Lawn Boundary Detection and Segmentation

Nitin Somashekhar
Dipak Sairamesh
Northeastern University Boston, MA
somashekhar.n@northeastern.edu
sairamesh.d@northeastern.edu

Abstract

Automating the process of mowing the lawn might avoid a lot of tedious manual labor. Traditional autonomous lawn mowers use wires to enclose the lawn space and manually map it. Through an analysis of the characteristics of the grass terrain, a special method is developed in this project to automatically estimate the lawn boundary and space. A simple yet effective technique is utilized to accurately identify lawn space and boundaries in continuous frames. The outcomes and technique of actual experiments carried out on a variety of grassy and non-grassy terrains demonstrate the success of the present investigation and its suitability for use in future undertakings.

1 Introduction

Autonomous lawn mowers frequently execute the work of mapping the limits of a lawn and detecting its borders in order to define the space for the robot to move in. Due to the variations in grass type, size, color, texture, and season around the globe, detecting lawn areas is a difficult challenge to address. Classic image processing techniques mentioned in [1] cannot be utilized to recognize grass areas successfully because of these variations and different types of lawns. Using segmented and masked photos of several types of lawn spaces, this project proposes a unique deep learning technique to train a model, and then define the lawn space and edges of the lawn using image processing. The ADE20K dataset includes segmented data for a variety of classifications. The dataset will be produced by removing photos from the "grass" class and producing masked versions of those images. By down sampling the picture and then up sampling it to preserve the most information through the layers, an encoder-decoder convolutional UNet architecture will used to train the dataset to reduce the computational cost for pixel wise convolution operations. The final objective is to fine-tune the model's hyperparameters and using different loss functions to maximize accuracy, and employ robust image processing techniques and models to accurately detect lawn spaces and edges of the lawn.

Reference code :

Network: milesial/Pytorch-UNet: PyTorch implementation of the U-Net for image semantic segmentation with high quality images (github.com)

Metrics : VainF/DeepLabV3Plus-Pytorch: Pretrained DeepLabv3 and DeepLabv3+ for Pascal VOC & Cityscapes (github.com)

2 Related Work

2.1 Fast Semantic Segmentation Model PULNet and Lawn Boundary Detection Method by Xiaoxia Li, Jingjing Chen, Yuanzheng Ye, Shunli Wang and Xueyuan Wang

The technique suggested in this study uses a novel semantic segmentation model called PULNet, which is based on a ResNet50 network to precisely detect lawn area and borders. The encoder model employs a pooling pyramid to increase the invariances of features to multi scale changes and image translation, and the ADE20K data is utilized for training. The model uses dilated convolution to expand the effective receptive field of the feature extraction network improving high-level semantic information and using conditional random fields to replace the SoftMax classification layer to improve the accuracy of classification. Each feature map is interpolated bilinearly to a size of 1/32 of the original picture, with the model's hyperparameters set as follows: The ADE20K training method utilized the same parameter settings as ResNet, with the input picture scaled to 512x512 and the number of categories set to 150, where 0 represents the background. An eight-neighbor coding method is designed to accurately locate the border of the lawn which concludes the research. This is a method for identifying grass borders using an 8x8 kernel that recognizes the shifting trend of pixel values along the margins. Experiments on the ADE20K dataset obtained the mean Intersection over Union (mIoU) and mean Pixel Accuracy (mPA) 32.86% and 75.65% respectively. The average speed is 82.7 frames per second on a platform with GTX 1080Ti GPU and as such can be used for fast and accurate lawn semantic segmentation with boundary detection.

