Testing Different CNN Architectures for Semantic Segmentation for Landscaping with Forestry Robotics

M.E. Andrada[1], J.F. Ferreira,[1,2], D. Portugal[1] and M. Couceiro[3]

Abstract— **The increasing lack of manpower has driven forestry to become increasingly mechanized, leading to the emergence of forestry robotics. In this article, we present the results of our evaluation of a set of state-of-the-art convolutional neural network-based solutions for semantic segmentation using the Bonnetal open-source training and deployment framework, together with a custom-made solution based on an adaptation of an alternative decoder and encoder for that framework, the *Adapnet++–eASPP* architecture, in the context of a robotic perception pipeline designed to perform landscaping in woodlands to reduce the amount of living flammable material (the Fuel class) for wildfire prevention. Results show that, overall, *Adapnet++–eASPP* was the most robust and comprehensive encoder for our application, demonstrating a consistently high average level of performance in comparison to the other architectures, and displaying the greatest robustness of the group. With this solution, we demonstrated to be able to satisfy our requirements of a low rate of false positives for the Fuel class and operational performance of 10fps.**

I. Introduction

One of the most effective measures for forest fire prevention is to foster landscaping maintenance procedures, namely to "clear" forests, actively reducing fuel accumulation by seasonal pruning, mowing, raking, and disposal of undesired living combustible material, such as herbaceous plants, arboreal vegetation, bush and shrubbery [1]. It is imperative to devise technological solutions to allow workers to engage safely, while simultaneously speeding up operations. Engineering and computer sciences have been starting to be employed to deal with this issue, converging to one particular domain: robotics.

Despite many advances in key areas, the development of fully autonomous robotic solutions for precision forestry is still in a very early stage. This stems from the huge challenges imposed by rough terrain traversability [2], for example due to steep slopes, to autonomous outdoor navigation and locomotion systems [3], but also by limited perception capabilities [4], and reasoning and planning under a high-level of uncertainty [5]. Artificial perception for robots operating in outdoor natural environments has been studied for several decades. For robots operating in forest scenarios, in particular, there is research dating from the late 80s-early 90s – see, for example, [6]. Nevertheless, despite many years of research, as described in surveys over time (e.g., [7], [8], [9]), a substantial amount of problems have yet to be robustly solved.

The SEMFIRE project [10] proposes the development of a multi-robot system (MRS) to reduce the accumulation of live combustible material (e.g. bush, herbaceous plants, etc.), thus assisting in landscaping maintenance procedures (e.g. mulching). This is an application domain with an unquestionable beneficial impact on our society and the proposed project will contribute to fire prevention by reducing wildfire hazard potential.

In this work, our aim is to use semantic segmentation to classify n number of classes in a forestry environment, including living combustible material. More specifically, we intend to:

- identify live flammable material for mulching as our main goal;
- identify tree species to protect (and avoid as obstacles);
- detect people and animals for safety purposes.

To this end, we evaluated the performance in this context of a set of state-of-the-art convolutional neural network (CNN) solutions using an open-source training and deployment framework, together with a custom-made solution based on an adaptation of an alternative decoder and encoder for that framework, which we describe in sections III and IV. We will finish this paper by drawing conclusions and describing future work (section V).

II. Related Work

Semantic segmentation is a natural progression from traditional object classification techniques to create intricate and complex robotic systems. This technique consists in classifying each pixel of an image allowing for better path planning, autonomous navigation and object classification [11]. An example can be seen in Fig. 1, in which each pixel was classified as one of the 6 classes shown in Table I.

Semantic segmentation for vegetation detection in general and plant species discrimination in particular has attracted a lot of interest in the past few years. A lot of work has recently been spurred by the botanical scientific community (see, for example, [12]). Additionally, agricultural robotics has contributed with a significant amount of research for many years now, namely on crop-weed discrimination (e.g.

*This work was supported by the Centre of Operations for Rethinking Engineering (CORE, ref. CENTRO-01-0247-FEDER-037082), and the Safety, Exploration and Maintenance of Forests with Ecological Robotics (SEMFIRE, ref. CENTRO-01-0247-FEDER-03269) research projects co-funded by the "Agencia Nacional de Inovacao" within the Portugal2020 programme.

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

[2] Computational Neuroscience and Cognitive Robotics Group, School of Science and Technology, Nottingham Trent University, Nottingham, UK joao.ferreira@ntu.ac.uk

[3] Ingeniarius, Rua Coronel Veiga Simao, Edificio B CTCV, Coimbra, Portugal micael@ingeniarius.pt

[13], [14]). There is also work on vegetation segmentation for robot navigation, path following and traversibility for outdoor robotics outside of urban areas (e.g. [15]). Finally, there is research that has focused on vegetation classification for very specific applications such as [16], who propose a solution for the classification of floodplain vegetation by fusing structural and spectral data rendered by LiDAR and CASI systems.

