Model Pruning in Depth Completion CNNs for Forestry Robotics with Simulated Annealing

M. Eduarda Andrada[1], Joao F. Ferreira[1,3], George Kantor[4], David Portugal[1], Carlos Henggeler Antunes[2]

***Abstract*— In this article, we present an analysis of model compression in depth completion neural networks for forestry robotics, considering the increasing demands of real time autonomous solutions. Specifically, we implement a single state simulated annealing meta-heuristic for model pruning in the ENet and MSG-CHN neural networks for depth completion. We run experiments in three different datasets and analyze how different levels of pruning affect the accuracy and speed of the models. Experimental tests show that increasing sparsity has different effects depending on the neural network and dataset. ENet has negligible difference in accuracy and it would greatly benefit from lowering the amount of FLOPs, while MSG-CHN displays an inconsistent behavior depending on the dataset. This suggests that while both models benefit from model compression techniques, the optimal sparsity level depends on environment, dataset and neural network.**

I. Introduction

Despite the advances in robotics and computer vision, there are still no fully autonomous robots in forestry environments. While artificial perception has been studied extensively in natural environments (e.g., [1], [2], [3]), a vast number of issues that demand robust solutions have not been developed thus far such as perceiving the full environment, for example. Depth completion has received significant recognition, specifically for fusing Light Detection And Ranging (LiDAR) and camera sensors (Figure 1). Knowing the surrounding depth is crucial for an unmanned ground vehicle (UGV) to perceive and move into the world. A popular example of a widely used dataset for benchmark regarding depth extrapolation for 3D LiDARs and red, green and blue channels (RGB) cameras has been created by the Karlsruhe Institute of Technology (KIT), called KITTI [4].

Even though simple computer vision techniques and depth only neural networks achieve sufficient results, newer techniques, such as [5], [6], and [7], project significantly better outcomes by using multimodal Convolutional Neural Networks (CNNs) of a depth map and RGB images to identify the missing points as in Figure 2.

*This work has been supported by a CMU Portugal Ph.D. grant (ref. PRT/BD/152194/2021) from the Portuguese Foundation for Science and Technology (FCT), and the Safety, Exploration and Maintenance of Forests with Ecological Robotics (SEMFIRE, ref. CENTRO-01-0247-FEDER-03269) research project co-funded by the "Agência Nacional de Inovação" within the Portugal2020 programme.

[1]Institute of Systems and Robotics, University of Coimbra, Portugal. {duda.andrada, jfilipe, davidbsp}@isr.uc.pt

[2]INESC Coimbra, Department of Electrical and Computer Engineering, University of Coimbra, Portugal. ch@deec.uc.pt

[3]School of Science and Technology, Nottingham Trent University, Nottingham, UK. joao.ferreira@ntu.ac.uk

[4]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, U.S.A. gkantor@andrew.cmu.edu

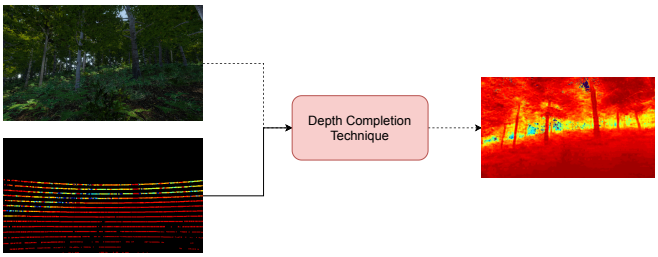

Fig. 1. Depth completion process using a sparse depth map and an optional RGB image.

Depth completion multi-modal neural networks (NNs) in particular achieve promising results by utilizing the sparse depth image and the equivalent RGB frame to extrapolate the remaining pixels according to objects and their boundaries [6]. However, it is a computationally costly process that requires lighter options to guarantee real-time performance [8]. Therefore, optimization techniques, such as model compression, are essential for real time operations.

With the increasing number of studies involving meta-heuristics over the last decade, they are a promising choice for NN optimization, a topic which has been receiving significant attention nowadays [9]. Meta-heuristics provide adequate solutions to general combinatorial problems that do not have algorithms with guaranteed performance, such as multi-level graph partitioning [10]. They have been recently used for automating NN design in neural architecture search [11]; knowledge distillation [12] and model pruning [13].

