UKRAINIAN CATHOLIC UNIVERSITY

MASTER THESIS

Automatic Plant Counting using Deep Neural Networks

Author:
Oleh PIDHIRNIAK

Supervisor:
Orest KUPYN

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in the

Department of Computer Sciences
Faculty of Applied Sciences

Lviv 2019

Declaration of Authorship

I, Oleh PIDHIRNIAK, declare that this thesis titled, “Automatic Plant Counting using Deep Neural Networks” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UKRAINIAN CATHOLIC UNIVERSITY

Abstract

Faculty of Applied Sciences

Master of Science

Automatic Plant Counting using Deep Neural Networks

by Oleh PIDHIRNIAK

Crop counting is a challenging task for today's agriculture. Increasing demand for food supplies creates a necessity to perform farming activities more efficiently and precisely. Usage of remote sensing images can help to better control the population of the plants grown and forecast future yields, profits and disasters. In this study we offer a series of approaches for plant counting using foreground extraction algorithms, deep neural networks. The study introduces innovative to the field approach of densely distributed plants counting using density map regression with the accuracy of 98.9% on palm oil trees dataset.

Acknowledgements

I would like to address my gratitude to my father Yaroslav Pidhirniak for introducing me to the world of the science and my mother Tetyana Pidhirniak for her everlasting support. Special gratitude I address to Oleksandr Romanko for cultivating an interest in me to the field of Data Science as whole and agriculture in particular. Also I want to appreciate my supervisor Orest Kupyn for his support and advices.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

As a society of the third millennium, humankind starts to master every aspect of its day to day life, making it more efficient in terms of time and resources. One of the most critical sides of human life is her nutrition, and we can notice severe advancements in the agricultural sector that satisfies these human needs. Improvements such as precision farming, fertilizers, etc. are helping to utilize Earth resources most efficiently, but there is still much room for advancement.

Recent development in Computer Vision methods and techniques allowed to achieve human-level accuracy in various applied tasks such as object detection and counting. This collective work of thousands of researchers is already making life more comfortable and allowing more efficient use of resources in areas such as driver-less cars, face recognition, photos enhancements, etc.

The agricultural sector has a well-defined task for these new developments in computer vision sphere - crops counting for better yield prediction, segmentation of problematic crop areas such as plants beaten down due to weather activity, flooded regions of the fields, detection of plant diseases, etc. Solving described tasks allows farmers to better prepare for upcoming dangers to their yield, minimize their losses and maximize harvest and profits.

In recent decades farmer started using aerial and satellite images in order to understand their crops better. Usage of such means provided the way to look at fields at previously unseen scales. However, it is not always clear how to process such high amounts of visual data, and often farmers use human force to process it which makes it not cost and time efficient. High-resolution images generated by such activities are a perfect target for application of computer vision methods. Imagery from airplanes allows running crop counting algorithms and segmentation algorithms to find hazardous or problematic areas. With the development of drone manufacturing farmers gained the ability to use autonomous drones to fly over their fields and take low height images of the plants. Such images allow scientist to use computer vision algorithms in order to identify plant diseases or harmful pests. Early detection of such problems allows farmers to prevent further spread of problem agents and utilize resources more efficiently.

In this work, we will make an overview of existing solutions to the task of counting and segmentation of crops on aerial images. Also, we will provide our solution to the mentioned tasks using our datasets. We will describe the process of data preparation, tested approaches to for segmentation and counting tasks.tasks.

Chapter 2

Related Work

This work is centered around the task of object instance counting, which obliges us to perform an overview of methods and techniques used for such tasks.

2.1 Artificial Neural Networks

Let's assume that we have a set of objects U that can be characterized with properties $(x_1, ..., x_n)$. Each object has an assigned value or label y. The main task that for the artificial neural network is to construct a function $F : X \mapsto Y$ such that given a vector of object characteristics X it returns the label or value that object is associated with. If y is a numeric value, such a task is called regression; in case of categorical value, we call such task classification. Neural networks are machine learning algorithms meaning that they need some initial dataset D to train on, the bigger the size of one the better. The main goal of the training process of a neural network is to minimize an error function

$$L(D, F) = \sum_i F(x_i) \neq y_i \tag{2.1}$$

, meaning finding optimal function

$$W^* = argmin_W L(D, F(W)) \tag{2.2}$$

The neural network itself is a sequence of transformations of input vector x with functions $F_1, ..., F_n$ where F_n outputs value y associated with object described by vector x. F_i is the so-called neural network layer consisting of neurons. A neuron is an object that is characterized by its input size, weight vector and activation function. The neural network layer is simply a collection of neurons that work on the same characteristics of an object. The function F1 is one layer of such neurons, and after applying the function, we get some new space of features. Then we apply another such layer to this feature space. There may be a different number of neurons, some other non-linearity as a transforming function. Thus, consistently applying these transformations, we get the common function F - the transformation function of the neural network, which consists of the sequential application of several functions. Artificial neural networks have gained high popularity recently due to their state-of-the-art performance in problems of signal processing, handwriting recognition, speech to text, weather forecasting, and face recognition.

