AgriAdapt: Towards Resource-Efficient UAV Weed Detection using Adaptable Deep Learning

Octavian M. Machidon
Faculty of Computer and Information Science, University of Ljubljana
Slovenia
octavian.machidon@fri.uni-lj.si

Andraž Krašovec
Faculty of Computer and Information Science, University of Ljubljana
Slovenia
ak6688@student.uni-lj.si

Alina L. Machidon
Faculty of Computer and Information Science, University of Ljubljana
Slovenia
alina.machidon@fri.uni-lj.si

Veljko Pejović
Faculty of Computer and Information Science, University of Ljubljana
Slovenia
veljko.pejovic@fri.uni-lj.si

Daniele Latini
GEO-K s.r.l., Rome
Italy
daniele.latini@geo-k.co

Sarathchandrakumar T. Sasidharan
"Tor Vergata" University of Rome, DICII, Rome
Italy
chandubajaj095@gmail.com

Fabio Del Frate
"Tor Vergata" University of Rome, DICII, Rome
Italy
fabio.del.frate@uniroma2.it

ABSTRACT

The 2022-2023 food crises and the ongoing human population growth make the efficient use of the available agricultural land a pressing matter. However, weeds present a major obstacle towards efficient land use, and cause up to 40% yield loss in all major crops, leading to more than $100 billion USD annual global economic loss. Camera equipped unmanned aerial vehicles (UAVs) represent an attractive tool for automatic weed detection. Yet, high computation and energy requirements of deep learning models restrict automatic real-time inference to expensive high-end UAVs, preventing wider adoption of this promising solution for weed detection.

In this work we present AgriAdapt, a solution for lightweight on-UAV weed detection that is based on novel slimmable U-Net neural architecture for weed detection. A defining property of AgriAdapt is its adaptability to operating conditions – the resource usage can be scaled dynamically according to the needs for battery preservation, input difficulty, or other factors. In this paper we present preliminary experiments on a newly-collected weeds dataset and detailed assessment of resource savings enabled by AgriAdapt.

CCS CONCEPTS

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**; • **Computer systems organization → Embedded and cyber-physical systems**; • **Computing methodologies → Neural networks**.

KEYWORDS

unmanned aerial vehicles, weed detection, slimmable neural networks

ACM Reference Format:
Octavian M. Machidon, Andraž Krašovec, Alina L. Machidon, Veljko Pejović, Daniele Latini, Sarathchandrakumar T. Sasidharan, and Fabio Del Frate. 2023. AgriAdapt: Towards Resource-Efficient UAV Weed Detection using Adaptable Deep Learning. In *The 2nd Workshop on Networked Sensing Systems for a Sustainable Society (NET4us '23), October 6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3615991.3616405

NET4us '23, October 6, 2023, Madrid, Spain

ACM ISBN 979-8-4007-0365-2/23/10…$15.00
https://doi.org/10.1145/3615991.3616405

1 INTRODUCTION

Year 2022 brought the highest worldwide increase in food prices observed in the last decades[1], with the poorest countries being hit the worst. Primarily caused by global issues, such as the COVID-19 pandemics, wars and the related disruption to supply chains, climate change, and other factors, the ongoing food crisis emphasised the need for regional self-sustainability and efficient local agriculture.

Weeds are amongst the most impacting abiotic factors causing up to 40% yield loss in all major crops, leading to more than $100 billion USD annual global economic loss [15]. Timely detection of weeds is thus of key importance. Camera-equipped unmanned aerial vehicles (UAVs) represent a promising technology upon which an economically viable, robust, and scalable solution for automatic weed detection could be built. These devices can provide close-up images of the fields, and deep neural networks (DNNs) have already been constructed to reliably detect weeds in such images [2, 9].

