XXVIII Congresso {virtual} de Iniciação Científica da Unicamp 2020

IDENTIFYING MATURE AND IMMATURE TOMATOES IN A GREENHOUSE WITH DEEP LEARNING

Yuri F. Borrmann, Matheus Ferraciolli, Guilherme Folego & Luiz Henrique A. Rodrigues

1 INTRODUCTION

Researches with tomatoes images are frequently related to maturity and fruit size estimation, because it helps to plan the harvest and to project sales. For example, Sari et al. (2016) and Sari & Adinugroho (2017) used different approaches to classify the maturity of segregated tomatoes. This means that they used images containing only tomatoes in a simple background. Other researches that worked with isolated tomatoes include developing algorithms to estimate their volume and mass (Nyalala et al., 2019), detecting external defects (da Costa et al., 2020) and detecting fruits, flowers and the maturity gradient (de Luna et al., 2019).

In a real world application, it is necessary to monitor tomatoes in crop conditions. Deep learning techniques such as convolutional neural networks (CNN) have shown promising results to identify objects in images (LeCun et al., 2015). For example, Liu et al. (2019) achieved an F1-Score of 92.15% while detecting mature tomatoes in these condition. Meantime, the identification of green fruits in the background of leaves could be a challenge: Habaragamuwa et al. (2018), working with green strawberries identification, reached an average precision of approximately $10\,p.p.$ lower when compared with results of mature strawberries.

In this research, we propose a deep learning model that identifies images which contain tomato fruits. In addition to the challenge previously mentioned, another obstacle is that the CNN needs a large amount of data to train. Transfer learning techniques can be used to solve this problem, which means to adapt the knowledge of pre-trained models to a new problem (Pan & Yang, 2010). Additionally, data augmentation techniques have proven to increase significantly the model performance in agriculture problems (Bargoti & Underwood, 2017).

2 MATERIALS AND METHODOLOGY

2.1 DATASET

All images were collected at the producer Projeto Mais[1], located in Campinas, SP, Brazil. We took the photos with a Samsung Galaxy J8 camera ($4\,608 \times 3\,456$ pixels resolution). Due to the complexity of CNNs, images with high resolution require high computational processing. So it is common to split images into smaller patches to train models. Folego et al. (2016) indicate that using patches is beneficial to avoid information loss and to better use the details of images.

Using this method, we split each image into patches with sizes similar to the input of most neural networks (*i.e.*, 224×224 pixels). One original image generates 225 patches with resolution of 307×230 pixels. All patches have the same size and there is no overlap between them.

We split 52 images in high resolution into $11\,700$ patches. Then these patches were labelled in two categories: "with tomato" and "without tomato". All images with any fraction (the size does not matter) of a tomato were considered "with tomato" samples. Images that did not have any tomatoes parts were classified as "without tomato" samples. Some patches were removed because

Table 1: Images distribution and their patches division.

Set	Images	Patches	With tomato	Without tomato
Training	37	8 361	2 739	5 622
Validation	5	1 121	336	7 85
Test	10	2 115	584	1 531

it was impossible to distinguish if we were observing a leaf or a green tomato. In these cases, the challenge of classifying green tomatoes in the middle of green leaves stands out.

Our dataset was divided randomly trying to follow the proportion: 70% of patches to train the model; 10% to validation; 20% to test (Table 1). During this process, we assured that all patches of the same original image belonged to the same set (training, validation or test).

2.2 Training and validation

As parameter of analysis, we used the same metric as Habaragamuwa et al. (2018), Bargoti & Underwood (2017), Dias et al. (2018) and Liu et al. (2019): F1-Score. Also, to improve the study of the model performance, we generated the receiver operating characteristic (ROC) curve and we calculated the area under the curve (AUC).

As shown in the literature, transfer learning is a technique that may improve or not results, depending on the dataset. Thus first we validated the use of three training methods: from scratch, with fine-tuning, and with fine-tuning just in the last layer. Due to the better results on previous experiments and other similar problems (Rahnemoonfar & Sheppard, 2017; Habaragamuwa et al., 2018), we used in these first experiments the neural network ResNet152 pre-trained in the ImageNet (Russakovsky et al., 2015) dataset.

With the best precedent method we validated techniques of data augmentation, which increased results of previous works (Bargoti & Underwood, 2017; Perez & Wang, 2017). The first arrangement is flipping images in vertical and horizontal, both with 50% probability of being flipped. The second arrangement is a combination of the first one with transformations in bright, contrast, saturation and hue. The third is the same as the second but with random rotation. Finally, we compared resizing patches directly to the input size of ResNet152 or applying a random crop in each patch.