2.2 Identification & Segmentation of Lawn Grass Based on color & visual texture classifiers by ALEXANDER SCHEPELMANN

The automated lawnmower CWRU Cutter can avoid obstacles on its own. Previously, the robot used LIDAR to locate obstacles, but the cost of the sensor makes its inclusion in commercial versions too expensive. Similar information may be obtained via cameras for a far lower cost, but first, valuable information must be gleaned from the incoming images. This can be computationally expensive. Furthermore, vision-based techniques may be extremely sensitive to changes in illumination. This thesis offers a technique for classifying grass based on color and visual texture that may be used outside. While texture measurements are based on edge detection and quantified using computationally affordable first and second order statistics, neighborhood-based color measurements are computed using the HSL color model. Individual measurements are then combined to create a binary representation of mow able terrain in an image. Performance is quantified by measuring recognition performance on a set of sample neighborhoods that contains common backyard obstacles. Edge detection, an image processing approach, is used to identify various boundaries based on discontinuities in color between a centered pixel and its surrounding neighbors. This way of detecting lawn borders is simple and effective. By employing a gradient throughout the area and graphing the magnitude at a central pixel, the discontinuities are computed. The edge reaction is bigger when the magnitude is larger. Another simple technique for recognizing things of interest is to use visual texture. Texture-based approaches show how to use texture to extract incredibly minute characteristics about specific objects for real-time, vision-based navigation and obstacle avoidance.

2.3 Vision-Based Mowing Boundary Detection Algorithm for an Autonomous Lawn Mower by Tomoya Fukukawa, Kosuke Sekiyama, Yasuhisa Hasegawa∗∗, and Toshio Fukuda

In this paper, a boundary detection technique for autonomous lawn mowers is proposed. For effective mowing, an autonomous lawn mower needs excellent moving precision. A vision system is used to tackle this issue by identifying the boundary between two locations, i.e., before and after grass mowing. The mowing border is vague and cannot be directly recognized. Therefore, as described above, we categorize the input picture of the lawn field into two areas using a texture classification algorithm with a bank of filters. Based on a chi-squared statistic, threshold processing is used to classify the data. Then, using Random Sample Consensus, the boundary line is found from the categorized regions (RANSAC).

This paper investigates mowing border detection for automated lawn-mowing applications. The system is based on the principles of texture classification and uses a number of filters to track changes in the grass boundary's mowing-related properties. The feature changes were recorded using a filter known as the Gabor filter. For the categorization of input photos from two areas, i.e., the images taken before and after the lawn-moving procedure, 36 such filters with a kernel size of 17 x 17 pixels were utilized. The findings demonstrated the capability of the suggested approach to identify boundary points and gauge a boundary line with some outliers. In the top portion of the photos, outliers frequently appear. This is a result of the distant locations' low resolution making it difficult to gather enough texture data.

2.4 Edge detection for weed recognition in lawns by Lorena Parraa, Jose Marinb, Salima Yousfic, Gregorio

Techniques for pre-processing images are essential for reducing computational time. One such method is reducing the image size. Prior to pre-processing, the grass characteristics from the photos were recovered as a baseline for this paper's method assessment. In turn, this aids in lowering the volume of processed data. The presence of weeds in the lawn was determined using the grass data and edge detection. Techniques for smoothing and post-processing are used to improve accuracy and decrease inaccuracy. The suggested technique is meant to be applied in drones that record video and transmit data about the location of the weeds. A method for detection in control vehicles of weeds management systems was suggested by the operational algorithm. Low-cost and real-time operations are prioritized in the work that has been done, and it may be expanded to detect a number of additional grass disturbances caused by illnesses or natural phenomena. Another topic of concern is performance in various lighting scenarios. Following the identification of vegetation pixels, weed detection is often accomplished using processing methods by combining data on the variations in color, form, texture, location, size, and spectrum between weeds and crops. This study suggests the best edge detection and post-processing strategies for weeds and turfgrass presence detection with a threshold value. Up to 12 different edge detection filters were tested, and the results of the experiments were compared. The three filters that produced the most encouraging findings were chosen. Then, multiple post-processing alternatives were assessed, including various mathematical operators and cell sizes for aggregation methods. The optimal threshold value was then tested using fresh photos and various indices that consider false positives and false negatives.

2.5 Analysis of Hybrid Classification Approach to Differentiate Dense and Non-dense Grass Regions by Sujan Chowdhury, Brijesh Verma & David Stockwell

To distinguish between dense and non-dense zones, vegetation categorization using aerial or satellite images from video data acquired in an outside setting is studied. Texture feature and several classifiers, such as Support Vector Machine (SVM), Neural Network (NN), k-Nearest Neighbor (k-NN), and Naive Bayes, are employed therein. The basis classifiers are the machine learning algorithms. In the actual world, in addition to the presence of vegetation borders, there are other hurdles that call for further classifier tuning and effective segmentation. The examination of a hybrid classification strategy used in this paper to identify vegetation, namely the kind of roadside grasses, using films taken by the Queensland transport and main roadways is presented. From roadside video data, the suggested framework can discern between dense and open grasslands. While the majority of current research has been on infrared pictures, the suggested technique employs image texture features to classify vegetation regions. The primary contribution of this research study is an analysis of the hybrid strategy combining texture feature and several classifiers. For training and testing purposes, several pictures were produced from video footage comprising varying situations of roadside vegetation.

