



## Image Processing Failure and Deep Learning Success in Lawn Measure

---

Jessie Wilkins, Vu Nguyen and Bahareh Rahmani

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

July 12, 2020

# Image Processing Failure and Deep Learning Success in Lawn Measurement

<sup>1</sup>J. Wilkins, <sup>1</sup>M. V. Nguyen, <sup>1,2</sup>B. Rahmani

<sup>1</sup>Math and Computer Science Department, Fontbonne University, St. Louis, MO, USA

<sup>2</sup>Computer Science and Data Science Department, Maryville University, St. Louis, MO, USA  
b.rahmani@wuslt.edu

**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 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 but 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 and other deep learning techniques.

**Keywords.** Lawn Measurement, Convolutional Neural Network, Thresholding, Edge Detection, Contouring, Tensorflow, Keras, Regression

## 1. Introduction

The lawn measurement project has to start nearly from scratch, and creative ways should find to deal with the limited quality and quantity of the samples.

We used the lawn area of houses from satellite images. The process is simple: find the address of the house, use online software to measure the area of the lawn, record the measures, and repeat the steps. When the company sends out people to measure the lawn area manually, they want some rough estimate beforehand in areas. It is a very repetitive job, but a human could only do it. Hence, the lawn care company had to pay the person at least the minimum wage, and the person had to spend many hours trying to measure hundreds of houses. Therefore, the company spends thousands of dollars and inordinate amount of time to measure. Eyestrain and fatigue are other issues that cause 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 method used by the lawn care company is artificial neural networks. Artificial neural networks have a long winding history in the computer science field. The idea of neural network in general started in biology and influenced artificial neural networks. Their mathematical conception dates back to 1943 in a paper of Warren S. McCullough and Walter Pitt [3].

In the 1950s, several attempts ended 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 corruption from phone lines by predicting the next bit. In the 1980s, neural networks revived as a topic of interest, and several advances to the artificial neural network architecture 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 worked 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 weakness of image processing methods in calculating lawn area. Then, we show that deep learning and convolutional neural network work very well. Datum description comes in section 2. Section 3 discusses about theoretic background. Results and conclusions come after in section 4.

## **2. Data Description**

Creating the neural network for the deep learning problem is only half the battle. The other half is collecting all the necessary data. Collecting the specific dataset of satellite images of houses with their measured lawn area is a big challenge. While there were datasets that involved satellite images, they often involved much larger areas and usually implemented in classification problems. Despite this problem, there were imperfect, but still effective ways of collecting data. The best solution is measuring the lawn area of houses on Google Maps and cropping these images. While image quality was somewhat lower, it was the best dataset found. We used an online software Area Calculator (Area Calculator Using Maps) to measure each area in square meters. Then we used Krita, an image editing software (Digital Painting. Creative Freedom) to crop the larger picture and make smaller ones. In total, we collected 65 pictures of houses and their lawns. This dataset was too small to use as training data. In order to remedy this problem, we used the ImageDataGenerator class 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 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 everything you need to change each picture. The change is often random and within a given parameter range. This class used to iterate through each picture in the dataset to create an augmented duplicate of each image 50 times. Fig1 is a sample of duplicating images. 3000 images created with this method.

## **3. Theoretical backgrounds**

### **3.1. Convolutional Neural Network**

The convolutional neural network (CNN) is for classification rather than regression that is the main part of the current problems. Most importantly, CNN introduces the Keras library, which used to carry out a great deal of the process. The accuracy of the neural network 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.



