Nursery AI

Nina Aleskerova [1] **Batyrkhan Gainitdinov** [1] **Artem Galliamov** [1] **Ivan Legenchuk** [1] **Maksim Pankratov** [1]

Abstract

Farmer's work has such an obligatory chore as monitoring the height of each plant to control the harvest. We propose the solution of this problem - measure the heights by Unmanned Aerial Vehicles (UAV) and Deep Learning methods to save human efforts, time, and money. We gathered our own datasets manually by means of UAV, cropped them and created masks for images to use them as additional features for the final classification. We compared the most popular baseline models such as ResNet18, AlexNet and MobileNetV3 Small using several quality metrics and selected the best one in terms of F1-score. After that, we compared the quality of the predictions for two datasets that we have worked with.

Github repo: link to github

Video presentation: first link to the video (Dropbox)

Second link to the video (Youtube)

1. Introduction

Optimization of the farmers work can be reached by replacing the long and unavoidable job with particular techniques which use remote and mobile instruments. Thus, the methods which workers use to measure the height of the plantations are time consuming and bring some limitations. Those are some ground surveys, aerial photogrammetry, which needs to have collection of images, taken from different angles and synthetic aperture radar (SAR), prone to show low resolution in case of unsatisfactory relationship between plants biomass and vegetation properties, also requiring precise installation of sensors to spacecraft. The idea is that the conjunction of developing Deep Learning algorithms with remote and inexpensive surveying instruments will further give much benefits for farmers.

There are several ways to perform remote imagery. One of them is with the help of very high resolution satellites (VHR), those are Worldview-2, GeoEye-1, which is, obviously, has high spatial camera resolution of 1m, but to get such data it might be economically inefficient. Thus to optimize the problem unmanned aerial vehicles (UAV) are used and widespread nowadays, moreover such option of surveying continues to expand, especially for the purposes of the agricultural area.

UAV has high resolution plots as well as VHR, but also the advantage of the method is low cost of used hardware system, which is remotely controlled by the pilot or programmed microprocessor. These customizable and flexible systems perfectly replaces the farmer and ease the process of monitoring the health conditions of the harvest and the way they interact with the environment. Despite the fact, that initial idea of UAV usage was for military purposes, this mobile and safe method was deftly adapted for the agriculture industry, such as vegetation monitoring, cattle detection, inspection of plants deceases and seed and land cover classification. Another advantage of the instrument is providing RGB orthoimages - orthomosaics, which in turn enable to avoid using 3D images of the crops and safe time in terms of computations. Application of UAV in combination with Deep Learning algorithms for image processing can give much benefits and show great results; moreover, Convolutional Neural Networks are able to extract image features and recognize patterns, which is fully fits the solution of agricultural problems using high resolution plots.

The contribution of this study is to propose a method for automatically assessing the growth dynamics of agricultural plants from a given set of images taken from a drone. The obtained results require further improvement; nevertheless, they provide a basis for thinking about new ideas and ways of elaboration of the method.

The main contributions of this report are as follows:

- In Section 2 we provide a comprehensive overview of old, recent and state-of-the art methods for solving the problem of vegetation height monitoring.

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Nina Aleskerova <Nina.Aleskerova@skoltech.ru>, Batyrkhan Gainitdinov <B.Gainitdinov@skoltech.ru>, Artem Galliamov <Artem.Galliamov@skoltech.ru>, Ivan Legenchuk <Ivan.Legenchuk@skoltech.ru>, Maksim Pankratov <maksim.pankratov@skoltech.ru>.

Final Project of the Deep Learning 2021 Course, Skoltech, Moscow, Russian Federation, 2021.

- In Section 3 we explicitly describe data creation and processing, along with algorithms, models, methods and approaches used to solve the problem stated in our project.
- In Section 4 we provide an extensive description of the experiments we have completed.
- In Section 5 we summarize the results that we obtained and discuss our ideas for improving the method.

2. Related Work

Deep Learning models are widely used in the agriculture area, so in the (Santos et al., 2019) the various applications of different Neural Network architectures used for particular purposes are shown, those are seed classification using ResNet-18, Soil/Root segmentation and cattle detection using data, obtained by means of UAV; More than 100 genotypes with highly genetic diversity were used to calculate the biomass of the forage grass for the livestock production with the help of Convolutional Neural Networks for regression task, especially AlexNet and ResNet18 (de Castro et al., 2020). Object detection techniques are used in (Fan et al., 2018) for Tobacco Plant Detection using remote data obtaining with the help of UAV with high spatial resolution of 35 mm.