As seen in Figure 3, model pruning is a technique that analyzes which parameters could be removed from the NN, while maintaining a balance between speed and accuracy. Works such as [14], [15] and [16] use learn based methods to achieve high model compression by reducing the number of weights or parameters. Consequently, lowering the number of parameters is expected to reduce floating point operations (FLOPs), a common measure of GPU performance [14]. In the realm of CNN model compression, this would lower inference time and contribute to the real time application of depth completion CNN-based methods.

This work presents a proof-of-concept study to showcase the performance of a simulated annealing optimization technique for model pruning in depth completion CNNs. Our goal is to assess its impact on depth completion performance using two different CNNs and three distinct datasets.

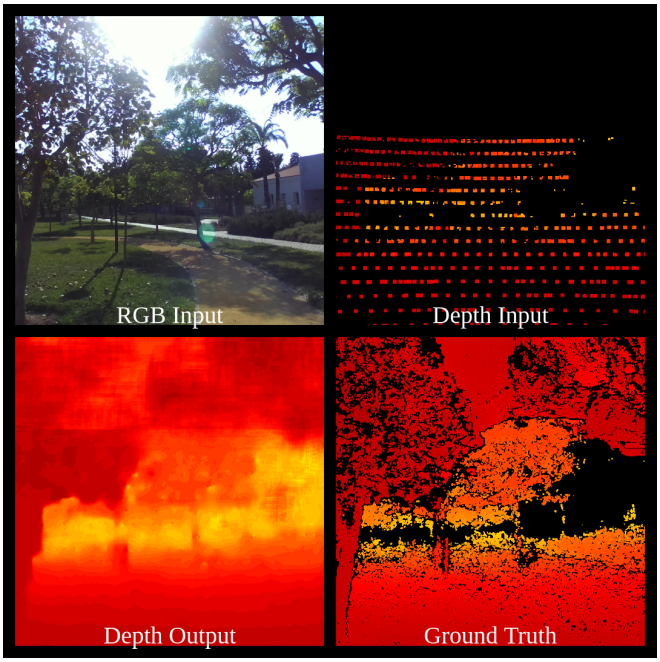

Fig. 2. Example of depth completion image from UASOL [17] dataset.

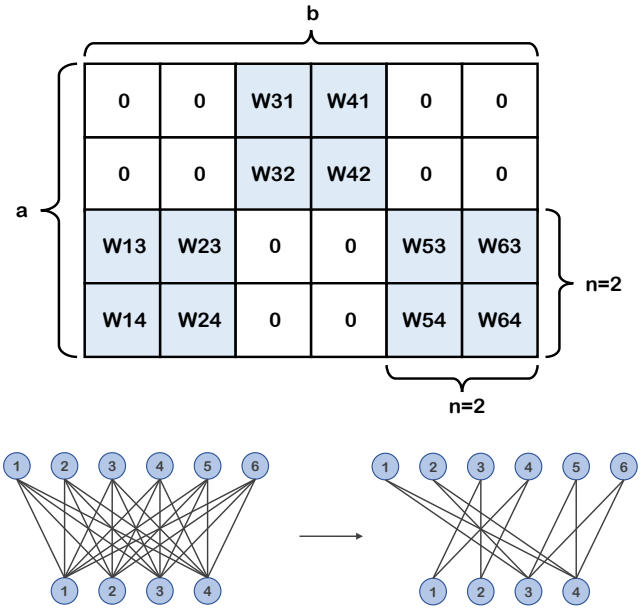

Fig. 3. Model pruning concept in convolutional neural networks. Note that the 0s in the matrix are tentative parameters for removal. Adapted from [18].

II. Proposed Approach

The neural networks used in this work were selected based on their performance in the KITTI depth completion benchmark, widely used for outdoor urban environments [4]. While not ideal, this is the closest ranked dataset that can be adapted to a forestry environment due to its popularity, number of publications and models based on it, and the sensors are the most similar to our projects' needs. After analyzing which NNs would best fit our case study, ENet [8] and multi-scale guided cascade hourglass network (MSG-CHN) [5] were selected as each provides a needed characteristic according to our design requirements.

