Project report for

"The Analysis of Space-time Events with Deep Learning Methods"

NKFIH PD128699
09/01/2018-08/31/2022

Tamas Zsedrovits

Introduction

The amount of data gathered over the last few decades has exponentially increased on many fronts, from consumer electronics to scientific measurements. More and more data is collected, stored, transferred, and processed as various sensors become more and more affordable. Since it is impossible to pick out the most important portions of the data, it has become necessary in many applications to extract pertinent information from large amounts of data. Traditional data processing algorithms are frequently unable to handle this volume of data. The issues are numerous. For instance, the computations are too time-consuming, or the results are only accurate for a small portion of the input.

The following three research goals were established for this project based on my prior work and the field's current state-of-the-art. 1) Examine how Self-Organizing Maps (SOM, Kohonen Network) can be used to solve the issue of finding events in large amounts of data. 2) creating and using Where-What Networks for the detection of objects. 3) Look into pruning techniques and how they might be used on the designed networks. A new network structure to process incoming video from a drone for object detection with GPGPU or FPGA implementation, an extension of the SOM network, and a technique to reduce the neural network size while maintaining performance were among the potential outcomes. The goal was to publish three papers at international conferences and two peer-reviewed journal articles.

One journal paper and three conference publications were published throughout the project. Additionally, one conference abstract has been submitted with the option to submit an extended abstract (which is still being reviewed), and another journal paper's draft is almost complete. The results of this unpublished work are described in greater detail in the following pages, while the results of these publications are briefly introduced.

KNN Algorithm Predictor for Data Synchronization of Ultra-Tight GNSS/INS Integration

The journal article [1] presents a method for localizing aircraft using extremely tight GNSS-INS coupling. The shortcomings of each system can be balanced out by the combined strengths of the GNSS and INS systems. This paper introduces a new technique of using a predictor on GNSS output before integrating with the INS. The time synchronization between the INS and GNSS must be achieved to simultaneously estimate the errors from the two systems. Therefore, a prediction between the sampling instants of the GNSS receiver is necessary. While GNSS receiver information is available once every second, INS information is available every 0.1 seconds. For GNSS receivers, various predictors can be used, with the best one being chosen based on the outcomes of the predictions. In order to satisfy the synchronization process between INS and GNSS and to predict the output of the GNSS receiver when its signal is lost (data blocking) for a short period of time, the KNN predictor algorithm searches the database for data that is similar to the current data.

Due to its advantages, such as anti-jamming immunity and increased dynamic ranges, the ultra-tight integration of GNSS and INS was utilized in this paper. Different scenarios, both with and without blocking for a brief period of time, involved clutching the GNSS data. The results show that as the GNSS-data-blocking period is extended, the error values in the three axes (X, Y, and Z) get more significant. The error values rise when the GNSS data-blocking time is prolonged. Additionally, for the same blocking periods in the GNSS data, the errors obtained using GNSS/INS with predictor are lower than those obtained using INS alone and GNSS without predictor. Using GNSS/INS with the KNN predictor will result in a smooth change in the standard deviation in all three axes, even when there is a blocking time in the GNSS receiver, according to the measurements presented in the paper. The paper's topic was not initially mentioned in the research proposal, but the accurate estimation of the UAV ego-motion is crucial for object detection [2].

3D Cave Mapping with UAVs

The conference paper [3] presents a proof-of-concept system to produce an environment map, and it could all be implemented onboard a drone. We currently have a camera that can recognize a specific object and calculate its distance using laser data and a laser-based mapping system that can produce 2D maps. These components are housed on a small 3D printed module that is simple to mount on a drone. Maps can only be created in 2D and must be expanded to 3D. Given the currently available technology, there are two viable options for an onboard sensor setup. Two 2D LiDAR sensors rotated 90 degrees to one another, and a solid-state 3D LiDAR. An Intel RealSense L515 camera is being tested at the moment [4]. The project's ultimate objective is to develop an autonomous system allowing the drone to map the cave without user input. Even though it's a long shot, this objective calls for a real-time map to be accessible. An onboard drone can simultaneously create a lower resolution map in real-time instead of making the map offline. The measurements can be recorded to develop high-resolution offline maps.

Bio-motivated vision system and artificial neural network for UAV obstacle avoidance

The conference papers [5], [6] and the initial testing results describe the system developed for object detection and avoidance for small UAVs in an indoor environment. The findings demonstrate that, while there is no discernible difference between convolutional networks trained with and without preprocessing in terms of performance, preprocessing can shorten training times and reduce the size of training sets needed to achieve adequate performance.

