Multi-class Fruit Classification using RGB-D Data for Indoor Robots

Lixing Jiang, Artur Koch, Sebastian A. Scherer and Andreas Zell

***Abstract*— In this paper we present an effective and robust system to classify fruits under varying pose and lighting conditions tailored for an object recognition system on a mobile platform. Therefore, we present results on the effectiveness of our underlying segmentation method using RGB as well as depth cues for the specific technical setup of our robot. A combination of RGB low-level visual feature descriptors and 3D geometric properties is used to retrieve complementary object information for the classification task. The unified approach is validated using two multi-class RGB-D fruit categorization datasets. Experimental results compare different feature sets and classification methods and highlight the effectiveness of the proposed features using a Random Forest classifier.**

I. Introduction

A. Motivation

Object category classification and recognition is an essential and widely studied task in computer vision. Since the introduction of low-cost RGB-D sensors like the Microsoft Kinect, the demand for RGB-D-based approaches has become even more universal. By utilizing the additional depth-information and derived features, identification of objects becomes even more precise and thus more feasible for practical applications [1]–[6].

Inspired by those advances, we utilize a mobile service robot as an identification system for fruits. A simple practical scenario would be a supermarket, where the robot should be able to deduce the prices of different fruits without additional input of the customers.

The task at hand involves two main goals. Firstly, from a technical point of view, we extend the setup of our existing mobile platform (MetraLabs Scitos G5, see Fig. 1) to provide RGB-D data for the fruits to be classified. Secondly, we build an accurate and robust fruit classification system tailored to utilize the RGB-D data of our specific hardware setup.

Hence, we firstly introduce our segmentation process, which utilizes background subtraction on RGB and depth image data. Then, we review several visual feature descriptors and 3D intrinsic shape measures [3], [7]–[10]. Afterwards, we give details on our feature vector, which fuses color, texture, geometry and 3D shape measures for a compact representation of essential RGB-D data characteristics. Compared to single visual feature descriptors or geometrical features, the combined descriptor provides complementary cues about the fruit category even under varying lighting conditions. Finally, we apply several machine learning (*ML*) methods and evaluate their performance. The results show that combining visual color and depth cues can noticeably improve the accuracy.

L. Jiang, A. Koch and S.A. Scherer are with the Chair of Cognitive Systems, headed by Prof. A. Zell, Computer Science Department, University of Tuebingen, Sand 1, D-72076 Tuebingen, Germany {lixing.jiang, artur.koch, sebastian.scherer, andreas.zell}@uni-tuebingen.de

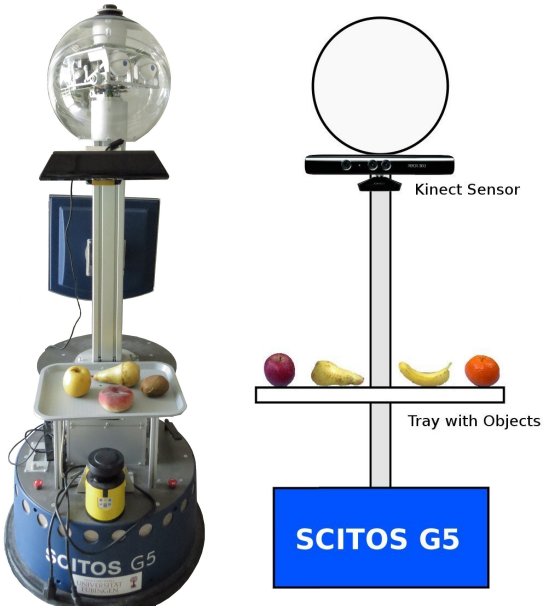
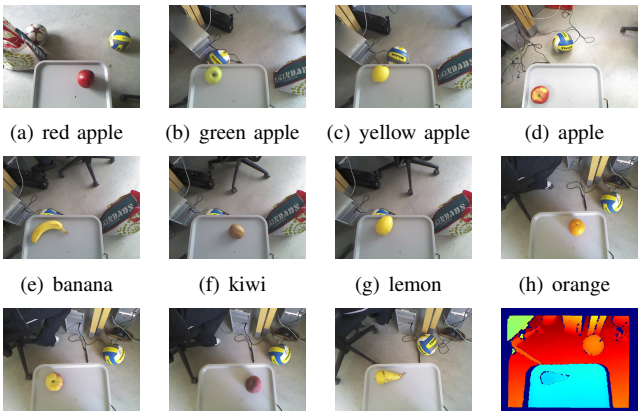


Fig. 1. SCITOS G5 robot equipped with a tray and the Microsoft Kinect.

