FOREST
A Flexible Object Recognition System

Julia Moehrmann and Gunther Heidemann
Institute of Cognitive Science, University of Osnabrück, Albrechtstr. 28, 49076 Osnabrück, Germany

Keywords: Image Recognition System, Development, Image Recognition, Image Annotation, Ground Truth Annotation.

Abstract: Despite the growing importance of image data, image recognition has succeeded in taking a permanent role in everyday life in specific areas only. The reason is the complexity of currently available software and the difficulty in developing image recognition systems. Currently available software frameworks expect users to have a comparatively high level of programming and computer vision skills. FOREST – a flexible object recognition framework – strives to overcome this drawback. It was developed for non-expert users with little-to-no knowledge in computer vision and programming. While other image recognition systems focus solely on the recognition functionality, FOREST covers all steps of the development process, including selection of training data, ground truth annotation, investigation of classification results and of possible skews in the training data. The software is highly flexible and performs the computer vision functionality autonomously by applying several feature detection and extraction operators in order to capture important image properties. Despite the use of weakly supervised learning, applications developed with FOREST achieve recognition rates between 86 and 99% and are comparable to state-of-the-art recognition systems.

1 INTRODUCTION

While images play an ever more important role in everyday life, image recognition has only succeeded in specific areas like, e.g., bar code or fingerprint recognition. A wide application of computer vision techniques by normal Internet users in the near future is very unlikely. This is mainly due to the complexity of existing image recognition systems. Software frameworks like MATLAB or OpenCV provide extensive functionality, but require programming skills and knowledge about which methods to use for building a recognition system. While users who are interested in developing an image recognition system may already have programming skills, acquiring the necessary computer vision skills requires a lot of time and effort. A software framework which is applicable by non-expert users would have to fulfill a series of requirements. Ideally, the development of a new image recognition system should follow the few simple steps shown in Figure 1. The user decides on a recognition task, selects an appropriate image data source for the task, e.g., a webcam, and annotates the training data. The vision system then learns a classifier based on the image features and the ground truth data, without the need for user interaction. Despite the simplicity of this process, it represents exactly the development process for creating a custom image recognition system with FOREST. In contrast to existing software frameworks FOREST considers all steps of the development process, i.e., the selection of training data, ground truth annotation, calculation of the recognition system, the investigation of classification results and the investigation of possible skews in the training data, not only the vision functionality. The major contribution of FOREST therefore is the presentation of a software tool, which is not used as a collection of algorithms like existing frameworks, but as an out-of-the-box development tool which is in-

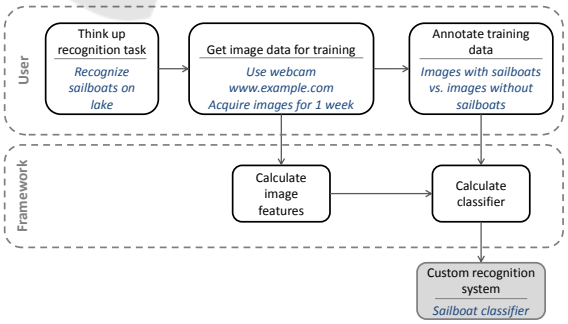

Figure 1: Workflow for development of custom recognition system, divided by manual tasks (user box) and automatic tasks (framework box). Text in lower half (italics) provides and example for the task.

Moehrmann J. and Heidemann G..
FOREST - A Flexible Object Recognition System.
DOI: 10.5220/0005175901190127
In *Proceedings of the International Conference on Pattern Recognition Applications and Methods* (ICPRAM-2015), pages 119-127
ISBN: 978-989-758-077-2

