Comparative Evaluation of Deep Learning Architectures for Environment and Obstacle Recognition in Robotic Lawn Care

Julius Siaulys [1,] and Agne Paulauskaite-Taraseviciene [1]

[1] *Kaunas University of Technology, Faculty of Informatics, Studentu 50, Lithuania*

Abstract

Robotic lawn mowers typically rely on boundary wires, which are installed around the perimeter of the lawn to define the mowing area. While boundary wires have been a reliable technology for robotic lawn mowers, there are certainly limitations and inefficiencies associated with them. AI technologies have the potential to improve the performance and capabilities of robotic lawn mowers, and reduce the reliance on boundary wires. By using computer vision technologies to recognize obstacles and environments in robotic lawn mower fields, it could potentially eliminate the need for boundary wires and reduce the risk of damage to objects on the field and the health risks to small animals. This study aims to evaluate different deep learning architectures for recognizing obstacles and environments in robotic lawn mower fields, using the Ade20k dataset. The differently merged datasets were evaluated on the full dataset of 150 classes, 21 classes and 3 merged subsets of classes. The results of the comparison of seven different deep learning models showed that merging classes is practical for this task and to improves semantic segmentation accuracy results by up to 1.4 times.

Keywords

Robot mowers, deep learning, accuracy metrics, image recognition, semantic segmentation, optimization

1. Introduction

Mowing the lawn is a common task for many people living in the countryside, but it is a tedious and time-consuming process. In addition, there are many restrictions on when you can mow your lawn. One of the main restrictions is bad weather conditions, because it's not recommended to mow your lawn when it's wet or damp, as the grass can become easily damaged, and it can be dangerous for the operator to walk on slippery surfaces. When mowing the lawn after a rainfall, wet grass can be a challenge. The moisture causes the grass blades to stick together and clump up in the mower blade area, which can lead to clogs and an uneven cut, also, mowing during or after rain can cause soil compaction and harm the grass roots. Moreover, there are noise regulations that restrict when you can mow your lawn, especially in urban or residential areas. Many cities and towns have noise ordinances that specify when outdoor activities, including lawn mowing, are allowed. One solution to solve these problems altogether is to use lawn mowing robot.

Most robot mowers use an electromagnetic field created by a boundary wire that is installed around the perimeter of the area to be mowed [1-3]. This boundary wire serves as a virtual fence that guides the robot mower and prevents it from leaving the mowing area or entering areas that are off-limits. The boundary wire is usually installed prior to using the robot mower and requires some initial setup time. If new obstacles appear in the area, such as landscaping changes or newly installed objects, the boundary wire may need to be readjusted to accommodate the changes. In addition, the use of a boundary wire

IVUS 2023, May 12, 2022, Kaunas, Lithuania
EMAIL: julius.siaulys@ktu.lt (J. Siaulys); agne.paulauskaite-taraseviciene@ktu.lt (A. Paulauskaite-Taraseviciene);
ORCID: 0000-0002-2425-7260 (A. 1); 0000-0002-8787-3343 (A. 2);

can limit the flexibility of robot mowers, and it may also pose a risk to animals and other living beings that come into contact with the wire.

The study was conducted in the UK to analyze the impact of robot mowers on European Hedgehogs [4]. The study found that depending on the model, the encounter could be fatal to the hedgehogs. To address these concerns, some newer robot mowers incorporate advanced sensors and safety features to detect and avoid obstacles, including animals. The ultrasonic sensors or cameras can be used to detect obstacles and adjust their path accordingly [5,6].

The aim of this research is to develop an algorithm that can be integrated into an autonomous system without external devices, using cameras as input and making decisions to drive, avoid or cross based on the real-time situation in front of the robot. This could be used not only on private properties but also in large-scale commercial areas. In addition, the system could be integrated into communicating robots that could work together to increase efficiency. The system aims not only to reduce human resource costs but also to increase overall environmental sustainability, as robotic lawnmowers use electricity as a source of energy compared to petrol used in traditional lawnmowers.