The remainder of the paper is structured as follows: Section I-B gives an overview of the related work. In Section II we present the modifications to the hardware of our mobile robot. Section III introduces the datasets we acquired for training and evaluation. After describing our segmentation approach in Section IV we present and review several visual descriptors and 3D intrinsic shape measures in Section V. Finally, we present the results obtained from different classifiers in Section VI and conclude the paper in Section VII.

B. Related Work

Although there have been enormous advances in object classification and recognition in computer vision and robotics in recent years, it still represents a challenging field of study [1], [2], [4], [5], [8], [10]–[12]. Previous fruit classification and recognition approaches applied global low-level visual features in color, edge and texture properties [12], [13]. Rocha et al. [12] proposed an automatic fruit and vegetable classification system using color, texture, shape and local appearance features in 2D RGB images. They reported an error rate of approximately 3%, thus being able to correctly classify 15 different fruit and vegetable categories at an accuracy of 97% on 2,633 image samples using Linear Discriminant Analysis (LDA).



(a) red apple (b) green apple (c) yellow apple (d) apple

(e) banana (f) kiwi (g) lemon (h) orange

(i) saturn peach (j) peach (k) pear (l) depth

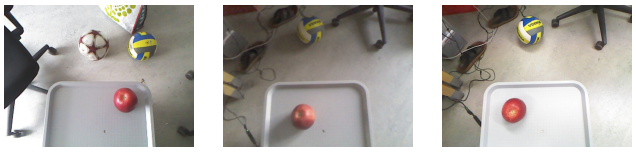
Fig. 2. Raw RGB(-D) image data samples of fruits.

As an alternative to global low-level features, several object classification or recognition schemes build on more sophisticated local features (e.g. [14], [15]) that focus on the description of a local neighbourhood around a point of interest. Although local features show a good performance regarding textured objects or regions of interest, they do not perform well for homogeneous regions. To address this problem, several approaches extend to RGB-D data. Hinterstoisser et al. [6] presented a template matching method for the detection of textureless objects. Karpathy et al. [3] provided a method for discovering object models from 3D meshes in indoor environments. For this purpose, they propose different intrinsic shape measures to achieve a good segmentation result. Lastly, Lai et al. [2] published a large scale hierarchical RGB-D object dataset that contains several different object classes, e.g. fruits and vegetables, and may be used for performance evaluation.

II. Technical Setup

As shown in Fig. 1, the development platform used for this paper is a Scitos G5 service robot from MetraLabs. It is equipped with a laser scanner that is used for Monte-Carlo-based self-localization and an integrated PC for on-board processing. Additionally, the touchscreen is used for IO-tasks and human-machine interaction.

For our work, we extended the experimental platform with a gray plastic tray, where the object samples (i.e. fruits) to be recognized can be put on. The tray is mounted at a height of approximately $0.6\,m$ parallel to the ground plane and serves as the general experimental area. Additionally, a Microsoft Kinect was mounted at a vertical distance of approximately $0.5\,m$ orthogonal (i.e. pointing downwards) to the tray. The Kinect RGB-D sensor concurrently records both color and depth images at a resolution of 640×480 pixels with 30 frames per second. Since the off-the-shelf Kinect XBox 360 sensor specifies a minimum distance of approximately $0.8\,m$ for the retrieval of depth data, we equipped the robot with a Kinect for Windows that supports the so-called near mode. With near mode enabled, the Kinect for Windows provides depth data for objects at a minimum distance of $0.4\,m$ without loss in precision.



(a) daylight (b) dim lighting (c) artificial light

Fig. 3. Same apple at different poses and lighting conditions.

This specific setup gives the test subjects (e.g. customers) an intuitive access for the placement of fruit samples. Furthermore, it provides us with the desired image samples that are necessary for the classification task, as can be seen in Fig. 2. And it finally also introduces some constraints that can be exploited in the upcoming segmentation task.

