Displays 74 (2022) 102286

Contents lists available at ScienceDirect

Displays

journal homepage: www.elsevier.com/locate/displa

Tuning of grayscale computer vision systems☆,☆☆

Pavel Škrabánek [a,*], Natália Martínková [b,c]

[a] Institute of Automation and Computer Science, Brno University of Technology, Technická 2896/2, 616 69 Brno, Czech Republic
[b] Institute of Vertebrate Biology, The Czech Academy of Sciences, Květná 8, 603 65 Brno, Czech Republic
[c] RECETOX, Masaryk University, Kotlářska 2, 611 37 Brno, Czech Republic

ARTICLE INFO

MSC:
68T45
65K10

Keywords:
Computer vision
Parameter optimization
Performance evaluation
WECIA graph
Weighted means grayscale conversion

ABSTRACT

Computer vision systems perform based on their design and parameter setting. In computer vision systems that use grayscale conversion, the conversion of RGB images to a grayscale format influences performance of the systems in terms of both results quality and computational costs. Appropriate setting of the weights for the weighted means grayscale conversion, co-estimated with other parameters used in the computer vision system, helps to approach the desired performance of a system or its subsystem at the cost of a negligible or no increase in its time-complexity. However, parameter space of the system and subsystem as extended by the grayscale conversion weights can contain substandard settings. These settings show strong sensitivity of the system and subsystem to small changes in the distribution of data in a color space of the processed images. We developed a methodology for Tuning of the Grayscale computer Vision systems (TGV) that exploits the advantages while compensating for the disadvantages of the weighted means grayscale conversion. We show that the TGV tuning improves computer vision system performance by up to 16% in the tested case studies. The methodology provides a universally applicable solution that merges the utility of a fine-tuned computer vision system with the robustness of its performance against variable input data.

1. Introduction

Inputs to computer vision systems can be grayscale images [1], video sequences [2], or views from multiple cameras [3], among others. In practice, color images are the most common, and the color information is usually represented by the RGB (red, green, blue) color model. Color provides powerful and significant discriminative information, but many computer vision tasks are preferentially solved on images that have been converted to grayscale [4]. The main reasons for the conversion are to decrease the time-complexity of computer vision systems and to reduce the amount of training data required to achieve good performance [5].

A color image I, represented using the RGB color model, consists of red, green, and blue components I_R, I_G, and I_B, respectively. The conversion of an RGB image $I = (I_R, I_G, I_B)$ of $K \times L$ pixels into a grayscale image Y is a mapping of the image I from an $\mathbb{R}^{K \times L \times 3}$ representation to an $\mathbb{R}^{K \times L}$ representation. The conversion can be accomplished in many ways [6]. Most grayscale conversion methods [5,7–12] have been optimized to produce perceptually accurate results for human use [6]. Optimizing grayscale conversion with respect to the human visual system poses a disadvantage for computer vision tasks [5,13]. The utilization of grayscale conversion methods reflecting human perception is not always well justified or warranted for computer vision tasks, because the conversion needs to contribute to maximizing features essential for the task to be solved [5]. In other words, human perception optimization might hinder achieving human goals using the computer vision.

Grayscale conversion methods based on weighted linear combinations of the red, green, and blue image channels are appropriate for object [4] and texture recognition [5] and for edge detection [14]. The weighted means-based conversion methods have very good ratios between time-complexity and performance [5]. The weighted means-based conversion methods can be described by a general formula

$$Y = w_R I_R + w_G I_G + w_B I_B, \tag{1}$$

☆ This paper was recommended for publication by G. Guangtao Zhai.

☆☆ The work was supported by the Ministry of Education, Youth and Sports of the Czech Republic, INTER-COST project, Czech Republic LTC18053 European COST Action CA15140, Brno University of Technology, Czech Republic Grant No. FSI-S-20-6538, and the Czech Science Foundation, Czech Republic Grant No. 17-20286S.

* Corresponding author.
E-mail address: pavel.skrabanek@vut.cz (P. Škrabánek).

https://doi.org/10.1016/j.displa.2022.102286
Received 14 February 2022; Received in revised form 20 July 2022; Accepted 7 August 2022
Available online 17 August 2022

where Y is a grayscale image and w_R, w_G, and w_B are the weighting coefficients of the red, green, and blue components, respectively. It holds that $w_R, w_G, w_B \in [0, 1]$ and $w_R + w_G + w_B = 1$.

The performance of a computer vision system or of its part (a subsystem) is influenced by the settings of all operations incorporated into the pipeline of the system or subsystem, respectively. To reach a desired performance of the system or subsystem, an appropriate setting of all relevant operations must be found, including the setting of the weighting coefficients w_R, w_G, and w_B in systems and subsystems with implemented weighted means grayscale conversion (1) [4,15]. Let us call such systems and subsystems "the grayscale computer vision systems and subsystems."

Grayscale computer vision systems and subsystems have typically more than three parameters, and hand-tuning of their setting becomes random or impossible. Utilization of an optimization method that allows a fully automatic search for the optimal setting is recommended. Unfortunately, parameter spaces of grayscale computer vision systems and subsystems can contain substandard settings [15]. The substandard settings show strong sensitivity of the systems and subsystems to changes in the distribution of data in a color space of processed images such that small changes can cause significant changes in the system and subsystem performances [15,16]. Substandard settings that lead to chaotic responsiveness of the systems and subsystems are undesirable for practical applications.

The magnitude of the performance changes depends on pipelines of the systems or subsystems and on the available data. Recognizing what constitutes a detrimental performance change requires involvement of a domain (computer vision) expert to identify and eliminate the substandard settings. The role of the experts is to assess the overall performance of the system or subsystem in order to ensure selection of a reliable setting (i.e., selection of the optimal setting is influenced by the expert judgment [17]). Experts can assess performance of the systems or subsystems using numerical data or they use a graphical tool, such as a weighting coefficients impact assessment (WECIA) graph, to facilitate their involvement in the tuning process [16].

To help find stable and reliable settings of computer vision systems and subsystems that use grayscale conversion, we developed a methodology for Tuning of the Grayscale computer Vision (TGV) systems. The TGV methodology is an expert-guided optimization whereby domain experts ensure that the substandard settings are avoided. To ensure flexibility, TGV assesses performance of grayscale computer vision systems and subsystems using user-defined objective functions. The experts supervise the optimization process using WECIA graphs. WECIA graphs were designed to display classification performance of image classification systems. We generalized the use of WECIA graphs to any computer vision task while taking into account performance, and any other property of a grayscale computer vision system or subsystem. To improve clarity of the graphs [18], we used a perceptually uniform color map instead of the originally proposed jet color map. We implemented the methodology in MATLAB, and here we demonstrate utilization and properties of TGV on an object categorization (grape detection [15]) and an image segmentation task (vessel segmentation [19]).

The key contributions of the article are:

- We propose a methodology for tuning of grayscale computer vision systems. Parameter settings determined by the methodology result in computer vision systems with desired performance, energy consumption, or time-complexity.
- We confirm the need to involve a computer vision expert in the parameter tuning process.
- We demonstrate the discrepancy between human and computer vision requirements on grayscale conversion.
- We propose consistent co-estimation of all computer vision system parameters, including the grayscale conversion weights, and expert evaluation of the stability of the optimized setting.