2.2 Convolutional Neural Networks

Convolutional Neural Networks architectures are inspired by the results of biological research of visual cortex area of the human brain conducted by D. Hubel and T. Wiesel. The main ideas that have been used are the localization of the zones of perception and the division of neurons by functions within one layer. The task of neuron, in this case, is to monitor its receptive field and recognize the image on which it was trained. Simple neurons are collected in groups (planes). Within one group, simple neurons are tuned to the same stimulus, but each neuron watches its fragment of the receptive field. Together, they look through all the possible positions of this image. All simple neurons of the same plane have the same weight, but different receptive fields. One can imagine the situation differently, that this is one neuron that knows how to try on its image at once to all positions of the original image. All this allows one to recognize the same image regardless of its position. Recent advancement in the area of convolutional neural networks shows that they already can outperform humans in some image recognition tasks.

2.3 Object Detectors

Next step in the evolution of CNN's and classifiers based on them are object detectors such as single shot detector Liu et al., 2015 and "you only look once" (YOLO) Redmon et al., 2015 and its descendants - YOLOv2 and YOLOv3. These detectors are performing well on various multiclass dataset such as Pascal VOC[Everingham et al., 2010], COCO[Lin et al., 2014], etc. and theoretically can be applied to the task of crop counting on remote sensing images. Another type of detectors are R-CNN, Fast R-CNN, and Faster R-CNN He et al., 2017 that use features pyramid networks(FPN) Lin et al., 2016 in order to perform bounding box prediction of objects we are interested in. This approach extends possibilities of further data processing since last layers of the underlying neural networks provide multiclass labels, bounding boxes and accuracy scores.

2.4 Foreground extraction

Foreground/background extraction task segmentation in still images is of great practical importance in image editing and processing. This task is exhaustively described in Rother, Kolmogorov, and Blake, 2004. The algorithm starts it work with initial user provided rectangle around the foreground region (foreground region should be completely inside the rectangle). Then algorithm segments it iteratively to get the best result. Firstly it uses a Gaussian Mixture Model(GMM) model the foreground and background. GMM learns and create new pixel distribution. That is, the unknown pixels are labelled either probable foreground or probable background depending on its relation with the other hard-labelled pixels in terms of color statistics. Based on obtained pixel distribution a weighted graph is built with lower weights assigned to edges connecting pixels with larger difference in color. Further the mincut algorithm is applied. The process is continued until the classification converges. In case of crop counting task this algorithm is useful for plants with sparse distribution over the field area.

2.5 Crop counting tasks

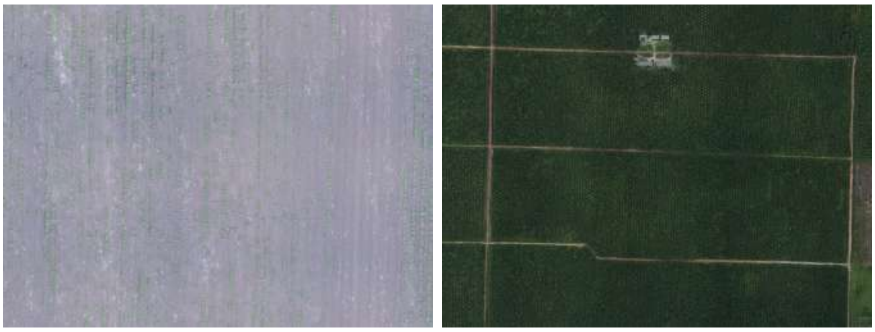
One of the methods that operates on data similar to ours was described in D.A. Pouliot, 2002. Authors here apply local maximum filter and analyze local transects coming out from a potential tree crown. They achieved tree-detection accuracy of 91%. In A. Manandhar, 2016 authors use polar shape matrix in order to perform palm trees detection using circular autocorrelation. One of the most popular works in the area of palm detection on unmanned aerial vehicles images is Malek et al., 2014. Authors here perform extraction of keypoints using the Scale-invariant Feature Transform (SIFT) with further classification on pretrained Extreme learning machine. Koon Cheang, Koon Cheang, and Haur Tay, 2017 proposed a deep neural network approach to the task of palm counting. In order to complete the counting task they were using sliding window approach – taking patches from full scale images using a window of the size close to average palm bounding box. After obtaining the patch image is forwarded to a neural net based classifier that makes a decision about the class of the image.

In further chapters we will describe implementation and performance of sliding window-classifier, and various object detectors on our datasets, discussion about benefits and drawbacks of each method will be provided. Since every method requires a different data to perform learning and processing we will also discuss methods to prepare raw data for each described approach.

Chapter 3

Data

In order to gain a better understanding of machine learning algorithms that need to be applied to complete instance segmentation task we made an overview of an existing dataset and assembled a plan to transform raw data into a format suitable for algorithms. Available to us data sources contained data of three kinds:



(A) Sugar beet.

