2018

Development of Vegetation Mapping with Deep Convolutional Neural Network

Sae-han SUH
South Dakota State University

DEVELOPMENT OF VEGETATION MAPPING WITH DEEP CONVOLUTIONAL NEURAL NETWORK

BY

SAE-HAN SUH

A thesis submitted in partial fulfillment of the requirements for the

Master of Science

Major in Computer Science

South Dakota State University

2018

DEVELOPMENT OF VEGETATION MAPPING WITH DEEP CONVOLUTIONAL NEURAL NETWORK

SAE-HAN SUH

This thesis is approved as a creditable and independent investigation by a candidate for the Master of Science in Computer Science degree and is acceptable for meeting the thesis requirements for this degree. Acceptance of this does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Sung Shin, Ph.D.

Thesis Advisor Date

George Hamer, Ph.D.

Acting Head, Department of Electrical Engineering and Computer Science Date

Dean, Graduate School Date

ACKNOWLEDGMENTS

I sincerely thank Professor Sung, Shin for giving me an opportunity to work in the field of computer science. I would also like to thank Professor Alireza Salehnia and Professor Kwanghee Won for their time and valuable intellectual endeavors.

I also take this opportunity to express a deep sense of gratitude to thank my wife, who has read the drafts and commented on topics for the improvement of this paper. It was all thanks to her, that kept me going on, and this paper would have never been possible without her help.

TABLE OF CONTENTS

ABBREVIATIONS

API	Application Programming Interface
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
DA	Data Augmentation
GPS	Global Positioning System
JIT	Just-In-Time
NIN	Network-In-Network
OCR	Optical Character Recognition
PA	Precision Agriculture
ResNet	Residual Network
SSCM	Site-Specific Crop Knowledge
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle
USGS	United States Geological Survey
VGG	Visual Geometry Group

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

DEVELOPMENT OF VEGETATION MAPPING WITH DEEP CONVOLUTIONAL NEURAL NETWORK

SAE-HAN SUH

2018

The Precision Agriculture plays a crucial part in the agricultural industry about improving the decision-making process. It aims to optimally allocate the resources to maintain the sustainable productivity of farmland and reduce the use of chemical compounds. [17] However, the on-site inspection of vegetations often falls to researchers' trained eye and experience, when it deals with the identification of the non-crop vegetations. Deep Convolution Neural Network (CNN) can be deployed to mitigate the cost of manual classification. Although CNN outperforms the other traditional classifiers, such as Support Vector Machine, it is still in question whether CNN can be deployable in an industrial environment. In this paper, I conducted a study on the feasibility of CNN for Vegetation Mapping on lawn inspection for weeds. I want to study the possibility of expanding the concept to the on-site, near real-time, crop site inspections, by evaluating the generated results.

1. INTRODUCTION

Precision Agriculture is expected to provide farmers with a decision support system to improve productivity at a reduced manual effort. With the increased occurrence of the food crisis, the combination of deep learning with this domain has gained much attention. Identification or classification of plants is still a challenging task because of the lack of appropriate datasets and the identification difficulty from early stage figure of plants. Therefore, the current trend is favorable to CNN which does not need manually-crafted features. It is likely to apply the findings from a specific CNN model to the other datasets, rendering the former more robust. [7]

The recent successes of the Convolutional Neural Networks in object detection have been revolutionizing the Computer Vision; the success of the AlexNet at the ImageNet Large Scale Visual Recognition Competition 2012 [19] proved that the neural network could outperform any classifiers in this field. [7] Though the recent advances in the Computer Vision are promising, we cannot deny the fact that the current, state-of-the-art models become increasingly complex and time-consuming. In various industrial & commercial scenarios, engineers and developers often face a demand for a system suited for a tighter time/spatial budget than the research environment. [26]

Current CNN-based model is generally trained with the vast dataset called ImageNet. At least, each image is annotated with one of its 1000+ classes. [10] In the Precision Agriculture, however, input images are often unannotated. Data source also varies significantly from the satellite imagery to the one from an onboard camera fitted in an Unmanned Aerial Vehicle (UAV) or any human-crewed vehicle. The overall resolution,

number of imagery and the features per pixel are dependent upon the capability of the onboard camera. One must consider these variables if he/she wishes to apply a CNN-based classification method to one of the Precision Agriculture fields. [11]

The current state-of-the-art CNN models are usually trained with computing devices fit for intensive computations. The first CNN that won the ImageNet Large Scale Visual Recognition Challenge 2012, AlexNet, is written with Nvidia® CUDA to run with Graphical Processing Unit; without such a device, the training of a CNN is practically infeasible. Since the embedded computing environment places tight restrictions on its system resource and power management, I assume that many of these state-of-the-art networks or models could fail to be trained or fitted in such an environment. Given a set of different resolution factors and an embedded computing environment, I study the feasibility of applying the CNN to the vegetation classification based on these observations.

I first study the effect of the layers on the performance of a classifier. The deep learning shows that it is useful in detecting and classifying the objects in a given dataset, but such a model would require considerable computational power. Hence, I would like to find empirically a number of layers in a network which would not impair the classifier with a reasonable classification result as the current state-of-the-art model. Second, I study the feasibility of classification on the images with the reduced resolution and the upsampled images. I will try to find a minimum image resolution that could guarantee the right classification with an acceptable score.

