Refining Ice Layer Tracking through Wavelet combined Neural Networks

Debvrat Varshney [1] **Masoud Yari** [1] **Tashnim Chowdhury** [1] **Maryam Rahnemoonfar** [1]

Abstract

Rise in global temperatures is resulting in polar ice caps to melt away, which can lead to drastic sea level rise and coastal floods. Accurate calculation of the ice cap reduction is necessary in order to project its climatic impact. Ice sheets are monitored through Snow Radar sensors which give noisy profiles of subsurface ice layers. The sensors take snapshots of the entire ice sheet regularly, and thus result in large datasets. In this work, we use convolutional neural networks (CNNs) for their property of feature extraction and generalizability on large datasets. We also use wavelet transforms and embed them as a layer in the architecture to help in denoising the radar images and refine ice layer detection. Our results show that incorporating wavelets in CNNs helps in detecting the position of deep subsurface ice layers, which can be used to analyse their change over time.

1. Introduction

The Greenland Ice Sheet (GrIS) has the potential to increase sea level rise by 7.4m (Shepherd et al., 2020). Model projections estimate that associated imbalances in the climate, such as snow accumulation rates, excessive melt-water runoff will continue to increase with a warming climate for the rest of the century (Shepherd et al., 2020; Montgomery et al., 2020). In this regard, analyzing the change in the ice sheet is imperative to project its climatic impact. One of the ways to analyse this change is through airborne observations, such as those during the NASA Operation Ice Bridge (OIB) mission.

The OIB mission flew a Snow Radar sensor, operated by the Center for Remote Sensing of Ice Sheets (CReSIS), which captured the horizontal profile of internal ice layers in grayscale images. Each ice layer present in these images was accumulated at a specific year, and is present at a specific depth (See Figure 2 first column). By capturing such images every year, one can calculate the change in thickness of each of these layers, which is required to calculate the overall mass loss from GrIS.

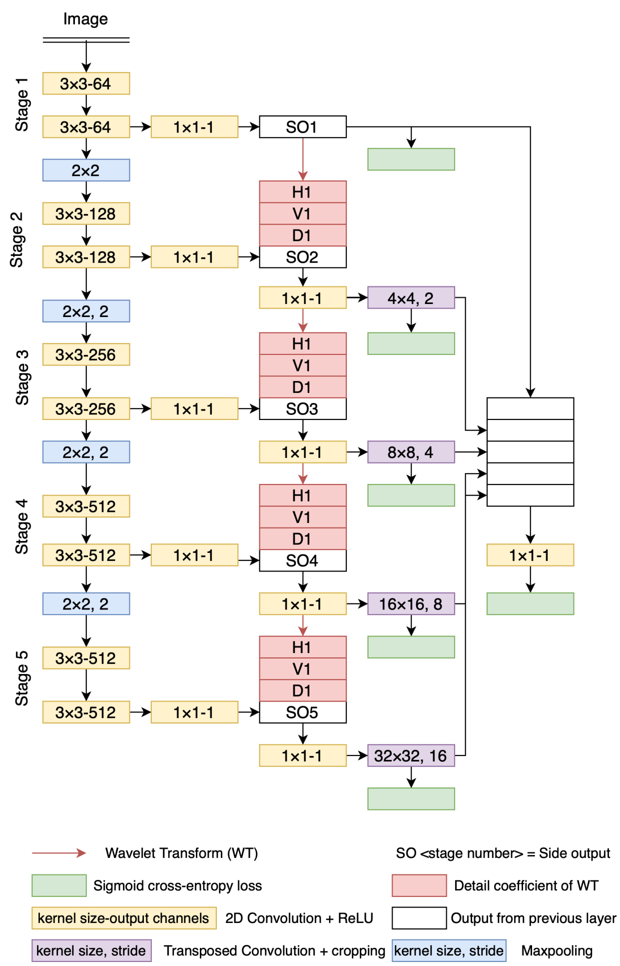

Figure 1. An end-to-end wavelet combined multi-scale architecture to learn image denoising and enhance edge detection. H1, V1, and D1 denote the horizontal, vertical, and diagonal detail coefficients of a level-1 wavelet transform, respectively.

