Geospatial Grass Segmentation for Lawn Sizing

Joel Northrup[1], Jayden B. Pye[2], Matthew Naples[3]
[1]University of Iowa, Department of Statistics and Actuarial Science, Iowa City, USA
[2]University of Iowa, Department of Industrial and Systems Engineering, Iowa City, USA
[3]University of Iowa, Department of Statistics and Actuarial Science, Iowa City, USA
{Joel-northrup; jayden-pye; matthew-naples}@uiowa.edu

May 15, 2020

ABSTRACT

Many lawn care companies in the United States price their services based on the area of grass on a property. Lawn care contractors must give quick and accurate quotes to potential clients in order to streamline sales. Furthermore, manually measuring a lawn takes time and requires professional training. With the advent of online measuring tools, lawn care contractors today are able to measure a lawn's size remotely. Nevertheless, this is still a time consuming task. Additionally, there exist possibilities for human error when measuring the lawn remotely by satellite. One potential solution to generating a quick and accurate quote to a client would be implementing a deep learning model to predict lawn size. As a first step to predicting lawn size for homes in the United States, we decided to implement the Mask R-CNN model built by the Facebook Research Institute. The Mask R-CNN model identifies instance segmentation on a pixel level. Using the pre-trained weights of the same model trained on the COCO dataset, we will use transfer learning to find masks that appropriately cover the lawns in our dataset. Our data consist of manually labeled satellite images of residential properties in the Cedar Rapids area taken from Google Maps. The output of the model is a pixel mask layer over the lawn which can then in a later project be converted into an area measurement in square feet. This square footage can then be used by lawn companies to quickly and accurately price their services for clients simply with a satellite image of their property.

I. INTRODUCTION

Lawn sizing is an essential business process to the lawn care industry, as it helps price their services to clients. Manual lawn sizing requires training, takes time, and ultimately costs money. In addition, quickly quoting clients helps streamline sales for contractors and allows them focus their time and resources elsewhere. Predicting the size of their lawn utilizing deep learning is a fast and labor free solution to quote clients. Ultimately, we would like to be able to predict the lawn size of any property in the United States. This project is a first step in the process of achieving that goal. It is a similar approach to a smaller version of the problem. We want to predict which pixels of an image correspond to a lawn (which can in a future project be used to find the area of the lawn) using images of residential properties in the Cedar Rapids area (as opposed to the United States as a whole). We discuss later how we can extend this project to solve the bigger problem. We use the Mask R-CNN model developed by the Facebook Research Institute to predict lawn pixels. In addition, pretrained weights from a Mask R-CNN model trained on the COCO dataset will be applied to our model using transfer learning. The dataset consists of satellite images of residential properties in the Cedar Rapids area taken from Google Maps. Each image will be represented by a three dimensional tensor with pixel intensity values in each color channel. Additionally, each image label will be a two-dimensional tensor known as a mask with each cell representing a pixel image. A value of 1 corresponds to a given pixel being identified as grass and a 0 corresponding to a pixel that is not. These mask labels for the training dataset will be done by our team using RectLabel.

The model output will be the same format as the training label input. Each cell and corresponding value will have the same interpretation as the training labels given. The output mask layer can be aggregated into a pixel area and then converted from pixels to square footage to accurately estimate the area of a property's lawn. This predicted square footage can ultimately be used by the lawn care industry to quote a potential client. All this can be done with a simple satellite image of the client's property.

II. RELATED WORK

A. Mask R-CNN

The mask region based convolutional neural network model developed by the Facebook Research Institute implements two object detection techniques, a bounding box and pixel mask. The model extends the Faster R-CNN model, the bounding box approach, and adds a branch for predicting an object mask in parallel. The Mask R-CNN model has 3 outputs for each candidate object; a class label, a bounding-box offset, and the object mask [2]. The Mask R-CNN model is both intuitive and simple to train whilst only a small increase in computational expense when compared to the Faster R-CNN. One key component to the Mask R-CNN, however, is the inclusion of pixel-to-pixel alignment called RoIAlign. This feature of the model helps address the misalignments between the region of interest and extracted features of the output mask by preserving the explicit per-pixel spatial correspondence of the input for the region of interest [2]. Nonetheless, the model outperforms all existing single model entries on all tasks, including the COCO 2016 challenge winners [2].

