v. 4, n. 5, 2024

Journal of Engineering Research

CLASSIFICATION OF CORN PRODUCTIVITY USING THE FEW-SHOT LEARNING APPROACH

Gabriel Tonon Cimatti
IT Department – Universidade Estadual de Ponta Grossa (UEPG)
Ponta Grossa – PR – Brazil

Alaine Margarete Guimarães
IT Department – Universidade Estadual de Ponta Grossa (UEPG)
Ponta Grossa – PR – Brazil

Eduardo Fávero Caires
Department of Soil Science and Agricultural Engineering – Universidade Estadual de Ponta Grossa (UEPG)
Ponta Grossa – PR – Brazil

Gabriel Passos de Jesus
Post-graduation Program in Applied Computing – Universidade Estadual de Ponta Grossa (UEPG)
Ponta Grossa – PR – Brazil

Abstract: Estimating productivity is important for agriculture, and machine learning (ML) techniques have contributed to making it happen more quickly and efficiently. Considering the difficulty of acquiring agricultural data on a large scale, few-shot learning (FSL) methods are an alternative. The objective was to evaluate the use of different image composition methods obtained by Remotely Piloted Aircraft, associated or not with plant height, for classifying corn productivity, using traditional and FSL-based AM techniques. The results with FSL showed that the Siamese network model can be viable without using the average plant height.

INTRODUCTION

Brazil is the third largest corn producer in the world, behind only the United States and China [FPA 2023]. In agriculture, productivity is the biggest indicator of a good harvest, and having a prior notion of this number, based on the initial state of the crop, can be crucial for the farmer to know if an intervention is necessary in the way of cultivation and in the treatment of crops. plant. However, achieving an approximation of the productivity value at different stages of crop development is complex due to the high cost and long duration of this procedure.

The use of approaches based on Artificial Intelligence (AI) has brought good results for estimating the productivity of different crops when associated with aerial images obtained by different sources, such as satellites and remotely piloted aircraft (RPA), as can be seen in Prestes (2020). Among the AI approaches is machine learning, which learns patterns in historical data from different sources, such as images. Typically, the greater the amount of data, the better the learning performance. However, obtaining data involves time and cost. Thus, techniques have been studied that allow extracting knowledge from a few examples, known as few-shot learning (FSL) [Wang et al. 2020]. Siamese Neural Networks, based on the concept of deep learning, adopt the FLS technique [Chicco 2021]. No article was found that uses the FSL technique to predict grain productivity.

This work aimed to evaluate different methods of composing images obtained by RPA, associated or not with plant height, to classify corn productivity, using traditional techniques and techniques based on FSL.

MATERIAL AND METHODS

The experimental areas used in this study belong to the School Farm of ``Universidade Estadual de Ponta Grossa`` (UEPG), in the Campos Gerais region of Paraná, in the municipality of Ponta Grossa. There are two areas, called "Area 1" (Figure 1) and "Area 2" (Figure 2), in which corn crops were cultivated.

IMAGE CAPTURE AND DATA EXTRACTION

The images were captured through flights carried out using an eBee model fixed-wing RPA (SenseFly), equipped with an RGB camera. The experimental areas were divided into 32 plots, corresponding to different treatments carried out. This way, the plots were demarcated in the images to extract a fraction of the image specific to each plot. In each of the 32 plots, the height of ten plants was collected in the field and the average height of the plants per plot was calculated. (Figure 3). Grain production was obtained by harvesting by plot, weighing the grains and correcting moisture by 13% (standard value). The productivity calculation for each plot considered the final weight of the grains in it (in kg) divided by the area of the plot (in ha).

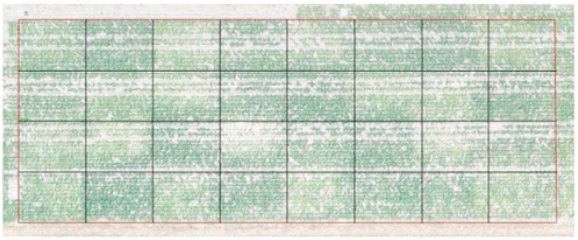

Figure 1. Image obtained from Area 1, with the demarcation of the 32 plots.

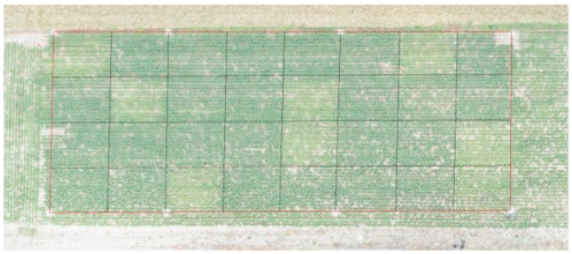

Figure 2. Image obtained from Area 2, with the demarcation of the 32 plots.