Processing the Snow Radar images from every year can be complicated due to the data size, and also because the images can be noisy, with the ice layers being visually indistinguishable from each other. In this regard, we need effi-

[1]Computer Vision and Remote Sensing Laboratory, University of Maryland Baltimore County, Baltimore, MD, USA. Correspondence to: Maryam Rahnemoonfar <maryam@umbc.edu>.

Tackling Climate Change with Machine Learning Workshop at ICML 2021.

cient, automated algorithms which can denoise the dataset and extract complex features from the images. Recently, deep learning has been applied on Snow Radar images (Yari et al., 2019; 2020; Rahnemoonfar et al., 2021; Varshney et al., 2020) , due to its success in building efficient computer vision algorithms, and also for being robust across large datasets. These works note that the models can be improved further, if the inherent noise in the images could be learnt and reduced in an end-to-end architecture.

Wavelet transforms (Mallat, 1989) are signal processing techniques which help in image denoising. They can represent images in a multiresolution format and depict the contextual as well as textural information of an image at different scales (Huang et al., 2017). In this work, we show that by taking wavelet transforms at intermediate stages of a neural network architecture (Figure 1), the denoised output produced after every stage is more 'learned' and helps in detecting ice layers as sharp edges.

2. Related Works

Recently, there has been a lot of work with detecting ice layers from radar images, as well as improving the computational complexity and feature detection of deep learning by incorporating wavelet transforms. In this section, we highlight the related literature in both these domains.

2.1. Deep Learning for Ice Layer Detection

Deep learning has been extensively applied on remotely sensed radar images (Yari et al., 2019; 2020; Rahnemoonfar et al., 2021; Varshney et al., 2020) for its ability in feature extraction and utility in automatically processing large datasets. Yari et al. (2019) used a multi-scale deep CNN to track the internal ice layers in Snow Radar images. The authors of the work also experimented with pretraining on BSDS dataset (Arbeláez et al., 2011) and found that the network does not work well because of the inherent noise in the Snow Radar dataset. This work was further expanded in Rahnemoonfar et al. (2021) where the authors trained the multi-scale network on synthetic Snow Radar images, to make the training more robust. A multi-scale network was also used in Yari et al. (2020) where the authors used a model trained on images from the year 2012, and fine tuned it by training on different years. Further, Varshney et al. (2020) also found that using pyramid pooling modules, a kind of multi-scale architecture, helps in learning the spatio-contextual distribution of the ice layer pixels. All these works used multi-scale networks in order to extract the ice layer edges from Snow Radar images and noted that noise in Snow Radar images is an issue which needs to be addressed.

2.2. Wavelet combined CNNs

Recently, there has been a surge to combine and replace layers in a typical CNN with a wavelet transform layer such as Liu et al. (2019), Williams & Li (2018), Huang et al. (2017), Han & Ye (2018), and Bae et al. (2017). Williams & Li (2018) found that downsampling through a wavelet transform layer can lessen the creation of jagged edges as compared to max and average pooling layers. Liu et al. (2019) noted that wavelet transform can be used to enlarge the receptive field of a kernel and improve efficiency while downsampling feature maps. Doing so also helped in avoiding information loss which happens in a typical pooling operation. Huang et al. (2017) used a wavelet based loss function and a wavelet combined CNN for super resolution of multi-scale face images. Bae et al. (2017) discovered that training on wavelet subbands can help in residual learning. These works concluded that embedding wavelets in neural networks helps in achieving sharper boundaries, less artifacts and improves feature learning.

3. Dataset

We use Snow Radar images captured by CReSIS in 2012 and publicly hosted at CReSIS (2012). The images have a vertical resolution of less than 4cm per pixel. The dataset contains 2361 training images and 260 test images. There are both multi-class labels and binary labels available for each image. We use the latter as they are more useful for binary edge detection. .