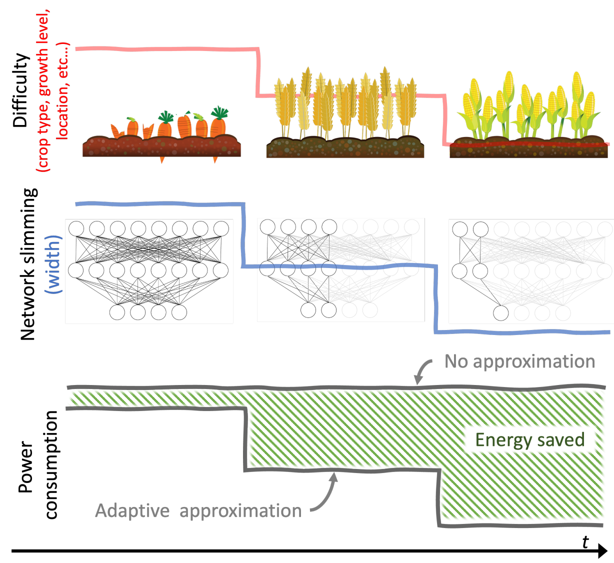
However, the full potential of computer vision for weed detection can be realized only if the processing happens directly on UAVs. This would enable UAVs to provide information for real-time location-specific actioning. For instance, a UAV could detect weeds, which an on-the-ground robot could immediately exterminate. The biggest obstacles towards the realization of the above are the limited processing capabilities and battery charge of UAVs. DNN models require substantial resources that prevent the use of on-device models on anything but the most powerful (and expensive) UAV models, which clashes with the critical need for efficient precision agriculture on small farms in the world's less developed regions.

To enable wider proliferation of UAV-based weed detection we have to surmount the following challenges. First, the standard weed recognition DNNs are often prohibitively large for on-UAV operation. Second, while one-off compression models may allow these models to run on resource-constrained devices, one-fits-all solution may lead to unneeded quality loss across heterogeneous device landscape. Finally, the exact energy savings from such devices are not known and a full system integration is yet to be demonstrated and assessed.

In AgriAdapt (Figure 1) we make the following contributions while addressing the above challenges:

- We introduce two novel resource-optimized neural network architecture that excels in weed recognition, yet remains lightweight in terms of the number of model parameters;
- We enable adaptive compressibility of the above models by casting them to the slimmable neural network framework;
- On an originally collected weeds dataset we profile computation and energy savings enabled by AgriAdapt;
- We implement a UAV prototype that hosts a slimmable weed detection framework and is able to adapt it in real time.

Figure 1: Illustration of the AgriAdapt concept: through real-time compression (slimming) of the neural network (middle) we can adapt the resources used according to the difficulty of each inference input (top) thus enabling energy savings (bottom).



This work presents our preliminary experiences developing and evaluating AgriAdapt. We observe a significant potential for resource usage reduction with AgriAdapt and we believe that the resource savings will translate to a wider availability of on-device weed detection across a range of UAVs.

2 RELATED WORK

2.1 Deep learning for weed detection.

In recent years, a number of studies have examined the possibility of weed detection from remote sensing images using deep learning (DL) [1, 6, 24]. Due to their strong, automatic feature learning capabilities, DL-based approaches outperform traditional machine learning algorithms in all automated agriculture tasks [21]. Nevertheless, one of the most important research gaps in the field of DL for weed detection, and the one that prevents the practical use and integration of DL with a real-time automated system, is the assessment

[1]https://www.fao.org/worldfoodsituation/foodpricesindex/en/

of the temporal perofrmance of on-board DL, which is addressed in only a few studies.

Chechliński *et al.* [4] implement a custom convolutional neural network architecture combining elements from four different models (U-Net, MobileNets, DenseNet and ResNet), for performing image segmentation into 3 classes: crops, weed or soil. The evaluation of this model showed a precision score between 47–67% and an inference time of 50-100 ms per frame on a Rasberry Pi 3B+ platform, for slightly different configurations of the same architecture.

Partel *et al.* [18] evaluate the performance of three different model architectures: Faster R-CNN with Resnet 50, Faster R-CNN with Resnet101, and YOLOv3 with Darknet53 for real-time weed detection on an intelligent precision spraying system. The test results showed that all three models performed well from the point of view of performance (precision and recall values). The best performing network was Faster R-CNN with Resnet 50, while YOLOv3 achieved the lowest values of all three, but still had an acceptable precision of 95%. However, the processing time evaluation showed significant differences between the three models, with YOLOv3 being more than three times faster than the other two, enabling the processing at almost 15 frames per second. Yet, this work utilized a powerful NVIDIA GTX 1070 Ti graphical processing unit (GPU) with 2432 CUDA cores and a clock frequency of 1607 MHz, which is impractical for real-world UAV-based processing.