ENet, as seen in Figure 4, is a robust and highly layered CNN. It uses RGB and depth images as input to output an extrapolated depth map in a color-dominant branch before inputting the same sparse depth and this new map into a depth-dominant branch to output a more comprehensive solution. Due to the nature of this model, it has over 120 convolutional layers and it is one of the best performing model with code available at the benchmark, with an RMSE value of 741.3 mm. However, it has a high inference time because of the number of FLOPs, 350 GFLOPs, that affects real time operations. For reference, the popular model Resnet-50 [19] uses 7 GFLOPs.

In contrast, MSG-CHN (see Figure 5) is a lightweight model with 45 convolutional layers in total. It uses multiple depth and RGB image sizes in three different encoders and decoders to achieve its final output, as shown in Figure 6. Although it is not as accurate as ENet, with current benchmark Root Mean Square Error (RMSE) of 762.2 mm, its number of FLOPs is 31 GFLOPs, an eleventh of ENet's.

We chose to test the performance on three datasets, UASOL garden [17], KITTI [4] and SynPhoRest [20], based on the quality, diversity and number of images they contain.

UASOL is a high-resolution dataset recorded in the University of Alicante. It includes both RGB and full depth map of the outdoor environment in the camera frame, which has a great amount of vegetation while also including some urban elements such as surrounding buildings. Since it includes forestry elements and high quality images, this dataset is ideal to analyze model compression behavior.

SynPhoRest is a synthetic dataset with 480p RGB and fully complete depth map of a dense Portuguese forest simulation. It comprises all the forestry elements expected to truly autonomously navigate a UGV. Moreover, it allows for a comparison test between real versus synthetic images and how these neural networks generalize them. Therefore, a dataset comprising both forestry environment and complete synthetic images provides great insight on model pruning effectiveness.

Lastly, KITTI is an autonomous car dataset from an urban environment. It consists of RGB and LiDAR depth map images from an automotive perspective. Although it does not encompass many forestry characteristics, it is an adequate basis of comparison considering that both NNs were modeled to achieve best results in its surroundings.

Overall, each dataset has distinct features that affect different parameters in a model. Thus, they allow for a robust and comprehensive analysis of model compression in depth completion neural networks.

The model pruning approach used in the analysis is performed by the Simulated Annealing (SA) meta-heuristic. SA is a technique inspired on the tempering process which slowly cools the metal until it reaches a final stable temperature. The use of SA in neural networks is enhanced to allow for different sparsity when compressing the model.

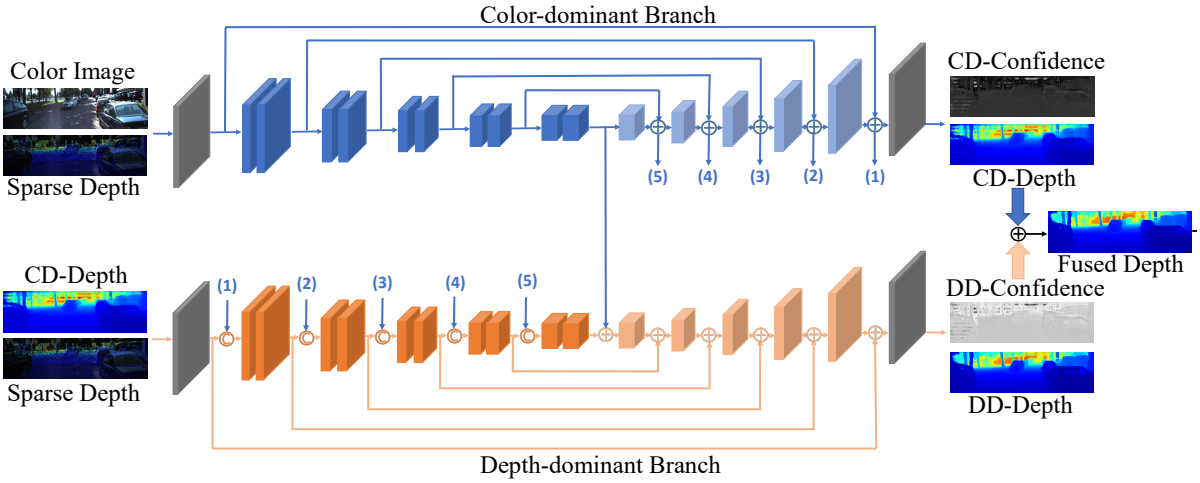

Fig. 4. ENet's two sets of decoder and encoders with 120 convolutional layers in total. Reproduced from [8].