2.6 U-Net: Convolutional Networks for Biomedical Image Segmentation by Olaf Ronneberger, Philipp Fischer, and Thomas Brox

There is widespread agreement that deep networks need thousands of annotated training samples to be successfully trained. This paper demonstrates that a network of this type can be trained end-to-end from a relatively small number of pictures, outperforming the previous best technique (a sliding-window convolutional network) on the ISBI challenge for segmenting neural structures in electron microscopic stacks. Training With Caffe's stochastic gradient descent implementation, the input pictures and their related segmentation maps are utilized to train the network. A sound initialization of the weights is crucial in deep networks with several convolutional layers and varied network routes. This may be done by selecting the starting weights from a Gaussian distribution with a gamma distribution for a network with this design (alternating convolution and ReLU layers). Convolutional networks are frequently employed for classification tasks in where the output to an image a single class label is. The intended output, or the assignment of a class label to each pixel, should incorporate localization in many visual tasks, particularly in biomedical image processing. Thousands of training photos are typically out of reach for biological jobs as well. The fundamental concept is to add additional layers to a typical contracting network, replacing the pooling operators with up-sampling operators. As a result, the output's resolution is increased by these layers. High resolution characteristics from the contracting path are mixed with the output that has been up-sampled in order to localize. Based on this knowledge, a subsequent convolution layer may subsequently learn to put together a more exact result. The u-net (averaged over 7 rotated versions of the input data) achieves without any further pre- or postprocessing a warping error of 0.0003529 and a rand-error of 0.0382.

2.7 Focal Loss for Dense Object Detection by Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár

The most accurate object detectors available today are built on the R-CNN-popularized two-stage method, which involves applying a classifier to a small collection of potential object locations. Contrarily, one-stage detectors that are used across a consistent, dense sample of probable item locations have the potential to be quicker and easier, but up to now, they have lagged behind the accuracy of two-stage detectors. This study looks at the reasons why this is the case. The main contributing factor is the high foreground-background class imbalance that was observed during training of dense detectors. This study suggests altering the standard cross entropy loss to lessen the weight given to losses allocated to cases with clear classifications in order to remedy this class imbalance. The innovative Focal Loss method protects the detector from being overloaded with easy negatives during training by concentrating training on a small number of challenging cases. It also proposes a straightforward dense detector called RetinaNet to assess the efficacy of this loss. Results show that when trained with the focal loss, RetinaNet is able to match the speed of previous one-stage detectors while surpassing the accuracy of all existing state-of-the-art two-stage detectors. The Focal Loss is designed to address one-stage object detection scenarios in which there is an extreme imbalance between foreground and background classes during training (e.g., 1:1000). An image's total focus loss is calculated as the focal loss added over all 100k anchors, normalized by the number of anchors allocated to a ground-truth box. Since the majority of anchors are straightforward negatives and experience little loss values due to the focus loss, it is then normalized by the number of assigned anchors rather than the total number of anchors.

3 Methods

3.1 Preprocessing

The proposed methodology starts with an RGB image as input which is converted into an image of size 540x500 to maintain a uniform size before feeding it to the model. As the input image may consist of some noise and we are only interested in the identifying the lawn space and the boundaries of the lawn, the image is blurred using a median filter and a Gaussian filter to denoise the image. A median filter is used to preserve the edges and remove some high intensities noise in the image, followed by a Gaussian blur to smoothen the image over the edges and retain information around the edges which is crucial for us.

3.2 Model

A U-net model architecture is used to perform binary sematic segmentation of images to classify lawn spaces. The input to the model is a pre-processed image of size 540x500. The model output consists of two classes namely, background and lawn. The output of a model is a 2x540x500 image with each channel. To categorize lawn spaces, binary semantic segmentation is performed on input images using a U-net model. A 540x500-pixel pre-processed image serves as the model's input while the model's output consists of two classes namely, “background” which is class - 0 and lawn which is class -1. The model's output is a two-channel, 540x500 image with the first channel corresponding to class 0 and the second channel corresponding to class one.