2. Background

2.1. Tuning of computer vision systems and subsystems

In computer vision, parameter tuning is the process of finding a parameter setting θ that provides a desired performance (high true positive rate, low false positive rate, etc.) and properties (low energy consumption, low time-complexity, etc.) of a computer vision system or subsystem S. An adequate setting must be found for all m parameters p of S according to an objective function f. Thus, the parameter tuning is an optimization problem, where we search for optimal input arguments of the function f. Specifically, we search for the optimal parameter setting θ^* that provides the highest (max problem) or the lowest (min problem) objective function value. The search is carried out on a parameter space Θ of S, where Θ is a set of the settings θ. It is given as $\Theta = X_{p_1} \times \cdots \times X_{p_m}$, where X_{p_i} is a domain of the ith parameter p_i, $i \in \mathcal{I}$, and $\mathcal{I} = \{1, \ldots, m\}$. It means that $\theta = (p_1, \ldots, p_m)$ and $\theta^* = (p_1^*, \ldots, p_m^*)$, where p_i^* is the optimal setting of the ith parameter p_i for $\forall i \in \mathcal{I}$.

The objective function f comprises of quantitative assessments of the performance and properties of S. If S employs a learning algorithm, S must be trained on a set of annotated samples T before its evaluation. The training–evaluation sequence must be carried out for each $\theta \in \Theta$ separately [15]. The evaluation is accomplished on an evaluation set E, where E consists of annotated samples and $T \cap E = \emptyset$. Thus, the objective function can be formally written as

$$v = f(\theta, S, E, T), \tag{2}$$

where v is an objective function value and $v \in \mathbb{R}$. Evaluation of the system or subsystem S for $\forall \theta \in \Theta$ results in a set V of evaluation results v, where $V = \{v | v = f(\theta, S, E, T) \wedge v \in \mathbb{R} \wedge \theta \in \Theta\}$.

The search for the optimal parameter setting θ^* can be automated using an optimization algorithm. Algorithms such as grid-search [20], random search [21], Bayesian optimization [22], a robust parameter estimator [23], and genetic algorithms [4] can be used for this purpose. Usually, the algorithms carry out the search on a user-defined subset $\hat{\Theta}$ of the parameter space Θ. The subset is given as $\hat{\Theta} = \hat{X}_{p_1} \times \cdots \times \hat{X}_{p_m}$, where $\hat{X}_{p_i} \subset X_{p_i}, \forall i \in \mathcal{I}$. The parameter setting θ^* is optimal with respect to the subset $\hat{\Theta}$ of the parameter space Θ.

2.2. Grayscale computer vision system and subsystem

Computer vision systems and subsystems that utilize (1) are termed grayscale computer vision systems and subsystems. Let us denote these systems and subsystems as S_G and a setting of the grayscale conversion (1) as a triplet $\mathbf{w} = (w_R, w_G, w_B)$. Let us further consider that a setting of S_G is characterized by the m-tuple $\theta = (\mathbf{w}, p_2, \ldots, p_m)$, where $m \geq 2$. The parameter space of S_G is given as $\Theta_G = X_{\mathbf{w}} \times X_{p_2} \times \cdots \times X_{p_m}$, where $X_{\mathbf{w}}$ is a domain of the grayscale conversion setting $\mathbf{w}$.

2.3. WECIA graph

The WECIA graph $\mathcal{G}$ (Fig. 1) was designed to display a dependence of S_G performance on setting of the weighting coefficients $\mathbf{w}$ for one setting of the remaining $m-1$ parameters p. It consists of an equilateral triangle within which a given plotted point represents proportions of the color components used to create a grayscale image Y from a color image $I = (I_R, I_G, I_B)$ according to (1). The proportions of the red, green, and blue components are given by the weights w_R, w_G, and w_B, respectively. The axis related to the red component I_R is the left side of the triangle. The proportion of the red, given by w_R, is plotted on the axis where w_R increases downward. The same principle is used for the remaining two axes. The bottom and right sides are related to the green and the blue components I_G and I_B, respectively. The proportions of the green and the blue components w_G and w_B increase in the rightward and upward directions, respectively. Performance of

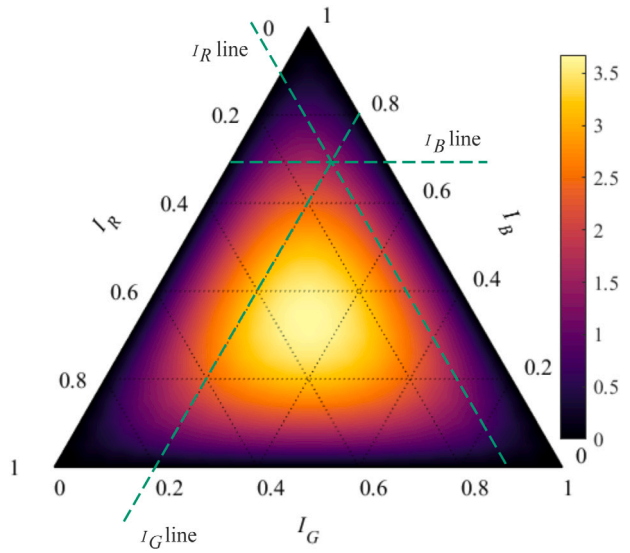

Fig. 1. The WECIA graph shows a dependence of performance and properties of a grayscale computer vision system or subsystem S_G on setting of the weighting coefficients $\mathbf{w} = (w_R, w_G, w_B)$. The performance and properties are evaluated using an objective function $v = f(\theta, S_G, E, T)$, where $\theta = (\mathbf{w}, p_2, \ldots, p_m)$ and $m \geq 2$. One graph displays the dependence for one setting of the $m-1$ parameters p. The sides of the triangle are axes of the graph and they are related to the weighting coefficients w_R, w_G, and w_B. The objective function values v are displayed using color in the area inside the triangle. Reading of v can be accomplished using auxiliary lines (broken line). Positions of the lines are given by an inspected setting of the weighting coefficients (in this case, $w_R = 0.1$, $w_G = 0.2$, and $w_B = 0.7$). The intersection of the lines positively determines v where its value can be estimated using the color bar (in this case, $v \approx 1.4$).

S_G, evaluated using $v = f(\theta, S_G, E, T)$, is represented using color in the area inside the triangle.

Detection of substandard settings requires evaluating S_G performance for various settings of the weighting coefficients $\mathbf{w}$ while keeping all other parameters of S_G unchanged. To get an expressive image of an influence of $\mathbf{w}$ on the performance, the evaluation results v should uniformly cover the surface bounded by the triangle, which means that the evaluation should be carried out for evenly sampled settings of the weighting coefficients w_R, w_G, and w_B. We define settings aimed at construction of $\mathcal{G}$ using two conditions: (i) $w_R, w_G, w_B \in \{0, \Delta w, 2\Delta w, \ldots, 1\}$, and (ii) $w_R + w_G + w_B = 1$. The step Δw is determined by a desired number of points on one axis of the diagram k, where $\Delta w = (k-1)^{-1}$, $k \geq 3$ and $k \in \mathbb{N}$. Settings $\mathbf{w} = (w_R, w_G, w_B)$ satisfying these conditions form a subset $\hat{X}_{\mathbf{w}}$ of the domain $X_{\mathbf{w}}$. To avoid visually artificial features or obscuring the real details, we use the inferno color map (Fig. 1), which aims to provide accurate perceptual uniformity [18].

3. TGV methodology

If carried out purely by means of an optimization algorithm, parameter tuning of a grayscale computer vision system or subsystem S_G can result in a substandard setting [15]. The TGV methodology enables one to identify and avoid such settings. Consider a situation wherein a solution provided by an optimization algorithm is a proposal of the optimal setting and one or more domain experts are then required to validate the fitness of the setting. The experts assess the proposal by means of the WECIA graph. If the experts consider the setting to be substandard, the selection–validation sequence must be repeated. The search for a new proposal is carried out on a reduced parameter space. This loop must be repeated until the experts approve a proposed setting or until all settings $\theta \in \Theta_G$ are labeled as substandard. The first setting approved by the experts is the best setting, which ensures the desired performance of S_G given the objective function. Let us consider this setting to be the optimal setting θ^*.

