UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Ashraf Abbasov

Text Region-Based Convolutional Neural Network for Precision Agriculture

Master's Thesis (30 ECTS)

Supervisor(s): Kallol Roy, PhD
Indrek Virro, MSc

Tartu 2023

Text Region-Based Convolutional Neural Network for Precision Agriculture

Abstract: Application of Neural Networks in Precision Agriculture is now more widespread than ever. Neural networks have been extensively used in various tasks in precision agriculture, such as plant detection, disease detection, yield estimation, and soil classification.

In this thesis, we build a blueberry plant image dataset for object detection with additional directional text that indicates the where in the image blueberry plant is. We train the Region-based Convolutional Neural Network (RCNN) model twice. First, using its original architecture that utilizes the Selective Search algorithm to create region proposals. Then, we modify the model by replacing Selective Search algorithm with additional text data to generate region proposals. Through performance analysis of both models on the test data, we show that the text method saves significant time on both training and inference while having good enough accuracy to compete with original model.

Keywords:
agriculture, neural networks

CERCS: P176 Artificial intelligence

Tekstiga piirkondlike ettepanekute meetoditega konvolutsiooniline närvivõrk täppispõllumajanduse rakenduse jaoks

Lühikokkuvõte:

Neuraalvõrkude rakendamine täppispõllumajanduses on nüüd laialdasemalt kui kunagi varem. Närvivõrke on laialdaselt kasutatud täppispõllumajanduse eri ülesannetes, nagu taimede tuvastamine, haiguste tuvastamine, saagikuse hindamine ja mulla klassifitseerimine.

Käesoleva lõputöö koostame objektide tuvastamiseks mustikataime pildiandmestiku koos suunatekstiga, mis näitab, kus pildil mustikataim asub. Koolitame kaks korda piirkonnapõhise konvolutsioonilise närvivõrgu (RCNN) mudelit. Esiteks, kasutades selle algset arhitektuuri, mis kasutab piirkonna ettepanekute loomiseks valikulise otsingu algoritmi. Seejärel muudame mudelit, asendades valikulise otsingu algoritmi täiendavate tekstiandmetega, et luua piirkonna ettepanekuid. Testiandmete mõlema mudeli jõudlusanalüüsi võime näitame, et tekstimeetod säästab aega nii koolituse kui ka järelduste tegemisel, olles samal ajal täpselt hea täpsusega, et konkureerida algse mudeliga.

Võtmesõnad:

põllumajandus, tehisnärvivõrgud

CERCS: P176 Tehisintellekt

Contents

List of Figures

1 Introduction

In recent years, precision agriculture has become increasingly important due to various reasons. Precision agriculture allows optimization of crop yields and reduction of waste. Ability to timely detect crop diseases and treat them with correct type and amount of chemicals is another key advantage of precision agriculture. These benefits of precision agriculture incentivised the usage of more and more state of the art technologies such as Neural Networks.

Recently there has been more and more studies that utilize neural networks for precision agriculture. Convolutional Neural Network has been used to accurately distinguish the healthy and diseased plants based on the leaf images [Fer18]. Another similar study used Deep Convolutional Neural Networks to detect tomato leaf diseases with high accuracy [AJ22]. Integrated deep learning algorithm has been used and achieved 99.50% accuracy in different research focusing on wheat disease detection [XCZ+23]

Advanced methods were not only used for disease detection but also yield estimation. Vineyard yield estimation for yield quality optimization were conducted using CNN and transformer models [OSF+22]. One research estimated the soybean yield using Deep Learning together with Generalized Regression Neural Network. Average 97.43% accuracy was achieved predicting the total weight of soybean pods [LDN+22]. Neural Networks perform efficiently on more challenging yield estimation tasks such as fruit estimation on-plant. Faster RCNN with modified IoU function has been utilized to to estimate mango, apple and orange tree yields with high enough accuracy. [BRS21]

Neural Networks has been widely used to optimize irrigation and minimize water waste in precision agriculture. In one study, Irrigation system controller were designed based on the non-linear function between amount of water provided and moisture levels in root

of the plants. This controller then was operated by Neural Network model to provide enough water to increase the moisture to desired level [CPTS08]. Another similar research uses Neural Network to predict when the moisture level will decrease to undesired level to schedule watering efficiently. The results showed in increase in efficient consumption of water and yield [GZJ+21]. Prediction for water demand of green beans were done using several machine learning models. Combination of Long short-term memory network and Convolutional Neural Network performed best using root depth, basal crop coefficient and other metrics as inputs while Support Vector Regressor performed worst. [MAAES+23]

1.1 Problem

In this thesis, we are going to use blueberry plant images provided by Virro and his colleagues at Estonian Life and Sciences University (Figure 1). Our main objective is to train an object detection model using these images. Main model architecture that we are going to use and make some alterations and experiments is Region proposals with Convolutional Neural networks (RCNN) [GDDM13]. Important alteration to RCNN in our thesis will be the usage of text annotations to replace the extremely time consuming Selective Search [UvdSGS13] algorithm which is the algorithm chosen to generate region proposals for RCNN in original paper. Firstly, we are going to build a dataset from provided blueberry images by defining blueberry plants with bounding boxes. In addition to bounding boxes, we are going to add text annotations to each image in the dataset to generate region proposals. In our RCNN architecture for Convolutional Neural Network part of the architecture we are going to use pretrained VGG16 CNN model [SZ15]. This model will be further trained with blueberry dataset twice with region proposals from selective search algorithm and region proposals from our text annotations.