Moreover such option to obtain data from UAV has expanded in recent years (Bendig et al., 2014), since this way of taking agricultural landscape images, which are orthomosaics with high-resolution, provide photogrammetric analyses with artfully mapped landscapes using inexpensive hardware system. Regarding the paper, authors referred to the height of the plants, but in this case, the height data was gotten using UAV-based multi-temporal crop surface models (CSMs). (de Castro et al., 2020) uses RGB dataset provided with UAV DJI Phantom 4 PRO with the camera resolution of 5472×3648.

The environmental effects on the forage, the crops growth and their health conditions examined during growing season, i.e. leaf nitrogen concentration and plants height parameters were predicted using Machine Learning Algorithms with multispectral imagery of maize plants taken with UAV (Osco et al., 2020).

In the next paper (Radke et al., 2020) the remote imagery for detailed vegetation information, especially the monitoring height of vegetation, was performed by means of using aerial photogrammetry, synthetic aperture radars and light detection and ranging (LiDAR) which gives important insights into forest age and habitat quality. The new architecture Y-net with convolutional layers was designed for these purposes and demonstrated R^2 of 0.83.

In addition, crop plots were processed by applying accurate segmentation with the help of the edge detection, Hough line detection and segment reduction techniques on the plots from UAS (Unmanned Aerial System imagery) in order to assess the state of different varieties and treatment regimes in a timely and cost-effective manner. (Robb et al., 2020)

3. Algorithms and Models

We have successfully completed replication of some scripts from (de Castro et al., 2020) using the unofficial PyTorch implementation [1]. We had to tweak certain parts of the code in order for it to fit the changed setup. The script which was replicated: *cropper.py*, it processes the RGB orthomosaics, Figure 1, obtained with UAV and crops the plants into separate images. This is achieved by blocks, which are defined by user, determining number of rows and columns inside the particular polygons, Figure 2, 3, 4

Figure 1. Orthomosaic

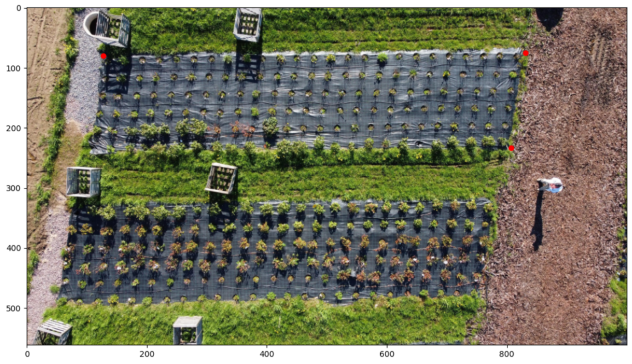

Figure 2. Experimental polygon

[1] https://github.com/wvmcastro/tiffviewer

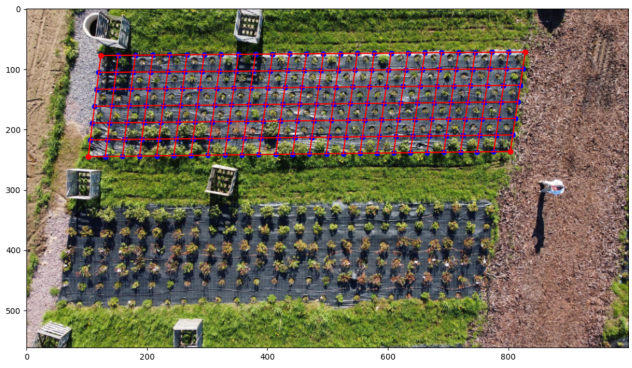

Figure 3. Cropped plots

Figure 4. Separate image

3.1. Dataset

Vegetation images were manually taken at Leningrad Area nursery "Ryzhiki"[2], where the rhododendrons genotype was photographed. Plant beds are presented with the following sizes: 50 x 12, 47 x 12, 30 x 8, and 25 x 6. Each plant within the beds was measured in order to examine the true value of height. Generally, the growth of the vegetation ranges from 0 to 80 centimeters.