2. MATERIAL AND METHODS

2.1 Convolutional Neural Network

Convolutional Neural Network, CNN, is the current state-of-the-art object detection and the imagery classification system. The traditional approach to the imagery classification is divided into two steps. First, it extracts a set of features, carefully crafted by human experts. Second, by using the extracted features, the experts choose to use one of the classification systems. The first step is difficult because the accuracy of the classifier depends on the design of the feature extractor. Thus, a large amount of pattern recognition in image classification is only used to describe and compare different sets of features for a task. [21] However, the need for an appropriate feature extractor is that the learning techniques used by the classifier have been limited to low dimensional space with easily separable classes. [21] [22] With the availability of a large volume datasets and its differing characteristics, the classifier and learning techniques cannot rely on the 'learned' feature vectors, but on the dataset itself.

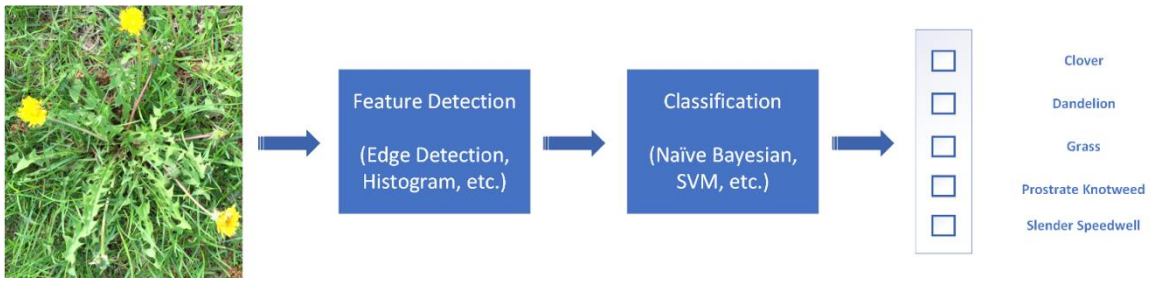

Figure 1. Illustration of a traditional system design for pattern

The biological researches in the 1950s initially inspire CNN. The CNN is modeled after the organization of the visual cortex of animal and tries to 'learn' features by adjusting the hyperparameters in the network during its training. Its independence from

the manually crafted features gives a major advantage to the CNN over the other classification systems, such as Support Vector Machine (SVM), Bayesian Classifier, and so forth. The training and use of CNN have been primarily enabled by the availability of modern hardware with relatively low-cost, the corresponding hardware application programming interface (API) (e.g., Nvidia® CUDA) and libraries. (e.g., Berkeley Artificial Intelligence Research Lab's Caffe, Google® TensorFlow) Before the AlexNet's superior performance (15.3 % error rate vs. (second place) 26.2 % error rate) [20] shown at ImageNet Large Scale Visual Recognition Competition 2012, CNN has had many applications in the image and video recognition, natural language processing, and so on. In the 1990s, AT&T's neural research group developed a weaving neural network for check reading, and several Optical Character Recognition (OCR) and handwriting recognition systems designed by Microsoft were based on the CNN itself. [23]

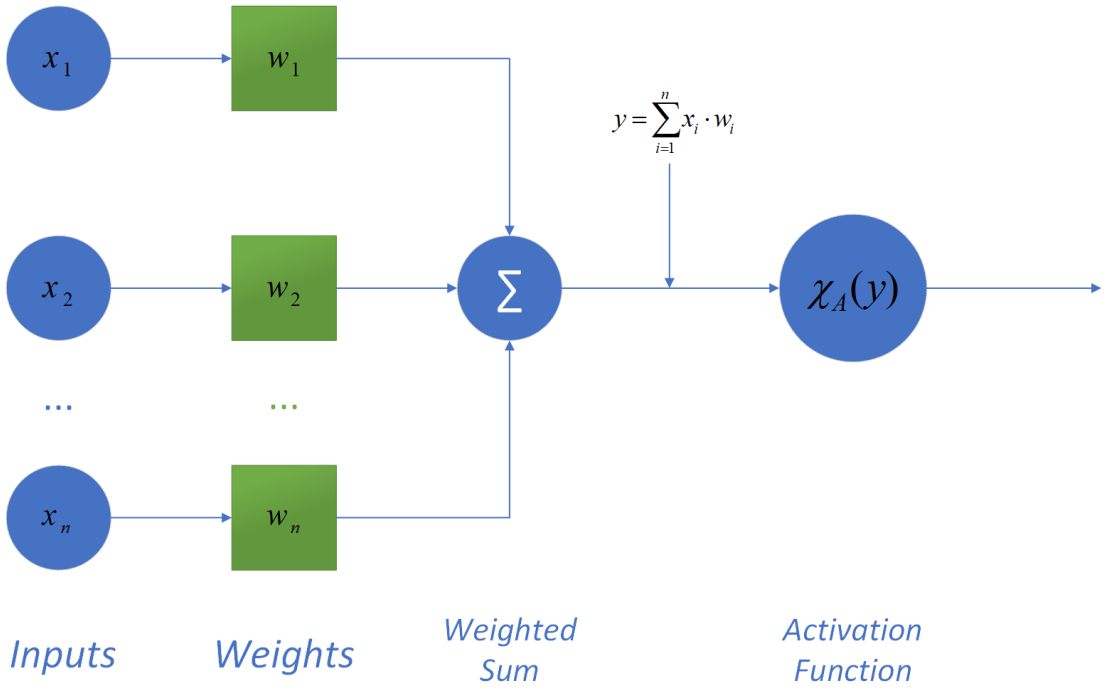

Figure 2. Diagram of a single layer, Perceptron

Note that the characteristic function can be any activation function, such as Sigmoid or ReLU (Rectified Linear Unit), and A can be a set of elements satisfying a specific condition.