Semantic segmentation for tree species discrimination, in particular in forestry contexts, has also received substantial attention in the past decade – alas, a considerable amount of this work relates to satellite or aerial image processing (see, for example, [17], [18]). These works, while interesting, are only marginally relevant to field robotics in forestry, since in the latter robots are at ground-level. Conversely, a smaller subset of this research has been dedicated more specifically to robots in forestry environments or to processing images taken on the ground (generally referred to as "natural images"; see, for example, [19], [20]). Additionally, semantic segmentation has also been used for navigation to distinguish forest trails from less traversable ground [21]. Nonetheless, Very few research efforts have systematically addressed semantic segmentation for field robotics in forestry at ground-level, with the notable exception of the work by [22], who developed an architecture consisting of two modality-specific encoder streams fusing intermediate encoder representations into a single decoder using a proprietary self-supervised model adaptation fusion mechanism which optimally combines complementary features. As intermediate representations are not aligned across modalities, they also introduce an attention scheme for better correlation. This solution is proposed by the authors to segment RGB, depth and/or NIR-based modalities using 5 classes ("sky", "trail", "grass", "vegetation", "obstacles") with very promising results.

Successful semantic segmentation can be linked to advancement on convolutional neural networks (CNN) and its variants such as encoder-decoder architecture, fully convolutional and residual networks. Notable state-of-the-art examples are U-Net [23], MobileNet [24] and ResNet [25]. In essence, all these architectures have a basic structure in common. They contain convolution and pooling layers which filter the image to extract different features while also reducing its spatial size. Newer architectures are generally improvements on these seminal architectures; for example, Adapnet++ [22] uses the same structure as ResNet50 but it adds new residual units to it.

Each of these architectures has its own advantages and disadvantages regarding precision and processing time. More specifically, densely layered neural networks provide higher accuracy at the expense of slower inference processing times. Finding the right balance is integral in real-time robotics applications, therefore, CNN architectures need to be carefully designed with these trade-offs in mind.

III. Proposed Approach

Our particular application and sensor set-up inform our specific requirements concerning semantic segmentation. For our purposes, we are interested in segmenting the image pixel-wise according to the 6 classes listed in Table I (see also Fig. 1). The image inputs for our solution currently consist of three image streams conveyed by a multispectral camera, namely NIR (Near Infrared), Red and Green (NGR) at a rate of 10 frames per second (fps), a frame-rate which imposes a requirement on the execution times for inference to be under 100ms. Another particularly important requirement for our application is to minimize false positive classifications in the Fuel class, a conservative approach which ensures the preservation and safety of local flora and fauna, respectively. To this end, we evaluated the performance of a set of state-of-the-art neural network based-solutions together with a custom-made solution based on an adaptation of an alternative decoder-encoder architecture.

TABLE I

Semantic classes and respective color coding.

Classes	Colors
Background	Black
Live flammable material (aka Fuel)	Red
Canopies	Green
Trunks	Brown
Humans	Yellow
Animal	Purple

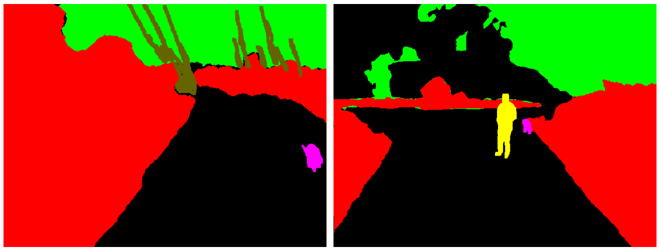

Fig. 1. Two examples of ground truth labeling of multispectral images using the classes listed in Table I.

The set of state-of-the-art, off-the-shelf architectures that we tested was comprised by *MobileNetV2* [24] and *ResNet50* [25]). These were chosen because *MobilenetV2* is reportedly a fast and accurate network while *ResNet50* is slower but more robust.

In addition to these two architectures, the encoder-decoder architecture proposed by [22], consisting of the *Adapnet++* encoder and the *eASPP* (efficient Atrous Spatial Pyramid Pooling) decoder, was adapted and implemented. More specifically, *Adapnet++* is a modification of current *ResNet50* with the addition of 2 residual units to the architecture, while *eASPP* is an adaptation of *ASPP* (Atrous Spatial Pyramid Pooling) developed by DeepLab [26], and was created to reduce the amount of parameters while maintaining the accuracy in the results. A simplified representation of this architecture can seen in the Figure 2. Since *Adapnet++* is based on *ResNet*, it was important to compare how they both performed for our application.

Very importantly for real-time operation, the three architectures meet our 10 fps (100ms) requirement (see Table V).

