BIO Web of Conferences **130**, 01016 (2024) https://doi.org/10.1051/bioconf/202413001016
BFT-2024

Information system for monitoring number of fruits and their condition using computer vision technology

A.V. Shimokhin[1*], *A.A.* Luchinovich[1], *S.N.* Boltovsky[1], and *A.S.* Gritsay[2]

[1]Omsk State Agrarian University named after P.A. Stolypin, 1, Institutskaya ploschad, Omsk, 644008, Russian Federation
[2]Omsk State Technical University (OmSTU), Omsk, Russian Federation

Abstract. The main objectives of process automation are to improve the quality and productivity of process execution. An automated process has more stable characteristics than a manually executed process. One of the tools of automation is artificial intelligence in general and computer vision in particular. Computer vision (also called machine vision) is a scientific trend in the field of artificial intelligence and related technologies for acquiring images of real-world objects, processing them and using the obtained data to solve various kinds of applied tasks without (full or partial) human participation. From an engineering perspective, computer vision seeks to automate tasks that can be performed by the human visual system. Computer vision tasks include ways of acquiring, processing, analysing, and understanding digital images and extracting data from the real world to produce numerical or symbolic information. In the context of crop farming, automation is primarily capable of increasing yields and improving the quality of the final product, by better controlling environmental conditions and timely detection of any abnormalities in the development and growth of plants and fruits, by increasing productivity, by replacing a variety of more accurate and responsive automated systems (e.g. irrigation or temperature control) with human involvement, and thereby simplifying the role of humans in cultivation, and by improving the quality of the final product. In the paper, the main objective of the designed information system is the recognition in a digital image of plant fruits with subsequent classification based on known colour ranges in advance. In order to achieve this goal, the following tasks should be solved: minimisation of possible negative effects of distortions of the original image; analysing the image to look for tomato fruits on it.

1 Introduction

Computer vision technology in the context of fruit growing allows analysing the condition (e.g., growth stage) of tomato fruits and keeping records of them.

Classification of tomato fruits into ripening stages is possible based on their external colour. The main stages of tomato fruit ripening are as follows:

* Corresponding author: av.shimokhin@omgau.org

https://doi.org/10.1051/bioconf/202413001016

– milk stage;
– brown stage;
– pink stage;
– red stage [1].

At the milk stage of maturity, the fruits have a hard rind, light green with whitish tinge of surface colour, light green pulp with the beginning of ossification around the seeds.

Fruits in this stage are well stored and finally ripen in 2-3 weeks [1].

The next stage of ripening is brown (another name is blanche). At this stage, the tomato fruit begins to acquire the inherent colour of the variety and almost reaches its optimum size, but is still hard and firm to the touch. Tomatoes at this stage of development can be stored, but, unlike milk ripeness, they need much less time to fully ripen - usually 6 to 7 days [1].

Fruits at the pink stage are dense, with light pink or bright orange colouring. Up to 50% of the fruit surface may be yellowish-brown in colour.

The final stage of tomato development is the red ripening stage. Fruits at this stage are dense, of full biological maturity and of a colour typical of the botanical variety. Fruits of this stage have a short storage period (about a few days) and should be consumed or preserved as soon as possible. A distinctive feature of the ripe tomato is the characteristic shine of the peel [1].

Since different varieties often have different external colours at all stages of ripening, it is necessary to determine the characteristic colours for each individual variety for a correct classification.

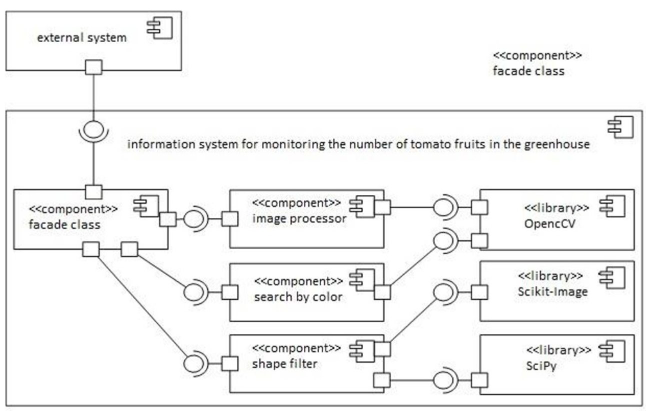

Fig. 1. Diagram of components of the designed information system

The diagram (Figure 1) demonstrates that the developed information system is intended to be used as a plug-in library to some external system.

