Ship Detection in SAR Using Extreme Learning Machine

Liyong Ma(✉), Lidan Tang, Wei Xie, and Shuhao Cai

School of Information and Electrical Engineering,
Harbin Institute of Technology, Weihai 264209, China
maly@hitwh.edu.cn

Abstract. Ship detection is an important issue in many aspects, vessel traffic services, fishery management and rescue. Synthetic aperture radar (SAR) can produce real high resolution images with relatively small aperture in sea surfaces. A novel method employing extreme learning machine is proposed to detect ship in SAR. After the image preprocessing, some features including entropy, contrast, energy, correlation and inverse difference moment are selected as features for ship detection. The experimental results demonstrate that the proposed ship detection method based on extreme learning machine is more efficient than other learning-based methods with prior performance of accuracy, time consumed and ROC.

Keywords: Ship recognition · Extreme learning machine
Synthetic aperture radar (SAR)

1 Introduction

Ship detection is an important issue in many aspects, vessel traffic services, fishery management and rescue. Traditional ship detection method such as patrol ships or aircrafts are costly and limited by many circumstances, coverage area and weather condition. Particularly, because of many air cash in recent years, ship detection has become more and more important for ship monitoring and ship searching to save people in time.

Synthetic aperture radar (SAR) can produce images of objects, such as landscapes and sea surfaces. SAR is usually mounted on mobile platforms such as aircrafts. The movement of platform can acquire a lager synthetic antenna and provide better azimuth resolution. Generally speaking, the larger aperture is, the higher image of resolution it will be, no matter the aperture is physical or synthetic. For these reason, SAR can produce a real high resolution image with relatively small aperture. There are many other advantages for us to use SAR images. SAR images can be obtained in many circumstances, regardless of whether it is during day or night, rain or snow. SAR system can be so useful when optical tools can not be used.

X. Gu et al. (Eds.): MLICOM 2017, Part II, LNICST 227, pp. 558–568, 2018.
https://doi.org/10.1007/978-3-319-73447-7_60

SAR has been widely used for ship detection. A K-means clustering and land masking method was reported on a novel method in coastal regions of SAR images in [1]. A morphological component analysis method was developed in [2] to achieve satisfactory results in complex background SAR images. A new technique using color and texture from spaceborne optical images as a complementary to SAR-based images was employed in [3].

Some ship detection methods using SAR based on common machine learning algorithms were also reported. Neural network based method was used in SAR to acquire better results in rough water and false alarm rates in [4]. Texture features from SAR image to discriminates speckle noise from ship was also reported in [5]. A method based on support vector machines (SVM) combined with grid optimization was employed for false alarm removal in [6], and tensor based approaches were also reported in [7,8]. There are some papers that use deep learning in SAR for ship detection [9], they present a high network configuration used for ship discrimination.

Extreme learning machine (ELM) is a machine learning algorithm with simple optimization parameters, fast speed and good generalization performance [10–12]. It has been widely used in many applications of image processing and machine vision [13–15]. In comparison with SVM, ELM has a simple implementation and requires less optimization works. Meanwhile, ELM has better learning performance with fast speed for its better generalization ability while SVM can just achieve sub-optimal solutions with higher computational complexity. In this paper, we perform ship recognition in SAR employing ELM method. One of the main contributions of this paper is that an efficient ship recognition method employing ELM for SAR is proposed.

2 Extreme Learning Machine

ELM algorithm includes two steps. The first step is data mapping where input data are mapped into the hidden layer employing random feature mapping or kernel learning approach. The second step is output. The final output can be obtained by multiplying the middle results with their corresponding weights. Different from the traditional learning process of three-layer neural networks where all the parameters are tuned iteratively and severe dependency of the parameters between different layers limits the learning performance, the hidden layer is non-parametric in ELM. This simple policy leads to the smallest training error and the smallest norm of weights, therefore ELM can achieve superior generation performance over other learning approaches for neural networks. ELM is able to improve the performance of single hidden layer neural network which is low and easy to be trapped in local minimums.