To achieve the greatest flexibility in model selection

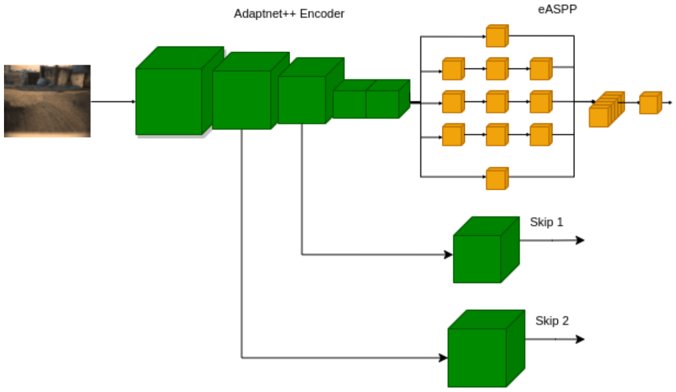

Fig. 2. Simplified representation of the Adapnet++ – eASPP architecture where the green blocks all go through a convolution, ReLU and Batch Normalization and the yellow blocks are 3x3 convolutions with 3, 6, and 12 dilation rate [22].

and/or implementation and the best possible performance in training, deployment and inference, we used an off-the-shelf, stable, easy to use, robotics-friendly tool with a modular codebase called Bonnetal [27], which implements semantic segmentation using CNNs designed to be easily integrated with the ROS (Robotics Operating System) framework while still taking full advantage of available computational resources via user-friendly configuration. This tool allows developers to easily develop new research approaches while avoiding the effort of re-implementing them from scratch or modifying the available code until it becomes at least marginally usable for the research purpose. Fundamentally, Bonnetal mixing and matching any encoder and decoder sets (including backbones such as MobileNet and ResNet) using any of a set of backends (e.g. the PyTorch framework) for the build. Furthermore, it allows changes to hyper-parameters, dataset and architectures effortlessly due to its configuration file and structure.

To complement Bonnetal, for evaluation/benchmarking of the possible solutions we used a specialized tool named Weights and Biases [28]. This tool logs each desired metric to identify the ideal backbone. In our approach, considering our requirements, the metrics used were Jaccard Index (or *IoU*), *Recall* and the *F1 score*:

$$JI/IoU = \frac{TP}{TP + FN + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = \frac{2TP}{2TP + FN + FP}$$

TABLE II

Jaccard Index comparison between different architectures using transfer learning techniques.

Test #	Encoder	Decoder	Pre-Weights	Class Weights	mIoU
#1a	MobileNetv2	ASPP	None	None	0.314
#1b	MobileNetv2	ASPP	✓	None	0.405
#2a	ResNet50	ASPP	None	None	0.419
#2b	ResNet50	ASPP	None	✓	0.544
#2c	ResNet50	ASPP	Decoder only	None	0.562
#2d	ResNet50	ASPP	✓	✓	**0.729**
#3a	Adaptnet++	eASPP	None	None	0.388
#3b	Adaptnet++	eASPP	None	✓	0.425
#3c	Adaptnet++	eASPP	✓	✓	0.660

where:

- TP = *True Positive* – correctly classified pixels
- FP = *False Positive* – pixels wrongly classified
- FN = *False Negative* – pixels wrongly indicate class is absent

These metrics were selected because each provide insightful information on an encoder's quality. Specifically, IoU provides the most comprehensive test as it calculates the positive result in comparison to all the possible values, while *Recall* analyzes the quality of the positive values. For example, if the value for People is near 1, it means it rarely misidentifies people with any other class. This would be ideal as we would like to avoid humans to be identified as fuel. Lastly, the $F1$ score calculates as the Jaccard Index but it attributes a larger weight to true positives therefore potentially providing a more coherent outcome.

IV. Experimental Results

In the description that follows, training was performed using a GeForce RTX 2070 GPU and Core i7 8th Gen CPU on the SEMFIRE dataset, which is composed by 500 labeled NGR images (NIR, Green and Red Channels) according to the classes listed in Table I.

A. Ablation experiments with transfer learning

The first set of tests conducted were to identify if any kind of transfer learning would be advantageous for an image that it is not RGB. Since our dataset contains NGR images, it was not known if the pre-trained models would transfer features well. Therefore, a few tests were conducted utilizing the same hyper-parameters – architecture configurations and quantitative results for all tests are summarized in Table II.

In Test #1, the *MobileNetv2* backbone was used with the *ASPP progressive* decoder and no frozen layers, both with and without pre-weights (Figs. 4 and 3, respectively). The ImageNet pre-trained model, a dense dataset with 1.2 million RGB images and 60 different classes [29], was used. As can be observed, using no pre-weights results in substantial misidentifications, most likely due to the dataset's size.

Empirical testing shown on Table II shows transfer learning techniques do indeed improve the *IoU* value in all neural networks analyzed. Even though the pre weights image are

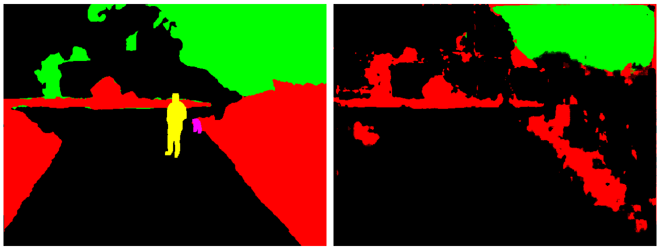

Fig. 3. Example output for model consisting of *MobileNetv2* backbone, *ASPP* progressive decoder and fine tuning trained on Bonnetal with no pre-weights. Ground truth image is shown on the left and corresponding prediction on the right.