(B) Palm trees.

FIGURE 3.1: Example of available data.

This data was provided by Skyglyph © company from their archives. These are the photos from aerial vehicles taken from height 50-100 meters. The data was a single image of field area with resolution 11254px × 6858px.

After preliminary analysis it was decided to create markup on figure 3.1b for areas containing palm trees. Labeling was performed in software called Labelbox. Regarding image 3.1a decision was made to process it without labeling every plant instance.

Labeling of palms on image 3.1b was done basing on one class - "palm." The complete count of palm trees after labeling was 15947 palms. The data was initially labeled in the form of

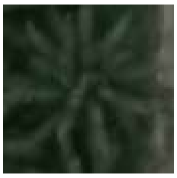


FIGURE 3.2: Example of item from class "Palm"

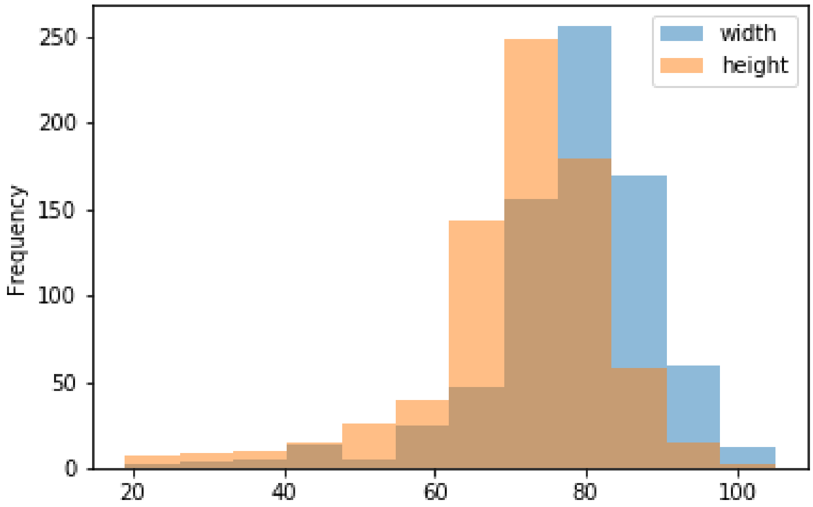


FIGURE 3.3: Palms sizes distribution

a palm bounding box and exported in Pascal VOC format described in Everingham et al., 2010. Example of an item from class palm can be seen on 3.2. Such labeling allowed for further extraction of palms center needed for various machine learning algorithms described further in this work.

During labeling we obtained a distribution of palm trees sizes provided on 3.3.

To get a better understanding of the data, we provide statistics on palm sizes in table 3.1. As one can notice palms tend to be of a similar size with comparatively small deviation from the mean.

	Width px	Height px
mean	78.36	71.31
std	11.33	12.27
min	25.00	19.00
max	105.00	104.00

TABLE 3.1: Palms sizes statistics.

Chapter 4

Proposed Methods

The primary objective of this work can be described with the following task - given the image of the field perform segmentation of single instance of plant and calculate the number of plant instances. Such a task can be achieved in several ways varying from getting instances to count directly to using a model to mark every instance with a marker with further counting of markers.

In the case of sparse spatial distribution of plant instances, the objective can be achieved with foreground extraction algorithm such as GrabCut [Rother, Kolmogorov, and Blake, 2004]. Plant instances can be counted by introducing geometric metric and its further application to segmented areas.

Cases with more complex data where plant are populated closely, and overlap require more sophisticated approaches. The most crucial step for segmenting such data is to select distinguishable keypoints from a single plant specimen that the algorithm will search for. In this work, we will mainly concentrate on the application of deep neural networks for finding such keypoints. The first approach that was applied for such data was image scanning with sliding window and further forwarding the cut patches to image classifier that decides whether the provided object belongs to class "palm" or not. Another approach is to use classifiers fine-tuned to calculate regression task. We can modify some of the popular classifiers with two dense layers at the to provide the count of instances on the images depending on feature maps that classifier learned.

Last, but not least method that can be used is "encoder-decoder" like approach in U-Net network [Ronneberger, Fischer, and Brox, 2015]. The benefit of this solution is that we preserve global context while achieving good localization characteristics.