The external system can be any software or hardware-software complex based on a platform supporting the Python interpreter (for example, a smartphone or a single-board computer with a video camera). This external system provides the information system under development with the images required for tomato fruit recognition and uses the result depending on the task at hand. The developed information system also provides for the possibility of further processing and analysis of the obtained results (information about the size and position of tomato fruits in the image) by the external system (for example, the use of artificial intelligence to improve the accuracy of detection of tomato fruits or to expand the possibilities of recognition in difficult conditions).

The developed algorithm can be divided into several main stages.

The first stage is the preparatory stage. It receives the source image on which tomato fruits should be recognised, colour ranges, within which the colours of the searched tomato fruits are supposed to be found, and user parameters of the external geometrical shape of the searched tomato fruits. If the colour ranges and geometric shape parameters have not been communicated, the provided default ranges will be used later in the colour search step and the external shape filtering step. Also at this stage, parameters responsible for switching on/off the following functions of the developed system are received from the external calling system:

– brightness levelling;
– colour balance adjustment;
– object separation.

The second stage is image preprocessing and in turn consists of two stages:

– minimising negative effects that can be caused by different lighting conditions,
– minimising negative effects associated with general image noise.

The third stage is the stage of image analysis in order to search for tomato fruits on the image. This stage is divided into three stages:

– colour search stage;
– binarised image processing stage;
– stage of filtering objects by external form.

Colour search stage. The principle of actions occurring at this stage consists in consecutive checking of all pixels of the image for the occurrence of their colour in the range or ranges obtained at the first stage. The result of this stage is a binarised image in which white colour indicates the positions of pixels satisfying the above conditions, all other space is filled with black colour. Each colour range transmitted in the first stage corresponds to its own binarised image. This processing serves to filter small objects that possess the desired colours and have a small size (of the order of 30x30 pixels). These small objects are most likely not the sought tomato fruits. The next stage is used for filtering larger objects.

The stage of filtering objects by external shape. - The stage of searching for objects in the image with shape similar to tomato fruit and filtering all other objects. Object shape analysis is based on the following main parameters:

– square;
– roundness;
– convexity;
– inertia coefficient.

The result of the image analysis stage is a list with information about the position and size of the found objects and their binarised images for each separate colour range. As a result of the developed information system operation, the list of found objects is formed in accordance with the initial colour ranges and returned to the external system for further processing.

Before analysing and searching for objects directly, it is necessary to prepare the original image. This is to minimise the effects of distortion and enhance certain features of objects to transform the image to a form suitable for further analysis. Examples of such operations include exposure correction, colour balancing, image noise reduction, sharpening and image shape correction [2-4].

This stage in the developed information system has the following algorithm presented in Figure 2.

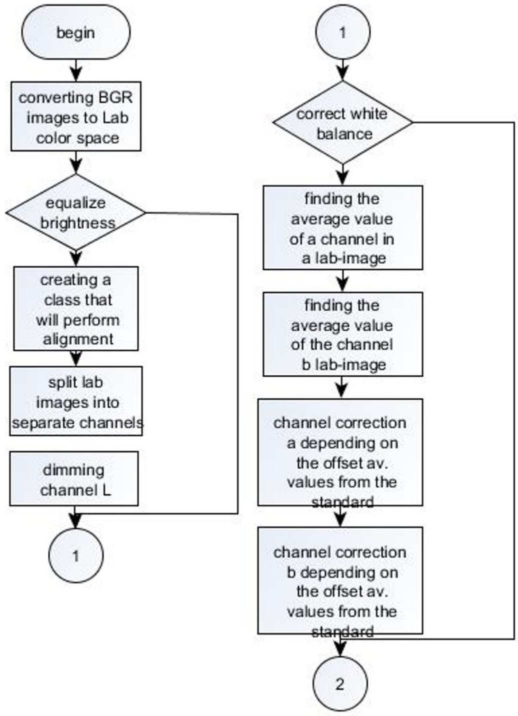

Fig. 2. Image preprocessing algorithm

The most significant problems that may prevent further recognition can be divided into two categories:

– lighting problems;

– problems related to noise and distortion of individual pixels in the image.