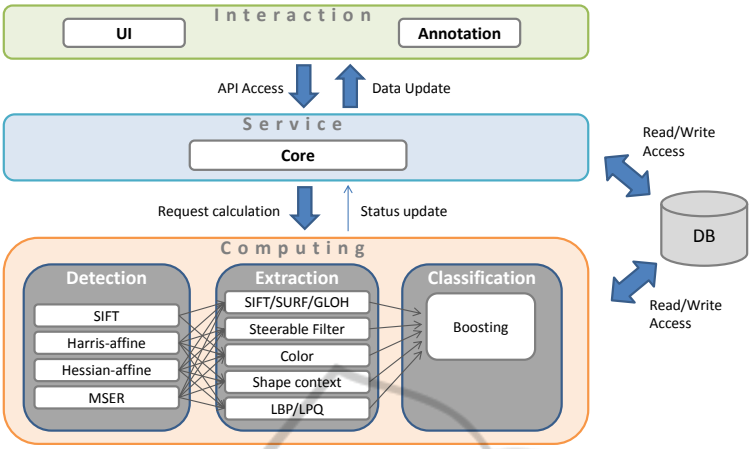

Figure 2: FOREST system design showing three modular layers which represent functionality for user interfaces and user interaction (interaction layer), management of development processes and scheduling of calculation tasks (service layer), and image processing, feature extraction and classification functionality (computing layer). Database is used as shared resource by service and computing layer to avoid transferring data between Java-based service layer and Matlab-based computing layer. Distribution of layers to different servers is possible and intended.

tuitive to use and guides users through all steps of the development process. It does not expect users to have programming skills or any knowledge of computer vision. This leads to certain issues FOREST has to solve. For one, the recognition task intended by the user is not known by the system which means that it has to be able to deal with arbitrary data sets and recognition tasks. Additionally, the missing expert knowledge does not allow the integration of any kind of prior knowledge, e.g., concerning features that could be useful or concerning the parametrization of feature extraction methods. FOREST is capable of achieving high recognition rates on standard and custom data sets by extracting a large set of image features and selecting appropriate features automatically. The selection of appropriate image features is based on the ground truth data provided by the user. The ground truth data is weakly annotated, i.e., each image is annotated as a whole, to enable users to perform the annotation task as efficiently as possible. Despite the lack of region based annotations, results are comparable to state-of-the-art systems.

For reasons of clarity, we define the terms software framework and recognition framework as a software for building and developing custom recognition systems. A custom recognition system in this context is a recognition system which has been trained on a specific, i.e., custom or user-defined, task.

2 REQUIREMENTS

There are a series of requirements that a recognition framework must implement in order to be usable by non-experts. These requirements can be divided into soft requirements and hard requirements. Soft requirements consider human factors which influence the architecture, whereas hard requirements directly consider technical aspects. We identified the following soft requirements for a software framework which allows non-expert users to develop image recognition systems:

- The system must not require any expert knowledge about computer vision or machine learning algorithms. It cannot be expected that users have this kind of knowledge or are willing to acquire it. Similarly, it cannot be expected that users understand the method of using image features and their structure.
- The system has to be usable instantaneously. It must require no training. Beside the technical knowledge the software itself must not present an obstacle itself. This could be the case if too many specific features are available or if technical terms are used.
- The overall time involved for the user in developing a custom image recognition system should be minimal. Similarly, necessary user interaction should be reduced to a minimum.
- Information should be presented to the user in a visual and intuitive way. Abstract representations are preferable over exact representations if they are more intuitive to understand.

Hard requirements are partially derived from these soft requirements:

- Application of different computer vision methods

to compensate for missing expert knowledge and possible variety in recognition tasks.

- Extensibility of software framework regarding image data sources and computer vision methods to allow for future developments. This requirement also implies a high modularity of the software.
- The software framework must not make strict requirements concerning client-side hardware and must not require buying software licenses.

Most of these requirements should be obvious when considering that such a software framework is intended for use by standard Internet users without any expert knowledge. Requirements concerning the instant usability are necessary to ensure potential users are not discouraged by a seemingly complex setup. This also involves that the software framework should – at least in its basic version – be free to use.

3 SYSTEM DESIGN

The technical requirements discussed above are reflected in the system design of FOREST (cf. Figure 2). The framework consists of three major components: the interaction layer, the service layer and the computing layer. The upper two layers are implemented in Java, whereas the computing layer is implemented in Matlab. The great advantage of this design is that multiple Matlab instances may run on physically distributed servers. Calculations are distributed to these Matlab Servers by the service layer. Although the advantage resulting from such a distribution is limited by the database communication this design is well suited to speed-up calculations without the need of having developers care about parallelization inside their image processing code.

Data is stored inside a database to allow for an efficient organization, e.g., of extracted image features. The database setup was chosen to prevent having to transfer data between the computing and service layer which could result in conversion problems.

This system design, with the service and computation layer running on distributed servers was chosen to provide an easily accessible setup. Users only need to install components from the interaction layer locally (although this could be avoided as well) in order to access the systems functionality. This allows for a fast access to the framework and a basically non-existent obstacle for using FOREST.