Figure 1. Blueberry plant

1.2 Contributions

We suggest new very primitive algorithm based on directional text to generate region proposals to replace the Selective Search algorithm. This simple algorithm then later can be turned into text classification to classify several sentences about the image to break the problem down into our simple algorithm. Both models - Selective Search proposed regions and text proposed regions - are compared by average precision and time they take to process an image. Next few chapters will go in depth into architecture of RCNN and both region proposal algorithms.

2 Background

The following section will examine the various components that make up the RCNN architecture and offer insights into the methods employed in this thesis. We will delve into the technical details of the RCNN architecture and provide a comprehensive overview of the research methodology adopted in this study.

2.1 Convolutional Neural Network

Convolutional Neural Networks are type of a deep learning model that primarily used on object detection, classification for image and video input types. CNNs are made of several layers and each layer does specific computation on the input data (Figure 2). These layers are input layer, convolutional layer, pooling layer, and fully-connected layer. Input layer consist of pixels values of the image. Convolutional layer consist of set of learnable filter that extracts the important features from image data by applying convolution. Convolution layers usually are followed by non-linear activation filter (e.g ReLU) to introduce non-linearity to the system. Pooling layer will then reduce the dimensionality of the input by reducing spatial size. This helps to extract the more dominant features while decreasing the computational demand that is required to process the values. Max pooling and Average pooling are two methods of pooling. Max pooling selects the maximum value in the area covered by kernel while average pooling takes average value of the whole area that overlaps with filter. The output of the pooling layer is then fed into a fully connected layer, which maps the high-level features extracted by the convolutional and pooling layers to the output class labels or scores. The fully connected layer may also include dropout regularization to prevent overfitting.

Convolution is a mathematical operation done with two functions that generates a new function based on the how these two functions change each other. In CNN, Convolution kernels are one of the two function while image is the other. A convolutional

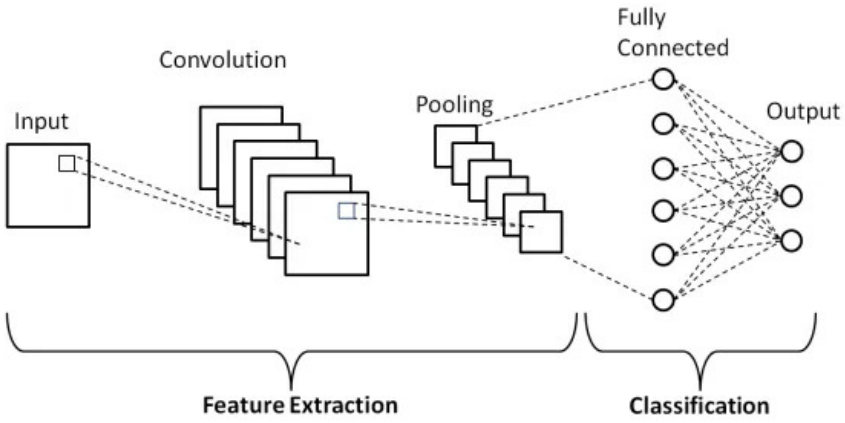

Figure 2. Convolutional Neural Network architecture [PR19]

kernel, also known as a filter, is used in convolutional neural networks (CNNs) to extract relevant features from input data. The kernel is a small matrix of weights that is applied to the input data by sliding over the data and computing the dot product between the kernel and the input patch at each position Figure 3. The resulting feature map summarizes the degree of similarity between the kernel and the input at each position. CNNs typically use multiple kernels, each with different values, to detect a variety of features, such as edges, corners, and blobs.

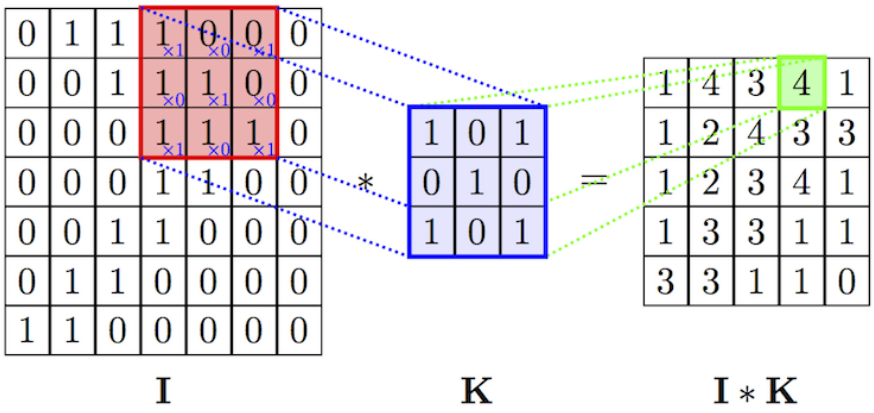

Figure 3. Convolution operation [SM17]

2.2 R-CNN

R-CNN stands for Region-based Convolutional Neural Network. RCNN is a family of object detection algorithms that combines the power of deep convolutional neural networks with region proposal methods. The main idea behind RCNN in region proposals where set of regions created which is possible to have desired objects to be detected in the image. To achieve this several region proposal algorithms can be used such as Selective Search. These algorithms generate set of possible regions based on low-level image features such as color, texture and contrast. After the region proposals are created, each possible region is separately classified using a large CNN. Specifically, the CNN is used to extract a fixed-length feature vector from each region proposal. This feature vector is then fed into a category-specific linear Support Vector Machines, which classifies the region containing each object class of interest. During training, the parameters of the CNN are optimized to minimize a multi-task loss function that includes a classification loss, a bounding box regression loss, and optionally a mask segmentation loss in the case of Mask RCNN. The main advantage of RCNN is its high accuracy in object detection, which has made it a popular choice in many applications, including autonomous driving, surveillance, and robotics. However, RCNN is computationally expensive due to the need to classify a large number of region proposals, which can make it impractical for real-time applications. To address this issue, later variations of RCNN, such as Fast RCNN and Faster RCNN, introduced various optimizations to reduce the number of region proposals and speed up the object detection process. These optimizations include using the convolutional feature maps directly for region proposal generation and sharing computation across region proposals to reduce redundant computation.