4. Methodology

The architecture of our proposed network is shown in Figure 1. The backbone architecture is a VGG13 architecture (Simonyan & Zisserman, 2015) without the terminal fully connected layers, and the last pooling layer. On the last layer of every stage (see Figure 1), i.e. before each max pooling layer, we do a 1×1 convolution, to create an intermediate feature map which we call as a 'Side Output'. From stage 2 to stage 5, the dimension of each side output is half of the previous side output due to the downsampling by a max pool operator in between two stages. We then take a level 1 Discrete Wavelet Transform (DWT) (Mallat, 1989; Williams et al., 2018) of the i^{th} side output, where $i \in [1, 5]$, and concatenate the detail coefficients H, V, D (Huang et al., 2017; Williams et al., 2018) of DWT with the $(i+1)^{th}$ side output. The detail coefficients produced by a DWT have half the dimension of its input, hence, their dimensions match with each successive side output layer and can be concatenated. Then, we convolve a 1×1 kernel with the concatenated product and upsample it to the dimensions of the input radar image, using transposed convolution. We also crop the upsampled product in order to rectify any

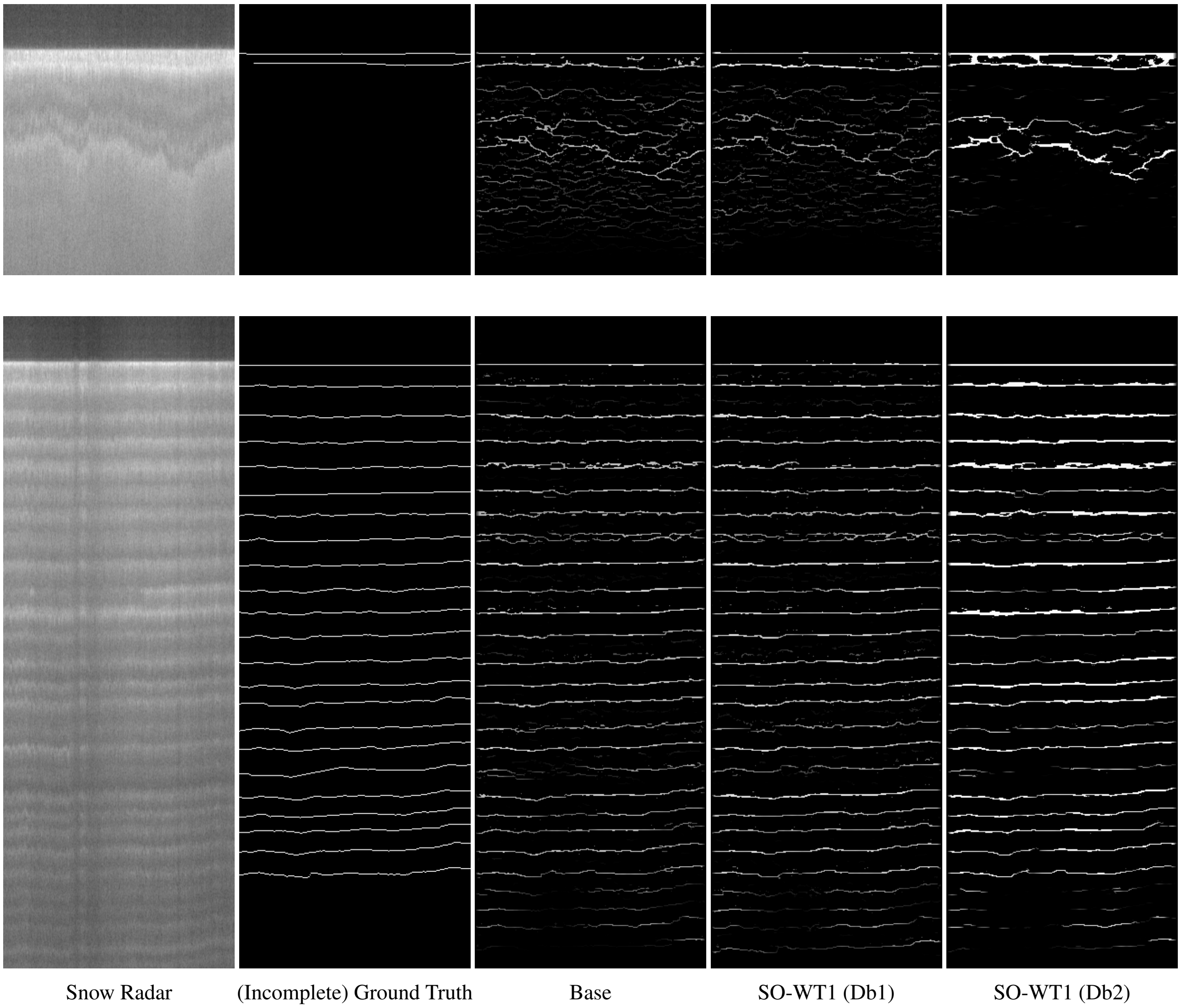

Figure 2. Qualitative comparison of model outputs. The first and second columns show sample Snow Radar images and their incomplete ground truth labels, respectively. The third column shows outputs from the Base network. The fourth and the fifth column show outputs from the SO-WT1 architecture with db1 and db2 wavelets, respectively. All network outputs are after NMS processing.

minor dimensional mismatch. All the upsampled feature maps from the five stages are finally concatenated, followed by a 1×1 convolution to form a 'fuse' layer. We train the network in a deeply supervised manner (Lee et al., 2015) and take a cumulative loss function from each of the five stages as well as the fused output (Equation 3). The position where we calculate loss is marked with the green cells in Figure 1.

We compute the binary cross entropy loss (l) for every pixel i as:

$$l(x_i; W) = \begin{cases} \alpha \cdot \log(1 - x_i) & \textit{if}\, y_i = 0 \\ \beta \cdot \log(x_i) & \textit{if}\, y_i = 1 \end{cases} \tag{1}$$

