Tomato detection with computer vision

Nikola Marić
University of Ljubljana, Faculty of electrical engineering
Tržaška cesta 25, 1000 Ljubljana
nm9529@student.uni-lj.si

Abstract

First and possibly most important step in implementing computer vision in agriculture is detection. In this paper we explore and test the performance of the U-net architecture in a segmentation task that is applied on images of tomatoes in complicated environment conditions with illumination variation, branch and leaf occlusion, tomato overlap, tomatoes in different stages of ripening and in a variety of sizes. All of these represent challenges in tomato detection. We rely heavily on data augmentation and explore the U-nets ability to significantly improve results trough data augmentation. With the use of data augmentation we improve the average IoU result from 82,6 % to 87,62 %. We also compare the U-net model to algorithms created based on SIFT, HOG and Cov descriptors in a classification task based on their ability to extract and utilise the correct features in such a challenging environment.

1. Introduction

First and possibly most important step in implementing computer vision in agriculture is detection. If we are not able confidently separate our objects of interest from the unnecessary background, then all further functions, such as disease detection, following the development of plants, automating collection using robots and so on, cannot be implemented. In this paper we will be taking a look at methods for detecting tomatoes.

In recent years convolutional neural networks (CNNs) were applied to the problem of image segmentation, specifically the so called U-net architecture [9] was introduced. One of the issues in using a CNN is the requirement for a large database. Sometimes millions of images are needed. This is a big issue for agriculture applications because they usually only have a couple of hundred, to a couple of thousand images to work with. One of the main advantages of the U-net architecture, that we will be utilising, is that it can be trained on relatively small data sets. We will also be taking a look at how data augmentation influences the final result and if it is a good tool for enlarging data sets for use in combination with the U-net architecture. Apart from image segmentation with a CNN, we will be taking a look at how algorithms using SIFT [7], HOG [3] and Cov [12] descriptors for binary classification perform. The task for the algorithms is to decide weather a given picture has a tomato in it or not.

This paper is organised as follows. In section 2 we talk about where this paper fits in and what research it is related to. In section 3 we explain the methods and data sets that we will be using. In section 4 we go trough the experiments and analyse the results. In section 5 we discuss and compare the results.

2. Related work

Various computer vision approaches have been deployed to tackle the issue of fruit detection. Initial methods that were introduced [13][1] focused on using colour and distinctive specular reflection patterns and other data gained with special cameras to detect fruit. Sa et. al. [11] proposed using a Faster Region-Based CNN for fruit detection using imagery obtained from two modalities: colour (RGB) and Near-Infrared (NIR). However their method encountered limitations with different fruit sizes, with not being able to properly detect some smaller fruit. Specifically for tomato detection, an algorithm using the Histograms of Oriented Gradients (HOG) descriptor and a Support Vector Machine (SVM) classifier was introduced by Liu et. al. [5] for the use in harvesting robots. The method provided good results with a relatively low execution time (0.95s per image). The problem with this method is that the images used were ether of a tomato or some type of background (leafs, twigs and others), meaning that the algorithm would only differentiate if the given image was of a tomato or background. Also, all of the tomatoes were ripe. The method wasn't tested with tomatoes in different stages of development. Our paper is most similar to Liu et. al. [6] who introduced an algorithm called YOLO-Tomato that is based on YOLOv3 object detection method [8], claiming above 90% accuracy in different conditions using the Intersection-over-

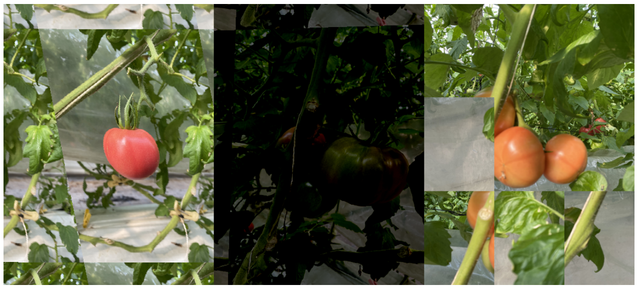


Figure 1. An example of three images that were heavily augmented

Union (IoU) evaluation metric. The data set used consists of 966 images of tomatoes in different growing circumstances.