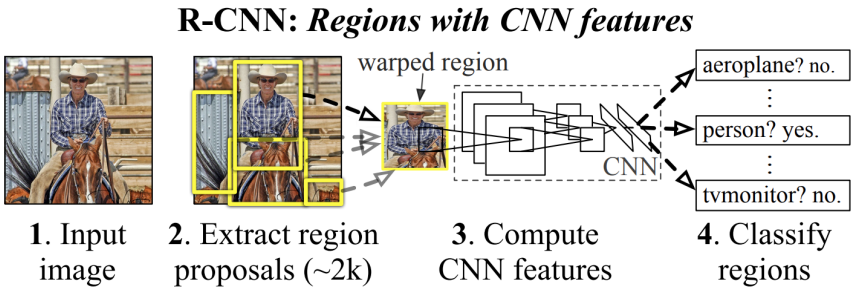

Figure 4. R-CNN architecture [GDDM13].

2.2.1 Selective Search

Selective Search is a region proposal algorithm used in object detection [UvdSGS13]. It calculates the hierarchical grouping of related regions according to color, texture, size and shape suitability. Selective Search uses the graph-based segmentation method proposed by Felzenszwalb and Huttenlocher as its starting point [FH04]. More specifically, it uses the segmentations from oversegmented image as initial input. Then, bounding boxes are extracted from all oversegmentations as region proposals. Obtained region proposals are combined based on similarities. This process is repeated and in each repetition bigger region proposals are added to overall region proposals. This bottom-up approach of generating larger region proposals from the smaller region proposals is what makes Selective Search algorithm hierarchical. This flow can be seen in Figure 5 Similarities between region proposals are defined by following 4 characteristics: color, texture, size, shape compatibility.

Color similarity. Each color channel of the image are used to calculate histogram consisting of 25 bins. Then these bin are merged into an array which results in 75-dimensional color descriptor. To compare color descriptor similarity between each image

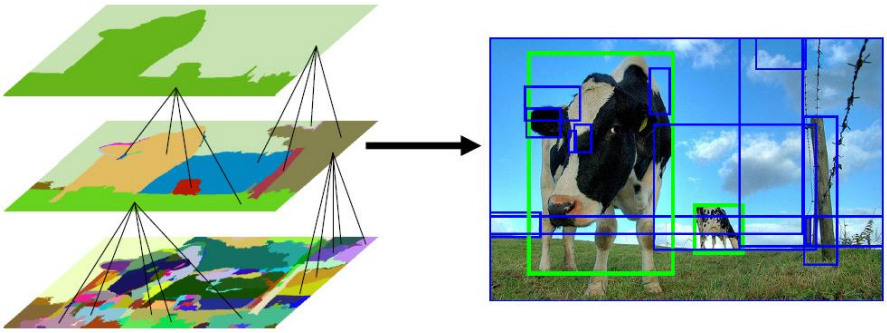

Figure 5. Selective Search segmentation visualization [vdSUGS11].

Equation (1) is used.

$$s_{color}(r_i, r_j) = \sum_{k=1}^{n} min(c_i^k, c_j^k) \tag{1}$$

c_i^k is the histogram value of k^{th} bin in color descriptor.

Texture similarity. To get texture features, the Gaussian derivatives are found at 8 different orientations for each color channel. Subsequently, a 10-bin histogram is computed for each orientation and each color channel, resulting in a feature descriptor that is 240-dimensional (10 bins x 8 orientations x 3 color channels). To compare texture descriptors similarity between each image Equation (2) is used.

$$s_{texture}(r_i, r_j) = \sum_{k=1}^{n} min(t_i^k, t_j^k) \tag{2}$$

t_i^k is the histogram value of k^{th} bin in color descriptor.

Size similarity. Size similarity is needed to make certain that differently scaled region proposals are generated all over the image. It achieves this by combining smaller regions early. If this was not taken into account, one single box would eat all the smaller neighboring regions which would cause differently scaled region proposals to be created only at this spot of the image. Size similarity is calculated using Equation (3).

$$s_{size}(r_i, r_j) = 1 - \frac{size(r_i) + size(r_j)}{size(im)} \tag{3}$$

$size(im)$ is image size in pixels.

Shape compatibility. This defines how well regions r_i and r_j overlap with each other. To eliminate openings if r_j fits into r_i then two regions should be merged. However, if theser regions do not touch each other then regions are not compatible shapes and shouln't be merger. Shape compatibility equation is given in Equation (4).

$$s_{fill}(r_i, r_j) = 1 - \frac{size(BB_{ij} - size(r_i) - size(r_j)}{size(im)} \tag{4}$$

$size(BB_{ij})$ is bounding box around r_i and r_j

.