Denote the training sample as $(\mathbf{x}_i, \mathbf{t}_i)$, $i = 1, ..., N$, the input feature vectors $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N]^T \in \mathbb{R}^{D \times N}$, and the label of the supervised sample output $\mathbf{t}_i = [t_{i1}, t_{i2}, ..., t_{iM}]^T \in \mathbb{R}^M$, where N is the sample number, D is the dimension size of the input sample feature vector, and M is the number of network output nodes those are used to solve multi-classification problems.

Denote the number of hidden layer nodes as L, the output of a hidden node indexed by i is

$$g(\mathbf{x}; \mathbf{w}_i, b_i) = g(\mathbf{x} \cdot \mathbf{w}_i + b_i). \tag{1}$$

where $i = 1, ..., N$, $\mathbf{w}_i$ is the input weight vector between the i-th hidden nodes and all input nodes, g is the activation function, and b_i is the bias of this node. A feature mapping function which connects the input layer and the hidden layer is

$$\mathbf{h}(\mathbf{x}) = [g(\mathbf{x}; \mathbf{w}_1, b_1), g(\mathbf{x}; \mathbf{w}_2, b_2), ..., g(\mathbf{x}; \mathbf{w}_L, b_L)]. \tag{2}$$

Denote the output weight between the i-th hidden node and the j-th output node as β_{ij}, where $i = 1, ..., L$, and $j = 1, ..., M$. The value of j-th output node can be obtained by

$$f_j(\mathbf{x}) = \sum_{i=1}^{L} \beta_{ij} \times g(\mathbf{x}; \mathbf{w}_i, b_i)). \tag{3}$$

Thus the output vector of the input sample $\mathbf{x}$ at he hidden layer can be described as

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x})..., f_M(\mathbf{x})] = \mathbf{h}(\mathbf{x})\boldsymbol{\beta}, \tag{4}$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \end{bmatrix} = \begin{bmatrix} \beta_{11} & \dots & \beta_{1M} \\ \vdots & \ddots & \vdots \\ \beta_{L1} & \dots & \beta_{LM} \end{bmatrix}. \tag{5}$$

The above ELM calculation can be summed up as following. After randomly select the value of input weights $\mathbf{w}_i$ and the bias of the neural network b_i, we can obtain the output H of hidden layers, therefore the output weights $\boldsymbol{\beta}$ can be obtained. During the training, $\boldsymbol{\beta}$ is obtained based on solving an optimization problem. And during the recognition, the maxim f_j is selected as the ELM output class label.

3 ELM Based Ship Detection Method

A ship detection approach employing ELM classification for SAR images is proposed in this paper. We will discuss image pre-processing, feature extraction and network training of the proposed approach as follows.

3.1 Image Pre-processing

We used SAR image that is derived from TerraSAR-X images dataset [16]. SAR image is calibrated and geocoded for pre-processing. In this stage, the main aim is to extract ship candidate area with reducing alarm rate as much as possible. Pre-processing includes feature selection and image segmentation. Due to the ship shapes are regular, long and thin, selecting the appropriate length-width ratio as the given threshold can eliminate non-ship objects, such as irregular islands and clouds. After segmentation, we can remove non-target objects and

gain ship candidate region. Then we use median filter to eliminate noise and enhance the accuracy of ship recognation. Finally we can cut out sample images based on the size of the ships to locate target areas and extract out all the ship samples and water samples with the size of 30×30 pixels.

Feature extraction is the extraction of ships and seawater texture. The texture are important information for the ship detection. To obtain more information, sample images processing tasks are employed to extract texture features. To get the texture information, a popular gray level co-occurrence matrix method [17] is employed. The element of the matrix is the coherent distribution for probability of a gray scale in constant distance. The matrix is obtained after scaning all the pixels with the reflection of co-occurrence time probability of related pixels in space. Therefore it can provide the joint probability distribution of the pixels. The gray level co-occurrence matrix is used for five texture parameters, they are entropy, energy, contrast, correlation and inverse difference moment.