3. Methods

In this section we will be going trough methods and the data set that we will be using for our experiments. We will be conducting two different experiments using different methods. First we will be doing image segmentation using the U-net architecture [10]. Afterwards we will be doing tomato classification, similar to [5] using SIFT, HoG and Cov descriptors.

3.1. The base data set

The base data set that we will be using is created on the basis of the LaboroTomato [4] data set. It consists of three types of images:

- RGB images of tomatoes taken in a greenhouse at different stages of ripening. The picture were rotated and resized to 512x384 pixels. There are 804 pictures in total, out of which 643 were used for training and 161 for testing.
- masks, which are black and white images where tomatoes are white and everything else is black.
- ground truth images where we have a comparison of every RGB image with its corresponding mask and bounding boxes drawn for each tomato.

3.2. Data augmentation

To enrich the training data set, we performed a series of augmentations to the original images and their corresponding masks. For augmentations we used the Albumentations [2] library for python. We performed eight different augmentations in total: HorizontalFlip, VerticalFlip, CLAHE, ElasticTransform, RandomBrightnessContrast, RandomGamma, RandomGridShuffle, RandomSizedCrop. The final data set consists of 7716 images. All of the images retained their size (512x386). You can see an example in figure 1.

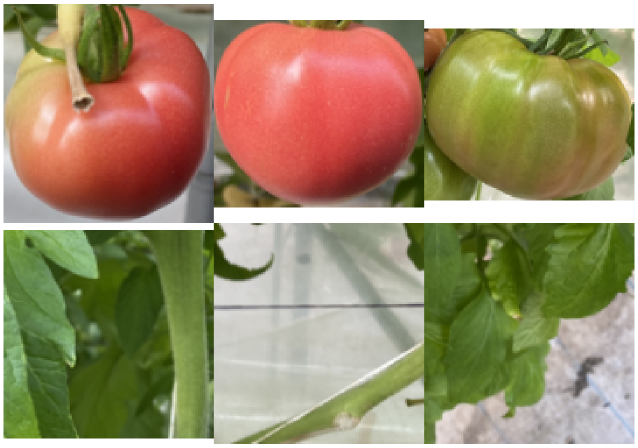


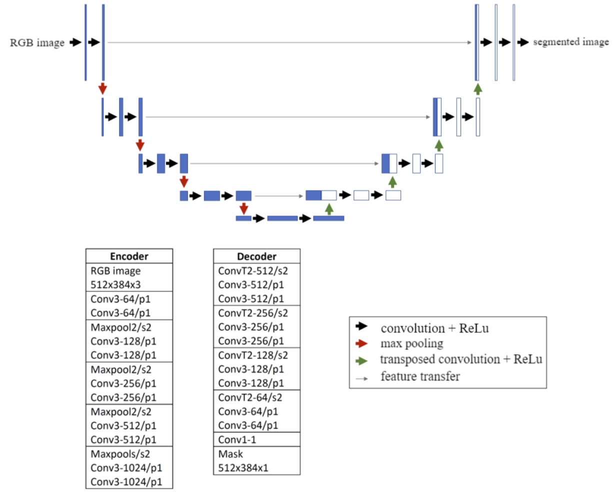
Figure 2. An example of images used for the classification task

For tomato classification using SIFT, HOG and Cov descriptors we had to split the original images into images of tomatoes and those of the background. This was done by cropping out every tomato into its own image by using the bounding boxes provided in the original data set. The background images were chosen randomly from the remaining background in the size that ranged from the smallest to the largest bounding box and in the same number as tomatoes. The final data set consisted of 9643 images of tomatoes and 9643 background images. You can see an example in figure 2.