The general structure of SA consists of starting the optimization process with a solution, which can be randomly generated, with an initial value for the temperature control parameter. In each iteration, a new solution within a neighborhood of the current one competes with the latter. If the new solution is better than the current one, the algorithm replaces the current one with the new one. Otherwise, if the new solution is worse, it can still replace it according to a probability acceptance function, which depends on the temperature and the difference of the evaluation function value for the two competing solutions, in an attempt to avoid the search process to be trapped in local optima. As the temperature decreases until a pre-determined final temperature value, it becomes more difficult that a new solution which is worse than the current one replaces the latter.

The SA version in Algorithm 1, adopts guided search based on prior experience, which removes more weights from layers with the most parameters as it affects the overall accuracy the least [21]. Alg. 1 searches for action by checking if the change of weight pruning rates in the layers, which defines the neighborhood, is accepted according to the SA scheme. The evaluation function is the RMSE. If the new weight values lead to better results, the new model is accepted and the temperature is lowered. Otherwise the change is accepted according to the probability acceptance function. When the final temperature is reached, the algorithm returns the best model pruned as the final solution.

The initial values for minimum, maximum temperature and cool down rate shown in Algorithm 1 ensure an appropriate compromise between number of iterations and results, as a higher number of cycles did not lead to any improvement in results. C_t represents the sparsity level that we intend to achieve in each training. In this experiment, it was set between 10% and 75%.

To ensure consistency and fair analysis of the results, two evaluation metrics were considered: floating points operations (FLOPs) and root mean squared error (RMSE). FLOPs

Algorithm 1 Simulated Annealing Pruning Algorithm where $model_{initial}$ is the original CNN model, C_t is overall pruning rate based on weight number, $perturbation$ is the change of weight pruning rates, $T_{max} = 100°C$, $T_{min} = 20°C$, and $\eta = 0.9$.

$INIT(model_{initial}, T_{max}, T_{min}, C_t, \eta)$
$model_{best} \leftarrow model_{inicial}$
$RMSE_{best} \leftarrow EVALUATION(model_{best})$
$T \leftarrow T_{max}$
while $T > T_{min}$ **do**
    Generate new model($model_{next}$) based on
$perturbation$ values.
    $RMSE_{next} \leftarrow EVALUATION(model_{next})$
    if $RMSE_{next} > RMSE_{best}$ **then**
        $Model_{best} \leftarrow Model_{next}$
    else
        $\Delta RMSE \leftarrow RMSE_{best} - RMSE_{next}$
        if $exp(\frac{\Delta RMSE}{T}) > random[0,1)$ **then**
            $Model_{best} \leftarrow Model_{next}$
        end if
    end if
    $T \leftarrow \eta * T$
$Model_{final} \leftarrow Model_{best}$

is a GPU performance metric which is ideal due to different graphics card available in the UGV and the laboratory. Therefore, it allows for easily replicability with different computer specifications. RMSE is a metric that analyzes the similarity between the ground truth and the prediction, the benchmark basis for depth completion defined in Equation 1. In our case, we compare the sparsity value for each increment with the original model values for RMSE, named $\Delta RMSE$.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2} \quad (1)$$

For this experiment, both neural networks were tested

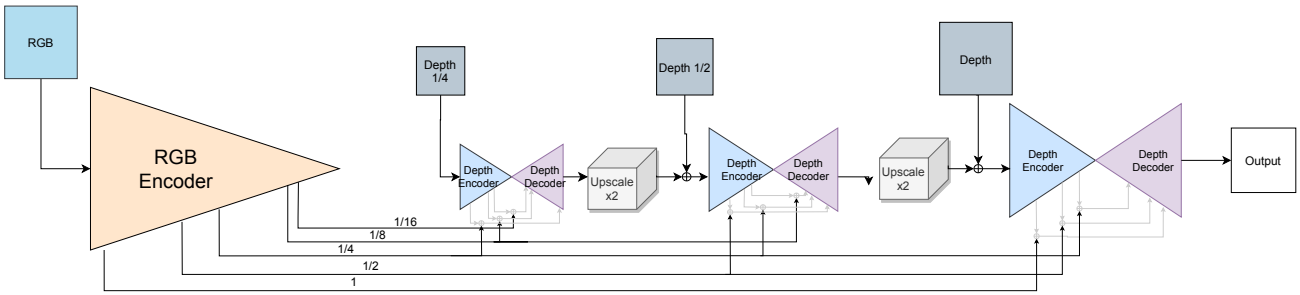

Fig. 5. MSG-CHN design with four encoders and three decoders with a total of 45 convolutional layers. Reproduced from [5].