Using the Pillow library of the Python programming language [Van Rossum and Drake Jr 1995], the image of each area was cropped, by plot, resulting in 32 new images for each area, all with the same dimensions. After that, two more cuts were made, one where the images were divided into 128 images and the other into 256 images, always of equal sizes. This was done to test whether increasing the database would help with the machine learning task.

CONFIGURING THE IMAGES AND MACHINE LEARNING APPROACH

Using the WEKA software [Frank et al. 2016] and the Auto-WEKA package [Chris Thornton et al. 2013], several tests were carried out, with different ways of image interpretation. The Auto-WEKA package is a tool that seeks to find the best classification algorithm with its best parameters for your database. It is possible to define maximum time and space limits to carry out such a task.

The task defined was classification. Productivity was converted to categorical data of two classes: "low/medium productivity" (< 15,000 kg/ha) and "high productivity" (>= 15,000 kg/ha). The threshold for these classes was defined considering the high productivity rates occurring in the areas, with the aim of identifying classification models for high levels of productivity. For area 1, 16 images were obtained for each class. For area 2, 20 and 12 images were obtained for the low/medium and high productivity classes, respectively.

The cross-validation technique was used for validation, using accuracy as a metric. The separated images were interpreted in three different ways: using a basic image filter that separates them into three histograms (red, green and blue), each with 32 bins; using the average color of the RGB pixels of each image, generating average attributes for red, green and blue, ranging from 0 to 255; using the average pixel color in HSB (hue, saturation and brightness) of each image, resulting in average attributes for hue, saturation and brightness.

The two images (areas 1 and 2) were divided into a base of 128 images and another of 256 images, for each area, totaling 4 different image bases. Each separate image was interpreted in four ways. Tests were carried out with and without height for all variations and, finally, a test with only height as a parameter.

USING THE FEW-SHOT LEARNING TECHNIQUE

The FSL technique was applied to the databases using Python and Siamese Neural Networks. Using the matplotlib library, the RGB values of the images were extracted and, together with the height, three different models were created: one with just the image, one with the image and the height and the last just with the height. This was done for the bases of 128 images and 256, for each of the areas (areas 1 and 2).