Then, we trained the following models: AlexNet (Krizhevsky, 2014), ResNet18, ResNet50 and ResNet152 (He et al., 2016), VGG16 and VGG19 (Simonyan & Zisserman, 2015); Inception V3 (Szegedy et al., 2016), SqueezeNet V1.0 and SqueezeNet V1.1 (Iandola et al., 2016). And to interpret the classification produced by our neural networks, we used the class activation mapping (CAM) visualization algorithm (Zhou et al., 2016). Lastly, to solve the confusion problem of a green tomato fruit in the green background, we followed Habaragamuwa et al. (2018) recommendation to use a CNN as a feature extractor and then train a support vector machine (SVM) classifier (Cortes & Vapnik, 1995) with these features.

3 Results and discussion

The results of different learning methods are shown in the Table 2. As expected in cases that the amount of data are limited, fine-tuning the network shows better results Tajbakhsh et al. (2016). In our problem, fine-tuning all the network is the best solution. In Table 3 we present the results of different data augmentation techniques. Then, we also validate applying a random crop in our patches instead of resizing it. The results are in Table 4.

Based on the last experiments, we defined that our models should be trained fine-tuning all the network, flipping images in vertical and horizontal, applying transformations in brightness, contrast, saturation and hue, and using random crop. The results of each neural network are presented in the Table 5. They are compatible with other papers in similar problems (Rahnemoonfar & Sheppard, 2017; Habaragamuwa et al., 2018), where ResNet152 shows the best F1-Score.

[1] `https://www.casabugre.com.br/empresa/o-projeto-mais`

Table 2: Results of ResNet152 with different learning methods.

Learning Method	F1-Score
From scratch	0.5911
With fine-tuning just in the last layer	0.7742
With fine-tuning in all the network	**0.8947**

Table 3: ResNet152 training results with different data augmentation.

Data augmentation	F1-Score
Without data augmentation	0.8847
Flipping in vertical and horizontal	0.8935
Flipping and transformations in bright, contrast, saturation and hue	**0.8947**
Flipping, transformations and rotations	0.8934

The best F1-Scores were from ResNet and VGG families. In opposition, the Inception V3 is akin to these neural networks (lots of layers and parameters) and has demonstrated the worst results in recall and precision. A deeper analysis is necessary to understand the reasons for this results.

We have as example of a ResNet152 false negative case the Figure 1, that shows the activation mapping of our CNN. It is considered that red pixels means higher activations. In this example, even the model having activations in the region of the tomato, the classification was "*without tomato*". The same was repeated in others false negatives cases.

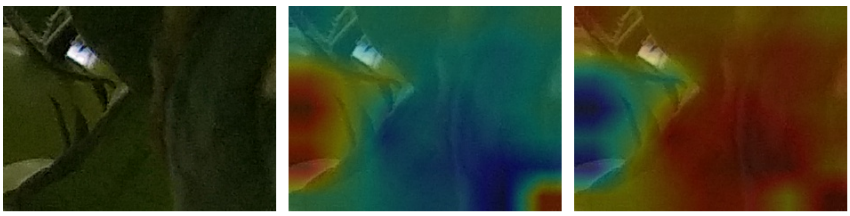

Figure 1: Class mapping activation example for tomato (in the middle) and without tomato (in the right).

With weights updated previously, we used the ResNet152 to extract features and train the SVM classifier. Using the RBF kernel with parameters $C = 1$ and $\gamma = 1/n_{features}$, the F1-Score was 0.9527. Using the linear kernel with parameters $C = 1$, the F1-Score was 0.9454.

The classification results of SVM models were higher to those of neural networks (Table 6). The error of the best SVM model in the test set was 30 images with tomato and 25 without tomato. From the 30 tomato images, 23 of them were green tomatoes with overlapping green leaves. This emphasizes the difficulty of classification in this challenging scenario.

4 Conclusion

In this research we developed an automated system based in deep learning techniques to identify images that contains tomato fruits. We validated the use of different CNNs, as well as techniques of data augmentation and transfer learning.

During the construction of the dataset, we observed the difficulty of annotating images with green tomatoes in the middle of green leaves. This recognition is an obstacle even to a human, so it was expected that the model classification in these cases would result in more errors.

The best model has an F1-Score of 0.9527, using the ResNet152 (with weights updated to this problem) to extract features and an SVM to classify them. This result shows that our method has better performance to classify images when compared to previous works in similar problems.

Table 4: ResNet152 training results with two specific data augmentation techniques.