2. Related works

The easiest way to build a robot mower is to use ultrasonic sensors [7]. Ultrasonic sensors can be used to help the robot mower navigate around obstacles, by measuring the distance to objects in front of it and adjusting its course accordingly [8,9]. However, they are not the only type of sensor that can be used for this purpose. Other sensors such as infrared sensors or lidar sensors can also be used for obstacle detection and avoidance [10,11,12]. Measuring distance using ultrasonic sensors as a reference point is not a viable solution as they have a blind spot and the obstacle has to be at the minimum and maximum distances [13]. The accuracy also depends on the temperature and humidity of the air, as these circumstances affect the speed of sound in air [14]. However, an obstacle, such as a bush, may not be detected due to gaps between branches. Thus, another possible solution is to use an infrared passive sensor and camera. In the study, to ensure safety, if the robot detected any obstacle, the system stopped and the user had to reset the parameters to restart the application [15].

To replace the boundary wire navigation, the use of beacons has been tested. For navigation a triangulation method can be employed [16]. While accuracy was satisfactory, this requires additional external equipment. Using GPS sensors and real-time kinematics (RTK), the navigation achieved an accuracy of 4 cm when moving and 4.5 cm when standstill. However, clouds also have a significant impact on accuracy. Other solutions therefore combine several sensors such as GPS, ultrasound and cameras. The images from the cameras were processed by algorithms that distinguished the colour green and allowed the robot to move and cut according to the algorithm's results. This solution is quite accurate, but in the presence of obstacles, which are also made up of green, it can go unnoticed, such as a green water hose [17-19]. Another solution which is open source is called “OpenMower” that allows users to customize it with sensors like GPS, Compass, ultrasound, bumper. With the data coming from all these sensors it is possible to use a variety of algorithms to navigate and mow the lawn, although it has limitations discussed before, it offers a great platform to develop more efficient robot mower system by integrating new types of sensors such as cameras.[32]

3. Deep learning models

For image recognition tasks, both object detection and segmentation are commonly used. Object detection is often preferred because it has smaller hardware requirements and can be more efficient. However, in this particular case, the task requires the precise identification of the boundary between mowable and non-mowable surfaces, which is not possible with object detection alone. Therefore, we have decided to focus on semantic segmentation, which allows us to accurately classify every pixel in the image. While segmentation may require more computational resources than object detection, it provides more detailed and accurate results, making it a better choice for this specific task. In our study, various deep learning models for semantic segmentation on a given dataset were included in order to perform a comparative analysis of model results. The models tested in this study involves FCN [21], LEDNet [22], ICNet [23], ContextNet [24], Deeplab [25], FastSCNN [26], and UNet [27]. All of these models were chosen because of their high performance in semantic segmentation. The Xception65 backbone was chosen as it has demonstrated high accuracy in previous studies and has been shown to be efficient in terms of computational resources [28]. The FCN model is a popular architecture for semantic segmentation, which consists of a series of convolutional layers that allow for end-to-end training and produce a dense prediction map [21]. One of the main advantages of LEDNet is its efficiency, because the model has a small number of parameters and can run in real-time on resource-constrained devices, such as smartphones and embedded systems [22]. The ICNet Its multi-resolution approach and lightweight architecture for real-time semantic segmentation of high-resolution images where both accuracy and efficiency are important [23]. ContextNet is another promising architecture for semantic segmentation tasks in real-time [24]. It combines features at different scales, allowing it to capture both local and global contextual information. One of the main advantages of the DeepLabv3+ architecture is its ability to produce high-quality segmentation results [25]. The model has achieved state-of-the-art performance on several benchmark datasets, including PASCAL VOC, COCO, and Cityscapes. The use of atrous convolutions and the ASPP module allows the model to capture multi-scale information, which is important for accurately segmenting objects at different scales and resolving boundary ambiguities.

The FastSCNN model is another lightweight architecture designed for real-time semantic segmentation tasks [26]. It uses a combination of depthwise separable convolutions and pyramid pooling to produce accurate results while requiring fewer computational resources. Finally, the UNet model is an encoder-decoder architecture that was introduced in 2015 for biomedical image segmentation. It has a skip-connection structure that allows for the preservation of spatial information and produces high-quality segmentation results [27].