3.1 Recognition Functionality

The generic recognition functionality of FOREST, which allows the development of recognition systems for arbitrary data sets, is achieved by applying a series of region detection and feature extraction methods. Currently available methods are shown in the computing layer in Figure 2. All methods for ROI detection and feature extraction are applied to the training image data. This is necessary, since users cannot be expected to make an educated decision about which method(s) to use for their specific data set. The huge amount of potential recognition tasks requires that possibly interesting image regions must be detected at this stage. Therefore a larger set of ROI is extracted here, rather than a smaller one. Currently available methods for ROI detection are SIFT (Lowe, 2004), Harris and Hessian affine invariant region detectors (Mikolajczyk and Schmid, 2004), and MSER (Matas et al., 2002). The resulting set of ROI are passed on to the feature extraction methods. Among the currently included feature descriptors are SIFT, color features which comprise Color Layout Descriptors (CLD), Dominant Color Descriptors (DCD), and color histograms (Manjunath et al., 2001), and other popular descriptors. In contrast to recognition systems like (Opelt et al., 2006; Zhang et al., 2005; Hegazy and Denzler, 2009) the result of the feature extraction stage does not only consist of feature sets from two or three feature types, but uses a larger set of different features.

As indicated by the arrows in Figure 2, the different region detection operators are applied to the image and the results are used by the different feature extraction methods, thereby producing a huge feature set which contains a variety of features representing different image properties. The resulting heterogeneous feature set is then passed on to the classifier. The boosting classifier used by FOREST was proposed in (Opelt et al., 2006). The boosting classifier identifies discriminative features from the heterogeneous feature set by calculating weak hypotheses for every positive training feature vector and selecting those with the highest discriminative ability. A weak hypothesis is defined by a feature vector v_t^{wh} of type $t \in T$ and a threshold θ. An image I is classified as positive if $\min_{j=1,..,|v_t^I|}(||v_t^{wh}, v_t^j||) < \theta$, i.e., if one vector of type t from image I is similar enough to the vector v_t^{wh}.

Annotation of the ground truth data for the classifier is described in the next section. It has to be mentioned, however, that the framework supports strong and weak annotation, i.e. the annotation of image regions and the image as a whole. So far the framework

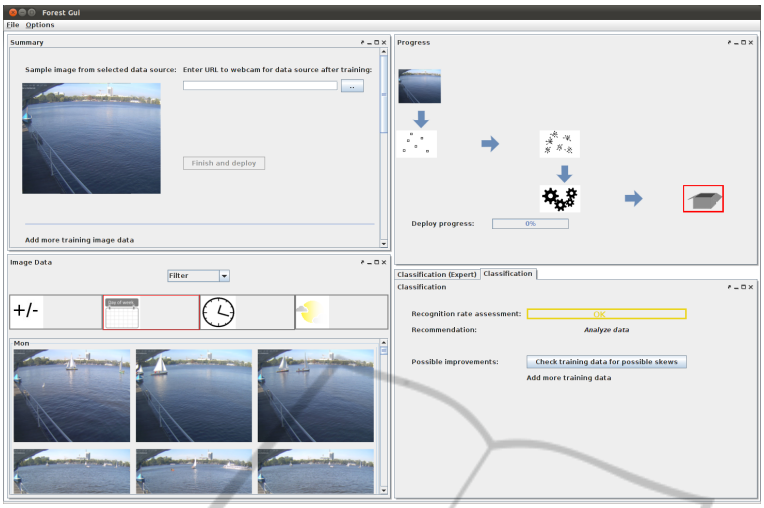

Figure 3: Graphical user interface which guides users through the development process. Summary of image acquisition setup (upper left), progress visualization (upper right), summary of acquired image data (lower left), and estimation of classifier performance (lower right) if available. Webcam data source: http://www.webcam.barca-hamburg.de.

was exclusively used and evaluated with weakly supervised learning, since this greatly reduces the annotation effort for the user.

The extraction of a huge heterogeneous feature set and the calculation of the boosting classifier are relatively expensive from a computational point of view. However, this is not considered as a drawback for FOREST for the following reasons:

- All steps in the development process where the user has to actively participate/interact with the system are highly optimized and the time required by FOREST for automatic processing can be used otherwise.
- The classifier usually employs a limited number of different feature types. Therefore it is unnecessary to apply all operators in the recognition phase, which allows for an efficient recognition.