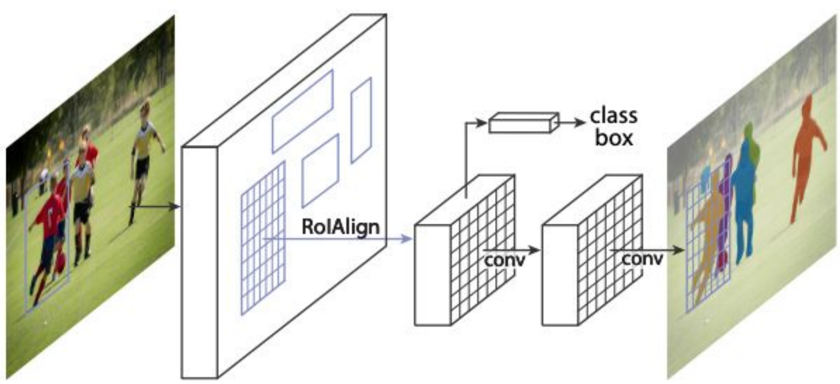

Figure 1. Mask R-CNN Model [2]

B. Transfer Learning

In many applications, training a machine learning model requires a sufficient amount of data and computational time to train on that specific data. However, one way to address the issue of not having enough data or enough computational power is to implement transfer learning. The core idea of transfer learning is to use experience gained in learning one task to help improve learning performance in a related, but different, task [5]. The logic behind transfer learning is that the earlier layers generalize a task while the later layers are more specialized for that given task. Thus, using the earlier layer weights on a similar task and training on the later layers will save computational time and improve model performance. The way this methodology can be implemented is on a case by case basis. That is, the success of transfer learning from one task to another is based on several factors such as task similarity, transferable learning layers of the model, or the selected model's input and output [5]. The process of implementing transfer learning, however, is quite simple. First, select a trained model to transfer its weights to the new model implementation. Then, select the layers, usually the last few, to train on the data set for the required task. Finally, train the model using the preloaded weights.

III. DATA

A. Data Collection and Annotation

As there does not exist an annotated lawn satellite image dataset, we collected and annotated our own data. To obtain the satellite images for our dataset, we utilized Google Maps. We then annotated the satellite images with the software program RectLabel, drawing polygons over areas that we recognized as grass. These annotations were saved as .xml files. This process of collecting the lawn satellite data and annotating is quite time consuming. Thus, we only collected and annotated 69 images. This small dataset size, however, is sufficient for this application as we address the issue by implementing data augmentation.

B. Data Augmentation

We used the following methods to augment images: rotation about the image center, horizontal and vertical translation from its original position, changing the image scale, flipping the image, a four point perspective transform, and linear contrasting (applying the following linear scaling to each pixel value 'v' in the image: 127 + alpha*(v-127), where alpha ranged between .8 and 1.2). The transformations were applied to the images in the order they are described.

Figure 2. Augmented Satellite Images

In order to implement these augmentation methods, we used methods from the imgaug library. To integrate these functions into our model, we defined a few custom methods and instance variables in Matterport's "Dataset" [1] class such that augmenting the data is as simple as calling augment_data() on an instance of the GrassDataset class and specifying the number of images to augment, the number of augmentations per image, and an imgaug Sequential augmenter object.

III. METHODOLOGY

A. Model Configuration

We implemented the Mask R-CNN model based on the existing implementation by Matterport Inc. released by MIT under their license [1]. The model itself is based on the open-source libraries Keras and Tensorflow. The set number of classes predicted was set to two classes, 'grass' and the background image. We found that our initial learning rate of 0.006 was slightly high for this application and would result in the exploding gradient problem while training. After training our model many times, we found that the ideal parameters for our model were a learning rate of 0.0009 and 131 training steps per epoch. The minimum confidence level for detection was 90%. This level seems to be a fair assumption in this application and generated satisfactory results when detecting grass. Lastly, the maximum number of ground truth instances was 10. Meaning, the model will only detect 10 objects per image. Given that there is only one class to detect and there are usually only 1 - 4 objects to detect per image, limiting the detection to 10 objects was a logical way of reducing the training time. The rest of the model configurations are taken from the base configurations for Matterport's implementation of the Mask R-CNN.

B. Loading Dataset

Our dataset needed to be manipulated so that it matched the required model inputs. To do this we created a GrassDataset class to define how the model should load our data. First, we defined a function that allowed for the reading of the annotation .xml files. This function returned the image's respective height, width, and depth as well as the mask layer coordinates in the form of a list of polygons (specifically, a list of Polygon objects from imgaug.augmentables.polys [4]). Next, we wrote our own load mask function that used these polygons to return the mask layers in the form of a boolean or binary mask where a value of 1 corresponds to that pixel being identified as grass and a 0 corresponding to a pixel that is not. Additionally, other functions were defined for the bounding box region, loading the images, augmenting the dataset, and increasing the number of arguments when adding an image to our dataset. After loading the data, the dataset was split into a training and testing set. Following this step, we ran our augmentation function discussed in the previous section to increase our training set size and reduce overfitting on our limited dataset.