The main problem of recognition is related to lighting. This is different colour balance in different source images. This problem may result from different lighting conditions of the scene at the time of capturing the object and may lead to the fact that the same object (or similar objects) in different photos under different lighting conditions or photo equipment settings will have different colours, which prevents further analysis of the image.

The white balance of an image depends mainly on the colour temperature of the scene lighting. Colour temperature (spectrophotometric or colorimetric temperature; denoted Tc and measured in kelvins) is defined, according to Planck's formula, as the temperature of a completely black body at which it emits radiation of the same colour tone as the radiation under consideration.

To solve the above problem, the developed information system uses colour balance adjustment. An important purpose of this adjustment is to correctly display certain colours, especially neutral colours. Hence, the general method is sometimes referred to as grey balance, neutral balance or white balance. Colour balance changes the overall mixture of colours in an image and is used for colour correction. Generalised versions of colour balance are used to correct colours or to deliberately alter them for artistic effect.

The results of the described method are demonstrated below in Figure 3 on the example of two identical photos with different white balance parameters.

Fig. 3. Adjusting colour balance in an image with predominantly warm tones (original image on the left, balanced colour and brightness on the right)

On this example it is possible to see that on initial photos the colour of the same object, but with different white balance parameters, differs, that can prevent further image analysis and search of tomato fruits on it by their characteristic colour.

It should be noted that most cameras already have a colour balance correction mechanism implemented. For this reason, the described information system provides possibility to switch off colour balancing in case colour balancing is performed directly when capturing an object. This action is performed in the same way as switching off light equalisation.

2 Materials and Methods

After image pre-processing the main stage of this information system follows. It directly searches for objects similar to tomato fruits. This stage can be divided into the following stages demonstrated in Figure 4.

The first two stages were partially covered when describing the general scheme of the algorithm (search by colour and filtering of small objects). It is worth paying attention to the last three stages presented in Figure 5 above.

The stage of filling the black inner regions of objects in the binarised image is a preparation for the subsequent separation of contiguous objects. The appearance of unwanted bores on objects is most often caused by non-ideal lighting conditions in the process of capturing the object. And the necessity of this stage is justified by the specificity of work on separation of objects merged in a binarised image.

The stage of separation of adjoining objects is necessary for solving the corresponding problem. It consists in the fact that two objects having close colours and touching each other, form one large common object in the binarised image, which leads to errors in recognition. The final stage of recognition is to search the resulting binarised image for objects with a shape similar to a tomato fruit.

Colour search serves as the basis for tomato fruit detection in the image in the developed information system. The principle of its operation consists in sequential checking of all pixels of the image for entering the specified colour range.

Removal of background and leaves in the developed system is based on exclusion of pixels of certain colours from the image. Namely the pixels, which in the Lab colour range have b* channel value less than 138 and in the RGB colour range the red component of the final colour less than 72.

Colour search in this information system is carried out using image in HSV colour model.

In case of HSV colour coordinates are:

– hue;
– saturation;
– value.

https://doi.org/10.1051/bioconf/202413001016

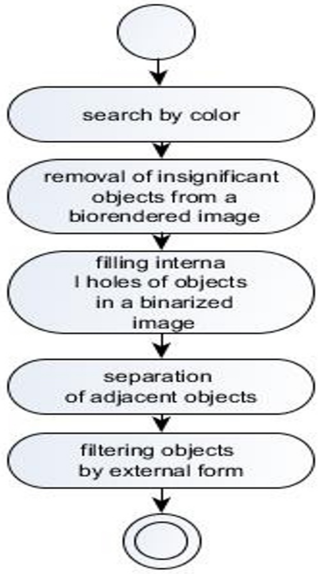

Fig. 4. Diagram with stages of tomato fruit retrieval in an image

In contrast, the HSV model allows to manipulate brightness and saturation directly without affecting tone, which makes it easier to work with colour. In the context of colour retrieval, this allows for more accurate selection of colour ranges. The result of this step is a binarised image with selected found objects, where white indicates pixels whose colour falls within the range being searched. An example is shown in Figure 5.

