**WÜRSCH — Diva**

<4 images, plus some Word figures>

**DivaServices—A RESTful Web Service for Document Image Analysis Methods**

Würsch, M., Ingold, R. and Liwicki, M.

This paper presents an open-source web service providing researchers in all fields with state-of-the-art computational methods for several Document Image Analysis (DIA) steps. Research on automatic DIA focuses mainly on developing and refining automatic processing steps, e.g., text line extraction, binarization, and layout analysis. While many state-of-the-art methods perform satisfactorily, the algorithms applied to obtain the results are not easily accessible for other researchers. Just making the source available online is not sufficient, as it typically requires a cumbersome installation of required libraries and reading long manuals about the usage. Our approach is to directly make the methods available as web services that can be accessed via RESTful HTTP requests, the current state of the art in online web communication. Thus, the resulting services can be integrated easily into document processing workflows by any software engineer without specific knowledge of the mathematical and algorithmic details of DIA. We will build on standards for result presentation, such as the Text Encoding Initiative (TEI)1 and the International Image Interoperability Framework (IIIF).2

**State of the Art**

Our research is motivated by the availability of many different web-based tools for researchers with a humanist background wanting to do document image analysis (e.g., *SALSAH* [Schweizer and Rosenthaler, 2011], Transcribe Bentham [Causer and Wallace, 2012], or the Genizah project [Wolf et al., 2011]). Most of these tools were either developed to solve a specific task or lack the inclusion of (semi)-automatic methods. Several projects using web services for DIA have been proposed in recent years. One such example is the Document Analysis and Exploitation (DAE; Lamiroy and Lopresti, 2011) that provides different algorithms as web services and allows for workflow creation. Our aim is to expand this research with a focus on making the algorithms available for researchers with only little computer science knowledge by providing them also with simple web interfaces as showcases building on the web services and demonstrating how to use them to integrate computational methods into their research.

**Methodology**

We propose an open-source framework for providing algorithms to the public. For this we designed a RESTful web service architecture exposing all information using the JavaScript Object Notation (JSON). The intention is to include a wide assortment of services for different tasks:

• Image processing and enhancement in order to make the desired content more easily visible or to make the processing of further automatic analysis simpler. Those methods include, for example, binarization methods (Otsu 1979), Laplacian of Gaussian (LoG), Difference of Gaussian (DoG).

• Document layout analysis methods allowing automatic extraction of texts, text lines, or images. These methods include pixel- (Wei, 2013) and interest point– (Garz et al., 2011) based approaches.

• Optical Character Recognition (OCR) to support the transcription of the documents.

• Methods for palaeographic analysis, such as script identification (Ghosh et al., 2010), writer identification (Fiel et al., 2013), and water mark analysis.

• Methods for feature extraction and feature selection, so that computer scientists can directly work on extracted meta-information without any specific knowledge in DIA. For example, the following methods are included: Local Binary Patterns (LBP; Nicolaou et al., 2014), Scale-Invariant Feature Transform (SIFT; Lowe, 1999), Gabor features (Chen et al., 2010), standard feature search algorithms, as well as several feature selection methods (Wei et al., 2014).

• Machine Learning algorithms: Support Vector Machines (SVMs), k-nearest neighbor algorithm (k-NN), Gaussian Mixture Models (GMMs).3

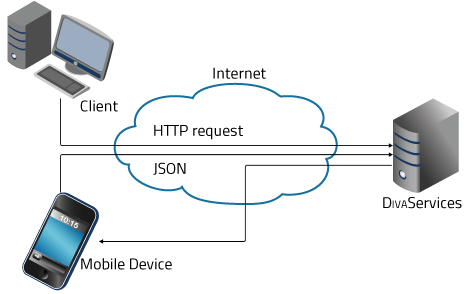
• Evaluation metrics for the automatic assessment of results and to allow computer science researchers to compare their systems. There we will build on the standards laid out in DAE.