3.2 Feature Extraction

Because ship has limited area, length and width range, we extracted the ship candidate area by selecting the appropriate ratio of the length-width and area size range. We not only segment target but also eliminate non-target objects. The following equation is used to calculate the ratio of length-width.

$$r = L_{MER}/W_{MER} \tag{6}$$

where r is the ratio of length-width, L is the length of the spindle direction, and W is the width of spindle direction. The ratio distinguishes between a rectangular objects and irregular objects.

Entropy. Since texture information is also important for ship detection, we use entropy as one of texture features. Entropy can provide the uniformity and complexity information of the texture, and it is calculated as

$$f_{entropy} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} p(i,j) \times [-\ln p(i,j)] \tag{7}$$

where $p(i,j)$ is the element of the position (i,j) in gray level co-occurrence matrix which has been described in Sect. 3.1.

Contrast. Contrast feature i able to reflect the degree of the image sharpness and the depth of the texture groove. It defers the calculation of the intensity contrast linking pixel and its neighbor over the whole image. Contrast can be calculated as

$$f_{contrast} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (i-j)^2 \times p(i,j)^2. \tag{8}$$

Energy. Energy is the sum of square of elements in GLCM. It reflects the uniformity of the image gray scale and texture roughness. Energy is calculated as

$$f_{energy} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} P(i,j)^2 \tag{9}$$

Correlation. Correlation measures the similarity of the spatial gray-level co-occurrence matrix elements in the row or column direction, reflecting the local gray correlation in the image. It is calculated as

$$f_{Correlation} = \frac{\sum_i \sum_j i \times j \times P(i,j) - \mu_x \times \mu_y}{\sigma_x^2 \times \sigma_y^2} \tag{10}$$

Inverse Difference Moment. It reflects the homogeneity of the image texture, and measures how much of the image texture changes. When its value become larger, the local texture is very uniform. The following is its calculation formula.

$$f_{Inverse\ difference\ moment} = \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} \frac{P(i,j)}{1+(i-j)^2} \tag{11}$$

3.3 Network Training

In our ship detection method in SAR using ELM network, the input layer is connected to the input feature vector which is texture features as described before. The output layer has one node used to mark ship or not. After defining the ELM network structure, the network can be used for sample training. The parameters of input weights $\mathbf{w}_i$ and biases b_i in ELM are randomly selected. Consequently the calculation of $\boldsymbol{\beta}$ is critical for ELM training.

Denote the actual output vector as $\mathbf{Y}$, the input vector $\mathbf{X}$, We can obtain the output vector from (3) as

$$\mathbf{Y} = \mathbf{H}\boldsymbol{\beta}, \tag{12}$$

where

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} \\ &= \begin{bmatrix} g(\mathbf{x}_1;\mathbf{w}_1,b_1) & \dots & g(\mathbf{x}_1;\mathbf{w}_L,b_L) \\ \vdots & \ddots & \vdots \\ g(\mathbf{x}_N;\mathbf{w}_1,b_1) & \dots & g(\mathbf{x}_N;\mathbf{w}_L,b_L) \end{bmatrix}, \end{aligned} \tag{13}$$

and

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{Y}_N \end{bmatrix} = \begin{bmatrix} y_{11} & \dots & y_{1M} \\ \vdots & \ddots & \vdots \\ y_{N1} & \dots & y_{NM} \end{bmatrix}. \tag{14}$$

The object of ELM method is to minimize two errors, they are the training error $||\mathbf{T} - \mathbf{H}\boldsymbol{\beta}||^2$ and the norm of output weight $||\boldsymbol{\beta}||$. This problem can be converted to an optimization problem as below

$$\min \quad \psi(\boldsymbol{\beta}, \boldsymbol{\xi}) = \tfrac{1}{2}||\boldsymbol{\beta}||^2 + \tfrac{C}{2}||\boldsymbol{\xi}||^2 \tag{15}$$

$$\text{s.t.} \quad \mathbf{H}\boldsymbol{\beta} = \mathbf{T} - \boldsymbol{\xi}. \tag{16}$$

where $\boldsymbol{\xi}$ is the output value error between the actual output and the desired output, and C is the regularization factor which is used to improve the training generalization performance with controlling the tradeoff between the closeness to the training data and the smoothness of the decision function. The above optimization problem can be solved employing Lagrange multiplier technique. When the matrix $(\mathbf{I}/C) + \mathbf{H}^T\mathbf{H}$ is not singular, solution $\boldsymbol{\beta}$ can be calculated as

$$\boldsymbol{\beta} = \left(\frac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1} \mathbf{H}^T\mathbf{T}. \tag{17}$$

