

IMAGE PROCESSING FAILURE AND DEEP LEARNING SUCCESS IN LAWN MEASUREMENT

J. Wilkins¹, M. V. Nguyen¹ and B. Rahmani^{1, 2}

¹Math and Computer Science Dept., Fontbonne University, St. Louis, MO, USA

²Computer Science and Data Science Dept.,
Maryville University, St. Louis, MO, USA

ABSTRACT

Lawn area measurement is an application of image processing and deep learning. Researchers used hierarchical networks, segmented images, and other methods to measure the lawn area. Methods' effectiveness and accuracy varies. In this project, image processing and deep learning methods were used to find the best way to measure the lawn area. Three image processing methods using OpenCV compared to convolutional neural network, which is one of the most famous, and effective deep learning methods. We used Keras and TensorFlow to estimate the lawn area. Convolutional neural network or shortly CNN shows very high accuracy (94-97%). In image processing methods, thresholding with 80-87% accuracy and edge detection are the most effective methods to measure the lawn area while the method of contouring with 26-31% accuracy does not calculate the lawn area successfully. We may conclude that deep learning methods, especially CNN, could be the best detective method comparing to image processing learning techniques.

KEYWORDS

Lawn Measurement, Convolutional Neural Network, Thresholding, Edge Detection, Contouring.

1. INTRODUCTION

While there are definitely major business applications for lawn care companies, this project is a great experiment into an area of image processing that is rarely looked at: image regression. So, the project had to be started nearly from scratch, and creative ways were found to deal with the limited quality and quantity of the samples.

At the lawn care company, the lawn area of houses from satellite images had to be measured. The process is often rather simple: find the address of the house, use online software to measure the area of the lawn of the house, record the measurements, and repeat the steps. While the company sends out people to manually measure the lawn area, they want some rough estimate beforehand in areas. It is a very repetitive job, but it could only be done by a human. So, the lawn care companies have to pay the person at least minimum wage, and the person has to spend many hours trying to measure hundreds of houses. Therefore, the company is spending thousands of dollars and inordinate amount of time to measure houses. Eye-strain and fatigue are another issue that causes inconsistency and low accuracy. The lawn care company could save thousands of dollars and hundreds of human hours in measuring the lawns of houses accurately (at least within a margin of error).

Weiqi Zhou and his colleagues in 2008 tried to measure the lawn properties including lawn and house area remotely. They used hierarchical networks and classified segmented images for this purpose [1]. Alexander Schepelmann in his master thesis, measured the lawn area using color and visual texture classifier [2]. One of the methods that could be used by the lawn care company is artificial neural networks. Artificial neural networks have a long winding history in the computer science field. While the idea of a neural network in general has its basis in biology and did influence artificial neural networks, their mathematical conception dates back to 1943 in a paper of Warren S. McCulloch and Walter Pitt [3]. In 1950s, several attempts were made to simulate a neural network by Clabaugh, et. al. In 1959, Stanford researchers Bernard Widrow and Marcian Hoff successfully implemented it in an algorithm [4]. They removed echoes and other data corruption from phone lines by predicting the next actual bit. In the 1980s, neural networks were revived as a topic of interest, and several advances to the artificial neural network architecture were made. After that productive decade, progress slowed down; however, due to even further developments in both artificial neural networks and increased processing power in computers, neural networks are now extremely popular and making large advances (Hardesty). Relevant to our query about measuring lawns from satellite images, convolutional neural networks are a major advancement in recent years for a variety of reasons. Specifically, convolutional neural networks would work theoretically well because the lawn measurement problem is an image problem. The convolutional neural network pulls out the features such as grass texture, tree patterns, and other image features, the dense network layers analyze those features and determine the lawn area.

In this project, first we show the disability of image processing methods in calculating house and lawn area, then show that deep learning and convolutional neural network would work very well in this purpose. Data will be described in section 2. Section 3 discusses the theoretical background. Results and conclusions come after in section 4.

