Advanced Feature Extraction Workflow for Few Shot Object Recognition

Markus Brüning, Paul Wunderlich, and Helene Dörksen

Abstract

Object recognition is well known to have a high importance in various fields. Example applications are anomaly detection and object sorting. Common methods for object recognition in images divide into neural and non-neural approaches: Neural-based concepts, e.g. using deep learning techniques, require a lot of training data and involve a resource intensive learning process. Additionally, when working with a small number of images, the development effort increases. Common non-neural feature detection approaches, such as SIFT, SURF or AKAZE, do not require these steps for preparation. They are computationally less expensive and often more efficient than the neural-based concepts. On the downside, these algorithms usually require grey-scale images as an input. Thus, information about the color of the reference image cannot be considered as a determinant for recognition. Our objective is to achieve an object recognition approach by eliminating the "color blindness" of key point extraction methods by using a combination of SIFT, color histograms and contour detection algorithms. This approach is evaluated in context of object recognition on a conveyor belt. In this scenario, objects can only be recorded while passing the camera's field of vision. The approach is divided into three stages: In the first step, Otsu's method is applied among other computer vision algorithms to perform automatic edge detection for object localization. Within the subsequent second stage, SIFT extracts key points out of the previously identified region of interest. In the last step, color histograms of the specified region

M. Brüning (✉) · P. Wunderlich · H. Dörksen
inIT – Institute Industrial IT, Technische Hochschule Ostwestfalen-Lippe, Lemgo, Deutschland
e-mail: markus.bruening@th-owl.de

P. Wunderlich
e-mail: paul.wunderlich@th-owl.de

H. Dörksen
e-mail: helene.doerksen@th-owl.de

V. Lohweg (Hrsg.), *Bildverarbeitung in der Automation*, Technologien für die intelligente Automation 17, https://doi.org/10.1007/978-3-662-66769-9_4

are created to distinguish between objects that feature a high similarity in the extracted key points. Only one image is sufficient to serve as a template. We are able to show that developing and applying a concept with a combination of SIFT, histograms and edge detection algorithms successfully compensates the color blindness of the SIFT algorithm. Promising results in the conducted proof of concept are achieved without the need for implementing complex and time consuming methods.

Keywords

Object recognition · SIFT · Color histograms · Computer vision · Few shot

1 Introduction

Recognizing objects based only on few or even one samples gained a lot of attention in recent years and is currently a hot topic in computer vision and machine learning [1]. There are numerous fields of application in which recognizing objects based on few images is desired and already under investigation: In dermatological disease diagnosis within the medical domain few-shot learning is applied to support doctors based on few given examples [2]. In the agriculture domain, the classification of healthy and diseased plants is of crucial importance as it preserves and improves the yield [3].

Achieving fast and reliable object recognition having only one or few images is also of interest for industrial applications: In case of customized products in small batch series production with reaching a "lot-size-of-one" only very few images can be taken after assembly [4]. In such a case, images of the customized items cannot be provided in the run-up to learn a deep-learning based classifier.

There are several reasons why in some cases only few data exists [1]:

1. Imitate the way humans learn: Only providing that much data that a human would require
2. Cases in which events only rarely occur
3. Reducing the amount of data subsequently reduces the data gathering effort as well as the computational cost

The target use-case of this work is object recognition applied in a conveyor belt system. A concept drawing of the aforementioned conveyor belt example is shown in Fig. 1: A set of objects A, B, C, ... moves along a conveyor with an off-the-shelf camera mounted on top of it. While the object travels on the belt it passes the cameras field of vision. During this time, images of the object can be recorded. In a second run, detected objects are checked for similarity with the previously recorded set of objects: It can be detected which objects have been recorded before and their position can be estimated.