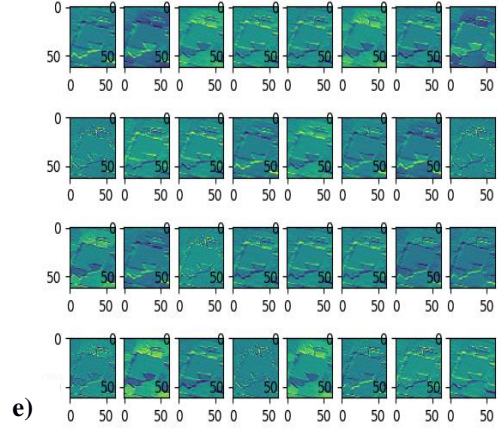
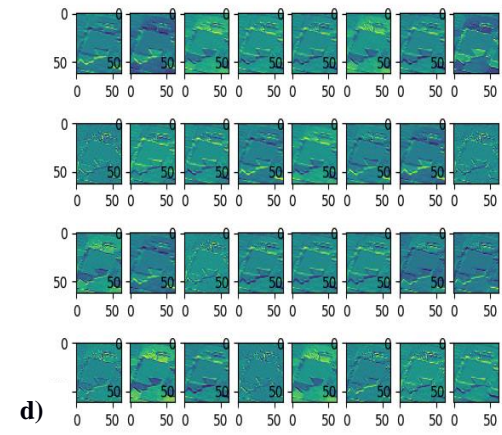
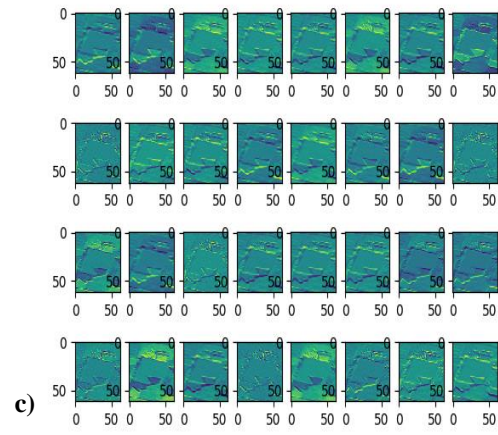
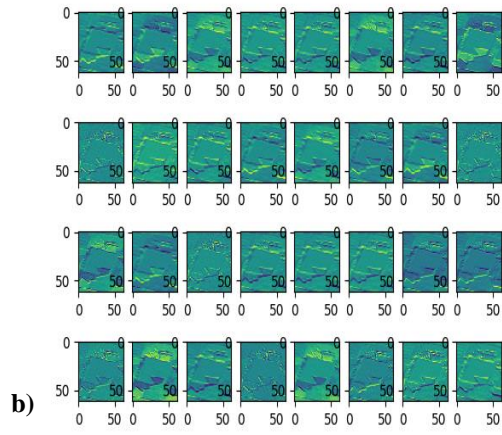
**Fig. 1.** Left-Image before Data Augmentation and Right-Image after Data Augmentation

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. 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 start 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 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]. 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 procedure is as follows. 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. Fig 2 shows an example.

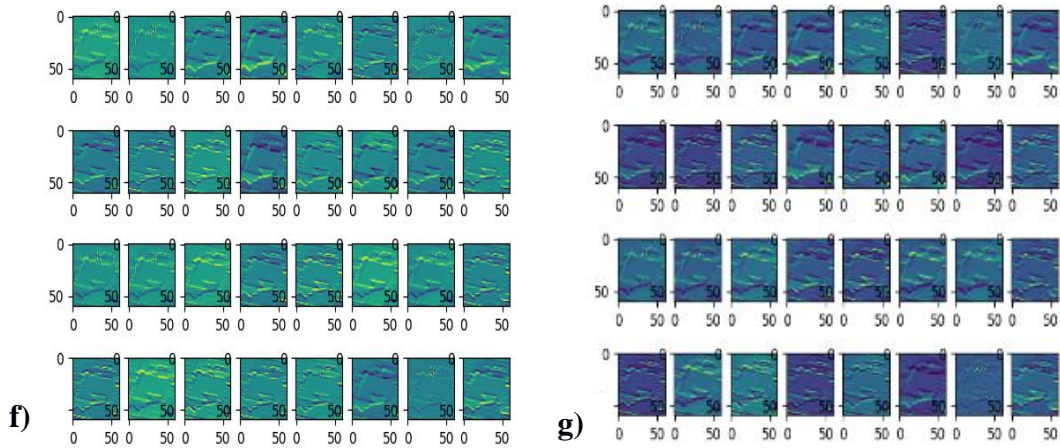
The fine-tuning of the neural network in Figure 2 involves a different number of neurons, layers, regularize' parameters, and optimizers.