In this research denoted models have been trained on the same dataset - Ade20k [20]. Xception65 have been chosen as the backbone architecture [29]. The Deeplab model was also trained with a MobileNet backbone instead of the traditional ResNet backbone. The testing results of our DeepLabv3+ model's accuracy using three different backbones are shown in Table 1. As the results were quite similar, we included the MobileNet backbone and the Xception65 backbone to validate our implementation and results

Table 1. Performance of DeepLabv3+ with different backbones architectures [28]

DeepLabv3+ backbone	mIOU σ	F1σ	Accuracy	Precision	Sensitivity	Specificity
Xception_65	0.910 ± 0.015	0.925 ± 0.014	0.968 ± 0.005	0.916 ± 0.028	0.935 ± 0.008	0.977 ± 0.008
MobileNetV2	0.890 ± 0.015	0.907 ± 0.013	0.959 ± 0.007	0.888 ± 0.034	0.928 ± 0.018	0.968 ± 0.012
ResNet101	0.904 ± 0.013	0.920 ± 0.013	0.965 ± 0.005	0.912 ± 0.046	0.929 ± 0.042	0.975 ± 0.014

4. Experimental setup

4.1. Data

The ADE20K dataset is a large-scale image dataset for semantic segmentation, which means that each pixel of an image is labeled with the corresponding object class [30]. The dataset contains 20,210 images and covers a wide range of indoor and outdoor scenes. It includes diverse objects such as humans, animals, vehicles, furniture, and various environmental elements such as vegetation, sky, and water. ADE20K dataset has 150 semantic object classes. These classes are grouped into 3 main categories: stuff, objects, and things. The stuff category includes materials and textures such as grass, sand, and sky. The object category includes objects with clear boundaries such as cars, bicycles, and chairs. The things category includes objects with complex shapes and structures such as humans, animals, and trees.

In grass mower real-life application, it is not necessary to include all the classes from the ADE20K dataset. In this case if we want to detect grass, pavement, various objects that could be in such scene, we do not need to include classes such as skyscrapers or planes. By removing these classes, we can reduce the complexity of the problem, and focus on the relevant classes that are present in the target environment.

In this study, we have removed all unnecessary classes that would not be useful for our purposes, resulting in a subset of the dataset that includes only 21 of the original 150 classes. Classes that were selected provided in the Figure 1. This subset would allow a potential robot mower to detect environment such as grass, water, sand and as well it would be able to recognize objects such as animals, toys and other common objects that could appear on the lawn.

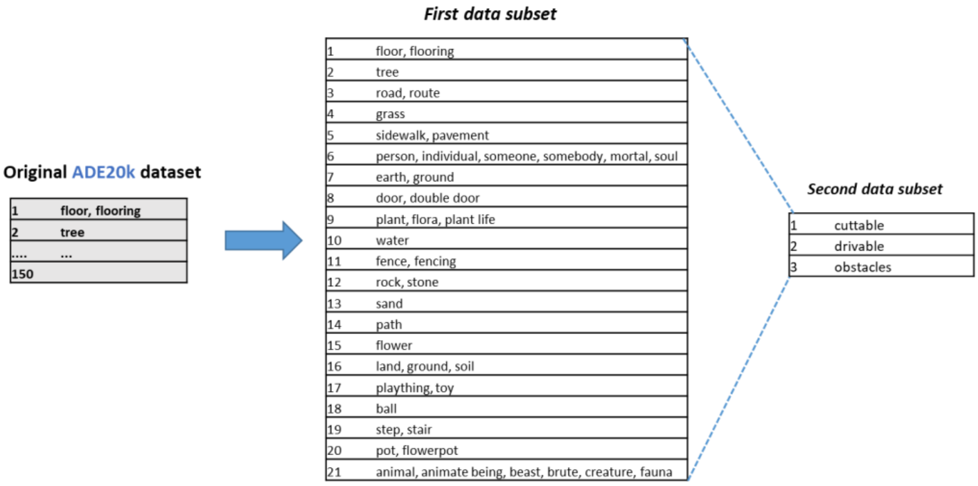

Figure 1: two-iteration optimization of the dataset ADE20k

Another subset of the data was created that includes only three classes: "cuttable," "drivable," and "obstacles." To form these classes, we merged the 21 classes that we previously discussed. This subset was created in an effort to optimize the decision-making process for a robot mower, which needs to determine when to turn on or off its blades and whether to drive over or avoid certain areas. The partitioning of the dataset was not changed from the original ADE20K dataset. Therefore, the training, validation, and testing sets remained the same, with the original split ratio.

4.2. Training parameters

All models were trained for 100 epochs, with a base image size of 520 pixels, the learning rate was set to 0.01. Optimizer of choice was SGD with a momentum of 0.9 and weight decay of 1e-4 (0.0001). PyTorch library [31] was chosen for optimizer, model and other implementations.