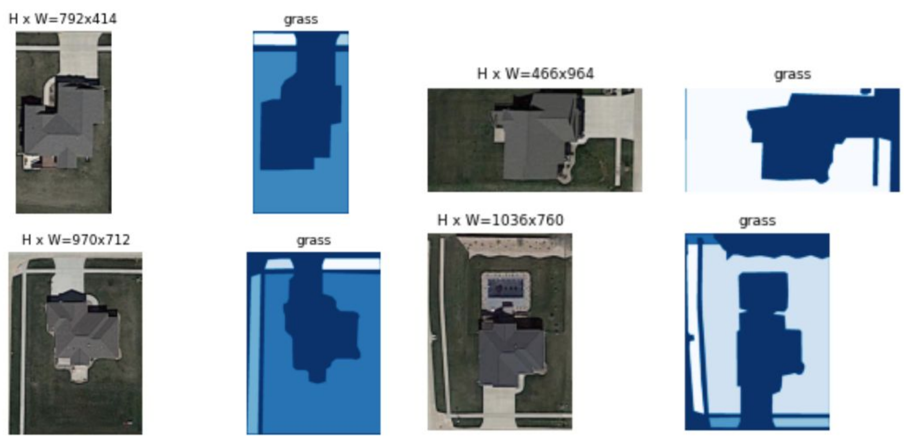

Figure 3. Examples of Loaded Images and Mask Labels

C. Training the Model

The implementation of the Mask R-CNN model discussed in this report uses transfer learning to decrease the computational load whilst training the model and increase overall model performance. First, we loaded the pretrained weights from the MS COCO dataset to our model, excluding the last few layers. The excluded layers are as follows, mrcnn_class_logits, mrcnn_bbox_fc, mrcnn_bbox, mrcnn_mask. These layers are not included in the transferred model weights as it enables the model to match the number of classes to the new data set. The model will be only trained on the head layers as the earlier layers are already trained to generalize features and object geometry within an image. Additionally, training only the head layers decreases the training time and computational load without significantly decreasing model performance. Finally, the model was trained at a learning rate of 0.0009 and with 100 epochs.

We trained our model on google colab pro with a single GPU. The gradients were clipped to 5.0 and weights are decayed by 0.0001 each epoch.

IV. RESULTS

The model reported an average Intersection over Union (IoU) of 62.42% on our test dataset. This metric reflects the overlap of the predicted mask layer and the true mask layer and therefore is a good measurement of performance for this application. Given this value, we are pleased with the model's initial performance. Additionally, the mean Average Precision (mAP) was 86.02% and was also the metric used to measure model performance for the MS COCO challenge. This measurement of performance includes the IoU as well as the Precision and Recall. Precision measures how accurate the model predictions are while Recall measures how well the model finds each positive case. The reason our model performs very well for mAP is that there are only two potential predictions, grass or not grass. Nonetheless, the model performs very well overall. Below are sample outputs of our model from our validation set.

