NEURAL NETWORK BASED OBJECT RECOGNITION USING COLOR BLOCK MATCHING

Kay Boehnke, Marius Otesteanu, Philipp Roebrock
Politehnica University of Timisoara
Communications Department
Bd. Vasile Parvan, no. 2
300223 Timisoara
Romania
Kay.Boehnke@gmail.com
Marius.Otesteanu@etc.utt.ro
Philipp.Roebrock@informatik.fh-gelsenkirchen.de

Wolfgang Winkler, Werner Neddermeyer,
University of Applied Science
Faculty of Computer Science
Neidenburger Str. 43
45877 Gelsenkirchen
Germany
Wolfgang.Winkler@informatik.fh-gelsenkirchen.de
Werner.Neddermeyer@informatik.fh-gelsenkirchen.de

ABSTRACT
This paper describes an algorithm for the fast classification of color regions in pictures with the help of neural networks. The algorithm divides the picture into discrete blocks which are analyzed independently. Average values of the three color channels are extracted from the block and classified by a neural network. Classification is made by a modified backpropagation network. After the classification a "best fit" search is used to find the best matching block. Depending on the block size it is possible to find and classify a learned color in a picture with the help of this sequential search algorithm. The proposed classification meets real time requirements in an industrial application. Therefore some optimizations were necessary which are explained in this paper. Furthermore, different pre-processing and segmentation algorithms and color space transformations were analyzed and tested regarding their effectiveness for fast image processing.

KEY WORDS
Color classification, neural networks, real-time block matching

1. Introduction
Colors signals arise in everyday situations, especially in the flora and fauna colors are used as warning signal but also as an attractant. Colors detection is a natural component in our human visual system.

Unfortunately color plays only a minor roll in object recognition and classification in industrial applications. In the 80th and beginning of the 90th of the last century color image analysis was restricted by high costs of adequate industrial color cameras. In the last decade the boom in the consumer market causes a big drop in prices of color video cameras. Prices of expensive industrial framegrabber boards profit by growing distribution of hardware-related consumer TV-cards. More and more color-based research activities are transferred in industrial applications.

To separate regions of colors in an image many approaches use threshold-based segmentation of provided RGB-values. This neural network approach offers the possibility to define "multidimensional" thresholds. Similar colors with significant differences in the RGB-color space can be easily merged in one class. The use of learning systems counters changes in illumination effectively. This solution is implemented to meet real time requirements enabled by parallel computing.

The next chapter provide a short overview of the used algorithms. The system is described in detail in the third chapter. Chapter 4 shows the potential of the proposed system in an industrial application. Conclusion and future aspects are provided in the last chapter.

2. Related work
The approach described here is based on the direct classification of color values of image pixels in the Red-Green-Blue (RGB) color space. In Haberaecker[1] a neural network with three input neurons for every color channel and na output neurons are used. na is the number of colors to be classified. Color classes are created manually. RGB values of corresponding colors are used for the training of the neural network [2][3].

Similar to [1] our approach uses the channel information from color images, but only three input neurons altogether, one for every color channel. According to Batlle et al. [4] we use a bottom-up approach extracting color regions from the image. Therefore, the picture is divided into blocks[5] like block matching algorithms used in stereo vision. This enables the analysis of local features in the pixel neighbourhood [6].

Beside color image segmentation [7][8] neural networks are used in color reduction in digital color images as well.
Papamarkos et al.[9] propose a technique to reduce color information in digital images with the help of a combination of different neural networks.

In contrast to most of the previous approaches the described technique in this paper focuses on fast computation for real time applications and uses a neural network with its advantages[3].

3. System description

A digital color picture consists of pixels which are represented as a vector of three color values. Mathematically, a picture can be described by the equation

$$ S = s(x, y, n) $$

It is possible to select the color channel with the parameter \( n = \{0,1\ldots(N-1)\} \). In an RGB color image there exist 3 channels for the particular colors red, green and blue. In the following these images are called RGB pictures. The values \( s(x, y, n) \) in an RGB picture are defined as:

$$
\begin{align*}
  s(x, y, 0) &= R \\
  s(x, y, 1) &= G \\
  s(x, y, 2) &= B
\end{align*}
$$

In a digital 24bit- RGB picture, the values of each channel are between 0 and 255. In the picture the color impression arises by additive mixing. For every color in an RGB picture, a point in a coordinate system can be assumed.