2. DATA DESCRIPTION

Creating the neural network for the deep learning problem is only half of the battle. The other half is collecting all the necessary data. The specific dataset of satellite images of houses with their measured lawn area is a big challenge. While there were datasets that involved satellite imagery, they are often involved much larger areas and usually implemented in classification problems. Despite this problem, there have been imperfect but still effective ways of collecting the data needed for this project. The most immediate solution is measuring the lawn area of houses on Google Maps and cropping these images out of the screenshots.

While image quality was somewhat lower, it was the best dataset that could be found. We used an online software Area Calculator (Area Calculator Using Maps) to measure each total lawn area in square meters. Then we used Krita, an image editing software (“Digital Painting. Creative Freedom”) to crop the larger picture and make smaller individual pictures. The resulted images contain one of the measured areas with its surrounding lawn. In total, the author collected 65 pictures of houses and their lawns. This dataset was too small to actually use as training data. In order to remedy this problem, we used the ImageDataGenerator class mentioned below in order to duplicate each picture. In many deep learning image processing problems, artificial data is a legitimate way to reduce potential overfitting and increase performance. One way is to duplicate a picture but then slightly change it in some way in order to produce an essentially different picture; this can be done by image’s rotating, inverting or flipping, distorting, changing the brightness of the image, and more. The ImageDataGenerator class has a method that can do all of those manipulations by just specifying in the constructor parameters of how you want to change each picture. The change is often random and within a given parameter range. This class was used to iterate through each picture in the dataset and create an augmented duplicate of each

image 50 times. Figure 1 is a sample of duplicating images. 3000 images created with this method. We took a portion of created dataset as a test dataset. Test data for validation determines the overfitting.

3. THEORETICAL BACKGROUNDS

3.1. Convolutional Neural Network

The convolutional neural network (CNN) helps to achieve a basis in deep learning projects. It is popular for classification rather than regression that is the main part of the current problems. Most importantly, CNN was included in the Keras library, which has been used to carry out a great deal of the process. Ultimately, the neural network did not reach the goal of 90% accuracy, but the accuracy was between 85% and 90%. Unfortunately, it did not necessarily explore overfitting in-depth and still did not apply to the current work [5, 6]. Hence, we used Tensorflow that was more complex but too low-level to easily implement.



Figure 1. Left-Image of before Data Augmentation; Right-Image of after Data Augmentation

There are many ways to implement convolutional neural networks. Sometimes, it works by defining and implementing complex functions with code from scratch [7] or by using Keras with Tensorflow as a backend [8]. A convolutional neural network is a 2D network meaning that it could take in 2-dimensional data. This fact is the kind of neural network used for image processing because pictures represented as 2-dimensional through their height and width.