III. Datasets

A. Subset of External Object RGB-D Dataset

To test feature descriptors and the classifiers, we carried out all our experiments on the fruit samples of the object RGB-D dataset proposed by Lai et al. [2]. The dataset overall includes 32 instances of seven different fruit categories with 21,284 image samples.

B. Collected Fruit RGB-D Dataset

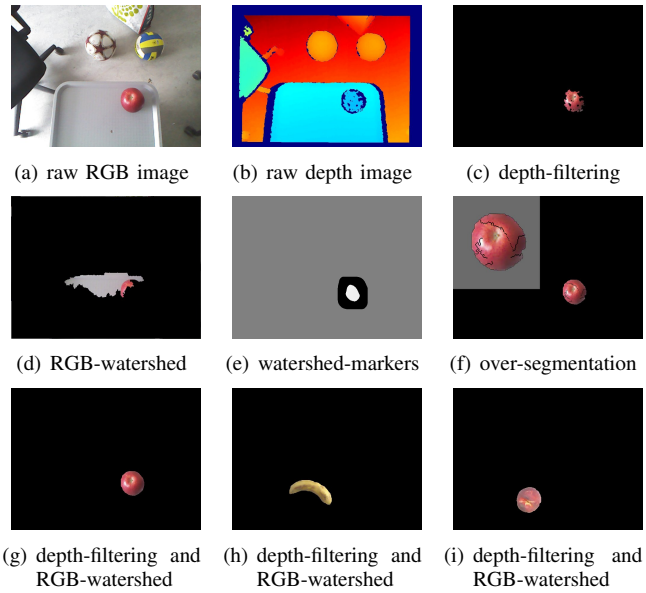
To prove the effectiveness of our approach on our actual robot, we collected our own fruit RGB-D dataset using our system described in Section II. The structure of our dataset includes categories and instance levels which are similar to the larger scale RGB-D object dataset [2]. This dataset builds a set of approximate 330 RGB-D sample images for each of a total of seven categories under three different lighting conditions that include daylight, dim lighting and artificial light in the night. Fig. 3 shows the same fruit in the three different lighting conditions. It can be seen that color, brightness as well as background for the same fruit may vary greatly due to different lighting conditions. The images also show that the poses of the same fruit instances were randomly perturbed, even for subsequent sample images, to be able to analyse performance in regard to variance in orientation and position.

Working on RGB-D datasets, it is essential to correctly align depth and RGB images, to be able to identify correspondencies between depth- and RGB-image pixels, and vice-versa. We use the transformation implemented in hardware within the Kinect before storing the image samples. Our recorded dataset for evaluation consists of 2,333 samples with a resolution of 640×480 pixels in total. Fig. 2 shows some examples of the resulting RGB and depth images.

In the following, we refer to the external dataset as *obj-dataset* and to the collected dataset as *own-dataset*.

IV. Segmentation

The purpose of the segmentation process is to detect object candidates in a complicated scene. An effective segmentation technique may greatly decrease computational complexity for the later stages of the pipeline, since it reduces the input data to only the significant regions that should be processed.



(a) raw RGB image (b) raw depth image (c) depth-filtering

(d) RGB-watershed (e) watershed-markers (f) over-segmentation

(g) depth-filtering and RGB-watershed (h) depth-filtering and RGB-watershed (i) depth-filtering and RGB-watershed

Fig. 4. Segmentation example.

Additionally, prior image segmentation is the basis of region-based feature extraction, since it allows using global features, which afterwards can directly be computed from the preprocessed image. Therefore, we apply a combined segmentation on RGB- as well as depth-image data.

A. Depth Filtering

Filtering out irrelevant background regions is essential for our subsequent work, since it reduces computational complexity greatly and minimizes the search space. Using the specifications of our setup, the goal is to segment out the tray and test objects placed on it, by taking advantage of the known 3D tray shape and its distance to the sensor. Therefore, we first apply a simple pass-through filter on each dimension of the 3D target domain using depth data. The constraints for each dimension can simply be derived from our fixed setup and the a priori known camera pose. The filtered points satisfying the X-, Y- and Z-axis constraints, are considered to belong to the object or the tray.