Junior and Ulson [14] assessed four YoloV5 architectures on a custom dataset with different weed species. The mean average precision achieved was between 0.25 and 0.3 without transfer learning, and between 0.5 and 0.7 with transfer learning. The inference time for processing images of 480 × 640 pixels ranged between 0.016 and 0.125 s while the number of frames per second processed varied between 62 and 8. The running times were measured on a Samsung Odyssey laptop with GTX1050 GPU.

Liu and Bruch [16] tested the YOLOv2 model with different feature extraction layers (ResNet-50, ResNet-101, MobileNet, InceptionResnet V2, SqueezeNet, VGG16, VGG19) for identifying lettuces. The mean average precision achieved by the various configurations of the YOLOv2 model varied between 0.5 (for the InceptionResnetV2) and 0.9 (for the VGG19). The authors report that the NVIDIA Jetson TX2 computer on which the models are deployed can process an image in around 30 ms, however no actual measurements for the actual inference times of the models are provided.

2.2 Deep learning compression techniques.

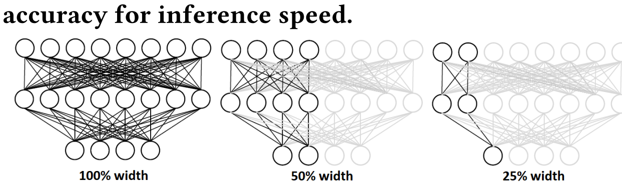
Various model compression techniques may be harnessed to reduce the complexity of DNNs, and thus, improve the inference time. These techniques deal with reducing the size and/or the computational burden of deep learning models for deployment in resource constrained environments, without significantly sacrificing accuracy. Some of the early neural network compression methods such as knowledge distillation [11], weight pruning [10], sharing [10] or quantization [8], allowed achieving only one compressed version of the bigger DNN with often permanently reduced inference accuracy. However, dynamic real-world scenarios may benefit from alternating between a more compressed network (e.g. when the battery level is low) and a more powerful one (e.g. when the input data is particularly difficult to classify). Hence, the need arises for adaptable compression techniques where several levels of compression can be supported within a single solution, so that a suitable performance-complexity trade-off can be struck at the runtime. Such dynamic compression techniques include dynamic network parameter pruning [7], dynamic parameter quantization [25], and early exiting [22], to name a few.

3 SLIMMABLE NEURAL NETWORKS FOR WEED DETECTION

3.1 Slimmable Neural Networks preliminaries

Slimmable Neural Networks (SNN) [27] exploit the observation that an increase or decrease in the number of DNN parameters disproportionally impacts the classification accuracy and that a graceful degradation of inference accuracy can be achieved with a significant reduction in the number of parameters. The SNN approach enables a reduction in the number of active network parameters on the fly to a fraction of the network's width selected from a pre-defined subset (as illustrated in Figure 2). To avoid the need to retrain the model after each configuration change, SNNs rely on switchable batch normalization (S-BN) layers. During the training, S-BN privatizes all batch normalization layers for each network width, meaning that all (sub-)networks are jointly trained at all the different widths, ensuring that acceptable inference accuracy can be achieved as the network complexity reduction increases.

Figure 2: Illustration of the Slimmable Neural Network [27] compression technique. The network can dynamically change the percentage of parmeters (widths) at runtime, often, but not always, trading inference accuracy for inference speed.



