Methods of Image Processing and Computer Vision

Vadim Stary[a] and Tomas Matyska[b]

[a] Department of Air Defence Systems, University of Defence, Brno, Czech Republic, e-mail: vadim.stary@unob.cz
[b] Department of Information Technologies, University of Hradec Králové, Hradec Králové, Czech Republic, e-mail: tomas.matyska@uhk.cz

Abstract

The article contains the overview of the most widely used methods for image processing, brief description and adumbration of possible utilization. In the introduction there are described general assumptions, advantages and disadvantages of each method. Methods are verified on suitable demonstrational tasks. Based on this tasks and results general logical models were created. Tested and described image processing methods are: shape (edge) detection, blobs detection, template matching, feature matching, using linear classifiers and artificial neural networks.

Key words: Image recognition; Image processing, Image analysis, Artificial intelligence, Neural networks

1. Introduction

Nowadays, lot of image processing methods and applications exists. In almost every kind of human activity is possible to find the usage of image analysis, for example in retail stores for automatic check out, in security for people identifying or in military for missile guidance. In general, all these applications use mathematical principles of correlation methods of 2D or 3D variables and create the base of various algorithms for computer aided image processing. The application and method variety makes this subject area very wide. The aim of this article is to analyse the most widespread methods and algorithms and verify them in demonstrational tasks. We can assume, the “boom” of various usages of the image processing methods will continue in the future. As the eminent American physicist Edward Teller published in his book in 1962, “Today’s science is tomorrow’s technology.”[1].

2. Terminology, basic principles and initial conditions

This chapter describes terminology, basic definitions and initial problems for purposes of this article. First important thing is to define meaning of detection, recognition and identification. These terms can be defined thru the example of the aircraft detection by the observer. First step is to detect the presence of the entity (YES or NO) in the area of interest. Second, is to recognize the type of entity (bird, aircraft, balloon, etc.) and the final step, is to do the identification of the type (friendly tactical fighter aircraft F-15E Strike Eagle). For computer image processing is important to digitize the input image data. Conversion is done by the sampling and quantization process, this includes the common problem for all A/D convertors, the finding the optimum between the data rate and the information loss. Type of used information from the image, defines the basic image processing method distribution. The most common is usage the brightness intensity matrix for each channel (colour). For quick and less sophisticated applications only one channel and the monochromatic image are used. Other methods use at least 3 channels, RGB (red, green, blue), BGRA (alpha) with transparency information, HSV (hue, saturation, value - brightness), etc. This information is presented as the number of intensity value of each pixel. Number of bits depends on the requested model colour depth, e.g. 24 bits (True Colour) for RGB (8bits per colour) means 16 777 216 possible colours.

2.1. Difficulties and problems

Image processing algorithms deals with several general problems. These are – rotation, image size changes, optical deformation or inconsistent external conditions (light intensity or light reflection). Applications variety disallows the universal usage of image processing method. Every application defines different requirements and it might not be suitable for some of the methods. The stress is put on the definition of requirements and priority for suggested usage.

3. Evaluated methods

3.1. Shape (edge) detection

These methods analyse characteristic object profile. In most cases, the input image is monochromatic. Base preprocessing includes edge detection and Hough transformation to find the “imperfect” edges and classified it to the class of shapes (line, circle, etc.). More difficult shapes can be detected with usage of hash table [2]. Output lines can be classified thru the decision tree depending on the application. Demonstrational task TIC-TAC-TOE game was based on this method. The goal was to detect the cross and the circle in a 3x3 square field. Image source is local or IP camera, with resolution VGA and above. Demonstrational task GUI is on Fig 1, and the functional logical algorithm is on Fig 2.

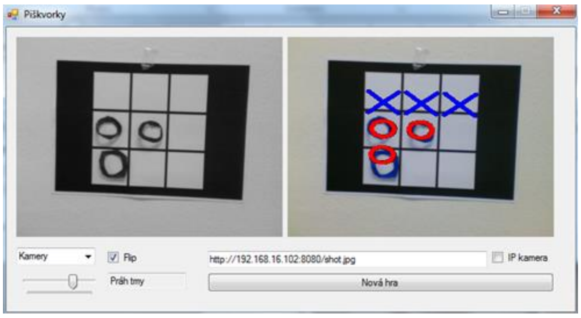


Fig. 1. TIC-TAC-TOE game GUI

Example was created in Visual Studio 2013 by C# programming language.