Three architectural components are integrated into the CNN: local receptive fields, shared weights and spatial/temporal sub-sampling. [21] These are the key components to ensure the learning of a neural network as well as some degree of shift, scale, and distortion invariance of the network. The authors of [21] assert that the sequential use of local receptive field and subsampling is inspired by the Perceptron. In 1957, Frank Rosenblatt constructed Perceptron to simulate the neuronal response to a random input. The conception was nearly simultaneous to work by Hubel and Wiesel in the 1960s to determine how the mammalian visual cortex works. The authors of [23] assert that CNN is specifically designed to capture three properties of the visual cortex:

1. The visual cortex is arranged in a spatial map; it possesses a two-dimensional structure, mirroring the structure of the image in the retina.
2. Visual cortex incorporates many ‘simple’ cells. A single cell’s main activity can be characterized by a linear function of the image in a small, spatially localized receptive field.
3. Visual cortex also incorporates many ‘complex’ cells. A complex cell’s activity is nearly identical to the simple one, but there’s one significant difference; their activity is invariant to the position of the feature. [23]

These particularities are the key features the CNN aims to capture for emulating the visual cortex. Although the CNN differs from the biological neural network, one cannot deny that it has played a crucial role in the history of deep learning. The CNN is a successful application of insights gained through brain research on machine-learning applications. [23] Here’s the simplified illustration of the CNN:

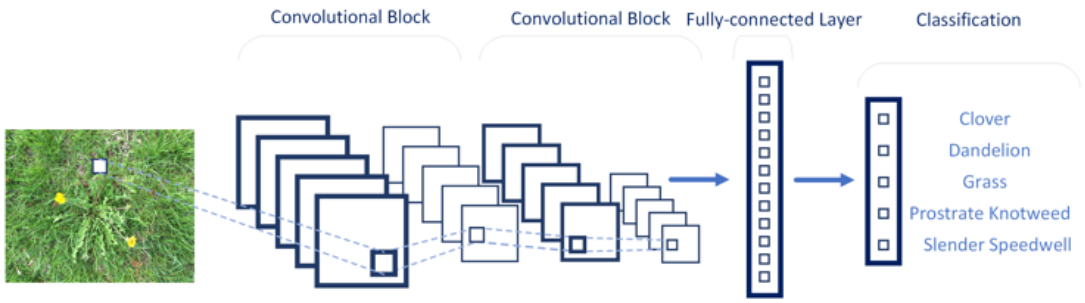

Figure 3. Illustration of a Convolutional Neural Network Design

VGG-16 (Visual Geometry Group) is one of the well-known CNN architectures, due to its simplicity in its construction. The network has 16 convolutional layers, in total,

and is divided into five blocks. Unlike the AlexNet (the winner of the ILSVRC 2012), it uses a series of convolutional layers with the small receptive field (3×3), instead of a single convolutional layer with the bigger field (7×7). The use of multiple non-linear convolutional layers in each block enables the network to learn discriminative features more easily. [19] As a result, the resulting architecture attains less number of parameters, which renders it easy to optimize than its predecessors (e.g., AlexNet)

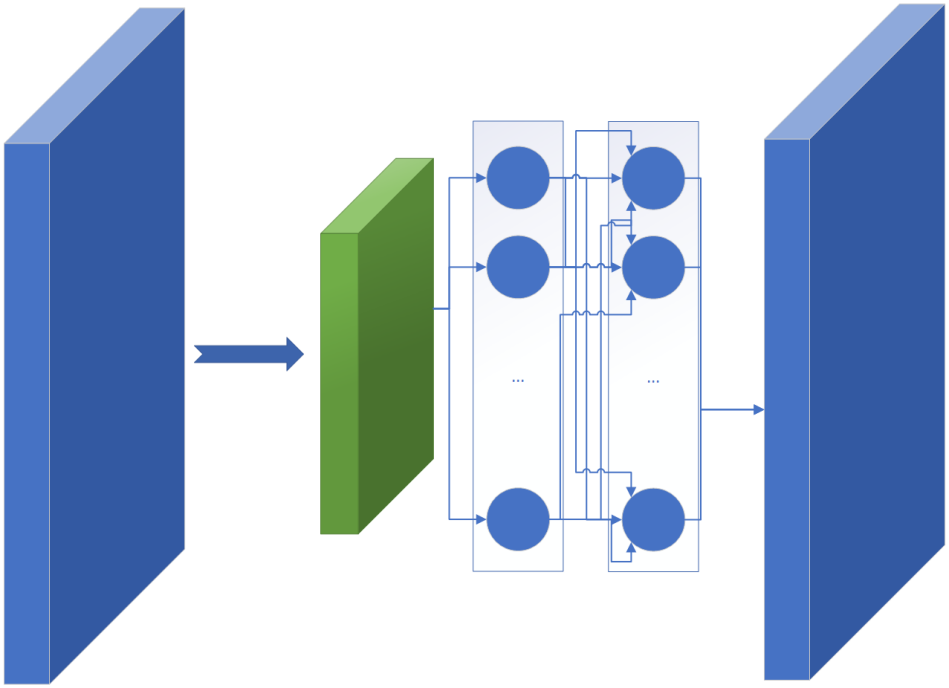

Figure 4. Illustration of a Network-In-Network (NIN) between two convolutional layers (blue cubes)