where x_i is the sigmoid activation map(s) obtained from a network with weights W. y_i is the ground truth label of the corresponding pixel in the input image I having a total of $|I|$ pixels. Further, α and β are defined as:

$$\begin{aligned} \alpha &= \lambda \cdot \frac{|Y^+|}{|Y^+| + |Y^-|} \\ \beta &= \frac{|Y^-|}{|Y^+| + |Y^-|} \end{aligned} \tag{2}$$

where $|Y^+|$ denotes the count of all positive labels, i.e. those marked as being an edge pixel ($y_i = 1$) and $|Y^-|$ denotes the count of all negative labels i.e. those marked as nonedge ($y_i = 0$). λ is a hyperparameter used to balance these positive and negative labels.

The total loss is computed as:

$$L(W) = \sum_{i=1}^{|I|} \left(\sum_{k=1}^{K} l(x_i^k; W) + l(x_i^{fuse}; W) \right) \tag{3}$$

k depicts each of the side outputs or stages, i.e. $K = 5$.

We name our proposed architecture, the one shown in Figure 1 as 'SO-WT1', with which we use a Debauchies wavelet (db2). We also experiment with a Haar wavelet (db1), a baseline network called 'Base', which does not have any wavelet transforms, and an architecture called 'WT4' where we take wavelet transforms of the input radar image, rather than the side output from the architecture. Both Base and 'WT4' are shown in Appendix A. All architectures share the same hyperparameters as those used in Rahnemoonfar et al. (2021).

5. Results

We perform non-maximal suppression (NMS) on the network outputs after which we calculate ODS (optimal dataset scale) and OIS (optimal image scale) F-scores using Dollár (2016). We tabulate these in Table 1 and plot the ODS precision recall curves for all the networks across various dataset-scale thresholds in Figure 3. Using F-score for evaluating ice layer detection is necessary since it can give a balanced outlook of both sensitivity and precision of layer detection. We primarily use the ODS F-score to evaluate our network outputs since it finds an optimal threshold for the entire dataset, made up of varying images from wet and dry snow zones, images having constructive or destructive interference etc. We also show the qualitative results of our network outputs, post-NMS, in Figure 2.

