FRUIT SORTING BASED ON TEXTURE ANALYSIS AND SUPPORT VECTOR MACHINE CLASSIFICATION

RAKUN, J.; BERK, P. & LAKOTA, M.

Abstract: *This paper describes an approach to classify fruit based on their texture properties. To analyse texture and to build a feature vectors for each pixel six different statistical moments are used. Then support vector machines are applied to analyse the texture feature vector of each fruit in order to classify it on of two groups. To verify the usefulness of the approach a simple fruit sorting machine was build. It is controlled by a personal computer that analyses the video stream and controls two servomotors that open the paths to two different fruit groups. It was proven that the approach reached on average 95,9% accuracy in our tests.*

Key words: *fruit sorting, texture, support vector machines, morphological operator*

Authors´ data: Assist. Prof. **Rakun**, J[urij]*; MSc. **Berk**, P[eter]*; Assoc. Prof. **Lakota**, M[iran]*; *University of Maribor, Faculty of Agriculture and Life Sciences, Pivola 10, SI-2311, Hoce, Slovenia, **University of Maribor, jurij.rakun@um.si, peter.berk@um.si, miran.lakota@um.si

This Publication has to be referred as: Rakun, J[urij]; Berk, P[eter] & Lakota, M[iran] (2015). Fruit Sorting Based on Texture Analysis and Support Vector Machine Classification, Chapter 19 in DAAAM International Scientific Book 2015, pp.209-218, B. Katalinic (Ed.), Published by DAAAM International, ISBN 978-3-902734-05-1, ISSN 1726-9687, Vienna, Austria
DOI: 10.2507/daaam.scibook.2015.19

1. Introduction

One of the crucial steps in fruit production is to sort each fruit in different categories. This is done to assess the quantity as well as the quality and set the optimal price. But because the job of sorting fruit can be a time consuming job that is also prone to human error, the sorting is usually done with the help of different sorting machines. They evaluate and sort fruit according to the shapes, outer, inner quality and even nutrients.

An example of fruit sorting machine (Evrosad, 2015) is made by French company MafRODA with a capacity to sort 10 tons per hour, where classification is made based on fruit colour, weight, quality and size. The price of such sorting line is around 3,8 million €.

Different fruit sorting machine producers use different approaches to sort fruit. In general they use digital cameras to observe the surface (Strum & Hofacker, 2009), electronic weights to measure the weight and special artificial light sources that can help to evaluate the quality of the fruit. (Greefa, 2015)

By applying different light sources to the fruit inner quality of the fruit can be evaluated, for example; degrees Brix that correspond to the sugar content and physiological errors, such as inner bubbles and brown cores. These approaches do not damage the inspected fruit and are perfectly safe to use. (Greefa, 2015)

Another quality assessment factor is weight of the fruit. By weighting, dried fruit can be separated from the good one. Again, different approaches to measure weight are applied, for example Greefa uses two; the 3-point weighing system that is to be found in the transfer unit of the fruit or it uses 3-point weighing system that is to be found in the central weighing section, below the fruit carriers (Greefa, 2015).

Modern sorting lines in general use CCD cameras to evaluate the fruit. With the help of the camera the fruit is measured multiple times from different angles. This can give an estimate about the size of the fruit as well as its quality based on the colour characteristics of the fruit. (Greefa, 2015)

In order to develop new algorithms for fruit sorting machines a simple sorting machine was developed and is described in section 2 of this paper with an approach to detect known fruit types based on their texture characteristics. To do so a multistage algorithm was developed that first transforms an image into HSV colour space (subsection 2.1), produces 6 degree texture dependent future vectors (subsection 2.2), uses support vector machine to first learn the model (subsection 2.3) and later for classification and finally, an optional step of area opening (subsection 2.4) to improve the results (section 3).

2. Material and methods

In order to test the approach a simple sorting machine was build depicted on Fig. 1. The enclosure was made out of stainless steel in shape of a letter Y and tilted, so the fruit would roll down with the help of gravity. Each fruit is inspected with the help of The Imaging source's DBK31AU03.AS digital camera (Fig. 2), connected to a personal computer. There the image of the fruit is analysed and according to the contents the fruit is diverted in one of two paths; depending on its properties. Each path can be open with the help of two servomotors controlled by the National Instrument's NI USB-6009 card (Fig. 3).