GUI description:

- Left image – input from the selected camera, converted into the 1 channel (monochromatic image) and the resolution is decrease to the one half.
- Right image – output image after move projection. This image is enabled when the external light conditions are sufficient.
- Scroll menu – input camera selection
- Checkbox Flip – horizontal image inversion
- Slider – brightness level settings
- Input text field – external image source address.
- Checkbox IP camera – IP camera usage
- Button new game – game reset.

Algorithm functionality is shown in the figure 2. Image pre-processing consists of grayscale conversion, resolution decrease, histogram optimization for light conditions and Canny edge detection technique [3].

Canny edge detector consists of Gauss filter for noise reduction and for every pixel, vector gradient of image intensity function, is calculated using Sobel operator [4]. Final image consists of pixels with maximum vector gradient depending on hysteresis level.

Play field detection in pre-processed image is done by finding continuous edges, which are approximated as curves. Among all, the 4 edge curves (squares) are kept, the rest are filtered. Remain squares are transform into the ideal square form (optical deformation elimination) and evaluated whether they fulfil the play field specification (light square with dark edge).

Circle detection is based on Hough transform, where parametric description is (1).

$$x = x_0 + r \cdot \cos(\theta)$$
$$y = y_0 + r \cdot \sin(\theta) \tag{1}$$

Final step is to project the detected situation into the image and make counter move (simple decision-making algorithm is implemented). Image projection is reverse process to a play field transformation. Crosses and circles symbols are insert into the empty squares and whole image is "deformed" into a real perspective by a transform matrix and the final image is projected over the initial image (right image in Fig. 1).

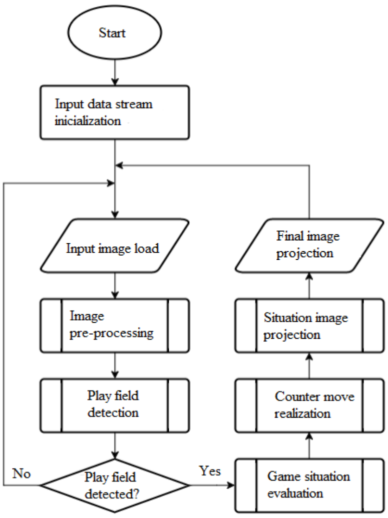


Fig. 2. TIC-TAC-TOE functionality algorithm

3.2. *Blob detection*

This method is based on image distribution into the regions (blobs) in which pixels has similar attributions (brightness, colour, etc.). Blob detection method is very fast, invariant to deformation and rotation. Disadvantage is external light condition dependence. This dependence can be corrected by brightness level settings. Other disadvantage is an image distribution into logical segments (with similar colour) not into real objects. Heterogeneous object can be detected as a multiple objects.

Detection is principle is mostly based on one channel monochromatic image, where different level of brightness are detected or HSV 3 channel colour dimensions. In the figure 3, there is an example of blob detection algorithm usage for "skin colour" detection. In the right side of the figure, there are sliders for HSV level settings for colour definition.

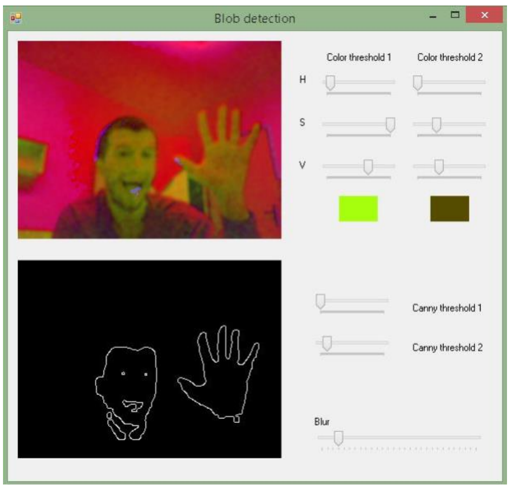


Fig. 3. Blob detection example

3.3. *Template matching*

This is one of “traditional” methods, which represents mathematical principles of two dimensions correlation analysis. In general, this method compares and evaluates the image with required template. For every pixel, the degree of similarity is calculated based on ZNCC (zero mean normalized cross correlation) or NCC (normalized cross correlation) algorithm. This method is robust and resistant to a brightness variance; nevertheless the requested computing power is high and it is not appropriate for real time applications. Nowadays a modification of this method is evolved for detection acceleration. For example BPC (bounded partial correlation) or ZBPC (zero partial correlation) uses Cauchy-Schwarz inequality for the obvious discrepant position elimination [5].