Figure 1. Test area and sample images from the drone camera. The first row shows the first test site, the gym of the University. The second row shows the second test site, the dining hall of the University. The left column is an overview of the environment with the obstacles. The middle column is the image from the drone camera, and the right column is the annotated image showing the object masks. The red mask is the display panel, and the green mask is the table with chairs.

Various types of neural networks were developed and tested for detection, including YOLO [7] and U-net [8] variants. New publications on the Where-What-Network (WWN) during the project's second year have brought attention to the fact that this structure is not yet sufficiently developed to be used in practice and is probably not the best candidate for further research due to its computational complexity [9]. Therefore, we maintained only the core concept of WWN in the conference publication written during the reporting period, namely that the developed algorithm uses a simplified human retina model during preprocessing.

Additional investigation into object detection and control for avoidance maneuvers is covered in the journal draft. A new dataset was gathered and annotated in order to continue working on object detection and avoidance. We used chairs and tables in addition to the display panel, which was the only object type used (Figure 1). Additionally, the environment grew more complex. The tests were conducted in the dining hall rather than in the University's gym. During the measurements, the lighting in the dining room was also changing.

We performed neural network parameter tuning for the journal paper and created three test versions. The experiment demonstrated that all variants offered the control module adequate input for avoiding collisions and navigating the space. The control was also updated with image-based visual servoing [10] (Figure 2). We state that our system can adapt to various lighting conditions and avoid two types of objects. The necessary dataset size and the learning time are also reduced because of the bio-motivated preprocessing mimicking the mammal retina. We plan to publish the annotated dataset too in a public repository and a describing journal paper. The system can be augmented for further development with more object types and a more advanced navigation module.

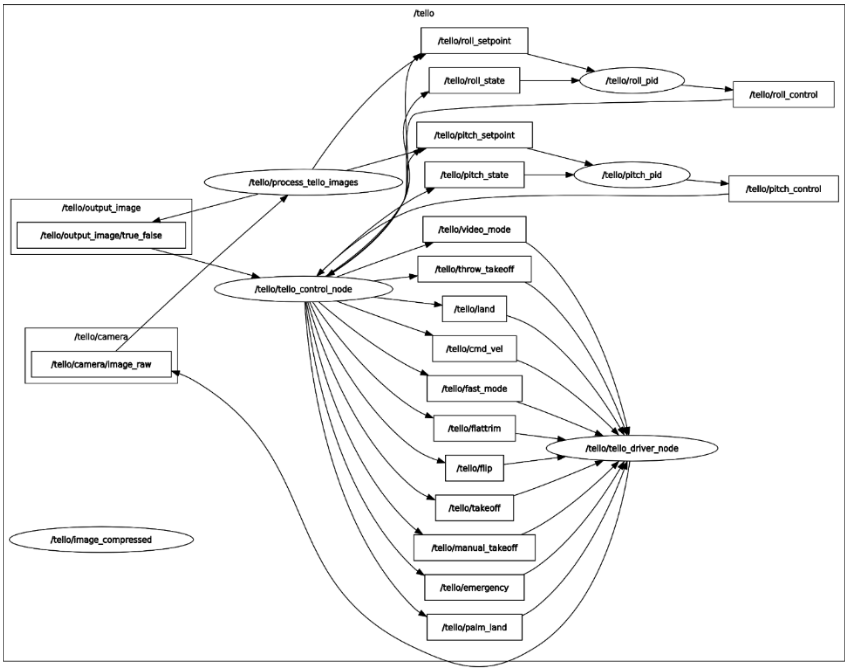

Figure 2. RQT graph of ROS nodes and topics running on the control computer. tello_driver_node provides the topics for drone commands, process_tello_images node runs the preprocessing and the U-net, tello_control_node calculates the free area and sends the adequate control command to the driver node. Two independent PID controllers calculate the drone's appropriate pitch and roll commands.

Event detection with Self-Organizing Maps for Feature Extraction of Wind Fields

In the conference abstract sent to the 22nd Conference on Artificial Intelligence for Environmental Science [11] as part of the 2023 American Meteorological Society meeting, a new method was introduced to detect vortices in flow fields. The primary motivation is that it is essential to understand flow interactions in complex terrain for better modeling and forecasting. The paper introduces a method that can be used to automatically detect different kinds of events in big datasets, such as the observations of two MATERHORN field campaigns [12], [13].