The TGV methodology can be implemented using various optimization algorithms. To demonstrate the principle of the methodology, we present implementation of the TGV methodology into a one-stage grid-search algorithm. The algorithm carries out the search for the proposals of optimal settings on a user-defined parameter space $\hat{\Theta}_G$. The parameter space $\hat{\Theta}_G$ is given by the number of points k on one axis of $\mathcal{G}$ and by user-defined domains $\hat{X}_{p_2}, \ldots, \hat{X}_{p_m}$ of the parameters $p_2, \ldots, p_m$, respectively. The domains $\hat{X}_{p_2}, \ldots, \hat{X}_{p_m}$ are finite sets of parameter settings considered during the tuning process.

The one-stage grid-search algorithm performs two successive steps when searching for the optimal setting [20]. First, it evaluates the objective function $v = f(\theta, S_G, E, T)$ for $\forall \theta \in \hat{\Theta}_G$. This step results in a set of objective function values V. In the second step, the algorithm determines the optimal setting θ^* using $\forall v \in V$.

In the presented implementation of TGV, the grid-search algorithm provides proposals of optimal settings that the expert will evaluate. If we search for a setting providing the highest objective function value v, then the jth proposal of the optimal setting is given as

$$\theta^j = \arg\max_{v \in V^j} v. \tag{3}$$

If the smallest v is required, the jth proposal is given as

$$\theta^j = \arg\min_{v \in V^j} v. \tag{4}$$

The search for θ^j is carried out on the jth set of objective function values V^j, where $V^j \subsetneq V^{j-1} \subsetneq \cdots \subsetneq V^1$. The set V^1 is formed by the grid-search algorithm at the initialization of TGV. It is given as

$$V^1 = \left\{ v | v = f(\theta, S_G, E, T) \wedge \theta \in \hat{\Theta}_G^1 \right\}, \tag{5}$$

where $\hat{\Theta}_G^1 = \hat{X}_{\mathbf{w}} \times \hat{X}_{p_2} \times \cdots \times \hat{X}_{p_m}$. As $v \in V^j$ are tied with settings $\theta \in \hat{\Theta}_G^j$ by f, the jth search is carried out on the jth parameter space $\hat{\Theta}_G^j$, where $\hat{\Theta}_G^j \subsetneq \hat{\Theta}_G^{j-1} \subsetneq \cdots \subsetneq \hat{\Theta}_G^1$. Note that $j \in \{1, \ldots, |\hat{\Theta}_G^1 \parallel \hat{X}_{\mathbf{w}}|^{-1}\}$.

TGV creates a WECIA graph $\mathcal{G}^j$ for the jth proposal $\theta^j = \left(\mathbf{w}^j, p_2^j, \ldots, p_m^j\right)$. $\mathcal{G}^j$ displays objective function values $v \in V^j$ for $\forall \theta \in \hat{\Theta}^j$, where $p_2 = p_2^j, \ldots, p_m = p_m^j$. The experts assess the overall evaluation results that are displayed. If the results indicate a substandard setting, all displayed objective function values v and corresponding settings θ must be removed from V^j and $\hat{\Theta}^j$, respectively. This operation results in a new set of objective function values V^{j+1} and a new parameter space $\hat{\Theta}^{j+1}$. The sets V^{j+1} and $\hat{\Theta}^{j+1}$ are used in the $(j+1)$th iteration of TGV. Once the expert approves a proposed setting, the setting is considered to be the optimal setting θ^* and the search terminates.

We implement the presented union of the grid-search with the TGV methodology as a function TGV in MATLAB script language (Supplementary Material 1). The function requires a designated evaluation set E as one of its inputs. A training set T must be provided only if the system or subsystem S_G employs a learning algorithm. A pseudocode of the function for the max problem (3) is given in algorithm 1. For the min problem, the line 10 must be changed according to (4).

4. Experiments

4.1. Detection of grapes

Single grape detectors aim to recognize grapes in real-life images. A grape detector based on a histogram of oriented gradients (HOG) descriptor and on a support vector machine (SVM) classifier is efficient in detecting white grape varieties in RGB images [15]. The detector classifies input object images of dimensions 40 × 40 pixels into two nonoverlapping classes: “berry” (“positive”) and “not berry” (“negative”).

The vision pipeline of the grape detector consists of three successive stages, where each stage comprises one operation (Fig. 2). At the first

Algorithm 1 Union of the one-stage grid-search algorithm with the TGV methodology (max problem).

1: **function** TGV($f, S_G, E, T, k, \hat{X}_{p_2}, \ldots, \hat{X}_{p_m}$)
Require: An objective function f; a grayscale computer vision system/subsystem S_G; an evaluation set E; a (optional) training set T; the number of points k on one axis of a WECIA graph $\mathcal{G}$; subsets of domains $\hat{X}_{p_2}, \cdots, \hat{X}_{p_m}$ of the parameters $p_2, \cdots, p_m$, respectively.
Ensure: Optimal setting of the parameters $\theta^* = (\mathbf{w}^*, p_2^*, \ldots, p_m^*)$. $\theta^* = -1$ when no setting has been approved.

One-stage grid-search:
2: $\Delta w \leftarrow (k-1)^{-1}$
3: $\hat{X}_{\mathbf{w}} \leftarrow \{(w_R, w_G, w_B) | w_R, w_G, w_B \in \{0, \Delta w, 2\Delta w, \ldots, 1\} \wedge w_R + w_G + w_B = 1\}$
4: $\hat{\Theta}_G^1 \leftarrow \hat{X}_{\mathbf{w}} \times \hat{X}_{p_2} \times \cdots \times \hat{X}_{p_m}$ ▷ user-defined subset of parameter space Θ_G
5: $V^1 \leftarrow \{v | v = f(\theta, S_G, E, T) \wedge \theta \in \hat{\Theta}_G^1\}$ ▷ evaluation results obtained $\forall \theta \in \hat{\Theta}_G^1$

TGV:
6: $\gamma \leftarrow 0$ ▷ flag where $\gamma = 0$ for a substandard setting and $\gamma = 1$ for an expert approved one
7: $j \leftarrow 0$
8: **repeat**
9: $j \leftarrow j + 1$
10: $\theta^j = \arg\max_{v \in V^j} v$ ▷ note that $v = f(\theta, S_G, E, T)$ and $\theta^j = \left(\mathbf{w}^j, p_2^j, \ldots, p_m^j\right)$
11: $\mathcal{G}^j \leftarrow \left\{(\mathbf{w}, v) | \theta \in \hat{\Theta}_G^j \wedge v \in V^j \wedge p_2 = p_2^j \wedge \ldots \wedge p_m = p_m^j\right\}$ ▷ the WECIA graph $\mathcal{G}$ for $p_2^j, \ldots, p_m^j$
12: $\gamma \leftarrow$ EXPvalidate ▷ $\gamma = 1$ when θ^j approved by the expert, otherwise $\gamma = 0$ (substandard setting)
13: **if** $\gamma = 0$ **then** ▷ removing all entries related to a substandard setting
14: $\hat{\Theta}_G^{j+1} \leftarrow \left\{\theta | \theta \in \hat{\Theta}_G^j \wedge p_2 \neq p_2^j \wedge \ldots \wedge p_m \neq p_m^j\right\}$ ▷ given by $p_2^j, \ldots, p_m^j$
15: $V^{j+1} \leftarrow \left\{v | v \in V^j \wedge \theta \in \hat{\Theta}_G^{j+1}\right\}$ ▷ note that $v = f(\theta, S_G, E, T)$
16: **if** $\hat{\Theta}_G^{j+1} = \emptyset$ **then** ▷ if no other setting is available for the expert evaluation,
17: $\theta^j \leftarrow -1$ ▷ no optimal setting exists
18: $\gamma = 1$ ▷ thus terminate the search
19: **end if**
20: **end if**
21: **until** $\gamma = 0$
22: $\theta^* \leftarrow \theta^j$
23: **return** θ^*
24: **end function**