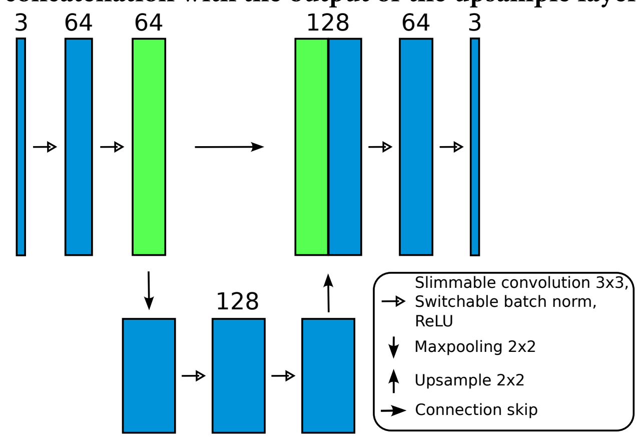
3.2 Towards dynamic neural network architecture for weed recognition

U-Net [20] is a popular network architecture often used in computer vision for segmentation tasks. U-Net is built on a fully convolutional foundation. The nearly symmetric structure of the network is divided into two main parts: the contracting path, which consists of repeated convolutions, each followed by a rectified linear unit and a max pooling, and the expansive path, which consists of a sequence of transposed 2D convolutional layers, each followed by a convolution and a concatenation with the corresponding feature map from the contracting path, plus two final convolutions, each followed by a rectified linear unit. Due to its well documented performance in different tasks, including weed recognition [9], we base our futher NN architecture development on U-Net.

Nevertheless, our goal is to bring UAV-based weed recognition to a wide range of settings, predominantly small farms, where scarce resources call for low-cost technological solutions. U-Net may incur significant computational overhead and prevent low-cost UAVs from running real-time inference onboard. Thus, we also investigate **Squeeze U-Net** [3]. This lightweight image segmentation network is built on the U-Net foundation and inspired by Squeeze Net [12]. By replacing the full convolutions in both the contracting and the expansive path of U-Net with fire modules (each consisting of point-wise convolutions followed by an inception layer comprising two parallel convolutions with different kernel sizes), Squeeze U-Net achieves a 12-fold reduction in model size and a a 3.2-fold reduction in multiply-accumulate operations (MACs) compared to U-Net.

Starting from the state-of-the-art image segmentation architecture – U-Net – and the state-of-the-art dynamic NN compression concept – Slimmable Neural Networks – we now construct **Slimmable U-Nets**. We build and train a U-Net and a Squeeze U-Net model harnessing an existing PyTorch implementation of slimmable layers provided by the authors of the original SNN paper [27]. Converting a standard U-Net into a slimmable one requires replacing the convolutional and batch normalisation layers with their slimmable counterparts. We build our slimmable U-Net accordingly. The contracting part of our network consists of two repetitions of slimmable convolution and batch normalisation layers, followed by a maxpooling layer, and ending with two additional repetitions of slimmable convolution and batch normalisation layers. On the expanding part of the model, the network consists of an upsampling layer followed by slimmable convolution and batch normalisation layer. The output is then concatenated with the output of the first pair of convolutions in the contraction part. Finally, we apply two more slimmable convolution and batch normalisation layers and finalise with a single slimmable convolution layer that yields us a segmentation mask. This network architecture is depicted in Figure 3.

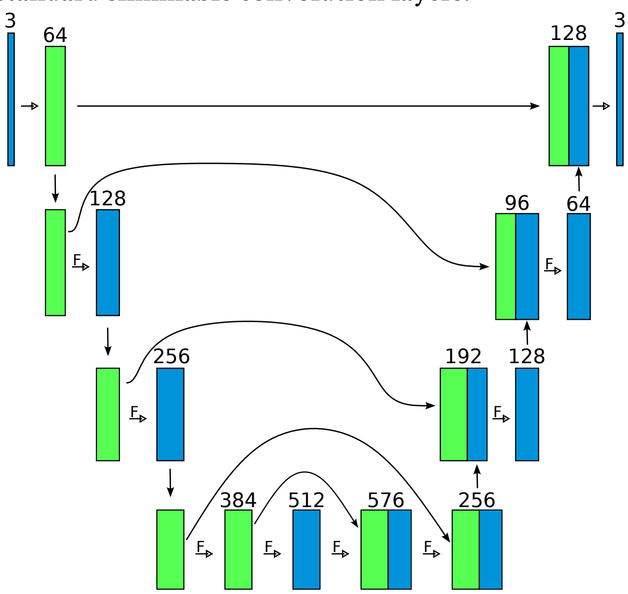
Figure 3: The Slimmable U-Net architecture, consisting of a single contraction/expansion module. The boxes represent data dimensionality as it passes through the network. Height of the box represents the image resolution, while the width represents the number of channels. Number of channels of the 100% network width is also denoted on top of the boxes. Arrows between the boxes represent the operations performed, legend for which is displayed in the bottom right corner of the figure. Green boxes represent the skip connection and concatenation with the output of the upsample layer.