Note that the cube-like structure (green) means the receptive field. Between the receptive field and the next convolutional layer, there exists a 'micro' network (multi-layer perceptron) instead of a linear layer

In the history of CNN, VGG network architecture (especially VGG-16 and VGG-19) is also one of the first networks which incorporated some ideas of the Network-In-Network (NIN) structure into the design of the convolutional neural network. [19] [25] The authors of [25] argued that the neural network suffers from its low generalization capability since the shallow 'softmax' has the poor level for abstraction and the fully-connected layer on top of the network architecture tends to show overfitting. Moreover, the fully-connected layers are heavily dependent upon the regularization. Taken this observation into consideration, the authors of the VGG network architecture incorporated additional convolutional layers. This increases the non-linearity of the resulting network's decision capability without jeopardizing the receptive fields of convolutional layers.

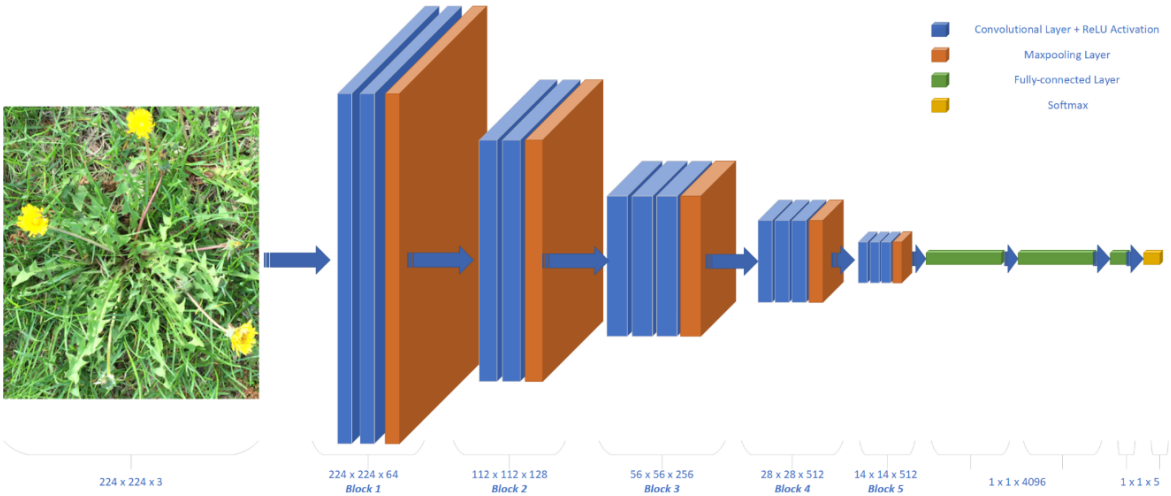

Figure 5. VGG-16 Architecture

VGG network architecture (especially VGG-16) shows that the depth of a neural network plays a critical role in discriminating the features and classifying the images.

Beginning with the VGG network architecture, the researchers' main focus is shifting towards building deeper neural networks.

In 2015, Kaiming He et al. [2] [26] discovered that more the layers are stacked, the more the network suffers from the degradation issue. The degradation is a counter-intuitive phenomenon where both the training and test error rates increase when the number of layers in the CNN increases. This holds true even if there is a change of dataset, according to the authors of [2]. As the depth of a neural network is increased, so is the computational burden. Among numerous architectures of the CNN, ResNet (Residual Network) framework is an effort to overcome the training issues. Unlike other architectures, ResNet framework contains the 'skip connection,' which will be explained later in this paper. This feature renders the network easier to optimize than the traditional CNN framework.