3.4. *Feature matching*

In contrast of template matching, this method is invariant, principle is based on finding of “key points” of image and their pairing with requested object. Feature matching method can be divided into the three steps, detection, description and matching. In general, this method is very fast, robust and optical deformation resistant. Disadvantage is a “problem” with multiple object detection, where algorithm usually matches the best candidate for pairing.

Several methods of detection are used:

- BRISK (Binary Robust Invariant Scalable Keypoints) – offers robust detection of binary invariant points.
- SURF (Speeded Up Robust Features) – uses own SURF descriptor.
- FAST (Features from Accelerated Segment Test)
- AGAST (Adaptive and Generic Corner Detection Based on the Accelerated Segment Test) – generates decision trees for appropriate descriptor (e.g. FREAK – Fast Retina Keypoint).
- Other

Descriptors provide key point definitions. List of important descriptors:

- SURF descriptor – describes intensity distribution close to the key point.
- SIFT (Scale Invariant Feature Transform) – describes key points by an image gradient vector.
- FAST, BRIEF (Binary Robust Independent Elementary Features) with ORB (Oriented FAST and Rotated BRIEF) algorithms are at two orders of magnitude faster than SIFT [6]
- FREAK (Fast Retina Keypoint) – is based on human eye analogy.

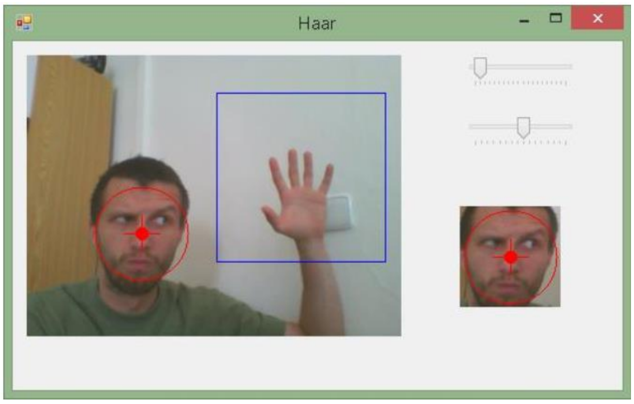
Examples of matchers:

- FLANN (Fast Library for Approximate Nearest Neighbours) – includes several algorithms and system for autonomous selection of optimal one based on input image. It is included in OpenCV framework.
- Brute Force matcher - takes the descriptor of one feature in first set and is matched with all other features in second set using some distance calculation. And the closest one is returned. It is included in OpenCV framework

3.5. *Cascade linear classifier*

Cascade linear classifier method was published in 2001 by Paul Viola and Michael Jones published in [7]. Original purpose was for face recognition, but it can be used for other applications. Method is based on HLF (Haar-like features) which issues with general face structure (light and dark areas). These areas are tested to corresponding to different HLF. Value of each HLF is difference of number of pixels under the white area and number of pixels under the black area. Because of effectivity, AdaBoost algorithm for choosing HLF is used. [8]. Weak classifiers are composed into a cascade structure and create strong classifiers for learning cycle. For every searched object (face, eye, smile…) is important to define weak and strong classifiers. Many of them are available for free on the internet in XML form. As an example, there is a simple application for face and palm detection, see Fig. 4. In the Fig. 4, the position of face is bordered with red circle and the palm position with the blue square. Bars on the right regulate the sensitivity level (scale factor and number of neighbour estimation confirms). This method is widely used for automatic focus and face detection in cameras and mobile phones. Modification of this method in cooperation with depth camera is also used in Microsoft Kinect sensor for Xbox 360 and Xbox One.

Fig. 3. Linear classifiers example



3.6. *Artificial neural networks*

One of possible usage of artificial neural networks is also image processing. In contrast of previous methods, this one can be used also for image synthesis and due to its principles can be apply in real time applications. Requested computing power depends on number of input neurons (image size), number of hidden neurons and layers, kind of activation function and way of learning process functions. Number of output layer neurons is usually equal of number of requested states. For example for number detection 0 to 9, output layer should have 10 neurons, in binary code 4 neurons. For image processing can be used genetic or backpropagation algorithms. Quality of achieving a solution is defined by a fitness function. This function is a result of the learning process of neural network. Learning process (supervised, unsupervised) is based on training examples which consists of learning, validation and evaluation parts. Training serves for setting weights of synapsis, validation part stops the learning process in case of overfitting and evaluation part serves for neural network testing.

For image processing the MNIST (Mixed National Institute of Standards and Technology) training database with 10 000 handwritten digits is often used (record is 9979 correct digits identification [9]).