In [1] this results in a feature space for the classification of a color. In this feature space a certain color represents a cluster with a certain distance to another color cluster. The feature space is equivalent to the RGB color space. The neural network can also be used for other non-uniform or uniform color spaces[10]. Converting RGB- images to other color spaces requires complex pre-processing computations and does not add any information.

To improve the classification and reduce the execution speed the algorithm is equipped with a block processing. The creation of blocks in the picture is made discretely. A picture with the height h and the width w is divided into r partial sub- pictures. It applies:

$$
 r = \frac{h \times w}{n \times n}
$$

where n is the edge length of a block. In the classification there problems arise concerning the analysis of pixel which are close to the right edge and the lower end of the picture, because no full blocks can be generated in these areas anymore. Assuming that the block size is small in relation to the picture size and in order to avoid special time intensive treatments these pixels of the marginal areas are not considered in the experiments.

Features can be extracted from a block consisting of n*n pixel. The feature extraction plays a central role for the color classification. The values of the several color channels of each pixel are regarded as a starting point for the analysis. In the simplest case these values can be used directly for the classification, as described in the book of Haberaecker[1]. The feature extraction in the blocks of a picture is made by calculating the average value in every color channel with the formula.

$$
 m_i = \frac{1}{N} \sum_x \sum_y s(x, y, i) \quad \text{with } i = \{0,1,2\} \quad (4)
$$

The three average values of the block now represent a feature vector:

$$
 M = (R, G, B) \quad (5)
$$

as a result of the average calculation which is calculated for each block. Regarding the clusters in the feature space this means a reduction of its size in the space and with that a reduction of the noise in the image. At the same time, the average values are normalized to the interval \([0..1]\) for noise reduction and input neuron normalization. For the classification of the features a backpropagation network is chosen here. The number of input values is fixed with the number of color channels. The topology was initially fixed with one hidden layer. The Cichocki/Unbehauen's[2] term:

$$
 2^{n_{\text{min}}} \geq z \Rightarrow n_{\text{min}} \geq \log_2(z) \quad (6)
$$

is necessary to automatically increase the number of hidden neurons. The number of hidden neurons \( n_{\text{min}} \) is directly adapted to the number of training patterns \( z \). The maximum number is limited to 16 in the implementation, because it is not appropriated to train a neural net with 2\(^{16}\) training pattern or more. Too many hidden neuron causes the network to overfit, so equation 6 provide a good initial value. The number of outputs of the neural network depends on the number of classes (colors) which have to be learned. The transfer function of the neurons in the hidden layer and output layer is fixed as a sigmoid function.

$$
 f(\text{net}) = \frac{1}{1 + e^{-\text{net}}} \quad (7)
$$

The network is trained by a normal backpropagation method with momentum factor[2]. At the end of the classification the neural network shows a resulting output vector in the form:

$$
 A = \{a_1, a_2, \ldots, a_n\} \quad (8)
$$
where \( n \) is the maximum number of outputs/colors.

For each color \( x \) results a target vector in dimension \( n \) from the canonical unity vector.

\[
t_x(i) = \begin{cases} 
1 & \text{if } x = i \\
0 & \text{else} 
\end{cases} \quad \text{for } i = (1, 2, \ldots, n) \tag{9}
\]

\[
\begin{bmatrix}
100 \\
010 \\
001 \\
\ldots \\
\end{bmatrix}
\]

Assuming that the trained network calculates a similar output vector \( T \) for the color \( x \) the equation:

\[
d(T, A) = \sqrt{\sum_{k=0}^{N-1} (T_{i,k} - A_{i,k})^2} \tag{10}
\]

is basically used for the determination of the weight changes during the training of the network. This Euclidean calculation of the network error can also be used for the determination of the quality of the classification. However, also other distance measurements are possible for the determination of the quality. On the one hand, the Euclidean distance of the vectors just mentioned is commonly used, but on the other hand, the city-block distance:

\[
d(T, A) = \sum_{k=0}^{N-1} |T_{i,k} - A_{i,k}| \tag{11}
\]

which has a shorter calculation time is also possible for the distance calculation.

