A Comparative Study of Local Detectors and Descriptors for Mobile Document Classification

Marçal Rusiñol*†, Joseph Chazalon†, Jean-Marc Ogier† and Josep Lladós*
*Computer Vision Center, Dept. Ciències de la Computació
Edifici O, Univ. Autònoma de Barcelona
08193 Bellaterra (Barcelona), Spain.
†L3i Laboratory, Université de La Rochelle
Avenue Michel Crépeau
17042 La Rochelle Cédex 1, France.

***Abstract*—In this paper we conduct a comparative study of local key-point detectors and local descriptors for the specific task of mobile document classification. A classification architecture based on direct matching of local descriptors is used as baseline for the comparative study. A set of four different key-point detectors and four different local descriptors are tested in all the possible combinations. The experiments are conducted in a database consisting of 30 model documents acquired on 6 different backgrounds, totaling more than 36.000 test images.**

I. Introduction

Document image classification is one of the main topics of interest among the document image analysis community [1], mainly because it is a required step in many different contexts, from retrieval to document understanding. Depending on the particular application, document classes are defined either in terms of textual contents, document structure or visual similarity.

A document image classification system based on textual contents means that document images will be casted to the same class if they "talk" about the same topics. Obviously such systems require an initial OCR step in order to automatically extract the text from the document images. A subsequent step devoted to transform text strings to a feature vector is applied. The bag of words technique [2] is the most common approach, although more complex topic modeling approaches such as latent semantic analysis [3] or latent Dirichlet allocation [4] provide better results. From this numeric representation, any statistical classifier can be trained to finally decide to which class assign an incoming document.

A second family of document image classifiers would be the approaches that are based on document structure similarity. Such methods allow to assign a document class whenever the documents share the same physical or logical structure, regardless of their contents. A first layout analysis step is devoted to decompose the document images into blocks so the document similarity can be expressed in terms of the spatial relationships among those blocks. Such spatial relationships are represented by particular data structures such as attributed relational graphs [5] or X-Y trees [6]. Although such representations have an important discriminatory power, they present the drawback that computing a mapping between two layout structures is computationally expensive.

Finally, document image classifiers based on visual similarity will group document images into the same class if they "look" similar. Usually the proposed descriptors use statistics computed over low-level features in order to encode such visual appearance of the document. Such low-level features are directly the average pixel intensity as in the works proposed in [7], [8], or a little more elaborate strategies such as multi-oriented run-lengths as proposed in [9]. Such representations have the advantage that they are normally cheap to extract and to embed in a statistical classifier. However they offer in general a limited discriminative power when it comes to a more fine-grained document image classification.

Although those three families try to solve different user definitions of what a document class or similarity means, they all present the same shortcoming. Such document image descriptions are in general global, due to the fact that historically the document images to process in our community came from a digitization device such as a scanner or a fax machine. Even lately, we started to directly process digitally-born documents directly in electronic format. However, with the quick growth of the camera quality in mobile phones and the ubiquity of such devices, in the last years, to deal with mobile captured document images has become an interesting research topic.

When acquiring a document image with a mobile gimmick we can not expect the same kind of document images to process. Such mobile digitization process entails illumination problems, geometric distortions due to the perspective effect, occlusions or cluttered background, etc. In such scenarios, it is doubtful that the classical global document image descriptors would be able to perform properly. However, some renowned object recognition techniques from the Computer Vision field, based on local descriptors, have been designed to be able to tackle such inconveniences, so researchers from the Document Analysis field started to apply such techniques to deal with document images.

One might still wonder whether those object recognition frameworks are really adaptable to processing document images, and more specifically, which local key-points detectors and local descriptors from the vast plethora that has been proposed in the last decade are the most promising and have the best performances when dealing with the particular case of document images.

In this paper we focus on the specific problem of document image classification with document images acquired with mobile devices. We propose to tackle such problem with an approach founded on a direct matching of local descriptors followed by a voting strategy. We analyze the performance of different state-of-the-art local key-point detectors and descriptors depending on a number of different factors such as the document type or the acquisition conditions. Such analysis is conducted in a large-scale scenario with more than 36.000 images in the test set.