4.3. Performance Metrics

Pixel accuracy (PA) and mean intersection over union (mIoU) are two commonly used evaluation metrics for image segmentation. Pixel accuracy measures the percentage of correctly classified pixels while mIoU measures the overlap between predicted and ground truth segmentation masks. mIoU is preferred as it provides a more comprehensive evaluation of performance, especially for imbalanced datasets and allows for a more detailed analysis of individual classes.

$$PA = \frac{\Sigma_i n_{ii}}{\Sigma_i t_i}, \tag{1}$$

$$MIoU = \frac{1}{n_c l} \Sigma_i \frac{n_{ii}}{(t_i + \Sigma_j n_{ji} - n_{ii})}, \tag{2}$$

where $n_c l$ – number of classes included in ground truth segmentation, n_{ji} – number of pixels of class i predicted to belong in to class j, t_i – total number of pixels of class i in ground truth segmentation

5. Obtained Results

The obtained results with 150 classes are shown in Figure 2. The average PA for all included models is above 60%, while the best results were obtained using the Deeplab Xception model with an accuracy of 76.45%, which is almost the same as the FCN model's PA value of 76.43%.

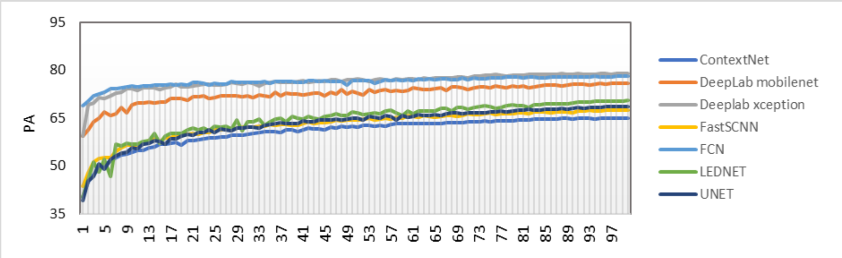

Figure 2: Pixel accuracy (PA) of different models over 100 iterations using 150 classes

However, mIoU results are lower, ranging from 15% to 42% (see Figure 3). This indicates that while the model is correctly classifying majority of the pixels, it is struggling to accurately identify certain regions of the image which results in two visible clusters of performance.

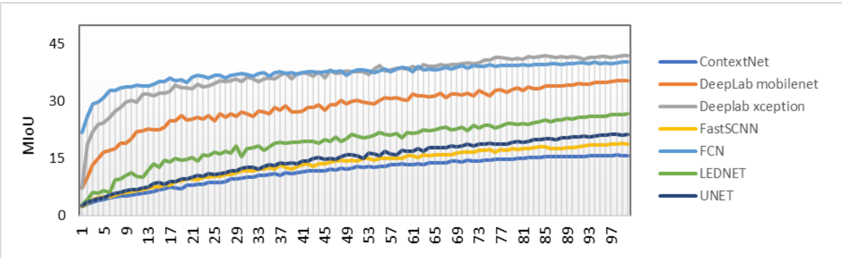

Figure 3: MIoU of different models over 100 iterations using 150 classes

After reducing the number of classes to 21, both PA and mIoU values significantly increased (Figure 4 – Figure 5). The FCN and Deeplab models were ranked in the top 3, just as they were in the 150-class dataset. The other models also showed consistent rankings. In this instance, the evaluated image segmentation models achieved stability in both metrics, pixel accuracy, and mean Intersection over Union (mIoU), before the 100-epoch mark. Nevertheless, to enhance the performance of the models even further, it would be advantageous to train them for a larger number of epochs.

