Pixel Classification for Skin Detection in Color Images

Bartosz Binias, Mariusz Frąckiewicz, Krzysztof Jaskot and Henryk Palus

1 Introduction

Segmentation of color image regions that contain skin pixels is a very important and challenging task of modern image processing. The general objective of described problem is to return the output image, whose every pixel was classified as either representing skin or not [1, 2]. Such information can be then used in various applications in computer vision [3, 4]. Important and interesting tasks, where skin segmentation is required, are automatic face [5–7] or gesture detection [8–10]. Additionally, many of the most effective filters of adult-only content are based on the information from the segmented skin regions [11, 12]. Image coding using regions of interest as presented in [13], is another very important example of the application of skin segmentation algorithms.

The most common approach to skin detection problem is pixel-wise, color-based classification [4]. In such methods each pixel is being classified independently from its neighbours only on the basis of its color features [14, 15]. Such discrimination between skin and non-skin pixels can be performed using skin color model represented either as a set of rules or thresholds or derived from used machine learning algorithm [4]. However, it must be noted that basing solely on the color information may not be sufficient for the task. It is a well known fact, that most popular color spaces such as the *RGB* and *HSV* or perceptually uniform *CIELab* suffer from many shortcomings (for more details see [16]). In order to improve the quality of pixel-based skin detection many approaches have been proposed. Among these worth mentioning are texture-based methods, adaptation techniques and spatial analysis. A very detailed description of this techniques can be found in [4].

In this research a direct, pixel-based approach was chosen. The main purpose was to evaluate the performance in this specific task of two very popular machine

B. Binias (✉) · M. Frąckiewicz · K. Jaskot · H. Palus
Silesian University of Technology, Akademicka 16,, 44-100Gliwice, Poland
e-mail: Bartosz.Binias@polsl.pl

A. Nawrat et al. (eds.), *Advanced Technologies in Practical Applications for National Security*, Studies in Systems, Decision and Control 106,
https://doi.org/10.1007/978-3-319-64674-9_6

learning classifiers: Regularized Logistic Regression and Artificial Neural Network with Regularization trained with Backpropagation [17]. The research focused mostly on the aspect of model's error evaluation and parameter tuning. Developing new algorithms [18–21] and computer simulation [22] are classical means to achieve progress of the technology.

2 Data Description

Data used in this research was 'Skin Segmentation Dataset' provided for the UCI Machine Learning Repository [23]. The dataset consists of 50859 examples marked as skin samples and 194198 non-skin samples. Available features are pixel's values in B, G and R channels, coded with 8 bits. The skin dataset was collected by random sampling of these values from images of various gender, age and race groups obtained from Color FERET Image Database and PAL Face Database from Productive Aging Laboratory.

All 245057 available examples were randomly divided into three subsets. The training set was created by randomly selecting 60% of all skin samples and 60% of samples from the other class. That way an original proportion between both classes has been sustained. Analogously, the cross-validation and test sets were separated from the remaining examples. Each of these consisted of 20% of all skin and non-skin examples, sampled without repetition.

3 Description of Algorithms

3.1 Logistic Regression with Regularization

The goal of the regression is finding a set of parameters $\Theta \in \mathbb{R}^{n+1}$ (where n goes for the number of features and additional dimensionality stems from the bias term) that minimizes cost function presented in Eq. 1 [24]:

$$J(\Theta) = \frac{1}{M} \sum_{i=1}^{M} \left(h_\Theta(x^{(i)}) - y^{(i)} \right)^2 \tag{1}$$

where M is the total number of training examples, defined by input variables (features) $x \in \mathbb{R}^{M\times(n+1)}$ and output variables $y \in \mathbb{R}^{n+1}$ (labels). Because of the dimensionality of a single example $x^{(i)} \in \mathbb{R}^{n+1}$ the vectorized notation of the hypothesis $h_\Theta(x^{(i)})$ can be written as follows (we assume that $x_0 = 1$ for each example):

$$h_\Theta(x^{(i)}) = \Theta_0 x_0 + \Theta_1 x_1 + \cdots + \Theta_n x_n = \Theta^T x^{(i)} \tag{2}$$