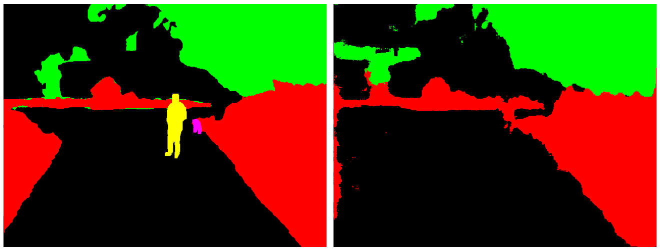

Fig. 4. Example output for model consisting of *MobileNetv2* backbone, *ASPP* progressive decoder and fine tuning trained on Bonnetal using Imagenet pre-weights. Ground truth image is shown on the left and corresponding prediction on the right.

based on the usual RGB channels (i.e. a different modality from NRG), it did improve the quality of training. Moreover, both *Adapnet++* and *Resnet50* improve classification using special label weighting for each class instead of the default value **1**, a technique used to help with class imbalance when there are classes with few instances [30].

Therefore, we assumed that transfer learning was beneficial in all cases, and proceeded to test the effect of freezing layers on transfer learning for ResNet. For Test #2, *ResNet50* was used as a decoder while keeping the same encoder (*ASPP progressive*). Fig. 5 shows the outcome for this architecture using ImageNet pre-weights, 500 epochs and frozen encoder layers. Conversely, Fig. 6 shows results for the same architecture using 700 epochs and special label weighting for each class instead of the default value **1**, a technique used to help with class imbalance when there are classes with few instances [30]. Using this setup resulted in improved results for *ResNet50* in rarely occurring cases such as Animals. It also showed that false positives were minimized when comparing to *MobileNetv2*, possibly because *ResNet50* is a more robust encoder.

In conclusion, Test #2 shows that traditional transfer learning (i.e. freezing encoder layers when the features are similar) does not produce optimal quality. This can be due the different modalities used for transfer learning and training the final model (i.e. RGB vs NGR), but more tests are needed to confirm this hypothesis. Overall, transfer learning techniques such as weight initialization with class imbalance and pre-trained weights have shown to improve the overall quality of the semantic segmentation classification.

Finally, Test #3 served to compare with the *Adapnet++–eASPP* architecture if encoder layers are not frozen (i.e. best-case scenario), showing slightly lower but still similar performance to *Resnet50*, which is understandable as the backbone of the former is an adaptation of the latter.

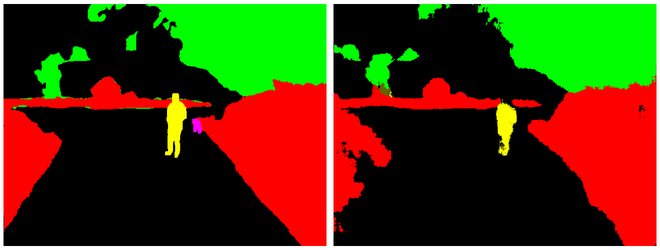

Fig. 5. Example output for model consisting of *ResNet50* backbone, *ASPP* progressive decoder and fine tuning trained on Bonnetal using Imagenet pre-weights with frozen encoder layers. Ground truth image is shown on the left and corresponding prediction on the right.

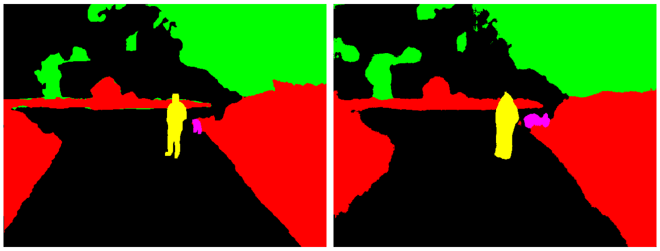

Fig. 6. Example output for model consisting of *ResNet50* backbone, *ASPP* progressive decoder and fine tuning trained on Bonnetal using Imagenet pre-weights for the whole model. Ground truth image is shown on the left and corresponding prediction on the right.

B. Hyper-parameter optimization experiments

The second set of experiments was designed to determine the optimal set of hyper-parameter values for each architecture. Given that the first set of experiments showed that transfer learning does indeed improve results, the next set of tests already included pre-trained models and weight initialization for class imbalance. As mentioned in section III, the hyper-parameters were modified using the Weights and Biases tool [28]. Weights and Biases uniformly changes each input between a fixed range to find the optimal validation loss, which was chosen because it is inversely proportional to the evaluation metrics – in other words, a lower loss yields higher *IoU*, *Recall* and *F-1* Score values. Each configuration was trained for 500 epochs and for each backbone, tests were conducted at least 20 times. The hyperparameters in question are listed and explained in Table III.

Fig. 8 and 9 show the results from the sweep made by Weights and Biases for *Adapnet++*, *ResNet50* and *MobileNetv2*. . These show that a specific configuration may yield a negligible *IoU* value for Fuel using the *MobileNetv2* backbone while its maximum is near the values found for ResNet50 and *Adapnet++*. This same effect can be better seen in *IoU* for Humans, where *ResNet50* has a lower variance between best and worst results due to the robustness of the system holding up regardless of the hyperparameters.