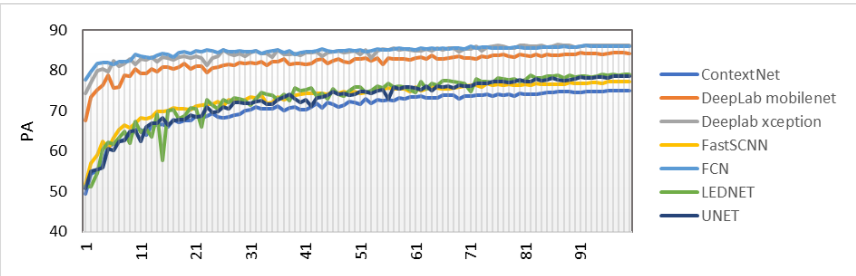

Figure 4: PA of different models over 100 iterations using 21 classes

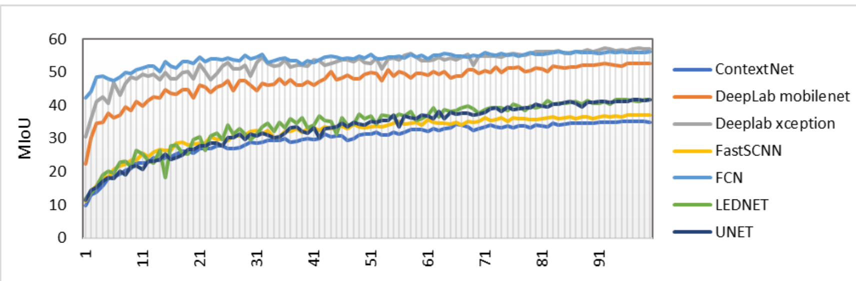

Figure 5: MIoU of different models over 100 iterations using 21 classes

The results for 3 class dataset are provided in Figure 6 and Figure 7. FCN and Deeplab with Xception backbone performed the best – reached ~92% accuracy. Deeplab with Mobilenet backbone had slightly worse accuracy, that was measured at 90.7%. In the case of the image segmentation models with only three classes, the metrics of the models stabilized before the 100-epoch mark. This suggests that a shorter amount of training is required when the number of classes is reduced.

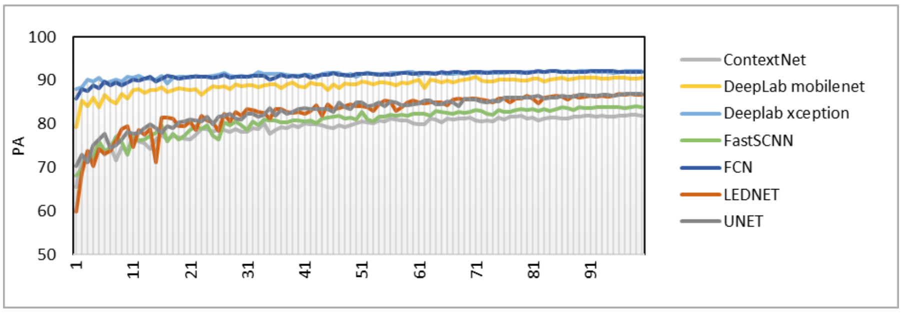

Figure 6: PA of different models over 100 iterations using 3 classes

Using MIoU metric, Deeplab with Xception backbone and FCN reached 79.8% accuracy, closely followed by Deeplab with Mobilenet 77.5% (See Figure 7).

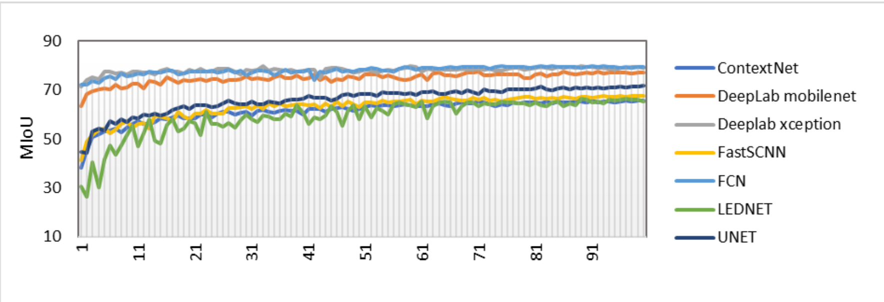

Figure 7: MIoU of different models over 100 iterations using 3 classes

Initially, using a 150-class dataset, the model achieved a pixel accuracy of 71%. However, after trimming down the dataset to 21 classes, the accuracy increased to around 80%. After merging the 21 selected classes into three categories, the pixel accuracy of the model further improved to around 88%.