a)  b)  c) 

Fig. 5. Result of the colour search stage: a) original image, b) binarised image, c) combined demonstration image

As you can see from the example in the previous figure, colour selection does not work perfectly. Some points can be neglected (e.g., jagged edges), and the follaowing steps are devoted to neutralizing more significant ones (e.g., a black area inside the object or a set of scattered white pixels that do not belong to the main object). Also, as mentioned above, this information system provides the ability to search through several colour ranges in parallel. Demonstration of this function is presented in Figure 6.

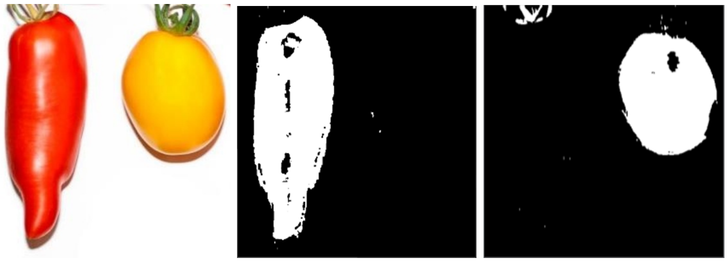

Fig. 6. Example of searching through multiple colour ranges

To do this, an external system using this information system needs to pass a dictionary (in this context, a dictionary is one of the Python language data structures) with minimum and maximum values for several ranges, as in the example below in Figure 7.

```
colour_ranges = {"red": {"min": (0, 132, 0), "max": (179, 255, 255)},
                 "green": {"min": (32, 50, 40), "max": (50, 255, 237)}}
```

Fig. 7. Example of a dictionary with colour range parameters

For more convenient selection of values for colour ranges it is possible to use special software developed for this purpose, which is not included in the information system under consideration.

As a result of colour search performed by the developed system, it may happen that an object potentially being a tomato fruit will have a black spot on the binarised image even if the search ranges are set correctly. This effect is demonstrated in Figure 8.

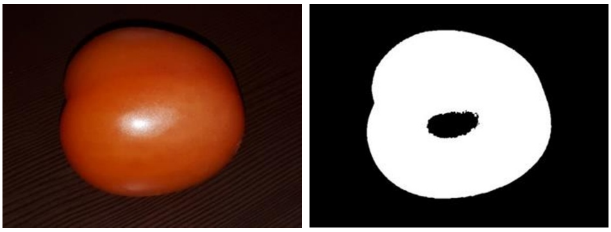

Fig. 8. Example of non-ideal colour selection

The above-described image defect may affect the correct separation of closely spaced objects due to the specificity of this operation.

The way to solve this problem is based on finding the outer contours of the object and filling all the space within this contour. OpenCV library provides such a possibility.

Contours can be considered as a curve connecting all continuous points along the boundary that have the same colour or intensity.

For better accuracy, binary images are used with the developed information system. This solution is based on the fact that in OpenCV, contour search consists of finding a lighter object against a darker one. An example of the found contour is shown in Figure 9 (for demonstration purposes, the contour is highlighted in green).

 https://doi.org/10.1051/bioconf/202413001016

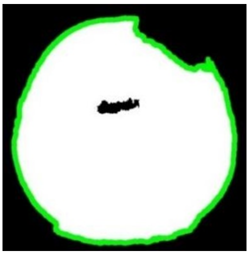

Fig. 9. Example of found external contour of an object.

After the contour is found, the whole of its inner space should be filled with white colour. Below in Figure 10 is an example of the same image that was previously given as an example of the previous method of removing black areas from objects in a binarised image.

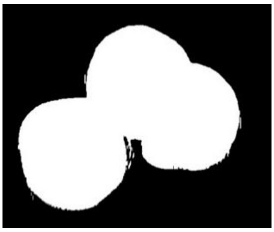

Fig. 10. The result of colouring black areas of objects using contour search

When comparing Figures 8 and 10, we can see that in the second example, the external contours of objects are not deformed in any way, and on this basis, it was decided to use the method of painting with contour search in the information system under consideration.