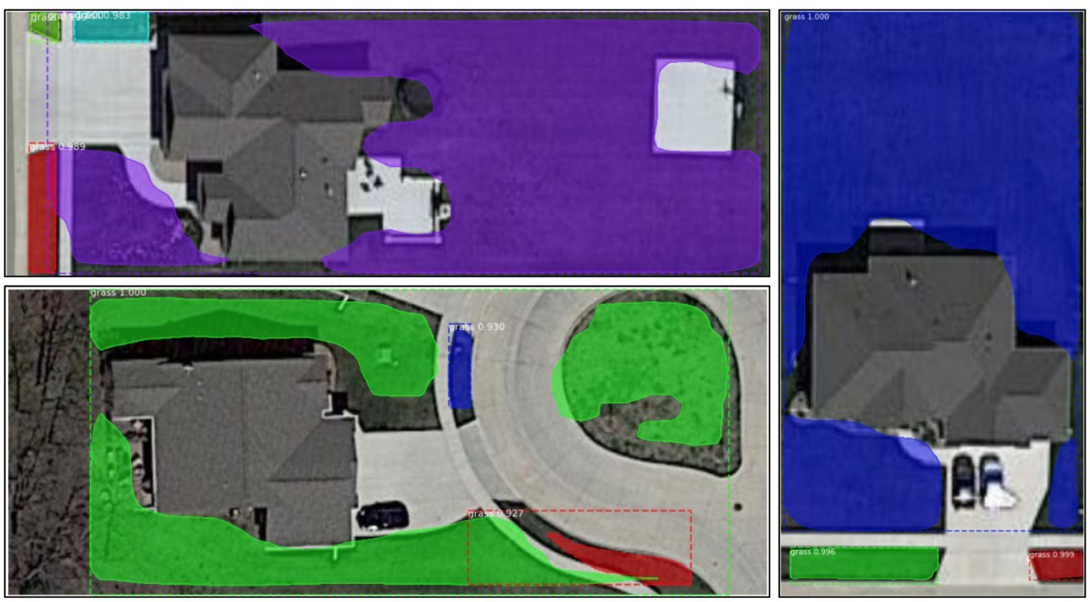

Figure 4. Sample Outputs of Test Images

V. DISCUSSION

While our implementation of the Mask R-CNN model performed remarkably well, there are a few ways to improve on our results and expand our scope of applicability. We could have increased the size and variability of our data in two ways: (1) increase the size of the original annotated data, and (2) include more methods of augmentation. Increasing the size of the dataset, as well as the number of relevant geographic regions from which that data is drawn, could increase model performance on any given test data from the United States. To expand the dataset, researchers could stratify the target population (lawns in the United States) into relevant geographic locations likely to have a similar "lawn types" (which can be specified by relevant factors such as lawn size, grass type, etc.) and randomly gather images from each stratification.

In addition to increasing the size of the dataset with real images, we also believe that other augmentation techniques would increase performance, specifically “thresholding”. Thresholding is a process where pixels beyond a certain intensity threshold are rendered as white pixels, and all others black. If thresholds are set properly, the dividing edge between “lawn” and “not lawn” may be more clearly delineated. This method of augmentation has increased accuracy in another project using similar data [6]. In addition to image quantity, We also have to consider the image quality. All images were collected by taking screenshots within Google Maps. Because of this, some of our images are blurry or not very clear. In the opinion of these authors, higher quality images would contribute to better accuracy in our model.

Another concern with our model is the prediction of grass in a property that has a considerable amount of foliage or tree cover. For properties that have considerable foliage, our model would have a difficult time predicting lawn square footage. This is not overly concerning, however, as this is also a problem that a human would encounter while measuring a property via satellite images.

VI. CONCLUSION

Overall, the results from our implementation of the model are promising but not sensational. They were satisfactory given the limited amount of data at our disposal. The model is close to being able to solve the problem of accurately masking lawn from properties in the Cedar Rapids area, which is an adequate first step in solving the ultimate problem of predicting lawn size of any house in the United States. We’ve discussed a few steps to broaden our solution to fit the scope of the larger problem.

ACKNOWLEDGEMENT

The authors of this report, Joel Northrup, Jayden Pye, and Matthew Naples contributed equally to the content of this report and work. This project was completed to satisfy the requirements of ISE:6380 Deep Learning at the University of Iowa by Dr. Stephan Baek.

REFERENCES

[1] Abdulla, W. (2017). Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. Retrieved February 28, 2020, from https://github.com/matterport/Mask_RCNN

[2] He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017, October 22). Mask R-CNN.Retrieved March 14, 2020, from https://research.fb.com/publications/mask-r-cnn/

[3]Johnson, J.W. (May 1, 2018). Adapting Mask-RCNN for Automatic Nucleus Segmentation. Retrieved March 14, 2020, from https://arxiv.org/abs/1805.00500

[4] Jung, A. (2020, February 3). Augment Polygons. Retrieved May 2, 2020, from https://nbviewer.jupyter.org/github/aleju/imgaug-doc/blob/master/notebooks/B03%20-%20Augment%20Polygons.ipynb

[5] Taylor, M., & Stone, P. (2009, July 9).Transfer Learning for Reinforcement Learning Domains: A Survey. Retrieved April 29, 2020, from http://www.jmlr.org/papers/volume10/taylor09a/taylor09a.pdf

[6] Wilkins, J., Nguyen, M.V., & Rahmani, B. (2020, April 22). Image Processing Failure and Deep Learning Success in Lawn Measurement. Retrieved May 1, 2020, from https://arxiv.org/abs/2004.10382