As it was mentioned before, the main source for imagery was UAV, especially the model was Mavick mini 2. It provides RGB orthomosaics with high resolution of 4K/30p and 1080/60p video. The footage was taken on 8th of March, i.e. in winter environmental conditions, Table 1 and on 24th. And on 25th of May - summer conditions, Table 2. The provided imagery were get at different day time for the May, thus the shooting conducted in the morning and in the evening with different angle of incidence of sun's rays, resulted in different image conditions. Thus, we replaced standard data augmentation procedures and prevented Deep Learning models from overfitting. Regarding spring imagery, the footage was taken at lunch time.

[2] https://ryzhiki.com/

Table 1. Weather conditions on 8th of March at 2pm

Air temperature, C^o	Air humidity, %	Wind speed, m/s	Soil temperature at the depth of 10 cm, C^o
-7.7	65.5	2.4	-2.1

Table 2. Weather conditions on 24th of May at 10am and 7pm respectively

Air temperature, C^o	Air humidity, %	Wind speed, m/s	Soil temperature at the depth of 10 cm, C^o
13.8	63.3	7.3	9.8
11.9	63.9	3.2	10

The height of UAV flight was 18m with shooting angle equal to 90 degrees.

3.2. Data Preprocessing & Generation

As it is described in the Experiments and Results section, using just images for training neural networks didn't give us such good results as we expected. The reasons for that can be different: the quality of the separate cropped images was not fairly good, the size of the dataset is not large enough for a network with large number of parameters to sufficiently fit to the data, the images diversion was not enough to solve this problem etc. So we had to look for other ways to increase the quality of our predictions. A possible option was to try to increase the images quality by some neural networks, for example, Image Super-Resolution (Sun & Chen, 1907). But this approach did not seem to be as promising because of the nature of images as using semantic segmentation for obtaining more information from the data as input so we decided to learn a segmentation model for further using masks of images on the next stage of the process.

Since all the pretrained models from torchvision library use completely different dataset we had to train our own segmentation model, and for that we needed a marked-up dataset. So we conducted the following process of marking the data:

1. selected a subset of images from the whole dataset which we were going to mark. The criteria was to

take as various ones as possible, naturally considering instances from every target class, including the ones that do not contain a plant at all (having label 0). One can try to measure images' covariance matrices or some other metrics for comparing their variety but the ground pixels often had too bright colors close to the target ones, that is why it seemed to measure ground's diversity rather than plants' so we did not apply it in the final selection. We just took images from different rows and fields with various labels.

2. after that, we created a function using OpenCV tools for making a mask of the image. The idea of it is as follows: we select by hand two representative rectangles from the source image. The first one should contain pixels of the plant on this image and the second one should contain the ground pixels. Then we create two masks corresponding to these rectangles and subtract the second one from the first one. After that, we can apply some morphological transforms to the resulting image: opening, closure and dilate with tuned numbers of iterations. Depending on the shape of the plant on the image, these operations might help to catch it better.

3. we applied this fuction to the chosen subset of images to obtain masks. However, these masks were three-channels images and for our convenience we needed one-channel image. We dealed with this issue in the custom dataset class for segmentation by thresholding the image and making it binary.

4. then we trained segmentation neural network on the obtained data. For this purpose, we chose UNet (Ronneberger et al., 2015) since it is a popular segmentation network and quite easy in implementation. As a loss function to UNet we applied dice loss (Huang et al., 2018). We will tell about the metrics used for training in the following section.

5. and finally, we applied the trained model to all the images in our dataset to obtain the masks. After that, we were able to use it further in our learning process.

3.3. Approach for Dealing with the Artificial Data

Having not only images but also their masks for the classification assumes some freedom in choosing the method of their joint use for training. The simplest solution here is just separately train two networks on these different inputs and then to average their outputs to get the final prediction. We can go further and create a hierarchical neural network that would, for example, cluster images by masks and then classify these clusters individually. But this approach would require the comprehensive study of this area and several experiments with its architecture so we left it as an idea for future development. Instead, we created a new artificial dataset each element of which now consisted of four channels - three channels for image and one additional channel with the binary mask information. Using this data with this additional information we became able to finally train our classification model and to check whether this new feature would increase the quality of our predictions.