Network	Wavelet	ODS	OIS
Base	NA	0.726	0.764
WT4	Db1	0.728	0.759
SO-WT1	Db1	0.740	0.766
SO-WT1	Db2	**0.746**	**0.780**

Table 1. ODS and OIS F-measures obtained by different networks

From Table 1, we see that SO-WT1 networks gave the highest ODS F-scores, with db2 performing better than db1. This can also be seen from Figure 2 where SO-WT1 with db2 gave much sharper and brighter outputs, while the other network outputs detected intermediate fluctuations as ice layer-edges. What is more interesting to note is that wavelet based networks were able to detect deeper layers which the Base network could not. Further, although the db2 post-NMS output is much sharper, some middle portions of the layers could be missing from them. That is a trade-off, and one can choose the wavelet depending upon use case. The WT4 network performed slightly worse since the wavelets in it are generated through the input radar image, and are comparitively constant matrices added into the architecture. On the other hand, for SO-WT1, the wavelets are generated from a convolutional layer, and will depend on the weights of that layer, which would change after each iteration of training. Hence, wavelets built in the latter architecture are more 'learned' than the former architecture. Thus, not only does wavelet transformation help in denoising the image, they help in model learning and detecting deeper ice layers in the form of sharp edges for our dataset.

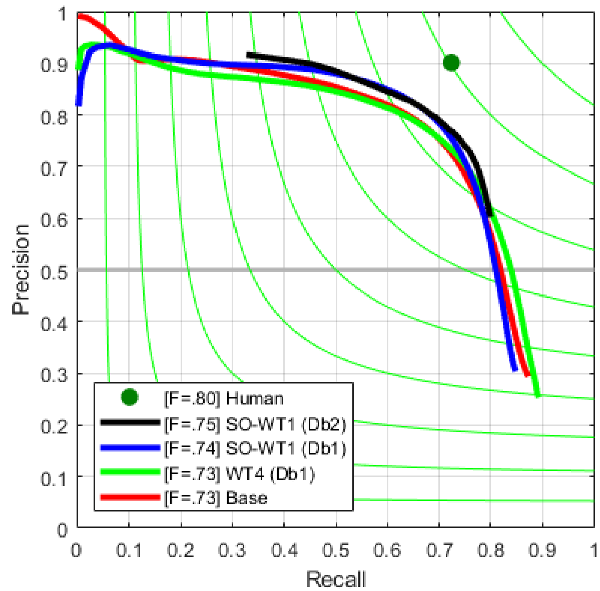

Figure 3. ODS F-score and precision-recall curve obtained by the four networks

6. Conclusion

The Greenland Ice Sheet is melting very quickly, and it is essential to assess the rate of this change. Typically, annual state of this ice sheet is captured through Snow Radar images which are very noisy. This work saw the use of wavelet transformations in convolutional neural networks to denoise the images and improve ice layer detection by detecting deeper layers with sharp boundaries. We also learnt that incorporating wavelet transforms after a convolutional layer, helps in building more 'learned' detail coefficients of the wavelet. Such a combination of CNNs with wavelet is ideal for denoising and feature learning. A generalized and robust model in this way can be applied to multiple ice datasets from different years, such that the change in the position of ice layers can be quantified to enhance the predictability of climate models.

Acknowledgements

This work is supported by NSF BIGDATA awards (IIS-1838230, IIS-1838024), IBM, and Amazon.

References

Arbeláez, P., Maire, M., Fowlkes, C., and Malik, J. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916, 2011. doi: 10.1109/TPAMI.2010.161.

Bae, W., Yoo, J., and Ye, J. C. Beyond Deep Residual Learning for Image Restoration: Persistent Homology-Guided Manifold Simplification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1141–1149, 2017. doi: 10.1109/CVPRW.2017.152.

CReSIS. NASA OIB Greenland, 2012. URL `https://data.cresis.ku.edu/data/temp/internal_layers/NASA_OIB_test_files/image_files/greenland_picks_final_2009_2012_reformat/2012/`.

Dollár, P. Piotr's Computer Vision Matlab Toolbox (PMT). `https://github.com/pdollar/toolbox`, 2016.

Han, Y. and Ye, J. C. Framing U-Net via Deep Convolutional Framelets: Application to Sparse-View CT. *IEEE Transactions on Medical Imaging*, 37(6):1418–1429, 2018.

Huang, H., He, R., Sun, Z., and Tan, T. Wavelet-SRNet: A Wavelet-based CNN for Multi-scale Face Super Resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1689–1697, 2017.