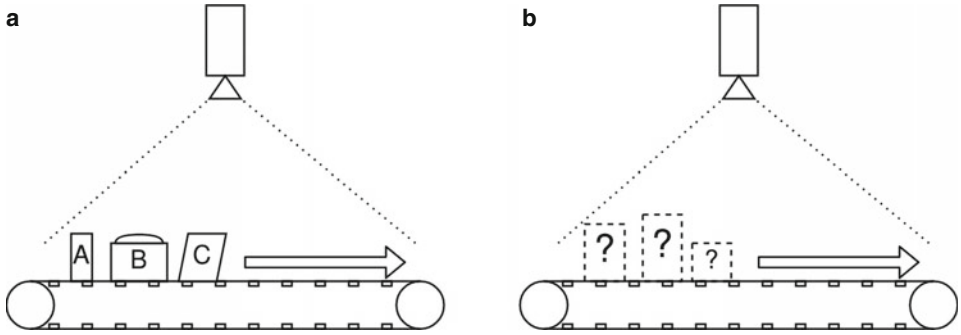

Fig. 1 The baseline scenario for this work: A "learning-phase" (**a**) and a "recognition-phase" (**b**).

1.1 State of the Art

The resulting major challenge of this scenario is the small number of images that can be recorded. Object detection methods in general can be divided into two architectural approach categories:

- **Neural:** Deep-learning methods, one- and two-stage detectors like RCNN [5], YOLO [6] and SSD [7] [8].
- **Non-neural:** Histogram of Oriented Gradients (HOG) [9], the Viola Jones Detector [10], Scale Invariant Feature Transform (SIFT) [11] and others [8].

Neural and non-neural approaches both have individual advantages and disadvantages: Non-neural methods do not have a requirement for training data or complex neural networks. On the other hand, the processing time for decision making might be longer [12]. Neural approaches however might deliver more accurate results (especially on challenging backgrounds) but require a larger training dataset [12].

When working with few or even only one image(s) common off-the-shelf deep learning methods cannot be applied due to the fact that deep learning does not perform well on smaller datasets [13]. The challenge of building a classifier only based on very few images is called *few-shot learning*. A related learning problem for this task category is called transfer learning in which knowledge is transferred from a domain with has sufficient data available [1]. In addition it can be checked if the dataset can be artificially enlarged using *data augmentation* which adds different kinds of invariance to the available images.

Regarding the non-neural based approaches, popular feature extraction systems like SIFT [11], SURF [14] or AKAZE [15] have a common drawback: Besides the ability of successfully identifying and localizing distinctive features in images, many methods of this category require grey-scale images as an input for further processing. This obviously abstracts away valuable information about the coloration present in an image.

Especially the neural based methods gained a lot of interest and development in recent years. Specialized few-shot learning (FSL) [1] and one-shot learning (OSL) [13] approaches made great progress but increased the complexity and engineering effort. The first idea of one-shot learning has been investigated by [16] in 2006. There are several approaches to tackle few-shot learning applications. They usually require some kind of prior knowledge which is used on different perspectives like the *data*, the *model* or the *algorithm* [1].

1.2 Related Work

The color-blindness of feature description algorithms like SIFT is no novelty in research and has been similarly investigated before: Suhasini et al. presents an approach for image retrieval using Invariant Color Histograms. The authors use the HSV instead of the RGB color space [17]. In [18] Chang et al. uses color coocurrence histograms (CH) for recognizing objects in images. Color-CH give information about the separation distance of pairs of colored pixels. This is an addition to normal color histograms, as these do not contain information about geometry features. The authors show successful object recognition on cluttered background, partial occlusions and flexing of the object. Ancuti and Bekaert identified that SIFT has proven to be the most reliable descriptor but is vulnerable to color images [19]. In this work color coocurrence histograms are also used combined with the SIFT approach. The results in context of image matching outperform the original version, detecting an additional number of correct matched feature points.

1.3 Research Question

The research question and the subsequent aim of this work is how a simple object recognition system can be realized without using prior knowledge. Regarding the usage of SIFT this paper evaluates a method for extending SIFT with using coloration information as an additional deciding factor.

To pick up the conveyor belt example from above (shown in Fig. 1) the following challenges are identified:

1. *Few images:* Due to the short recording time on the conveyor belt
2. *Plain background*
3. *Low variation:* Objects are only visible from one viewpoint
4. *Unknown class of objects:* No dataset or prior knowledge from related problems is available

The system should efficiently recognize objects in plain images by only providing few or one image as a template. Due to the usage of established image processing the system

is able to run on hardware with low computational power, instead of requiring expensive hardware components like GPUs.