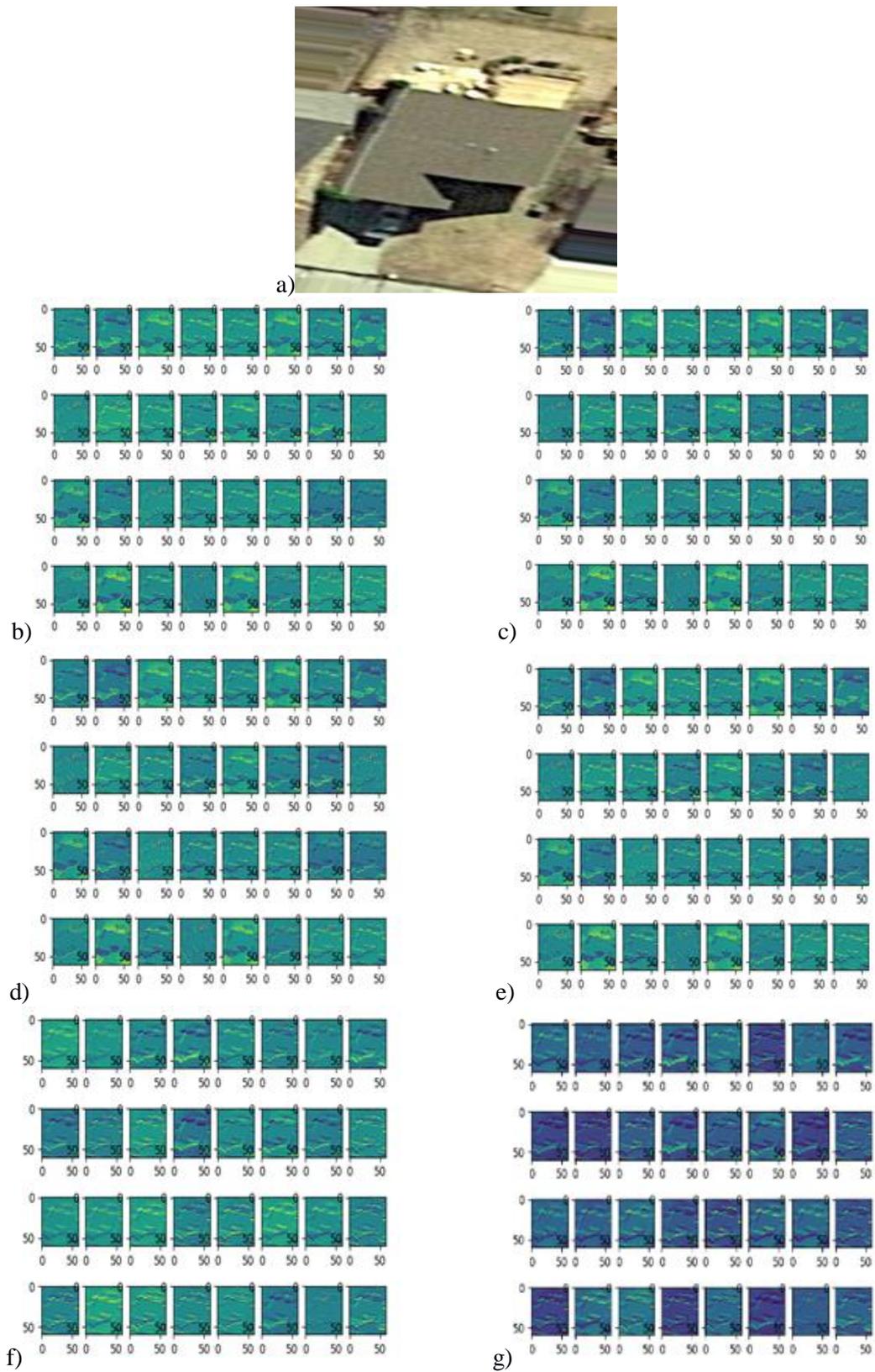


Figure 2. a) The original picture; b) Activation layer 1; c) Activation layer 2; d) Activation layer 3; e) Activation layer 4; f) Activation layer 5; g) Activation layer 6

However, the inputted arrays are 4 dimensions because it also includes the number of color channels and the number of pictures. It is 1-dimensional CNN for sentences and 3-dimensional CNN for voices. We started with three to four convolutional layers with two dense output layers at the bottom to process the information from the previous layers. After a few iterations, we used a different version of the convolutional layer in Keras that could be supposedly better performance called the separable convolutional layer. The separable convolutional layer or depth wise separable convolutional layer is an advanced version of the convolutional layer that can often increase the performance of the deep learning image processing models [3].

Besides, we implemented a new activation function in each layer called an ELU (instead of the previous RELU function) which again is often associated with producing better performance in neural networks [2]. Then, we implemented Batch Normalization to increase performance significantly, reduce the overfitting in the model, and deactivate a given layer in a certain percentage of the time [2]. We apply regularizations at each layer to prevent overfitting [9]. We also added the usual convolutional layer complements such as pooling layers to reduce the size of each picture at each layer [2]. The final model after intensive training on several models was a nine-layer convolutional neural network with 6 convolutional and 3 dense layers. This idea is probably the only neural network that specialized in regression analysis with the given satellite dataset. The basic procedure is as following: An array containing all the picture's arrays passed through the first convolutional layer, which pulls out certain low-level features from the picture. Then, the ELU activation function determines what values/features should move to the next convolutional layer. A pooling layer reduces the size of the picture and passes from every two convolutional layers. After passing through the six convolutional layers, the resulted features as a flattened list, processed by the three dense layers, which determine the area, correlated with the given feature [10]. By pulling out the features of the pictures through these convolutional layers and activations, the model simplified the data and emphasized certain features. Figure 2 shows an example for results of this process. Besides, the fine-tuning of the neural network, which involves trial and error attempts of a different number of neurons, layers, regularize' parameters, and optimizers, was also included in this example.