Otherwise, when the matrix $(\mathbf{I}/C) + \mathbf{H}\mathbf{H}^T$ is not singular, solution $\boldsymbol{\beta}$ can be calculated as

$$\boldsymbol{\beta} = \mathbf{H}^T \left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{T}, \tag{18}$$

where $\mathbf{I}$ is an identity matrix. In practice, when the number of training features of samples is greater than the one of hidden neurons, we use (17) to obtain the output weights, otherwise we use (18).

To improve the stability of ELM in calculating the output weights, an efficient solution is to find high quality mapping between input and hidden layers. RBF function is one of the most efficient mapping functions, and it used in our ELM based bubble defect detection method. $\mathbf{H}^T\mathbf{H}$ in (17) or $\mathbf{H}\mathbf{H}^T$ in (18) is called ELM kernel matrix, and $\mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x}_j)$ is ELM kernel. In our proposed method, the following Gaussian function is selected as the kernel

$$\phi(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma^2}\right). \tag{19}$$

The training process is performed as described above. After the training is finished, the trained ELM can be used for image detection in SAR.

4 Experimental Results

In our experiments, these SAR images acquired from TerraSAR-X Data Samples [16] have been radiometrically calibrated. Two sample images in our test dataset are illustrated in Fig. 1.

Some other usually used methods are assessed in our experiments to verify the efficiency of our proposed method. It has been widely validated that neural network based classification has poorer performance than SVM based one. Our test methods are K-nearest neighbors (KNN) [1], SVM [6], CNN [9] and ELM

Fig. 1. Sample images from the dataset

methods. After testing the different parameters with experiments, we select the optimized parameters for the best performance from the allowed ranges employing optimization search. In our ELM based method, the number of hidden nodes is set to 200.

4.1 Accuracy

The comparisons of classification performance in a variety of sample numbers are performed. The training sample number is selected from 120 to 480 with the step of 120, and these 4 groups are marked as group A, B, C and D, respectively. Samples in each group are stochastically selected from the dataset, and half of the sample is selected from the ship images, and others from non-ship ones. Test sample data contains 10 ship samples and 10 seawater samples. Each experiment has been tested 10 times. The final test accuracy is the average of 10 test experiments.

The comparison of classification correction rate of different methods is listed in Table 1. And these experimental results are also illustrated in Fig. 2. As shown in the figure, with the increasement of training samples, the classification accuracy of our proposed method gradually increases as well. The classification accuracy is more prior to other methods when fewer training samples are employed. Our proposed ELM based method is able to keep the superior performance of accuracy and the faster speed with the different sample numbers. It means that our proposed ELM method has most satisfactory performance with varies of sample number.

Table 1. Classification accuracy of different methods

Samples number	Method	Accuracy	Time consumed (s)
A (480 samples)	KNN	0.850	0.002
	SVM	0.805	0.249
	CNN	0.350	11.975
	ELM	0.900	0.161
B (360 samples)	KNN	0.850	0.002
	SVM	0.765	0.167
	CNN	0.300	5.837
	ELM	0.880	0.140
C (240 samples)	KNN	0.700	0.002
	SVM	0.750	0.150
	CNN	0.350	4.780
	ELM	0.865	0.110
D (120 samples)	KNN	0.750	0.001
	SVM	0.680	0.024
	CNN	0.350	3.502
	ELM	0.870	0.060