For the binary classification task the preferred output of the hypothesis function would be either 0 or 1. To enable that a slight modification of the hypothesis function is required. For that a sigmoid (logistic) function is used. The improved hypothesis is presented in Eq. 3:

$$h_{\Theta}(x) = \frac{1}{1 + e^{-\Theta^T x}} \tag{3}$$

In order to assign bigger penalization to predictions that differ highly from the required output hypothesis $h_{\Theta}(x)$ should be additionally logarithmized. To avoid problems with algorithm overfitting the training data the regularization of Θ parameters (apart from bias-related Θ_0) [24] was introduced in the form of λ multiplier. The final form of the minimized cost function of the Regularized Logistic Regression method used for the binary classification is presented in Eq. 4:

$$J(\Theta) = -\Big[\frac{1}{M}\sum_{i=1}^{M}\big(y^{(i)}log\big(h_{\Theta}(x^{(i)})\big) + (1 - y^{(i)})log\big(1 - h_{\Theta}(x^{(i)})\big)\big)\Big] \tag{4}$$

Finding optimal parameters Θ can be performed iteratively with the use of gradient-based numerical optimization techniques such as Gradient Descent or Conjugate Gradient. For such methods to work the derivative of cost function with respect to each parameter must be calculated and provided. However, it must be noted that the Θ_0 parameter should not be regularized. Therefore, the rule for upgrading this parameter in iteration $p + 1$ presented in Eq. 5 does not take the regularization term into account.

$$\Theta_0^{(p+1)} = \Theta_0^{(p)} - \alpha\frac{1}{M}\sum_{i=1}^{M}\Big(h_{\Theta}(x^{(i)}) - y^{(i)}\Big)x_0^{(i)} \tag{5}$$

For other parameters Θ_j, where $j = \{1, 2, \ldots, n\}$, the formula for finding their improved values in new iteration $p + 1$ using Gradeint Descent based method with step α is presented in Eq. 6:

$$\Theta_j^{(p+1)} = \Theta_j^{(p)} - \alpha\Big[\frac{1}{M}\sum_{i=1}^{M}\Big(h_{\Theta}(x^{(i)}) - y^{(i)}\Big)x_j^{(i)} + \frac{\lambda}{M}\Theta_j^{(p)}\Big] \tag{6}$$

3.2 Artificial Neural Network Model with Regularization

A typical Artificial Neural Network consists of structures known as layers [24]. Among them distinguished is the *input layer* and the *output layer*. Other ones are remotely referred to as *hidden layers*. Each layer is constructed of basic calculation units called *neurons*. If layer j has s_j neurons and layer $j + 1$ has s_{j+1} units then the weights of connections between neurons in particular layers $\Theta^{(j)} \in \mathbb{R}^{s_{j+1}\times(s_j+1)}$. The process of Neural Network's output calculation is known as the *Forward Propagation* where the vector of neuron's activations in layer j $a^{(j)} \in \mathbb{R}^{s_j+1}$ (with added bias) is being calculated as was presented in Eq. 7. In first, input layer we treat inputs as activation, as in $a^{(1)} = x^{(i)}$.

$$a^{(j+1)} = g(\Theta^{(j)} a^{(j)}) \tag{7}$$

In proposed model a sigmoid activation function $g(\Theta, a)$ was used for each neuron (Eq. 8).

$$g(\Theta, a) = \frac{1}{1 + e^{-\Theta^{(j)T} a^{(j)}}} \tag{8}$$

The cost function minimized by the Artificial Neural Network with Regularization of weights is presented in Eq. 9, where K stands for the number of classes, L is the total number of layers in the network [24].