In fine-tuning of the neural network, many of parameters were tested with a class from the Scikit-Learn library in Python called GridSearchCV (`Sklearn.model_selection.GridSearchCV`) [11]. GridSearchCV is a way of automating the trial and error process as well as testing the true accuracy or performance of a neural network. One of the concepts in this library is cross-validation or CV. Essentially, CV takes the existing dataset and separates it into further sub-sets that can be used for validation. The GridSearch part of GridSearchCV tests data by iterating through hyper-parameters. Hence, the GridSearchCV finds the best parameter too. While this method served its purpose well, it limited the dependability; specifically, in reducing overfitting. Because the validation partly relied on duplicated data from the same dataset, it does not show the best accuracy for reducing overfitting. Still, it was useful to see which parameters would over-fit the most and then try to reduce it by adjusting the dropout functions which calculates manually. Overall, it led to the best performance as of now. Figure 3 shows the final cropped model.

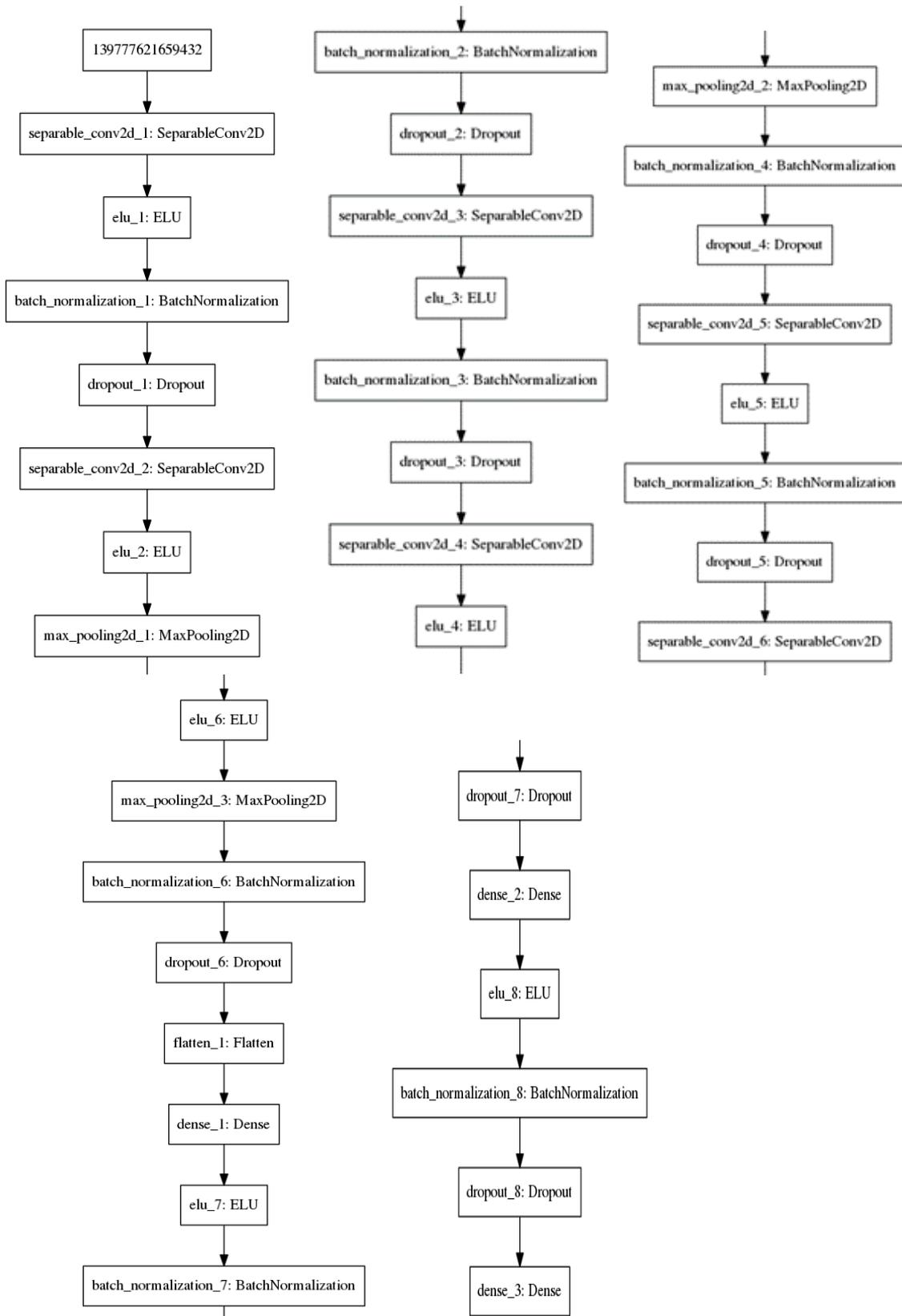


Figure 3. The final cropped images of the main model used, the model starts with the most left image and ends at the most right image

3.2. Thresholding and Augmentation