In the next step we apply RANdom SAmple Consensus (RANSAC) plane fitting to get an estimate of the tray plane represented by its Hessian Normal form. After obtaining the set of points belonging to the plane, we are able to separate the object from the plane and thus get the object's point cloud. Although RANSAC succeeds well in separating the plane from the object, the depth-based filtering itself does not provide adequate results for the segmentation process.

As can be seen in Fig. 4(c), the depth-masked RGB image obtained by projecting the depth-pixels onto the corresponding RGB-pixels exhibits many gaps. This is due to the fact that the depth sensor does not gather valid depth measurements on surfaces that are exposed to too much light, which is a well-known issue of the Kinect. Additionally, sampling artifacts and (re)projection errors may lead to this noise in the depth data. These gaps can generally be observed in the inner region of the object's depth-mask. Another issue can be observed at the outline of the projected 3D-shape. The sensor again is not able to generate depth information, since the surface normals of the object are close to being perpendicular to the directions of the camera rays. Thus, the infrared pattern projected by the sensor is subject to strong distortions in these areas and cannot be recognized.

Unfortunately, basic morphological operators do not provide satisfying results to solve this problem. Therefore, to optimize the result of the segmentation, we pass the depth-mask obtained from depth-filtering to the next segmentation stage building upon RGB data.

B. RGB Segmentation

To refine our previously obtained depth-mask, we apply the watershed transform (F. Meyer et al. [16]), which is a morphological algorithm for image segmentation. It uses a grey-level representation of the input image that may be interpreted as a topographic surface. During the sequential flooding of the minima on the grey value relief, it partitions the gradient image into watershed lines and catchment basins. The result of the watershed transform produces closed object contours and requires low computation times as compared to other more sophisticated vision-based segmentation methods. However, practically, this traditional transform leads to over-segmentation due to noise in the data, as shown in Fig. 4(f). In order to deal with the over-segmentation, we use the marker-based watershed approach in combination with the previously computed depth mask.

The structure of this segmentation method is as follows. At first, we need to define the marker regions as seed nodes, that identify foreground, background and uncertain regions by different labels. As can be seen in Fig. 4(d) and 4(e), this is an important step towards the watershed-based segmentation process, since it has a significant influence on the flooding process. To obtain preliminary foreground and background regions in the image, we perform simple and fast morphology operators (erosion and dilation) on the depth segmentation result. The foreground region is computed by performing erosion on the depth mask, the background boundary is computed by dilation. By taking the union of the complements of the foreground mask and the background mask ($\overline{B \cup F} = \bar{B} \cap \bar{F}$) we get the uncertain area, that is the region of interest for the watershed process.

Using these regions, we apply the classical marker-based watershed transform. Figs. 4(g-i) show segmentation results using the combined RGB- and depth-based segmentation. As can be seen, the method provides good segmentation quality for our fruit data and features low computational costs as compared to other segmentation methods (e.g. GrabCut [17]). After successful segmentation, color, texture and shape descriptors can be extracted to describe the significant object region(s) in the image.

V. Feature Extraction and Object Representation

In automatic object categorization, it is important to get a semantic representation and an understanding of the underlying input data. Because the available data generally consists

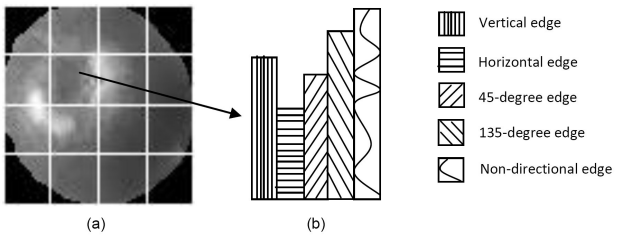


Fig. 5. Edge histogram descriptor.

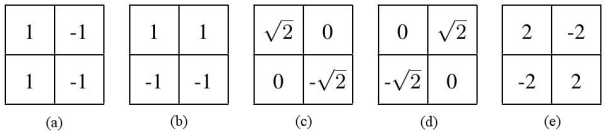


Fig. 6. Filters for edge detection. (a) vertical edge filter, (b) horizontal edge filter, (c) 45° edge filter, (d) 135° edge filter, (e) isotropic edge filter.

of unstructured multidimensional arrays of pixels or voxels represented by 2D RGB images or 3D point clouds, features that are able to characterize images or regions of interest need to be extracted and used. Features usually aim to capture different properties of the image or object represented by the image itself, and the choice of features is of significant importance for the description and recognition of the object. Therefore, in this section, we review color, texture, shape appearance features for RGB image data and intrinsic shape measures for 3D points in order to propose a system to solve a multi-class fruit classification problem using a combined feature vector.