3.4. Main Model

As a model for our final classification we considered state-of-the-art architecture - ResNet18. Nowadays Resnet is one of the best architectures and it is applied in various problems including one similar to ours ((de Castro et al., 2020)). After choosing the architecture we need to adapt the exact model to our task.

We set the task as classification task. This is due to the fact that we measure the plants in decimetres and have integer values from 2 to 8 and also 0 (meaning there is no plant on the image at all). This small set of available labels convinced us that it is better to consider the problem as classification one. According to this decision we had to adjust the standard structure of Resnet18 model to our task. We set 8 output channels in the last layer to receive as a result the probabilities of each class (which means height).

In our case we work with the dataset of 4-channel pictures (picture itself and mask). Thus, we had also to adapt the first layer of Resnet18 model as by default it expects standard 3-channel pictures as the input.

3.5. Evaluation Metric

As it was already mentioned, the problem statement implies a regression problem, but since the dataset was created and labeled manually and the target variable was measured with an accuracy of decimeters, we were able to reduce the problem to the multi-class classification. Despite loss in precision in real-life measurements, this approach has several advantages for us. The most important one is that we had the opportunity to assess the quality of our predictions with more intuitive and interesting metrics along with classic ones. We kept in mind that the disctibution of instances along classes was quite unbalanced (for example, 26:36:266:138:2:302:352:50 for winter dataset) while choosing the metric. Our final choice became weighted F1-score since it calculates metrics for each label, and finds their average weighted by support (the number of true instances for each label). Thus, this metric seems to be quite suitable for our task with imbalanced dataset. In addition to this, we also measured usual accuracy and wMAPE (weighted mean absolute percentage error).

4. Experiments and Results

First of all, we conducted a simple experiment with training state-of-the-art models on our dataset without any modifications. This gave us the resulting up to 50-54% accuracy depending on the choice of models and hyperparameters but did not improve further. Moreover, models tended to overfit quite significantly. Therefore, we applied the approach described in the previous section in order to check whether adding new information about the image would boost the performance of the same neural network.

The process of marking up the data is described in the subsection 3.2. It was not such an easy process not only because of limited time and resources but also because of image low resolution and similarity in the color of target plants and the background. Still, we managed to create masks for training dataset that would catch the majority of the object. On the Figures 5 and 6 one can see examples of mask made by us for one object from each winter and summer dataset. In general, we marked up more than 60 images for each of the dataset.

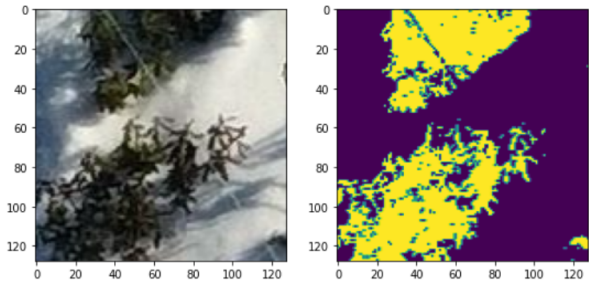

Figure 5. The marking of a random image from the winter dataset

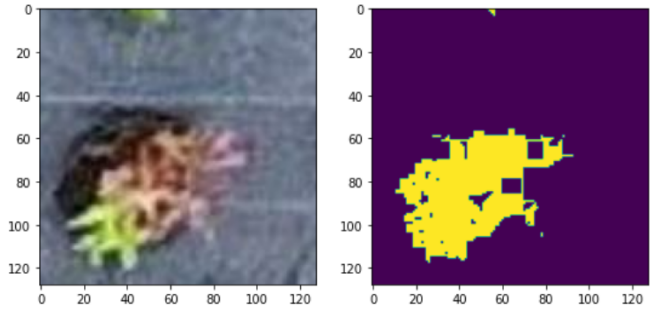

Figure 6. The marking of a random image from the summer dataset

After that, we trained segmentation network UNet on these training images and applied it to all the images in our datasets. On the Figures 7 and 8 one can see examples of networks output for random images from both winter and summer datasets. From the look of it we can suggest that coping with summer dataset was easier for the network, maybe, because of larger difference in pixels colors.