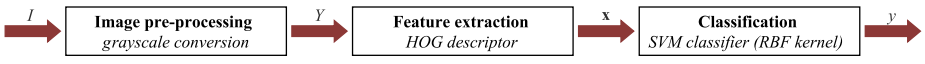

Fig. 2. Vision pipeline of the grape detector. The object categorization system consists of three successive stages (image pre-processing, feature extraction, and classification), where one operation is employed at each stage. The grayscale conversion (1) at the first stage converts an input RGB image $I = (I_R, I_G, I_B)$ to a grayscale image Y. The HOG descriptor extracts a vector of features $\mathbf{x}$ from Y at the second stage. At the third stage, the SVM classifier with the RBF kernel assigns the image I a label y with respect to $\mathbf{x}$.

stage, an object image $I = (I_R, I_G, I_B)$ is converted from the RGB color model into a grayscale image Y according to (1). Second, the HOG descriptor extracts a vector of features $\mathbf{x}$ from Y. In the last stage, the SVM classifier with a radial basis function (RBF) kernel assigns I a label y using $\mathbf{x}$.

The detector has eight parameters p. The grayscale conversion (1) contributes the setting of weights $\mathbf{w}$. The descriptor has five parameters: range r (in degrees), number of bins b, size of cells $\mathbf{c}_s$ (in pixels), number of cells in blocks $\mathbf{c}_b$ (in cells), and number of overlapping cells between adjacent blocks $\mathbf{c}_o$ (in cells) [24]. The performance of the classifier is influenced by a regularization constant C and a kernel width σ [15]. The setting and the parameter space of the grape detector is given as $\theta = (\mathbf{w}, r, b, \mathbf{c}_s, \mathbf{c}_b, \mathbf{c}_o, C, \sigma)$ and $\Theta_G = X_{\mathbf{w}} \times X_r \times X_b \times X_{\mathbf{c}_s} \times X_{\mathbf{c}_b} \times X_{\mathbf{c}_o} \times X_C \times X_\sigma$, respectively. The domains of the HOG descriptor parameters are: $X_r = \{180, 360\}$, $X_b \in \mathbb{N}_{>0}$, and $X_{\mathbf{c}_s}, X_{\mathbf{c}_b} \in \mathbb{N}_{>0}^2$, respectively. It holds that

$$X_{\mathbf{c}_o} = \left\{\mathbf{c}_o | \mathbf{c}_o = \lceil 0.5\mathbf{c}_b \rceil \wedge \mathbf{c}_b \in X_{\mathbf{c}_b}\right\}. \tag{6}$$

The domains of SVM parameters are: $X_C, X_\sigma \in \mathbb{R}_{>0}$.

We propose two case studies aimed at tuning of the grape detector given alternative behavior of the computer vision performance. The goal of case study 1 is to find a setting providing the highest classification accuracy acc [25]. For this case study, the objective function is given as

$$v = \text{acc}. \tag{7}$$

In case study 2, we search for a setting providing high classification accuracy while maintaining low time-complexity of the detector. Considering these requirements, we propose an objective function where the time complexity is proportional to the length of the feature vector. The objective function is given as

$$v = \text{acc} - |\mathbf{x}|', \tag{8}$$

where $|\mathbf{x}|'$ is a min–max normalization of the feature vector length. The length is given as

$$|\mathbf{x}| = b \prod_{i=1}^{|\lambda|} \lambda_i, \tag{9}$$

where

$$\lambda = \mathbf{c}_b \times \left| \left(|I| \times \mathbf{c}_s^{-1} - \mathbf{c}_b\right) \times \left(\mathbf{c}_b - \mathbf{c}_o\right)^{-1} + (1,1) \right|. \tag{10}$$

We use a set GX-3 as the evaluation set E that consists of 2000 "positive" and 2000 "negative" samples [15]. Because SVMs are supervised learning methods, the detector must be trained before its evaluation. We use a set T-3 consisting of 288 "positive" and 288

Fig. 3. Vision pipeline of the vessel segmentation system. The image segmentation system consists of four successive stages (image pre-processing, transformation, segmentation and cleaning). Within the image pre-processing, a RGB fundus image $I = (I_R, I_G, I_B)$ is converted to a grayscale image Y according to (1). At the second stage, Y is transformed into Y_W using IUWT. Global thresholding of Y_W results in a binary image Y_B at the third stage. At the last stage, Y_B is cleaned by removing small objects and filling small holes, respectively. This process results in a binary image Y_C.

"negative" samples as the training set T [26]. We consider the following subsets of parameter domains: $\hat{X}_\tau = \{180, 360\}$, $\hat{X}_b = \{9, \ldots, 21\}$, $\hat{X}_{\mathbf{c}_s} = \{(2,2),(3,3)\}$, $\hat{X}_{\mathbf{c}_b} = \{(2,2),(3,3)\}$, $\hat{X}_C = \{1, 10, 100\}$, and $\hat{X}_\sigma = \{1, 10, 20, 30, 40\}$. The subset $\hat{X}_{\mathbf{c}_o}$ is given by $\hat{X}_{\mathbf{c}_b}$ and (6). Further, $k = 21$.

4.2. Vessel segmentation

The purpose of a vessel segmentation system is to locate blood vessels in retinal images. A computer vision system developed for fast retinal vessel segmentation relies on isotropic undecimated wavelet transform (IUWT) [19]. Inputs of the system are RGB human fundus images. Its outputs are binary images with detected retinal vessels. The spatial dimensions of the input and output images are identical.

The system consists of four successive stages (Fig. 3). First, an input RGB image $I = (I_R, I_G, I_B)$ is converted into the grayscale image Y according to (1). Second, Y is transformed using IUWT into Y_W. Third, the segmentation of Y_W is carried out by means of a global thresholding and results in a binary image Y_B. Fourth, cleaning of Y_B results in a binary image Y_C. This stage involves removing small objects and filling small holes, respectively.

In addition to the setting of the weights $\mathbf{w}$, the vessel segmentation system has four other parameters p. Performance of IUWT is influenced by a vector $\mathbf{l} = (l_1, \ldots, l_{n_l})$ of n_l wavelet levels l. Its domain is given as $X_\mathbf{l} = \{\mathbf{l} | l_1 \in \mathbb{N}_{>0} \wedge l_{i+1} = l_i + 1\}$, where l_i is the ith element of the vector $\mathbf{l}$. The segmentation is controlled by a threshold τ with a domain $X_\tau = \{\tau | \tau \in \mathbb{R} \wedge 0 \le \tau \le 1\}$. The cleaning requires specification of a minimum size of an object to be kept o, and a minimum size of a "hole" h (size of a region surrounded by pixels detected as vessels that should remain as retinal background). Both, o and h are in pixels. Their domains are $X_o, X_h \in \mathbb{N}_0$. The setting and the parameter space of the vessel segmentation system is given as $\theta = (\mathbf{w}, \mathbf{l}, \tau, o, h)$ and $\Theta_G = X_\mathbf{w} \times X_\mathbf{l} \times X_\tau \times X_o \times X_h$, respectively.