A. Visual Feature Extraction

Our visual feature extraction method focuses on low-level features of the RGB image. The extraction process is carried out by using some descriptors from the MPEG 7 library [7]–[9]. Since the regions obtained by segmentation are generally homogeneous in color and texture, the following descriptors were chosen.

Scalable Color Descriptor (SCD): Color is the most fundamental property of visual content. The herein used descriptors characterize the color distribution inside the object region. The *SCD* [7] is based on color histograms extracted in the hue-saturation-value (*HSV*) color space, which is used for storage efficiency. It can be used as a global color feature by measuring the color distribution over an entire image. The descriptor extraction begins with the calculation of the color histogram with 256 bins. Therefore the Hue (*H*) component is quantized into 16, saturation (*S*) and value (*V*) into 4 bins each. To attain a more efficient encoding, the histograms are compressed using a $1D$ *Haar* transform. By iteratively applying the transform, the dimension of the descriptor can be reduced (e.g. $128, 64, 32, 16$) since the respective Haar-coefficients are used in the representation.

Edge Histogram Descriptor (EHD): Texture is an important feature to describe structure and neighbourhood information in an image. The *EHD* represents the spatial local distribution of edges in the image. A local edge histogram is computed for each of the 16 sub-regions, which are obtained by subdividing the source image into a regular 4×4 grid, see Fig. 5. Five different edge filters are used for feature extraction, as shown in Fig. 6. The resulting descriptor is composed of $5 \times 16 = 80$ values, with each bin representing different semantic information based on the location of the sub-region and the corresponding edge type.

B. Object Shape Representation

Shape clearly also offers important semantic information, since humans can recognize many objects given their shape alone. Shape features working on RGB-D images should utilize area, contour and shape information of an object in both RGB and depth images. In the following, we give a brief overview of the employed shape measures, introduced in the work of Karpathy et al. [3].

Compactness (Co): Characterizes the spherical similarity of $3D$ objects.

Symmetry (Sy): Describes the reflective symmetry along the principal axes for each cropped object.

Local Convexity (LC): Measures the convexity of the object representation in local regions.

Smoothness (Sm): Rewards points with more uniformly distributed neighbouring points in a local region.

These measures form the descriptor of the object's 3D shape (***3DSM***) based on depth data defined as:

$$\mathbf{3DSM} = \{Co, Sy, LC, Sm\}$$

Image Moments (Hu7): Image moments are easily computable scalar values that describe the distribution of pixels belonging to an object and their intensity. They are typically chosen to describe certain geometrical properties of an object. In this, we use seven well-known image moment invariants proposed by Hu [18], since they are invariant to image scale, rotation and (in parts) reflection.

Our final shape descriptor (***SH***) of each object candidate consists of the image moments and the intrinsic shape measures:

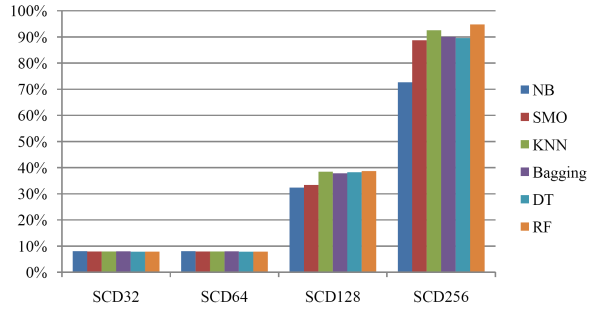
$$\begin{aligned} \mathbf{SH} &= \{Hu7, \mathbf{3DSM}\} \\ &= \{Hu7, Co, Sy, LC, Sm\} \end{aligned}$$