The organization of the rest of the paper is as follows. In Section II we overview the document classification strategy which is based on matching local descriptors. In Section III, we overview the set of different state-of-the-art local key-point detectors and descriptors that we use in this analysis. Section IV reports the experimental results, while finally in Section V we draw our conclusions.

II. Document Classification by Matching Local Descriptors

In order to perform the document image classification by matching local descriptors we followed a similar approach than the one previously presented in [10].

In the "training" phase, we need an example of each document class in order to extract and index their local descriptors. Given a collection of documents D and a sample document $d_i \in D$ of a class i, we extract their local key-points K_i. Each key-point $k_i^j \in K_i$ consists of the coordinates of the key-point, a scale factor and an orientation. We then compute the local descriptors F_i obtaining a feature vector $f_i^j \in F_i$ for all the key-points k_i^j.

All local descriptors F_i are then stored in an indexing structure with their associated document index i. In this paper we use the FLANN [11] indexing framework in order to compute similarities between the stored descriptors and the local descriptors from an incoming image. When using binary descriptors the Locality Sensitive Hashing indexing structure is used within FLANN. For integer or floating point descriptors we use a KD-tree indexing structure.

When an incoming document image arrives, we compute its local key-points and associated local descriptors. Such local descriptors are then matched against the whole indexing structure. We accumulate votes to the matched document class i. When all the local descriptors from the incoming image have been matched against our indexed corpus, the document class receiving more matches is the one taken as the corresponding class.

Two additional points are considered in order to make the results more reliable. On the one hand, we use a ratio-test, as in the original SIFT paper [12], in order to just keep the local matches that are really discriminative. That is, a match is considered only if the distance to the nearest local descriptor and the distance to the second local descriptor is sufficiently different. On the other hand, a RANSAC step [13] allow to filter out matches that might be correct in terms of similarity of local descriptors but that do not agree with the rest of matches in terms of the projective transform that we should apply to go from a set of local key-points in the incoming image to the set of local key-points in the model document image.

III. Local Detectors and Descriptors

We will overview in this section the off-the-shelf local detectors and local descriptors that we used in our study. We have used the baseline OpenCV's implementation[1] of the local detectors and descriptors.

A. Local Key-point and Region Detectors

Let us overview the four different key-point detectors that we have used in our comparison.

- **SIFT**: Key-point detector proposed by D. Lowe in [12] in which key-points are extracted as maxima of the Difference of Gaussians over a scale space analysis at different octaves of the image. Dominant orientations are assigned to localized key-points.
- **SURF**: Key-point detector proposed by H. Bay et al. in [14] which detects blobs based on the determinant of the Hessian matrix.
- **ORB**: Key-point detector proposed by E. Rublee et al. in [15] which uses an orientation-based implementation of the FAST corner detection algorithm.
- **MSER**: Key-region detector by J. Matas et al. in [16] which is based on finding stable and extremal regions when applying a connected component analysis after iteratively thresholding the image.