3.2.1 Dataset

The ADE20K dataset consists of more than 20,000 detailed semantic segmentation images and 150 different classes are present in the dataset, which includes both interior and outdoor images of grass, sky, people, and roads. It is one of the most used datasets for training models for segmentation problems. The dataset used comprises of RGB images as input and masked images as the target output. The dataset uses polygon annotation to mark the boundary of a particular category. The target class is grass for our study, for which the images which had grass category in it were extracted. The polygon datapoints of this category were used to create masked images for each input image. The masked image was a grayscale image with pixels with values 0 and 1 belonging to the background class and grass class, respectively. The dataset consisted a total of 3171 images . Fig 1 shows a selection of the dataset's input images together with the corresponding masked images. For viewing purposes, the masked images' pixel values are multiplied by 255.

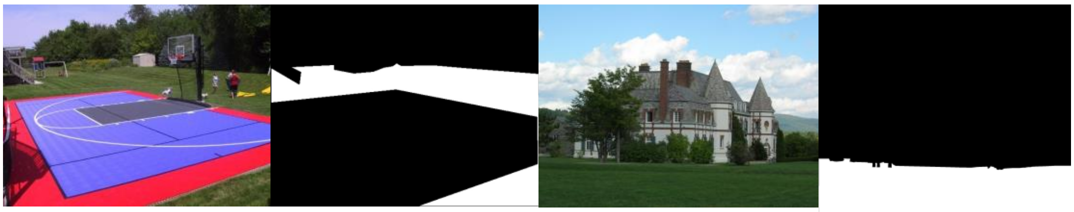

Figure 1 : Dataset images

3.2.2 Network Architecture

Fig. 2 presents the proposed U-net network architecture used for training images for semantic segmentation. A similar network architecture is proposed in [6] where the model is trained for biomedical image segmentation. As seen in Fig 2 the model comprises of two parts, encoder, and decoder. The encoder path follows a similar architecture as a convolution network with convolution and pooling layers. Each encoder layer consists of two 3x3 convolution that is followed by a batch normalization , rectified linear unit (ReLU) activation function and a maxpooling operation with a stride of 2 which down samples the input a factor of two. It is a common practice in deep learning to follow this flow where convolution is followed by batch normalization, an activation function and finally ends with a maxpooling layer. Each convolution layer is convolved with a 3x3 kernel.

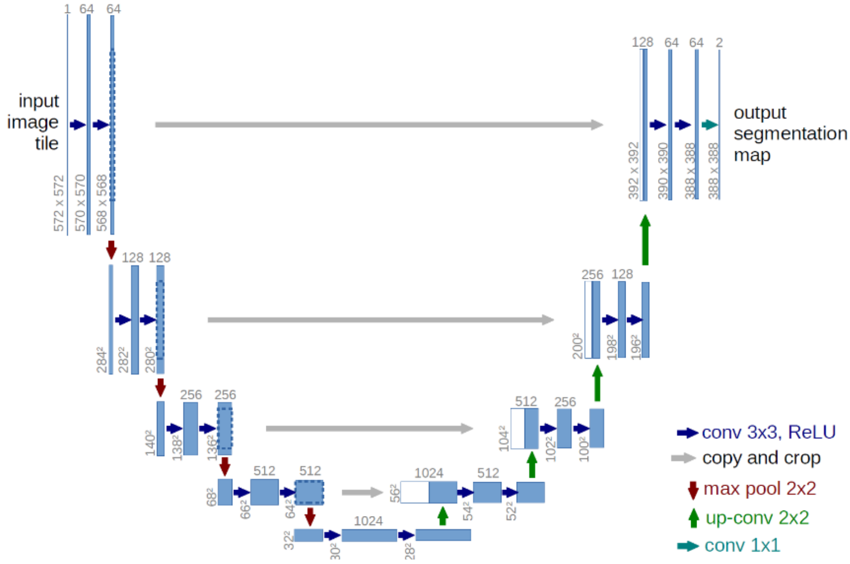

Figure 2 : U- network architecture

Batch normalization is used to standardize the input in each layer and follows a standard normal distribution across a batch. This helps the model to be more stable through multiple layers and trains the neural network faster. Fig 3 shows the graph for the mathematical expression for a ReLU activation function where any negative input causes the function to return 0, while a positive input returns the same value which is expressed as f(x)=max(0,x)