Final similarity. Sum of all the similarities mentioned above is defined as Final similarity given in Equation (5).

$$s(r_i, r_j) = a_1 s_{color}(r_i, r_j) + a_2 s_{texture}(r_i, r_j) + a_3 s_{size}(r_i, r_j) + a_4 s_{fill}(r_i, r_j) \quad (5)$$

r_i and r_j are the two regions in the image that the similarity is calculated for. $a_i \in 0, 1$ decides if the similarity value is used or not.

2.2.2 Support Vector Machine

Support Vector Machine is a machine learning method that is mainly used to classify both linear and non-linear data [AH+13]. Main working principle of SVMs are to identify a hyperplane that separates two classes of datapoints. To separate two classes there are many possible hyperplanes that can be chosen. SVM tries to find hyperplane that has the maximum distances from the closest datapoints of both classes (Figure 6). Dimension of the hyperplane depends on the dimension of input features. If the input features have the dimension of 2 then the separating hyperplane will be just a line. If the input is 3-dimensional then separating hyperplane will be two dimensional plane. However, it is not always that the data can be separated clearly with a hyperplane. In these cases, SVMs map input data into higher dimensions to try to find seperating hyperplane in higher dimensions. Doing this transformation is computationally expensive and actually converting input dimension to higher dimensions to find separating hyperplane would consume a lot of resources. To avoid this SVMs use what is called kernel trick. Kernel trick implicitly maps input features into higher dimensional space to find linearly seperable hyperplane. Instead of converting coordinates of input features into high dimensional space, kernel trick enables the computation of dot products between the input samples. Kernel trick uses kernel function to calculate scalar value that represents the similarity between two inputs in higher dimension.

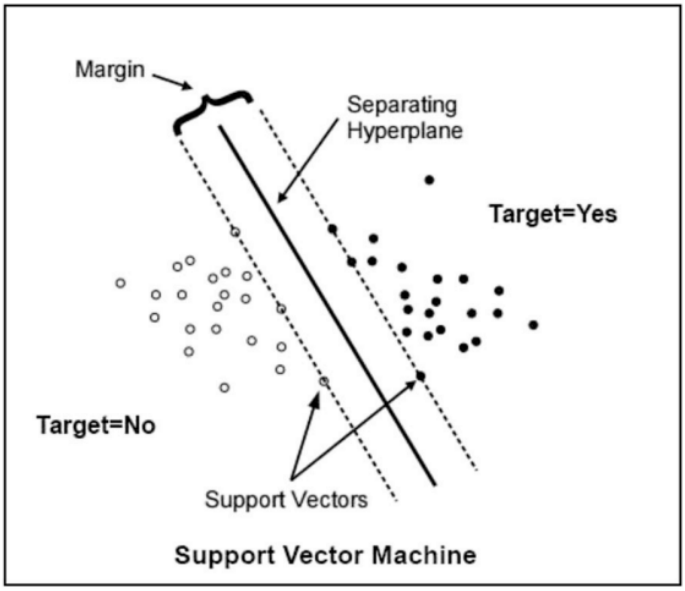

Figure 6. SVM optimal hyperplane [RA19]

2.2.3 Intersection over Union (IoU)

Intersection over Union is an evaluation metric to measure the accuracy of object detection models. It is usually used to evaluate the performance of Convolutional Neural Network models such as RCNN, Faster RCNN, YOLO. To calculate IoU, we need ground truth boxes which are hand labeled for each image that shows the objects we would like to detect. After inference model will output the predicted bounding boxes. To calculate IoU we need to find the intersection rectangle of ground truth boxes and predicted boxes. Then we calculate the area of union by taking the area covered by both boxes. Dividing intersection are to union area will result in the final IoU (Figure 7).

Apart from being used as evaluation metric IoU is also used as a treshhold for finding false positive detections. Generally, 0.5 is set as threshold but depending on the circumstances treshhold can be set as desired. For example, if the IoU threshold is set to 0.5, any predicted bounding box with an IoU value lower than 0.5 is considered a false positive.

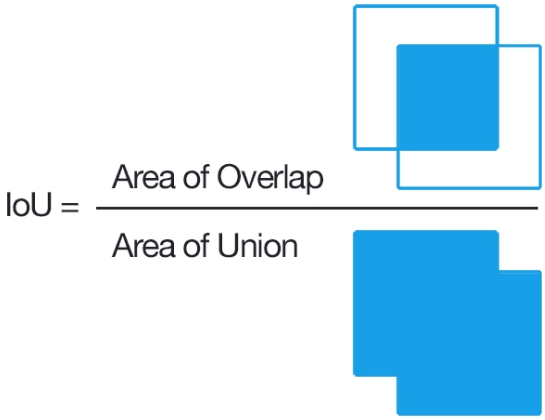

Figure 7. IoU calculation [Adr16]

If IoU value is greater than 0.5 then this prediction would be true positive.

2.3 VGG16

VGG16 is a Convolutional Neural Network architecture that was proposed for ILSVR (ImageNet) competition in 2014 [RDS+15]. VGG16 was trained on ImageNet dataset to detect 1000 classification categories and won the competition in Classification and Localization category. 16 in VGG16's name stands for the number of layers that have trainable parameters. VGG16 has 13 convolutional layers each followed by ReLU activation function, 5 Max Pooling layers and finally 3 fully connected layers last one with followed by softmax to calculate the probability of every class - in total 21 layers which only 16 has learnable weights Figure (8).