**a)**







**Fig. 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

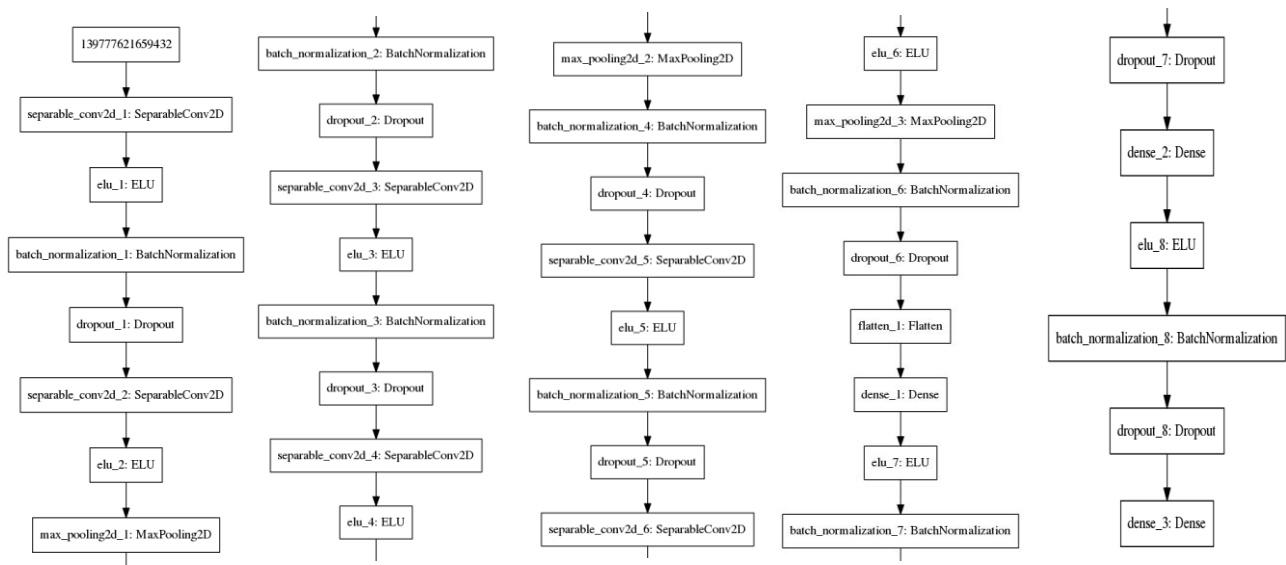
Many of these parameters tested with a class from the Scikit-Teach library called GridSearchCV (Sklearn.model\_selection.GridSearchCV) [11]. GridSearchCV is a way of automating the trial and error process and testing the true accuracy or performance of a neural network. One of the concepts in this library is cross-validation or CV. Essentially, CV takes the existing dataset and separates it into further sub-sets 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 serves its purpose well, it limits the dependability; specifically, in reducing overfitting. Because the validation partly relies on duplicated data from the same dataset, it does not show the best accuracy for reducing overfitting. Still, it is useful to see which parameters would over-fit the most and try to reduce it by adjusting the dropout functions, which calculate manually. Figure 3 shows the final cropped model.

### 3.2. Thresholding and Augmentation

Though the model significantly fine-tuned, it is 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. Hence, the neural network focuses on the essential features needed to measure the lawn area. Three different methods studied including thresholding, counter finding, and edge detection [13]. Thresholding requires a range of pixel color densities and converts them to black and white images. The purpose of thresholding 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 irrelevant functions that are not related to lawn area, thus simplifying the measurement of lawn area. Figure 4 shows an image before, after Thresholding, and after both Thresholding and Augmentation.

### 3.3. Contouring and Augmentation

The next method is to find the outline of the house and its lawn area. The contour method can find the edges and curves of the house (Figure 4). Despite the addition of additional information, the contour drawing failed to exceed the performance of the original neural network.