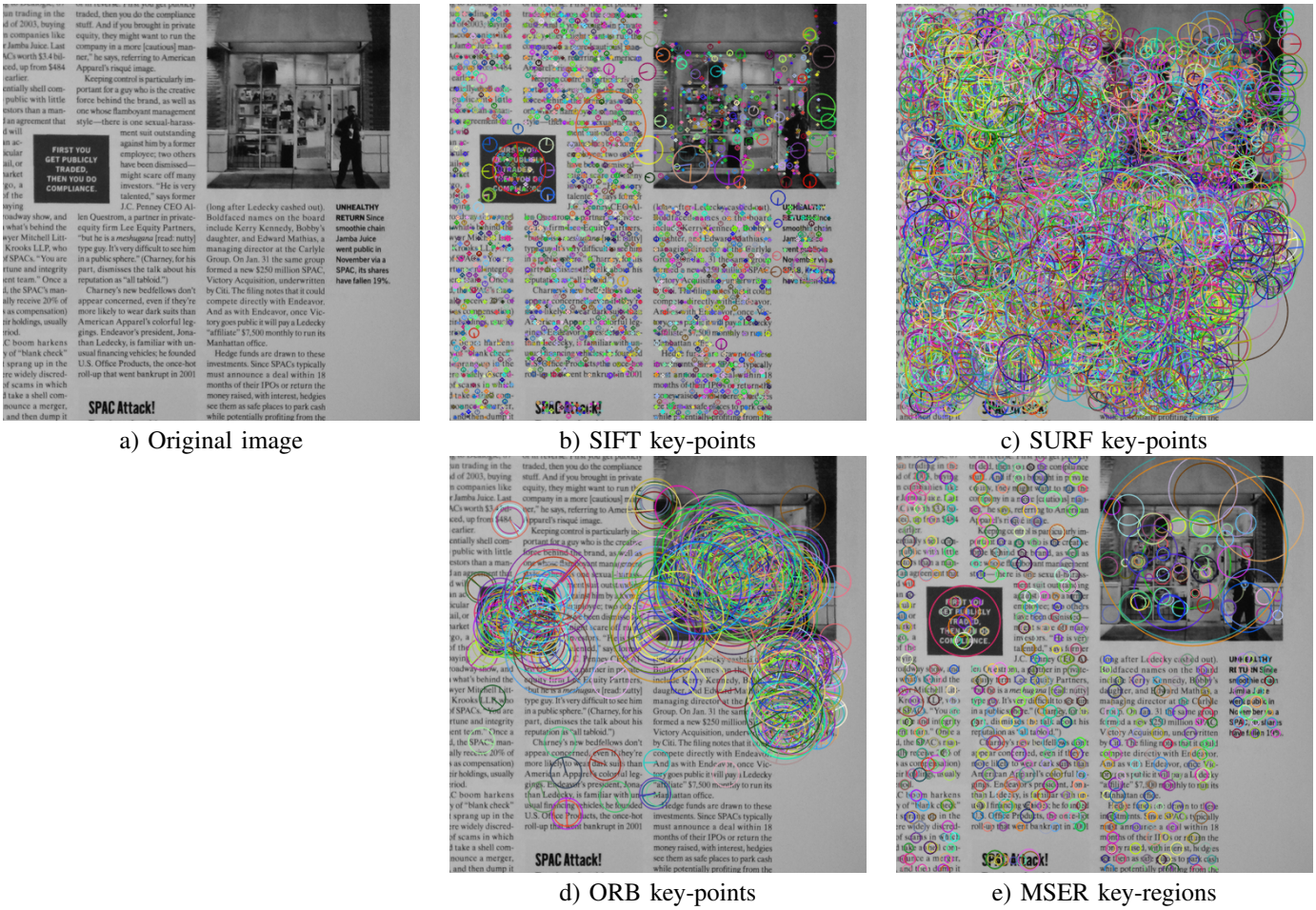
We present in Figure 1 an example of the extracted local key-points within a portion of a document image. It is worth noting the differences in terms of which kind of information is retained by the different key-point extractors, as well as the amount of key-points issued by the different algorithms with their default parameters. Obviously, such difference in the amount of key-points will result in higher matching times for the algorithms yielding high amounts of key-points when analyzing document images.

B. Local Descriptors

Let us overview the four different local descriptors we have used in our comparison.

- **SIFT**: is a local descriptor proposed in [12] which coarsely describes edges appearing in key-point frames by an orientation histogram over the gradient image.
- **SURF**: is a local descriptor proposed in [14] which is based on the computation of Haar wavelet responses in a dense fashion within the key-point frames.
- **ORB**: is a binary local descriptor proposed in [15] which is a rotation aware version of the BRIEF descriptor [17]. It basically encodes the intensity differences among a set of pairs of pixels.
- **BRISK**: is a binary local descriptor proposed by S. Leutenegger et al. in [18] which is also based on a pair-wise comparison of pixel intensities.

[1] http://www.opencv.org



a) Original image b) SIFT key-points c) SURF key-points

d) ORB key-points e) MSER key-regions

Fig. 1. Example outputs of the different key-points detectors for the same portion of a document image.

It is worth noting that both SIFT and SURF yield an integer-valued histogram while ORB and BRISK produce binary strings. Such binary descriptors are matched against each other with a Hamming distance which entails a much faster distance computation than the Euclidean distance calculation done for SIFT and SURF descriptors.