In addition, we produce the **Slimmable Squeeze U-Net** by replacing the convolution, batch normalisation, and the transposed convolution layers with their slimmable counterparts. Unlike the convolution and batch normalisation layers, we implement the transposed convolution layer by applying the *conv_transpose2d* function to the varying number of filters based on the current network width. In the contracting part of the network we employ two repetitions of two fire modules, followed by a maxpooling layer and finishing off with four additional fire modules. The expanding part consists of four repetitions of transposed convolution, followed by concatenation with the appropriate output from the contracting part, as per U-Net paradigm, and a fire module. The output is then upsampled, concatenated, convoluted, upsampled, and convoluted to finally obtain the segmentation mask. The entire network architecture is illustrated in Figure 4.

We now train on a weed detection dataset (described in subsection 5.1) both the Slimmable U-Net and the Slimmable Squeeze U-Net architectures we developed on four widths: 100%, 75%, 50% and 25%. Our images are annotated for both, weeds, and lettuce, therefore, our resulting segmentation mask consists of three separate classes – weeds, lettuce, and

Figure 4: The Slimmable Squeeze U-Net architecture, consisting of three contraction/expansion modules. The boxes, box colours, and arrow symbols retain the same role as in Figure 3. Additionally, the *F* labelled arrows represent the usage of fire modules, instead of standard slimmable convolution layers.



background. We utilise a weighted cross entropy loss function to train the network. Weeds and lettuce classes have a weight of 0.45 each, while the background is set to 0.1. Increasing the weight of the background class results in everything being inferred as the background. We train our networks on 168 images, and utilise 48 different images to evaluate the network performance. Images are resized to 128x128 pixels. Each network is being trained for 1000 epoch, at a starting learning rate of 0.0001 that exponentially decreases with a gamma value of 0.99.

4 UAV SYSTEM

For the purpose of the AgriAdapt experiment we built a custom UAV system, namely a hexa-rotor with a maximum take-off weight up to 6 kg (depicted in Figure 5. It is equipped with a Plug & Play System, which allows the pilot to switch between different payloads. Real-time mission management is possible as well as the Autonomous Waypoint navigation thanks to Pixhawk PX4 flight controller, which integrates Inertial Measurement Unit (accelerometers and gyros) and Global Positioning System (GPS) receiver. The UAV has been configured to carry Nvidia Jetson Nano, including an independent battery power supply, and radio transceiver module for remote communication with the ground control machine.

The Jetson Nano is also connected to the PX4 flight controller, using a serial UART protocol to access the telemetry information. The payload consists of an autofocus camera from Arducam with Sony IMX519 sensor and 16MP of spatial resolution. In order to ensure the maximum quality of the acquisition, the camera is stabilized through a 2-axis gimbal with brushless motors.

The system is characterized by a maximum flight time of up to 20 minutes. The Italian Civil Aviation Authority (ENAC) has given the Certification of Design attesting the compliance to Italian and European (EASA) laws.

Figure 5: The UAV system designed for the AgriAdapt experiment equipped with an ArduCam camera and an NVIDIA Jetson Nano Board.

5 EVALUATION

5.1 Dataset

We validate our approach on a preliminary weed detection dataset composed of hundreds of raw UAV images [17]. This dataset was acquired using consumer segment unmanned aerial vehicles, over a test fields cultivated with various species.

As first, the dataset has been labeled in YOLO v7 PyTorch format distinguishing between plants and weeds present in the field. Auto-orientation of pixel data (with EXIF-orientation stripping) and resizing to 640x640 (using stretching) were the only pre-processing techniques applied to each of the images in the dataset.