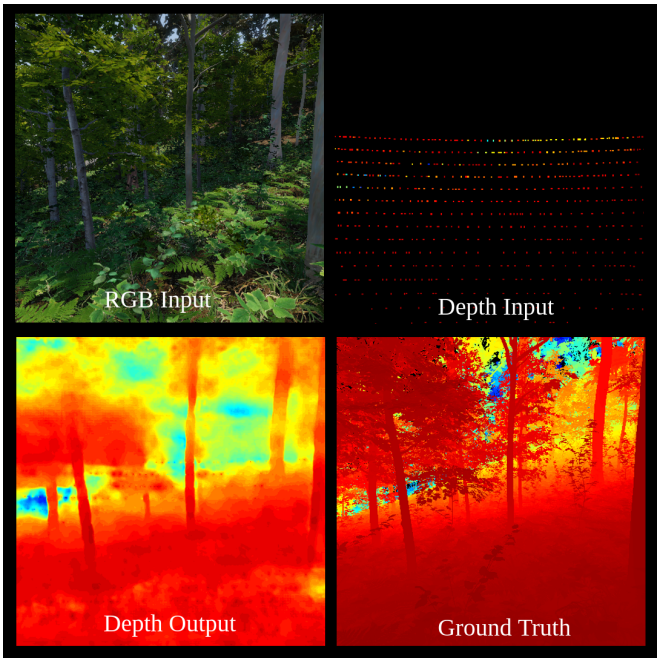

Fig. 6. MSG-CHN technique applied on a forestry synthetic dataset.

and analyzed in three different datasets, UASOL, KITTI and SynPhoRest, and evaluated with FLOPs and RMSE metrics at 5% sparsity increments with a sparsity range of 10% and 75%. Moreover, ENet and MSG-CHN were trained with the same hyperparameters and number of epochs to guarantee reliability and experiment reproducibility for each compression step.

III. Results & Discussion

Table I shows the FLOPs and RMSE changes from baseline values for each sparsity increment resulting from the ENet model pruning process. Figure 7 shows a visual representation of RMSE with respect to sparsity for each dataset. The RMSE values were an unexpected result from our preliminary assumption, which considered that lowering the parameters model would decrease the accuracy. All three datasets had only a marginal increase in error at the highest sparsity, 0.75, with the highest value of 0.4% at KITTI.

At 0.1 sparsity the behavior was inconsistent – UASOL and KITTI improved results while SynPhoRest worsen them significantly, increasing RMSE by 225.4%. This could indicate that at 10% compression, the resulting perturbation is highly unstable and the removed parameters affect prediction in unforeseeable ways.

Under those circumstances and taking into consideration our design requirements, the ENet model highly benefits of trimming redundant or unnecessary parameters and it can lower FLOPs by half and still maintain the same accuracy quality.

Table II and Figure 8 show the MSG-CHN model exhibits a behavior substantially different from the ENet model. Each dataset has a highly different performance when increasing sparsity. UASOL achieves the first and second best RMSE results at 0.75 and 0.7, respectively, while reducing FLOPs by 85%. The model pruning in the KITTI lowers the error by 64.9% at 0.1 sparsity and 4.8% at 0.65 sparsity. On the other hand, SynPhoRest increases the error with higher sparsity, with 46.6% increase at 75%, almost 5 times higher than at 65%. This difference between datasets shows that a lighter model such as MSG-CHN is already more optimized than ENet and the stochastic choice of parameter removal creates an random behavior in the model accuracy.

Nevertheless, given that at 0.6 sparsity there is a small increase of 4.5% of error in the SynPhoRest dataset while lowering both UASOL and KITTI by -8.2% and -4.5% respectively, it is possible to optimize the MSG-CHN with a 71% FLOPs reduction, to 9.43 GFLOPs and still achieve positive outcome.