IV. Experimental Results

A. Dataset

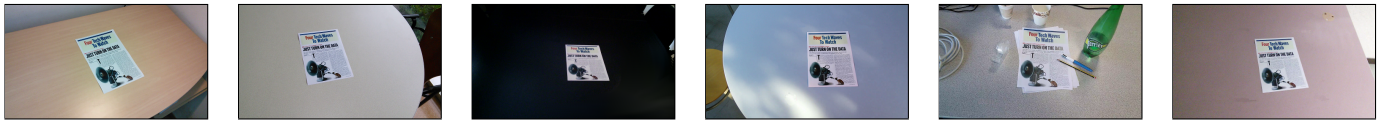
To build our dataset, we took six different document types coming from public databases and we chose five document images per document type. We have chosen the different types so that they cover different document layout schemes and contents (either completely textual or having a high graphical content). In particular, we have taken data-sheet documents and patent documents retrieved from the Ghega dataset [19]. Title-pages from medical scientific papers from the MARG dataset [20]. Colour magazine pages from the PRIMA layout analysis dataset [21]. American tax forms from the NIST Tax Forms Dataset (SPDB2) [22]. And finally typewritten letters from the Tobacco800 document image database [23]. We show an example of each of those six different document types in Figure 2. We removed some small noise and margins from the original document images and finally rescaled them to all have the same size and fit an A4 paper format.

Each of these document models were printed using a color laser-jet and we proceeded to capture them using a Google Nexus 7 tablet. We recorded small video clips of around 10 seconds for each of the 30 documents in six different background scenarios. The videos were recorded using Full HD 1920×1080 resolution at variable frame-rate. Since we captured the videos by hand-holding and moving the tablet, the video frames present realistic distortions such as focus and motion blur, perspective, change of illumination and even partial occlusions of the document pages. In addition of the video clips, we also have captured an 8Mp picture of each of the documents to be used as models for the matching classification. We present an example of the different scenarios in Figure 3. Summarizing, the database consists of 180 video clips comprising 36.444 frames.

The classification task consists in automatically recognizing for each frame which of the thirty different documents appears.

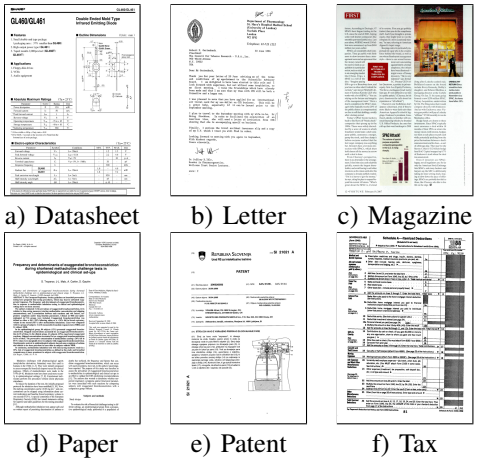
B. Results

We report in Tables I and II the obtained classification accuracies and the required processing times for all the possible combinations of local key-point detectors and local descriptors. The best performances are reached when using the SIFT descriptor over SIFT key-points. The performance



Background 1 Background 2 Background 3 Background 4 Background 5 Background 6

Fig. 3. Sample backgrounds used in our dataset when capturing the same magazine document.



a) Datasheet b) Letter c) Magazine

d) Paper e) Patent f) Tax

Fig. 2. Sample documents used in our dataset. a) Data-sheet from Ghega, b) letter from Tobacco800, c) magazine from PRIMA, d) paper from MARG, e) patent from Ghega and f) tax form from NIST.