$$J(\Theta) = -\Big[\frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{K} \big(y_k^{(i)} log\big(h_\Theta(x^{(i)})\big)_k + (1 - y_k^{(i)}) log\big(1 - h_\Theta(x^{(i)})\big)\big)_k\Big] + J_{reg}(\Theta) \tag{9}$$

where $J_{reg}(\Theta)$ is the regularization term (Eq. 10) [24].

$$J_{reg}(\Theta) = \frac{\lambda}{2M} \sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} \big(\Theta_{ji}^{(l)}\big)^2 \tag{10}$$

The most popular procedure of Artificial Neural Networks training is the *Backpropagation* algorithm. The detailed description of the algorithm can be found in [25].

4 Error Model Evaluation

Before the selection of the best parameters for the method it is important to properly evaluate its error on training and cross-validation datasets. Such effort can help to determine whether the model is capable of explaining the variance in the data properly without overfitting to the training examples. In this research, the base input of classification algorithm's consisted of three features related to each pixel's (example's) value in one of the *RGB* color space channels. All features were scaled to the range [0, 1]. The reason for that operation is the use of the Nonlinear Conjugate Gradient based method for optimization of algorithms' cost functions. Providing that features are on the similar scale improves convergence of this method.

4.1 Learning Curves

In order to determine whether any of the proposed models struggles with high bias or variance problem the adequate learning curves were calculated as the function of classification error's dependence on the number of samples. High bias errors on training and cross-validation sets would converge with the increase of samples provided for training. However, at some point they will both set on a relatively high value. Such behaviour indicates that examined model does not explain the classes sufficiently. Models with high variance are characterized by low error achieved during classification of training set and higher number of misclassified samples in cross-validation set. The reason for that is the overfitting of the model to the training dataset. Formula used for error calculation is presented in Eq. 11.

$$J_{err}(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \tag{11}$$

Learning curves for both models are presented in Figs. 1 and 2. They were calculated on the basis of the performance of classifier trained on the increasing number of $m \leq M$ samples. Because error of testing set classification was measured only on m examples it tends to increase with m. The size of cross-validation dataset remained unchanged during the whole testing.

The learning curves calculated for the Regularized Logistic Regression classifier are presented in Fig. 1. It can be noted that the increase of training samples does not reduce the classification error of cross-validation data. Additionally, both error functions converge to a relatively high value. These characteristics are a clear indicators that the model suffers from high bias problem. This means that it is not capable of properly describing the classes present in the data.

Analysis of the curves presented in Fig. 2 shows that the Artificial Neural Network model does not suffer from high bias or variance. After the sufficient number of training samples is provided it can be observed that errors calculated on training and cross-validation sets converge to common value. This proves that proposed model generalizes well on the distinct features of the data. Additionally, low values of errors acquired for higher numbers of samples indicate that it is able to properly explain the differences between classes.

4.2 Tuning of Regularized Logistic Regression Model

Error analysis for Logistic Regression Model performed in the previous subsection revealed that it suffers from high bias. The best solution of this problem would be to add more features, that would help to better describe the data. In this research an addition of some higher order polynomial features (up to third degree) was proposed. Assuming that the feature vector for i-th example (with n features and unitary

Fig. 1 Learning curve of Regularized Logistic Regression model ($\lambda = 0$)