As an input image VGG16 takes 224x224 RGB image which has 224x224x3 dimensions. Every convolutional layer uses 3x3 convolution kernel with stride of 1. This consistency of kernel size is also followed for Max Pooling layers, every one of them having the same 2x2 size with 1 stride. Number of filters for every convolutional layers increase as we go down the model architecture. First convolutional layers have 64 filters, second and third have 128 and 256 filters respectively, while fifth and fourth layers have 512 filters each.

In final part of the model, three fully connected (FC) layers follows the last and fifth convolutional layer. First two FCs have 4096 channels while last one has 1000 matching the number of classes. Last but not least output from last FC is followed by softmax layer to compute the probabilities for every class.

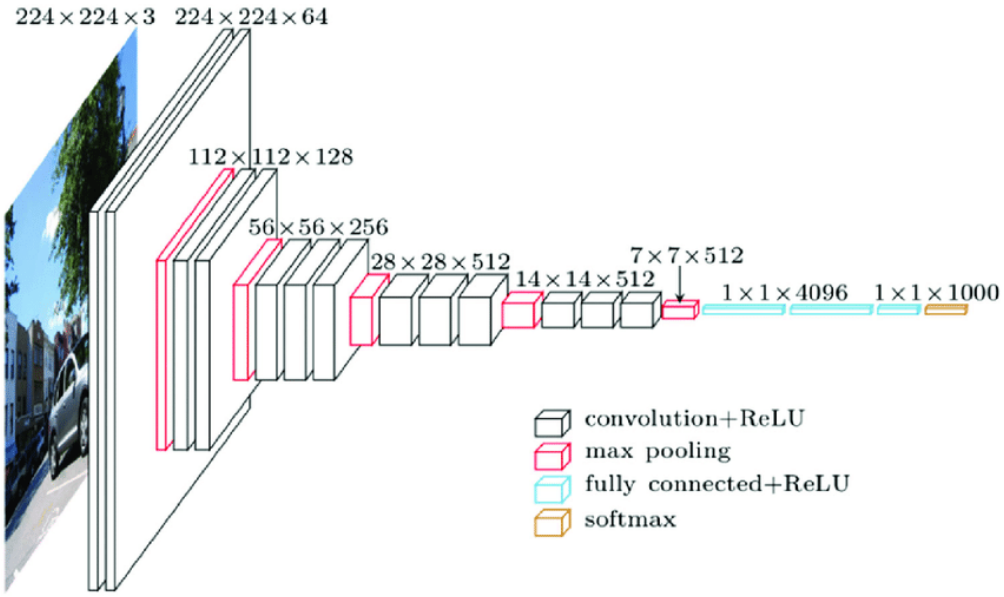

Figure 8. VGG16 Architecture

2.4 Evaluation Metrics

Evaluation metrics are important utility that are used to judge the efficiency and performance of machine learning models. These metrics offer a way to compare various models and assist to measure how well the model is doing. In object detection, evaluation metrics are used to find out how well the model is capable of localizing and detecting the objects in an image or video.

3 Data and Methods

In this section, we will discuss the data that was used to train the models, as well as the specific models and their training setups.

3.1 Dataset

Blueberry plant images provided by Indrek Virro and colleagues from Estonian Life and Sciences University was used to train the models. These images were collected from berry field with the size 28 hectares in Vehendi Village, Elva Municipality, Tartu County. To capture the images of blueberry plants tripod with 360 camera was attached at 1.6 meters height onto the moving platform (Figure 9). In total, there were **280** number of images that contained blueberry plants (Figure 10).

Figure 9. Mobile Platform used for collecting blueberry plant images

To make use of these images to train the models blueberry plants needs to be annotated with bounding boxes manually. To achieve this, we used *labelImg* library from *pip* package manager (Figure 11). We also added directional text to each image to be able

to create region proposals for RCNN from this text. We are going to discuss how these directional texts are used to generate region proposals in Chapter 3.1.

Figure 10. Images from blueberry plant dataset

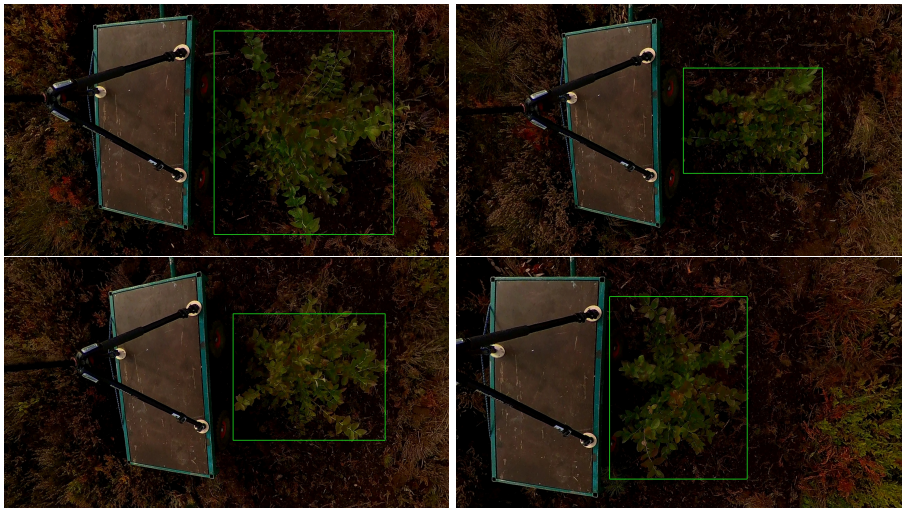

Figure 11. Labeled Images from blueberry plant dataset

3.2 RCNN with Selective Search and VGG16

After building the dataset, we use RCNN architecture and employ transfer learning from VGG16 pre-trained model. We freeze training for all VGG16s layer's weights except second and third layers from last and replace the last layer with one fully connected layer which has 1x1x1 dimensions making the model binary classification indicating our use case of finding only one object - blueberry plant (Figure 12). Last layer of VGG16 is normally a dense layer with 1000 outputs which indicates the original ImageNet dataset classes to be detected by VGG16.

