances of cloud coverage could affect the image quality since clouds could be mistaken by the model's algorithm as topographical features. Similarly, depending on when the picture was taken by the Sentinel-2 satellite, glare bouncing off the snow-clad area could result in poor image quality. The issue of potential cloud coverage was tackled by filtering out images with more than 5% cloud coverage during the dataset creation process, but no solution has yet been found for removing images with a set amount of measured solar glare bouncing off snowy and icy

regions other than the manual removeal of said images based on researchers' best judgement from the dataset post-generation. As such, images with solar glare were not filtered from the dataset.

3) Lack of consistency between topographical features across different DEM images: Upon inspection of the dataset, possible inconsistencies between DEM and satelite image pairs that have seemingly similar topographical features were found. The cause of these inconsistencies is unknown.

4) Future steps for improving the quality of the dataset: Steps were taken to mitigate the effect of these technical difficulties, but more work could always be done to improve the quality of the dataset. Future steps for improving the quality of the dataset are discussed in the conclusion section of this paper.

Before the dataset was used to train the model, each image was preprocessed to assure uniformity. Since the images in the dataset were created by randomly generating latitude and longitude values that created smaller boxes within a range of sizes, the images all had slightly different shapes. The transform function in the Torchvision library was used to resize the images to a prescribed square shape to ensure that all images fed to the model as inputs had the same size. The results of this paper will show that the performance of the model varied drastically as the resolution of these images was altered, making it an important hyperparameter. Two resolutions were considered during the hyperparameter optimization phase of this project – 64 by 64 and 128 by 128.

III. Architectures Tested and Methodology

A. UNet baseline (with and without data augmentation)

A UNet is a convolutional neural network that was developed for biomedical image segmentation [7]. Its architecture consists of a specific encoder-decoder scheme: (1) The encoder (also called the contraction path) reduces the spatial dimensions in every layer and increases the channels, and (2) the decoder (also called the expansion path) increases the spatial dimensions while reducing the channels. The concatenation of feature maps from encoder to the decoder helps give localization information. The specifics of the architecture are shown in Figure 2 and described as follows:

1) Encoder (left side): It consists of the repeated application of two 3x3 convolutions. Each conv layer is followed by a ReLU. A max pooling operation is applied to reduce the spatial dimensions. At each downsampling step, we double the number of feature channels, while we cut in half the spatial dimensions.

2) Decoder (right side): Every step in the expansive path consists of a 2x2 transpose convolution, which halves the number of feature channels. We also have a concatenation with the corresponding feature map from the contraction path, and a 3x3 convolution (each followed by a ReLU). At the final layer, a 1x1 convolution is used to map penultimate features to an output prediction with the appropriate number of channels.

B. Basic data augmentation

The baseline UNet described above was trained with and without basic data augmentation. The augmentation applied

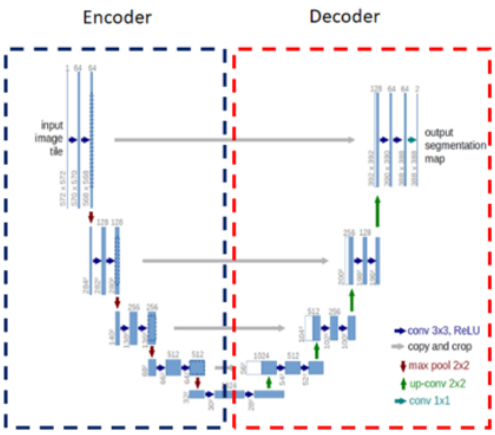

Fig. 2. Visual diagram depicting UNet model architecture from Ronneberger, O. et al. [7]

consisted of a 50% chance of applying a horizontal flip to image/label pairs during training, and a 50% chance of applying a vertical flip. The goal of these augmentations is to help make the network more robust, by preventing it from relying too heavily on shadow position and orientation in satellite imagery (meaning better performance on datasets taken from different hemispheres or at different times of the day or year). Future tests will include evaluating the model at different augmentation application probabilities to determine an optimal augmentation policy, as well as potentially include more augmentations such as scaling the images, changing the contrast or saturation of the image, or rotating the image by some prescribed angle.