Fig. 8 shows comparison results using the *IoU* metric, demonstrating that changes in hyper-parameters significantly affect the performance of each architecture. For instance,

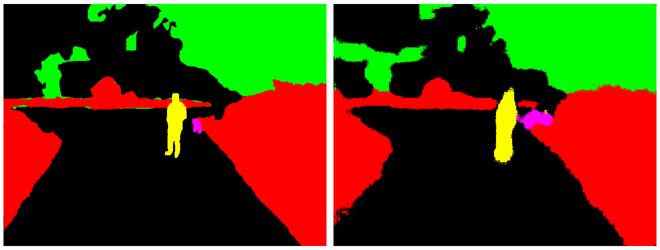

Fig. 7. Example output for model consisting of *Adapnett++* backbone, *eASPP* progressive decoder and fine tuning trained on Bonnetal using Imagenet pre-weights for the whole model. Ground truth image is shown on the left and corresponding prediction on the right.

TABLE III
CONFIGURABLE TRAINING PARAMETERS AND THEIR PURPOSES.

Parameters	**Purpose**
Max LR	Stochastic Gradient Descent (SGD) maximum learning rate. It is an adjustment in the weights of our network with respect to the loss gradient descent
Min LR	Warmup initial learning rate
Up Epochs	Decides the number of epochs used for warmup
Down Epochs	Decides number of epochs used for warmdown
Max/Min Momentum	SGD momentum max/min when LR is max/min respectively
Final Decay	Learning rate decay per epoch from Min LR
W Decay	Weight decay value for L2 regularization
Batch Size	Number of training examples utilized in one iteration
Backbone/Decoder Dropout	Number of layers dropped out while training for backbone/decoder
Backbone/Decoder Batch Normalization Decay	Decay on batch normalization to reduce noise on the backbone/decoder

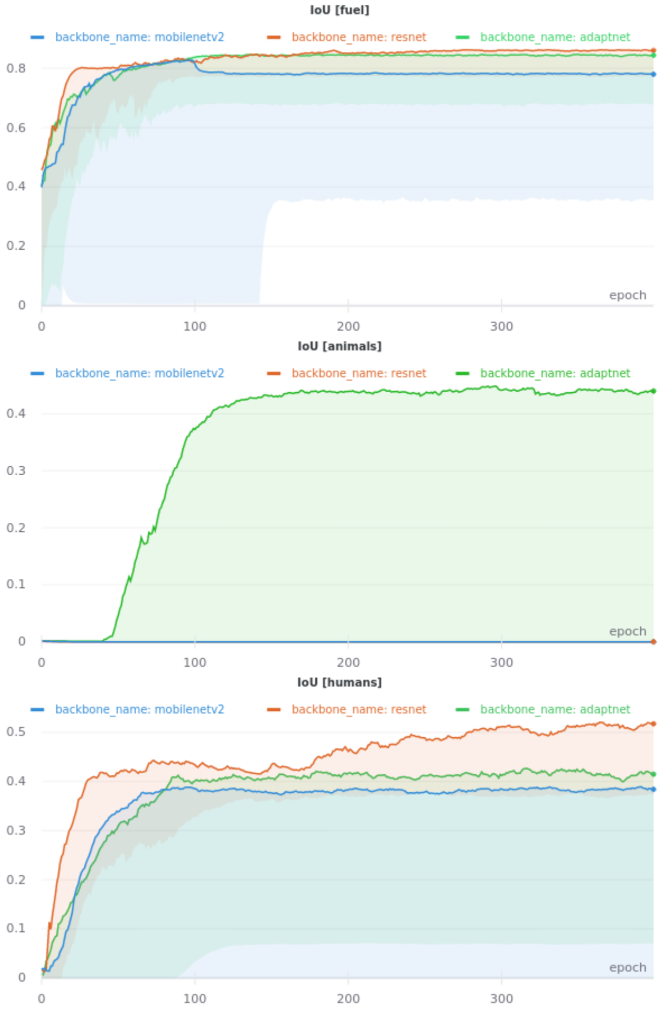

Fig. 8. Comparison plots of IoU *vs* Epoch for the *Adapnet++*, *ResNet50* and *MobileNet* encoders, for the Fuel, Animals and Humans classes resulting from the sweep made using Weights and Biases (respectively, from top to bottom; green for *Adapnet++*, yellow for *ResNet50* and blue for *MobileNetv2*). Lighter, faded colors represent all values depending on the chosen hyperparameters, while darker, contrasting colors correspond to the best results.

when considering the IoU for the Fuel class, *MobileNetV2* produces results close to 0 or near the best results of *ResNet50*. The same phenomenon happens for *Adapnet++* in the segmentation of humans and animals. This demonstrates the importance of choosing the right hyper-parameters and finding which has the greatest influence on segmentation quality. As can be seen in the plot, segmentation quality does not differ much in a class with a great number of samples such as Fuel. The highest values for each architecture are similar, with *ResNet50* achieving the best results and *MobileNetV2* the worst performance. However, in classes with smaller number of samples, such as Humans and Animals, results differ significantly. The Humans class has similar values between all three, but *ResNet50* widens the gap to the other two architectures after 200 epochs, reaching a 0.1 gap between itself and *Adapnet++*. Conversely, the Animals class exhibits a more unpredictable outcome. In fact, by 400 epochs, only *Adapnet++* is able to detect animals at all in every experiment. This effect can be due to the other networks needing more epochs to start recognizing that particular class more effectively. It also shows that the new residual units from *Adapnet++* have a positive effect when identifying classes with low number of instances.