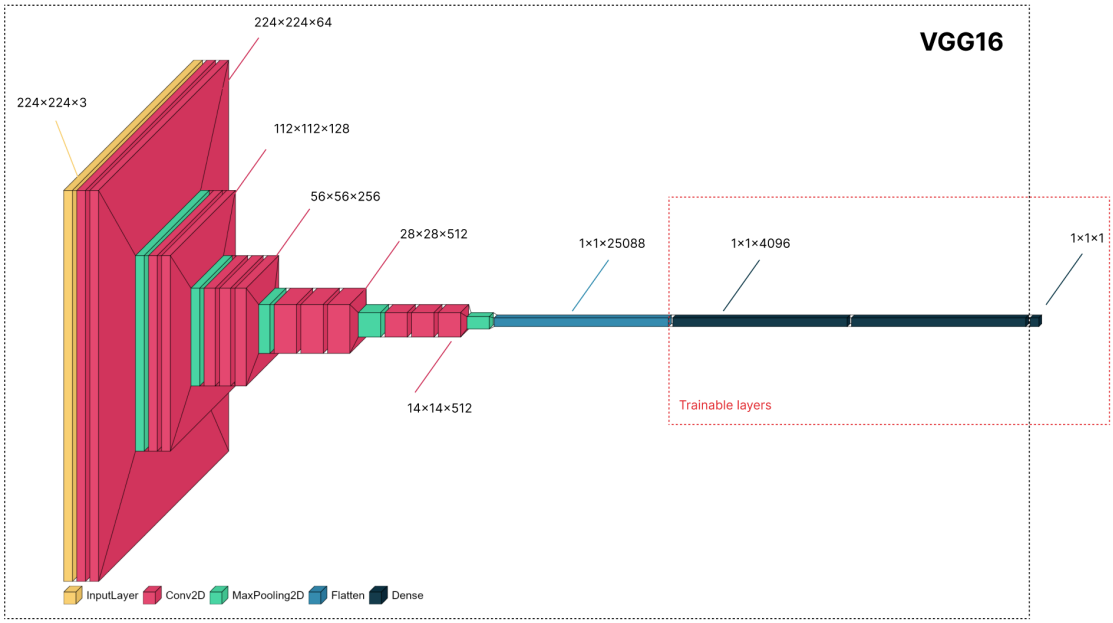

Figure 12. VGG16 Architecture for Transfer learning

VGG16 model for transfer learning (Figure 12) will be trained using the region proposals generated from Selective Search algorithm to fine tune the model to be able to learn features of blueberry plants and generate feature vectors for each region proposals to be used later to classify these region proposals into two classes whether they contain blueberry plant or not.

Training this model will have following steps:

1. The Selective Search algorithm will generate a large number of region proposals for each input image by combining and merging initial region proposals based on their similarity in color, texture, size, and shape. These region proposals are ranked based on their final similarity scores. We are going to use **top 2000 proposed regions** for each image.

2. We are going to build a new dataset for transfer learning by first resizing all the region proposals into 224x224x3 size to be able to pass them down the VGG16 model. Then we are goint to select **30 Positive** and **30 Negative** example regions from 2000 proposed regions for each image. The region is considered Positive example when Intersection over Union of the region with ground truth bounding box is greater than 0.5. Regions having IoU value less than 0.5 with the labeled bounding boxes are considered negative examples. Each positive and negative regions will have corresponding 1 and 0 labels in this new dataset.

Once, VGG16 model modified for transfer learning is trained to convert region proposals into feature vectors, we then use the same model by only changing last layer to optimize one linear SVM as we have only one object to be detected in all images which is blueberry plant. This approach of having one linear SVM for every class instead of using the original classifier is discussed in detail in original RCNN paper [GDDM13]. Main idea is that SVM will perform better because SVM is trained using hard negative mining method while fine-tuned model uses randomly selected negative examples. We are going to follow the same steps as original RCNN paper and train an SVM to detect if proposed regional bounding boxes contain blueberry plants or not. To finetune the SVM we use ground truth bounding boxes as positive examples. As negative examples we take proposal boxes created by Selective Search which have IoU value less than 0.3 with true bounding boxes.

Figure 13. Top 2000 proposed regions by Selective Search algorithm

3.3 Text based region proposals

In this thesis, we propose a novel method to replace the Selective Search algorithm for generating region proposals in object detection tasks. Our approach utilizes text data that was added for each image during the compilation of the blueberry dataset. Specifically, we divide each image into 9 distinct sections, which are annotated by their location in the image as following: **top-left, top-middle, top-right, middle-left, middle, middle-right, bottom-left, bottom-middle, and bottom-right**. These annotations correspond to the approximate location of the blueberry plant in each image. We then use these text annotations to generate a large number of region proposals for each image in the dataset. These proposals are randomly-sized and generated only in the regions of the image where the text annotations indicate the blueberry plant is located. An example of 200 region proposals generated using this approach is shown in Figure 14. The generated region proposals are then used to perform transfer learning with a VGG16 model. The model is fine-tuned using these region proposals, and an SVM classifier is trained on the resulting feature vectors. This process is similar to the one discussed in Section 3.2, with the primary difference being the use of text-based region proposals instead of the Selective Search algorithm.By replacing Selective Search with our text-based region proposal method, we aim to fasten the inference and training of object detection model. Our approach is particularly useful when object location information is available in the

form of text annotations, as it can reduce the number of region proposals generated and improve the overall speed of the model.