FOREST does provide the functionality to explicitly set parameters for region detection and feature extraction operators, in order to be usable by expert users also. However, non-expert users are not expected to tune any parameters. Instead, FOREST uses the default parameters proposed in the literature.

4 GRAPHICAL DESIGN AND VISUAL SUPPORT

Users are supported in the development process by an intuitive user interface. In the initial step, all users have to do is select an image data source and specify an image acquisition criterion, e.g., duration, in case the data source is an online resource. In case of a webcam data source users may specify the location of the webcam. This results in the acquisition of additional information for each image, like weather and visibility information at the specified location. These additional attributes can be used to filter the training data and investigate it for possible skews.

After the image acquisition information was specified users are redirected to a general overview shown in Figure 3. The overview shows a summary of the image acquisition specification (upper left), the current development step and progress (upper right), the acquired training image data (lower left), and the classifier performance estimation (lower right) if it is already available. The progress panel gives users a feedback about the current status of the development process. The overview of the acquired training data in the lower left panel provides the possibility of displaying all images, filtering for specific attributes, or displaying the distribution of images. As can be seen in Figure 3 users can filter the image data by different attributes depicted as icons: annotation (+/- icon), date, time, and weather. If training images are taken only for similar weather conditions, within the span of a few days or always at the same time, they will tend to be very similar and exhibit low variance. An additional view using a scatterplot matrix of these attributes can also help to detect correlations and skews, as shown in Figure 4.

The classifier is calculated automatically and evaluated using 10-fold cross-validation. The performance of the classifier is then estimated by FOREST based on the average correct recognition rate. In order to give users an easy-to-understand feedback about

the recognition capabilities of their custom recognition system the assessment is colored green, yellow, or red, indicating very good, OK, or bad performance. Users are then given a hint by the system about how to proceed. In Figure 3, the recognition performance is assessed to be *OK* and the user is given the hint to investigate the training data or to add more training data. The investigation of the training data can be started using the provided button. The user is then directly led to the scatterplot matrix with possibly interesting panels highlighted in red (cf. Figure 4). The scatterplot matrix visualization shows the histogram of a single attribute on the diagonal and the scatterplots on the upper triangle. In the highlighted upper row a skew between positive (1) and negative (-1) training data can be observed. To be more precise, it is obvious that the training data set consists of $\approx 95\%$ negative training samples and less than 50 positive training images. A data set skewed like this can easily lead to a degraded recognition performance. Although the effects of skewed raining data sets are well known, to the best of our knowledge, no attempt has been made to investigate such skews, especially by non-expert users.

It is also possible to view more detailed information about the classifier performance, e.g., average and best classification rate over the number of weak hypotheses used by the boosting classier. This information is considered to be too detailed for beginners and is therefore accessible in a background tab.

Beside the development process, the annotation of ground truth data is an important task which cannot be automated. A specialized user interface has been presented before in (Moehrmann and Heidemann, 2013) to allow for an efficient annotation of ground truth data using a semi-supervised process which arranges images according to similarity.

5 EVALUATION

The recognition ability of custom recognition systems developed with FOREST is shown in this section. For the evaluation no manual adaptations took place, i.e., no preprocessing of the data took place, except a resizing for high resolution images and all methods were run with their default parametrization. The setup therefore corresponds to a non-expert user developing a recognition system.

The evaluation considers artificial examples, like the Graz-02 (Opelt et al., 2006) and the Caltech-101 (Fei-Fei et al., 2004) data sets, however it also considers real-world examples where recognition tasks were defined for local webcam data. The resulting recognition performance depends on the number of weak hypotheses used by the boosting classifier. We calculated the results for $wh = 1,..,300$ weak hypotheses. In general, recognition rates converge around 20 to 100 weak hypotheses. More hypotheses do not have a negative impact due to the small weights they are assigned to in the calculation of the boosting classifier and therefore no overfitting effects can be observed. we present a compact version of the results by giving the average recognition rates for 200 to 300 weak hypotheses. We also provide the number of weak hypotheses at which the results converge, i.e., at which the improvement reduces significantly. A common way to represent the results would be to provide ROC curves. However, this would involve calculating the error rates for different thresholds of the strong boosting classifier. Since non-expert users will not be able to interpret this threshold, FOREST does not consider a modification. The results are meant to represent the real performance of custom recognition systems developed by non-expert users.