C. TransUNet with single active skip layer transformer

Transformer Neural Networks were originally developed and used primarily for natural language processing tasks. More recently, transformers have been repurposed for computer vision applications, and have shown promising results in a variety of different applications.

Traditional transformers take in a vector (in NLP this vector is the representation of a sentence) input, and output a similar vector. They are unique in that they use self-attention to "understand" how each component of the sentence relates to every other component. They do this by creating a Query, Key, and Value vector for each word in the sentence by multiplying the word's vector by learned Query, Key, and Value matrices each separately. Each word is then given a score value derived by taking the dot product of the Query and Key vectors calculated for that word. This score value is soft-maxed and can be thought of as a value between zero and one given to every word in the sentence that represents how important each word in the sentence is to the word presently being encoded by the transformer. This means that each word in the sentences receives a new score when the next word in the sentence is being encoded. The value vectors for each word are then multiplied by their respective scores. This has the effect of minimizing the value vectors that are not as important or directly related to the word being encoded at the time, while keeping the important words' value vectors relatively unchanged. These updated value vectors are then summed to create the output of the self-attention

component of the transformer network for a single word in the sentence.

When utilizing transformers in computer vision, the "sentence" is an image that has been evenly split up into N number of "patches." A clever way of doing this is to pass the input image through a convolutional layer with the kernel and step size both equal to the preferred patch size, and then flattening the output into a vector. Each of the resultant patches (now in vector form) represents a single word in the image sentence.

For the purposes of this paper, the input image was 64 by 64 pixels, and was split into 16 patches, each having the dimensions of 16 by 16 pixels.

The implementation of this UNet was inspired by the implementation of a Transformer model within a UNet architecture by Chen, J. Et al [11]. In their paper, Chen, J. Et al. place their transformer model in the skip connection between the lowest most encoder-decoder pair of their UNet model. Figure 3 shows a visual representation of the network architecture utilized by Chen, J. et al.

For this paper, the transformer was placed in the skip connection between the uppermost encoder-decoder pair in the UNet. Figure 4 shows a visual representation of the architecture utilized in this paper.

D. TransUNet with multiple active skip layer transformer

It was hypothesized that implementing the vision transformers on lower-level skip connections further away from the final output of the UNet would allow the decoding convolutional layers of the model more time to smooth the output image, and move it away from the heavily pixelated look vision transformers are predisposed to outputting due to the manner in which they split up their input images and reconstruct them for their outputs.

To accomplish this, an updated Multi-Transformer UNet model was made, and it was built in such a way that a transformer block was placed in every skip-connection. The transformer blocks within each skip connection could be turned on or off when calling the model. If they were turned off, they behaved as a regular skip-connection, but if they were turned on, the skip connection was fed through a vision transformer (ViT) block. Figure 5 shows a visual of the model architecture.

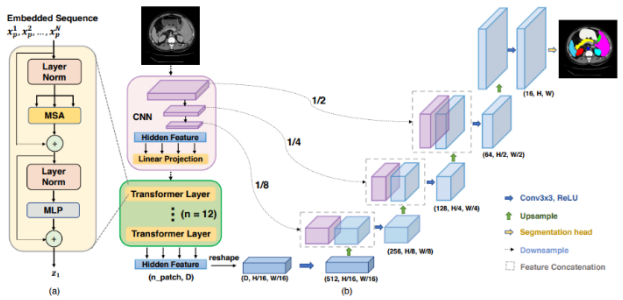

Fig. 3. Visual diagram depicting the TransUNet model from Chen, J. et al [9].