VI. Experimental Results

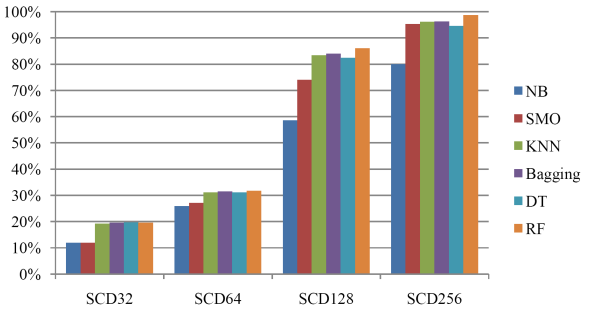
In the quest to discover the best features and classification algorithm, we carried out extensive experiments in order to analyse classification performance based on color, texture and shape image descriptors using multiple mutually exclusive image samples from the two datasets described in Section III. We trained and tested various classifiers using six machine learning algorithms, namely Native Bayes (*NB*), Sequential Minimum Optimization (*SMO*), k-Nearest Neighbors (*KNN*), Bagging based on *REPTree*, Decision Trees (*DT*) and Random Forests (*RF*), in the Waikato Environment for Knowledge Analysis (*Weka*) [19].

A. Evaluation of Different Features

Firstly, we evaluate different scales of the SCD feature descriptor on the previously chosen classifiers. Fig. 7 presents accuracy results of six machine learning approaches with 10-fold cross-validation as a function of the SCD scales on both datasets presented in Section III. The results show



(a) Accuracy of SCD on *own-dataset*.



(b) Accuracy of SCD on *obj-dataset*.

Fig. 7. Classification accuracy of SCD with different scales.

TABLE I

CLASSIFICATION ACCURACY OF EHD ON BOTH DATASETS.

Accuracy(%)	NB	SMO	KNN	Bagging	DT	RF
own-dataset	44.62	68.97	**83.54**	63.14	49.12	81.40
obj-dataset	53.09	75.78	**95.13**	73.46	61.80	91.15

that the SCD descriptor, at low scales, does not perform well in our fruit recognition task. Good classifier results are achieved using at least 128 bins. This is mainly due to lossy compression using the Haar transform, which, as expected, has a negative effect on the classification. For both datasets, the 256 bin version of the SCD using the Random Forest classifier obtains the best classification results.

Furthermore, we compared the EHD descriptor on both datasets. The results are presented in Table I. It can be seen that the texture descriptor solely is unable to achieve satisfying results for our classification task.

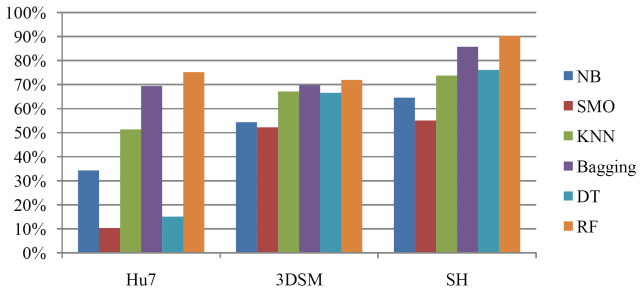
Fig. 8 shows the results of our shape descriptor evaluation. Again, the accuracy is shown as a function of the ML-methods using different shape descriptors including image moments (*Hu7*), 3D shape measures (*3DSM*) and the combined shape descriptor (*SH*). Although both shape features perform far from optimal individually, we can get significantly better results by combining 3D shape information and image moments. The best classification accuracy is again obtained using Random Forests.

B. Evaluation of the Combination of All Features

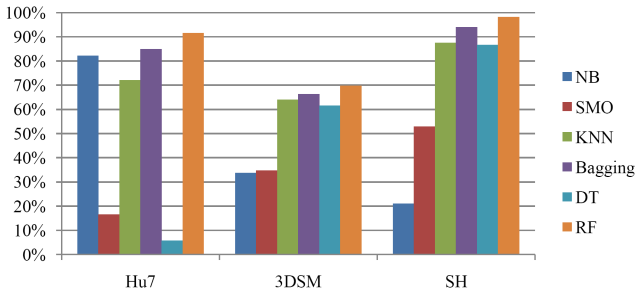
To further improve the classification accuracy, we evaluate the combined descriptor defined as

$$\mathbf{SESH} = \{SCD, EHD, Hu7, SH\}$$

We include color, texture, image moments and 3D shape measures. Fig. 9 shows the results regarding the classification



(a) Accuracy of shape features on *own-dataset*.