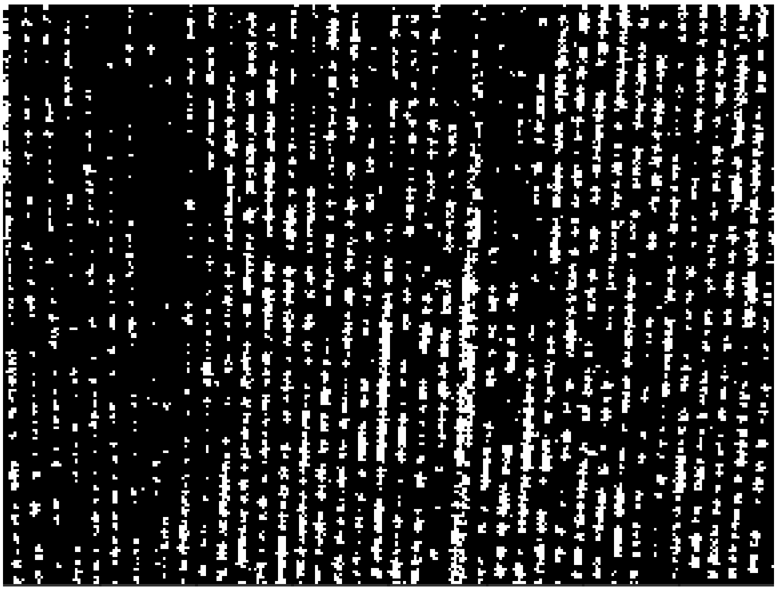


FIGURE 4.1: Mask of pixels with dominating green channel.

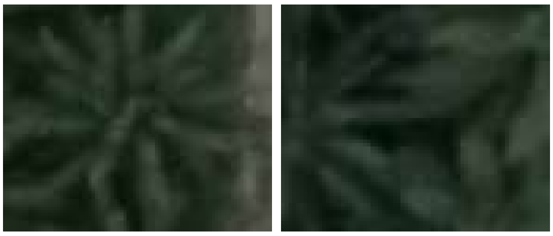
4.1 Foreground extraction

This approach performs well in cases with clearly distinguishable background and foreground in the image. Starting with a user-specified bounding box around the object to be segmented the algorithm estimates the color distribution of the target object and that of the background using a Gaussian mixture model. The last is used to construct a Markov random field over the pixel labels, with an energy function that prefers connected regions having the same label, and running a graph cut based optimization to infer their values. As this estimate is likely to be more accurate than the original, taken from the bounding box, this two-step procedure is repeated until convergence.

The initial guess was chosen as all pixels where green channel dominates over red and blue on 3.1a. After that initial guess was passed to the GrabCut algorithm to improve results of foreground extraction. When final segmentation is achieved the task is to count instances in the segmented areas. Since we can notice strong domination of vertical vectors of crops distribution the count plants can be roughly estimated by dividing blob height by average height of the plant.

4.2 Classifiers and a sliding window for dense crop clusters

There are a wide variety of image classifiers available right now in the field of computer vision. Modern implementations can easily distinguish between images on 4.2. To achieve the formulated task of counting the crop instances we need to supply such images to the classifier. This can be achieved by iterating over the image 3.1b with sliding window and forwarding the patches to the classifier.



(A) Palm. (B) Mid palm area.

FIGURE 4.2: Classifier classes

The main advantage of this method is the simplicity of implementation.

On the other hand, the accuracy of the counting depends on the size of the palm and sliding window size. This means that usage of the trained model on previously unseen images with different geometric parameters (palm size, camera height, directions of plants) becomes harder as user needs to adjust them when performing inference on new data. Also to ensure that all plants were covered by sliding window it has to make stride smaller than palm size. This means that there can be a couple of positive results for single palm which need to be merged in a subsequent step.

4.3 Density map regression

The main goal of this approach is to predict a heatmap (2D matrix) for each class so that we can sum elements of matrix elementwise and get the count of class instances. In order to achieve this goal, we need to generate ground truth data that the model will learn with. To conform to summarization criteria the simplest way is to use 2D Gaussian kernel:

$$G_{2D} = -\frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} \tag{4.1}$$

. Since the integral over 4.1 is always equal to 1 – an element-wise sum of such heatmap will be equal to the count of marked instances on the initial image. Loss function was chosen to be Frobenius norm of matrix $H_R = H_T - H_P$, where H_T – ground truth heatmap and H_P predicted heatmap.

$$||H_R|| = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |h_{ij}^2|} \tag{4.2}$$

It was chosen to use U-Net neural network that has "encoder-decoder" architecture. Encoder part consists of convolution blocks followed by a maxpool downsampling to encode the input image into feature representations at multiple different levels. The decoder part of the network consists of upsample and concatenation followed by regular convolution operations. The main contribution of U-Net in this sense compared to other fully convolutional segmentation networks is that while upsampling and going deeper in the network we are concatenating the higher resolution features from down part with the upsampled features in order to better localize and learn representations with following convolutions. Since upsampling is a sparse operation, we need a good prior from earlier stages to represent the localization better.