In order to investigate the questions and requirements, the following chapter proposes an approach by presenting the concept and details of an implementation. The subsequent chapter contains an experiment on a test-dataset and states its results. The last chapter concludes the work and gives an outlook on the topic.

2 Approach

2.1 Concept

To estimate the distinctive textural and shape features of an object present in an image, we choose the well established *SIFT algorithm*. It is an object recognition system that uses local images features which are invariant to scaling, translation, rotation and partially invariant to illumination changes [11]. Reasons for choosing this algorithm are superior results in comparative analysis [20].

An additional tool is used for the object recognition. The creation of *color histograms* represents the pixel-wise color distribution within an image.

The main steps in the presented workflow are depicted in Fig. 2. It represents a shortened version of the full workflow of Fig. 5:

In the **image capture and preprocessing** step the input image is taken by a commercially available camera with a resolution of 640x480 px. The choice of the camera type is arbitrary, as long as the image of the saved reference objects have been taken with the same camera to match the resolution and possible coloration shifts. A standard USB-webcam, a smartphone camera as well as a virtual image feed have been tested as input devices. The preprocessing separates the object's fore- and background of the image and creates a binary mask. This region-of-interest (ROI, the area containing the object) masks out the part of the image that is irrelevant for detection. Then, key point-descriptors of the ROI are **extracted using SIFT**. The found descriptors are subsequently matched with the available templates. This is the first deciding factor for classification. If there are multiple objects that feature a high similarity (from now on called "candidates"), the decision is ambiguous and a **color histogram** of the ROI is created. It is similarly compared with the template images. Thus, the histograms serve as an "arbiter". The final **decision** or assignment is firstly based on the result of SIFT and in a case of multiple candidates the result of the histogram comparison is made use of.

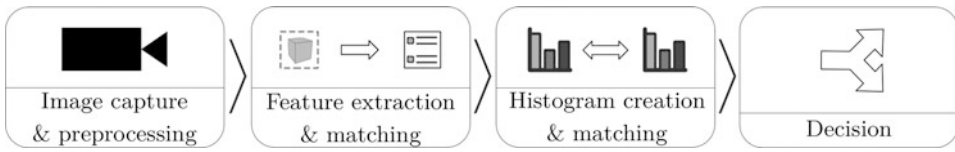

Fig. 2 Summarized programming flowchart.

2.2 Detailed description

The following description refers to Fig. 5. Text in bold notation points to the headings on the right-hand side. The workflow starts with the **preprocessing and image capture** including an initialization of the "known" objects. These are images of objects that have been captured before and are stored as image files. For every object a binary mask is created as explained before. A color histogram of the area provided by the mask is created. This is done for later use before the recognition loop to reduce processing time while detecting. As the color of the background is likely to be captured by the histogram, the corresponding color components have to be excluded from every objects histogram. This is achieved by creating a mask containing only the area of the object. This eliminates the capturing of unnecessary pixels.

The effect of not masking the color components of the background tested is demonstrated on three images shown in Fig. 3. A reference object (a) is compared with a rotated and translated representation of the object (b). Image (c) shows a similar object with a slight color-variation in some parts. The results of Table 1 show the histogram similarity derived from the calculated distance.

After the successful masking the loop is entered starting with image capturing. This begins with receiving an image from a simple USB-camera for example. The contour detection now tries to detect objects within the image. A successful detection provides the region-of-interest which contains the object. If none is found, the loop is iterated-through until a ROI is found. To precisely locate the object, the boundaries and contour of the object have to be located. Therefore, the contour is extracted by applying a method of topological structural analysis using border following [21]. From the hierarchical output

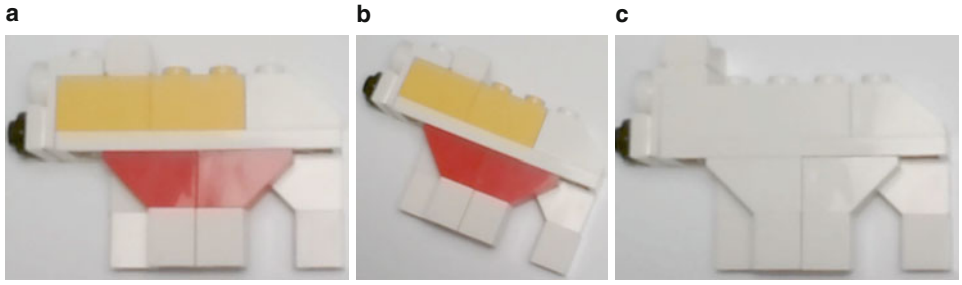

Fig. 3 (**a**) Reference image, (**b**) Reference image rotated and translated, (**c**) Differently colored object.