As an example, the demonstrational task for artificial neural network for handwrite numbers 0 to 9 was created.

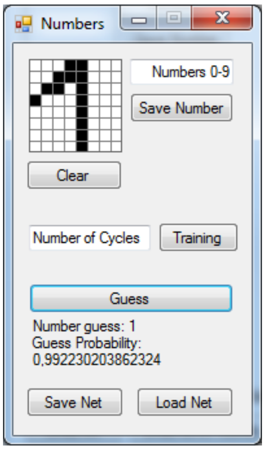


Fig. 5 GUI of number ID example

Usage example was created by C# in MS Visual Studio 2013. For universal usage the library with feedforward neural network was made. It includes definitions of structures, layers, neurons and genetic algorithm and backpropagation methods. GUI serves as a tool for creation training samples, saving and loading networks and for input number guessing.

Input is represent by a square 8x8 square field where you can draw a number. Algorithm was trained by 50 000 and 150 000 training cycles on 200 handwrite samples. Training sample is in text file as a string of 64 bites (0 for white pixel, 1 for black pixel). Training cycles is based on backward propagation of errors, and sets parameters (synaptic weight) of neural network. Probability of right estimation of number is calculated from value of output neuron divide with sum of all other output neurons values. Trained network reached success number estimation more than 99 %.

Quality and success rate of image recognition is given by the structure of artificial neural network, by the range of training samples and cycles. This method can be used in situations with frequent changes.

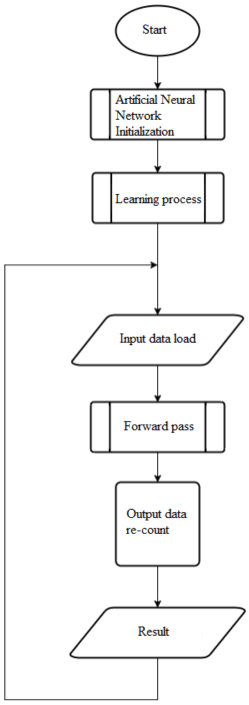


Fig. 6. Logical model of artificial neural network example

4. Conclusion

This article describes the most widely used methods of image processing. The main intent was to create a list of methods and algorithms, verify their advantages and disadvantages and prepare a base for the possible future usage. For verification, demonstrational tasks were created – the TIC-TAC-TOE game and the number identifier. For better understanding the logical models of the examples are also included. Application usage, especially in military or security, seems to be radar, camera targeting, flight navigation, technical diagnostics and steganography or face recognition. All of these applications already exists, but still needs an improvement and optimization. This article should bring a basic perspective in the field of image processing.

References

[1] TELLER, E., The Legacy of Hiroshima. Michigan: Doubleday, 1962, 146 p.

[2] HECKER, Y. C. and. BOLLE, R. M., On geo-metric hashing and the generalized Hough transform, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 24, No. 9, 1994, pp. 1328-1338. doi: 10.1109/21.310509

[3] CANNY, J. A Computational Approach To Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, 1986, pp. 679-698.

[4] JÄHNE, B. and SCHARR, H. and KÖRKEL, S., Principles of filter design, In *Handbook of Computer Vision and Applications*. Academic Press, 1999

[5] STEFANO, I. and MATTOCCIA, L. S. and TOMBARI, F., Pattern Recognition Letters. In: *ZNCC-based template matching using bounded partial correlation* [online]. 3. may. 2005 [cite.2016/11/1].Available: http://labvisione.deis.unibo.it/fede/papers/jprl05.pdf

[6] RUBLEE, Ethan, et al. ORB: An efficient alternative to SIFT or SURF. In: 2011 *International conference on computer vision*. IEEE, 2011. p. 2564-2571.

[7] VIOLA, P. and JONES. M., 2001. Rapid Object Detection using a Boosted Cascade of Simple Features, In: 2001 *Conference of computer vision and pattern recognition* [online]. [cite 2016/11/1]. Available in: https://www.cs.cmu.edu/~efros/courses/LBMV07/Papers/viola-cvpr-01.pdf

[8] WANG, R., AdaBoost for Feature Selection, Classification and Its Relation with SVM, *A Review*, *Physics Procedia*, Volume 25, 2012, Pages 800-807, ISSN 1875-3892, http://dx.doi.org/10.1016/j.phpro.2012.03.160.

[9] NIELSEN, M. Neural Networks and Deep Learning. Using neural nets to recognize handwritten digits [online]. 2016, version 1 [cite: 2016/11/1] Available http://neuralnetworksanddeeplearning.com/chap1.html