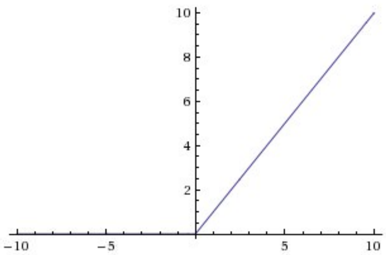

Figure 3 : RELU activation

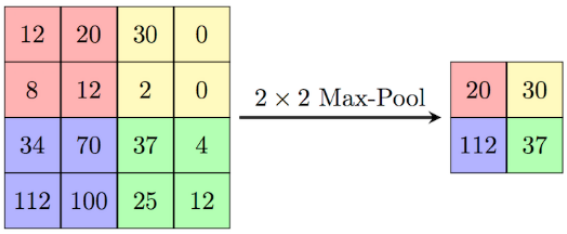

Figure 4 : Maxpooling (2 x 2)

A ReLU activation function is used to account for non-linearity and make the model more flexible. A maxpooling layer of kernel size 2x2 is used the down sample the image by a factor of two and increase the number of feature channels by two at each down sampling step. Fig 4 shows the maxpooling operation where within the kernel window the maximum value is chosen for the respective kernel. The network's decoder component, which is visible in the right portion of the Fig 2, is used to up sample the feature map that was previously down sampled. The feature map is up-sampled by a factor of two at each phase of the up-sampling process which is followed by a convolution with a 2x2 kernel size, which reduces the number of feature channels by half. The feature map from the corresponding encoder layer is concatenated with the feature map produced after passing it through a 2x2 convolution thereby doubling the feature maps. This feature map is passed through two 3x3 convolution layer and a ReLU activation layer.

A 64-feature vector generated by the final layer is mapped to the model's output, which consists of the classes 0-background and 1-grass. The feature vector is passed through a 1x1 convolution before being mapped to the output of the model.

3.2.3 Training

The network was trained with a total of 3171 images where 2537 (80%) images were used for training and 634 (20%) images were used for validating. Each input image had a corresponding segmentation mask of the same size with each pixel belonging to class 0-background or class 1- grass. The segmentation masked were used as target images for predicting the output. Table 1 shows the hyperparameters that were used to train the model. The input to the model is 540x500 RGB image.

Table 1 : Training Hyperparameters

Parameter	Value	Parameter	Value
Epochs	20	LR	$4x10^{-5}$
Batch Size	8	Input Size	540x500
Momentum	0.9	Decay Rate	$1x10^{-8}$

During training of the model, a learning rate of $4x10^{-5}$ was chosen as the optimal learning rate after training using learning rates in the range of $1x10^{-3}$ to $1x10^{-6}$. An optimizer called RMSProp was used to help in gradient descent when finding optimal weights, that is comparable to the Adam optimizer, a popular optimizer for deep neural networks. RMSProp address the issue of vanishing gradient problem i.e., gradient of neural networks tends to vanish or explode as the data propagates through the network. RMSProp tackles this problem by considering the moving average of squared gradients to normalize the gradient. This helps in decreasing the step of large gradients to prevent exploding and increasing the step for small gradients to prevent vanishing. The optimizer used a momentum of 0.9 and decay rate of $1x10^{-8}$ for the gradients. [7] Proposes the focal loss, a loss function that aids in resolving the issue of an unbalanced dataset. One of the classes may occasionally dominate the gradient and account for the majority of the loss in a dataset for binary classification. These negatives are easily classified examples which prevent the model on concentrating hard examples. To resolve this issue to give less preference to a class which can easily be classified and focus training on the hard examples of the target class, a modulating factor is introduced to the cross-entropy loss. The modulating factor is defined as
$(1 - p_t)^{\gamma}$ which is added to the cross-entropy loss, where γ is a tunable focusing parameter and is defined as $\gamma \geq 0$.Focal loss is defined as:

$$FL(p_t) = -(1 - p_t)^{\gamma} \log(p_t).$$

The focusing parameter γ can be tuned to make the model pay more attention to hard examples. Finally, a more balanced definition for Focal loss was used as :

$$FL(p_t) = -\alpha t(1 - p_t)^{\gamma} \log(p_t).$$

Where α can be defined as a weighing factor that belongs in the range of $\alpha \in [0, 1]$ for class 1 and $1-\alpha$ for class 0. The output of a network is an image with two channels dealing with two classes as the output. The output single channel image is obtained by taking the argmax of the predicted model output across the channel dimension, where pixels with values of 0 and 1 correspond to the background class and the grass class, respectively. The whole network was developed using PyTorch as the framework and trained for 20 epochs using a Tesla K80 GPU.