Table 1 Comparison and Similarity without masking

(a) compared to …	(b)	(c)
Without masking	48%	47%
With masking	56%	30%

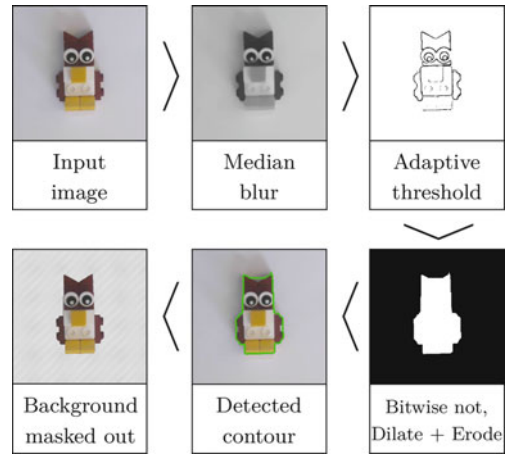

Fig. 4 The image preparation workflow.

only the outermost contour (the "parent") is used to limit the area the object appears in. The technical details of this image preparation is depicted in Fig. 4.

In detail, the image preparation workflow of Fig. 4 is realized as follows: The input image, in this case the owl figure, is converted to greyscale and a blurring filter is applied for noise reduction. Then, adaptive thresholding is used to extract the edges of the object. Methods without an adaptive property, like Canny Edge Detection [22], are prone to require a manual setting of parameters in order to detect the edges properly. The output of the thresholding step is subsequently inverted via a bitwise-not function. The application of two morphological operations, namely dilating and eroding, again reduces noise. The resulting image distinguishes the objects' area (indicated by white pixels) from the background (represented by black pixels). Up to this point, this black-and-white image represents a mask dividing fore- and background. The topological structural analysis using border following [21] is now easily applied on the prepared image. The outermost contour (the "parent") gives information about the objects border that is used to create the bounding-box. Therefore, the smallest and greatest x- and y-coordinates of the detected contour form the top-left and the bottom-right corner of the box.

The **feature extraction and matching** is performed using SIFT. The extraction procedure is restricted to the region-of-interest provided by the bounding box of the masked area. If only few (due to noise) or no features were found by SIFT it is assumed that no object is present in the region. Otherwise, the feature descriptors are matched with the ones from the list of known objects. The matches are stored in a "score list". This list is subsequently sorted ascending with the highest scores.

Now, candidates are appointed with the requirement of featuring a similarity of at least 50% in order to make a **decision**. This parameter is defined as similar and is chosen freely. If no candidate has been nominated, the object is seen as "unknown". If there is only one candidate it is a distinct decision. The case of having multiple objects (≥ 2) sharing a high similarity estimated by SIFT is determined by analyzing the coloration-in-