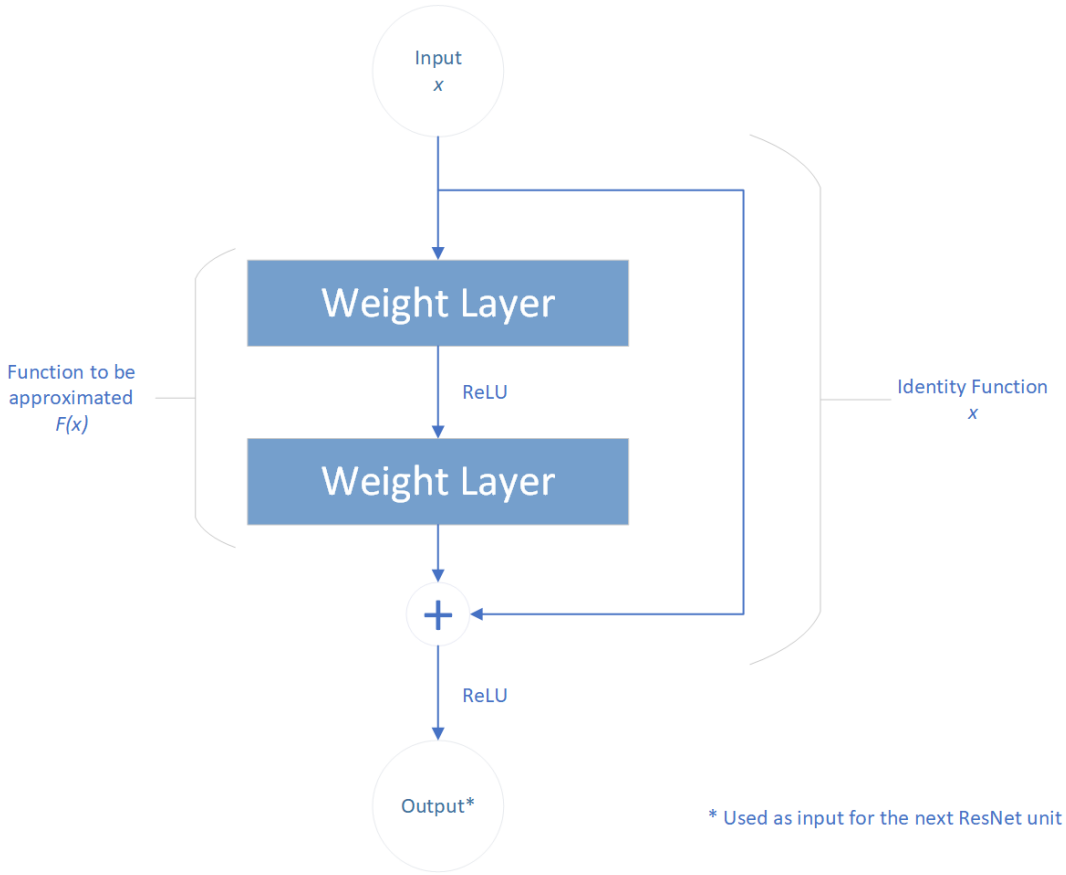

Figure 6. Basic Residual unit for ResNet Framework [2]

Kaiming He et al. described the concept of the ResNet framework as follows: suppose that the input and output of a neural network are of the same dimensions. Assume that there exists a certain function $\mathcal{H}(x)$, mapping to be approximated by a certain subset[1] of a neural network. (This is also the main hypothesis of a deep learning) Denote the set of inputs to the first layers by x. If one hypothesizes[2] that a set of nonlinear layers in a network is able to approximate functions in an asymptotical manner, then it implies that to approximate their residual functions in the similar manner is also possible:

$$\mathcal{H}(x) - x$$

[1] The authors of [2] assume that this subset of a neural network needs not to be a proper subset of the neural network.
[2] To the best of our knowledge, it is still an open problem.

Kaiming He et al. argues that rather than expecting a set of stacked layers to approximate $\mathcal{H}(x)$, it is recommended to design a neural network to approximate a residual function

$$\mathcal{F}(x) := \mathcal{H}(x) - x$$

Then, the function to be approximated becomes $\mathcal{F}(x) + x$. The authors of [2] concluded that "although both forms should be able to asymptotically approximate the desired functions (as hypothesized), the ease of learning might be different.".

In 2016, the developers of the original ResNet [6] improved their 'skip connection' design. In short, Kaiming He et al. pointed out that it is best to avoid the information of the shortcut; otherwise (e.g., performing a multiplication on the information from the skip connection), it would render the optimization of the network and even the backpropagation extremely difficult. So, the authors of the ResNet framework modified their ResNet building block to include the batch normalization and activation layers. This is called 'pre-activation'; with this modification, the original authors of the ResNet framework surpassed the upper bound of the number of convolutional layers to 1000 layers.

The ResNet framework has another strength regarding the computation and related complexity. Kaiming He et al. also pointed out that, whereas the original ResNet-34 model's depth is more profound than the one for the VGG-16 model, the former has one-

sixth of the computational complexity of the latter. [2] This renders both the VGG and the ResNet framework suitable for our study. I chose the ResNet framework for another basis of the study since several authors report that ResNet converges faster than the other network frameworks and is less likely to suffer from overfitting.

2.2 Precision Agriculture

Precision Agriculture (PA) is a relatively new concept of farming management. Its research aims to develop a support system in the decision-making process, by using the site-specific crop knowledge (SSCM), so that it can enable the farmers to optimize outputs on given inputs. At the same time, it can also preserve the resources of a farm. [18]

The inconsistent, excessive application of chemical substances has amounted to a series of undesirable consequences. They vary from nutrient imbalance, unforeseen damage (e.g., pesticide resistance) to reduced productivity. [17] A few kinds of literature indicate that PA can contribute to various objectives such as the longevity of certain farmland to the long-term sustainability of agricultural production. These studies confirm that PA could reduce the environmental influence of chemical substances, such as pesticides and fertilizer. PA aims to apply these substances to the area which needs the most attention. This targeted, localized, Just-In-Time (JIT) approach of the PA can be beneficial for the environment.

PA requires a mean to gather the necessary information on a specific farm. Traditionally, this could be done by Global Positioning System (GPS) and satellite

imageries, operated by public or private entities such as the United States Geological Survey (USGS). As of 2018, farmers can operate the Unmanned Aerial Vehicle (UAV), or drone, in a relatively inexpensive manner to gather the spatial variability of a farm. Then, they can analyze the fertility of their farm based on gathered intelligence.