TABLE I. CLASSIFICATION ACCURACIES

Descriptor	Detector			
	SIFT	SURF	ORB	MSER
SIFT	85.16	67.81	36.07	55.32
SURF	53.95	61.10	64.26	53.65
ORB	29.67	25.00	70.81	9.01
BRISK	69.51	79.66	39.20	63.23

TABLE II. PROCESSING TIMES (SECS.) PER IMAGE

Descriptor	Detector			
	SIFT	SURF	ORB	MSER
SIFT	2.974	5.166	3.886	0.273
SURF	1.824	2.439	0.593	0.100
ORB	0.334	0.222	0.033	0.065
BRISK	0.420	0.339	0.057	0.070

of the descriptor drops quite dramatically when it is computed over other key-points. Concerning the SURF key-point detector and descriptor, we appreciate that the SURF framework does not seem very suited for the specific task of dealing with document images. One the one hand, when computing local descriptors over SURF key-points, it is the BRISK descriptor that yields the best performances. On the other hand, the SURF descriptor achieves the best performances when used over ORB key-points. In both cases, the reached performances are quite far from the ones reached by SIFT. In addition, the processing times for the SURF descriptors are also important. Despite the fact that the SURF framework was proposed as a faster alternative to SIFT, the vast amount of key-points found when used over document images with its default parameters (we can appreciate it in Figure 1) provokes that using SURF might not be suitable in applications with real-time requirements. Computing SURF descriptors over SURF key-points is almost as costly as computing SIFT descriptors over SIFT key-points, with an important drop in accuracy. However, the SURF key-points were the ones which gave the best performance for the BRISK local descriptor. BRISK being a binary descriptor, achieves very good classification accuracies with much faster processing times than SIFT or SURF. However, being computed over such large amount of key-points, makes that the ORB framework is almost 10 times faster. Finally, we can appreciate that whatever descriptor we use, the MSER key-regions do not reach very good results. It is worth to note that MSER does not provide a region orientation *per se*, so in principle computing local descriptors over MSER key-regions would not reach invariance to rotation. However, in OpenCV's implementation, SIFT, SURF and BRISK descriptors actually compute a dominant orientation in order to do an orientation normalization whereas ORB expects the key-regions to have already their angular information. This fact would explain the important drop in performance that we observe when computing ORB descriptors over MSER key-regions. Summarizing, the best performances are obtained by using the SIFT key-point detector with the SIFT descriptor, but it is worth to note that in scenarios requiring real-time computation, the performances reached when using the ORB framework are also very promising, since although there is a drop of nearly 15% classification accuracy, the processing times are trimmed by two orders of magnitude.

We report in Tables III and IV the averaged classification accuracies for the different backgrounds and document types respectively. We report the performances reached with the most promising configurations from Tables I and II. We show that the acquisition conditions have a strong impact on the final performances. The classification abilities are hindered when dealing with images acquired in low-light conditions, as in backgrounds 3 and 6 or in scenarios with severe clutter and occlusions, as the background 5. The performance drop due to low-light conditions (backgrounds 3 and 6) affects in a more severe way the BRISK and ORB descriptors than the SIFT descriptor, in which such performance drop is not that severe. Such effect is easy to understand taking into account that both BRISK and ORB are based on pixel intensities comparison, and thus quite sensitive to illumination changes, whereas the SIFT descriptor is based on gradient orientations which is more robust to such distortions. We also observe that the performance of the detectors and descriptors under analysis can also be severely hindered depending on the nature of the document to deal with. Documents which contain mostly text with uniform layouts such as the papers, patents or datsheets are harder to describe than the other documents presenting either more graphical information or having a more "textured" layout, as in the case of the letters, magazines and tax forms.

TABLE III. CLASSIFICATION ACCURACIES FOR THE DIFFERENT BACKGROUNDS

Detector &	Background					
Descriptor	1	2	3	4	5	6
SIFT / SIFT	98.79	95.69	81.47	89.22	61.94	83.82
SURF / BRISK	97.42	95.77	54.07	91.91	68.44	70.37
ORB / ORB	96.85	85.63	55.03	81.67	40.75	64.90

TABLE IV. CLASSIFICATION ACCURACIES FOR THE DIFFERENT DOCUMENT TYPES

Detector &	Document Type					
Descriptor	Datasheet	Letter	Magazine	Paper	Patent	Tax
SIFT / SIFT	79.79	86.80	98.67	68.09	80.63	96.96
SURF / BRISK	65.51	79.95	98.03	64.76	74.25	95.48
ORB / ORB	51.47	73.06	94.68	45.57	77.03	83.02

V. CONCLUSIONS

We have presented in this paper a comparative study of local key-point detectors and local descriptors for the specific task of mobile document classification. The experiments, conducted in a database consisting of 30 model documents acquired on 6 different backgrounds, totaling more than 36.000 test images, show the dominance of the SIFT framework over other detectors and descriptors. Despite their lower performance, binary descriptors are also a good choice when the requirements of the application impose to have real-time responses. In such scenarios, both BRISK and ORB descriptors perform well, presenting processing times between one and two orders of magnitude difference in comparison with SIFT.