We split the dataset into 70% training, 20% validation and 10% test images and train the Slimmable U-Net and Slimmable Squeeze U-Net architectures for the four slimmable widths described in Section 3 to perform weed detection and classification on this dataset.

5.2 Weed detection performance

For measuring the quantitative performance of the proposed network, we assessed two well-established segmentation metrics: Intersection over Union (IoU) and Precision (P). Intersection over Union measures the overlap of each prediction with the ground truth, while Precision quantifies the ability of the model to locate the relevant objects, given as the fraction of the true positive predictions from all predictions.

The mathematical expressions for these metrics are given below in Equations 2 and 1:

$$IoU = \frac{TP}{TP + FN + FP} \tag{1}$$

$$P = \frac{TP}{TP + FP} \tag{2}$$

where TP are the true positives (all correct predictions of the actual classes), FP the false positives (all negative samples incorrectly identified as positive ones), and FN the false-negatives (all positive samples that were incorrectly classified).

SNN width	SU-Net	SSU-Net
25%	50.4%	0%
50%	53.3%	39.9%
75%	54.6%	42.8%
100%	54.7%	43.5%

Table 1: Intersection over Union (IoU) for both Slimmable U-Net (SU-Net) and Slimmable Squeeze U-Net (SSU-Net) for the weed detection task.

In terms of the IoU metric, the results (Table 1) show graceful performance degradation for the top three network widths as the network becomes more slim: a 1.4% reduction between 100% width and 50% width for the full-scale U-Net and a slightly larger decrease in performance of 3.6% among the same widths for the Squeeze U-Net, with the full-scale version outperforming its more compact counterpart overall. The smallest width (25%) shows a more significant decrease in performance for the U-Net (a 4.3% reduction), and is completely unusable in the case of the squeeze U-Net. This can be explained by the already compressed architecture of this model, which consequently might not support slimming to such drastically reduced width.

With regard to the precision, the results depicted in Table 2 demonstrate more graceful performance degradation in case of the full-scale U-Net (less than 1% among the top three widths) compared to its squeezed counterpart (4.3%). The 25% width version of the Squeeze U-Net is unusable, but what is striking is that for the other top three widths this network outperforms the full-scale U-Net. This could be due

SNN width	SU-Net	SSU-Net
25%	66.32%	0%
50%	68.98%	70.42%
75%	69.38%	73.33%
100%	69.80%	74.77%

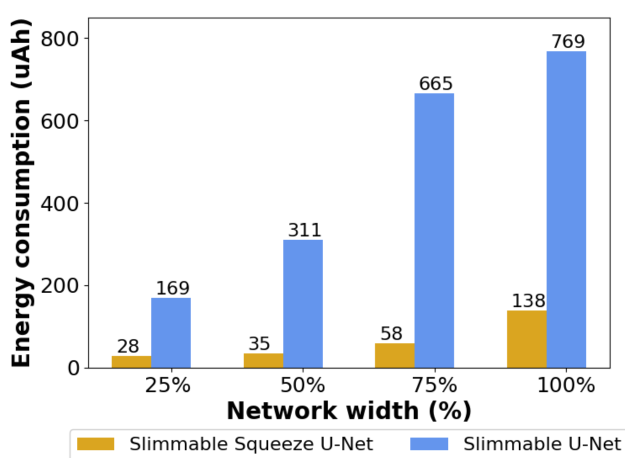
Table 2: Precision metric score for both Slimmable U-Net (SU-Net) and Slimmable Squeeze U-Net (SSU-Net) for the weed detection task.

to the Squeeze U-Net being more accurate in the overall predictions, thus exhibiting a higher precision, but less likely to classify pixels as weeds, thus exhibiting a lower IoU.