4.3.1 Training

All of the models were implemented using PyTorch deep learning framework Paszke et al., 2017. The training was performed on single Nvidia GeForce GTX 1060 on datasets described earlier. Part of dataset 3.1b was labeled with Gaussian kernels in centers of the palms resulting in 4.4. The model that we choose was U-Net with pretrained ResNet-34 convolution layers. Such

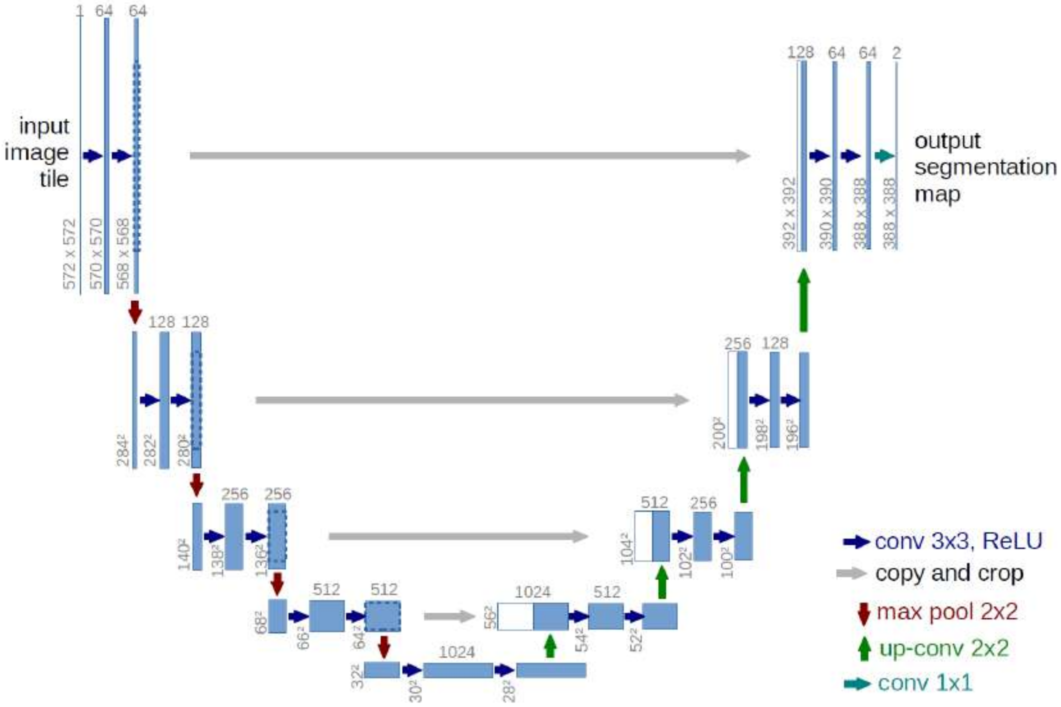


FIGURE 4.3: U-Net architecture

technique with reusing pretrained layers is called transfer learning and is described in Yosinski et al., 2014. Training was performed on a subset containing 942 palms which are 5% of whole dataset size. Train dataset was split into two parts 85% for training, 5% for validation

4.4 Deep Neural Networks for Regression

One more method that could be used for crop instance counting is using deep neural networks trained for classification task and fine-tuned for the regression task. The idea here is to add a final layer that would output the count of crop instances in the input image. This method is really simple to implement. On the other hand, this method generalizes poorly on previously unseen data in case of counting task.

4.4.1 Training

To probe this method we used ResNet-50 CNN pretrained on Imagenet dataset. The last layer of the network was dropped