Though the model significantly fine-tuned, it was still below 90% accuracy or even got to 85% especially on the test set. Therefore, we worked on another solution like OpenCV, which is a non-deep learning solution to the problem [12]. OpenCV is an image-processing library in Python. It has several methods and classes used to manipulate images that can extract desired features. OpenCV emphasizes or removes unnecessary features from the pictures, which lets the neural net focus on the essential features for measuring the lawn area. The 3 different studied methods include thresholding, contouring finding, and edge detection [13]. Thresholding requires a range of pixel color densities and converts them to black and white images. Its purpose is to make the house in the middle of each picture completely change to one color (white) and the rest of the lawn area to another color (black). Threshold eliminates functions unrelated to lawn area, thus simplifying the measurement of lawn area. Figure 4 shows an image before, after Thresholding, and after both Thresholding and Augmentation.

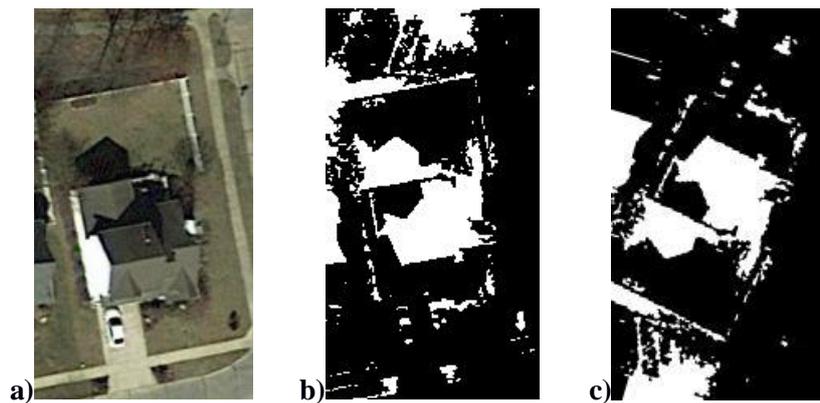


Figure 4. a) Image before Thresholding; b) Image after Thresholding; c) Image after Thresholding and Augmentation

3.3. Contouring and Augmentation

After several tests and some fine-tuning, Thresholding and Augmentation did not work very well because they could not even replicate the current model's accuracy. The next method was finding the contours of the houses and their lawn area. The contouring method finds the edges and curves of the houses (Figure 5). Despite adding extra information contouring also failed to exceed the performance of the original neural network.