(b) Accuracy of shape features on *obj-dataset*.

Fig. 8. Classification accuracy of shape features on both datasets.

TABLE II

CLASSIFICATION ACCURACY OF SHAPE FEATURES ON *own-dataset*.

Accuracy(%)	NB	SMO	KNN	Bagging	DT	RF
Hu7	34.29	10.09	51.35	69.40	15.09	75.18
3DSM	54.39	52.21	67.12	69.65	66.61	71.97
SH	64.55	55.04	73.73	85.73	76.08	**90.27**

TABLE III

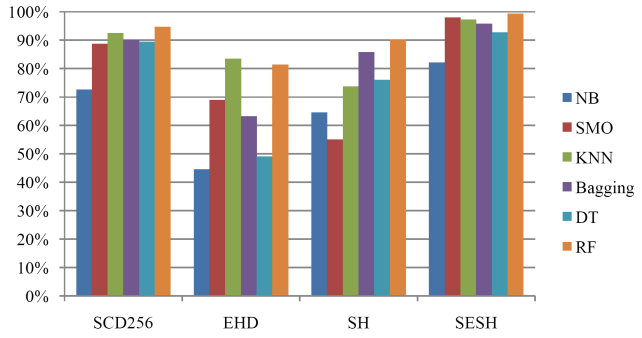
CLASSIFICATION ACCURACY OF SHAPE FEATURES ON *obj-dataset*.

Accuracy(%)	NB	SMO	KNN	Bagging	DT	RF
Hu7	15.15	16.64	72.04	84.89	5.76	91.49
3DSM	33.66	34.67	64.07	66.31	61.50	69.81
SH	20.99	52.84	87.53	94.00	86.67	**98.20**

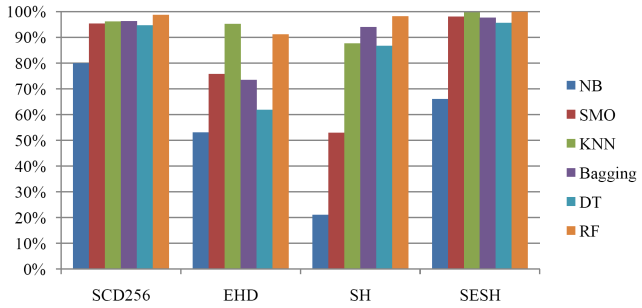
accuracy of different classifiers using SESH. For comparison, also the best performing single descriptors are visualized in the graph. As can be seen, the SESH descriptor outperforms all our tested classifier/descriptor combinations. Best results are achieved using Random Forests, where the accuracy rises up to 99.36% and 99.91% on both datasets, respectively.

C. Comparison on Datasets

The accuracy on our *own-dataset* is lower than that of the *obj-dataset* in all of our experiments. After analyzing the results in more detail, we found out that the similarity between the yellow apple and the lemon (see Fig. 2(c) and 2(g)) is often suspect to misclassification. Even for the human perception system, both samples are not easy to be distinguished from each other regarding the image data. For results with a very high accuracy ($\gtrapprox 95\%$), more than 50% of the errors arise through the (mis-)classification of the yellow apple as the lemon. The rest of the errors is dominated by inter-category level errors ($\geq 40\%$), where e.g. a peach instance is identified as another peach instance inside the



(a) Accuracy of combined features on *own-dataset*.



(b) Accuracy of combined features on *obj-dataset*.

Fig. 9. Classification accuracy of single descriptors and combined feature-vector on both datasets.

TABLE IV

CLASSIFICATION ACCURACY OF COMBINED DESCRIPTOR (SESH) ON BOTH DATASETS.

Accuracy(%)	NB	SMO	KNN	Bagging	DT	RF
own-dataset	82.12	98.03	97.26	95.76	92.71	**99.36**
obj-dataset	66.03	98.02	99.58	97.65	95.61	**99.91**

same category. With the remainder of the errors being of cross-category type, where a fruit is recognized as another fruit at very low error-rates ($\leq 10\%$).