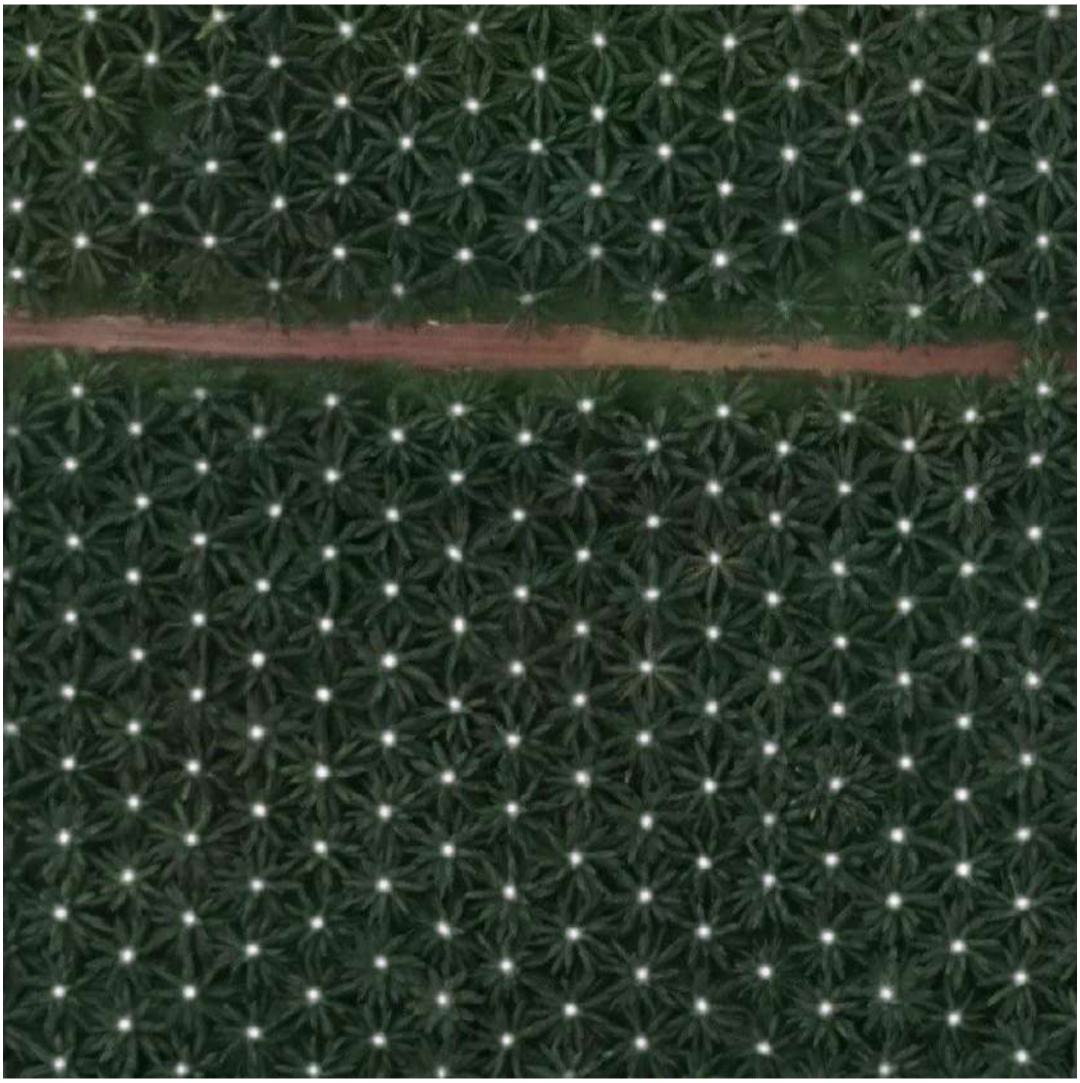


FIGURE 4.4: Palms heatmap

and replaced with two dense layers. The original ResNet-50 had a fully-connected layer with dimension 2048 × N, where N is the number of classes that we classify into. We added two layers in order to increase the capacity of the network. The first layer was a dense layer with dimension 2048 × 256 with ReLu activation function. The second layer with dimension 256 × 1.

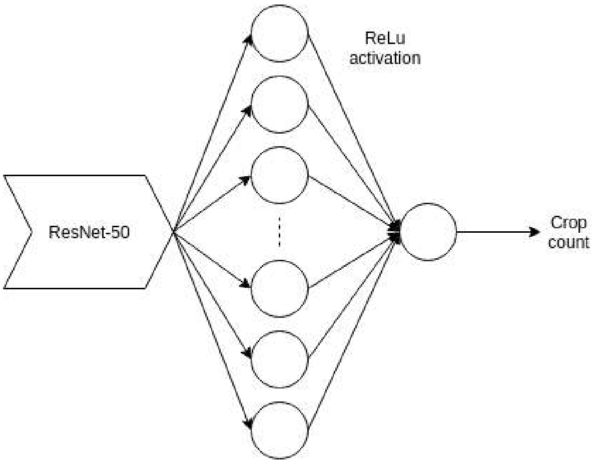


FIGURE 4.5: Fine-tuned architecture of Neural Network.

The network was using image patch of size 224 × 224 as input and number C as a count of plant instances as the value to approximate prediction to. Train subset used was the same as in 4.3. The subset was reformatted into a sequence of 70 image patches of size 224 × 224. In order to compare predictions to ground truth MSE loss was used, same as in 4.3. Ground-truth labels were normalized:

$$x_{normalized} = |\frac{x - m}{d}| \tag{4.3}$$

here x – original instances count, $m = 13.18px$ – mean count over all images, $d = 3.37px$ – standard deviation over all images.