Self-organizing Maps (SOM), also called Kohonen networks, were successfully used in the past to determine the main characteristics of large-scale meteorological systems [14]–[18]. The main advantage of using SOM is that it can drastically lower the dimension of the data. Meanwhile, it enhances essential features while keeping the topological connections which is present in the input.

To detect primary events like flow patterns and vortices automatically, we use SOM to learn the main characteristics of our data for a given period. This learned pattern provides the "background," which can be neglected during the feature extraction – an analogy to object detection in computer vision [19]. The background is coded with the weight vector of the SOM neurons. For the detection, a new input is coded with the Vector Quantization (VQ) technique, primarily used for lossy image compression[20]. The VQ forms a codebook for small parts or tiles of the image and describes images with a vector and the codebook. The coded input is reconstructed based on the codebook formed by the SOM neurons' weight vectors and neglected from the original input „image," calculating a saliency map. The features can be determined on the saliency map by a threshold or a more sophisticated method (like a small neural network) to avoid false detections due to noisy input.

The method's performance is tested in a simulation where the wind field is calculated as an analytical solution of the stream function [21]. The simulation consists of 313 timesteps and a 144x144 lattice where three vortices are generated with changing circulation rates and a dominant, permanent flow. The output of the simulation is a two-component vector field for each timestep. The SOM network's input is a 450-element 1D vector consisting of 3x3 tiles of the original vector field, the two components of wind placed in an interleaved manner throughout 25 timesteps. The background is generated with the first 25 steps consisting only of the permanent background flow with a small random fluctuation.

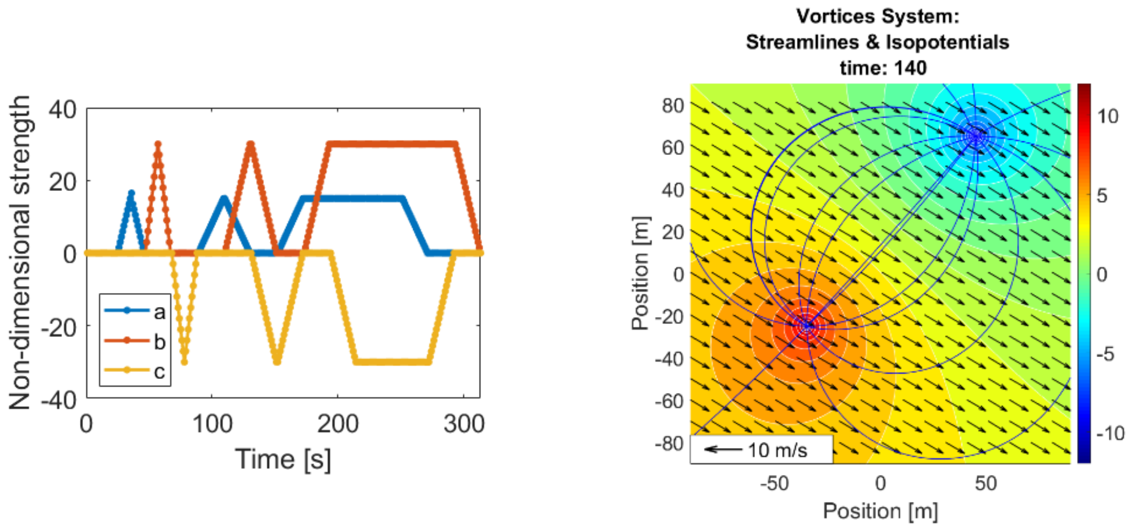

Figure 3. The left graph shows the flow field rates of the three vortices generated in the simulation. The right graph shows an example vector field with isopotentials and streamlines in the background at 140s in the simulation.

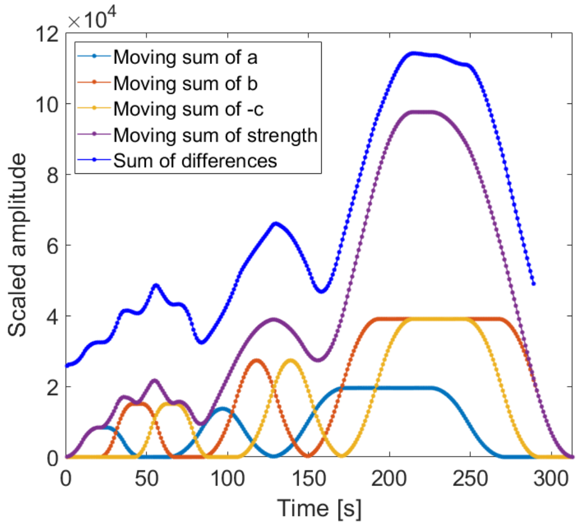

Figure 4. The graph shows the moving sum of flow field rates of each vortex and the whole vector field with a 25s window compared to the sum of differences calculated from the original vector field and the reconstructed vector field. The sum of differences follows the moving sum of the whole vector field in time.