VII. CONCLUSION

This paper approaches the multi-class classification of fruits on a mobile robot. For this purpose, we investigated the performance of different descriptors under varying sample conditions (e.g. pose and lighting) to find the best feature descriptor and ML-method for our fruit recognition task. We introduced our RGB-D-based segmentation approach with a focus on robustness and speed as well as the specific technical setup of our service robot. The results of our segmentation prove its effectiveness and that we can robustly separate the desired fruit object candidates in a complicated scene at low computational costs. Importantly, we present a unified RGB-D feature descriptor that combines low-level RGB features and 3D shape information. We evaluated different combinations of state of the art classifiers and several feature descriptors on both RGB-D datasets. The results demonstrate that the combined RGB-D descriptors are highly suited for our fruit recognition task. Furthermore, we observe that the Random Forest classifier is the best choice, although it was slightly outperformed by kNN for the EHD descriptor. The classification process using Random Forests with the proposed combined descriptor (SESH) delivers a peak accuracy of over 99% on both experimental datasets, which is very promising regarding further applications and extensions of our approach.

As future work, we plan to enlarge the RGB-D dataset and to further analyse other features and texture descriptors. Additionally, we intend to make the collected RGB-D fruit dataset accessible to the research community.

REFERENCES

[1] R. Socher, B. Huval, B. Bhat, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Advances in Neural Information Processing Systems(NIPS)*, 2012.

[2] L. Kevin, B. Liefeng, R. Xiaofeng, and F. Dieter, "A large-scale hierarchical multi-view rgb-d object dataset," in *IEEE International Conference on on Robotics and Automation(ICRA)*, (Shanghai, China), pp. 1817–1824, 2011.

[3] A. Karpathy, S. Miller, and L. Fei-Fei, "Object discovery in 3d scenes via shape analysis," in *IEEE International Conference on Robotics and Automation(ICRA)*, (Karlsruhe, Germany), pp. 290–294, May 2013.

[4] J. Fischer, R. Bormann, G. Arbeiter, and A. Verl, "A feature descriptor for texture-less object representation using 2d and 3d cues from rgb-d data," in *IEEE International Conference on Robotics and Automation(ICRA)*, (Karlsruhe, Germany), pp. 2104–2109, May 2013.

[5] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *IEEE International Conference on Computer Vision(ICCV)*, (Barcelona, Spain), pp. 858–865, 2011.

[6] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of texture-less objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 34, pp. 876–888, 2012.

[7] B. S. Manjunath, J. rainer Ohm, V. V. Vasudevan, and A. Yamada, "Color and texture descriptors," *IEEE Transactions on Circuits and Systems for Video Technology(CSVT)*, vol. 11, pp. 703–715, 2002.

[8] P. Mylonas, E. Spyrou, Y. Avrithis, and S. Kollias, "Using visual context and region semantics for high-level concept detection," *IEEE Transactions on Multimedia*, vol. 11, pp. 229–243, 2009.

[9] D. Zhang, M. M. Islam, and G. Lu, "A review on automatic image annotation techniques," *Pattern Recognition*, vol. 45, pp. 346–362, 2012.

[10] D. Zhang, M. M. Islam, and G. Lu, "Local features and kernels for classification of texture and object categories: A comprehensive study," *Int. J. Computer Vision*, vol. 73, pp. 213–238, 2007.

[11] P. V. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *IEEE International Conference on Computer Vision(ICCV)*, (Kyoto), pp. 221–228, 2009.

[12] A. Rocha, D. C. Hauagge, J. Wainer, and S. Goldenstein, "Automatic fruit and vegetable classification from images," *Computers and Electronics in Agriculture*, vol. 70, pp. 96–104, 2010.

[13] J. Zhao, J. Tow, and J. Katupitiya, "On-tree fruit recognition using texture properties and color data," in *IEEE Intelligent Robots and Systems(IROS)*, (Alberta, Canada), pp. 263–268, 2005.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[15] H. Bay, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, 2008.

[16] S. Beucher and F. Meyer, "The morphological approach to segmentation: the watershed transformation," *Optical Engineering*, vol. 34, pp. 433–481, 1993.

[17] C. Rother, V. Kolmogorov, and A. Blake, ""grabcut": interactive foreground extraction using iterated graph cuts," *ACM Trans. Graph.*, vol. 23, pp. 309–314, 2004.

[18] M.-K. Hu, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, pp. 179–187, Feb. 1962.

[19] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, 2009.