Therefore in the execution the following equation for the output vector:

\[ F = \min(T - A_x) \quad \text{with } T = \{t_1(i), t_2(i), \ldots, t_n(i)\} \tag{12} \]

is valid for the calculation of quality. The error vector \( F \) changes into a scalar quality metric \( q \) by applying:

\[ q = 100 \left(1 - \frac{\sum_{x=1}^{n} F(x)}{n}\right) \tag{13} \]

The reason for the usage of this distance is the reduction of computational costs. The required execution time with an optimized implementation is reduced by 70% compared to the Euclidean distance. During the search the algorithm goes sequentially through all blocks of the picture and always gets a quality metric as result. The block coordinates with the maximum value of quality are determined as final result of the classification result.

\[
\begin{align*}
\text{Table 1 Results for generalization} \\
\hline
\text{False recognition} & 5x5 \text{ Block} & 10x10 \text{ Block} \\
0\% & 0\% \\
\text{No recognition(green)} & 0\% & 0\% \\
\text{No recognition(yellow)} & 0\% & 1.8\% \\
\text{No recognition(blue)} & 0\% & 1.8\% \\
\text{No recognition(error)} & 0\% & 0\% \\
\hline
\end{align*}
\]

4. Experiments

At the industrial application described above three colors of markings at engine parts shall be differentiated (Picture 1). Here, background colors are considered and drawn into the right diagram as black dots. All green pixels in the green rectangle of the left picture are drawn green in the diagram on the right.

For the verification of the approach some classifications were analyzed in Matlab[11]. The training set consists of the manually selected regions of different pictures of the engine parts (Marking in Picture 1 in the middle of the triangle shape of the engine part). Due to this detailed information different tints and different illuminations can be integrated in the training. Altogether, the training set contains 797 samples. One part of 258 was defined as blue, 315 as yellow and 224 as green. The stop criteria were fixed on a residual error of 0.001 and on a maximum number of 1000 epochs. The test of the generalization ability was carried out with the help of the Leave-Out strategy. 100 unknown pictures of blue engine parts and 53 pictures of yellow highlighted engine parts were used for testing. Concerning the green marks there was not enough pictorial material available. Because of this these 6 pictures became resubstituted. None of the 159 test pictures was classified as wrong. At 11 pictures no colors were recognized or these were assigned to the rejection class. With the help of manual analysis of these five pictures it was realized that their declaration as "not OK" was completely correct because no markings were made in the industrial method at these engine parts. At the remaining 6 pictures the marking was smaller than the block size of 10x10 pixels. Therefore the quality has slipped under the threshold of 90 percent. These pictures could be classified correctly in the following test with a smaller block size (5x5). The analyze time with this block size of 10x10 pixel was 80ms in average for one image with the PC mentioned above. The test with a block size of 5x5 arose an average time of 136ms, but clearly improved the classification results shown in table 1.
The results of the generalization ability were also successful at further tests and reached recognition rates of over 99 percent. At the implementation the search of the color in the picture results in a fixed pixel because of the "Best fit" search. To this each classification output of a block is compared to all other combinations to display the best matching position.

It was noticed during the tests that with very small block sizes also points outside the marking were declared as "Best- Fit". In some sample images particularly blue tones were reflected on the metal of the engine parts. Picture 2 impressively shows the problems of the choice of smaller block sizes.

The numbers at the engine parts located above the marking have blue areas which possibly match the learned color even better than the actual blue marking. Additionally, small reflective surfaces and other markings cause problems concerning the classification. The solution of the problem is very simple: The block size must be adapted to the color region which has to be found. Other similar color regions have to be removed by the limitation of the search area in the picture. In addition these simple parameter changes lead to a faster execution of the algorithm.

Optimizations of the code were analyzed in order to decrease the execution times of the algorithm. These optimizations in machine code achieved great reductions concerning the execution time. With the reimplementation of mathematical standard functions and linearized calculations over 90 percent time-saving for an initial implementation was achieved. At the same time, there was no loss of accuracy detected.

5. Conclusion

A system for fast color classification is presented using an optimized neural network. In addition to the described application in chapter this algorithm could be used as a digital filter or in image segmentation task. The main advantage of the algorithm is its generality, so the solution will achieve good results in many kinds of future applications. Future work will focus on improvements like automatic adjustment of the block size. The unusual combination of block matching metrics and neural networks offers potential in future developments.

References