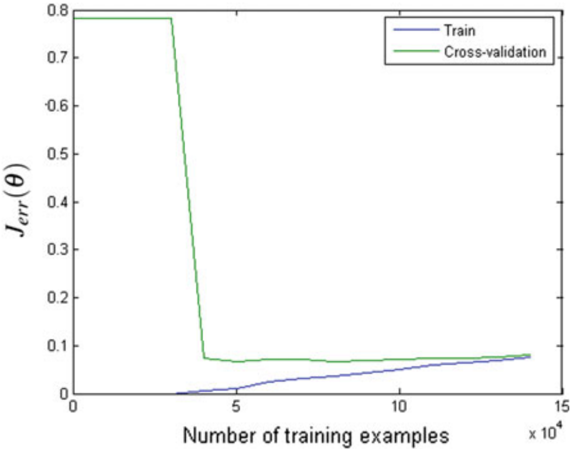
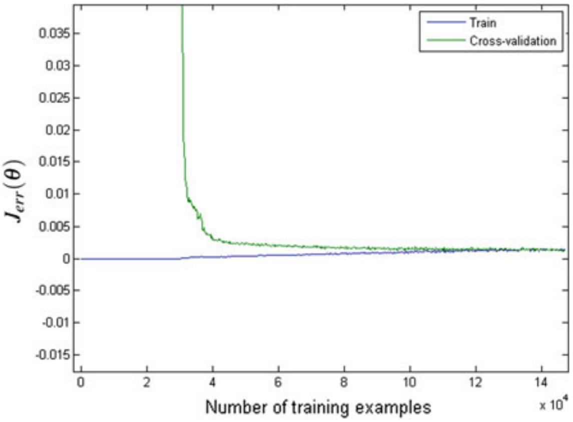


Fig. 2 Learning curve of Artificial Neural Network Model ($\lambda = 0$)



bias x_0) is of form $x^{(i)} = [x_0^{(i)} x_1^{(i)} \dots x_n^{(i)}]^T$, then addition of higher order would mean providing all the combination of features (except for bias) up to required order, i.e. $x_1x_2, x_2^2, x_1x_2x_3, x_3^2x_1$, *etc*. Presented in Fig. 3 are the learning curves calculated for the extended set of features. Achieved results prove that the extension of feature vector is the solution of high bias problem for this case. For the final tuning of the parameter the best weight of the regularization term was selected.

Finally, an additional tuning of the regularization weight was applied. Basing on the results presented in Fig. 4 $\lambda = 3$ was chosen as the best regularization parameter for the classification of testing data.

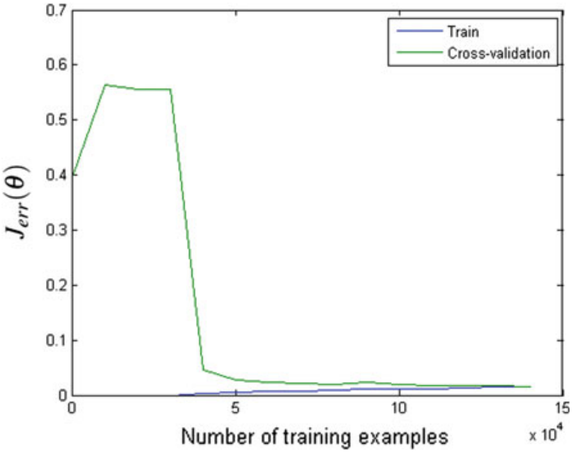


Fig. 3 Learning curve of Regularized Logistic Regression model with polynomial features of up to third order ($\lambda = 0$)

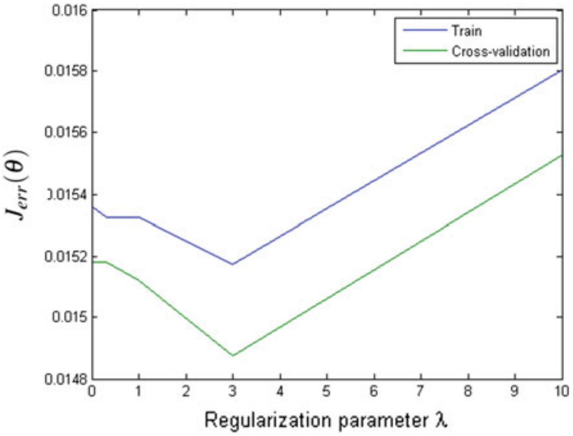


Fig. 4 Validation curve of Regularized Logistic Regression model presenting influence of regularization parameter λ on classification error

4.3 Tuning of Artificial Neural Network Model

The Artificial Neural Network Model trained with Backpropagation did not exhibit any signs of high bias or variance despite it being very basic with only one hidden layer. In order to find the best number of neurons in hidden layer model's classification error was calculated for training and cross-validation datasets with regularization parameter $\lambda = 1$. Thanks to such high regularization weight, achieved results were oriented mostly towards bias, not variance validation. Achieved validation curve is presented in Fig. 5.