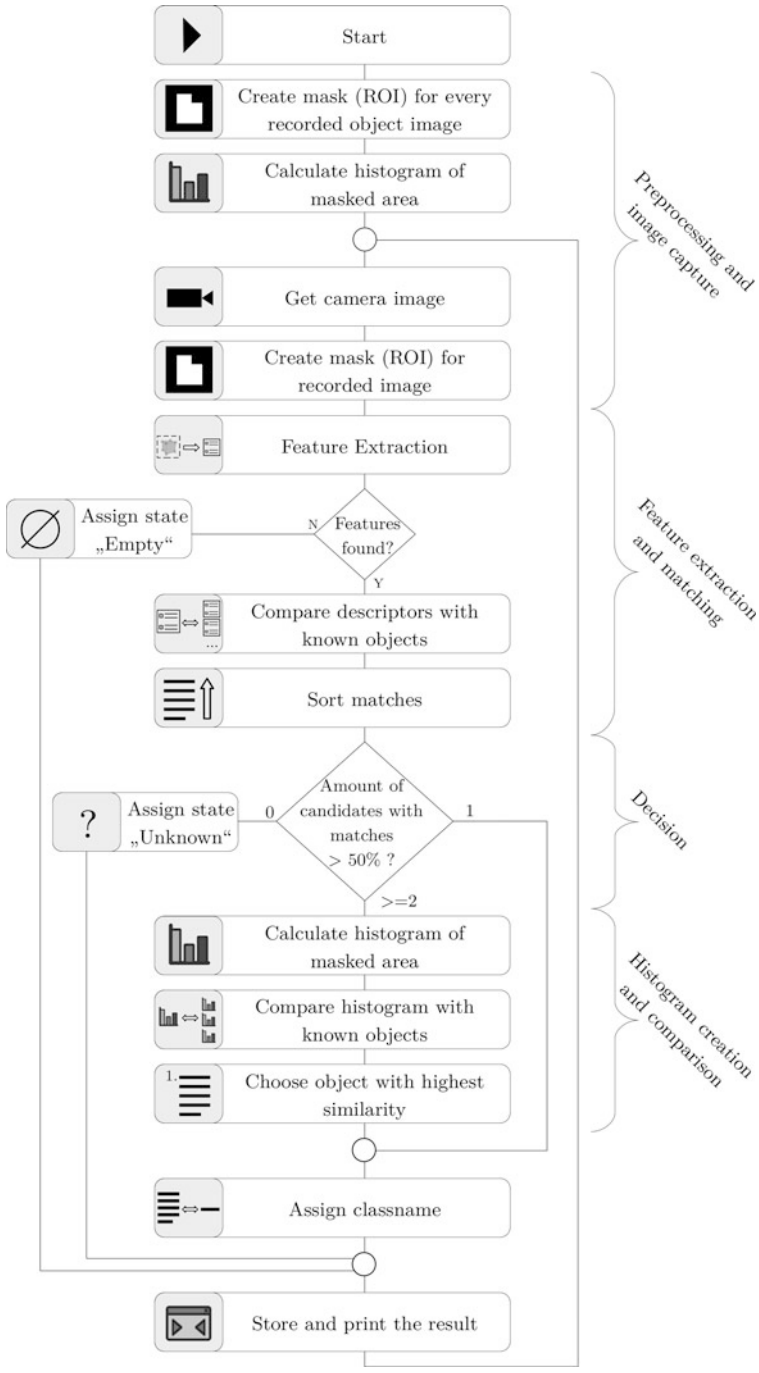

Fig. 5 Extended programming flowchart.

formation. Therefore, a **color histogram** of the recent image is created with the additional background mask as seen before. The histogram of the current image is compared with the ones calculated in the beginning and the results are stored in a list. This list is also sorted based on the scoring. Now, the object corresponding to highest score is estimated to be the match for the newly seen object.

2.3 Concept Drawing

See Fig. 5.

3 Experiments and Results

To validate the added color-variance of the SIFT algorithm and the overall functionality within an object recognition system a proof-of-concept is conducted. The objects themselves used in this context are small Lego® figures. They are originally "produced" in the SmartFactoryOWL[1] to demonstrate the workflow of a cyber-physical-system. For this scenario it is assumed that the bricks of the figures can be chosen in individual ways to fit a customers need.

Fig. 6 shows an overview of the dataset used in the experiment.

The dataset consists of a sum of 30 images representing 10 classes. Each class is captured three times: One image as depicted in Fig. 6 and two images slightly shifted and rotated by 45° and 180°. Four classes within the dataset are additionally present with minor color changes. This is done to challenge the recognition system: These objects a likely