Both neural network models greatly benefited from pruning and man-made design appears to be resource wasteful even when achieving the best results in the KITTI benchmark.

IV. Conclusion

Overall, techniques for model pruning were effective in optimizing inference for real time applications. SA Model compression has shown to improve the model efficiency without increasing the error significantly in both neural networks.

ENet had many redundant parameters that can easily be removed with model pruning without any considerable increase in RMSE. Although MSG-CHN is a smaller model

TABLE I

SIMULATED ANNEALING RESULTS ON ENET IN THE THREE DATASETS, UASOL, SYNPHOREST AND KITTI FOR 15 SPARSITY INCREMENTS.

	UASOL		SynPhoRest		KITTI	
Sparsity	$\Delta FLOPs$	$\Delta RMSE$	$\Delta FLOPs$	$\Delta RMSE$	$\Delta FLOPs$	$\Delta RMSE$
0.1	-5.1%	-14.3%	-5.5%	+225.4%	-6.0%	**-56.4%**
0.15	-9.8%	+0.0%	-8.4%	+0.0%	-9.1%	+0.0%
0.2	-9.8%	+0.0%	-12.1%	+0.0%	-11.7%	+0.0%
0.25	-15.4%	+0.0%	-16.0%	+0.0%	-13.9%	+0.0%
0.3	-20.2%	+0.0%	-17.7%	+0.0%	-17.7%	+0.0%
0.35	-21.6%	+0.0%	-22.1%	+0.0%	-20.5%	+0.0%
0.4	-25.8%	+0.0%	-23.6%	**+0.0%**	-26.1%	+0.1%
0.45	-28.6%	+0.0%	-30.0%	+0.1%	-26.7%	+0.1%
0.5	-31.1%	+0.0%	-31.0%	+0.1%	-30.2%	+0.1%
0.55	-37.3%	+0.0%	-35.3%	+0.1%	-34.1%	+0.1%
0.6	-36.2%	**-17.7%**	-39.7%	+0.2%	-39.1%	+0.2%
0.65	-42.9%	+0.0%	-40.2%	+0.2%	-41.0%	+0.3%
0.7	-42.4%	+0.0%	-48.4%	+0.6%	-43.7%	+0.5%
0.75	-48.3%	+0.0%	-48.2%	+0.3%	-46.6%	+0.4%

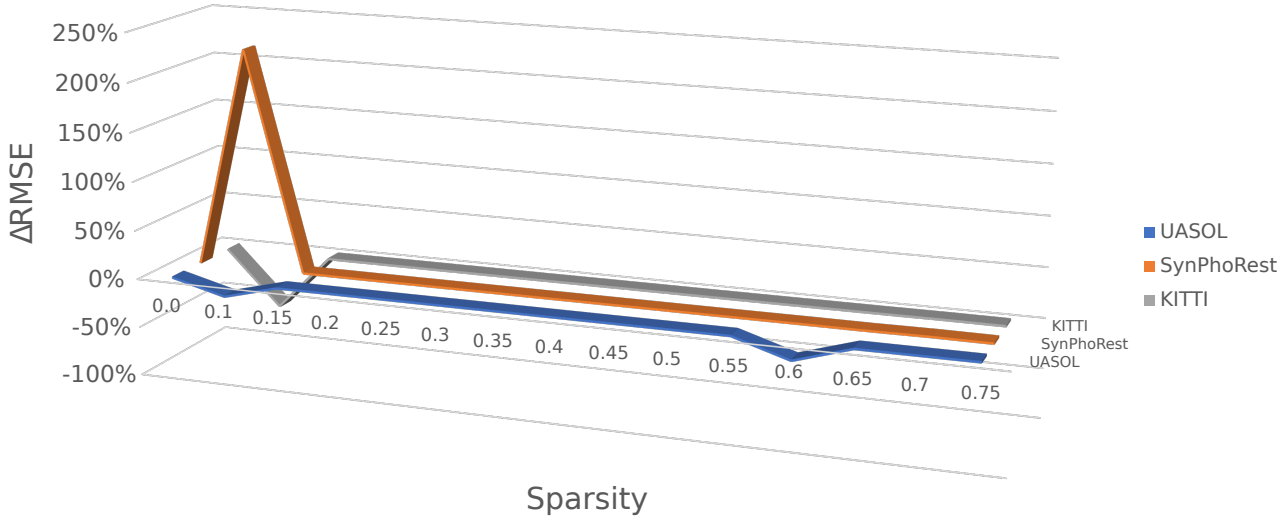

Fig. 7. ENet model compression graph which shows $\Delta RMSE$, difference between current sparsity RMSE and baseline value, on 3 different datasets.

and it had inconsistent performance between datasets while compressing, it is still possible to increase sparsity by 0.6 which lowers FLOPs by 71% with almost no effect in accuracy.