The updated model was trained on the same dataset as the previous models with every one of the transformer skip-connections turned on. This was done partially as a means of confirming that each transformer skip-connection worked, however having each transformer turned on at once may have had detrimental effects on the model's output. These effects will be discussed in section IV, subsection D.

IV. Results and Evaluation

A. Results of model arcitecture 1: UNet without data augmentation (baseline model)

The DEM data queried from Sentinel Hub was used to train the UNet baseline model in two different ways. In the first approach the model was trained using only the DEM images for labels, while in the second approach, the model was trained using the elevation data represented in the original DEM images and two additional label features, slope and aspect of

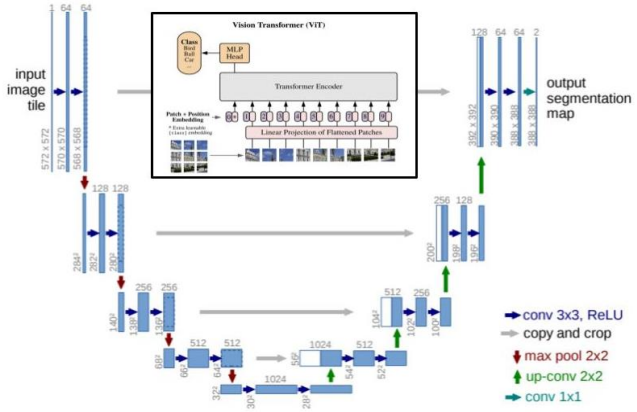

Fig. 4. Visual diagram of TransUNet model trained for this paper. A vision transformer (ViT, figure modified from Dosovitskiy, Alexey, et al. [11]) was implemented within the uppermost skip layer of the UNet Arcitechture.

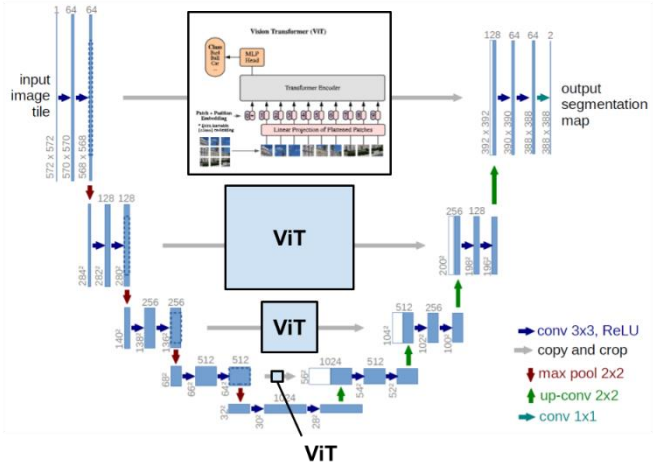

Fig. 5. Visual diagram depicting the Multi-Transformer UNet arcitechture. Four vision transformer neural networks were implimented within the skip connections of the baseline UNet model.

the terrain. The team was, however, unable to gain access to a GPU to speed the training of the model, so tuning was only feasible for certain hyperparameters (image size, number of epochs, etc.), and not with as much fidelity as would be ideal. The relevant results and hyperparameter tuning regimes that were able to be performed given hardware limitations are discussed below.

1) Using only elevation labels during training: The first iteration of the UNet model was run for 100 epochs with images of size 128x128 pixels, optimized using stochastic gradient descent (SGD) with learning rate 0.01 and batch size 16. The mean squared error (MSE) loss function was used as the training objective. Figure 6 shows the results of the model after 100 epochs and the training and validation loss curves during training.

While loss curves for both the training and validation sets show an overall downward trend, the loss curve for the training set in particular is especially noisy and does not consistently improve. This initial result prompted the researchers to make several changes to the hyperparameters and optimizer, including (1) reducing image size to simplify identifying topographical features, (2) increasing batch size to smooth the loss curve during training, and (3) moving from vanilla Stoachstic Gradient Descent to the Adam optimizer and decreasing the learning rate.