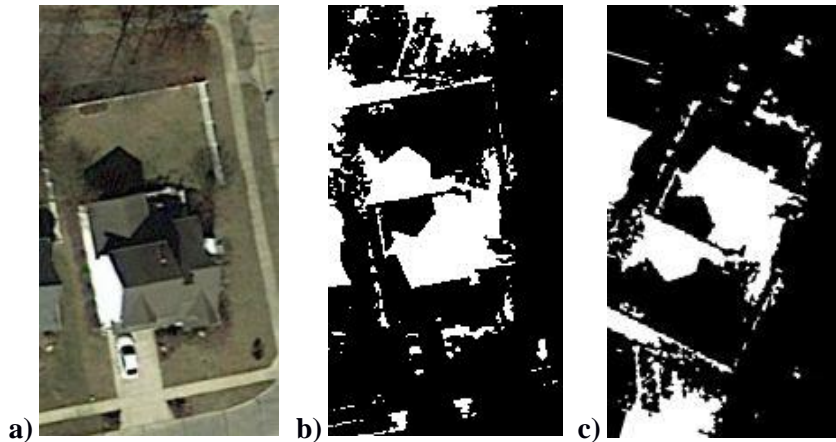


**Fig. 3.** The final cropped images of the main model used, the model starts with the most left image and ends at the most right image.

### 3.4. Canny Edge Algorithm

Finally, we used the Canny Edge algorithm to find the edge of the house tracked by the program, which can simulate the ideal result of the threshold (Figure 5).

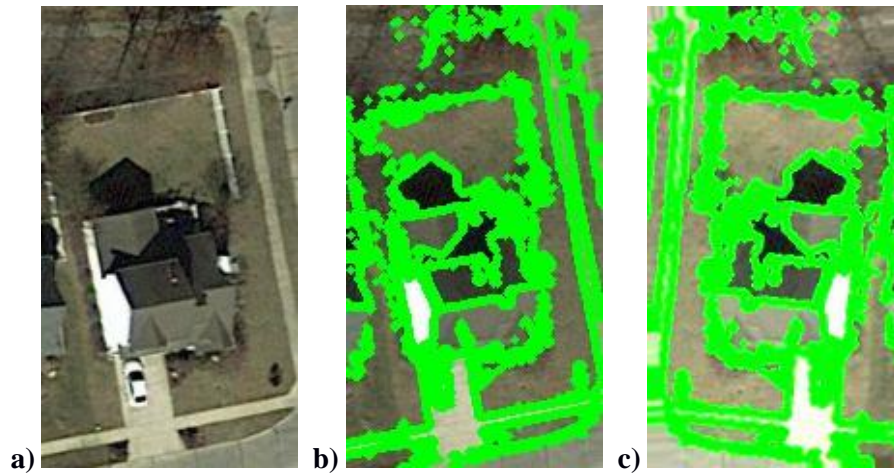
Unfortunately, the algorithm also failed to meet expectations. Despite OpenCV being a powerful tool in many image-processing applications, it was insufficient for this project's purpose.



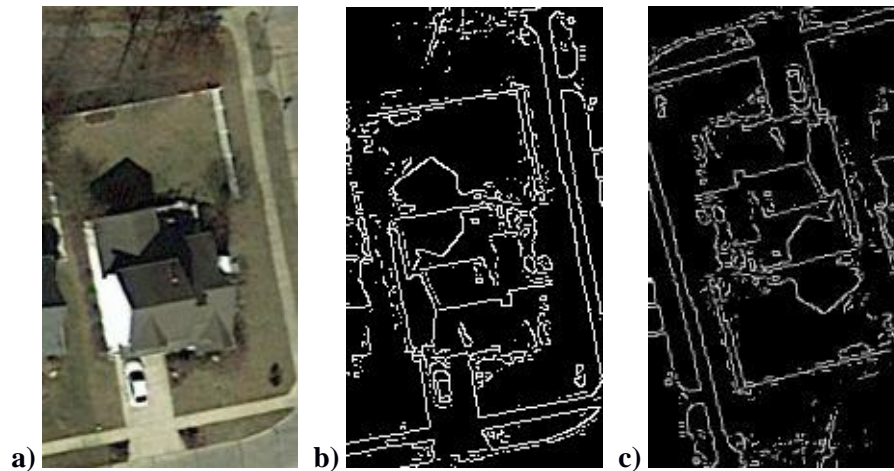
**Fig. 4.** a) Image before Thresholding b) Image after Thresholding  
c) Image after Thresholding and Augmentation

## 4. Results

The final accuracy of the convolutional neural network is not quite at 90%, but it was able to measure the 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, 1437 MSE is the given result which is around 38 square meters ( $\sqrt{1437} = \sim 38$  sq. m.) margin of error for each predicated lawn area. To get something like an accuracy percentage, the ratio of the margin of error to the average value subtracted from 1 ( $1 - (\text{error}/\text{average\_val})$ ).