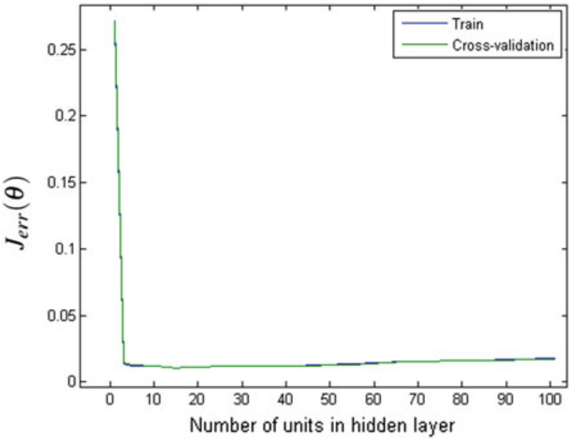


Fig. 5 Validation curve of ANN presenting the influence of hidden layer neurons on classification error ($\lambda = 1$)

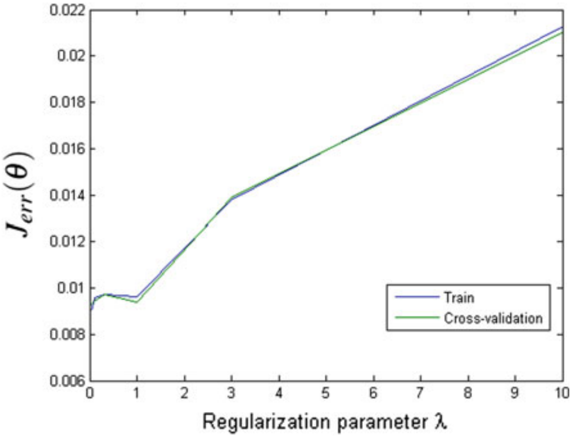


Fig. 6 Validation curve of ANN (with 15 neurons in hidden layer) presenting influence of regularization parameter λ on classification error

According to achieved results the best number of hidden neurons is 15. For that value the additional tuning was performed in order to select appropriate regularization parameter λ. Basing on the results presented in Fig. 6 $\lambda = 0.001$ was chosen as the best regularization parameter for the classification of testing data.

5 Results

Both models with the best settings selected in Sect. 4 were used for the classification of test data. However, it must be noted that after a careful evaluation of both model's errors it was determined that the Artificial Neural Network classifier proved to perform significantly better. For each model a confusion matrix was created on both cross-validation and test dataset. Because in this case proportion between classes is skewed evaluating algorithms only on the basis of their accuracy would not be meaningful. Having that in mind, classifier's precision, recall and specificity were also calculated. Precision refers to a number of True Positive cases over all cases classified as Positive (here as skin samples). To describe the ratio between True Positives and actual Positive examples, the Recall can be used. Specificity is the equivalent of the Recall for the Negative class. In Table 1 presented are results achieved for cross-validation and test sets for both proposed models.

Examination of the results achieved by the Artificial Neural Network model shows that it's performance on the testing and cross-validation datasets is nearly perfect. Very high values of all used measures are the indicators that not only the accuracy of the model is high, but also the general quality of classification. On the other hand, the performance of the Logistic Regression is less stable. Although it achieved comparable accuracy and slightly higher recall, it is significantly less precise. The lacking precision of skin segmentation algorithms is often listed as one of the most important issues of pixel-based detectors.