The above changes resulted in a much-improved performance of the UNet model using only the DEM images as labels. The loss curve showed a prominent downward trend visible even with half the original number of epochs, likely due to the use of the Adam optimizer and increased batch size. The reduction in image size also resulted in improved identification of topographical features, and the increased batch size reduced training time considerably. The results of this model are shown in Figure 7.

2) Using elevation, slope, and aspect labels during training: The second UNet model, referred to as the SEA model for the slope, elevation, and aspect labels it was fed, utilized three output labels as opposed to the baseline UNet model's one. In addition to elevation, this version of the model was given slope and aspect data as additional output labels, and was asked to generate slope and aspect maps alongside the elevation map for input original satellite imagery. The RichDEM library [13] was used to create labels for these features based on the original DEM data used for training.

Because this version of the UNet model was attempting to learn to predict three outputs instead of one, the method of training needed to be reconsidered. While there was still one loss function used to train the model, each predicted feature contributed its own loss term during training. These terms were weighted and summed into a combined loss that was used to train the model. Giving weights to each individual loss term created an additional set of hyperparameters that must be considered during hyperparameter tunning.

The SEA model produced the results found in Figure 8, and used a learning rate of 0.001, a batch size of 64, the Adam optimizer, equal weights for each loss function label term, and an image size of 64 by 64 pixels. The model appeared to plateau within 20 epochs, and did not improve upon the previous version of the UNet model that utilized only elevation labels yields superior results. However, this could change with further hyperparameter tuning.

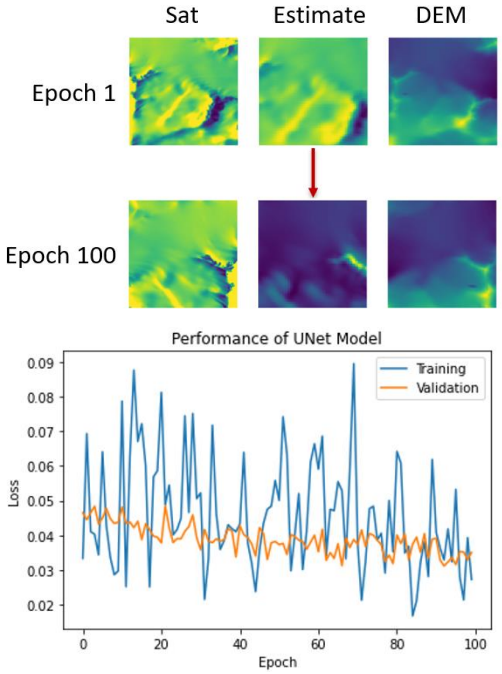

Fig. 6. The satelite image, model estimated DEM, and true DEM pairs for the first and one-hundredth epochs (top), and the loss curves for the model's performance on training and validation sets utilizing elevation data only, pre-hyperparameter tuning plotted against the number of epochs (bottom).

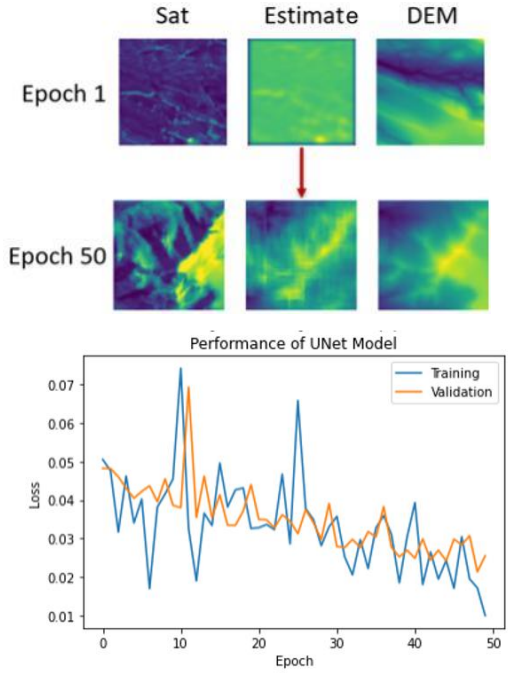

Fig. 7. The satelite image, model estimated DEM, and true DEM pairs for the first and one-hundredth epochs (top), and the loss curves for the model's performance on training and validation sets utilizing elevation data only, post-hyperparameter tuning plotted against the number of epochs (bottom).