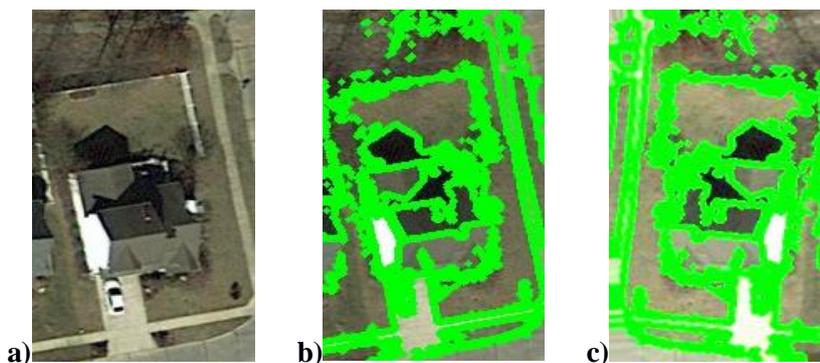


Figure 5. a) Image before Contouring; b) Image after Contouring and Gaussian Blurring; c) Image after Contouring and Augmentation

3.4. Canny Edge Algorithm

Finally, we used the Canny Edge algorithm to find the edges of the house traced by the program that was able to simulate the desired result of the thresholding (Figure 6). Unfortunately, this algorithm also failed to meet expectations. Despite OpenCV being a powerful tool in many image processing applications, it was insufficient for this project's purpose.

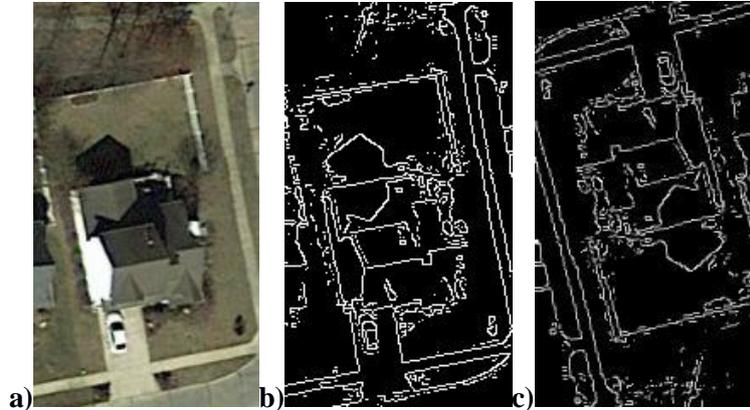


Figure 6. a) Image before Canny Edge Algorithm; b) Image after Canny Edge Algorithm; c) Image after Canny Edge Algorithm and Augmentation

4. RESULTS AND CONCLUSION

Table 1 shows the accuracy of different models. The final accuracy of convolutional neural network is not quite at 90%, but it definitely was able to measure lawn area within a given margin of error. Looking at the Mean Squared Error (MSE) of the prediction of the test data after the most recent training on 1,849 pictures, MSE of 1437 is the given result which is around 38 square meters margin of error for each predicted lawn area ($\sqrt{1437} = \sim 38$ sq. m.). To get something like an accuracy percentage, the ratio of the margin of error to the actual average value subtracted from 1 ($1 - (\text{error}/\text{average value})$). The average measurement of the data that was trained on this model (pictures 1-45) is roughly 294 sq. m. The accuracy is on average $\sim 87\%$ ($1 - (38/294) = \sim 87\%$). The median, which is approximately 276 sq. m. with the accuracy: $\sim 86\%$ ($1 - (38/276) = \sim 86\%$). The test result, unfortunately, is the result of some overfitting that has been occurring, but it is not that radically different from the training results. The mean squared error of the training vs the testing is only off by a less than 100 MSE even though there is a greater difference between training and validation MSE.

However, there is some evidence of volatility in this model; for example, in another test, the MSE was 2366 meaning an average error of 49 Squared meters, $\sim 83\%$ average accuracy, and $\sim 83\%$ median accuracy. While this would not be acceptable to any lawn care business yet, this is still a sign of hope that this neural network can at least predict the value within a reasonable margin of error.