We use test images from a digital retinal image for a vessel extraction (DRIVE) database as the evaluation set E [27]. E consists of 20 manually segmented images. Pixels representing the blood vessels (positive samples) are in a minority for each of the images. We consider the following subsets of parameter domains: $\hat{X}_\mathbf{l} = \{\mathbf{l} | l_1 \in \{1, \ldots, 4\} \wedge \max_{l \in \mathbf{l}} l \in \{2, \ldots, 5\} \wedge l_{i+1} = l_i + 1\}$, $\hat{X}_\tau = \{0.125, 0.15, \ldots, 0.25\}$, $\hat{X}_o = \{70, 75, 80\}$, and $\hat{X}_h = \{10, 15, \ldots, 30\}$. Further, $k = 21$.

We propose two case studies that tune the vessel segmentation system. In case studies 3 and 4, respectively, we search for a setting that provides the highest segmentation accuracy and the highest true positive rate tpr [28]. As the class distributions are highly imbalanced for all the images in the evaluation set E, class balanced accuracy $\mathrm{acc_B}$ must be used in case study 3 [29]. We formulate the objective function for this case study as an average class balanced accuracy

$$\upsilon = \frac{1}{n_I} \sum_{i=1}^{n_I} \mathrm{acc_B}(i), \tag{11}$$

where n_I is the number of images in the set E, and $\mathrm{acc_B}(i)$ is the balanced accuracy of the ith image. For case study 4, we define the objective function as

$$\upsilon = \frac{1}{n_I} \sum_{i=1}^{n_I} \mathrm{tpr}(i), \tag{12}$$

where $\mathrm{tpr}(i)$ is the true positive rate of the ith image in E.

Table 1
Course of the search for the setting providing the highest classification accuracy of the grape detector in case study 1 for the full TGV (second and third rows) and using standard grayscale conversion methods. The best performance is in bold.

Conversion	j	$\mathbf{w}^j$	r^j	b^j	$\mathbf{c}_s^j$	$\mathbf{c}_b^j$	C^j	σ^j	υ^j
TGV	1	(0.90,0.05,0.05)	1	10	(2,2)	(2,2)	1	1	0.8920
	2	(1.00,0.00,0.00)	1	9	(3,3)	(2,2)	10	20	**0.8898**
BT.601	1	(0.30,0.59,0.11)	1	12	(3,3)	(3,3)	10	20	0.8760
R	1	(1.00,0.00,0.00)	1	9	(3,3)	(2,2)	10	20	**0.8898**
G	1	(0.00,1.00,0.00)	1	12	(3,3)	(3,3)	100	20	0.8790
B	1	(0.00,0.00,1.00)	1	9	(3,3)	(2,2)	10	20	0.8020

The table consists of indexes j of proposed settings (column 1), the proposed settings θ^j (columns 2–7) and objective function values υ^j (column 8), where $\theta^j = (\mathbf{w}^j, r^j, b^j, \mathbf{c}_s^j, \mathbf{c}_b^j, \mathbf{c}_o^j, C^j, \sigma^j)$ and $\mathbf{c}_o^j$ is given by (6).

4.3. Performance comparison without optimized grayscale conversion

We compare performance of the TGV methodology with optimization of grayscale conversion and evaluation of the stability of the result to computer vision system optimization performed without including the weights $\mathbf{w}$ in the parameter space. We used fixed settings corresponding to the most common conversion methods in computer vision: ITU-R recommendations BT.601 $\mathbf{w} = (0.30, 0.59, 0.11)$, and utilization of isolated red $\mathbf{w} = (1.00, 0.00, 0.00)$, green $\mathbf{w} = (0.00, 1.00, 0.00)$ and blue channels $\mathbf{w} = (0.00, 0.00, 1.00)$ as grayscale images.

5. Results

5.1. Tuning of the grape detector

5.1.1. Case study 1 – Grape detector with setting providing the highest accuracy

While searching for a setting that provides the highest classification accuracy of the grape detector, we evaluated two proposed settings (Table 1, rows two and three). We labeled the first proposed setting θ^1 as substandard after evaluating the performance using the first WECIA graph $\mathcal{G}^1$ (Fig. 4(a)). The graph displays settings featuring rapid changes of υ in relation to small changes of the weighting coefficient setting $\mathbf{w}$. For example, $\Delta\upsilon \approx -0.45$ for $w = w^1 \pm \Delta w$, where $w^1 \in w_R^1, w_G^1, w_B^1$ and $\mathbf{w}^1 = (w_R^1, w_G^1, w_B^1)$. This indicates a setting that will likely prove problematic in classification of alternative datasets. As the graph $\mathcal{G}^2$ (Fig. 4(b)) does not contain such attributes, we approved the setting θ^2 (i.e., $\theta^* = \theta^2$). Settings providing the highest accuracy for the fixed grayscale conversion methods had 0 to 8.8% lower performance according to the objective function (7) compared to the selected TGV setting (rows 4–7 in Table 1).

5.1.2. Case study 2 – Grape detector with setting providing high accuracy and low time-complexity

Tuning of the grape detector according to (8) resulted in one proposed setting (Table 2, second row). The WECIA graph $\mathcal{G}^1$ does not reveal any anomaly in the displayed evaluation results (Fig. 5), and we approved the first proposed setting θ^1 (i.e., $\theta^* = \theta^1$). Settings providing the highest accuracy and lowest time-complexity for the fixed grayscale conversion methods underperformed the TGV setting by 0.3 to 9.1% according to the objective function (8) (rows 3–6 in Table 2).

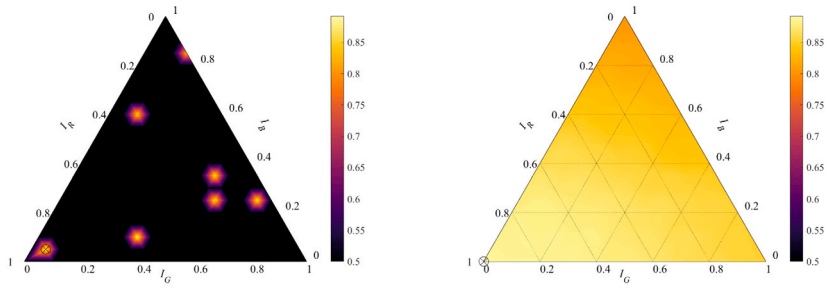

Fig. 4. WECIA graphs for the first (a) and second (b) proposed setting θ^1 and θ^2, respectively, in case study 1. Objective function value of the first proposed setting (crossed circle in (a)) is about 0.8920 (Table 1, $j = 1$) but the values in its neighborhood are only 0.5000. This phenomena can be observed for several other settings where $p_2 = p_2^1, \ldots, p_m = p_m^1$. Such settings are substandard. Objective function value of the second proposed setting (crossed circle in (b)) is about 0.8898 (Table 1, $j = 2$). Objective function values in its neighborhood are similar and they gradually change with respect to $\mathbf{w}$. Considering these facts, we approved the setting θ^2 as the optimal setting θ^*. The scale of colormaps is identical for both graphs.

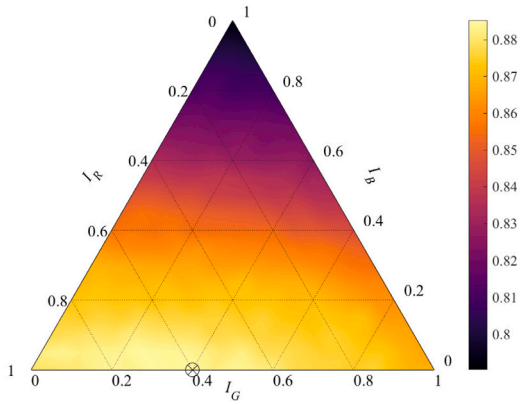

Fig. 5. WECIA graph for the first proposed setting in case study 2. Objective function value υ of the first proposed setting θ^1 (Table 2, $j = 1$), given by (8), is about 0.8853 (crossed circle). The displayed dependency of υ on $\mathbf{w}$ does not show any attribute typical for substandard settings (i.e., $\theta^* = \theta^1$).