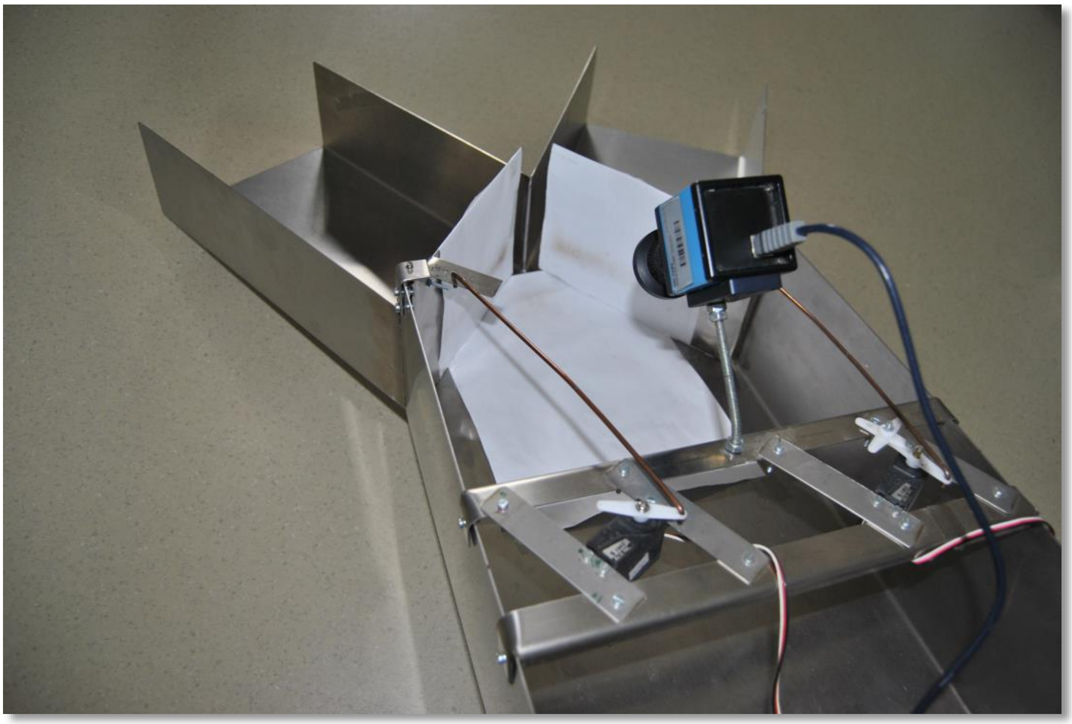

Fig. 1. A simple Y-shaped sorting line for binary classification

Fig. 2. The imaging source's DBK31AU03.AS digital camera used to capture new images on the sorting line

Fig. 3. NI USB-6009 controller card used to control the servomotors

Fig. 4 depicts an overview of the algorithm that is used in sorting machine. As an input it takes an image from the digital camera and first transforms it from RGB to HSV colour space. Transformed images are then used for two purposes; learning or classification. At first the system has to learn where the first step is to describe each pixel with a 6D feature vector based on the textures in its close proximity. Next the vectors are used to learn the system so it knows how to classify a new input image. As a result the system produces a binary image; eider it detected a known new object of known texture pattern or not and it is classified on the sorting line trough the right hatch. Each step of the algorithm is further described in the following subsections.