As mentioned before, there are 65 pictures accumulated. So, there are approximately 3000 duplicated pictures in total. However, the final result used only the 45 original pictures which were separated into training, validation, and test data. In this case, there were really 1,849 training samples, 150 validation samples, and 250 test samples when this neural network achieved the above accuracy. The results obtained by image processing methods have almost half the accuracy of deep learning methods. Our results could be a resource to help people working on

properties lawn area and other business companies [14,15,16].

Table 1. Mean Squared Error of training, validation, and testing datasets.

Model Used	Highest Accuracy (1-Error/Average of Original Data)	Model Results (Average Predicted Lawn Area)	Average Lawn Area of Used Data
CNN Training	-	-*	~298.97 m ²
CNN Validation	~94%	~280.12 m ²	~298.14 m ²
CNN Testing	~97%	~262.65 m ²	~254.17 m ²
Threshold Model Training	-	-*	~283.55 m ²
Threshold Model Validation	~80%	~228.87 m ²	~287.21 m ²
Threshold Model Testing	~87%	~221.22 m ²	~254.17 m ²
Contour Model Training	-	-*	~297.66 m ²
Contour Model Validation	~26%	~634.49 m ²	~280.35 m ²
Contour Model-Testing	~31%	~587.50 m ²	~254.17 m ²
Edges Model Training	-	-*	~297.89 m ²
Edges Model Validation	~85%	~237.80 m ²	~280.53 m ²
Edges Model Testing	~57%	~145.16 m ²	~254.17 m ²

* The average predicted result was not shown for the training data since the MSE recorded during training does not correspond with the MSE and the predicted lawn area observed through Keras predict function for the training data

REFERENCES

- [1] W. Zhou, A. Troy, M. Grove (2008) Modeling Residential Lawn Fertilization Practices: Integrating High Resolution Remote sensing with Socioeconomic Data. Springer.
- [2] A. Schepelmann (2010) Identification and Segmentation of Lawn Grass Based on Color and Visual Texture Classifiers, Master Thesis, Case Western Reserve University.
- [3] W.S. McCulloch, W. Pitts (1990) A Logical Calculus of the Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biology Vol. 52, No. 1/2. pp. 99-115. 1990.
- [4] V. Reddy, M. K. Kumar, K. A. Naik, Widrow Hoff (2011) Learning Algorithm Based Minimization of BER Multi-User Detection in SDMA, International Journal of Technology And Engineering System (IJTES): Jan –March 2011- Vol. 2 No. 3.
- [5] Geron (2017) Hands-on Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly.
- [6] F. Chollet (2017) Deep Learning with Python, Manning.
- [7] Vedaldi, K. Lenc (2016) MatConvNet Convolutional Neural Networks for MATLAB, ArXiv: 1412.4564v3.
- [8] R. Airola, K. Hager (2017) Image Classification, Deep Learning and Convolutional Neural Networks, Karlstad University.

- [9] S. Demyanov (2015) Regularization Methods for Neural Networks and Related Models, The University of Melbourne.
- [10] M. Heusel, D.A. Clevert, G. Klambauer, A. Mayr, K. Schwarzbauer, T. Unterthiner, S. Hochreiter(2015) ELU-Networks: Fast and Accurate CNN Learning on ImageNet, ICCV 2015.
- [11] Scikit-learn developers (2018) Scikit-learn User Guide, Release 0.19.2 June 2018.
- [12] A. Mordvintsev, K. Abid (2017) OpenCV-Python Tutorials Documentation, Release 1, 2017.
- [13] R. C. Gonzalez, R. E. Woods (2017) Digital Image Processing, Pearson.
- [14] Project Administration Handbook for Civil Engineering Works, (2018).
- [15] E. Schappacher (2018) Embracing web-based tools to measure lawns, Landscape Management.
- [16] P. Robbins, T. Birkenholtz (2002) Turfgrass revolution: measuring the expansion of the American lawn, Land Use Policy 20 (2003) 181–194, Pergamon.