Chapter 5

Experimental evaluation

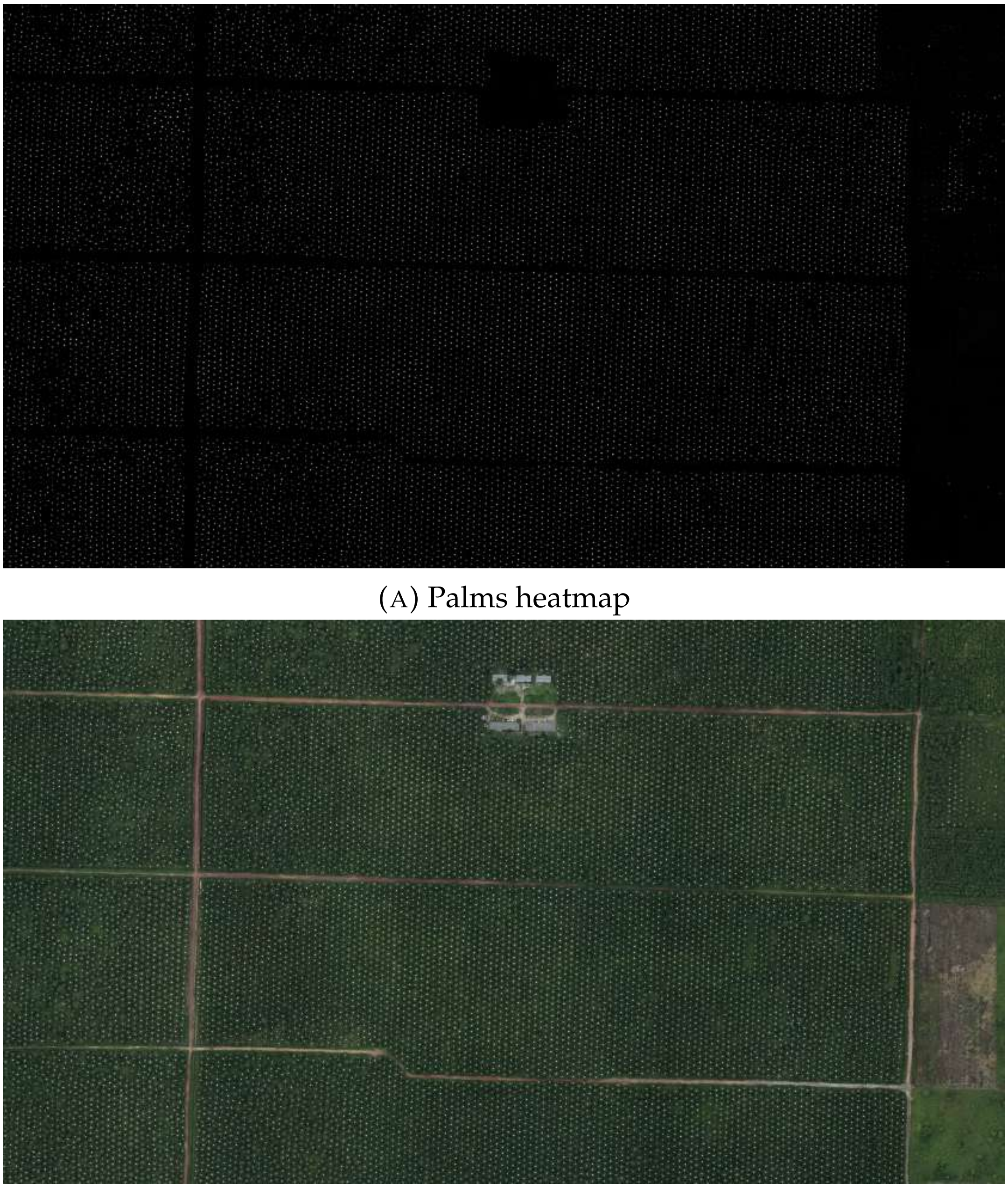
Here we will describe the results obtained from different approaches to counting tasks and compare them against each other. Our comparison will be focused around the ground truth palms count of 15947 and how well various models performed when counting on raw image. One more important thing to take into account is the size of the training dataset needed to achieve the desired accuracy level.

5.1 Density map regression

Application of density map regression model to the whole image resulted in heatmap on 5.1. Most palms were marked with white circles that deep neural network was taught to recognize and produce on unseen images. Summarization of the resulting heatmap gave us the count of 15784 palms which is 98.9% of accuracy.

5.1.1 Coordinates detection

Since we obtain a mask with clearly distinguishable circles of the same color, we can use blob detectors in order to calculate palm center coordinates. In order to detect the coordinates we used "SimpleBlobDetector" from Bradski, 2000. The result can be seen on 5.2



(A) Palms heatmap

(B) Palms heatmap stacked on top of input image

FIGURE 5.1: Density map regression model results

5.2 Classifier based regression

Application of the method described in 4.4 gave us the result of 20448 when applied to the whole dataset. Compared to the ground truth value of 15947 plant instances this gives us 28% of error. This margin is unacceptable for crop counting task since it induces large uncertainty when used in further modeling of plants lifecycles and revenue estimations. The method might

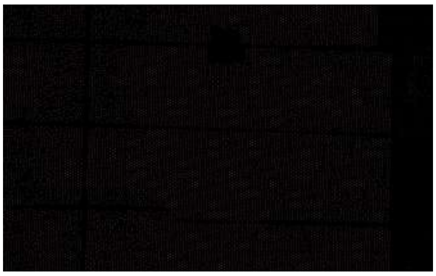


FIGURE 5.2: Blob detector results.

be improved with the extension of the training set with more data, but this task was left out of the scope of the current study since one of the requirements for the task was minimal size of the training dataset.

5.3 Foreground extraction

Mask that was obtained from the foreground extraction algorithm applied to the case of sugar beet had the area of 76023641 square pixels. Based on the average plant size of 42 × 48 pixels we can calculate that there are 37710 plant instances on the field. This method showed the accuracy of 93% which were enough to make further predictions and estimates with the help of experts in agriculture.

5.4 Sliding window with Classifier

Evaluation of approach when we try to classify patches of the initial image 3.1b was not completed, since this approach was triggering several classifier activations on palm class for single plant instance as can be seen on 5.3.