Lee, C.-Y., Xie, S., Gallagher, P., Zhang, Z., and Tu, Z. Deeply-Supervised Nets. In *Artificial intelligence and statistics*, pp. 562–570. PMLR, 2015.

Liu, P., Zhang, H., Lian, W., and Zuo, W. Multi-Level Wavelet Convolutional Neural Networks. *IEEE Access*, 7:74973–74985, 2019.

Mallat, S. G. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

Montgomery, L., Koenig, L., Lenaerts, J. T. M., and Kuipers Munneke, P. Accumulation rates (2009–2017) in Southeast Greenland derived from airborne snow radar and comparison with regional climate models. *Annals of Glaciology*, 61(81):225–233, 2020. doi: 10.1017/aog.2020.8.

Rahnemoonfar, M., Yari, M., Paden, J., Koenig, L., and Ibikunle, O. Deep Multi-Scale Learning for Automatic Tracking of Internal Layers of Ice in Radar Data. *Journal of Glaciology*, 67(261):39–48, 2021.

Shepherd, A., Ivins, E., Rignot, E., Smith, B., van Den Broeke, M., Velicogna, I., Whitehouse, P., Briggs, K., Joughin, I., Krinner, G., et al. Mass Balance of the Greenland Ice Sheet from 1992 to 2018. *Nature*, 579 (7798):233–239, 2020.

Simonyan, K. and Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*, 2015.

Varshney, D., Rahnemoonfar, M., Yari, M., and Paden, J. Deep Ice layer Tracking and Thickness Estimation using Fully Convolutional Networks. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 3943–3952. IEEE, 2020.

Williams, T. and Li, R. Wavelet Pooling for Convolutional Neural Networks. In *International Conference on Learning Representations*, 2018.

Williams, T., Li, R., et al. An Ensemble of Convolutional Neural Networks Using Wavelets for Image Classification. *Journal of Software Engineering and Applications*, 11(02):69, 2018.

Yari, M., Rahnemoonfar, M., Paden, J., Oluwanisola, I., Koenig, L., and Montgomery, L. Smart Tracking of Internal Layers of Ice in Radar Data via Multi-Scale Learning. In *2019 IEEE International Conference on Big Data (Big Data)*, pp. 5462–5468. IEEE, 2019.

Yari, M., Rahnemoonfar, M., and Paden, J. Multi-Scale and Temporal Transfer Learning for Automatic Tracking of Internal Ice Layers. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pp. 6934–6937, 2020. doi: 10.1109/IGARSS39084.2020.9323758.

A. Other Architectures

Our Base network, Figure 5, is a multi-scale architecture without any wavelet transforms, as that used in Rahnemoonfar et al. (2021), except that ours has a VGG13 backbone network. The side outputs in this architecture do not have any wavelet transforms associated with them. We also built a model, where we took a level-4 wavelet transform of the input radar image, and concatenated the detail coefficients from level i with side output $i+1$ of the Base network. This architecture, which we call WT4, is shown in Figure 6. Both architectures use a common legend, Figure 4, which defines the cells in the diagram and the convolutional operators associated with them. Further, Figure 7 shows an example of a level-2 wavelet transformation.

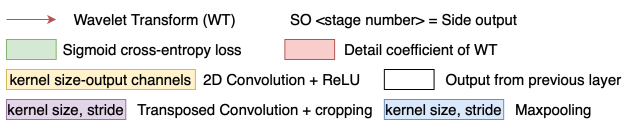

Figure 4. Legend for the Base and WT4 architectures in Figures 5 and 6, respectively.