During the analysis of requirements to the designed information system it was revealed that often there may be cases when several objects possessing the desired colour and closely located, touch and, thus, partially or completely overlap each other. The result is that in a binarised image such objects merge into one large white area, as in Figure 11.

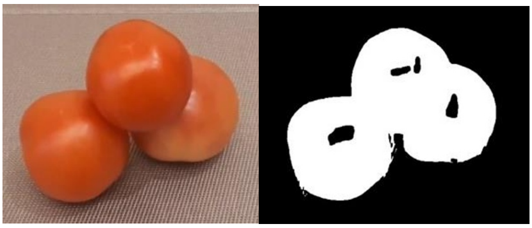

Fig. 11. Tomatoes merged into one large object, too close together relative to each other

Attempting to recognise the above image by the developed system without preliminary separation of objects will lead to the fact that in the end only one object or none at all will be recognised (since the shape of the resulting white area has little in common with the shape of a tomato fruit).

To solve this problem, we have considered a method - Watershed Algorithm.

The first step in applying the watershed algorithm in the developed information system for segmentation is the construction of a distance map for the binarised image, as shown in Figure 12.

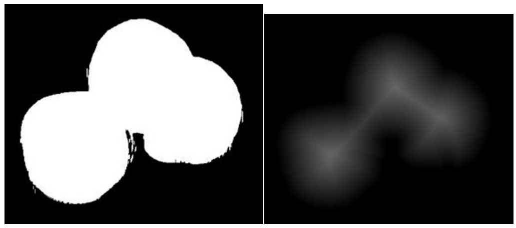

Fig. 12. Example of a distance map (right) for a binarised image

The distance map is based on the Euclidean Distance Transform (EDT). The map shown in Figure 14 is a demonstration of EDT. The distance_transform_edt function of the Scipy library for the Python programming language was used to build this map.

The next step is to find the local maxima of the obtained distance map. These points represent the centres of the assumed objects and it is from them that the segmentation of objects will start directly (the "water filling" described above). The function peak_local_max of the Skimage library is used to find local maxima.

The result of this function is binarised images for each individual object (segment of the original single object) found during the watershed algorithm. An example is shown in Figure 13.

The objects obtained as a result of the above segmentation will be analysed separately in the external geometric shape filtering step.

Fig. 13. Segments of the original merged object

It should be noted that this stage requires large calculations and, therefore, requires more time. If it is necessary to increase the speed of operation of the system under consideration and it is known that tomato fruits in the source images are not touching with high probability, the developed information system provides the possibility of switching off the stage of separation of partially overlapping objects (and together with it the stage of removing black areas of objects. Disabling occurs at the stage of transferring the source materials for recognition.

3 Discussion

The final stage of the developed information system is the search for objects whose shape is similar to the shape of tomato fruit in the images obtained at the previous stages.

There are several approaches to solving this problem using OpenCV library tools. We have chosen the approach related to the use of simple blob detector, which is a part of OpenCV library. This method is used in the developed information system. It allows to find areas on the image that have a common property (in the context of computer vision such area is called "blob") by the following key characteristics:

– colour;

– minimum distance between the centres of the areas;

– area (minimum and maximum values);

– roundness (minimum and maximum values);
– convexity (minimum and maximum values);
– inertia (minimum and maximum values).

Parameter setting is performed before creating a detector instance by assigning selected values and boolean variables to the corresponding fields of the SimpleBlobDetector_Params class to activate the corresponding filters. Let us consider the actions performed in this algorithm in more detail.

First of all, we need to create an instance of the SimpleBlobDetector_Params class containing all the settings for further filtering of objects by the detector (simple blob detector). An important and not quite obvious feature of this class is that some filters are enabled in advance and for further correct work with the detector in question you should make sure that all filters that are not planned to be used are switched off.

The next step is to disable the Colour filter. This filter will not be needed for further analysis of objects, as the colour selection in this developed information system was performed at the stage of colour search.