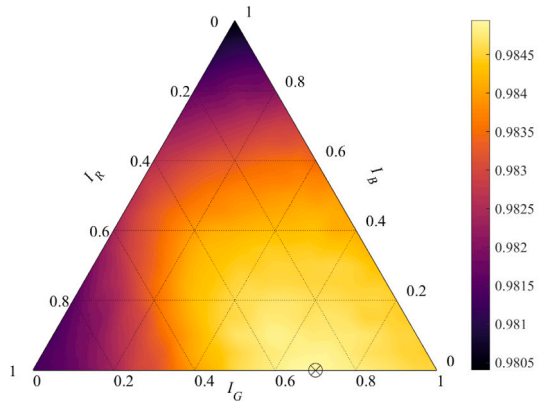

Fig. 6. WECIA graph for the first proposed setting within case study 3. Objective function value υ of the first proposed setting θ^1 (Table 3, $j = 1$), given by (11), is about 0.9850 (crossed circle). The displayed dependency of υ on $\mathbf{w}$ does not show any attribute typical for substandard settings (i.e., $\theta^* = \theta^1$).

Table 2
Course of the search for a setting providing high classification accuracy and low time-complexity of the grape detector in case study 2 for the full TGV (second row) and using standard grayscale conversion methods. The best performance is in bold.

Conversion	j	$\mathbf{w}^j$	r^j	b^j	$\mathbf{c}_s^j$	$\mathbf{c}_b^j$	C^j	σ^j	υ^j
TGV	1	(0.60,0.40,0.00)	1	9	(3,3)	(3,3)	10	10	**0.8853**
BT.601	1	(0.30,0.59,0.11)	1	9	(3,3)	(3,3)	10	10	0.8752
R	1	(1.00,0.00,0.00)	1	9	(3,3)	(3,3)	1	30	0.7947
G	1	(0.00,1.00,0.00)	1	9	(3,3)	(3,3)	10	10	0.8682
B	1	(0.00,0.00,1.00)	1	9	(3,3)	(3,3)	100	20	0.8825

The table consists of indexes j of proposed settings (column 1), the proposed settings θ^j (columns 2–7) and objective function values υ^j (column 8), where $\theta^j = (\mathbf{w}^j, r^j, b^j, \mathbf{c}_s^j, \mathbf{c}_b^j, \mathbf{c}_o^j, C^j, \sigma^j)$ and $\mathbf{c}_o^j$ is given by (6).

Table 3
Course of a search for the setting providing the highest segmentation accuracy of the vessel segmentation system in case study 3 for the full TGV (second row) and using standard grayscale conversion methods. The best performance is in bold.

Conversion	j	$\mathbf{w}^j$	$\mathbf{l}^j$	τ^j	o^j	h^j	υ^j
TGV	1	(0.30,0.70,0.00)	(1,2)	0.125	80	30	**0.9850**
BT.601	1	(0.30,0.59,0.11)	(1,2)	0.125	80	30	0.9848
R	1	(1.00,0.00,0.00)	(1,2)	0.125	70	30	0.9814
G	1	(0.00,1.00,0.00)	(1,2)	0.125	80	25	0.9845
B	1	(0.00,0.00,1.00)	(1,2)	0.125	70	10	0.9809

The table consists of indexes j of proposed settings (column 1), the proposed settings θ^j (columns 2–6) and objective function values υ^j (column 7), where $\theta^j = (\mathbf{w}^j, \mathbf{l}^j, \tau^j, o^j, h^j)$.

5.2. Tuning of the vessel segmentation system

5.2.1. Case study 3 – Vessel segmentation with setting providing the highest accuracy

Table 3 (second row) shows the course of a search for the setting providing the highest segmentation accuracy of the vessel segmentation system. We approved the first proposed setting θ^1 as optimal. Changes of the objective function values (averaged balanced accuracy) are proportionate to changes of the weighting coefficient setting (Fig. 6). Settings providing the highest accuracy for the fixed grayscale conversion methods performed by 0.02 to 0.4% worse than the TGV setting according to the objective function (11) (rows 3–6 in Table 3).

5.2.2. Case study 4 - Vessel segmentation with setting providing the highest true positive rate

The setting providing the highest true positive rate of the vessel segmentation system was determined in the first iteration (Table 4, row two). The gradual change of the objective function values (Fig. 7) matches with the expected performance of a computer vision system that is robustly set up. Settings providing the highest true positive rate for the fixed grayscale conversion methods performed by 0 to 16.4% worse than the TGV setting according to the objective function (12) (rows 3–6 in Table 4).

6. Discussion

We intended the TGV methodology to be used for parameter tuning of grayscale computer vision systems or subsystems. The methodology

Table 4
Course of the search for a setting providing the highest segmentation true positive rate of the vessel segmentation system in case study 4 for the full TGV (second row) and using standard grayscale conversion methods. The best performance is in bold.

Conversion	j	$\mathbf{w}^j$	$\mathbf{l}^j$	τ^j	o^j	h^j	υ^j
TGV	1	(0.00,1.00,0.00)	(2,3,4)	0.25	70	30	**0.8128**
BT.601	1	(0.30,0.59,0.11)	(2,3)	0.25	70	30	0.7973
R	1	(1.00,0.00,0.00)	(2,3)	0.25	70	30	0.6693
G	1	(0.00,1.00,0.00)	(2,3,4)	0.25	70	30	**0.8128**
B	1	(0.00,0.00,1.00)	(3,4)	0.25	70	30	0.6487

The table consists of indexes j of proposed settings (column 1), the proposed settings θ^j (columns 2–6) and objective function values υ^j (column 7), where $\theta^j = (\mathbf{w}^j, \mathbf{l}^j, \tau^j, o^j, h^j)$.

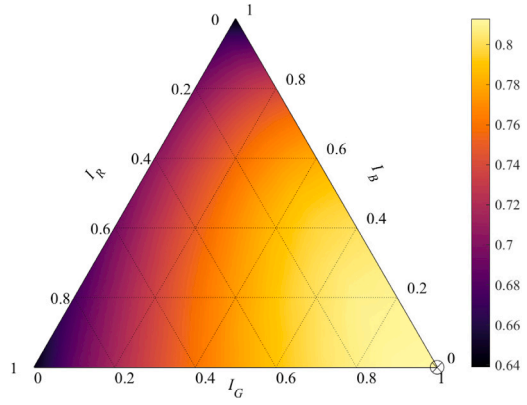

Fig. 7. WECIA graph for the first proposed setting within case study 4. Objective function value υ of the first proposed setting θ^1 (Table 4, $j = 1$), given by (12), is about 0.8128 (crossed circle). The displayed dependency of υ on $\mathbf{w}$ does not show any attribute typical for substandard settings (i.e., $\theta^* = \theta^1$).

provides a tool to identify and avoid the substandard settings that can occur in the parameter spaces of grayscale computer vision systems and subsystems. We confirmed the usefulness of the methodology in case study 1 (tuning of the grape detector for the highest accuracy), where the first proposed setting was substandard (Fig. 4(a)). The objective function value for this setting was 0.8920. The objective function value for the second proposed setting, which we approved as suitable, was 0.8898 (Table 1). The negligible drop in the objective function value for the second setting entitles us to consider the approved setting to be the optimal one, because the system will likely remain robust in practical applications.