3.3. Image segmentation using U-net

The U-net architecture was introduced by Ronnenberger et. al. [10] for image segmentation in the field of Biomedicine. It was since used in many different applications because of its ability to provide good results when trained on relatively small data sets, that have somewhere from a couple of hundred, to a couple of thousand images. The network architecture is also well suited for the use of data augmentation and is fast, being able to do a segmentation of a 512x512 image in less than a second on a NVidia Titan GPU [10]. It is a CNN in the shape of the letter ”U” 3 consisted of an encoder that has convolutional and max pooling layers, and a decoder that has transposed convolutional and max pooling layers. For the activation function, ReLu was used. A very important part of this architecture is the copying of features, from the encoder side to the decoder that is done at every level separately.

As input, the network takes a RGB image of tomatoes and as output it is supposed to give a binary image with all of the tomatoes colored white. For parameter optimisation the Adam method was used. As this is a binary segmentation, we will be using the binary cross entropy loss function. This function first applies the sigmoid function to the output of the network and then compares it to the image mask. The error is calculated for every pixel using the following



Encoder
RGB image 512x384x3
Conv3-64/p1 Conv3-64/p1
Maxpool2/s2 Conv3-128/p1 Conv3-128/p1
Maxpool2/s2 Conv3-256/p1 Conv3-256/p1
Maxpool2/s2 Conv3-512/p1 Conv3-512/p1
Maxpools/s2 Conv3-1024/p1 Conv3-1024/p1

Decoder
ConvT2-512/s2 Conv3-512/p1 Conv3-512/p1
ConvT2-256/s2 Conv3-256/p1 Conv3-256/p1
ConvT2-128/s2 Conv3-128/p1 Conv3-128/p1
ConvT2-64/s2 Conv3-64/p1 Conv3-64/p1
Conv1-1
Mask 512x384x1

Figure 3. U-net architecture that was used for tomato segmentation

formula:

$$\frac{1}{N}\sum_{i=1}^{N} -(y_i * \log{(p_i)} + (1-y_i) * \log{(1-p_i)})$$

where y_i is the correct label of the pixel i that we can take from the mask and p_i is the probability that the pixel belongs to the object that we are trying to segment.

For the evaluation we will be calculation the deviation of the predicted masks when compared to the ground truth masks. For that purpose we will be using the Jaccard index, also known as the Intersection over Union (IoU). We use this index because we are not interested in how similar our output image is to the mask, but how much of our object of interest was correctly identified and how much did we miss/falsely identify. This is exactly what the IoU metric provides. If we were to simply compare the two images, a part of them would always mach, even if no tomatoes were detected. IoU is calculated with the following formula:

$$IoU = \frac{TP}{TP+FP+FN}$$

where TP represents the number of tomato pixels in the output image that mach the tomato pixels of the ground truth. FP represents the number of pixels that were identified as a tomato, but belong to the background and FN represents the number of pixels that belong to a tomato, but were identified as the background.

3.4. Tomato classification using SIFT, HOG and Cov descriptors

We created three separate classification algorithms, each using one of the mentioned descriptors, for a two class classification issue. One class represents tomatoes and the other represents the background. We used the nearest neighbour classification method for all three algorithms in such a way that we compare the descriptors of the current image with the descriptors form the training set. This comparison is done by calculating the distance between each descriptor and the entire training set. Distance calculation is different for each descriptor. The descriptor gets assigned the class form its nearest descriptor in the training set and the entire image gets the class that the majority of its descriptors have. For instance, if 60% of descriptors of an image are classified as a tomato, the entire image is classified as a tomato.

For the SIFT descriptor approach, we first clustered the descriptors into 20 clusters for tomato images and 20 clusters for the background images. For each image we first extract the SIFT descriptors then we calculate the euclidean distance between each descriptor and the 40 clusters assigning each descriptor to its closest cluster. As the first 20 clusters represent class one and the other 20 class two. We classify the image by counting how many descriptors are in each group of clusters, the one with more gets the image.