2.3 Data Augmentation

The increase in computation power and steady flux of data collected from various sources have enabled the Convolution Neural Network to bear the state-of-the-art classification results. Although this trend seems to continue in natural language processing, and image and video classification, an important issue arises overfitting. It is mainly due to the relative scarcity of data corresponding to the label or the relatively small size of a dataset. Overfitting usually occurs during the training of a CNN with a small dataset. It prevents the CNN's capability to generalize on previously unlabeled data. Data Augmentation (DA) is a potential solution to most of the situations similar to our description.

DA is aimed to ‘inflate’ the volume of the training set, but keep the volume of the test set unchanged. A CNN trained with such an augmented dataset would be less likely to ‘change its opinion’ in case of change of the two variables above. [9] There exist principally two types of transformation: geometric transformations and photometric ones. The formers aim to increase the training set, by performing the geometry altering an image. This renders the CNN invariant to certain affine transformations. Examples are horizontal/vertical

flipping, cropping, and rotation. The photometric ones aim to achieve the same goal, by transforming the colors and the brightness of an image.

In principle, the DA is a rather inexpensive scheme to prevent overfitting and enhance the performance of a classifier, regarding its generalization capability. This class does not change for quite a few variants, and input can easily be transformed thanks to many geometric operations.

3. RESULTS AND DISCUSSION

3.1 Dataset Preparation

In this study, the input dataset is a set of photos taken personally around Brookings, SD, the USA during the April - May 2018 with one single iPhone 6. Each taken photo is cropped for various classes, a subset of plants typically on the lawns. I resized every image whose resolution is higher than 224 x 224-pixel size, a typical input dimension of our neural networks.

The dataset consists of 5,326 images of five classes specified below. For this paper, the dataset was randomly split into the training dataset and the test dataset by a 7:3 ratio. The class weight is set to mitigate the imbalanced number of samples in each class. I did take precaution in organizing the dataset since the classifier's performance is mainly dependent upon the quality of the dataset [4][8]. Each photo was taken at a 1m from the ground, to simulate the pseudo-horizontal point of shooting from the UAV or the camera attached to the tractor.

(a)  (b)  (c) 

(d)  (e) 

Figure 7. Sample images of the dataset:
a) Clover b) Dandelion c) Grass d) Prostrate Knotweed e) Slender Speedwell

Various data augmentation techniques were used: horizontal/vertical flipping, zooming, rescaling, and rotations $\{0°, 90°, 180°, 270°\}$. This series of techniques was reported to render the CNN less variant to various geometric transformations.

3.2 Training of a CNN

As discussed above, I used the ResNet framework to train a network. The following structures are used to evaluate the effectiveness of a classifier: ResNet-18, ResNet-34, and ResNet-50 as well as our ResNet-derived architectures. VGG-16 is selected for comparison with the ResNet framework. Our objective is to determine an optimal number of layers, given an individual neural network and the characteristics of the dataset.

There is no need to train a complex neural network if the one with lesser complexity performs as good as the former within the acceptable margin of error. Moreover, the number of hyperparameters increase proportionally to the depth of a network. Thus a danger of overfitting arises.

I conducted the training on the following setting: Intel® i7-7700k @ 4.20 GHz, 32 Gb Ram, a single Nvidia® GTX 1080, Ubuntu 16.04 LTS, Keras 2.0 API with TensorFlow 3.5 as the backend. Each training took about 35-40 minutes to be completed. For the training of each CNN model, the learning rate was set to 0.0001, decay rate to 0.0. The epoch was set to 50. The mini-batch size was set to 32 because the available GPU was limited to one single card. The Adam optimizer was used to minimize the loss values; it keeps the local structure in a low dimensional space. Since the modified ResNet-based and VGG-based networks do not have a set of pretrained weights with ImageNet, I limited our focus to the CNN with randomly initialized weights. Additionally, I trained each network 5 times and computed the average Top-1 accuracy on the test set.

The network architectures I trained additionally are derived from the ResNet architecture, namely ResNet-10 and ResNet-14, composed of exactly one convolutional layer (one bottleneck block for ResNet-14). From VGG-16, I derived the VGG-M*i*, VGG-MR*i*, VGG-B*i,* and VGG-FB*i*. The configuration will be found later in this paper. As previously stated, the primary objective for training various network architectures is that I

would like to find the smallest number of weight layers that does not degrade the classification result on our dataset.

3.3 Analysis of the Results

3.3.1 *On the number of layers*

	# Parameters	# Convolutional Layers
ResNet-10	4,913,413	10
ResNet-14	8,039,813	14
ResNet-18	11,189,893	18
ResNet-34	21,309,189	34
ResNet-50	23,582,597	50

Table 1. Estimate on the number of parameters of ResNet-derived architectures

	Top-1 Accuracy (Train set)	Top-1 Accuracy[3] (Test set)
ResNet-10	92.66%	87.88%
ResNet-14	92.56%	84.73%
ResNet-18	92.94%	87.09%
ResNet-34	93.13%	95.13%
ResNet-50	92.83%	87.85%

Table 2. Top-1 Accuracy of models based on various ResNet framework-based architectures

[3] Top-1 Accuracy is equivalent to the ratio that the classifier's prediction matches the ground truth. I will use this notation throughout this paper.