**Fig. 5.** a) Image before Contouring b) Image after Contouring and Gaussian Blurring, c) Image after Contouring and Augmentation



**Fig. 6.** a) Image before Canny Edge Algorithm b) Image after Canny Edge Algorithm c) Image after Canny Edge Algorithm and Augmentation

The average measurement of the trained data on this model (pictures 1-45) is roughly 294 sq. m. The accuracy is on average ~87% ( $1 - (38/294) = \sim 87\%$ ). median is about 276 sq. m. with the accuracy: ~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 less than 100 *MSE* even though there is a greater difference between training and validation's *MSE*.

However, there is some evidence that the model is volatile. For example, in another test, the *MSE* is 2366, which means that the average error is about 49 square meters, the average accuracy is about 83%, and the median accuracy is about 83%. While this idea would not be acceptable to any lawn care business yet, this fact is still a sign of hope that this neural network can at least predict the value within a reasonable margin of errors.

As mentioned before, there are 65 pictures collected. Hence, there are close to 3000 duplicated pictures in total. However, the result used only the 45 original pictures separated into training, validation, and test data. In this case, when the neural network reaches the above accuracy, there will be 1,849 training samples, 150 verification samples and 250 test samples. The results obtained by image processing methods have almost half 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 (error/Average of Original Data)	Model Results (Average Predicted Lawn Area)	Average Lawn Area of Used Data
CNN Training	-	-*	~298.97
CNN Validation	~94%	~280.12 m <sup>2</sup>	~298.14 m <sup>2</sup>
CNN Testing	~97%	~262.65 m <sup>2</sup>	~254.17 m <sup>2</sup>
Threshold Model Training	-	-*	~283.55
Threshold Model Validation	~80%	~228.87	~287.21
Threshold Model Testing	~87%	~221.22	~254.17
Contour Model Training	-	-*	~297.66
Contour Model Validation	~26%	~634.49	~280.35
Contour Model-Testing	~31%	~587.50	~254.17
Edges Model Training	-	-*	~297.89
Edges Model Validation	~85%	~237.80	~280.53
Edges Model Testing	~57%	~145.16	~254.17

\* The average predicted result is not available for the training data since the MSE recorded during training does not correspond to 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, Deep Learning with Python. Manning (2017).
7. Vedaldi, K. Lenc, MatConvNet Convolutional Neural Networks for MATLAB. ArXiv: 1412.4564v3, (2016).
8. R. Airola, K. Hager, Image Classification, Deep Learning and Convolutional Neural Networks. Karlstad University (2017).
9. S. Demyanov Regularization Methods for Neural Networks and Related Models. The University of Melbourne (2015).
10. M. Heusel, D.A. Clevert, G. Klambauer, A. Mayr, K. Schwarzbauer, T. Unterthiner, S. Hochreiter ELU-Networks: Fast and Accurate CNN Learning on ImageNet. ICCV (2015).
11. Scikit-learn developers, Scikit-learn user guide, Release 0.19.2,( June 2018).
12. A. Mordvintsev, K. Abid, OpenCV-Python Tutorials Documentation, Release 1, (2017).
13. R. C. Gonzalez, R. E. Woods, Digital Image Processing. Pearson. (2017).
14. Project Administration Handbook for Civil Engineering Works, (2018).
15. E. Schappacher, Embracing web-based tools to measure lawns, Landscape management, (2018)

16. P. Robbins, T. Birkenholtz, Turfgrass revolution: measuring the expansion of the American lawn, *Land Use Policy* 20 (2003) 181–194, Pergamon, (2002).