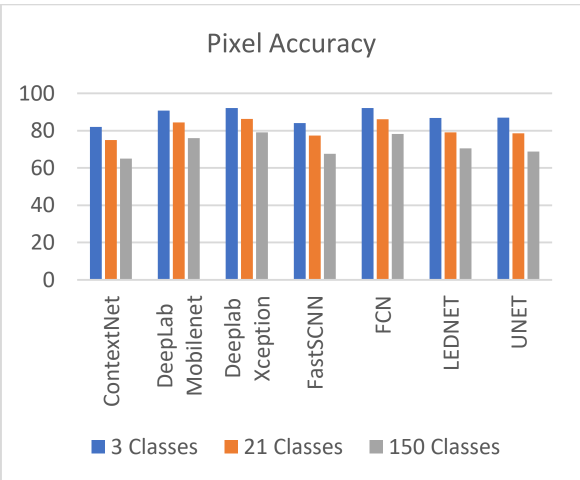

Figure 8: Pixel accuracy of different models

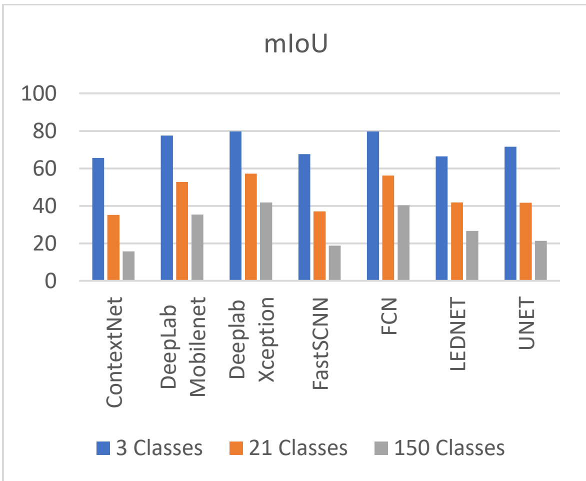

Figure 9: MIoU of different models

Mean intersection over union (mIoU) also improved as the dataset size was reduced. The mIoU score increased from around 28% using the 150-class dataset to roughly 45% when using the 21-class dataset. Furthermore, the mIoU score improved significantly to ~72% when evaluating the performance of the model on the three-category subset dataset (See Figure 8 – 9).

Visual example of inference can be seen in Figure 10. This particular inference displays that the green hose is being partially recognized, but due to the lack of data samples for classes such as “grass” and “garden objects” in ADE20k dataset, it is severely underperforming. This issue can be offset by introducing a custom dataset including more examples for these classes.

(a) (b)

Figure 10: Example inference on image containing grass surface and green hose with two different models: (a) 3 class xception65 trained on custom dataset resulted with PA value of 65.06% and MIoU - 42.33% (b) 3 class LEDNET trained on original dataset resulted with PLA – 1.3% and MIoU – 1.96%.

5.1. Inference speed

Real-time performance is a critical aspect of safe robot mower operation. Therefore, we measured the inference time using a CPU (Intel i5 6600k) by running inference 100 times per model with a 1058x1880 pixel resolution image.

Table 2 Semantic segmentation model inference speed in seconds

FCN	ContextNet	Deeplabv3+ Xception65	Deeplabv3+ Mobilenet	ICNET	LEDNET	FASTSCNN
6.36 ± 0.2	0.812 ± 0.020	8.41 ± 0.1	2.36 ± 0.07	1.88 ± 0.11	2.20 ± 0.13	0.923 ± 0.024

Based on the results presented in Table 2, it can be observed that the most accurate models had the slowest inference times. However, it is worth noting that Deeplabv3+ achieved comparable accuracy when using either Xception65 or MobileNet backbone, but inference speed was measured to be roughly 3.5 times faster when using Deeplabv3+ with MobileNet.

Therefore, it may be reasonable to retrain the FCN model with MobileNet backbone to further increase the inference speed above Deeplabv3+ with MobileNet backbone. It should be noted that the tested image in our study contained nearly 2 million pixels, while in a real robot mower system, the images would likely be closer to 512x512 pixels resolution, which would reduce the number of pixels to roughly 0.26 million. This reduction in pixel count would result in an expected inference time roughly 7 times faster than the results reported in Table 2.

Additionally, in order to further increase performance, it may be beneficial to use a GPU instead of a CPU for inference.