Figure 14. 200 proposed regions by Text based region proposal method. Text annotation for this image was **middle**.

4 Experiments and results

In this section we are going to discuss the training setup of RCNN models and performance comparison of Selective Search and Text based region proposal methods.

4.1 Training setup

To perform the transfer learning with VGG16 model we need to get region proposals for every image by running Selective Search algorithm on every training image. High computational complexity of Selective search makes this process very time consuming as it involves computing hierarchical segmentation on an image. On average this process took **63.2 seconds per image**.

Then, using these proposed regions we build training dataset by picking 30 positive and 30 negative examples from top 2000 region proposals of each image decided by the IoU value of region proposals with the ground truth boxes which are above 0.5. This 0.5 are chosen same as the original RCNN paper. Once this data is ready, we then perform transfer learning by training the model discussed in Figure 12. This model was trained for 3 epochs with 64 batch size. Adam optimizer with 0.001 learning rate was used. As loss function the choice was binary crossentropy loss as we are trying only to detect if the proposed region is blueberry plant or not.

After we are done with transfer learning we replace the last binary classification layer of the model with one SVM layer. In the original RCNN paper authors use as many SVM layer as there are objects. As we only want to detect blueberry plants we have one SVM layer to be trained. This SVM layer however are not trained with the same data we have used to train binary classifier. To train this model with SVM layer we are going to take ground truth boxes as positive examples and out of all the region proposals we are going to take 5 boxes which have IoU value less than 0.3 as negative examples.

We train this new model using Hinge loss. As for the choice of optimizer Adam with

0.001 learning rate was utilized. This model was trained for 10 epochs with batch size 32.

We apply same two step training method for Text based region proposals with exact same procedure then to make accuracy and speed comparisons on both models. Building training data for the initial transfer learning step is much faster when using the Text based region proposal method only taking few milliseconds, in contrast to Selective Search algorithm where it was 63.2 s/image.

4.2 Results

After the training of models, both models were tested with test data containing 56 images. For this inference process, each test image needs to go through region proposal generation algorithms, Selective Search in the first model, Text based in the second model. As we discussed, in Section 4.1, selective search algorithm is very slow due to performing hierarchical segmentation, each test image also takes much more time to perform inference with Selective Search. On average, it takes **3.2** minutes per test image for model with Selective Search to be able to predict bounding boxes. However, with Text based region proposals it takes 2.1 minutes per test image for model to detect objects. This faster speed comes with a cost. The Average Precision of the model with Selective Search was **78%** while Text based region proposals method gives us **51%** accuracy. Average precision was calculate by plotting the Precision-Recall (PR) curves and finding the Area under the PR curve (AUCPR) Figures 15, 16. Precision-Recall curve was found by calculating the precision and recall for every value IoU from 0 to 1 increased by 0.1. As we only have one object class which is blueberry plant our AP value is the same as mAP value.

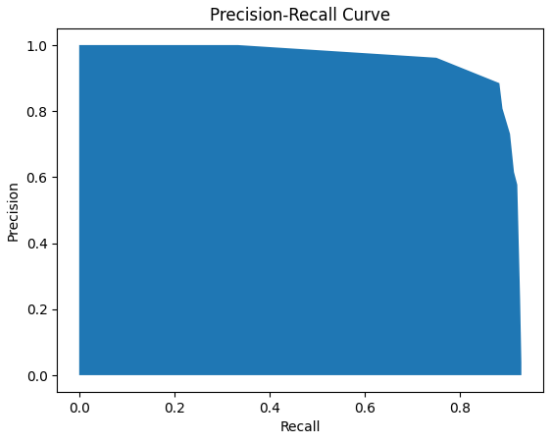

Figure 15. Precision-Recall curve of the model with Selective search

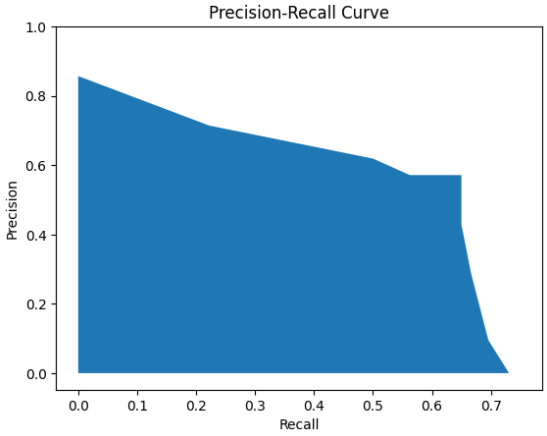

Figure 16. Precision-Recall curve of the model with Text based region proposals

This poorer performance of the model with Text based region proposals method can be explained by several factors. First is the the random nature of the method. As we are generating random bounding boxes around the area based on the text. There is a possibilty that the generated boxes does not have the best IoU with the ground truth boxes. Another reason is the amount of data not being big enough to train the model to an extent that will have similar perfomance of the model with Selective Search algorithm. These results can be interpreted as No Free Lunch theorem for optimization. As, text based region proposals method is faster than Selective Search, this speed boost comes with the compromise of performance.