5.3 Time and energy consumption

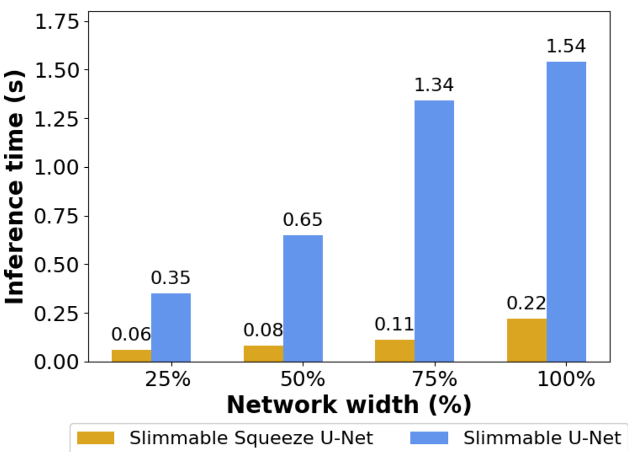
We perform time and energy measurements on a NVIDIA Jetson Nano 4GB [5] board running Ubuntu 20.04.6 LTS. We run NN models using Python 3.8.10 [26] and PyTorch [19] 1.12.0., and measure the power consumption using a Monsoon power monitor tool [13], a commonly used tool for power measurements in embedded computing [23]. The results of these experiments are presented in Figure 6. At the same time, we benchmarked the average inference duration for one sample image for both networks and show the results in Figure 7.

Figure 6: Average energy consumed for processing one image using the two Slimmable U-Nets on the NVIDIA Jetson Nano board.



The measurements show a consistent linear decrease in both the energy consumption and the inference time for both network architectures as the slimming moves towards

Figure 7: Average inference time for processing one image using the two Slimmable U-Nets on the NVIDIA Jetson Nano board.



smaller widths. Taking the 50% width, which is the smallest usable width for both networks, as a reference, we observe that the slimming brings energy savings of 2.4× for the Slimmable U-Net and 3.9× for the Slimmable Squeeze U-Net, compared to the 100% width network in both cases.

The timing measurements follow a similar pattern, with the 50% width networks being 2.3× faster (U-Net) and 2.7× faster (Squeeze U-Net) than their full-width versions.

6 CONCLUSION AND FUTURE WORK

In this paper we present the design, implementation, and preliminary evaluation of AgriAdapt – a solution for lightweight on-UAV weed detection that is based on a novel slimmable U-Net neural architecture for weed detection. Our solution enables dynamic adjustment of the number of parameters used for NN computation, while providing a graceful degradation of inference performance.

The initial results show that through network slimming, AgriAdapt enables energy savings of up to 2.4× compared to using a non-slimmed U-Net and up to 3.9× in case of the Squeeze U-Net, with negligible performance drop – roughly 1% drop in precision for U-Net and 4% for Squeeze U-Net, while for IoU the reduction is similar in both cases.

The encouraging results from the preliminary evaluation described in this paper, presents a basis for our future efforts that will be focused on the exploration of contextual factors, such as image properties, outside brightness, location, and others, that are potentially impacting the difficulty of the weed detection task. More specifically, we aim to devise a context-based adaptation algorithm that will at each inference point select a NN width that maximizes the inference performance, while minimizing resource usage.

ACKNOWLEDGMENTS

The research presented in this paper was funded by the ARRS project "Context-Aware On-Device Approximate Computing" (J2-3047), by the Slovenian Research Agency research core funding No. P2-0098 and P2-0426, and by the H2020 Smart4All 3rd FTTE grant "AgriAdapt".

REFERENCES

[1] Muhammad Hamza Asad and Abdul Bais. 2020. Weed detection in canola fields using maximum likelihood classification and deep convolutional neural network. *Information Processing in Agriculture* 7, 4 (2020), 535–545.

[2] M Dian Bah, Adel Hafiane, and Raphael Canals. 2018. Deep learning with unsupervised data labeling for weed detection in line crops in UAV images. *Remote sensing* 10, 11 (2018), 1690.

[3] Nazanin Beheshti and Lennart Johnsson. 2020. Squeeze u-net: A memory and energy efficient image segmentation network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*. 364–365.

[4] Łukasz Chechliński, Barbara Siemiątkowska, and Michał Majewski. 2019. A system for weeds and crops identification—reaching over 10 fps on raspberry pi with the usage of mobilenets, densenet and custom modifications. *Sensors* 19, 17 (2019), 3787.