6. Conclusions

Different deep learning models have been tested by addressing the image processing and analysis tasks of a lawn mower robot. A dataset ADE20K with different objects was used to train the models, and the experiments showed that in this case it is appropriate to reduce the dataset and to perform the optimization by focusing only on the most relevant objects. The obtained results of the experiments showed that the strategy of reducing the number of classes from 150 to 21 and merging them into 3 categories was successful and led to a significant increase in pixel accuracy (PA) and MIoU. The results show that the most accurate models are Deeplab Xception and FCN. For the Deeplab Xception and

FCN models, optimizing the classes from 150 to 3 MIoU, the average accuracy improved by 1.4 times, and PA by 1.08 times. For the worst-performing models, the accuracy of MIoU improves even more, reaching 1.8 times for MIoU and 1.11 times for PA. Additionally, reducing the number of classes also led to faster training times for the models. The experiment indicated that by reducing the amount of classes, the models were fully trained faster, thus reducing the overall computational cost. After testing the inference speed, we noticed that changing the backbone from Xception65 to MobileNet significantly improved the inference speed, reducing it from 8.4 seconds to 2.36 seconds, while only lowering the accuracy by 2%. These findings suggest that carefully selecting and optimizing the relevant classes in a dataset can improve the performance and efficiency of deep learning models for image processing tasks. To further optimize the use of semantic segmentation, it would be advisable to explore various backbones in order to reduce inference time while maintaining high accuracy. The incorporation of advanced image recognition technology in robot mowers is anticipated to enhance energy efficiency through the decision-making module that disables the cutting blade motor when the mower is not driving over a mowed surface. Furthermore, the use of cameras may expand the working area and improve safety.

7. References

[1] Gardena, Gardena Sileno Life specification, 2023, URL: https://www.gardena.com /int/products/lawn-care/robotic-mower/robotic-mower-sileno-life-750-m2/967845303

[2] Husqvarna, AutoMower 435X AWD specification, 2023 URL: https://www.husqvarna.com /ie/robotic-lawn-mowers/automower-435x-awd

[3] LandxCape, Landxcape LX790 robot mower specification, 2023, URL: https://landxcape-robotics.com/product/landxcape-lx790-600m²

[4] S. Rasmussen, A. Schroder, R. Mathiesen, J. Nielsen, C. Pertoldi, and D. Macdonald, "Wildlife Conservation at a Garden Level: The Effect of Robotic Lawn Mowers on European Hedgehogs (Erinaceus europaeus)", Animals, vol. 11, no. 5, p. 1191, 2021, doi: 10.3390/ani11051191.

[5] S. Lohar, L. Zhu, S. Young, P. Graf, and M. Blanton, "Sensing Technology Survey for Obstacle Detection in Vegetation", Future Transportation, vol. 1, no. 3, pp. 672–685, Nov. 2021, doi: 10.3390/futuretransp1030036.

[6] S. Duan, H. Ma, J. Zhang, G. Liu, "Development of an Automatic Lawnmower with Real-Time Computer Vision for Obstacle Avoidance", 2021, International Journal of Computational Methods, 19. 10.1142/S0219876221420019.

[7] O.O.E.Ajibola and S.Olajide, "Design and Construction of Automated Lawn Mower. Proceedings of the International Multi Conference of Engineers and Computer Scientists, 2021, IMECS 2021, pp. 1-5.

[8] B. K. Tuncalp, C. Cataltepe, S. Karavil, "Autonomous Lawn Mower Development", Conference: International Conference on Artificial Intelligence and Applications, 2020.

[9] S.V. Tanaji, C. S. Chandrakant, P. S. Shashikant, G. O. Raju, G. S. Bhalchandra, "Automated Mower Robo", Onternational Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056 Vol. 5 (1), 2018

[10] P. Anush, "Infra Red Assisted Navigation for Automatic Lawn Mower Robot", International Journal of Recent Technology and Engineering. 8. 2273., 2019, doi: 10.35940/ijrte.B1251.0982S1119.

[11] M. H. Wu, J. C. Yu and Y. C. Lin, "Study of Autonomous Robotic Lawn Mower Using Multi-Sensor Fusion Based Simultaneous Localization and Mapping", International Conference on Advanced Robotics and Intelligent Systems (ARIS), 2022, pp. 1-4, doi: 10.1109/ARIS56205.2022.9910445.

[12] M. Hoffmann et al., "Coverage Path Planning and Precise Localization for Autonomous Lawn Mowers," 2022 Sixth IEEE International Conference on Robotic Computing (IRC), Italy, 2022, pp. 238-242, doi: 10.1109/IRC55401.2022.00046.