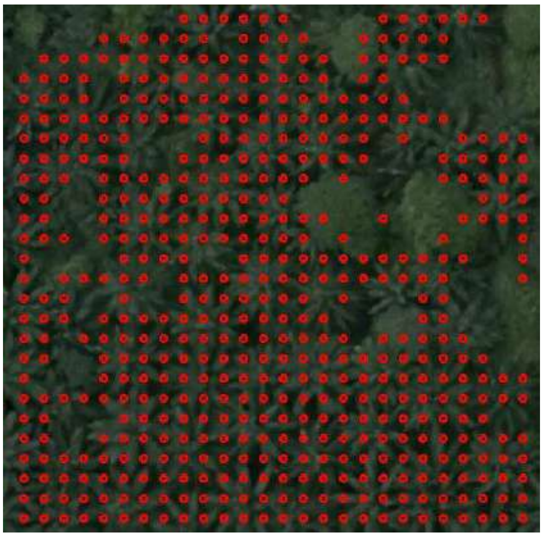


FIGURE 5.3: Sliding window iteration result.

Chapter 6

Conclusions

In this work, we described the task of plant instances counting. Reviewed possible variations of such tasks depending on plants spatial distribution density – cases with overlapping and sparse plants. Proposed different methods to count plant instances such as foreground extraction with further estimation based on average plant size in case of sparse distribution. Various Deep Neural Network approaches where proposed such as:

- classification of the patches provided by sliding window
- density map regression with U-Net architecture
- classifier fine-tuned for regression tasks.

Numerical evaluations were provided for density map regression application and classifier regression for palms counting task and estimation of sugar beet based on crop canopy cover metric.

Bibliography

A. Manandhar L. Hoegner, U. Stilla (2016). "Palm tree detection using circular autocorrelation of polar shape matrix". In: *SPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume III-3, 2016 XXIII ISPRS Congress*. Photogrammetry and Remote Sensing, Technische Universitaet Muenchen, pp. 465–472. URL: https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/III-3/465/2016/isprs-annals-III-3-465-2016.pdf.

Bradski, G. (2000). "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools*.

D.A. Pouliot D.J. King, F.W. Bell D.G. Pitt (2002). "Automated tree crown detection and delineation in high-resolution digital camera imagery of coniferous forest regeneration". In: *Remote Sensing of Environment 82*. Remote Sensing of Environment 82, p. 325. URL: https://pdfs.semanticscholar.org/90ec/b9d3e4197ca651988ed530c226e02eb3e58a.pdf.

Everingham, M. et al. (2010). "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88.2, pp. 303–338.

He, Kaiming et al. (2017). "Mask R-CNN". In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988.

Koon Cheang, Eu, Teik Koon Cheang, and Yong Haur Tay (2017). "Using Convolutional Neural Networks to Count Palm Trees in Satellite Images". In: *arXiv e-prints*, arXiv:1701.06462, arXiv:1701.06462. arXiv: 1701.06462 [cs.CV].

Lin, Tsung-Yi et al. (2014). "Microsoft COCO: Common Objects in Context". In: *arXiv e-prints*, arXiv:1405.0312, arXiv:1405.0312. arXiv: 1405.0312 [cs.CV].

Lin, Tsung-Yi et al. (2016). "Feature Pyramid Networks for Object Detection". In: *arXiv e-prints*, arXiv:1612.03144, arXiv:1612.03144. arXiv: 1612.03144 [cs.CV].

Liu, Wei et al. (2015). “SSD: Single Shot MultiBox Detector”. In: *arXiv e-prints*, arXiv:1512.02325, arXiv:1512.02325. arXiv: 1512.02325 [cs.CV].

Malek, S. et al. (2014). “Efficient Framework for Palm Tree Detection in UAV Images”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 7.12, pp. 4692–4703. ISSN: 1939-1404. DOI: 10.1109/JSTARS.2014.2331425.

Paszke, Adam et al. (2017). “Automatic differentiation in PyTorch”. In:

Redmon, Joseph et al. (2015). “You Only Look Once: Unified, Real-Time Object Detection”. In: *arXiv e-prints*, arXiv:1506.02640, arXiv:1506.02640. arXiv: 1506.02640 [cs.CV].

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *arXiv e-prints*, arXiv:1505.04597, arXiv:1505.04597. arXiv: 1505.04597 [cs.CV].

Rother, Carsten, Vladimir Kolmogorov, and Andrew Blake (2004). “"GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts”. In: *ACM Trans. Graph.* 23.3, pp. 309–314. ISSN: 0730-0301. DOI: 10.1145/1015706.1015720. URL: http://doi.acm.org/10.1145/1015706.1015720.

Yosinski, Jason et al. (2014). “How transferable are features in deep neural networks?” In: *arXiv e-prints*, arXiv:1411.1792, arXiv:1411.1792. arXiv: 1411.1792 [cs.LG].