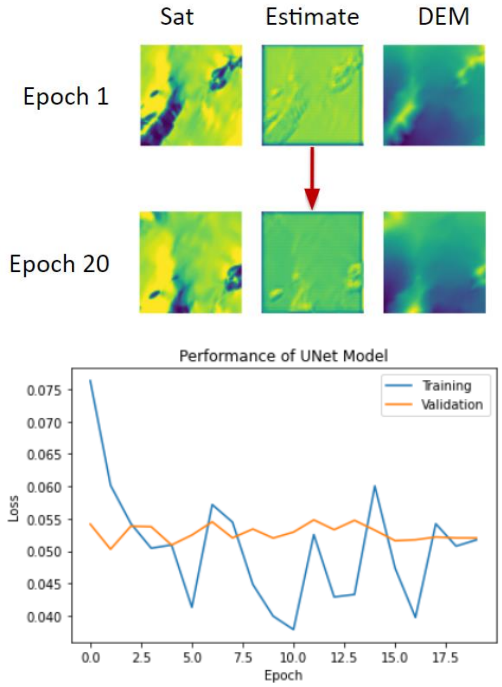

Fig. 8. The satelite image, model estimate of corresponding DEM, and true DEM pairs for the first and twentieth epochs (top), and the loss curves for the SEA UNet model's performance on the training and validation sets post-hyperparameter tunning plotted against the number of epochs (bottom).

B. Results of Model Architecture 1 with data augmentation

Next, basic data augmentation was introduced when training the baseline UNet models described above. During training, images and their corresponding labels featured a 50% chance of being horizontally flipped and an independent 50% chance of being vertically flipped.

The model was trained with an image size of 64x64 pixels, the Adam optimizer with learning rate 0.001, and a batch size of 64. Once again, the MSE loss function was used for training and evaluating performance. Figure 9 shows an input satellite image from the dataset, the model's predicted DEM after the final epoch of training, and the satellite image's corresponding true DEM. The loss curves for the training and validation sets across 20 epochs of training are shown in Figure 10.

The loss curves are not as steep and do not reach the same loss as the baseline model. Similarly to the baseline, however, the proposed model does not seem to plateau after 20 epochs. It could be that the proposed model trains slower but to an eventual lower loss than the baseline without augmentation. Allowing the model to train for a higher number of epochs could help to determine if this is the case.

Based on the first twenty epochs, the baseline model outperforms the proposed model utilizing some basic data augmentation techniques. In future research, each model will be run for a longer duration to ensure they each converge.

C. Results of Model Architecture 2: TransUNet with single active skip layer vision transformer

The third model proposed by this paper is the TransUNet model architecture with a single active skip layer vision transformer. The model was trained with an image size of 64x64 pixels, using the Adam optimizer with a learning rate of 0.001 and a batch size of 64. The MSE loss function was used for training and for evaluating performance.

Figure 11 shows an example prediction of the TransUNet model for an input satellite image on its first and twentieth epochs, alongside the corresponding ground-truth label. These outputs can be contrasted with Figure 12, which shows the outputs from the baseline UNet model that did not use a vision transformer in the uppermost skip connection, but was otherwise identical. The TransUNet model's outputs are considerably more pixelated. This could be due to the vision

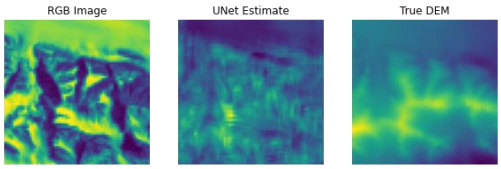

Fig. 9. The satellite image, UNet DEM estimate, and true DEM on the final (twentieth) epoch of the baseline UNet model with data augmentation.