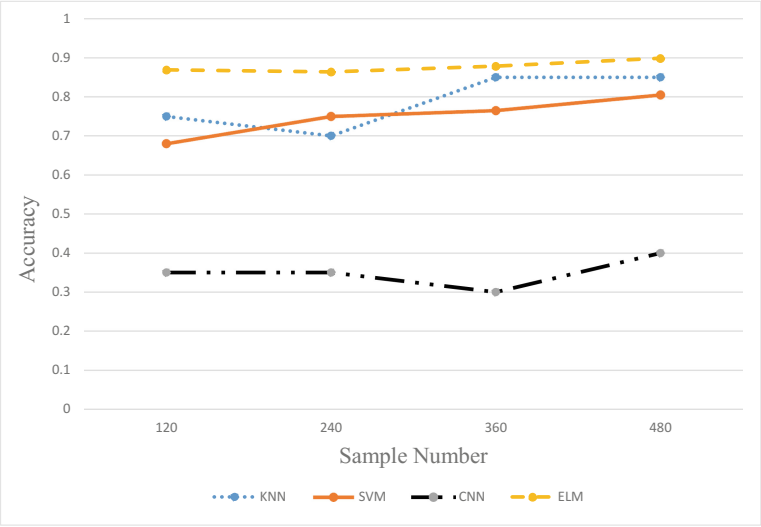

Fig. 2. Comparison of defect detection classification accuracy with different methods

4.2 Time Consumed

The speed is measured with the sum of training and test time, and the experimental results are also listed in Table 1. The time unit is seconds. The test data obtains 20 samples, which has 10 ship samples and 10 water samples. We performed the experiments with Matlab R2014a and Ubuntu 14.04 for python program on a computer with 64G RAM, 2.10 GHz E5-2620 CPU. And CNN method is tested with a GPU of GeForce GTX 1080 Ti. CNN has 50 iteration times for every training experiment. The data reveals that the proposed ELM algorithm is so fast that it can be employed in real time applications.

4.3 ROC and AUC

Receiver operating characteristics (ROC) curve and Area under the ROC curve (AUC) are widely used for classification performance comparison, and they are also employed in our experiments. We set the threshold varying from maximum to minimum and get the ROC curve. Interpreted as the probability that a classifier is able to distinguish a randomly chosen positive instance from a randomly chosen negative instance, AUC value is greater when the classification has good performance. The detailed ROC curves are illustrated in Fig. 3 in which the number of samples is 480. The curves of our proposed method are all on the above in the figures. It reveals that our proposed method has the best classification ability than other approaches. AUC value is listed in Table 2. Our proposed method obtain the greatest AUC value. All these experiments have shown that our proposed ELM based method has the best performance.

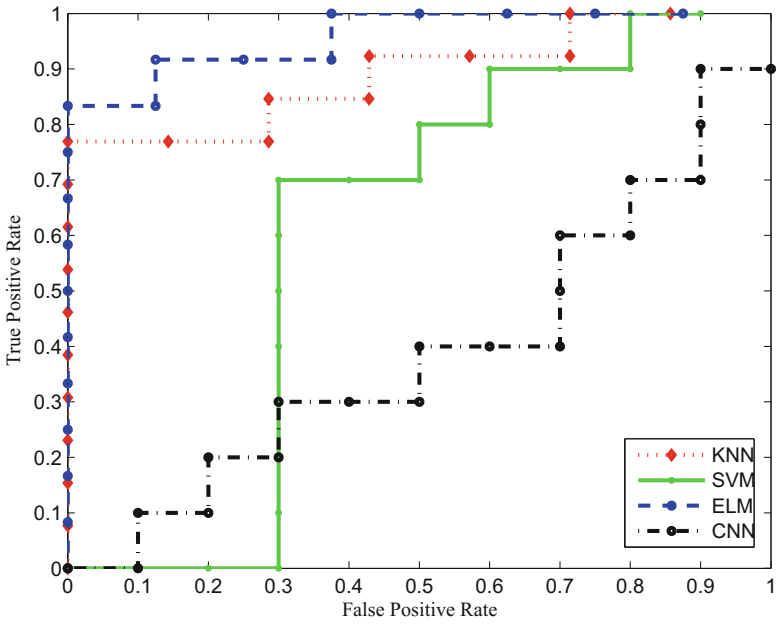

Fig. 3. ROC of different methods (sample number is 480).