For the HOG approach we calculate the euclidean distance between our descriptor and all of the descriptors form the training set, for each class. The image is classified by calculating the sum of all distances.

Cov approach is the same as HoG, except for the distance metric. In this case we didn't use the euclidean distance but a custom metric depicted by the following formula:

$$D = \sqrt{\sum_{i=1}^{N}((\log{(\lambda(C_i - C))})^2)}$$

where C_i represents covariance descriptors form the training set, C represents the current covariance descriptor and N is the number of covariance descriptors in the training set.

For evaluation of all three approaches the True positive rate (TPR) and False positive rate (FPR) metrics were used. They are calculated with the following formulas:

$$TPR = \frac{TP}{TP+FN}$$

$$FPR = \frac{FP}{FP+TN}$$

where TP represents the number of tomatoes that were classified as tomatoes, FN represents the number of tomatoes that were classified as the background, FP represents the number of background images that were classified as tomatoes and TN represents the number of background images that were classified as background.

4. Experiments and results

In this section we will be going trough the experiments that we have conducted. Firs we will be taking a look at tomato segmentation with the U-net model and after that,

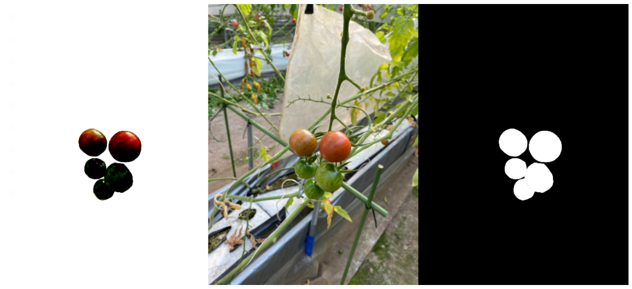


Figure 4. An example of successful segmentation

we will take a look at the classification problem with different descriptors.

4.1. Segmentation with U-net

Here we conducted two separate experiments. The first was training the U-net model on the base data set with 643 training images and 161 test images. We trained the model for 100 epochs with the batch size set to 2. The model converged relatively quickly in the 27th epoch with the IoU score of 82,6%.

This result was plagued by two issues. First was that some images have tomatoes in the background that are not marked on the corresponding masks, also some masks take the branches into account and some not. The second issue is the relatively small data set. To deal with this issue we decided to utilise the ability of the U-net to learn from augmented data.

For the second experiment we used the augmented data to enrich the training data set and trained the model on 7716 images. It was trained for 100 epochs with the batch size of 10. The model converged in the 72th epoch with the IoU score of 87,62 %. This is a considerable improvement when compared to the base data set. On figure 4 we can see an example of a successful segmentation.

With the use of augmented data we have successfully dealt with the issue of the data set size. The only problem that remains are the masks. As we can see in figure 6, there are some tomatoes that are covered by a branch, the mask that was used as the ground truth has the entire tomato marked including the branch, while our model tries to exclude the branch as it is not a tomato. The second issue that we noticed, figure 5, is that there are some background tomatoes that are not included in the masks but our model detects them. In most cases where the IoU is lower, the reason is the masks that were used for evaluation and not the model itself. Here we can actually see that our model has successfully learned the features that represent a tomato, regardless of its stage of development, size, lighting, orientation, shape, distance from the camera and so on.

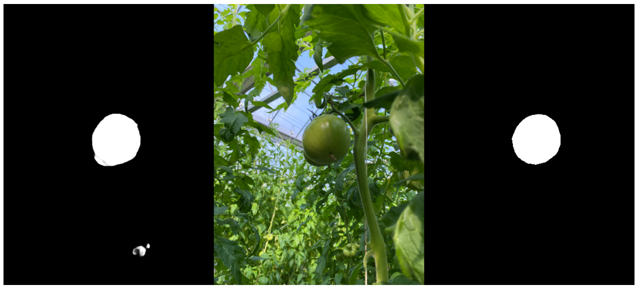


Figure 5. An example of a tomato that was detected, but wasn't a part of the mask