The next step is to switch on the area filter and load the minimum and maximum area values of the searched objects. The area filter is used for filtering too small or, on the contrary, too large objects. The range of the searched areas in the developed information system is calculated on the basis of the size of the initial image. This is necessary because objects with absolute values of areas will have different relative sizes on images with different resolutions. For use in the developed information system, it was decided to use values from 1/225 to 1/2.25 values of the area of the original image.

The next steps are to include and load the minimum values of the roundness, convexity and inertia filters. The values of these filters were calculated experimentally and are 0.6, 0.5, 0.4 respectively. The characteristics used by these filters require a separate explanation.

Roundness is a metric that allows us to determine how close to a circle a figure is geometrically. That means a regular hexagon has more roundness than a square. The numerical value of roundness is determined by the formula (1).

$$\mathrm{C} = \frac{4\pi S}{(P)^2} \tag{1}$$

where: C – roundness;
S – square;
P – perimeter.

It follows from this formula that a circle has a roundness equal to 1, and the roundness of a square is 0.785.

The numerical value of convexity is calculated by dividing the area of the figure by the area of its convex hull. The convex hull of a figure is the smallest convex geometric figure that completely covers the original figure. Thus, the value of convexity belongs to the set from 0 to 1.

Inertia Ratio shows how much the geometric figure is stretched. For clarity, Figure 14 shows examples of figures with different inertia ratios (the green figure on the right has a smaller ratio than the blue figure on the right) [5].

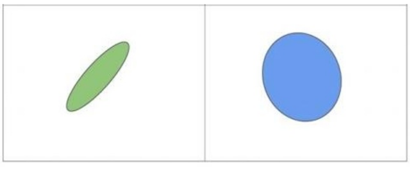

Fig. 14. Example of figures with different coefficients of inertia

https://doi.org/10.1051/bioconf/202413001016

For a circle this coefficient is equal to 1, for an ellipse - from 0 to 1, and for a line - 0.

After making the above settings, it is necessary to create an instance of SimpleBlobDetector class by passing the class with SimpleBlobDetector_Params settings to the constructor. To apply the detector, the detect method of this class should be called.

It should be noted that in the developed information system it is possible to change the detector parameters by passing the SimpleBlobDetector_Params class with the appropriate settings together with the source image at the stage of calling the initialising function of the system under consideration.

Figure 15 below shows two tomato fruits. It can be argued that this picture is produced under optimal conditions. Since:

– the original image is not noisy;

– optimal lighting of the scene (not harsh and without large shifts in cold or warm tones);

– fruit intact;

– fruits are far enough apart and not in contact with each other.

Fig. 15. Test image for recognition under optimal conditions

Fruit recognition under these external conditions is not hindered and tomato fruits are recognised with a fairly high accuracy, as in Figure 16.

Fig. 16. Recognition result under optimal conditions

As can be seen in the final demonstration image, two tomato fruits were found (and subsequently extracted) correctly.

A test image with artificially imposed noise will be used as a test image for tomato fruit recognition under noisy conditions.

For better demonstration of the noise, an enlarged fragment of the image was taken. It is possible to see, there is quite a lot of noise in the image. These noises do not affect the recognition of tomato fruit in this image by a human, but in the process of recognition by computer vision methods, the accuracy can drop significantly.

As demonstrated in the previous figure, the system worked correctly and all tomato fruits in the image were recognised.

Let's consider the impact of image noise in more detail. In

Figure 17 shows the comparison of binarised image fragments, which depict the found tomato fruit under different conditions.

 https://doi.org/10.1051/bioconf/202413001016

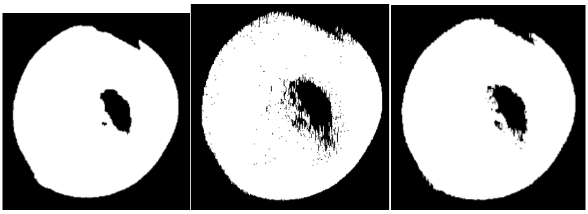

Fig. 17. Binarised images of tomato fruit when recognised under optimal conditions (left), under source image noise conditions without smoothing (middle) and under source image noise conditions with smoothing (right)

The above examples illustrate the negative impact of noise in the original image and the consequent need to deal with it. Noise can lead to incorrect operation of a detector that filters objects based on the shape of their binarised image.