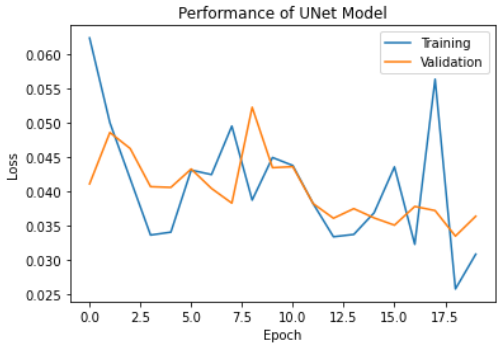

Fig. 10. Loss curves during training for the UNet Model with data augmentation on the training and validation datasets for the first twenty epochs.

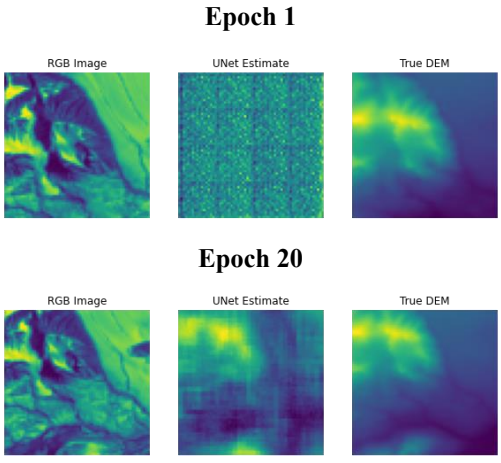

Fig. 11. TransUNet DEM predictions for 64x64 pixel images at Epochs 1 and 20 (the satellite images displayed in the leftmost column were fed into the model as full RGB images, but are only displayed here, and throughout the paper, on a blue to yellow gradient).

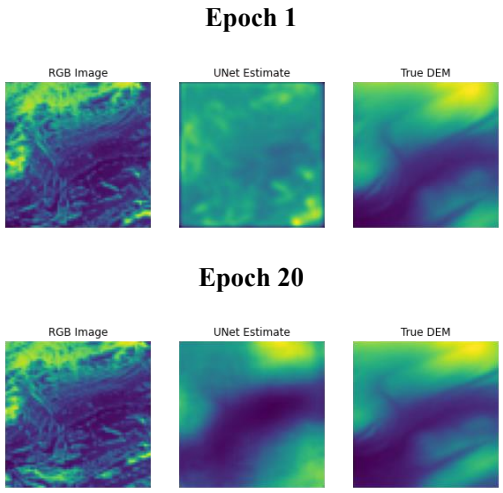

Fig. 12. Baseline UNet DEM predictions for 64x64 pixel images at Epochs 1 and 20 (the satellite images displayed in the leftmost column were fed into the model as full RGB images, but are only displayed here, and throughout the paper, on a blue to yellow gradient).

transformer underfitting. Loss values during training for the TransUNet on training and validation data are shown for images of size 128x128 pixels in Figure 13, and for images of size 64x64 pixels in Figure 14. It's unclear if either model converged with the epochs provided. Further testing with a higher number of epochs is necessary to make this determination.

D. Results of Model Architecture 4: TransUNet with multiple active skip layer transformers

The Multi-TransUNet model was trained on the same dataset as the previous models with a vision transformer applied to every skip connection. This was done in part to confirm that each transformer skip-connection worked as intended, though having each transformer enabled at once may have had detrimental effects on the model's output. Figure 15 shows the Multi-Transformer UNet model's elevation predictions on the first and twentieth epochs. The Multi-Transformer UNet model did not meaningfully learn over a period of 20 epochs. This is confirmed by the training and validation loss curves for the model shown in Figure 16. The validation loss curve does not trend downward, remaining approximately flat on average, while the training loss curve fluctuates noisily.