In future work, we will compare the simulated annealing model pruning with other popular compression methods and introduce data augmentation and optimization, such as fine-tuning, to assess whether it is possible to improve even further the resulting compressed models.

REFERENCES

[1] C. Thorpe and H. Durrant-Whyte, "Field robots," in *Proceedings of the 10th International Symposium of Robotics Research (ISRR'01)*, 2001.

[2] A. Kelly *et al.*, "Toward Reliable Off Road Autonomous Vehicles Operating in Challenging Environments," *International Journal of Robotics Research*, vol. 25, no. 5–6, pp. 449–483, 2006.

[3] S. Lowry and M. Milford, "Supervised and unsupervised linear learning techniques for visual place recognition in changing environments," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 600–613, 2016.

[4] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *International Conference on 3D Vision (3DV)*, 2017.

[5] A. Li, Z. Yuan, Y. Ling, W. Chi, C. Zhang *et al.*, "A multi-scale guided cascade hourglass network for depth completion," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 32–40.

[6] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy lidar completion with rgb guidance and uncertainty," in *2019 16th International Conference on Machine Vision Applications (MVA)*. IEEE, 2019, pp. 1–6.

[7] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, "Non-local spatial propagation network for depth completion," in *Proc. of European Conference on Computer Vision (ECCV)*, 2020.

[8] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "Towards precise and efficient image guided depth completion," in *ICRA*, 2021.

[9] K. Hussain, M. N. M. Salleh, S. Cheng, and Y. Shi, "Metaheuristic research: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2191–2233, 2019.

[10] D. Portugal and R. P. Rocha, "Partitioning generic graphs into k balanced subgraphs," in *Proceedings of the 6th Iberian Congress On Numerical Methods in Engineering (CMNE 2011), Coimbra, Portugal*, 2011, pp. 13–16.

[11] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan, "A

TABLE II

SIMULATED ANNEALING RESULTS ON MSG-CHN IN THE THREE DATASETS, UASOL, SYNPHOREST AND KITTI FOR 15 SPARSITY INCREMENTS.

	UASOL		SynPhoRest		KITTI	
Sparsity	$\Delta FLOPs$	$\Delta RMSE$	$\Delta FLOPs$	$\Delta RMSE$	$\Delta FLOPs$	$\Delta RMSE$
0.1	-17.6%	-11.2%	-17.4%	+113.3%	-16.7%	**-64.9%**
0.15	-21.4%	+0.3%	-24.5%	+0.0%	-23.5%	-0.2%
0.2	-32.2%	+0.5%	-30.9%	**+0.0%**	-37.3%	+1.2%
0.25	-35.8%	+0.2%	-36.1%	+0.4%	-43.7%	+1.0%
0.3	-42.6%	-0.1%	-46.2%	+0.4%	-48.4%	+2.5%
0.35	-53.8%	-0.9%	-44.6%	+0.6%	-51.8%	+1.1%
0.4	-53.9%	-1.8%	-58.7%	+0.6%	-58.3%	-1.7%
0.45	-63.1%	-3.2%	-60.3%	+1.2%	-62.7%	-4.4%
0.5	-63.9%	-4.9%	-63.1%	+1.8%	-63.8%	-4.8%
0.55	-69.6%	-6.3%	-67.0%	+2.3%	-69.2%	-4.8%
0.6	-70.4%	-8.2%	-71.1%	+4.5%	-72.5%	-4.3%
0.65	-77.7%	-10.5%	-74.2%	+9.6%	-74.9%	+1.6%
0.7	-78.4%	**-21.3%**	-79.0%	+32.6%	-77.2%	+2.5%
0.75	-84.3%	-21.2%	-80.5%	+46.6%	-80.3%	+1.3%