The result of incorrect operation with a noisy image with no smoothing in this case resulted in the inability of the developed information system to recognise tomato fruits in the image.

To simulate the situation with image analysis with incorrect colour balance settings, we use the original image with shifted colours towards warm shades shown in Figure 18.

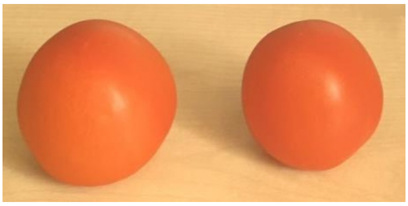

Fig. 18. Test image with incorrect colour balance

As you can see in the figure above, the colour of objects in the image has been distorted (the shade of red tomato fruits has shifted closer to orange). This factor may affect the correct operation of colour search. To neutralise the negative effects of this factor, white balance correction is performed. Demonstration of the result of the developed information system is shown in Figure 19.

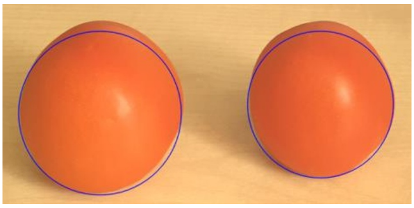

Fig. 19. Result of image analysis with incorrect colour balance

As can be seen in the above figure, the developed information system successfully recognised tomato fruits with distorted colour.

There is a problem of the procedure of separation and recognition of several partially overlapping objects in an image.

Figure 20 will be considered as an initial demonstration image in this section

 https://doi.org/10.1051/bioconf/202413001016

Fig. 20. Test image with several partially overlapping tomato fruits.

In this picture there are several tomato fruits overlapping each other. Recognising four tomato fruits in this image is not very difficult for a human being. But in the context of computer vision, it is quite common. The problem of recognising such groups is that in a binarised image all these objects, because they all have similar colours, form a single large white object, as in Figure 21.

Fig. 21. Binarised image derived from the original image.

The developed information system uses the watershed algorithm to solve such problems. The considered image shown in Figure 26 must be prepared before applying the watershed algorithm.

Preparation of a binarised image implies removal (painting) of black areas that are within the white object in the prepared image. After this preparation, we can proceed to segmentation by the watershed algorithm.

The result of solving the task is shown in Figure 22.

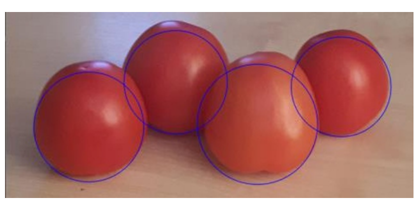

Fig. 22. Result of analysing an image with several partially overlapping tomato fruits

The above image shows that the developed information system correctly detected all four tomato fruits.

It should be noted that the information system under consideration is very likely to detect only those tomato fruits with the hidden part not exceeding approximately half of the fruit. Otherwise, the recognition accuracy will be drastically reduced.

Often the image contains not only tomato fruits, but also different kinds of extraneous objects (not taking into account the background object). As an example, Figure 23

Objects with colours uncharacteristic for tomato fruits are filtered out at the stage of search by colour and therefore cannot be present in the final binarised image and do not cause great difficulties at the stages of analysis of such an image. The problem is caused by those objects whose colours are included in the searched range. Binarised images of these objects should be sorted by other parameters. Such parameters can be properties of external geometric shapes of objects in the binarised image.

Fig. 23. Test image with foreign objects.

The OpenCV library contains tools for analysing parameters of geometric shapes. The result of filtering using such tools is shown in Figure 24.

Fig. 24. Result of image analysis with foreign objects present on it

It should be noted that this approach does not protect against misidentification as tomato fruit of objects that are very similar in colour and shape to the tomato fruit (e.g., red ball).

For correct classification based on tomato fruit colour (e.g., classification by stage of ripening or by variety), it is necessary to search for objects in the image of several colour ranges at once.

Consider an example of the following situation shown in Figure 25. The figure shows two tomato fruits of different varieties. It is necessary to classify these fruits by varieties on the basis of their external colour and to calculate their number.