Table 2. AUC of different methods

Samples number	KNN	SVM	CNN	ELM
480	0.747	0.500	0.390	0.833

5 Conclusions

In this work, a novel image processing application is developed to detect ship in SAR images. After the image processing and the feature extraction, ELM based classification is applied to detect ships in SAR images. Compared with other learning based classifiers, the proposed ELM based method can obtain better performance to detect ship. The proposed extreme learning machine based method is potential for object detection in other SAR applications. Research of the more efficient ELM based method with automatic feature selection will be studied in the future.

Acknowledgments. This work was partly supported by National Natural Science Foundation of China (No. 61371045).

References

1. Wang, S., Yang, S., Feng, Z., et al.: Fast ship detection of synthetic aperture radar images via multi-view features and clustering. In: International Joint Conference on Neural Networks, pp. 404–410 (2014)
2. Yang, G., Yu, J., Xiao, C., et al.: Ship wake detection for SAR images with complex backgrounds based on morphological dictionary learning. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1896–1900. IEEE (2014)
3. Selvi, M.U., Kumar, S.S.: Sea object detection using shape and hybrid color texture classification. Commun. Comput. Inf. Sci. **204**, 19–31 (2011)
4. Martan-De-Nicols, J., Mata-Moya, D., Jarabo-Amores, M.P., et al.: Neural network based solutions for ship detection in SAR images. In: International Conference on Digital Signal Processing, pp. 1–6. IEEE (2013)
5. Khesali, E., Enayati, H., Modiri, M., et al.: Automatic ship detection in single-pol SAR images using texture features in artificial neural networks. Int. Archiv. Photogrammetry Remote Sens. **XL-1-W5**, 395–399 (2015)
6. Yang, X., Bi, F., Yu, Y., et al.: An effective false-alarm removal method based on OC-SVM for SAR ship detection. In: IET International Radar Conference, pp. 1–4. IET (2015)
7. Ma, L.: Support tucker machines based marine oil spill detection using SAR images. Indian J. Geo-Marine Sci. **45**, 1445–1449 (2016)
8. Ma, L., Hu, Y., Zhang, Y.: Support tucker machines based bubble defect detection of lithium-ion polymer cell sheets. Eng. Lett. **25**, 46–51 (2017)
9. Schwegmann, C.P., Kleynhans, W., Salmon, B.P., et al.: Very deep learning for ship discrimination in synthetic aperture radar imagery. In: 2016 IEEE International Geoscience and Remote Sensing Symposium, pp. 104–107. IEEE (2016)
10. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**, 489–501 (2006)
11. Huang, G., Huang, G.B., Song, S., You, K.: Trends in extreme learning machines: a review. Neural Netw. **61**, 32–46 (2015)
12. Huang, G.B., Zhou, H., Ding, X., Zhang, R.: Extreme learning machine for regression and multiclass classification. IEEE Trans. Syst. Man Cybern. **2**, 513–529 (2016)
13. Wang, S., Deng, C., Lin, W., Huang, G.B.: NMF-based image quality assessment using extreme learning machine. IEEE Trans. Cybern. 255–258 (2016)
14. Yüksel, T.: Intelligent visual servoing with extreme learning machine and fuzzy logic. Expert Syst. Appl. **47**, 232–243 (2017)
15. Liu, X., Deng, C., Wang, S., Huang, G.B., Zhao, B., Lauren, P.: Fast and accurate spatiotemporal fusion based upon extreme learning machine. IEEE Geosci. Remote Sens. Lett. **13**, 2039–2043 (2016)
16. TerraSAR-X Data Samples. http://www.infoterra.de/free-sample-data
17. Benco, M., Hudec, R., Kamencay, P., et al.: An advanced approach to extraction of colour texture features based on GLCM. Int. J. Adv. Robot. Syst. **11**, article no 104 (2014)