To additionally test the performance of the Artificial Neural Network model it was tested and visually examined on the real images that did not belong the 'Skin Segmentation Dataset' provided for the UCI Machine Learning Repository. In Figs. 7 and 8 presented are results of segmentation of the images from PUT database. The PUT face database [26] contains high resolution color face images of 100 people acquired on a uniform background under controlled lighting conditions.

Analysis of the segmentation results proves the high quality of the developed classifier. The visual examination of the output image reveals that all the most important information as well as some really deep details of the original image were captured, thus providing a very satisfactory, precise and accurate skin segmentation result.

Table 1 Performance of both proposed classifiers evaluated on cross-validation and test data

Model	Logistic regression		Artificial Neural Network	
Data	CV	Test	CV	Test
Accuracy	0.9849	0.9831	0.9990	0.9987
Precision	0.9321	0.9248	0.9967	0.9947
Recall	1.0000	1.0000	0.9987	0.9989
Specificity	0.9809	0.9787	0.9991	0.9986

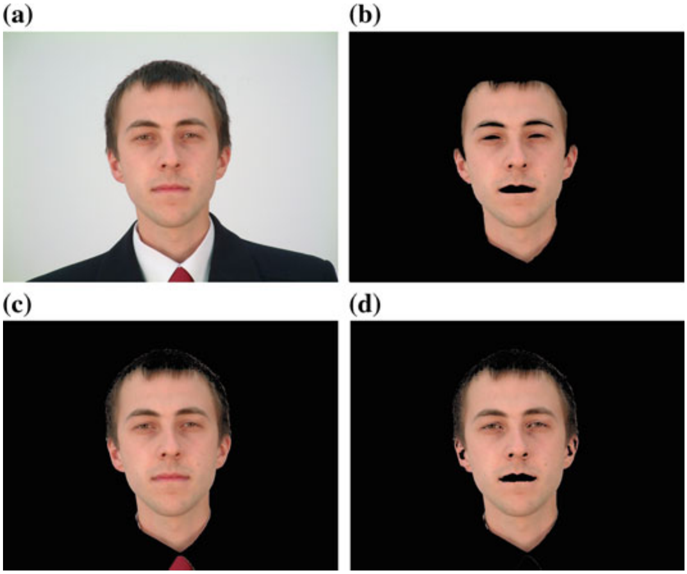


Fig. 7 Results of skin segmentation performed on real image: **a** original image, **b** manually segmented ground truth image, **c** result of segmentation with Regularized Logistic Regression, **d** result of segmentation with Artificial Neural Network

It must be noted that the algorithm allowed the occurrence of some False Positive classification in the right side of the image. The most problematic regions of the image are related to areas where hair connects with skin. Because color features of this regions correspond belong to the developed skin model it is impossible to avoid classifying them as Positive cases. However, such inaccuracies can be easily removed with the use of some image postprocessing techniques mentioned in the Sect. 1 of this paper. To furtherly evaluate a quality of the segmentation the Jaccard similarity coefficient [27] was calculated for both images. The obtained values were 0.897 for Regularized Logistic Regression and 0.920 for Artificial Neural Network model for the first image and respectively 0.943 and 0.948 for the second image. The Jaccard index is a statistic commonly used for the evaluation of the similarity and diversity of two sets. For a full similarity the Jaccard index would take the value of 1. High values achieved for Artificial Neural Network classifier correspond with visual evaluation of the results.