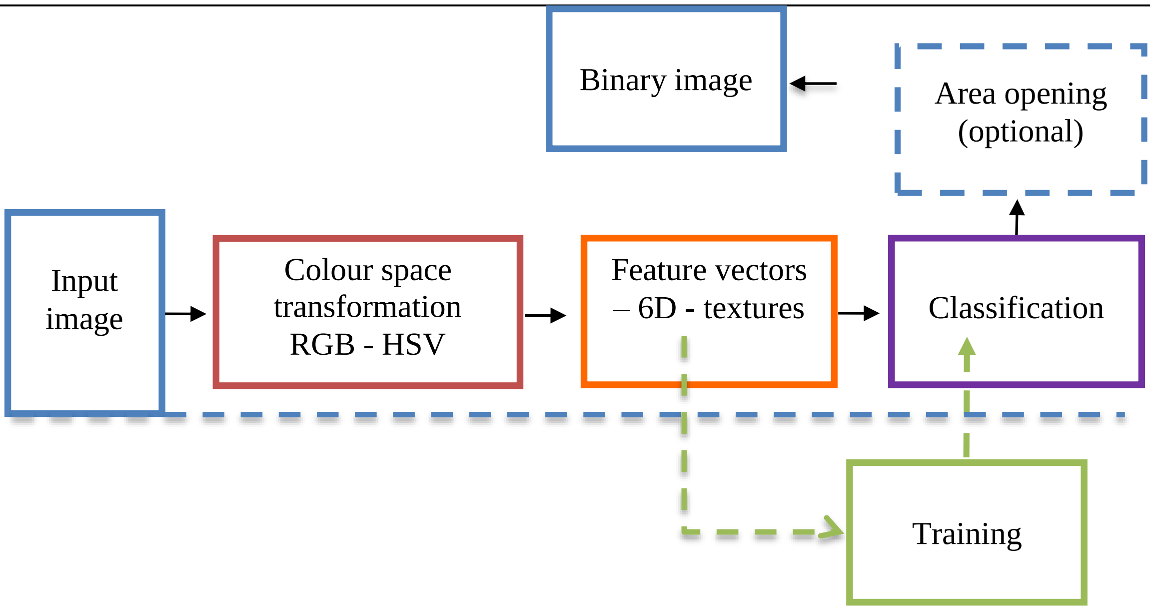

Fig. 4. An overview of the texture based sorting algorithm (green box – learning stage, violet box – classification stage)

2.1. Colour space – RGB to HSV

The first step is to transform the image from RGB to HSV colour space (Gonzales & Woods, 2008). By applying this transformation the pixels are described by hue, saturation and brightness instead of three basic colour shades - red, green and blue. Because we are trying to analyse the textures made up of different combinations of colour, only the first plane of the HSV representation is selected; hue.

2.2. Feature vectors

By analysing the hue plane of an image, every pixel can be described with the help of its feature vector by analysing a small neighbour hood of 7×7 pixels, with the current pixels as a centre of 7×7 window. For each pixel a 6-value vector is constructed by including a mean value, standard deviation, smoothness, third moment, uniformity and entropy (Gonzales et al., 2003).

Mean value is included as part of the future vector as it describes average colour shades in a window. This means the procedure not only analyses different patterns of textures, but also takes into account the colour values. If p is a histogram of hue values, L is the number of all pixels in a window and z_i is a value of current pixel the mean value is then calculated as shown by Eq. (1):

$$m = i = 0L - 1zip(zi) \tag{1}$$

Next is the standard deviation (Eq. (2)) that outlines the variation or dispersion of hue values. This corresponds to average contrast value in a window. If $u_2(z)$ is a variance then standard deviation is computed as:

$$\sigma = u2(z) \tag{2}$$

Smoothness value describes how smooth are the surfaces of the texture. The value is lower for areas where values do not change much and higher for those that have prominent textures with frequent extreme values. Smoothness is computed as follows:

$$R = 1 - 11 + \sigma 2 \tag{3}$$

Third moment computed by Eq. (4) summarizes the shape of the histogram of hue values. If it is symmetric relative to the mean value, the value of symmetry is close to 0, if it is skewed to the left the value is negative and if it is skewed to the right the value is positive.

$$\mu 3 = i = 0L - 1zi - m3p(zi) \tag{4}$$

Uniformity specifies how hue values are distributed in a histogram. If the values are all the same, the uniformity reaches a maximum value, as shown by Eq. (5) and less, if the values change.

$$U = i = 0L - 1p2(zi) \tag{5}$$

The last value is entropy that measures the randomness of the values in a window, computed by Eq. (6). Its value is higher if the values of the texture change randomly and lower for smoother or periodic textures.

$$e = -i = 0L - 1p(zi)log2p(zi) \tag{6}$$

2.3. Classification - support vector machines

The computed feature vectors correspond to different textures and they must be analysed. To do so, we have chosen a support vector machines or SVM as described by (Vapnik, 1999), (Haykin,1999), (Ivanciuc, 2007) and (Boser et al., 1992). The SVM approach can be used for linear regression analysis or pattern classification, as in our case. To analyse an image, we use statistical distribution for each of the pixels and, based on the texture patterns, conclude if the pixel belongs to an area with the texture we are trying to detect or not. This in affect produces a binarized image, revealing pixels that make up a familiar texture pattern.

SVM works by applying directed neural network in order to estimate a hyper plane that separates one group from the other. The process starts with a learning step that initializes the weights of the neural network in a way that the hyper plane is able to classify a member in one of two groups. Every group member is therefore presented with a help of *n*-dimensional feature vector and based on its contents it is separated with the help of a liner classifier or (*n*-1) dimensional hyper plane.