For the first model, the input database

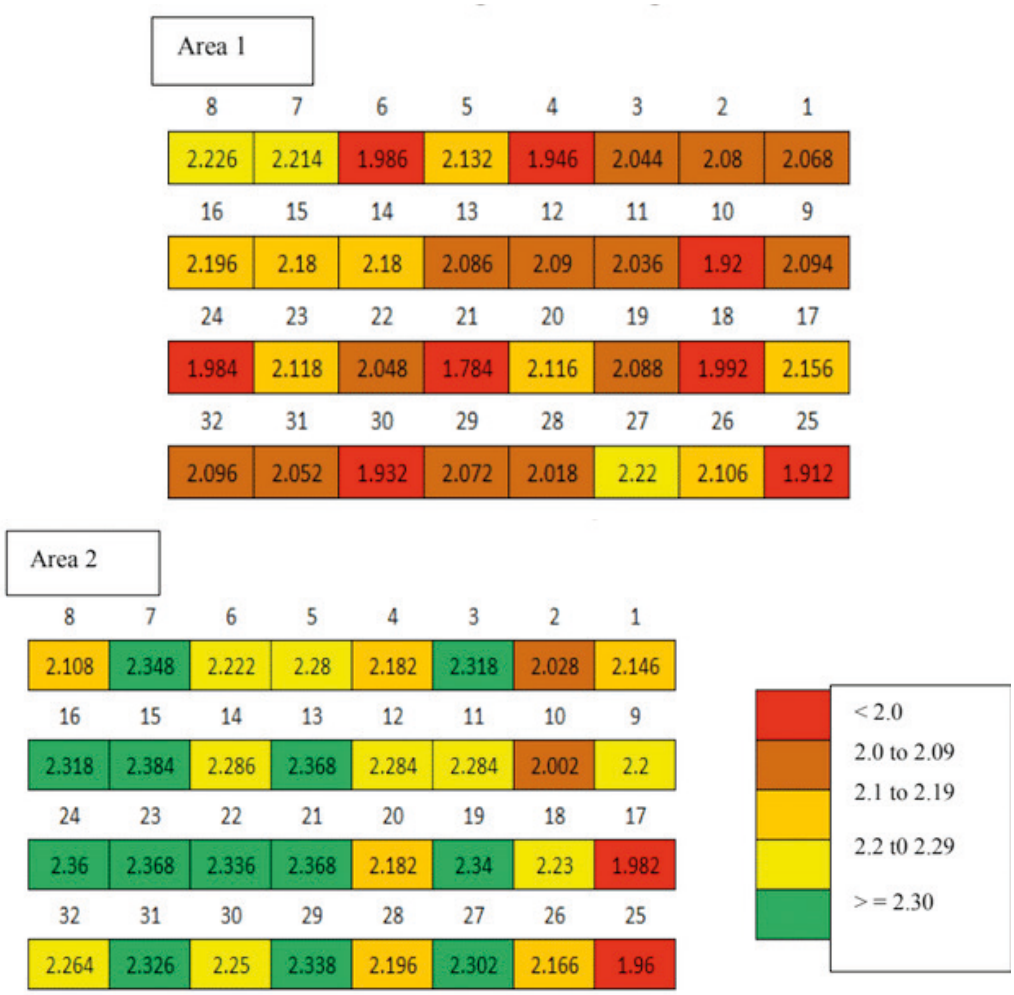

Figure 3. Average height of plants in areas 1 and 2.

was transformed into a multidimensional matrix whose dimensions are "The number of images" by "width in pixels" by "height in pixels" by three RGB values. In the second model, what changes is the last dimension of the matrix, from three entries (RGB) to four (RGB and height). In the third model, the base is a two-dimensional matrix: height by class.

These databases were applied to a Siamese neural network from the Keras library that consists of a combination of "Conv2D", "MaxPooling2D", "Flatten" and "Dense" layers. We adopted hold-out validation with a variable division between training (70-90%) and testing (10-30%) in the models, using accuracy as a metric.

RESULTS AND DISCUSSION

RESULTS WITH DIFFERENT IMAGE SETTINGS AND ALGORITHMS

The results obtained with the different image configurations and algorithms are presented in tables 5 to 8. Accuracy when height was taken into consideration.

It was always above 87% and in several cases above 95%, indicating the importance of plant height. However, it is not always necessary, as in some cases without height the accuracy was greater than 80% (tables 5 and 8). An accuracy lower than 90% is not ideal, but can be considered when data collection is very costly.

SIAMESE NETWORK RESULTS

The results obtained with Siamese networks presented a lower average accuracy than that achieved with the algorithms used in the WEKA software. However, in Table 9 an accuracy of 83% can be seen in the test with height and image using 128 images, which is comparable to those of the WEKA software.

The accuracy obtained with Siamese networks without using the average plant height reached 80%, showing that it is a reliable and easy-to-apply model for large-scale crops. Table 10 also presents a satisfactory accuracy of 72% in the test with only image in the division of 128 images.

CONCLUSIONS

The development of a classification model based on aerial photos and average plant height proved to be effective when used together with the WEKA software and can be applied to any corn crop.

The results obtained using the FSL technique showed that, with the appropriate database and the right configuration, the model with a Siamese network can be applied without using the average height, bringing great advantage to the farmer.

THANKS

This research was supported by the National Council for Scientific and Technological Development.

Area 1: 128	Filter with height	Filter without height	HSB with height	HSB without height	RGB with height	RGB without height	Without image with height
Algorithm	AdaBoost.M1	rules.PART	rules.PART	RandomForest	AdaBoost.M1	RandomForest	AdaBoost.M1
Correctly classified instances	127 99.2188%	104 81.25%	125 97.6563%	94 73.4375%	116 90.625%	104 81.25%	124 96.875%
Incorrectly classified instances	1 0.7813%	24 18.75%	3 2.3438%	34 26.5625%	12 9.375%	24 18.75%	4 3.125%
Kappa coefficient	0.9844	0.625	0.9531	0.4688	0.8125	0.625	0.9375
Square root of mean error	0.0881	0.3719	0.144	0.4127	0.2382	0.3537	0.1329

Table 5. Results from Area 1 with 128 images.