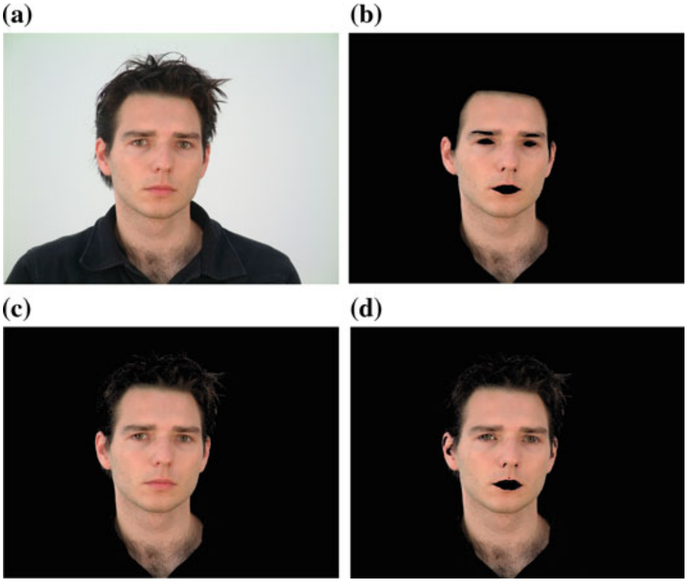


Fig. 8 Results of skin segmentation performed on real image: **a** original image, **b** manually segmented ground truth image, **c** result of segmentation with Regularized Logistic Regression, **d** result of segmentation with Artificial Neural Network

6 Conclusions

In this paper the performance of two prominent classification algorithms was evaluated and compared in the task of skin pixel detection. It was proven that the exploratory data analysis approach can produce some highly satisfying results without the need of referring to some image specific methods, especially during low-level processing.

Achieved results indicate that the Artificial Neural Network is capable of extracting some hidden features and structures in the data. This can be best observed when used classifiers are compared. The Regularized Logistic Regression's performance was poor until some additional features were created and provided as extra inputs to the classifier. At the same time the Artificial Neural Network (with only one hidden layer) performed significantly better using only basic features. This holds an important advantage over approaches where features must be designed by the data scientist during model creation. The reason for that is, that such automatic feature extraction is not constrained by the invention of the designer.

The further improvements to the method can still be applied in order to achieve even better results. An addition of different features derived from other known color spaces could prove to be a valuable extension of the method. Additionally, some advanced postprocessing like texture-based models, adaptation techniques and spatial analysis could be introduced to the proposed algorithm.

Acknowledgements This work was supported by Polish Ministry for Science and Higher Education under internal grant BKM/514/RAu1/2015/t-21 for Institute of Automatic Control, Silesian University of Technology, Gliwice, Poland.