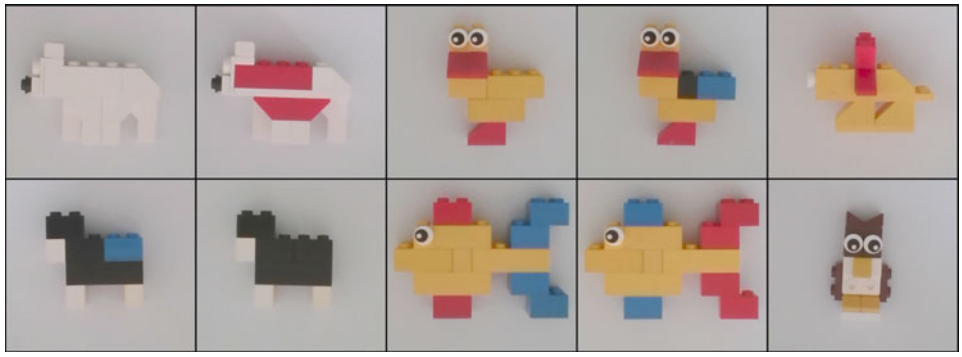

Fig. 6 An image of every class of the dataset (from left to right, top to bottom): A polar bear (2), duck (2), lion, sheep (2), fish (2) and an owl.

[1] https://smartfactory-owl.de

	bear_a	bear_b	duck_a	duck_b	fish_a	fish_b	lion	owl	sheep_a	sheep_b
bear_a	**46**	**40**	15	12	17	21	18	16	22	18
bear_b	**25**	**41**	18	12	16	17	13	15	22	20
duck_a	14	12	**32**	**38**	25	28	12	17	20	13
duck_b	17	12	**32**	**36**	22	27	11	13	19	17
fish_a	19	15	17	16	**35**	**43**	21	22	21	22
fish_b	20	15	17	16	**39**	**32**	17	22	22	20
lion	20	14	18	17	21	18	36	25	23	27
owl	12	6	11	11	11	14	11	45	8	9
sheep_a	28	21	21	23	29	35	22	27	**37**	**45**
sheep_b	26	22	22	22	26	30	23	29	**48**	**41**

Fig. 7 A similarity matrix of SIFT applied on the dataset. The objects refer to Fig. 6. Suffix "_a" denotes a normally colored object and "_b" a variant with slightly altered colors. Results of these objects are framed in a black box. All similarities in percent.

to look similar when observed in grey-scale, but show variations when analyzing the coloration.

The results are represented in form of a classification results matrix. Every object image is compared to every other object. The matrix entries represent similarities and are calculated as follows:

$$\text{SIFTsimilarity} = (\text{KeypointMatches}/\text{TotalKeypoints}) \tag{1}$$

$$\text{HistogramSimilarity} = (1 - \text{HistogramDistance}) \tag{2}$$

$$\text{Similarity} = (\text{SIFTsimilarity} + \text{HistogramSimilarity})/2 \tag{3}$$

The first matrix depicted in Fig. 7 shows how SIFT performs on the provided dataset. The calculated similarities between the objects with color variations (_a and _b) are generally very close but the highest similarity often points to the wrong object leading to a false classification. The difference towards the other classes is sufficient in order to tell these apart.

Evaluating the results of applying a combination of SIFT and color histograms reveals more distinctive decisions in the matrix of Fig. 8. The classes with the color variant feature a higher distance towards each other. The matrix shows that in every case the highest similarity belongs to the correct class, even though rather closely for some cases. The boundary towards the different classes is more distinctive as well. This is indicated by the more reddish coloration within the matrix.

	bear_a	bear_b	duck_a	duck_b	fish_a	fish_b	lion	owl	sheep_a	sheep_b
bear_a	**71**	**47**	19	17	17	19	19	32	30	27
bear_b	**40**	**69**	30	27	25	22	23	30	27	25
duck_a	18	27	**63**	**53**	49	44	48	33	21	16
duck_b	20	27	**50**	**65**	50	52	34	34	26	28
fish_a	18	25	45	47	**66**	**62**	46	31	18	22
fish_b	18	22	38	46	**60**	**65**	38	32	21	25
lion	20	24	51	38	46	39	65	33	18	19
owl	30	26	30	33	26	28	26	70	36	32
sheep_a	33	27	21	28	22	28	17	46	**75**	**59**
sheep_b	31	26	21	31	24	30	16	42	**60**	**72**

Fig. 8 A similarity matrix of the presented workflow applied on the dataset. All similarities in percent.

4 Conclusion and Outlook

Although the used algorithms are rather old in terms of image processing approaches, they have proven to still be useful and beneficial for the evaluated are of application.

In general, working with a low amount of images, in the "few-shot learning" domain, is still a relatively new topic in machine learning. Common state-of-the-art methods do not perform well on smaller datasets and especially may have problems with uni-colored backgrounds due to the risk of overfitting.