The detection performance of the method is measured, also describing the parameter tuning. Based on the measurements on simulated data, a 4x4 SOM network was enough to get the desired accuracy. Figure 5 shows the structure and the sample hits of the SOM network. Test of the method on measured and simulated data is also shown for observations and Weather

Research and Forecast (WRF) model output from the two MATERHORN field campaigns. The background or dominant flow for such experiments can be constructed with the help of Proper Orthogonal Decomposition (POD), where the first mode is the dataset's average.

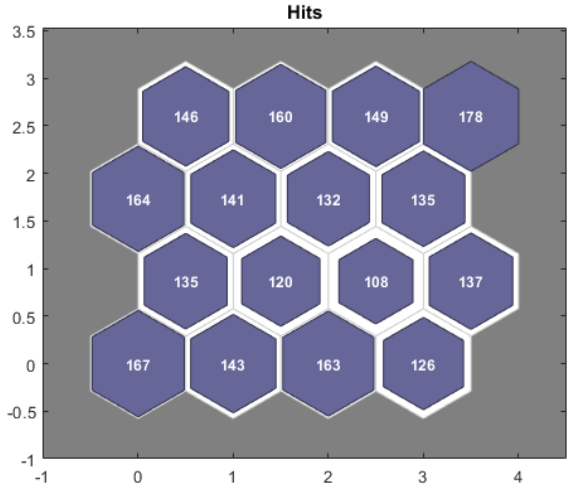

Figure 5. Structure and the sample hits of the 4x4 SOM network trained with MATLAB [22]

Network pruning

For the network pruning, I do not currently have publishable results yet. Several measurements were done to try and reproduce the original results, and we used publicly available datasets and source code for testing. Without going into details about the algorithms, we got the following results. The tables show that the original measurements are reproducible (Table 1).

I briefly introduce the tested methods. The SNIP algorithm's goal is to identify significant (or sensitive) connections so that less significant connections can be eliminated without the need for pruning and relearning cycles. An auxiliary variable, which displays the strength of the connection between two neurons based on the impact of altering a connection to the loss function, aids in achieving the desired result [23], [24]. In contrast to the SNIP algorithm, the GraSP method tries to minimize the change in the loss after the first training step by removing the weights that "reduce gradient flow" the best [25], [26]. With Soft Filter Pruning (SFP), filters can be removed from a model while it is being trained from scratch or after it has already been trained. Each training epoch involves optimizing and training the entire model using the training data. In addition to maintaining the original models' model capacity, it also avoids the greedy layer-by-layer pruning process and enables almost simultaneous pruning of all layers [27], [28].

In contrast to traditional approaches, "Pruning from Scratch" (PfS) enables researchers to obtain a pruned structure from randomly initialized weights immediately. When pruning the network, the micro-level layer settings, particularly the number of channels in the layers and the channel pruning strategy, receive the majority of attention. An auxiliary variable also determines each channel's significance and considers and optimizes the sparsity ratio in addition to the significance of the channels [29], [30].

Method	Network	Dataset	Parameters	Accuracy loss	
				Original paper	Measured
SNIP	LeNet5	MNIST	Batch size = 100 Target sparsity = 0.95	0.8%	0.8%
GraSP	VGG19	Tiny ImageNet	Batch size = 128 Target ratio = 0.98 Epoch = 300 Learning rate = 0.1	4.1±0.34%	3.73%
GraSP	VGG19	Cifar10	Batch size = 128 Target ratio = 0.98 Epoch = 160 Learning rate = 0.1	2.04±0.12%	1.84%
GraSP	VGG19	Cifar100	Batch size = 128 Target ratio = 0.98 Epoch = 160 Learning rate = 0.1	5.26±0.47%	4.99%
SFP	Resnet56	Cifar10	Batch size = 128 Pruning rate = 30% Epoch = 200 Learning rate = 0.01	0.49±0.2%	0.33%
SFP	Resnet110	Cifar10	Batch size = 128 Pruning rate = 20% Epoch = 200 Learning rate = 0.01	-0.25±0.41%	-0.02%
PfS	VGG19	Cifar10	Batch size = 128 Sparsity ratio = 0.52 Epoch = 200 Learning rate = 0.01	-0.31±0.08%	-0.25%

Table 1. Selected measurement results reproducing the original article’s performance evaluation.