Data augmentation	F1-Score	AUC
Resizing the patch to the same model input size	0.8808	**0.9854**
Random cropping the patch	**0.8947**	0.9243

Table 5: Results of different neural networks.

Neural Network	Recall	Precision	F1-Score	AUC	Inference (ms)
AlexNet	0.7722	**0.9415**	0.8485	0.9421	7.08
Inception V3	0.7587	0.8878	0.8181	0.9734	18.25
ResNet18	0.8185	0.9336	0.8723	**0.9786**	8.11
ResNet50	0.8527	0.9222	0.8861	0.9187	14.69
ResNet152	**0.8801**	0.9097	**0.8947**	0.9243	31.18
SqueezeNet V1.0	0.7962	0.9012	0.8455	0.9625	6.64
SqueezeNet V1.1	0.8253	0.8976	0.8599	0.9671	**6.21**
VGG16	0.8425	0.9283	0.8883	0.9050	23.90
VGG19	0.8596	0.9211	0.8893	0.9138	28.07

Table 6: Results using the SVM classifier with RBF and linear kernels.

SVM	Precision	Recall	F1-Score
RBF kernel	**0.9568**	**0.9486**	**0.9527**
Linear kernel	0.9422	**0.9486**	0.9454

Considering the publication of all code used in this research, the results can be reproduced and the models used in subsequent researches. To future works, it is necessary to build a larger dataset to reach even better results. The model developed in this research can be used as ground to solve problems of localization and segmentation.

REFERENCES

Suchet Bargoti and James Underwood. Deep fruit detection in orchards. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3626–3633. IEEE, May 2017.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep. 1995.

Arthur Z. da Costa, Hugo E.H. Figueroa, and Juliana A. Fracarolli. Computer vision based detection of external defects on tomatoes using deep learning. *Biosystems Engineering*, 190:131–144, 2020. ISSN 15375110.

Robert G. de Luna, Elmer P. Dadios, Argel A. Bandala, and Ryan Rhay P. Vicerra. Size classification of tomato fruit using thresholding, machine learning and deep learning techniques. *Agrivita*, 41 (3):586–596, Oct. 2019. ISSN 24778516.

Philipe A Dias, Amy Tabb, and Henry Medeiros. Apple flower detection using deep convolutional networks. *Computers in Industry*, 99:17–28, Aug. 2018.

Guilherme Folego, Otavio Gomes, and Anderson Rocha. From impressionism to expressionism: Automatically identifying van Gogh's paintings. In *2016 IEEE International Conference on Image Processing (ICIP)*, volume 2016-Augus, pp. 141–145. IEEE, Sep. 2016.

Harshana Habaragamuwa, Yuichi Ogawa, Tetsuhito Suzuki, Tomoo Shiigi, Masanori Ono, and Naoshi Kondo. Detecting greenhouse strawberries (mature and immature), using deep convolutional neural network. *Engineering in Agriculture, Environment and Food*, 11(3):127–138, Jul. 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:770–778, Dec. 2016. ISSN 10636919.

Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. pp. 1–13, Feb. 2016.

Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. Apr. 2014.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

Guoxu Liu, Shuyi Mao, and Jae Ho Kim. A mature-tomato detection algorithm using machine learning and color analysis. *Sensors (Switzerland)*, 19(9):1–19, 2019. ISSN 14248220.

Innocent Nyalala, Cedric Okinda, Luke Nyalala, Nelson Makange, Qi Chao, Liu Chao, Khurram Yousaf, and Kunjie Chen. Tomato volume and mass estimation using computer vision and machine learning algorithms : Cherry tomato model. *Journal of Food Engineering*, 263(July):288–298, 2019. ISSN 0260-8774.

Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct. 2010.

Luis Perez and Jason Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. Dec. 2017.

Maryam Rahnemoonfar and Clay Sheppard. Deep Count: Fruit Counting Based on Deep Simulated Learning. *Sensors*, 17(4):905, Apr. 2017.

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*, 115(3):211–252, Dec. 2015.

Yuita Arum Sari and Sigit Adinugroho. Tomato ripeness clustering using 6-means algorithm based on v-channel otsu segmentation. In *2017 5th International Symposium on Computational and Business Intelligence (ISCBI)*, pp. 32–36. IEEE, Aug. 2017.

Yuita Arum Sari, Sigit Adinugroho, R. V.Hari Ginardi, and Nanik Suciati. Enhancing tomato clustering evaluation using color correction with improved linear regression in preprocessing phase. *2016 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2016*, pp. 401–406, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–14, Sep. 2015.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-December:2818–2826, Dec. 2016. ISSN 10636919.

Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, May 2016.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2921–2929. IEEE, Jun. 2016.