Additionally, many approaches in few-shot learning require some kind of prior knowledge, for example regarding the data, model or algorithm [1]. Therefore, a detailed analysis of the environment by an expert is required in order to investigate the availability of similar datasets. All in all, using neural-approaches often results in lots of engineering to find the right models and parameters. We may see further development in the future.

The experiment conducted in this work shows a significant improvement compared to a SIFT-only-based classification. The "challenges"[2] included in the dataset resulted in a low number or ambiguous matches when only SIFT is applied. The proposed method of this work increased the number of correctly classified objects compared in two similarity matrices. The effect on the reduced dataset approaches zero, as it only includes heterogeneous objects which SIFT can successfully distinguish.

But the results also state that using histograms in addition to key point detection is no cure-all solution to determining small variations in color. Little variations in the color components due to illumination changes or other influences during recording can decrease the number of correctly classified objects. This is likely to occur in this case, as no professional equipment was used for recording.

[2] Objects featuring a high degree of similarity but slightly differently colored

On the upside, the proposed classification workflow was achieved by using only lightweight methods without the need of a training stage or a dataset for learning purposes. This is especially attractive for the usage on resource limited hardware. All in all, the workflow presented in this work offers several advantages towards deep-learning methods but offers room for improvement in detecting small coloration changes.

References

1. Wang Y, Yao Q, Kwok JT, Ni LM (2021) Generalizing from a few examples. ACM Comput Surv 53:1–34
2. Prabhu V, Kannan A, Ravuri M et al (2019) Few-shot learning for dermatological disease diagnosis. In: Proceedings of the 4th Machine Learning for Healthcare Conference 106, S 532–552
3. Nuthalapati SV, Tunga A (2021) Multi-domain few-shot learning and dataset for agricultural applications. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)
4. Jasperneite J, Hinrichsen S (2015) Wandlungsfähige Montagesysteme für die Fabrik der Zukunft. In: VDI-Tagung "Industrie 4.0" (Vortrag)
5. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition
6. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
7. Liu W, Anguelov D, Erhan D et al (2016) Ssd: Single shot multibox detector. In: European conference on computer vision
8. Zou Z, Shi Z, Guo Y, Ye J (2019) Object detection in 20 years: A survey. http://arxiv.org/abs/1905.05055
9. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). https://doi.org/10.1109/cvpr.2005.177
10. Viola P, Jones M (2001) Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR 2001
11. Lowe DG (1999) Object recognition from local scale-invariant features. In: Proceedings of the Seventh IEEE International Conference on Computer Vision
12. Subhashini D, Dutt SI (2020) A review on road extraction based on neural and non-neural networks. Int J Eng Res V9:1306–1309
13. Jadon S, Garg A (2020) Hands-on one-shot learning with python: Learn to implement fast and accurate deep learning models with fewer training samples using pytorch. Packt, Birmingham
14. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (surf). Comput Vis Image Underst 110:346–359
15. Alcantarilla P, Nuevo J, Bartoli A (2013) Fast explicit diffusion for accelerated features in nonlinear scale spaces. In: Procedings of the British Machine Vision Conference 2013
16. Fei-Fei L, Fergus R, Perona P (2006) One-shot learning of object categories. IEEE Trans Pattern Anal Machine Intell 28:594–611
17. Suhasini PS, Krishna KS, Krishna IV (2012) Combining sift and invariant color histogram in HSV space for deformation and viewpoint invariant image retrieval. In: 2012 IEEE International Conference on Computational Intelligence and Computing Research

18. Chang P, Krumm J (1999) Object recognition with color cooccurrence histograms. In: Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat No PR00149)
19. Ancuti C, Bekaert P (2007) SIFT-CCH: increasing the SIFT distinctness by color co-occurrence histograms. In: 2007 5th International Symposium on Image and Signal Processing and Analysis
20. Tareen SA, Saleem Z (2018) A comparative analysis of SIFT, surf, Kaze, AKAZE, Orb, and brisk. In: 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)
21. Suzuki S, Abe K (1985) Topological structural analysis of digitized binary images by border following. Comput Vis Graph Image Process 29(3):396
22. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell PAMI-8:679–698