Figure 4: Scatterplot matrix of training image data distribution considering annotation and additional attributes like weather, time, and date. Histograms of single attribute are shown on diagonal. Panels showing possibly skewed data are highlighted in red.

5.1 Artificial Data

The evaluation on artificial data sets is intended to show the general recognition capabilities of FOREST and the benefit of using a large feature set. For the Graz-02 data, a custom recognition system was calculated for each of the three categories *bike*, *car*, and *person*. The calculation was repeated ten times. In each iteration 150 images from the positive and negative category were randomly chosen for training. The evaluation used the remaining images. Results are given in Table 1. All results range above 86%, without any specific selection of training data samples or

Table 1: Results for recognition systems on Graz-02 data set, averaged over ten runs with randomly selected training data.

Category	Avg. rec. rate	#wh
Bike	86.34%	16
Car	86.86%	10
Person	86.1%	6

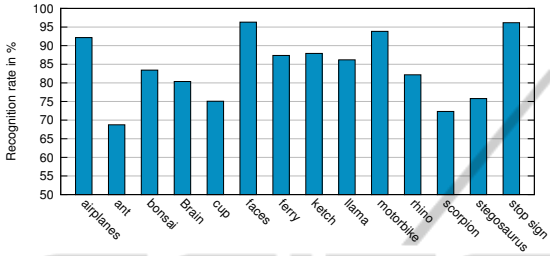

Figure 5: Recognition rates averaged over ten training episodes. Each episode used 30 randomly selected positive and negative images for training.

methods to use. These results are above those reported in (Opelt et al., 2006), especially for the *car* category which was reported with 67.2% and the *bike* category which was reported to be 73.5%. A detailed investigation of the selected weak hypotheses shows that the classification is indeed based on discriminative structures of the objects and persons.

In order to prove the general recognition capability of FOREST, 14 random categories were chosen from the Caltech-101 data. The overall performance of FOREST on this data set is limited by the feature descriptors used since weakly supervised learning is not expected to have a negative effect with this data set. The evaluation of all 101 categories is therefore obsolete in this context. The results of all categories are considered separately. Each recognition system was calculated ten times on a different set of randomly selected training images. For each training episode 30 positive and negative training images were used. For testing, 50 positive and negative images were used. The results are given in Figure 5 which shows high recognition rates for most categories. Weaker categories correspond to those which contain complex structures and a high variance as, e.g., ants or scorpions.

5.2 Real-world Data

The evaluation on real-world data sets is of importance since weakly supervised learning might have a stronger effect in such recognition scenarios. Real-world examples consider the recognition of open windows, sailboats on a lake, and cars parked in a no-parking zone. For the windows an internal office webcam was used, the image data for the other two examples was acquired from publicly available webcams.

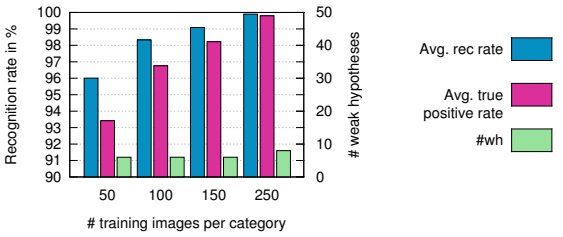

Figure 6: Results for recognizing open windows in an office room using different numbers of training images per category. Results are averaged over ten runs.

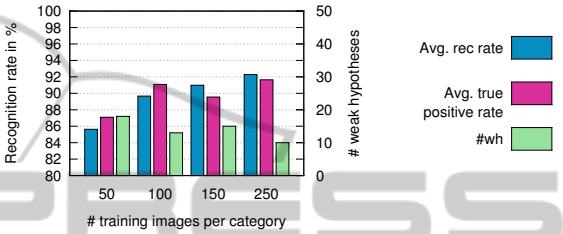

Figure 7: Results for recognition of sailboats on a lake using different numbers of training images per category. Results are averaged over ten runs.