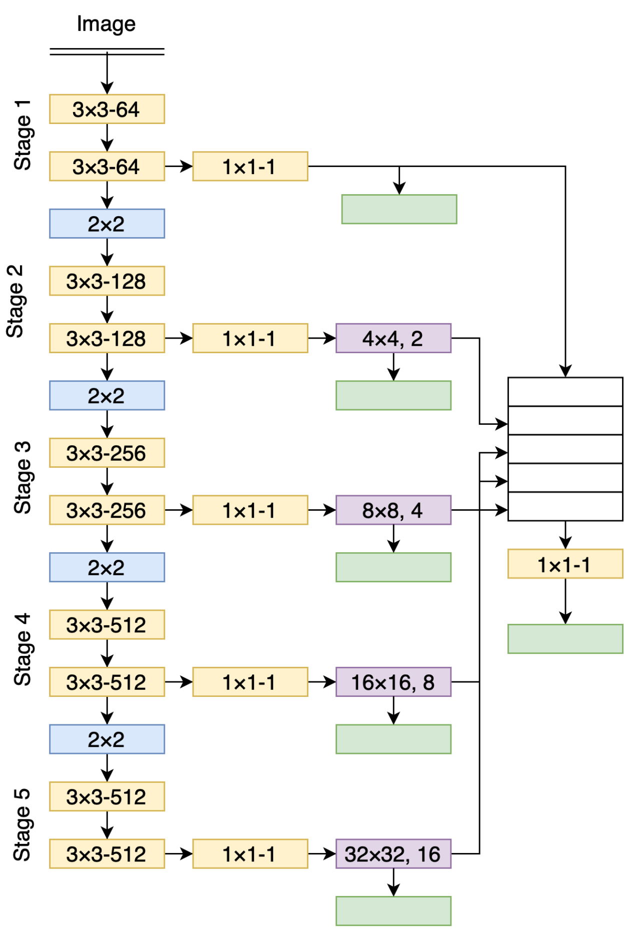

Figure 5. Out Base network, that is an architecture without wavelet transforms, same as the one used in Rahnemoonfar et al. (2021) but with a VGG13 (Simonyan & Zisserman, 2015) backbone.

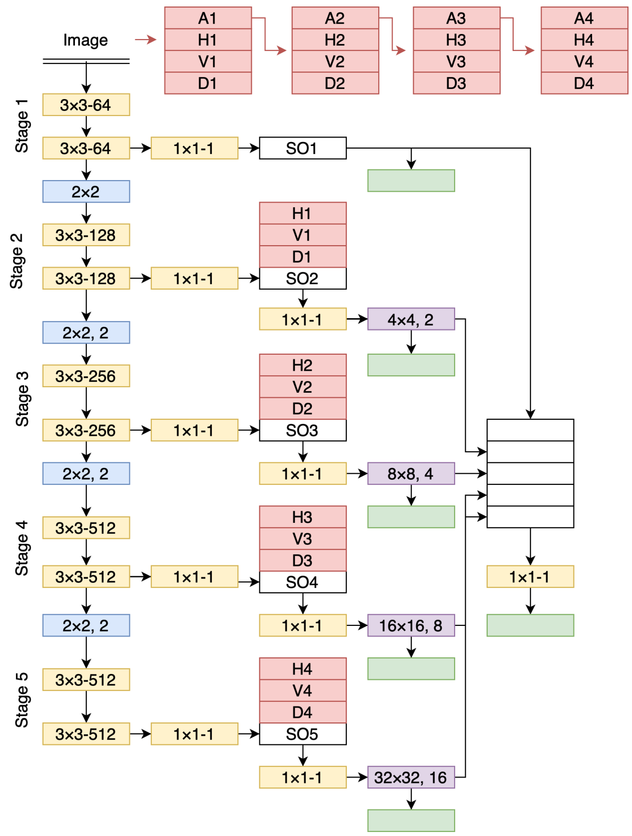

Figure 6. WT4 network where we take a level 4 wavelet transform of the input radar image, and concatenate the detail coefficients from level i with side output $i+1$, to refine multi-scale edge detection.

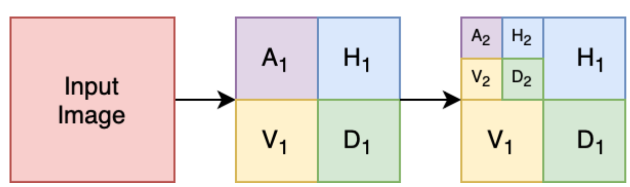

Figure 7. A level 2 wavelet transform of a given input image. The subscript denotes the level number. A, H, V, D denote the Approximation, Horizontal, Vertical, and Diagonal coefficients of a wavelet transform, respectively.