While the immediate results do not support the usage of a Multi-Transformer UNet model, further study is needed to determine just how detrimental, or helpful, each skip-connection transformer can be to the model on its own and when combined. Transformers are known for taking longer to train than some other architectures (including convolutional neural networks, which employ additional inductive biases), and the poor learning and validation curves shown could be a result of an error in the code, improper hyperparameter configuration, or insufficient training.

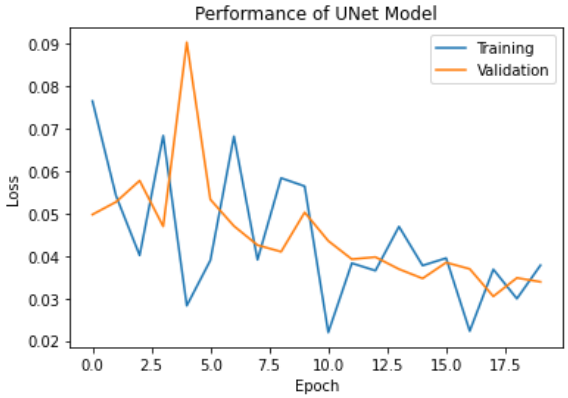

Fig. 13. Loss values during training of TransUNet model on Training and Validation sets of 128x128 images over the first twenty epochs.

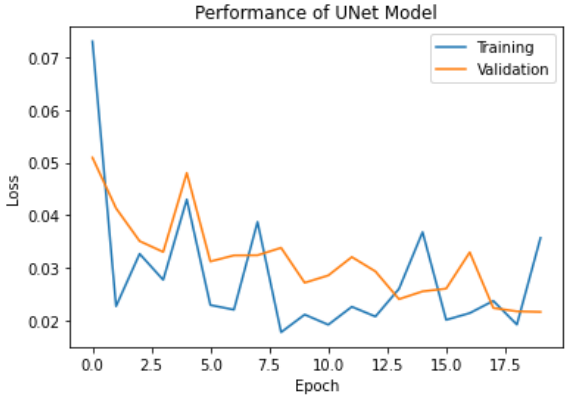

Fig. 14. Loss values during training of TransUNet performance on Training and Validation sets of 64x64 images over the first twenty epochs.

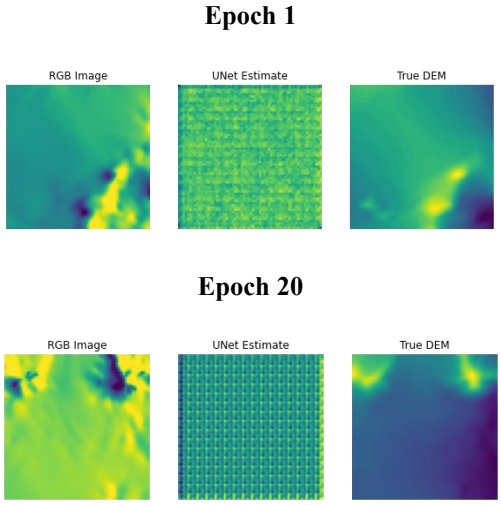

Fig. 15. Multi-Transformer UNet DEM predictions for 64x64 pixel images at Epochs 1 and 20 (the satellite images displayed in the leftmost column were fed into the model as full RGB images, but are only displayed here, and throughout the paper, on a blue to yellow gradient).

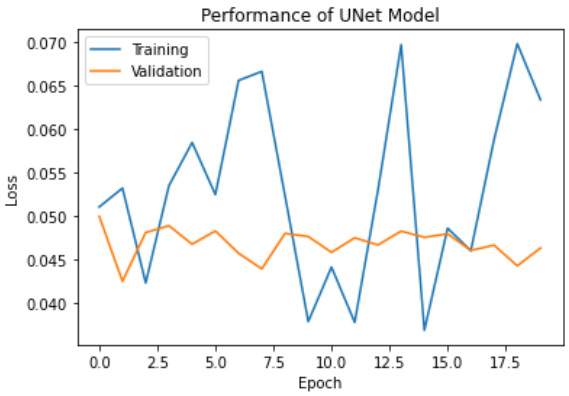

Fig. 16. Loss values during training of Multi-Transformer UNet model on Training and Validation sets of 64x64 images over 20 epochs.