3.3 Postprocessing

From the model a single channel output is obtained which is multiplied with 255 to make it a binary image. The pixels classified as grass will have a value of 255 and the pixels classified as background will have a value of zero. As semantic segmentation deals with pixel classification improper classification along the edges and within the classification space introduces lot of noise in the thresholded image.

Figure 5 : Model output

In Fig 5 it can be observed that the boundaries are pixelated along the edges of the lawn space and due to improper classification, it can be observed that few pixels are classified as 0 within the lawn space. To solve this problem, we need to merge the boundaries of the thresholded image and make it more uniform. The thresholded image is dilated using a 5x5 kernel. Dilation is used to increase the size of the pixels, which results in the range of the pixels' value increasing from 0 to 255 based on the value of the surrounding pixels. The pixels are now connected to each other and not separated by 0 pixels. This solves the problem of small pixels which are classified as the background within the lawn space because, due to dilation of surrounding pixels these pixels are not completely 0. Applying a median and gaussian blur filter to the image can further blur the edges and reduce noise. The edges become more regular and blend into the picture. Since the lawn's bounds don't need to be expanded while maintaining connectivity between pixels at the lawn's margins and preventing them from being separated by 0 pixels, erosion is applied on the picture pixels to reduce their size. The image is further thresholded so that any pixel values larger than 80 are equal to 255 in order to clearly delineate the lawn space and to facilitate subsequent processing. The resultant images now have quieter lawn areas with more consistent edges. The coordinates of the lawn space can now be kept as the location of the pixels in the image with the value of 255. This thresholded image is used to further extract the coordinates of the edges of the lawns. A Canny edge detector is used which works on the principle of finding changes in the gradient of an image to define an edge. The Canny edge detection technique is a multi-step process that includes noise reduction, determining the intensity gradient of the image, using non maximum suppression to remove unnecessary pixels, and hysteresis thresholding to determine which edges are all actually edges, and which ones are not. The last phase involves using contour detection to find the edges and determine their locations. In this way the lawn edge's pixel coordinates are extracted. Fig 6 (a) shows the raw thresholded output from the model. The image after applying erosion, dilation, and blurring is seen in Fig 6 (b). One can observe that edges are smoother and more defined. Fig 6 (c) depicts the edges found in the image after applying a Canny edge detector and performing contour detection.

Figure 6 (a) : Model output

Figure 6 (b) : Smoothened edges

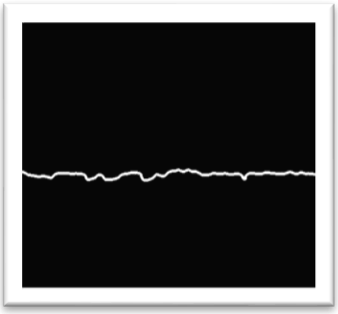

Figure 6 (c) : Edge detection

4 Results

The results of the proposed method can be evaluated by evaluating the metrics of the trained model and precision of defining the edges of the lawn. Table 2 summarizes the result of the model in terms of the three metrics that were calculated to evaluate it i.e., mean pixel accuracy (mPA) , dice score and mean intersection over union (mIoU).

Table 2 : Model Results

Mean Intersection over Union (mIoU)	0.74
Mean Pixel Accuracy (mPa)	0.82
Dice Score	0.576