Figure 1.Conceptual overview of the proposed DivaServices framework. Access to the provided methods and tools would all be standardized using HTTP requests and JSON as input/output format.

Besides a large set of own implementations we will integrate several open-source software like Tesseract4and OCROpus.5 This enables fast integration of many available image processing algorithms that have been in development for years and proven to produce reliable results.

A high-level overview of the proposed framework is provided in Figure 1. Access to the provided tools and algorithms would be standardized across all possible end-user applications using simple HTTP requests and JSON as input/output format. For accessing the methods we will follow the proposed URL format for RESTful services.

The current state of DivaServices is available at http://divaservices.unifr.ch. Using GET requests allows for browsing the available services. We are in the process of developing a web front-end that will allow for automated prototype creation of all available algorithms in order to allow for experimenting with them.

Since we only provide algorithms, creating specific workflows is left to developers designing client applications and can therefore be designed targeting the specific need of end users. At a later stage we aim at directly implementing some of the more common workflows directly into DivaServices.

As proof of concept6 a simple histogram-based line segmentation method was exposed using the proposed framework. This service was then integrated into the DivaDiaWT, a web-based interface that allows for the creation of transcriptions. The user interface of the DivaDiaWT is shown in Figure 2.

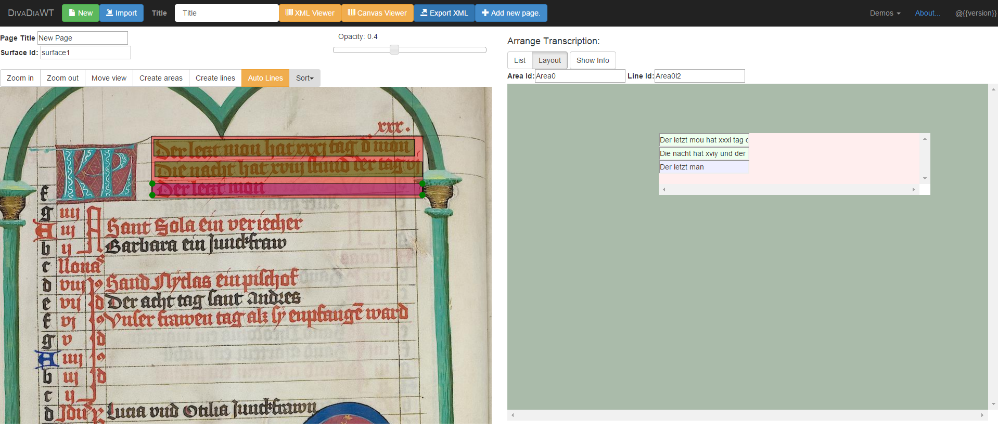


Figure 2.Overview of the DivaDiaWT. The original image is displayed on the left side, transcriptions on the right side. Transcriptions can be displayed in Layout mode, where they are aligned with the original image.

The line segmentation service is used on a user-marked region and automatically extracts lines from there. In Figure 3 a user created a box using his mouse around a region that he wants to have automatically processed into text lines. The result of the automated text line segmentation is shown in Figure 4.

We have set up a web server on which we run our RESTful web services. When the user wants to automatically segment a text area, a POST request is made to the server containing the following JSON:

{

"url": "http://www.e-codices.unifr.ch/loris/bbb/bbb-0360/bbb-0360\_001r.jp2/full/pct:25/0/default.jpg",

"top": "300",

"bottom": "500",

"left": "190",

"right": "750"

}

Listing 1.Example of a JSON sent together with the POST request for automatic line segmentation. The url points to the source image; the four location values mark the region within the image that the user selected.

[

{

"bottom": "180",

"left": "95",

"right": "469",

"top": "156"

},

{ …}

]

The web service downloads the image and processes the region marked by the user. This can lead to several detected text lines, and the server responds with a JSON file containing the bounding box of each detected text line:

Listing 2. The JSON the server sends back to the client application containing all bounding boxes of the detected text lines.

The DivaDiaWT parses this information and presents it as shown in Figure 4.