SVM can be defined according to (Haykin, 1999) in the following way. We use x_i to represent a feature vector, for which we select a response d_i, with values of 1 or -1 that classify the object according to the features in one of two groups. A hyper plane can then be written as:

$$\mathbf{w}T\mathbf{x} + \beta = 0, \tag{7}$$

where **w** represents a weight vector, β a bias and **x** an input feature vector. For an unknown feature vector $\mathbf{x}i$ the Eq. (7) changes to:

$$\mathbf{w}T\mathbf{x}i + \beta \geq 0, \text{ for } d_i = 1 \quad (8)$$

and

$$\mathbf{w}T\mathbf{x}i + \beta < 0, \text{ for } d_i = -1. \quad (9)$$

The vectors that comply with Eq. (8) and Eq. (9) fit the hyper plane perfectly and we name them support vectors. In order to make a classification we need to calculate the weight parameter **w** and a bias β. To do so, we use a quadratic optimization (Haykin, 1999) and Lagrange multiplayer approach (Haykin, 1999) that are covered elsewhere and will not be recapitulated here.

Once the support vectors are known, we can proceed with a classification of an unknown object, based on its feature vector. In our case the object pixels are represented with 6 dimensional feature vectors computed out of texture measures from subsection 2.2. However, the classification on an object will only be as good as it is the learning set of objects defined by its feature vectors used during the learning step.

2.4. Post processing - area opening

Area opening is a morphological operator (Gonzales & Woods, 2008) that eliminates smaller regions on binary images, caused by noise, imperfect classification, etc. It works by eliminating areas that have fewer then a pre set threshold value. For our experiments we have set a predefined threshold value of 150 pixels, which is 0,17% of our picture size or 4,6% of average test object size.

3. Results

In order to test the approach 19 randomly selected examples of fruit (apple and peach) were selected. The goal was to verify if the system can detect new fruit and if it can classify the pixels of an image correctly. The first 9 examples were of peaches and the other 10 were of apples. In all test cases the images were of 300×300 pixels in size and pixels were counted on a final binarized image and compared to original were pixels that make up the fruit were counted manually. This produced a number of true positives, false positives, true negatives and false positive pixels. The numbers are summarized in Tab. 1.

The results from Tab. 1 show that on average 93,2% of right fruit pixels were detected and 6,8% not. The reason for these 6,8% are border pixels where the values of feature vectors get influenced by the surrounding pattern.

Example	True positives	True positives [%]	False positives	False positives [%]	True negatives	True negatives [%]	False negatives	False negatives [%]
1	3677	98.1%	70	1.9%	83563	96.8%	2760	3.2%
2	1881	85.3%	324	14.7%	84940	96.8%	2855	3.2%
3	2198	93.2%	161	6.8%	84889	96.9%	2752	3.1%
4	2040	92.3%	170	7.7%	85099	96.9%	2691	3.1%
5	2704	92.7%	212	7.3%	84083	96.6%	3001	3.4%
6	3297	99.4%	19	0.6%	83783	96.7%	2901	3.3%
7	6481	92.4%	536	7.6%	80747	97.3%	2236	2.7%
8	1903	98.6%	27	1.4%	85526	97.1%	2544	2.9%
9	2322	92.5%	189	7.5%	84664	96.8%	2825	3.2%
10	3637	90.4%	388	9.6%	82777	96.3%	3198	3.7%
11	4002	89.3%	478	10.7%	82359	96.3%	3161	3.7%
12	4659	97.2%	134	2.8%	82552	96.9%	2655	3.1%
13	4155	96.5%	149	3.5%	82577	96.4%	3119	3.6%
14	4085	94.9%	220	5.1%	82588	96.4%	3107	3.6%
15	4591	99.5%	22	0.5%	82524	96.7%	2863	3.3%
16	3165	84.6%	577	15.4%	82932	96.1%	3326	3.9%
17	3643	88.2%	486	11.8%	82617	96.2%	3254	3.8%
18	3371	92.6%	269	7.4%	83365	96.5%	2995	3.5%
19	4422	93.2%	323	6.8%	82267	96.5%	2988	3.5%
Average:	3486	93.2%	250	6.8%	83361	96.6%	2907	3.4%
St. dev:	1168	4.4%	176	4.4%	1239	0.3%	269	0.3%

Tab. 1. The results of 19 randomly selected examples

The true negative values on the other hand reach on average 96,6% with 3,4% of false positives. These are small regions that get miss-classified due to noise, similar textures of background compared to fruit, small learning sets, etc. Because these regions are small the results can be improved by morphological operators, were we choose area opening described in subsection 2.4 and repeated the test. The results are summarized in Tab. 2.