The given values are the mean pixel accuracy (mPA), Dice score, and mean Intersection over Union (mIoU) for an image segmentation model. The mean pixel accuracy is the average pixel accuracy across all images in a dataset, with a value of 0.82 indicating that the model is correctly classifying pixels in images with an accuracy of 82%. The Dice score is a measure of the similarity between the predicted and ground truth segmentations, with a value of 0.576 indicating that there is a moderate level of overlap between the two sets of data. The mIoU is the average intersection over union for all classes in a dataset, with a value of 0.74 indicating that the model is correctly identifying the areas of overlap between the predicted and ground truth segmentations with an accuracy of 74%. The intersection over union for each individual class was calculated as 0.903 for class – 0 background and 0.583 for class -1 grass. The model is performing well for class 0, but it is not performing as well for class 1. This may be related to class 0 slightly dominating the gradient, contributing to the loss, and forcing the model not to learn challenging instances for lawns. Compared to the results in [6] a similar architecture was used to train a model for biomedical image segmentation, this paper achieved an mIoU of 0.92 and [1] uses a PULNet model trained on the ADE20 dataset and achieves an accuracy of 0.96. For evaluating the lawn boundary detected calculating a bounding loss as described in [8] but did not achieve conclusive results to be reported. Fig 7,8,9 shows the model output for the input image in the second column and the last column shows the final output with the lawn space defined and green and the lawn boundary marked in blue. We can observe clear lawn spaces defined and some noise is removed inside the lawn space which are not classified accurately as seen in the Fig 7. The lawn boundary is smoother as seen in the final result of the third column as compared to the output of the model seen in the second column.

Overall, when the model is evaluated , these values indicate that the image segmentation model is performing well, with good accuracy for pixel classification and reasonable overlap between the predicted and ground truth segmentations but has some exceptions where the model struggles to predict with an imperfect lawn as seen in Fig 9, where the lawn is surrounded by weeds and the weeds are predicted as grass. In Fig 8 the rear part of the vehicle which is green is predicted as grass which is a false positive prediction.

Figure 7 : Test set 1

Figure 8 : Test set 2

Figure 9 : Test set 3

5 Conclusion and Future Scope

This project proposes a deep learning approach for training a model to detect lawn spaces. The use of segmented and masked images, along with the encoder-decoder convolutional neural network architecture, allows the model to accurately identify lawns despite variations in grass type, size, color, texture, and season. Exact pixel coordinates of the lawn space are recovered and precisely characterized using the classification of grass. Using the model's output, the boundary of the grass is located using image processing techniques including Canny edge detection, contour detection, and morphological operations such as erosion and dilation. This approach achieved good results with a few exceptions and has the potential for further improvement by introducing more diverse datasets and optimizing the model's architecture. In the future, the project could expand to include other applications of deep learning in the realm of landscaping and outdoor space management. The use of different backbone model encoders, such as ResNet, DenseNet, and MobileNet, could improve the accuracy of the model. Additionally, the use of boundary loss to evaluate the boundary detection of lawns and models like active contour detection could help to estimate the lawn boundary more accurately. Overall, the project shows promise in using deep learning to address the challenge of identifying the boundaries of a lawn space.

References

[1] Li, X., Chen, J., Ye, Y., Wang, S., & Wang, X. (2021, February). Fast Semantic Segmentation Model PULNet and Lawn Boundary Detection Method. In Journal of Physics: Conference Series (Vol. 1828, No. 1, p. 012036). IOP Publishing.

[2] Schepelmann, A. (2010). Identification & segmentation of lawn grass based on color & visual texture classifiers (Doctoral dissertation, Case Western Reserve University).

[3] Fukukawa, T., Sekiyama, K., Hasegawa, Y., & Fukuda, T. (2016). Vision-based mowing boundary detection algorithm for an autonomous lawn mower. Journal of Advanced Computational Intelligence and Intelligent Informatics, 20(1), 49-56.

[4] Parra, L., Marin, J., Yousfi, S., Rincón, G., Mauri, P. V., & Lloret, J. (2020). Edge detection for weed recognition in lawns. Computers and Electronics in Agriculture, 176, 105684.

[5] Parra, L., Marin, J., Yousfi, S., Rincón, G., Mauri, P. V., & Lloret, J. (2020). Edge detection for weed recognition in lawns. Computers and Electronics in Agriculture, 176, 105684.

[6] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

[7] Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (pp. 2980-2988).

[8] The Kervadec, H., Bouchtiba, J., Desrosiers, C., Granger, E., Dolz, J., & Ayed, I. B. (2019, May). Boundary loss for highly unbalanced segmentation. In International conference on medical imaging with deep learning (pp. 285-296). PMLR.

[9] Baştan, M., Bukhari, S. S., & Breuel, T. (2017). Active Canny: edge detection and recovery with open active contour models. IET Image Processing, 11(12), 1325-1332.