In summary, even though *ResNet50* performs best on Humans and Fuel, *Adapnet++* achieves a better overall performance, since it is able recognize a specific class for which all others fail.

The $Recall$ metric, as mentioned previously, is the most appropriate metric for evaluating the performance of all three architectures in what concerns our requirement of avoiding as many false positives for Fuel as possible. For parameter optimization, it follows the same pattern as IoU regarding the range of detection between the worst and best models trained for each architecture. This aligns with our expectations as both depend on the same values of true positives and false negatives.

Fig. 9 shows impressive *Recall* values for Fuel, with *Adapnet++* and *ResNet50* achieving the same results at

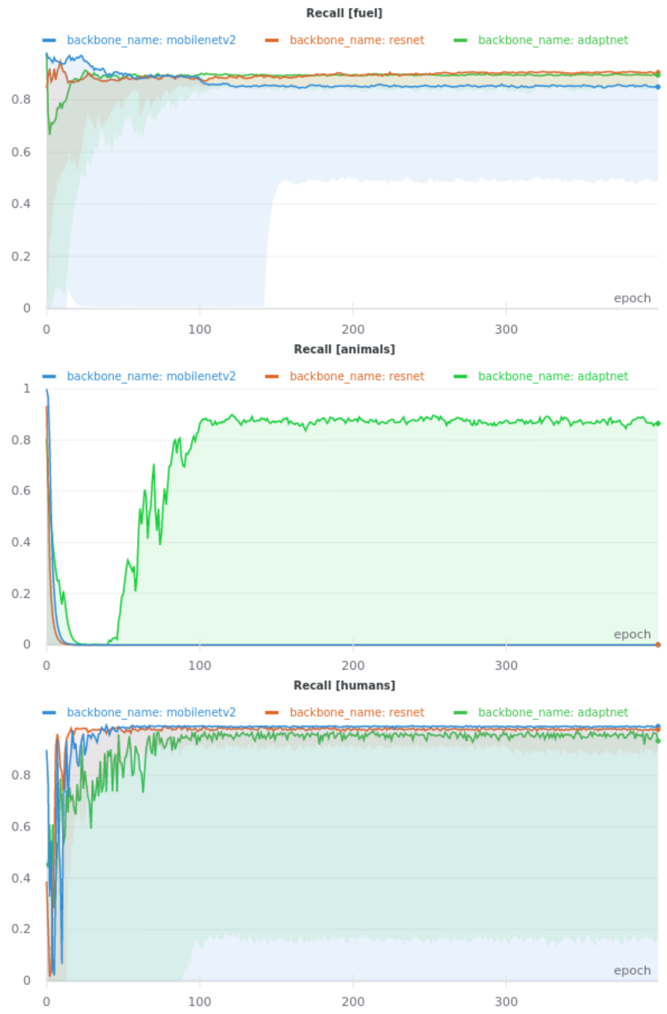

Fig. 9. Comparison plots of Recall *vs* Epoch for the *Adapnet++*, *ResNet50* and *MobileNet* encoders, for the Fuel, Animals and Humans classes resulting from the sweep made using Weights and Biases (respectively, from top to bottom). Coloring scheme follows same convention as with Fig. 8.

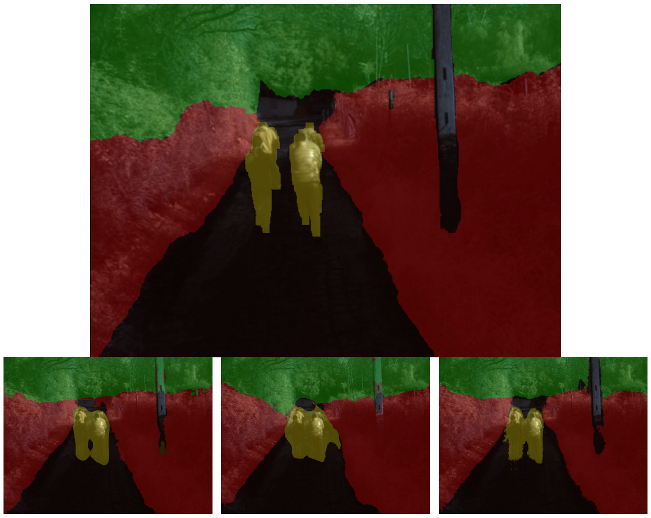

Fig. 10. Qualitative Results for *MobileNetv2* (left), *ResNet50* (middle) and *Adapnet++* (right) in comparison to the ground truth (top).

TABLE IV
RESULTS FOR BEST MEAN FOR EVALUATION METRICS.