Table 3 shows that the ResNet-derived networks, namely ResNet-10 and ResNet-14, achieved a comparable training result to our baseline model ResNet-34. However, the result also shows that the classifier does not benefit from the increase of depth of a neural network, with our dataset. It also shows that our ResNet-derived networks could retain the accuracy up to 87.88%. I hypothesize that the state-of-the-art CNN networks, including the ResNet networks, are meant to classify the vast amount of ImageNet dataset with more than 1000+ classes. Thus, this characteristic might result in such a stagnant series of results with our dataset. There is a possibility that the accuracy could be ameliorated if the pre-trained ImageNet weights were available to these networks. However, as discussed before, such sets of weights are not available with the implementations of our choice.

	# Parameters	# Layers
VGG-M1 (w/o FC layers)	17,477	1
VGG-B1 (w/o FC layers)	17,477	1
VGG-B2 (w/o FC layers)	107,103	2
VGG-M1	29,652,741	3
VGG-M2	42,571,653	4
VGG-MR0	123,477,125	7
VGG-MR2	123,661,637	9
VGG-16	134,281,029	16

Table 3. Estimate on the number of parameters of VGG-derived architectures

	Top-1 Accuracy (Train set)	Top-1 Accuracy (Test set)
VGG-M1 (w/o FC layers)	71.52%	68.45%
VGG-B1 (w/o FC layers)	82.59%	77.03%
VGG-B2 (w/o FC layers)	86.52%	81.27%
VGG-M1	83.33%	76.91%
VGG-M2	90.03%	85.87%
VGG-MR0	90.1%	86.6%
VGG-MR2	89.49%	87.94%
VGG-16	93.00%	88.14%

Table 4. Top-1 Accuracy of models based on various VGG framework-based architectures

I modified, in a similar way, the VGG-16 neural networks as follows:

- VGG-M*i* (*i* = 1, 2) denotes the VGG-like network with no convolutional layer except the *i*-th and its preceding block(s). Each block consists of only one convolutional layer and retains its max pooling layer.
- The VGG-MR*i* denotes the VGG-like network with a reduced number of convolutional layers to 1 except the *i*-th block and its preceding block(s)
- The VGG-B*i* denotes the VGG-like network with an ONLY convolutional *i*-th block and its preceding block(s). Each block consists of an ONLY one convolutional layer.

a) The study shows, in an empirical way, that reducing the number of convolutional layers by 2 (in some cases, up to 4) in the VGG-16 network could retain the accuracy more or equal to 85%. The existence of the max-pooling layers in the configuration could improve a bit the performance of the classifier, but not in a substantial way. (Compare the VGG-B2 and VGG-FB2)

b) Table 4 shows that the number of convolutional layers could matter regarding classification, more than its number in each block in the VGG network.

I could not find out the conclusive difference in performance between the model with fully-connected layers and the model without them since its existence often exhausted the resource of the testing environment. However, I found out the difference of two models above regarding the resulting model size; the model containing the fully-connected layers

is, at worst, ten times bigger than the one without the layers. I believe that it is due to the additional weights a fully-connected layer contains.

3.3.2 *On the images with a modified resolution*

The more we try to map a large area with the aid of a UAV or a satellite, the larger the scale of the image becomes. Hence, the volume of information in one pixel becomes disproportionally smaller. Therefore, I conducted several studies when applying the downsampling and upsampling the images. For this purpose, I reduced all the resolution of the train and validation datasets to 224 x 224, the standard spatial dimension of each network. Then, I modified the resolution of the resulted validation dataset, specified in the following table.

	Train crop size	Test crop size	Top-1 Acc % (Test)
ResNet-10	224 x 224	28 x 28	67.27%
ResNet-10	224 x 224	56 x 56	80.49%
ResNet-10	224 x 224	112 x 112	81.45%
ResNet-10	224 x 224	224 x 224	87.88%
ResNet-10	224 x 224	320 x 320	88.13%
ResNet-10	224 x 224	480 x 480	90.45%
ResNet-14	224 x 224	28 x 28	64.01%
ResNet-14	224 x 224	56 x 56	80.35%
ResNet-14	224 x 224	112 x 112	79.18%
ResNet-14	224 x 224	224 x 224	84.74%
ResNet-14	224 x 224	320 x 320	88.85%
ResNet-14	224 x 224	480 x 480	91.14%
ResNet-34	224 x 224	224 x 224	95.13%

Table 5. Top-1 Accuracy of models based on ResNet-10 and ResNet-14, with different test cropping size

The study shows the ResNet-derived networks managed to achieve a comparable result even if the input dimension is reduced. This shows that the generalization capabilities of the ResNet framework would not be impaired by the reduced number of residual blocks.

However, if the spatial dimension of the inputs is more than 50% smaller than the CNN's input dimension, it shows that the classification performance is dropped by 15%. If the resolution of the input is equal to one-tenth of the input dimension, the accuracy is sharply decreased to 64%. Table 6 shows a similar result to support our previous claim.

It is interesting that, with lesser number of convolutional layers, our VGG-derived networks, VGG-MR0 and VGG-MR2, produce marginally better Top-1 Accuracy results (5 – 13%) than our ResNet-derived networks, ResNet-10 and ResNet-14. However, one must take also into consideration that the ResNet-derived networks have significantly lower numbers of parameter than the numbers of VGG-derived networks; ResNet-derived networks have less than the one-tenth number of parameters than the VGG-derived networks.