Recognition of open windows in an office building is relevant to prevent theft due to neglect. Results for the recognition of open windows are given in Figure 6. The evaluation was run ten times with randomly selected training images and different numbers of training images per category. Even for a small number of training images recognition rates are very high. However, as the number of training images is increased it can be seen that the true positive rate increases significantly, resulting in near-to-perfect recognition performance.

The evaluation for the recognition of sailboats uses the same setup. The recognition rate increases with the number of training images. Here it can also be seen that the classifier requires less weak hypotheses for a larger number of training images. This is due to the fact that the training data provided more detailed information which allows the classifier to select highly discriminative features. A detailed investigation shows that, for a small number of training images, features in the water area are used for classification, whereas more training images lead to a small number of weak hypotheses targeting sailboats only.

In contrast to the other systems, the detection of cars in a no-parking zone was evaluated using a typical setup. That is, images were acquired over the course of one week. The recognition system was then

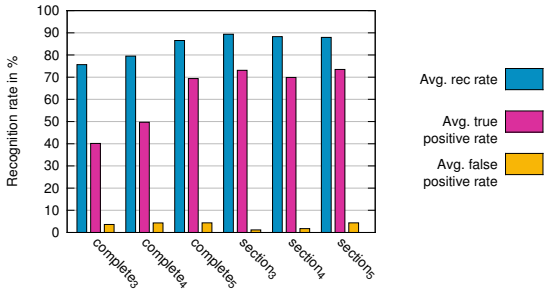

Figure 8: Results for recognizing cars in a no-parking zone. Recognition systems were calculated on all features from the image over a course of n days ($complete_n$) and over an image region selected by the user ($section_n$).

Figure 9: Webcam image of harbor and no-parking zone (highlighted in red). Webcam data source: http://www.frs.de/nc/de/frs-webcams/stralsund.html

trained on the images of the first n days and tested on the images of the following days. We expect this to be a typical setup since it is expected that users develop custom recognition systems in this manner. The data set contained approximately equal numbers of positive and negative training images. The results are given for $n = 3,4,5$ days in Figure 8, denoted as $complete_n$. The results show a significant difference between the average recognition rate and the true positive rate, which increases with the number of training images. Since the webcam mainly shows the harbor, the actual no-parking zone makes up only a small part of the image, as can be seen in Figure 9. When the image data source is initially selected by the user he or she also has the possibility of selecting an image region for observation. The recognition system then focuses recognition on this image section only. The results for image recognition systems for which a selection of the no-parking zone took place are given as $section_n$ (the selected image section was a rectangular region around the area showing the street). As can be seen in Figure 8 the $section_n$ results basically exhibit no differences for varying numbers of training images. Additionally, only very few weak hypotheses are required for achieving high recognition rates. Errors were mainly due to cars driving on the street, close to the no-parking zone. Unfortunately, the update rate of the webcam is not high enough to allow frame by frame comparisons or tracking of cars. However, we believe the recognition could further be improved using more training data, especially such data that considers more variance in weather and lighting conditions.

Table 2: Results for multi-class recognition systems. Recognition of the correct category is given by top_1. top_n refers to the correct category being included in the top n ranked categories.

Data set	top_1	top_2	top_3
Flowers17	79%	89%	92%
AT&T Faces	86.6%	92.35%	93.26%

5.3 Multi-class Recognition

FOREST supports the development of multi-class recognition systems. This evaluation uses the Flowers17 (Nilsback and Zisserman, 2006) and the AT&T Faces (Samaria and Harter, 1994) data sets, with a five-fold cross-validation. Average recognition results over all categories are given in Table 2 as top_n for $n = 1,2,3$. These modified recognition rates consider an image as being classified correctly if it corresponds to one of the n top-ranked categories by the classifier. As can be seen in the results, recognition rates are well above 90% if we consider $n = 2$. A recognition system like the one for flowers is intended for a certain community which is interested in the name and type of a flower. Such a system could benefit largely from a visualization of results for the best matching n categories with probabilities and sample images given. It would then serve as a decision basis for users. Due to the large variety in floral representations such a setup is most likely to succeed in a real-world application.

Results on the Flowers data set are comparable to those reported by (Nilsback and Zisserman, 2006) with a top_1 recognition rate of 81.3%. Despite the optimization of parameters by (Nilsback and Zisserman, 2006) FOREST reaches almost the same results with default parametrization only. Results for the face data set are very high although the data set contains only a small number of images per person. This suggests, that face recognition on private photo collections should be possible with high accuracy.