Example	True positives	True positives [%]	False positives	False positives [%]	True negatives	True negatives [%]	False negatives	False negatives [%]
1	3677	98.1%	70	1.9%	85261	98.8%	1062	1.2%
2	1881	85.3%	324	14.7%	86730	98.8%	1065	1.2%
3	2198	93.2%	161	6.8%	86679	98.9%	962	1.1%
4	2040	92.3%	170	7.7%	86889	99.0%	901	1.0%
5	2704	92.7%	212	7.3%	85872	98.6%	1212	1.4%
6	3297	99.4%	19	0.6%	85580	98.7%	1104	1.3%
7	6481	92.4%	536	7.6%	81164	97.8%	1819	2.2%
8	1903	98.6%	27	1.4%	87316	99.1%	754	0.9%
9	2322	92.5%	189	7.5%	86503	98.9%	986	1.1%
10	3637	90.4%	388	9.6%	84475	98.3%	1500	1.7%
11	4002	89.3%	478	10.7%	84057	98.3%	1463	1.7%
12	4659	97.2%	134	2.8%	84250	98.9%	957	1.1%
13	4155	96.5%	149	3.5%	84275	98.3%	1421	1.7%
14	4085	94.9%	220	5.1%	84286	98.4%	1409	1.6%
15	4591	99.5%	22	0.5%	84222	98.6%	1165	1.4%
16	3165	84.6%	577	15.4%	84630	98.1%	1628	1.9%
17	3643	88.2%	486	11.8%	84315	98.2%	1556	1.8%
18	3371	92.6%	269	7.4%	85063	98.5%	1297	1.5%
19	4422	93.2%	323	6.8%	83965	98.5%	1290	1.5%
Average:	3486	93.2%	250	6.8%	85028	98.6%	1240	1.4%
St. dev:	1168	4.4%	176	4.4%	1451	0.3%	282	0.3%

Tab. 2. The results of 19 randomly selected examples with area opening applied

Compared to the results from Tab. 1 a conclusion can be made that area opening does not affect the true/false positive values, as these regions are quite big. The false negative values on the other hand are smaller and get almost eliminated, where only 1,4% of values remain. Further inspection showed that these are border values and get connected to the region that corresponds to the location of fruit.

4. Conclusions

In this work we have presented an approach to detect and classify fruit pixels on a simple fruit sorting machine. Based on the results the fruit can be classified in on of the groups; known or good texture patterns in one, and other in the second group. For the classification a SVN based approach was used that classifies each hue valued pixels in one of two groups. As an additional step morphological step of area opening can be applied to further improve the results.

We have showed that in all 19 examples the procedure detects majority of fruit pixels, on average reaching 93,2%. The rest are border pixels where their features get influenced by surrounding area. Only 1,4% of pixels get misclassified and these are those that are connected to the area representing a fruit on a binarized image. If no fruit is present, they get eliminated. By taking into account true positives and true negatives the selected approach reaches on average 95,9% accuracy in our tests.

In order to improve the accuracy of the system, different steps could be tested. The first is region growing for the border pixels, which would improve the true positive numbers. The second is bigger learning data sets where the number of false negatives would drop even further.

5. References

Boser, B. E.; Guyon, I. M. & Vapnik, V. N. (1992). A Training Algorithm For Optimal Margin Classifiers, *5th Annual ACM Workshop on COLT*, ACM Press, pp. 144-152, ISBN:0-89791-497-X, New York, USA

http://www.evrosad.si - Evrosad, *Accessed on 2014-09-04*

Gonzales, R. C.; Woods, E. R. & Eddins, S. L. (2003). *Digital Image Processing using Matlab*. New Jersey, ISBN: 978-0130085191, Prentice Hall

Gonzales, R. C.; Woods, E. R. (2008). *Digital Image Processing Third Edition*. New Jersey, ISBN: 978-0131687288, Prentice Hall

http://www.greefa.nl/UK/products-measuring-systems.htm - Greefa, Measuring systems, *Accessed on 2014-09-04*

Haykin, S. (1999). *Neural Networks: A Comprehensive Foundation*, 2nd edition, New Jersey, ISBN: 978-0132733502, Prentice Hall

Ivanciuc, O. (2007). Applications of Support Vector Machines in Chemistry, *Reviews in Computational Chemistry*, vol. 23., Wiley-VCH, pp. 291-400, ISBN: 978-0470116449

Strum, B.; Hofacker, W. (2009). Optical monitoring and control of drying processes, *DAAAM international scientific book 2009*, pp. 501-512, ISBN: ISBN 978-3-901509-69-8, Vienna, Austria

Vapnik, V. N. (1999). An Overview Of Statistical Learning Theory, *IEEE Transaction on Neural Networks*, vol. 10, pp. 988-999, ISSN: 2162-237X