References

[1] S. A. Aziez, N. Al-Hemeary, A. H. Reja, T. Zsedrovits, and G. Cserey, “Using KNN Algorithm Predictor for Data Synchronization of Ultra-Tight GNSS/INS Integration,” *Electronics (Basel)*, vol. 10, no. 13, p. 1513, Jun. 2021, doi: 10.3390/electronics10131513. [Online]. Available: https://www.mdpi.com/2079-9292/10/13/1513

[2] A. Ess, B. Leibe, K. Schindler, and L. van Gool, “A mobile vision system for robust multi-person tracking,” *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008, doi: 10.1109/CVPR.2008.4587581.

[3] H. Laczko, B. Janossy, and T. Zsedrovits, “Towards 3D Cave Mapping with UAVs,” in *2021 17th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, Sep. 2021, pp. 1–4, doi: 10.1109/CNNA49188.2021.9610810 [Online]. Available: https://ieeexplore.ieee.org/document/9610810/

[4] “LiDAR Camera L515 – Intel® RealSense™ Depth and Tracking Cameras.” [Online]. Available: https://www.intelrealsense.com/lidar-camera-l515/. [Accessed: Sep. 28, 2022]

[5] M. Petho and T. Zsedrovits, “UAV obstacle detection with bio-motivated computer vision,” in *2021 17th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA)*, Sep. 2021, pp. 1–4, doi: 10.1109/CNNA49188.2021.9610823 [Online]. Available: https://ieeexplore.ieee.org/document/9610823/

[6] M. Petho, A. Nagy, and T. Zsedrovits, “A bio-motivated vision system and artificial neural network for autonomous UAV obstacle avoidance,” in *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Dec. 2020, doi: 10.1109/ISRITI51436.2020.9315423.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, vol. 2016-Decem, pp. 779–788, doi: 10.1109/CVPR.2016.91 [Online]. Available: http://ieeexplore.ieee.org/document/7780460/

[8] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” vol. 9351, 2015, pp. 234–241 [Online]. Available: http://link.springer.com/10.1007/978-3-319-24574-4_28

[9] D. Wang, H. Wang, J. Sun, J. Xin, and Y. Luo, “Face Recognition in Complex Unconstrained Environment with An Enhanced WWN Algorithm,” *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 18–39, Jul. 2020, doi: 10.1515/jisys-2019-0114. [Online]. Available: https://doi.org/10.1515/jisys-2019-0114. [Accessed: Aug. 11, 2020]

[10] F. Chaumette and S. Hutchinson, “Visual servo control. II. Advanced approaches [Tutorial],” *IEEE Robot Autom Mag*, vol. 14, no. 1, pp. 109–118, Mar. 2007, doi: 10.1109/MRA.2007.339609. [Online]. Available: https://ieeexplore.ieee.org/document/4141039/. [Accessed: Sep. 28, 2022]

[11] "22nd Conference on Artificial Intelligence for Environmental Science - 2023 AMS Annual Meeting." [Online]. Available: https://annual.ametsoc.org/index.cfm/2023/program-events/conferences-and-symposia/22nd-conference-on-artificial-intelligence-for-environmental-science/. [Accessed: Sep. 29, 2022]

[12] H. J. S. Fernando *et al.*, "The MATERHORN: Unraveling the Intricacies of Mountain Weather," *Bull Am Meteorol Soc*, vol. 96, no. 11, pp. 1945–1967, Nov. 2015, doi: 10.1175/BAMS-D-13-00131.1. [Online]. Available: https://journals.ametsoc.org/doi/10.1175/BAMS-D-13-00131.1. [Accessed: Sep. 29, 2022]

[13] "MATERHORN-X | Earth Observing Laboratory." [Online]. Available: https://www.eol.ucar.edu/field_projects/materhorn-x. [Accessed: Sep. 29, 2022]

[14] S. Berkovic, "Winter wind regimes over Israel using self-organizing maps," *J Appl Meteorol Climatol*, vol. 56, no. 10, pp. 2671–2691, 2017, doi: 10.1175/JAMC-D-16-0381.1.