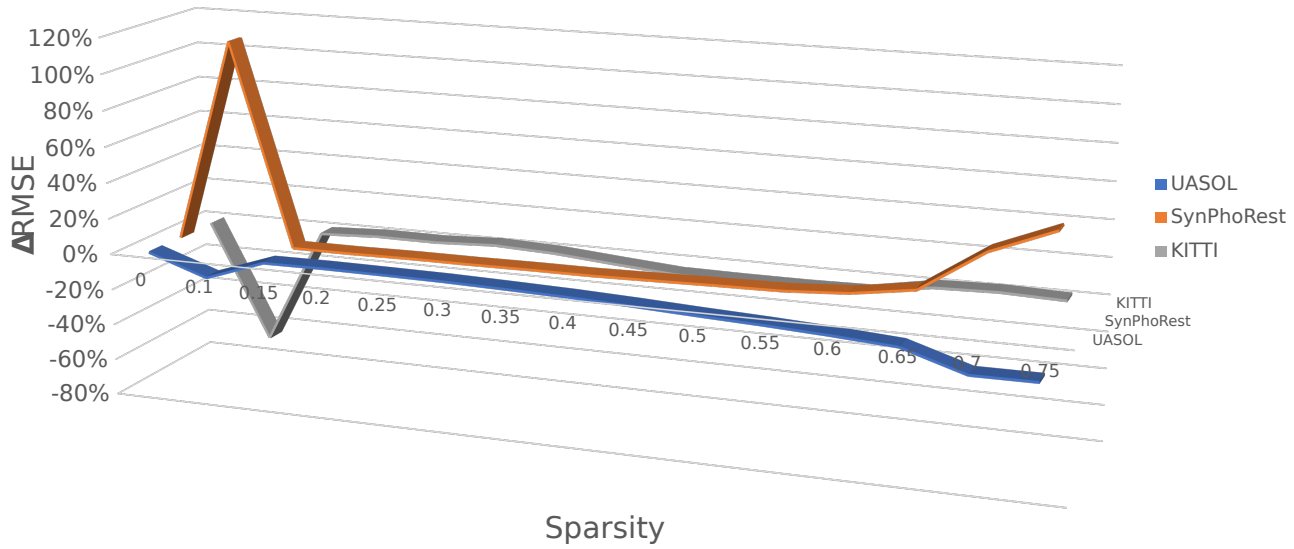

Fig. 8. MSG-CHN model compression graph which shows $\Delta RMSE$, difference between current sparsity RMSE and baseline value, on 3 different datasets.

survey on evolutionary neural architecture search," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[12] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, 2021.

[13] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang, "Pruning and quantization for deep neural network acceleration: A survey," *Neurocomputing*, vol. 461, pp. 370–403, 2021.

[14] Y. He, J. Lin, Z. Liu, H. Wang, L.-J. Li, and S. Han, "Amc: Automl for model compression and acceleration on mobile devices," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 784–800.

[15] T. Zhang, S. Ye, K. Zhang, J. Tang, W. Wen, M. Fardad, and Y. Wang, "A systematic DNN weight pruning framework using alternating direction method of multipliers," *CoRR*, vol. abs/1804.03294, 2018.

[16] T. Yang, A. G. Howard, B. Chen, X. Zhang, A. Go, V. Sze, and H. Adam, "Netadapt: Platform-aware neural network adaptation for mobile applications," *CoRR*, vol. abs/1804.03230, 2018.

[17] Z. Bauer, F. Gomez-Donoso, E. Cruz, S. Orts-Escolano, and M. Cazorla, "Uasol, a large-scale high-resolution outdoor stereo dataset," *Scientific data*, vol. 6, no. 1, pp. 1–14, 2019.

[18] L. Huang, J. Zeng, S. Sun, W. Wang, Y. Wang, and K. Wang, "Coarse-grained pruning of neural network models based on blocky sparse structure," *Entropy*, vol. 23, no. 8, p. 1042, 2021.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[20] R. Nunes, J. Ferreira, and P. Peixoto, "SynPhoRest - Synthetic Photorealistic Forest Dataset with Depth Information for Machine Learning Model Training," Mar. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6369445

[21] N. Liu, X. Ma, Z. Xu, Y. Wang, J. Tang, and J. Ye, "Autocompress: An automatic dnn structured pruning framework for ultra-high compression rates," 2019.