6 LITERATURE

Generic recognition systems try to solve a similar problem as FOREST. While generic recognition systems are able to recognize several object classes, a flexible recognition system like FOREST is meant to be adapted to an arbitrary recognition task. Nevertheless, generic recognition systems have been found to perform better when using multiple feature channels (Opelt et al., 2006; Zhang et al., 2005; Hegazy and Denzler, 2009; Varma and Ray, 2007).

The area of tangible user interfaces provides two examples for systems which require a flexible rather than a generic recognition functionality: *Crayons* (Fails and Olsen, 2003) and *Papier-Mâché* (Klemmer et al., 2004). Both systems provide the possibility of creating simple gesture recognition systems for interaction purposes. The underlying recognition functionality is, however, limited to very basic color information.

A recognition system which intends to use webcams is *Eyepatch*. It requires no expert knowledge in the areas of image recognition, but requires that the user applies and combines predefined classifiers. Another system which intends to use existing webcams is *Vision on Tap* (Chiu, 2011). It provides specific processing blocks which implement motion tracking, skin color recognition or face recognition. These can be combined in a visual computing application to create custom recognition systems. Although a nice variety of applications can be implemented using these building blocks, the resulting functionality is effectively limited.

7 CONCLUSIONS

A software framework, FOREST, for the development of custom, i.e. user-defined, recognition systems was presented. In order to be usable by non-expert users such a system has to fulfill a set of requirements which were discussed and implemented. In contrast to other existing systems FOREST considers all aspects of the development process from a non-expert users point of view. The image processing functionality is fully automated, requiring no programming skills or expert knowledge. Interactive steps in the development process were enhanced using semi-automatic techniques like, e.g., the identification of possible skews in the training data set. The user is even supported in the assessment of the classifier performance.

In contrast to existing software frameworks FOREST does not provide a collection of algorithm but instead allows the adaption of the recognition functionality to a specific user-defined recognition task. FOREST achieves this functionality by extracting a large heterogeneous feature set from the images and applying a boosting classifier which selects discriminative features based on the ground truth data provided by the user. The application of such a heterogeneous feature set allows the identification of important image properties despite the lack of knowledge about the (type of) recognition task even with weakly supervised learning.

REFERENCES

Chiu, K. (2011). Vision On Tap : An Online Computer Vision Toolkit. Master's thesis, Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences.

Fails, J. and Olsen, D. (2003). A Design Tool for Camera-based Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 449–456. ACM.

Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. In *IEEE CVPR Workshop on Generative-Model based Vision*.

Hegazy, D. and Denzler, J. (2009). Generic Object Recognition. In *Computer Vision in Camera Networks for Analyzing Complex Dynamic Natural Scenes*.

Klemmer, S., Li, J., Lin, J., and Landay, J. (2004). Papier-Mâché: Toolkit Support for Tangible Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 399–406. ACM.

Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *Intl. Journal of Computer Vision*, 60:91–110.

Manjunath, B., Ohm, J.-R., Vasudevan, V., and Yamada, A. (2001). Color and Texture Descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703–715.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *British Machine Vision Conference*, volume 1, pages 384–393.

Mikolajczyk, K. and Schmid, C. (2004). Scale and Affine Invariant Interest Point Detectors. *Intl. Journal of Computer Vision*, 60(1):63–86.

Moehrmann, J. and Heidemann, G. (2013). Semi-Automatic Image Annotation. In *Computer Analysis of Images and Patterns*, volume 8048 of *Lecture Notes in Computer Science*, pages 266–273.

Nilsback, M.-E. and Zisserman, A. (2006). A Visual Vocabulary for Flower Classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 1447–1454. IEEE.

Opelt, A., Pinz, A., Fussenegger, M., and Auer, P. (2006). Generic Object Recognition with Boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):416–431.

Samaria, F. and Harter, A. (1994). Parameterisation of a Stochastic Model for Human Face Identification. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pages 138–142. IEEE.

Varma, M. and Ray, D. (2007). Learning The Discriminative Power-Invariance Trade-Off. *IEEE Intl. Conference on Computer Vision (ICPR)*, 0:1–8.

Zhang, W., Yu, B., Zelinsky, G., and Samaras, D. (2005). Object Class Recognition using Multiple Layer Boosting with Heterogeneous Features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 323–330.