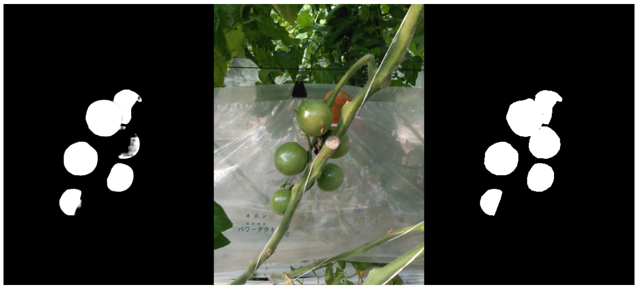


Figure 6. An example where the mask takes the branch into account

4.2. Classification using SIFT, HOG and Cov descriptors

The algorithms were trained on 15376 images and tested on 3910 images. Half of the images were of tomatoes and the other half of the background. The training and testing was done in Matlab 2022b.

The algorithm based on SIFT descriptors achieved a TPR of 61,18 % and a FPR of 32,08 %. Some images are without descriptors and those image were disregarded.

Using the HOG descriptor provided considerably better results, with TPR of 87,32 % and FPR of 15,64 %. Although the results are considerably better than with the SIFT descriptor, they are still unsatisfactory and pale in comparison with the U-net. This will be discussed further in the Discussion section.

The algorithm based on Cov descriptor performed similarly to the SIFT based algorithm, with a TPR of 76,15 % and FPR of 54,75 %. Here we have to take into account that, when compared to the SIFT based algorithm, this one is really lightweight, meaning it requires considerably less computational power.

5. Discussion

For the issue of tomato segmentation, the U-net architecture provides excellent results. The model that we have created is capable of detecting and separating tomatoes from the background in their natural environment, independent of size, lighting, orientation, shape, distance from the camera and the stage of ripening of the tomato. This makes it extremely usable as a first step in automating greenhouses. After detecting and separating tomatoes, the resulting images can be used in a series of different algorithms that would focus on tomatoes themselves, without the background interfering.

When comparing the U-net model to algorithms that use ”hand-made” descriptors for extracting features, the U-net completely eclipses them in performance. Even the best results using the HOG descriptor would result in a TPR of 87,32 %, which leaves a lot of tomatoes wrongly classified, meaning that the features extracted were not good enough to successfully differentiate all tomatoes in all cases form the background. On the other hand the U-net model detects tomatoes without issue. We presume that, if we created a similar algorithm that uses the U-net model on the images from the data set used by the HOG based algorithm, we could classify them on the basis of the number of ”white” pixels and the TPR would be very close to 100 %. This of course has to be tested.

For future work, we can focus on improving the mask used for training and testing the U-net model. This way we can get more concrete evaluation of the model that would represent the actual success of the segmentation.

References

[1] C. Bac, J. Hemming, and E. Van Henten. Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper. *Computers and electronics in agriculture*, 96:148–162, 2013.

[2] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, volume 1, pages 886–893. Ieee, 2005.

[4] Laboro.AI. LaboroTomato. `https://github.com/laboroai/LaboroTomato`. Accessed: 2023-03-22.

[5] G. Liu, S. Mao, and J. H. Kim. A mature-tomato detection algorithm using machine learning and color analysis. *Sensors*, 19(9):2023, 2019.

[6] G. Liu, J. C. Nouaze, P. L. Touko Mbouembe, and J. H. Kim. Yolo-tomato: A robust algorithm for tomato detection based on yolov3. *Sensors*, 20(7):2145, 2020.

[7] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[8] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[9] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[10] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[11] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool. Deepfruits: A fruit detection system using deep neural networks. *Sensors*, 16(8), 2016.

[12] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II 9*, pages 589–600. Springer, 2006.

[13] Q. Wang, S. Nuske, M. Bergerman, and S. Singh. Automated crop yield estimation for apple orchards. In *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, pages 745–758. Springer, 2013.