[15] R. Chávez-Arroyo, S. Lozano-Galiana, J. Sanz-Rodrigo, and O. Probst, "Statistical-dynamical downscaling of wind fields using self-organizing maps," *Appl Therm Eng*, vol. 75, pp. 1201–1209, 2015, doi: 10.1016/j.applthermaleng.2014.03.002. [Online]. Available: http://dx.doi.org/10.1016/j.applthermaleng.2014.03.002

[16] J. C. Espinoza, M. Lengaigne, J. Ronchail, and S. Janicot, "Large-scale circulation patterns and related rainfall in the Amazon Basin: A neuronal networks approach," *Clim Dyn*, vol. 38, no. 1–2, pp. 121–140, 2012, doi: 10.1007/s00382-011-1010-8.

[17] H. Kalinic *et al.*, "Comparison of two meteorological models using self-organizing maps," *MTS/IEEE OCEANS 2015 - Genova: Discovering Sustainable Ocean Energy for a New World*, no. ii, pp. 5–10, 2015, doi: 10.1109/OCEANS-Genova.2015.7271374.

[18] F. Matić, H. Kalinić, and I. Vilibić, "Interpreting Self-Organizing Map errors in the classification of ocean patterns," *Comput Geosci*, vol. 119, no. January, pp. 9–17, 2018, doi: 10.1016/j.cageo.2018.06.006.

[19] Y. Bernard, N. Hueber, and B. Girau, "Novelty Detection with Self-Organizing Maps for Autonomous Extraction of Salient Tracking Features," *Advances in Intelligent Systems and Computing*, vol. 976, pp. 100–109, 2020, doi: 10.1007/978-3-030-19642-4_10.

[20] R. M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, 1984, doi: 10.1109/MASSP.1984.1162229.

[21] E. Holzbecher, "Streamfunction and Complex Potential BT - Environmental Modeling: Using MATLAB," E. Holzbecher, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 281–301 [Online]. Available: https://doi.org/10.1007/978-3-642-22042-5_15

[22] "Self-organizing map - MATLAB selforgmap." [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/selforgmap.html. [Accessed: Sep. 29, 2022]

[23] N. Lee, T. Ajanthan, and P. H. S. Torr, "SNIP: Single-shot Network Pruning based on Connection Sensitivity," *Iclr*, Oct. 2018 [Online]. Available: http://arxiv.org/abs/1810.02340v2

[24] "GitHub - namhoonlee/snip-public: SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY." [Online]. Available: https://github.com/namhoonlee/snip-public. [Accessed: Sep. 29, 2022]

[25] C. Wang, G. Zhang, and R. Grosse, "Picking Winning Tickets Before Training by Preserving Gradient Flow," Feb. 2020, doi: 10.48550/arxiv.2002.07376. [Online]. Available: https://arxiv.org/abs/2002.07376v2. [Accessed: Sep. 29, 2022]

[26] "GitHub - alecwangcq/GraSP: Code for 'Picking Winning Tickets Before Training by Preserving Gradient Flow' https://openreview.net/pdf?id=SkgsACVKPH." [Online]. Available: https://github.com/alecwangcq/GraSP. [Accessed: Sep. 29, 2022]

[27] Y. He, X. Dong, G. Kang, Y. Fu, C. Yan, and Y. Yang, "Asymptotic Soft Filter Pruning for Deep Convolutional Neural Networks," *IEEE Trans Cybern*, vol. 50, no. 8, pp. 3594–3604, Aug. 2020, doi: 10.1109/TCYB.2019.2933477. [Online]. Available: https://github.com/he-y/soft-filter-pruning. [Accessed: Sep. 29, 2022]

[28] "GitHub - he-y/soft-filter-pruning: Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks." [Online]. Available: https://github.com/he-y/soft-filter-pruning. [Accessed: Sep. 29, 2022]

[29] Y. Wang *et al.*, "Pruning from Scratch," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 12273–12280, Sep. 2019, doi: 10.1609/aaai.v34i07.6910. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/6910. [Accessed: Sep. 29, 2022]

[30] "GitHub - zheng-ningxin/Pruning-from-scratch." [Online]. Available: https://github.com/zheng-ningxin/Pruning-from-scratch. [Accessed: Sep. 29, 2022]