Encoder	**Decoder**	**mIoU**	**mRecall**	**mF1-Score**
MobileNetv2	ASPP	0.5352	0.7672	0.6328
ResNet50	ASPP	0.5326	0.7283	0.6339
Adaptnet++	eASPP	**0.6475**	**0.8967**	**0.7677**

0.9 while *MobileNetV2* obtaining 0.85. This means that Fuel is rarely misclassified. However, to ensure the best performance, it is necessary to obtain high *Recall* values for Humans and Animals so they are not misidentified as fuel. *Recall* for Humans shows great promise as it is above 0.9 for all architectures. *MobileNetV2* marginally attained the best values at 0.995, while *ResNet50* and *Adapnet++* achieve slightly lower results at 0.979 and 0.934 respectively. However, as with the *IoU* metric, only *Adapnet++* is able to recognize animals up to 400 epochs at 0.865.

Fig. 10 shows qualitative results for all architectures for a representative example including instantiations of all classes except Animals. As seen in the figure, all three results visually seem to be near the expected ground truth with *Adapnet++* probably being the best match from visual inspection.

Lastly, Table IV shows the overall result for each encoder with all metrics, including the $F1score$. As shown previously, while *ResNet50* has a slight advantage over *Adapnet++* for specific Classes such as Fuel and Humans, the latter achieves the best mean results as it is the only architecture that recognizes animals at all.

Even though *Recall* values are high for Humans, Fig. 10 shows the architectures tend to overestimate the size of humans and fuel – in fact, most categorized the post in the background as fuel, which would clearly represent a safety hazard if the high-level modules of the mulching robot were to take that classification as accurate.

Moreover, the architectures many times mislabel, in a non-cluster-like fashion, most background features as either Human, Animal or Fuel, as can be seen in Fig. 11. Most likely, this happens because that section of the image is the side of a truck with a logo and there are no similar scenarios in the dataset. A possible solution for these issues is to include other modalities such as depth. In fact, it could also improve accuracy and lower amount of false positives.

C. Complexity and inference execution time analysis

Finally, all architectures were compared in terms of number of parameters used and inference execution times – results are summarized in Table V. As expected, *MobileNetv2* is by far the fastest running architecture matched by the considerably lower number of parameters it uses, while the other two architectures exhibit much closer figures, with *Adapnet++–eASPP* being lighter and faster than *ResNet50*.

V. CONCLUSIONS & FUTURE WORK

In this article, we presented the results of experiments conducted to test the performance of a set of state-of-the-art

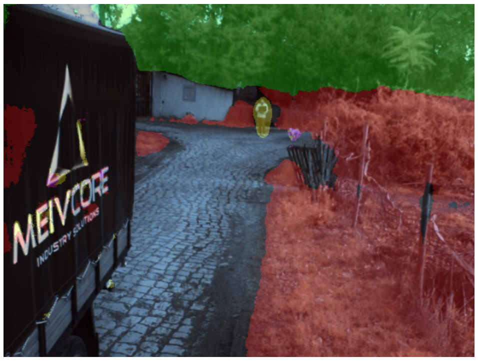

Fig. 11. Example of ground truth (left) and respective prediction (right) made by *MobileNetv2* backbone and ASPP progressive decoder with fine tuning showing mislabeling of features on the lettering on the side of the truck as being member of other than the expected Background class.

TABLE V
COMPLEXITY AND INFERENCE EXECUTION TIME FOR 640 × 480 INPUT IMAGES.

Encoder	**Decoder**	**# Parameters**	**Inf. Time (ms)**
MobileNetv2	ASPP	**2,154,862**	**11.6**
ResNet50	ASPP	49, 652, 118	53
Adapnet++	eASPP	32, 345, 870	40.9

neural network based-solutions using an open-source training and deployment framework, together with a custom-made solution based on an adaptation of an alternative decoder and encoder for that framework, in the context of a perception architecture for a forestry multi-robot system designed to perform landscaping in woodlands to reduce the amount of living flammable material for wildfire prevention. Results show that, overall, *Adapnet*++ was the most robust and comprehensive encoder for our application. It demonstrated a consistently high average level of performance in comparison to the two other architectures, and displayed the greatest robustness of the group.

While the experiments show promising results, we plan in the future to expand the architecture to allow for multimodal integration of different modalities adding to the available NGR channels, RGB or depth (from cameras or even from LiDAR sensors). For that purpose, in ensuing work we intend to test the expansion of the system with fusion layers and a depth encoder.

ACKNOWLEDGMENTS

We would like to thank Daryna Datsenko and Roman Prykhodchenko for their invaluable help with the preliminary experiments with Bonnetal and dataset labelling.

REFERENCES

[1] C. Ribeiro, S. Valente, C. Coelho, and E. Figueiredo, "A look at forest fires in Portugal: technical, institutional, and social perceptions," *Scandinavian Journal of Forest Research*, vol. 30, no. 4, pp. 317–325, 2015.

[2] B. Suger, B. Steder, and W. Burgard, "Traversability analysis for mobile robots in outdoor environments: A semi-supervised learning approach based on 3D-lidar data," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA 2015)*, (Seattle, Washington), May 2015.