[

{

"bottom": "180",

"left": "95",

"right": "469",

"top": "156"

},

{ …}

]

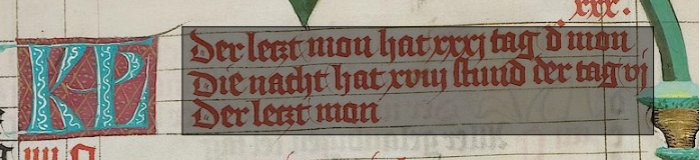


Figure 3. The user marked a box (visible as grey shaded area) that should be automatically divided into text lines using an algorithm provided by DivaServices.

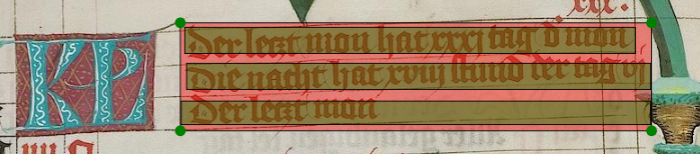


Figure 4. The result of the automated segmentation process.

**Notes**

1. http://www.tei-c.org.

2. http://www.iiif.io.

3. See Ma and Doermann (2003) for an overview of classification algorithms in script identification.

4. http://code.google.com/p/tesseract-ocr.

5. http://github.com/tmbdev/ocropy.

6. Code of the proof of concept is available at https://github.com/lunactic/Diva-WebService.

**References**

**Causer, T. and Wallace, V.** (2012). Building a Volunteer Community: Results and Findings from Transcribe Bentham. *Digital Humanities Quarterly,* **6**: 1–28.

**Chen, J., Cao, J., Prasad, R., Bhardwaj, A. and Natarajan, P.** (2010). Gabor Features for Offline Arabic Handwriting Recognition. *International Workshop on Document Analysis Systems*, pp. 53–58.

**Fiel, S., Hollaus, F., Gau, M. and Sablatnig, R.** (2013). Writer Identification on Historical Glagolitic Documents. *Document Recognition and Retrieval,* 902102–902102-10.

**Garz, A., Sablatnig, R. and Diem, M.** (2011). Layout Analysis for Historical Manuscripts Using Sift Features. *International Conference on Document Analysis and Recognition,* pp. 508–12.

**Ghosh, D., Dube, T. and Shivaprasad, A. P.** (2010). Script Recognition—A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **20**: 1–21.

**Lamiroy, B. and Lopresti, D.** (2011). An Open Architecture for End-to-End Document Analysis Benchmarking. *International Conference on Document Analysis and Recognition*, pp. 42–47.

**Lowe, D. G.** (1999). Object Recognition from Local Scale-Invariant Features. *International Conference on Computer Vision,* pp. 1150–57.

**Ma, H. and Doermann, D.** (2003). Word-Level Script Identification for Scanned Document Images. *Document Recognition and Retrieval,* pp. 124–35.

**Nicolaou, A., Slimane, F., Maergner, V. and Liwicki, M.** (2014). Local Binary Patterns for Arabic Optical Font Recognition. *International Workshop on Document Analysis Systems* (IEEE), pp. 76–80.

**Otsu, N.** (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics,* **9**: 62–66.

**Schweizer, T. and Rosenthaler, L.** (2011). SALSAH—eine virtuelle Forschungsumgebung für die Geisteswissenschaften. *EVA 2011*, pp. 147–53.

**Wei, H.** (2013). Layout Analysis on Historical Documents. *International Conference on Document Analysis and Recognition, Doctoral Consortium (ICDAR DC).*

**Wei, H., Chen, K., Ingold, R. and Liwicki, M.** (2014). Hybrid Feature Selection for Historical Document Layout Analysis. *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 87–92.

**Wolf, L., Littman, R., Mayer, N., German, T., Dershowitz, N., et al.** (2011). Identifying Hoin Candidates in the Cairo Genizah. *International Journal of Computer Vision,* **94**: 118–35.