V. Conclusions

In summary, this paper has compared different neural network architectures and their ability to accurately predict digital elevation maps for Arctic land depicted in satellite imagery. Specifically, the following can be said: (1) Providing the baseline UNet neural network with additional slope and aspect label data did not improve the performance of the model significantly; (2) Implementing a vision transformer within the uppermost skip connection improved the performance slightly, but implementing vision transformers within every skip connection impaired and complicated learning; and (3) simple data augmentation did not noticeably increase model performance within the first 20 epochs of training.

Future work should include: (1) Further experimentation with hyperparameter tuning for each model; (2) Training the models for a longer number of epochs so that researchers can be sure that each model can converge; and (3) Checking the dataset for imperfections that might be affecting the quality of the model outputs or the fidelity of model evaluation.

Even without in-depth hyperparameter tuning, the baseline UNet model showed promisingly accurate results when converting the satellite images to digital elevation maps. If given ample time to train, with a quality dataset, using a neural network to generate a digital elevation map from a satellite image could constitute a quick way to gain a reasonable estimation for the elevation of difficult to navigate terrain. This could be used not only for Arctic exploration, but for extraplanetary exploration as well, to better prepare autonomous rovers for exploratory missions. Including vision transformer neural networks within the baseline UNet architecture might be one way of increasing the accuracy of the neural network model's predictions, but further research is necessary to be certain.

References

[1] Chen, J. L., Wilson, C. R., & Tapley, B. D. (2006). Satellite gravity measurements confirm accelerated melting of Greenland ice sheet. science, 313(5795), 1958-1960.

[2] Cassella, C. (2020, August 12). Arctic sea ice could be gone by 2035, according to Earth's climate history. ScienceAlert. Retrieved September 29, 2021, from https://www.sciencealert.com/the-arctic-could-be-free-of-sea-ice-next-decade-if-it-warms-the-same-as-last-time.

[3] Ray, L., Price, A., Streeter, A., Denton, D., & Lever, J. H. (2005, April). The design of a mobile robot for instrument network deployment in antarctica. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation (pp. 2111-2116). IEEE.

[4] Pedersen, L., Wagner, M., Apostolopoulos, D., & Whittaker, W. R. (2001, May). Autonomous robotic meteorite identification in Antarctica. In Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164) (Vol. 4, pp. 4158-4165). IEEE.

[5] Ghorbanzadeh, O.; Blaschke, T.; Gholamnia, K.; Meena, S.R.; Tiede, D.; Aryal, J. Evaluation of Different Machine Learning Methods and.Deep-Learning Convolutional Neural Networks for Landslide Detection. Remote Sens. 2019, 11, 196. https://doi.org/10.3390/rs11020196.

[6] Sentinel Hub, https://www.sentinel-hub.com/. Accessed 3 Dec. 2021.

[7] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[8] Wang, W., Huang, Y., Wang, Y., & Wang, L. (2014). Generalized autoencoder: A neural network framework for dimensionality reduction. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 490-497).

[9] Chen, Jieneng et al (2021). TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation. ArXiv abs/2102.04306.

[10] Cao, Hu, et al (2021). Swin-unet: Unet-like pure transformer for medical image segmentation. arXiv preprint arXiv:2105.05537.

[11] Dosovitskiy, Alexey, et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

[12] Rothman, Denis (2021). Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more. Packt Publishing Ltd.

[13] Barnes, Richard (2016). RichDEM: Terrain Analysis Software. http://github.com/r-barnes/richdem.