	Train crop size	Test crop size	Top-1 Acc % (Test)
VGG-MR0	224 x 224	28 x 28	77.75%
VGG-MR0	224 x 224	56 x 56	81.96%
VGG-MR0	224 x 224	112 x 112	77.75%
VGG-MR0	224 x 224	224 x 224	86.6%
VGG-MR0	224 x 224	320 x 320	88.63%
VGG-MR0	224 x 224	480 x 480	88.48%
VGG-MR2	224 x 224	28 x 28	72.4%
VGG-MR2	224 x 224	56 x 56	82.34%
VGG-MR2	224 x 224	112 x 112	81.11%
VGG-MR2	224 x 224	224 x 224	87.94%
VGG-MR2	224 x 224	320 x 320	88.32%
VGG-MR2	224 x 224	480 x 480	88.54%
VGG-16	224 x 224	224 x 224	88.14%

Table 6. Top-1 Accuracy of models based on VGG based networks, with different test cropping size

Using regularization techniques (e.g., Dropout layer) could yield a better classification accuracy to our derived networks, but that could be left for our future study. However, the objective of this study is to observe if the CNN-based classifier with fewer layers can maintain the comparable classification accuracy. Table 6 shows that there is a marginal decrease in classification accuracy if the overall resolution of the test dataset decreases from the original resolution to its downsampled resolution. The Top-1 Accuracy for both of our derived networks (VGG-MR0 and VGG-MR2) decreases by 6 ~ 9% (5%) if our input image's resolution is reduced to 112 x 112 (56 x 56) pixels. If the input is further reduced to 28 x 28 pixels (12.5 % of the original resolution), the accuracy plummets by 10 ~ 15%.

4. CONCLUSIONS

Striking an appropriate balance between the size of a neural network and the characteristics of a dataset is critical. Using our dataset, I have shown that a simpler artificial neural network derived from the current state-of-the-art CNN framework could achieve a satisfiable classification result with marginally reduced accuracy. If one can afford a specific loss of accuracy for the sake of the feasibility, I believe that one could produce a vegetation map with a larger area, using a similar approach. Primarily, I find it somewhat surprised that a CNN could handle the downsampled images this well (6-10% performance drop), based just on the original input size. (224×224 pixels) The performance of derived networks remains to be seen, i.e., verifying a similar performance can be achievable with different datasets. Still, in that scenario, the experimental results imply that this could be applied for on-site inspection of farmland. Although it would

require a computer equipped with a decent GPU to train such a model, I hope that our modified models could fit any embedded device thanks to the reduced model size.

For the future work, I will apply our findings from this study to imageries containing the much larger area. I will also work with various CNN frameworks to validate empirically that our approach is feasible with most of the frameworks currently available.

LITERATURE CITED

[1] Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, *29*(9), 2352-2449.

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[3] Wang, J., & Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning (No. 300). Technical report.

[4] Torralba, A., & Efros, A. A. (2011, June). Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 1521-1528). IEEE.

[5] Zou, W. W., & Yuen, P. C. (2012). Very low-resolution face recognition problem. *IEEE Transactions on Image Processing*, *21*(1), 327-340.

[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016, October). Identity mappings in deep residual networks. In *European Conference on Computer Vision* (pp. 630-645). Springer, Cham.

[7] Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

[8] Zhu, X., Vondrick, C., Ramanan, D., & Fowlkes, C. C. (2012, September). Do I Need More Training Data or Better Models for Object Detection? In *BMVC* (Vol. 3, p. 5).

[9]

Taylor, L., & Nitschke, G. (2017). Improving Deep Learning using Generic Data
[10] Augmentation. *arXiv preprint arXiv:1708.06020*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information*
[11] *processing systems* (pp. 1097-1105).

Chen, X. W., & Lin, X. (2014). Big data deep learning: challenges and
[12] perspectives. *IEEE access*, *2*, 514-525.

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors*, *16*(8),
[13] 1222.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The*
[14] *Journal of Machine Learning Research*, *15*(1), 1929-1958.

Shen, L. (2017). End-to-end Training for Whole Image Breast Cancer Diagnosis
[15] using An All Convolutional Design. *arXiv preprint arXiv:1708.09427*.

Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated
[16] convolutions. *arXiv preprint arXiv:1511.07122*.

Masko, D., & Hensman, P. (2015). The impact of imbalanced training data for
[17] convolutional neural networks.

Bongiovanni, R., & LoInberg-DeBoer, J. (2004). Precision agriculture and
[18] sustainability. *Precision agriculture*, *5*(4), 359-387.

Suh, S. H., Kim, D. Y., Jhang, J. E., Byamukama, E., Hatfield, G., & Shin, S. Y. (2017, September). Identification of the White-Mold affected Soybean fields by

using Multispectral Imageries, Spatial Autocorrelation and Support Vector Machine. In *Proceedings of the International Conference on Research in Adaptive*

[19] *and Convergent Systems* (pp. 104-109). ACM.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for

[20] large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. *International*

[21] *Journal of Computer Vision*, *115*(3), 211-252.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning

[22] applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Duda, R. O., & Hart, P. E. (1973). Pattern classification and scene analysis. *A*

[23] *Wiley-Interscience Publication, New York: Wiley, 1973*.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*

[24] (Vol. 1). Cambridge: MIT Press.

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image

[25] classification using deep learning. *arXiv preprint arXiv:1712.04621*.

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *arXiv preprint*

[26] *arXiv:1312.4400*.

He, K., & Sun, J. (2015). Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5353-5360).