[5] NVIDIA Developer. 2022. Jetson Nano Developer Kit. https://developer.nvidia.com/embedded/jetson-nano-developer-kit visited on 2023-06-14.

[6] Mulham Fawakherji, Ali Youssef, Domenico Bloisi, Alberto Pretto, and Daniele Nardi. 2019. Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*. IEEE, 146–152.

[7] Xitong Gao, Yiren Zhao, Łukasz Dudziak, Robert Mullins, and Cheng-zhong Xu. [n. d.]. Dynamic Channel Pruning: Feature Boosting and Suppression. In *International Conference on Learning Representations*.

[8] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. 2015. Deep learning with limited numerical precision. *International conference on machine learning* (2015), 1737–1746.

[9] ASM Mahmudul Hasan, Ferdous Sohel, Dean Diepeveen, Hamid Laga, and Michael GK Jones. 2021. A survey of deep learning techniques for weed detection from images. *Computers and Electronics in Agriculture* 184 (2021), 106067.

[10] Yihui He, Xiangyu Zhang, and Jian Sun. 2017. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE international conference on computer vision*. 1389–1397.

[11] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[12] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016).

[13] Monsoon Solutions Inc. 2015. *Monsoon Solutions high voltage power monitor*. Retrieved February 3rd 2022 from http://msoon.github.io/powermonitor/HVPM.html

[14] Luiz Carlos M Junior and José Alfredo C Ulson. 2021. Real time weed detection using computer vision and deep learning. In *2021 14th IEEE*

International Conference on Industry Applications (INDUSCON). IEEE, 1131–1137.

[15] Muhammad Azim Khan, Shakeel Ahmad, and Ali Raza. 2019. *Integrated Weed Management for Agronomic Crops*. Springer Singapore, Singapore, 257–281. https://doi.org/10.1007/978-981-32-9783-8_14

[16] Bo Liu and Ryan Bruch. 2020. Weed detection for selective spraying: a review. *Current Robotics Reports* 1 (2020), 19–26.

[17] Van Manh Nguyen. 2021. *Weed Dataset Collection*. Retrieved March 15rd 2023 from https://manhnv103.github.io/weed-datasets/

[18] Victor Partel, Jinho Kim, Lucas Costa, Panos M Pardalos, and Yiannis Ampatzidis. 2020. Smart Sprayer for Precision Weed Control Using Artificial Intelligence: Comparison of Deep Learning Frameworks.. In *ISAIM*.

[19] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 234–241.

[21] Muhammad Hammad Saleem, Johan Potgieter, and Khalid Mahmood Arif. 2021. Automation in agriculture by machine and deep learning techniques: A review of recent developments. *Precision Agriculture* 22 (2021), 2053–2091.

[22] Simone Scardapane, Michele Scarpiniti, Enzo Baccarelli, and Aurelio Uncini. 2020. Why should we add early exits to neural networks? *Cognitive Computation* 12, 5 (2020), 954–966.

[23] Andreas Schuler and Gabriele Anderst-Kotsis. 2019. Examining the energy impact of sorting algorithms on Android: an empirical study. In *Proceedings of the 16th EAI Int. Conf. on Mobile and Ubiquitous Systems: Computing, Networking and Services* (Houston, Texas) *(MobiQuitous '19)*. ACM, NY, USA, 404–413.

[24] Sovi Guillaume Sodjinou, Vahid Mohammadi, Amadou Tidjani Sanda Mahama, and Pierre Gouton. 2022. A deep semantic segmentation-based algorithm to segment crops and weeds in agronomic color images. *Information Processing in Agriculture* 9, 3 (2022), 355–364.

[25] Ximeng Sun, Rameswar Panda, Chun-Fu Richard Chen, Aude Oliva, Rogerio Feris, and Kate Saenko. 2021. Dynamic network quantization for efficient video inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7375–7385.

[26] Guido Van Rossum and Fred L. Drake. 2009. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

[27] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. 2018. Slimmable neural networks. *arXiv preprint arXiv:1812.08928* (2018).