References

1. Kakumanu, P., Makrogiannis, S., & Bourbakis, N. (2007). A survey of skin-color modeling and detection methods. *Pattern Recognition*, *40*(3), 1106–1122.
2. Khan, Rehanullah, Hanbury, Allan, Stöttinger, Julian, & Bais, Abdul. (2012). Color based skin classification. *Pattern Recognition Letters*, *33*(2), 157–163.
3. Phung, S. L., Bouzerdoum, A., & Chai, D. (2005). Skin segmentation using color pixel classification: Analysis and comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(1), 148–154.
4. Kawulok, M., Nalepa, J., & Kawulok, J. (2014). Skin detection and segmentation in color images. In M. Emre Celebi & B. Smolka (Eds.), *Advances in low-level color image processing* (Vol. 11, pp. 329–366). Lecture notes in computational vision and biomechanics. Netherlands: Springer.
5. Kawulok, M. (2005). Application of support vector machines in automatic human face recognition. *Medical Informatics & Technology (MIT)*, *9*, 143–150.
6. Chaves-González, Jose M., Vega-Rodríguez, Miguel A., Gómez-Pulido, Juan A., & Sánchez-Pérez, Juan M. (2010). Detecting skin in face recognition systems: A colour spaces study. *Digital Signal Processing*, *20*(3), 806–823.
7. Hajraoui, Abdellatif, & Sabri, Mohamed. (2014). Face detection algorithm based on skin detection, watershed method and gabor filters. *International Journal of Computer Applications*, *94*(6), 33–39.
8. Kawulok, M. (2008). Dynamic skin detection in color images for sign language recognition. *Proceedings of the ICISP, LNCS*, *5099*, 112–119.
9. Grzejszczak, T., Kawulok, M., & Galuszka, A. (2016). Hand landmarks detection and localization in color images. *Multimedia Tools and Applications, 75*(23), 16363–16387.
10. Daniec, K., Jedrasiak, K., Nawrat, A., & Bereska, D. (2013). Gyro-stabilized platform for multispectral image acquisition. In A. Nawrat & Z. Kuś (Eds.), *Vision based systems for UAV applications* (pp. 115–121). Springer.
11. Zarit, B. D., Super, B. J., & Quek, F. K. H. (1999). Comparison of five color models in skin pixel classification. In *International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 1999. Proceedings* (pp. 58–63). IEEE.
12. Lee, J. S., et al. (2007). Naked image detection based on adaptive and extensible skin color model. *Pattern Recognition*, *40*(8), 2261–2270.
13. Chen, M. J., Chi, M. C., Hsu, C. T., & Chen, J. W. (2003). ROI video coding based on H.263+ with robust skin-color detection technique. *IEEE Transactions on Consumer Electronics*, *49*(3), 724–730.
14. Chai, D., & Bouzerdoum, A. (2000). A Bayesian approach to skin color classification in YCbCr color space. In *TENCON 2000. Proceedings* (Vol. 2, pp. 421–424). IEEE.
15. Palus H. (2006). Color image segmentation: Selected techniques. In R. Lukac & K. N. Plataniotis (Eds.), *Color image processing: Methods and applications* (pp. 103–128). Boca Raton: CRC Press.
16. Palus, H. (1992). Colour spaces in computer vision. *Machine Graphics and Vision*, *1*(3), 543–554.
17. Al-Mohair, K., & Suandi, S. A. (2012). Human skin color detection: A review on neural network perspective. *International Journal of Innovative Computing, Information and Control*, *8*(12), 8115–8131.

18. Świtoński, A., Josiński, H., Jędrasiak, K., Polański, A., & Wojciechowski, K. (2010). Classification of poses and movement phases. *Computer Vision and Graphics*, 193–200.
19. Ryt, A., Sobel, D., Kwiatkowski, J., Domzal, M., Jedrasiak, K., & Nawrat, A. (2014, September). Real-time laser point tracking. In *International Conference on Computer Vision and Graphics* (pp. 542–551). Springer: Cham.
20. Sobel, D., Jędrasiak, K., Daniec, K., Wrona, J., Jurgaś, P., & Nawrat, A. M. (2014). Camera calibration for tracked vehicles augmented reality applications. In *Innovative control systems for tracked vehicle platforms* (pp. 147–162). Springer International Publishing.
21. Jedrasiak, K., & Nawrat, A. (2008). Fast colour recognition algorithm for robotics. *Problemy Eksploatacji, 3*, 69–76.
22. Daniec, K., Iwaneczko, P., Jędrasiak, K., & Nawrat, A. (2013). Prototyping the autonomous flight algorithms using the Prepar3D® simulator. In *Vision based systems for UAV applications* (pp. 219–232). Springer International Publishing.
23. Bhatt, R., & Dhall, A. (2010). Skin Segmentation Dataset, UCI Machine Learning repository.
24. Du, K.-L., & Swamy, M. N. S. (2013). *Neural networks and statistical learning*. Springer Science & Business Media.
25. Werbos, P. J. (1994). *The roots of backpropagation: From ordered derivatives to neural networks and political forecasting* (Vol. 1). Wiley.
26. Kasinski, A., Florek, A., & Schmidt, A. (2008). The PUT face database. *Image Processing and Communications*, *13*(3–4), 59–64.
27. Jaccard, P. (1912). The distribution of the flora in the Alpine Zone. *New Phytologist*, *11*(2), 37–50.