Finally, the detailed analysis of the results shows that the performance of such detectors and descriptors can be severely hindered depending on the nature of the document to deal with. Mostly textual documents are harder to describe than other documents presenting more graphical information or with a more "textured" layout. The acquisition conditions have also a strong impact on the final performances. The classification abilities are distorted when dealing with low-light conditions or with severe clutter and occlusions.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish Ministry of Education and Science under projects TIN2014-52072-P, and TIN2012-37475-C02-02 and by the People Programme (Marie Curie Actions) of the Seventh Framework Programme of the European Union (FP7/2007-2013) under REA grant agreement no. 600388, and by the Agency of Competitiveness for Companies of the Government of Catalonia, ACCIÓ.

REFERENCES

[1] N. Chen and D. Blostein, "A survey of document image classification: problem statement, classifier architecture and performance evaluation," *International Journal on Document Analysis and Recognition*, vol. 10, no. 1, pp. 1–16, June 2006.

[2] F. Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[3] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science and Technology*, vol. 41, no. 6, pp. 391–407, September 1990.

[4] D. Blei, A. Ng., and M. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. 4–5, pp. 993–1022, 2003.

[5] A. Bagdanov, "Fine-grained document genre classification using first order random graphs," in *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, 2001, pp. 79–83.

[6] F. Cesarini, M. Lastri, S. Marinai, and G. Soda, "Encoding of modified X-Y trees for document classification," in *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, 2001, pp. 1131–1136.

[7] P. Héroux, S. Diana, A. Ribert, and E. Trupin, "Classification method study for automatic form class identification," in *Proceedings of the Fourteenth International Conference on Pattern Recognition*, 1998, pp. 926–928.

[8] P. Sidiropoulos, S. Vrochidis, and I. Kompatsiaris, "Content-based binary image retrieval using the adaptive hierarchical density histogram," *Pattern Recognition*, vol. 44, no. 4, pp. 739–750, April 2011.

[9] A. Gordo, F. Perronnin, and E. Valveny, "Large-scale document image retrieval and classification with runlength histograms and binary embeddings," *Pattern Recognition*, vol. 46, no. 7, pp. 1898–1905, July 2013.

[10] M. Rusiñol and J. Lladós, "Logo spotting by a bag-of-words approach for document categorization," in *Proceedings of the International Conference on Document Analysis and Recognition*, 2009, pp. 111–115.

[11] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, 2009, pp. 331–340.

[12] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, November 2004.

[13] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[14] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of the International Conference on Computer Vision*, 2011, pp. 2564–2571.

[16] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of the British Machine Vision Conference*, 2002, pp. 384–396.

[17] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, July 2012.

[18] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proceedings of the International Conference on Computer Vision*, 2011, pp. 2548–2555.

[19] E. Medvet, A. Bartoli, and G. Davanzo, "A probabilistic approach to printed document understanding," *International Journal of Document Analysis and Recognition*, vol. 14, no. 4, pp. 335–347, December 2011.

[20] G. Ford and G. Thoma, "Ground truth data for document image analysis," in *Proceedings of the Symposium on Document Image Understanding and Technology*, 2003, pp. 199–205.

[21] A. Antonacopoulos, D. Bridson, C. Papadopoulos, and S. Pletschacher, "A realistic dataset for performance evaluation of document layout analysis," in *Proceedings of the 10th International Conference on Document Analysis and Recognition*, 2009, pp. 296–300.

[22] D. Dimmick, M. Garris, and C. L. Wilson, "Structured forms database," National Institutte of Standards and Technology, Tech. Rep., 1991.

[23] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard, "Building a test collection for complex document information processing," in *Proceedings of the 29th Annual International ACM SIGIR Conference*, 2006, pp. 665–666.