[3] R. Siegwart, P. Lamon, T. Estier, M. Lauria, and R. Piguet, "Innovative design for wheeled locomotion in rough terrain," *Robotics and Autonomous Systems*, no. 40, pp. 151–162, 2002.

[4] M. K. Habib and Y. Baudoin, "Robot-Assisted Risky Intervention, Search, Rescue and Environmental Surveillance," *International Journal of Advanced Robotic Systems*, vol. 7, no. 1, 2010.

[5] S. Panzieri, F. Pascucci, and G. Ulivi, "An outdoor navigation system using GPS and inertial platform," *IEEE/ASME transactions on Mechatronics*, vol. 7, no. 2, pp. 134–142, 2002.

[6] F. A. Gougeon, P. H. Kourtz, and M. Strome, "Preliminary research on robotic vision in a regenerating forest environment," in *Proc. Int. Symp. Intelligent Robotics Systems*, vol. 94, pp. 11–15, 1994.

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

[8] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, P. Rander, S. Thayer, N. Vallidis, and R. Warner, "Toward Reliable Off Road Autonomous Vehicles Operating in Challenging Environments," *The International Journal of Robotics Research*, vol. 25, pp. 449–483, Jan. 2006.

[9] S. Lowry and M. J. 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.

[10] M. S. Couceiro, D. Portugal, J. F. Ferreira, and R. P. Rocha, "Semfire: Towards a new generation of forestry maintenance multi-robot systems," 2019.

[11] I. Ulku and E. Akagunduz, "A survey on deep learning-based architectures for semantic segmentation on 2d images," 2019.

[12] H. Goeau, P. Bonnet, and A. Joly, "Plant identification based on noisy web data: the amazing performance of deep learning (LifeCLEF 2017)," in *CLEF 2017 - Conference and Labs of the Evaluation Forum*, (Dublin, Ireland), pp. 1–13, Sept. 2017.

[13] A. Milioto, P. Lottes, and C. Stachniss, "Real-Time Semantic Segmentation of Crop and Weed for Precision Agriculture Robots Leveraging Background Knowledge in CNNs," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2229–2235, May 2018.

[14] I. D. García-Santillán and G. Pajares, "On-line crop/weed discrimination through the Mahalanobis distance from images in maize fields," *Biosystems Engineering*, vol. 166, pp. 28–43, Feb. 2018.

[15] A. Brunner and B. Gizachew, "Rapid detection of stand density, tree positions, and tree diameter with a 2D terrestrial laser scanner," *European Journal of Forest Research*, vol. 133, pp. 819–831, Sept. 2014.

[16] G. W. Geerling, M. Labrador-Garcia, J. Clevers, A. M. J. Ragas, and A. J. M. Smits, "Classification of floodplain vegetation by data fusion of spectral (CASI) and LiDAR data," *International Journal of Remote Sensing*, vol. 28, no. 19, pp. 4263–4284, 2007.

[17] J. Lisein, A. Michez, H. Claessens, and P. Lejeune, "Discrimination of Deciduous Tree Species from Time Series of Unmanned Aerial System Imagery," *PLOS ONE*, vol. 10, p. e0141006, Nov. 2015.

[18] C. Dechesne, C. Mallet, A. Le Bris, and V. Gouet-Brunet, "Semantic Segmentation of Forest Stands of Pure Species as a Global Optimization Problem," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 4, 2017.

[19] T. Hellström and A. Ostovar, "Detection of Trees Based on Quality Guided Image Segmentation," in *Second International Conference on Robotics and associated High-technologies and Equipment for Agriculture and forestry (RHEA-2014)*, pp. 531–540, 2014.

[20] M. Carpentier, P. Giguère, and J. Gaudreault, "Tree Species Identification from Bark Images Using Convolutional Neural Networks," *arXiv:1803.00949 [cs]*, Mar. 2018.

[21] A. Giusti, J. Guzzi, D. C. Ciresan, F.-L. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro, *et al.*, "A Machine Learning Approach to Visual Perception of Forest Trails for Mobile Robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 661–667, 2016.

[22] A. Valada, R. Mohan, and W. Burgard, "Self-supervised model adaptation for multimodal semantic segmentation," *International Journal of Computer Vision (IJCV)*, jul 2019. Special Issue: Deep Learning for Robotic Vision.

[23] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *CoRR*, vol. abs/1704.04861, 2017.

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

[26] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *CoRR*, vol. abs/1606.00915, 2016.

[27] A. Milioto and C. Stachniss, “Bonnet: An open-source training and deployment framework for semantic segmentation in robotics using cnns,” *CoRR*, vol. abs/1802.08960, 2018.

[28] L. Biewald, “Experiment tracking with weights and biases,” 2020. Software available from wandb.com.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[30] M. Zhu, J. Xia, X. Jin, M. Yan, G. Cai, J. Yan, and G. Ning, “Class weights random forest algorithm for processing class imbalanced medical data,” *IEEE Access*, vol. 6, pp. 4641–4652, 2018.