The TGV methodology takes advantage of the computing power of modern computers. It utilizes an optimization algorithm that predetermines the time and space-complexities of TGV. The presented version of the TGV methodology (algorithm 1) was based on the one-stage grid-search algorithm. Both its time and space-complexities are $O\left(\prod_{i=1}^{m} |\hat{X}_{p_i}|\right)$, where m is number of parameters p and $|\hat{X}_{p_i}|$ is number of settings at the user-defined domain of the ith parameter p_i. The exponential complexities make the presented version of TGV suitable for tuning of grayscale computer vision systems and subsystems with a small number of parameters. The disadvantage of the high time-complexity can be partially suppressed by parallelization of operations. For systems and subsystems with a high number of parameters, the TGV methodology should be combined with a different optimization algorithm. Such algorithms as Bayesian optimization [22] and genetic algorithms [4] are appropriate for this purpose.

A key tool of TGV is the WECIA graph, which is capable to display dependence of the objective function values υ on $\mathbf{w}$. The WECIA graph was originally designed for object categorization tasks [16]. It was used to display dependence of a classification performance on the setting of the weighting coefficients $\mathbf{w}$, where the performance was evaluated using such single-value performance measures as accuracy and true positive rate [16]. Herein, we generalized the evaluation process while introducing the objective function f, where f can be of any structure, but it must hold that $f : \theta \to \mathbb{R}$ for $\forall\theta \in \Theta$. The TGV can thus be applied to computer vision tasks including, but not limited to, image classification [30], image segmentation [28], object detection [31], object tracking [32], and stereo matching [33,34]. We have shown that utilization of the objective function f extends the range of the WECIA graph's potential applications.

Our results highlight the importance of tuning grayscale conversion. We observed a significant difference among the displayed dependencies of υ on $\mathbf{w}$ for the optimal settings (Figs. 4(b)–7). For the grape detector, the optimal settings of the weighting coefficients were $\mathbf{w}^* = (1.00, 0.00, 0.00)$ and $\mathbf{w}^* = (0.60, 0.40, 0.00)$ for case studies 1 and 2, respectively. The performance of the grape detector diminished with increasing proportion of the blue component in the grayscale image (Figs. 4(b)–5). The difference between the highest and lowest objective function values was about 0.1 for both case studies (Figs. 4(b)–5). That represents deterioration of the performances by around 10% with respect to those having the highest objective values. For the vessel segmentation system, the optimal settings of the weighting coefficients were $\mathbf{w}^* = (0.30, 0.70, 0.00)$ and $\mathbf{w}^* = (0.00, 1.00, 0.00)$ for case studies 3 and 4, respectively. For case study 3, the performance of the vessel segmentation is almost insensitive to the setting of the weights (Fig. 6). For case study 4, the performance rapidly decreases with increasing proportion of the red or the blue component in the grayscale image (Fig. 7). The difference between the highest and the lowest objective function value was about 0.17, which is to say the deterioration in performance is about 21% with respect to the highest objective value.

We tested the proposed parameter tuning against a situation where the grayscale conversions build in the computer vision systems remained fixed. We optimized settings of the other parameters for the four case studies with the respective objective functions (7), (8), (11), and (12). The setting of the weighting parameters was fixed according to ITU-R recommendations BT.601, and to grayscale images corresponding to the isolated red, green, and blue color channels (see Section 4.3). In all four case studies, utilization of the most common grayscale conversion method (BT.601) decreased performance of the systems by 0.02 to 1.55% (Tables 3 and 4, respectively). The fact that the BT.601 conversion optimized for human perception underperformed the parameter setting suggested by TGV is not surprising considering the fact that the search with fixed grayscale conversion was carried out on a subset of the parameter space $\hat{\Theta}_G$. When using single channel conversion methods, the performance decreases up to 16.41% (Table 4, last row). However, note that the utilization of a single color channel conversion has been determined by TGV to be optimal in the first and fourth case study (Tables 1 and 4, respectively). Single channel grayscale conversion is computationally efficient and provides good performance is biological imaging [19].

Differences in the optimal setting of the grayscale conversion weighting coefficients vary in accordance with the given computer vision system. While the red channel carries the most discriminative information for the grape detection (case studies 1 and 2), the green channel is key for the vessel segmentation (case study 4). In none of the tested case studies was the grayscale conversion close to the preferred conversions from the perspective of human perception. The results obtained indicate that the optimal setting of a computer vision system and subsystem is highly dependent on the required performance, on the computer vision system or subsystem, and on the process data. We conclude that the setting of the weighting coefficients $\mathbf{w}$ in grayscale conversion can substantially influence performance of computer vision systems and subsystems. We propose to use TGV to reach optimal performance of the systems and subsystems. Human evaluation of an appropriate setting of the weighting coefficients based on visual data is difficult if not impossible (compare grayscale images in Figs. 8 and 9 for the optimal settings in panels (b) and (c) with images provided by the standard conversion methods in panels (d), (e) and (f)).

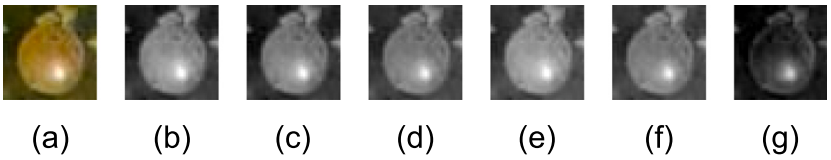

Fig. 8. Example of an input color image of the grape detector (a) and samples of grayscale images obtained using the weighted grayscale conversion setup according to the optimal setting for the case studies 1 (b) and 2 (c), ITU-R recommendation BT.601 (d), and for isolation of the red (e), green (f) and blue (g) color channels.

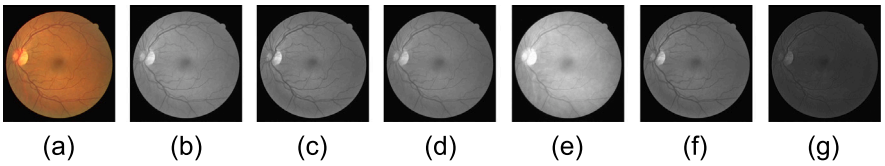

Fig. 9. Example of an input color image of the vessel segmentation system (a) and samples of grayscale images obtained using the weighted grayscale conversion setup according to the optimal setting for the case studies 3 (b) and 4 (c), ITU-R recommendation BT.601 (d), and for isolation of the red (e), green (f) and blue (g) color channels.

Can the involvement of the domain expert be eliminated to provide a tuning methodology robust against human judgment? We must unfortunately say no. Although the search for the optimal parameter setting can be fully automated, an expert must always specify the limit of acceptable performance changes for the given computer vision task before starting the search. The expert should also always consider cause for occurrence of substandard settings. For example, the isolated high-performance islands in case study 1 (Fig. 4(a)) may indicate overfitting of the classification model. Model overfitting could be eliminated by adding new, unique training samples to the training set.

7. Conclusion

Common software packages use the weighted grayscale conversion as the default conversion method. The weights are usually set up according to ITU-R recommendations BT.601 or BT.709. Our results show that the divergence from the standard grayscale conversion settings improves performance of the computer vision systems. The general formulation of the objective function (2) allows incorporation of specific requirements on computer vision systems, including requirements on performance, energy consumption, time-complexity, or space-complexity. The intuitive representation of evaluation results in WECIA graphs helps to improve resilience of computer vision systems aimed at image classification (case studies 1 and 2), image segmentation (case studies 3 and 4), but also object detection, object localization, stereo matching, or instance segmentation. Use of the general formula of the weighted means grayscale conversion followed by parameter tuning using the TGV methodology yields a solution merging the advantages of a fine-tuned computer vision system with the robustness of its performance against variable input data. Such an approach can be applied when developing new computer vision systems or in improving existing ones.