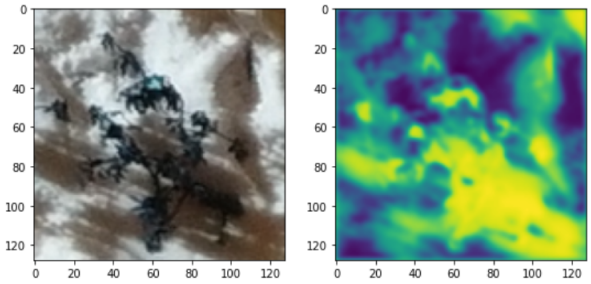

Figure 7. The output of the segmentation network on a random image from the winter dataset

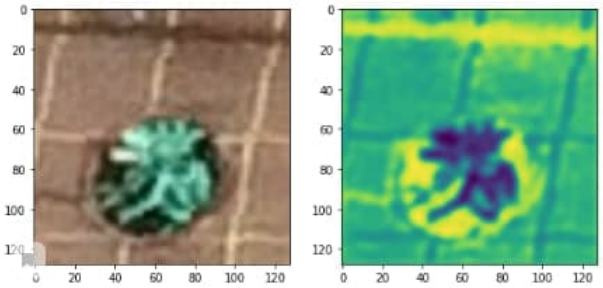

Figure 8. The output of the segmentation network on a random image from the summer dataset

Having masks for all the images, we created new dataset with each instance now presenting four-channel input: first three channels representing color of the original image in RGB format and the last channel being a binary mask for this image.

We trained state-of-the-art models used for our first experiments on this new generated data adapted for 4-channel input. We tried three different neural network architectures: Alexnet, Resnet18 and MobilenetV3 small and also tried using data augmentation and skipping it on the winter dataset since it is smaller and more challenging in terms of masks quality. The results of these experiments are shown in the table 3.

One interesting thing is that adding augmentation almost didn't affect the results measured in accuracy, weighted MAPE and weighted F1-score or even made it worse. But generally, without augmentation a network's quality almost did not change starting from several epochs whereas network that used augmentation was almost monotonously increasing its performance during all training process. Thus, adding augmentation may have still given us more stable result and would result in better quality if using it for larger

dataset and on a larger number of epochs.

ResNet18 has proven to be the best model in terms of both F1-score and accuracy. Therefore, we stuck with it while training the larger summer dataset.

To choose the proper loss, optimizer and scheduler we conducted several test-runs on a small part of the dataset and took those with application of which the model performed better. Thus, we have chosen Adadelta with learning rate = 1.0 as optimizer (was tested against Adam with learning rate = 1e-3), scheduler which multiplies the learning rate by 0.92 each 2 epochs (was tested with several different multipliers and frequency) and binary cross entropy loss (was tested against usual cross entropy).

Training the full pipeline, including UNet for segmentation part, took about 3.5 hours. On the figures 9, 10, 11, 12, 13 one can see how the metrics calculated on the test sample changed during epochs. It is noticeable that all presented test metrics start localizing after 6-7 epochs of training. This, probably, means that we need to increase the resolution of the images to improve the results. On these figures we can also see test weighted F1-score and train accuracy changing over epochs. Surprisingly, we see that train metric also localizes after 6-7 training epochs. Thus, we can conclude that we managed to avoid overfitting. This fact is also proven by the small difference between train and test metrics. For example, train accuracy localizes about 62% while test accuracy stays around 60%.

The comparison of the network's performance on both these datasets is shown in the table 4. As we can see, the network preformed significantly better on the summer dataset. We can interpret it in different ways: maybe this is because the data was made in different conditions and time, ensuring a natural data diversification; maybe this is because the amount of the data d is 33 times larger than for Winter dataset; and maybe (but this is still connected to the first two reasons) this is because of more accurate mask predictions for this dataset.