Area 1: 256	Filter with height	Filter without height	HSB with height	HSB without height	RGB with height	RGB without height	Without image with height
Algorithm	trees.REPTree	trees.J48	AdaBoost.M1	lazy.Ibk	AdaBoost.M1	functions.SMO	rules.PART
Correctly classified instances	248 96.875%	206 80.4688%	252 98.4375%	168 65.625%	256 100%	174 67.9688%	248 96.875%
Incorrectly classified instances	8 3.125%	50 19.5313%	4 1.5625%	88 34.375%	0 0%	82 32.0313%	8 3.125%
Kappa coefficient	0.9375	0.6094	0.9688	0.3125	1	0.3594	0.9375
Square root of mean error	0.1581	0.3731	0.1134	0.444	0.0036	0.566	0.1581

Table 6. Results from Area 1 with 256 images.

Area 2: 128	Filter with height	Filter without height	HSB with height	HSB without height	RGB with height	RGB without height	Without image with height
Algorithm	lazy.LWL	rules.OneR	rules.Jrip	rules.OneR	rules.JRip	Multilayer Perceptron	rules.Jrip
Correctly classified instances	121 94.5313%	80 62.5%	112 87.5%	85 66.4063%	112 87.5%	83 64.8438%	112 87.5%
Incorrectly classified instances	7 5.4688%	48 37.5%	16 12.5 %	43 33.5938%	16 12.5%	45 35.1563%	16 12.5%
Kappa coefficient	0.8819	0	0.7333	0.1925	0.7333	0.1304	0.7333
Square root of mean error	0.1789	0.6124	0.3048	0.5796	0.3171	0.4813	0.3171

Table 7. Results from Area 2 with 128 images.

Area 2: 256	Filter with height	Filter without height	HSB with height	HSB without height	RGB with height	RGB without height	Without image with height
Algorithm	lazy.LWL	AdaBoost.M1	RandomForest	lazy.LWL	trees.LMT	rules.Jrip	lazy.LWL
Correctly classified instances	241 94.1406%	221 86.3281%	224 87.5%	165 64.4531%	224 87.5%	203 79.2969%	224 87.5%
Incorrectly classified instances	15 5.8594%	35 13.6719%	32 12.5%	91 35.5469%	32 12.5%	53 20.7031%	32 12.5%
Kappa coefficient	0.8732	0.7194	0.7419	0.0923	0.7333	0.5035	0.7241
Square root of mean error	0.1966	0.2936	0.2616	0.464	0.3096	0.3944	0.2615

Table 8. Results from Area 2 with 256 images.

Area 1: 128	With image, with height	With image	With height	Area 1: 256	With image, with height	With image	With height
Accuracy	Training: 0,9515 Test: 0,8333	Training: 0,8462 Test: 0,8095	0.923	Accuracy	Training: 0,5217 Test: 0,5652	Training: 0,52121 Test: 0,5217	1
Kappa coefficient	-0.08333	0.3575	0.831	Kappa coefficient	0	0	1
Square root of mean error	0.6794	0.5651	0.0769	Square root of mean error	0.7845	0.5651	0

Table 9. Results from Area 1 with 128 and 256 images – Siamese network.

Area 2: 128	With image, with height	With image	With height	Area 2: 256	With image, with height	With image	With height
Accuracy	Training: 1 Test: 0,5833	Training: 0,989 Test: 0,7273	0.9	Accuracy	Training: 1 Test: 0,6087	Training: 1 Test: 0,6522	0.8653
Kappa coefficient	0.1495	-0.1192	0.8	Kappa coefficient	-0.3	-0.3558	0.6956
Square root of mean error	0.7338	0.7071	0	Square root of mean error	0.7845	0.8086	0

Table 10. Results from Area 2 with 128 and 256 images – Siamese Network.

REFERENCES

Chicco, D. (2021). Siamese neural networks: An overview. In *Artificial Neural Networks*, p. 73–94.

FPA. (2023) Brasil é destaque mundial na produção de milho. Disponível em: <https://agencia.fpagropecuaria.org.br/2022/09/30/brasil-e-destaque-mundial-na- producao-de-milho>. Acesso em: 25 abril 2023.

Prestes, C. D. P. (2020). Predição de produtividade de trigo por meio de dados espectrais e altura estimada da planta obtidos por meio de aeronave remotamente pilotada. Dissertação (Mestrado em Computação Aplicada) – Universidade Estadual de Ponta Grossa, Ponta Grossa.

Thornton, Chris et al. (2013). Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 847–855.

Van Rossum, G; Drake Jr, F.L. (1995). Python reference manual, Centrum voor Wiskunde en Informatica, Amsterdam.

Wang, Y., Yao, Q., Kwok, J. T. e Ni, L. M. (2020). Generalizing from a few examples: A survey on few-shot learning. ACM computing surveys, ACM New York, USA, p. 1–34.

Frank, E., Hall, M. A. e Witten, I. H. (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.