CRediT authorship contribution statement

Pavel Škrabánek: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Natália Martínková:** Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.displa.2022.102286.

References

[1] Z. Gao, L. Wang, L. Zhou, J. Zhang, HEp-2 cell image classification with deep convolutional neural networks, IEEE J. Biomed. Health Inf. 21 (2) (2017) 416–428, http://dx.doi.org/10.1109/JBHI.2016.2526603.

[2] Junichi Okuyama, Kana Nakajima, Kenta Matsui, Yuichi Nakamura, Kazuaki Kondo, Takahiro Koizumi, Nobuaki Arai, Application of a computer vision technique to animal-borne video data: extraction of head movement to understand sea turtles' visual assessment of surroundings, Animal Biotelemetry 3 (1) (2015) 35, http://dx.doi.org/10.1186/s40317-015-0079-y.

[3] Q. Yang, L. Wang, R. Yang, H. Stewénius, D. Nistér, Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling, IEEE Trans. Pattern Anal. Mach. Intell. 31 (3) (2009) 492–504, http://dx.doi.org/10.1109/TPAMI.2008.99.

[4] Ali Güneş, Habil Kalkan, Efkan Durmuş, Optimizing the color-to-grayscale conversion for image classification, Signal Image Video Process. 10 (5) (2016) 853–860, http://dx.doi.org/10.1007/s11760-015-0828-7.

[5] Garrison W. Kanan, Color-to-grayscale: Does the method matter in image recognition? PLoS ONE 7 (1) (2012) 1–7.

[6] M. Čadík, Perceptual evaluation of color-to-grayscale image conversions, Comput. Graph. Forum 27 (7) (2008) 1745–1754, http://dx.doi.org/10.1111/j.1467-8659.2008.01319.x.

[7] ITU-R Recommendation BT.601, Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios, 2011.

[8] Karl Rasche, Robert Geist, James Westall, Re-coloring images for Gamuts of lower dimension, Comput. Graph. Forum 24 (3) (2005) 423–432.

[9] Mark Grundland, Neil A. Dodgson, Decolorize: Fast, contrast enhancing, color to grayscale conversion, Pattern Recognit. 40 (11) (2007) 2891–2896.

[10] Ming Cui, Jiuxiang Hu, Anshuman Razdan, Peter Wonka, Color-to-gray conversion using ISOMAP, Vis. Comput. 26 (11) (2010) 1349–1360.

[11] Chengho Hsin, Hoai-Nam Le, Shaw-Jyh Shin, Color to grayscale transform preserving natural order of hues, in: Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, 2011, pp. 1–6, http://dx.doi.org/10.1109/ICEEI.2011.6021794.

[12] Tirui Wu, Alexander Toet, Color-to-grayscale conversion through weighted multiresolution channel fusion, J. Electron. Imaging 23 (2014) 23–23–6, http://dx.doi.org/10.1117/1.JEI.23.4.043004.

[13] Luca Benedetti, Massimiliano Corsini, Paolo Cignoni, Marco Callieri, Roberto Scopigno, Color to gray conversions in the context of stereo matching algorithms, Mach. Vis. Appl. 23 (2) (2012) 327–348, http://dx.doi.org/10.1007/s00138-010-0304-x.

[14] I. Ahmad, I. Moon, S.J. Shin, Color-to-grayscale algorithms effect on edge detection - a comparative study, in: 2018 International Conference on Electronics, Information, and Communication (ICEIC), 2018, pp. 1–4, http://dx.doi.org/10.23919/ELINFOCOM.2018.8330719.

[15] Pavel Škrabánek, Petr Doležel, Robust grape detector based on SVMs and HOG features, Comput. Intell. Neurosci. 2017 (2017) 1–17, http://dx.doi.org/10.1155/2017/3478602.
[16] Pavel Škrabánek, Sule Yildirim-Yayilgan, WECIA graph: visualization of classification performance dependency on grayscale conversion setting, Mendel 24 (2) (2018) 41–48, http://dx.doi.org/10.13164/mendel.2018.2.041.
[17] Cornelius Leondes, Image Processing and Pattern Recognition, Vol. 5, Academic Press, 1998.
[18] Fabio Crameri, Grace E. Shephard, Philip J. Heron, The misuse of colour in science communication, Nature Commun. 11 (1) (2020) 5444, http://dx.doi.org/10.1038/s41467-020-19160-7.
[19] Peter Bankhead, C. Norman Scholfield, J. Graham McGeown, Tim M. Curtis, Fast retinal vessel detection and measurement using wavelets and edge location refinement, PLOS ONE 7 (3) (2012) 1–12, http://dx.doi.org/10.1371/journal.pone.0032435.
[20] I.D. Coope, C.J. Price, On the convergence of grid-based methods for unconstrained optimization, SIAM J. Optim. 11 (4) (2001) 859–869.
[21] James Bergstra, Yoshua Bengio, Random search for hyper-parameter optimization, J. Mach. Learn. Res. 13 (2012) 281–305.
[22] B. Shahriari, K. Swersky, Z. Wang, R.P. Adams, N. de Freitas, Taking the human out of the loop: A review of Bayesian optimization, Proc. IEEE 104 (1) (2016) 148–175, http://dx.doi.org/10.1109/JPROC.2015.2494218.
[23] Charles V. Stewart, Robust parameter estimation in computer vision, SIAM Rev. 41 (3) (1999) 513–537.
[24] MATLAB R2018b, MathWorks, Inc., 2018.
[25] Pavel Škrabánek, Filip Majerík, Evaluation of performance of grape berry detectors on real-life images, in: Proceedings of the 22nd International Conference on Soft Computing MENDEL 2016, Brno University of Technology, Brno, Czech Republic, 2016, pp. 217–224.
[26] Pavel Škrabánek, Thomas Philip Runarsson, Detection of grapes in natural environment using support vector machine classifier, in: Proceedings of the 21st International Conference on Soft Computing MENDEL 2015, Brno University of Technology, Brno, Czech Republic, 2015, pp. 143–150.
[27] J.J. Staal, M.D. Abramoff, M. Niemeijer, M.A. Viergever, B. van Ginneken, Ridge based vessel segmentation in color images of the retina, IEEE Trans. Med. Imaging 23 (4) (2004) 501–509.
[28] Herng-Hua Chang, Audrey H. Zhuang, Daniel J. Valentino, Woei-Chyn Chu, Performance measure characterization for evaluating neuroimage segmentation algorithms, NeuroImage 47 (1) (2009) 122–135.
[29] Pavel Škrabánek, Petr Doležel, On reporting performance of binary classifiers, Sci. Papers Univ. Pardubice Ser. D XXIV (2017) 181–192.
[30] Marina Sokolova, Guy Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manage. 45 (4) (2009) 427–437, http://dx.doi.org/10.1016/j.ipm.2009.03.002.
[31] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, Silvio Savarese, Generalized intersection over union, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[32] L. Čehovin, A. Leonardis, M. Kristan, Visual object tracking performance measures revisited, IEEE Trans. Image Process. 25 (3) (2016) 1261–1274, http://dx.doi.org/10.1109/TIP.2016.2520370.
[33] F. Candocia, M. Adjouadi, A similarity measure for stereo feature matching, IEEE Trans. Image Process. 6 (10) (1997) 1460–1464, http://dx.doi.org/10.1109/83.624977.
[34] Kuk-Jin Yoon, In So Kweon, Distinctive similarity measure for stereo matching under point ambiguity, Comput. Vis. Image Underst. 112 (2) (2008) 173–183, http://dx.doi.org/10.1016/j.cviu.2008.02.003.