To perform this task, the developed information system sequentially searches for and generates binarised images for each colour range (corresponding to a particular variety). Then each binarised image is processed separately and lists of found objects are formed for each colour range.

Fig. 25. Test image with several tomato fruits of different varieties

Demonstration of the result of the developed information system is presented in Figure 26.

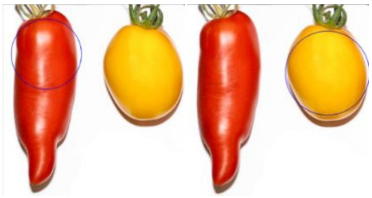

Fig. 26. Result of image analysis with several tomato fruits of different varieties with their classification by variety

The number of recognised tomato fruits can be obtained based on the length of the list with information about found objects returned by the developed information system at the end of image analysis.

The accuracy of tomato fruit recognition by the developed information system is directly related to the quality of the image and the extent to which the appearance of the tomato fruit is not distorted in the image. It follows that if an object has a shape or colour that is too uncharacteristic of tomato fruit (e.g., the fruit is damaged), it is most likely that the fruit will not be recognised in the image.

This category may also include fruit whose external geometric shape of the binarised image obtained in the colour search step is distorted.

The example above depicts tomato fruits overlapped by a stem and leaves. Since these overlapping objects have a different colour, they are not present in the final binarised image and are replaced by black areas. Because of this, the binarised image of the fruit is divided into several parts. As a consequence, each such part is analysed separately, which leads to erroneous recognition. The recognition accuracy of the developed information system is significantly reduced when foreign objects overlap the tomato fruit. For sufficiently accurate recognition it is necessary to avoid visual separation of tomato fruit in the image.

4 Conclusions

In this paper, examples of recognition of objects similar to tomato fruits in colour and external shape were presented. The results of the methods of minimising the effects of factors that negatively influence the image analysis in order to search for tomato fruits on it have been demonstrated. Weaknesses of the developed information system are described.

On the basis of the materials of this article, it is possible to draw conclusions. The developed information system for monitoring the number of tomato fruits correctly solves the following set tasks:

– correct tomato fruit recognition under image noise conditions;

– correct recognition of tomato fruits under different light conditions;

– correct recognition of adjoining and partially overlapping tomato fruits;

– correct filtration of foreign objects that do not possess features of tomato fruit appearance;

– correct classification of tomato fruits based on external colour.

It is also worth paying attention to the weaknesses of this information system:

– Possible incorrect recognition of a tomato fruit that is overly covered (half hidden) by a foreign object (in this case, the hidden fruit is most likely to be recognised as part of the fruit in the foreground);

– Possible incorrect recognition of a visually separated tomato fruit by a foreign object in the image (in this case, the fruit is most likely to be recognised as several separate objects).

It is important to take into account the weaknesses of the developed information system and, if possible, to avoid the above situations.

It should also be emphasised that the developed information system is built using computer vision algorithm in such a way that it did not use neural networks (training by examples), which can be highlighted as an advantage, because to set up the search for the necessary object does not need to search for a significantly large array of images. This approach to the development of information systems, in our opinion, is more acceptable for agriculture.

References

1. S. Dimitrios, Kasampalis, Pavlos Tsouvaltzis, Anastasios S. Siomos, Postharvest Biology and Technology **161**, 111036 (2020), https://doi.org/10.1016/j.postharvbio.2019.111036.
2. Lei Wu Pan, Yunqiang Yin, Xunyan Jiang, T.C.E. Cheng//Pattern Recognition **118**, 108022 (2021), https://doi.org/10.1016/j.patcog.2021.108022.
3. Jeevakala S., Brintha Therese A.//Biomedical Signal Processing and Control **41,** 21-30 (2018), https://doi.org/10.1016/j.bspc.2017.11.007.
4. V.V. Shipko, M.F. Volobuev, Optics and Spectroscopy **130,** 8. 1248-1255 (2022). DOI 10.21883/OS.2022.08.52911.3268-22.
5. Yamamoto, Kyosuke & Guo, Wei & Yoshioka, Yosuke & Ninomiya, Sensors **14**, 12191-12206 (2014)