[13] Anodas, "HC-SR04 Ultrasonic Distance Measuring Sensor," [Online]. Available: https://www.anodas.lt/en/hc-sr04-ultrasonic-distance-measuring-sensor

[14] M. Kelemen, I. Virgala, T. Kelemenova, L. Mikova, P. Frankovsky, T. Liptak and M. Lorinc, "Distance Measurement via Using of Ultrasonic Sensor," Journal of Automation and Control, 2015, vol. 3, no. 3, pp. 71-74
[15] J. C. Liao, S.H. Chen, Z.Y. Zhuang, B.W. Wu and Y.J. Chen, 2021. "Designing and Manufacturing of Automatic Robotic Lawn Mower", Processes, Vol. 9, no. 2: 358, pp. 1-21, https://doi.org/10.3390/pr9020358
[16] A. Sjogren, A. Gustafsson, M. Hoang and V. Josefsson, "Robotgrasklippare med ruttplanering," Goteborg, 2017, pp. 136.
[17] A. R. Reddy, N. V. Chaitanya, P. Abhishek and A. Suvarnamma, "Autonomous Solar Based Lawn Mower," International Journal of Pure and Applied Mathematics, 2018, vol. 119, no. 20, pp. 13129-13134
[18] J. M. Derander, P. Andersson, E. Wennerberg, A. Nitsche, E. Moen and F. Labe, "Smart robot lawn mower," Gothenburg, Chalmers University of Technology, 2018, pp. 70.
[19] I. Daniyan, V. Balogun, A. Adeodu, B. Oladapo, J. K. Peter and K. Mpofua, "Development and Performance Evaluation of a Robot for Lawn Mowing," Procedia Manufacturing, 2020, vol. 49, pp. 42-48
[20] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso and A. Torralba, "Scene Parsing through ADE20K Dataset," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 5122-5130, doi: 10.1109/CVPR.2017.544.
[21] J. Long, E. Shelhamer, T. Darrell, "Fully Convolutional Networks for Semantic Segmentation", 2014, CVPR, http://arxiv.org/abs/1411.4038
[22] Y. Wang, et al. "Lednet: A Lightweight Encoder-Decoder Network for Real-Time Semantic Segmentation", International Conference on Image Processing, ICIP, 2019, https://doi.org/10.1109/ICIP.2019.8803154
[23] H. Zhao, et al. "ICNet for Real-Time Semantic Segmentation on High-Resolution Images", Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018, LNCS. https://doi.org/10.1007/978-3-030-01219-9_25
[24] R. P. K. Poudel, U. Bonde, S. Liwicki, C. Zach, "ContextNet: Exploring context and detail for semantic segmentation in real-time", British Machine Vision Conference 2018, BMVC 2018.
[25] Y. Ren, et al. "Full convolutional neural network based on multi-scale feature fusion for the class imbalance remote sensing image classification", Remote Sensing, Vol. 12, No. 21, 2020, https://doi.org/10.3390/rs12213547
[26] R.P. K. Poudel, S. Liwicki, R.Cipolla, "Fast-SCNN: Fast semantic segmentation network". 30th British Machine Vision Conference 2019, BMVC 2019.
[27] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, J. Liang, Unet++: A nested u-net architecture for medical image segmentation. Lecture Notes in Computer Science, 2018, 11045 LNCS. https://doi.org/10.1007/978-3-030-00889-5_1
[28] D. Stifanic, et al. "Semantic segmentation of chest X-ray images based on the severity of COVID-19 infected patients", 2021**,** EAI Endorsed Transactions on Bioengineering and Bioinformatics, 1(3). https://doi.org/10.4108/eai.7-7-2021.170287
[29] F. Chollet, "Xception: Deep learning with depthwise separable convolutions", Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017https://doi.org/10.1109/CVPR.2017.195
[30] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A.Torralba, "Scene parsing through ADE20K dataset" Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017. https://doi.org/10.1109/CVPR.2017.544
[31] Pytorch. (n.d.). *SGD Pytorch documentation*. Retrieved March 25, 2023, from https://pytorch.org/docs/stable/generated/torch.optim.SGD.html
[32] OpenMower. (n.d.). Open Source Robot Mower project. Retrieved May 5, 2023, from https://github.com/ClemensElflein/OpenMower