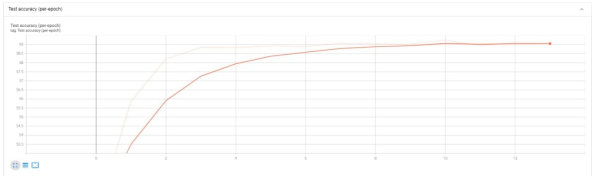

Figure 9. Test accuracy on summer dataset

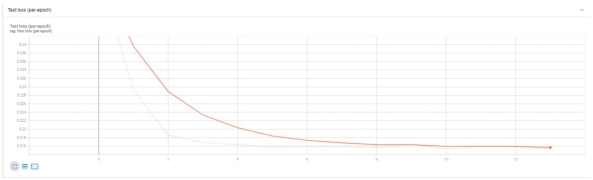

Figure 10. Test loss on summer dataset

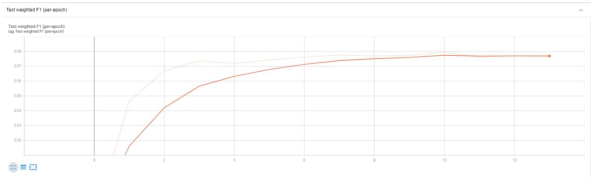

Figure 11. Test F1 score on summer dataset

5. Conclusion & Future Work

Let us first summarize the work that was done and the results we obtained. We cropped the dataset from images that were made by the drone and marked it up with a discrete target variable characterizing the height of the plant, and also marked up the subset of data with masks that highlight the plant on the photo. After that, we trained a segmentation neural network (UNet) on this subset in order to obtain the masks for all images. We used these masks to create a new data uniting image and mask information. After that, we trained ResNet18 on our new data uniting images and masks with Binary Cross-Entropy loss and using weighted F1-score as a quality metric. The results have shown that:

- ResNet18 works best as a baseline state-of-the-art model
- augmentation does not give quality profit in our case though makes the training process more stable
- the segmentation model on a small dataset with low resolution images still can give useful in some sense outcome
- weighted F1-score and accuracy are closely dependent in our case
- there is an increase in quality metrics in case of adding new information in the form of image masks (not reflected here in the tables but basically, the model increased in accuracy from 50-54% to 51-59% on both datasets)

It is worth noting that the whole training process was conducted separately on the two different datasets, summer and

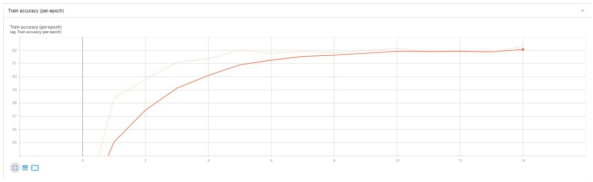

Figure 12. Train accuracy on summer dataset

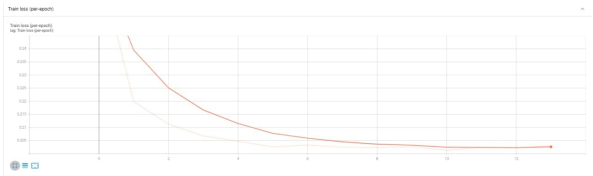

Figure 13. Train loss on summer dataset

winter, since the features characterizing them are too different. The results, due to this fact and to the different size and quality of the datasets, vary notably.

Now, looking at the results, we can suggest what could compromise the work of trained neural networks and lower the quality of the predictions. There are several ways for improving the outcome of our approach. First, looking at the images and comparing them with the ones from the baseline article it is natural to suggest that with higher-quality photos we would almost surely have got better results. Second, and this concerns the winter dataset, it would be better to have a larger size of the dataset and more variety in angle and height, time of day and other factors that affect image illumination. But what can be more important is that for training a segmentation neural network we needed far more marked-up data that we actually used due to the limits of time and resources and the size of the original dataset. Unsurprisingly, the masks obtained by the segmentation network were far from accurate and therefore not as informative as we would like them to be. The biggest problem has become false alarm on shadows which have similar color to the plants. Increasing resolution may help with this problem, which gives another perspective for future improvement of these results.

We can also try using different approaches for combining images and masks as input to neural networks. Using four-channel inputs may be not optimal due to the different nature of these channels, so perhaps, using hierarchical architectures or other neural network architectures combining images and masks as inputs in some way can improve quality of predictions.