5 Conclusion

It is evident that neural networks have become an indispensable tool in precision agriculture. Object detection, in particular, has been widely adopted to monitor crops, detection of crop diseases, and for many other use cases. However, integrating text data into object detection poses a significant challenge. In this thesis, we built a blueberry plant image dataset along with the accompanying text for each image. Using this dataset we then implemented the RCNN architecture with pretrained VGG16 model. Two models were built with this architecture only difference being the algorithm that is used to generate regional proposals. First model with Selective Search algorithm which is standard for RCNN architecure, while second model was with new approach that takes into account the text data containing the region in the image where the blueberry can be found and generates region proposal around that region. Upon comparison, it was found that the text-based approach was faster but had poor accuracy, indicating that further optimization and better method of generating region proposals is necessary for the approach to be effective. Nonetheless, this study presents a promising avenue for improving object detection methods in precision agriculture using text data.

6 Acknowledgements

I would like to thank my supervisors Kallol Roy and Indrek Virro who supported and supervised me on this thesis. I would also like to thank the University of Tartu for providing scholarship opportunities such as Dora Plus and Specialization Stipend which enabled me to pursue my studies.

Writing Assistance

ChatGPT is an instance of the GPT-3.5 architecture, a variant of the GPT (Generative Pre-trained Transformer) model developed by OpenAI that is able to process and generate natural language text [Ope]. Natural Language Model ChatGPT's assistance was leveraged in the writing process of this thesis to fix grammar mistakes and rephrase some sentences in academic style.

References

[Adr16] Adrian Rosebrock. Iou visualization, 2016. [Online; accessed May 9, 2023].

[AH+13] Sultan Aljahdali, Syed Naimatullah Hussain, et al. Comparative prediction performance with support vector machine and random forest classification techniques. *International journal of computer applications*, 69(11), 2013.

[AJ22] T. Anandhakrishnan and S.M. Jaisakthi. Deep convolutional neural networks for image based tomato leaf disease detection. *Sustainable Chemistry and Pharmacy*, 30:100793, 2022.

[BRS21] Santi Kumari Behera, Amiya Kumar Rath, and Prabira Kumar Sethy. Fruits yield estimation using faster r-cnn with miou. *Multimedia Tools and Applications*, 80(12):19043–19056, May 2021.

[CPTS08] Flavio Capraro, Hector Patiño, Santiago Tosetti, and Carlos Schugurensky. Neural network-based irrigation control for precision agriculture. 05 2008.

[Fer18] Konstantinos P. Ferentinos. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145:311–318, 2018.

[FH04] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, Sep 2004.

[GDDM13] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[GZJ+21] Zhe Gu, Tingting Zhu, Xiyun Jiao, Junzeng Xu, and Zhiming Qi. Neural network soil moisture model for irrigation scheduling. *Computers and Electronics in Agriculture*, 180:105801, 2021.

[LDN+22] Wei Lu, Rongting Du, Pengshuai Niu, Guangnan Xing, Hui Luo, Yiming Deng, and Lei Shu. Soybean yield preharvest prediction based on bean pods and leaves image recognition using deep learning neural network combined with grnn. *Frontiers in Plant Science*, 12, 2022.

[MAAES+23] Ali Mokhtar, Nadhir Al-Ansari, Wessam El-Ssawy, Renata Graf, Pouya Aghelpour, Hongming He, Salma M. Hafez, and Mohamed Abuarab. Prediction of irrigation water requirements for green beans-based machine learning algorithm models in arid region. *Water Resources Management*, 37(4):1557–1580, Mar 2023.

[Ope] OpenAI. Chatgpt: A large language model for conversational ai. https://openai.com/blog/chatgpt. March 23, 2023 Version.

[OSF+22] Alexander G. Olenskyj, Brent S. Sams, Zhenghao Fei, Vishal Singh, Pranav V. Raja, Gail M. Bornhorst, and J. Mason Earles. End-to-end deep learning for directly estimating grape yield from ground-based imagery. *Computers and Electronics in Agriculture*, 198:107081, 2022.

[PR19] Van Hiep Phung and Eun Joo Rhee. A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. *Applied Sciences*, 9(21), 2019.

[RA19] El-Houssainy A. Rady and Ayman S. Anwar. Prediction of kidney disease stages using data mining algorithms. *Informatics in Medicine Unlocked*, 15:100178, 2019.

[RDS+15] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[SM17] Ihab S. Mohamed. *Detection and Tracking of Pallets using a Laser Rangefinder and Machine Learning Techniques*. PhD thesis, 09 2017.

[SZ15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[UvdSGS13] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, Sep 2013.

[vdSUGS11] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *IEEE International Conference on Computer Vision*, 2011.

[XCZ+23] Laixiang Xu, Bingxu Cao, Fengjie Zhao, Shiyuan Ning, Peng Xu, Wenbo Zhang, and Xiangguan Hou. Wheat leaf disease identification based on deep learning algorithms. *Physiological and Molecular Plant Pathology*, 123:101940, 2023.

Appendix

I. Access to the code

https://github.com/ashrafabbasov/text-based-rcnn

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Ashraf Abbasov**,

*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Text Region-Based Convolutional Neural Network for Precision Agriculture,

   *(*title of thesis)

   supervised by Kallol Roy and Indrek Virro.

   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Ashraf Abbasov

15/05/2023