Model	Aug.	Loss	Acc.	Mape	F1	Epoch
ResNet	Yes	0.264	50.64	30.67	0.465	13
ResNet	No	0.265	56.17	23.45	0.514	3
AlexNet	Yes	0.306	37.45	35.76	0.257	6
AlexNet	No	0.284	48.94	30.75	0.408	15
MobileNet	Yes	0.319	34.47	44.99	0.278	15
MobileNet	No	0.726	20.0	47.86	0.198	12

Table 3. Comparison of different architectures with and without augmentation for the winter dataset

Dataset	Loss	Acc.	Mape	F1	Epoch
Winter	0.264	50.64	30.67	0.465	13
Summer	0.215	59.26	30.60	0.579	15

Table 4. Comparison performance of Resnet18 on Winter and Summer datasets

References

Bendig, J., Bolten, A., Bennertz, S., Broscheit, J., Eichfuss, S., and Bareth, G. Estimating biomass of barley using crop surface models (csms) derived from uav-based rgb imaging. *Remote. Sens.*, 6:10395–10412, 2014.

de Castro, W. V. M., Junior, J. M., Polidoro, C. H. S., Osco, L., Gonçalves, W. N., Rodrigues, L., Santos, M., Jank, L., Barrios, S., Valle, C., Simeão, R., Carromeu, C., Silveira, E., Jorge, L., and Matsubara, E. Deep learning applied to phenotyping of biomass in forages with uav-based rgb imagery. *Sensors (Basel, Switzerland)*, 20, 2020.

Fan, Z., Lu, J., Gong, M., Xie, H., and Goodman, E. Automatic tobacco plant detection in uav images via deep neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11:876–887, 2018.

Huang, Q., Sun, J., Ding, H., Wang, X., and Wang, G. Robust liver vessel extraction using 3d u-net with variant dice loss function. *Computers in biology and medicine*, 101:153–162, 2018.

Osco, L., Junior, J. M., Ramos, A. P. M., Furuya, D. E. G., Santana, D. C., Teodoro, L. P., Gonçalves, W. N., Baio, F., Pistori, H., Junior, C. A. S., and Teodoro, P. Leaf nitrogen concentration and plant height prediction for maize using uav-based multispectral imagery and machine learning techniques. *Remote. Sens.*, 12:3237, 2020.

Radke, D., Radke, D., and Radke, J. Beyond measurement: Extracting vegetation height from high resolution imagery with deep learning. *Remote. Sens.*, 12:3797, 2020.

Robb, C., Hardy, A., Doonan, J., and Brook, J. Semi-automated field plot segmentation from uas imagery for experimental agriculture. *Frontiers in Plant Science*, 11, 2020.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Santos, L., Santos, F. N., Oliveira, P. M., and Shinde, P. Deep learning applications in agriculture: A short review. In *ROBOT*, 2019.

Sun, W. and Chen, Z. Learned image downscaling for upscaling using content adaptive resampler. arxiv190712904 cs eess [internet]. 2019 [cited 2020 sep 6], 1907.

A. Individual contributions

Explicitly stated contributions of each team member to the final project.

Nina Aleskerova

Made the first experiment with training state-of-the-art models on the original dataset. Made the function for marking up images with masks. Created masks for segmentation part. Wrote UNet training process and the process of obtaining masks from it for all images. Conducted experiments with comparing different architectures and augmentation usage for state-of-the-art models on the winter dataset. Wrote Algorithms and Models (partially), Experiments and Results and Conclusion sections of the report. Created slides for presentation and recorded the video presentation.

Batyrkhan Gainitdinov

Wrote following parts of the report: Introduction, Related works, Dataset, the beginning of Algorithms and Models, Section 3.1. Running, debugging and completing the Cropper.py and Tiffviewer.py from the Github. Cropped the half of the whole dataset. Created masks for segmentation part. Created slides for presentation.

Artem Galliamov

Wrote following parts of the report: Abstract, Introduction, Related works; Running, debugging and completing the Cropper.py and annotate2.py from the Github; Cropped the half of the whole dataset; Created masks for segmentation part; Created slides for presentation; Recorded the video presentation.

Ivan Legenchuk

Created and prepared the datasets, cropped images, made the labeling of both datasets, matched images and labels. Made the corresponding parts in the presentation.

Maksim Pankratov

Writing source code for a custom dataloader. Creating masks for segmentation part. Selecting metrics. Debugging all the parts of code. Writing source code and conducting experiments with ResNet18. Editing the report and creating slides for presentation.