The Journal of Open Source Software

Handprint: a program to explore and compare major cloud-based services for handwritten text recognition

Michael Hucka[1]

1 Caltech Library, California Institute of Technology, Pasadena, CA 91125, USA

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Daniel S. Katz

Reviewers:

- @step21
- @rlskoeser

Submitted: 30 March 2022
Published: unpublished

Summary

Handprint (***Hand****written* ***p****age* ***r****ecognit****i****on* ***t****est*) is a command-line application that can invoke cloud-based services to perform handwritten text recognition (HTR) on images of documents. It accepts images in various popular formats, sends them to service providers, gathers the results, annotates copies of the images to show the results to the user, and optionally performs other operations. It currently supports HTR services from Amazon (Amazon, Inc., 2022a, 2022b), Google (Google, Inc., 2022), and Microsoft (Microsoft, Inc., 2022), but its architecture is modular and could be extended to other services. Handprint is a command-line program written in Python and can run on macOS, Windows, and Linux computers.

Statement of need

The goal of automating the recognition of text dates back, at minimum, to efforts in the 1950's to develop machines for banking applications (Berkeley, 1956; Dimond, 1957). The early methods were extremely limited in scope: they focused essentially on numbers only. Thanks to decades of advances in machine learning, document analysis, and computing power, methods have become so advanced that they can now be used to recognize cursive handwriting in dozens of human languages (Muehlberger et al., 2019). Today, handwritten text recognition (HTR) is even offered as a service by several computing companies over the Internet, on demand, for small fees—without the need to first train a system on samples of a person's handwriting. The recognition results are remarkably good overall, but there are differences in quality and features between the different offerings. Comparing the results produced by the competing services is complicated by the fact that they each have unique application programming interfaces (APIs).

The purpose of Handprint is to make it easier to test HTR services and compare the results, without the need for users to learn how to write program or work with the different APIs. With Handprint, users can easily test cloud-based HTR services on individual images, directories of images, and URLs pointing to images on remote servers, all without writing a line of code. If desired, users can also use Handprint in scripts as part of automated workflows.

Summary of Handprint usage

Configuration

The only configuration necessary after installation is to run Handprint with a command-line option to store the user's account credentials for each cloud-based HTR service provider. The command needs to be run once for each desired provider, and thereafter, Handprint will use the account information automatically. The documentation at https://caltechlibrary.github.io/handprint/ explains the simple file format in which the credentials need to be written.

The Journal of Open Source Software

Basic features

Handprint can read many common image formats: JP2, JPEG, PDF, PNG, GIF, BMP, and TIFF. Image paths or URLs can be supplied to Handprint in any of the following ways: (a) one or more directory paths or one or more image file paths on the local disk, which will be interpreted as images—either individually or in directories—to be processed; (b) one or more URLs, which will be interpreted as network locations of image files to be processed; or (c) if given the `-f` command-line option (`/f` on Windows), a file containing either image paths or image URLs to be processed. When using URLs, Handprint first downloads the image found at the given URL(s) to a directory of the user's choosing on the local disk. No matter whether files or URLs, each item should be an image of a single document page containing text.

Handprint's basic features include the ability to select a subset of services to use, save the full raw results from HTR services as JSON or text files, and use multiple processor threads to speed up processing. For example, using one of the sample images found in Handprint's source directory, the following command,

```
handprint --text-size 20 --display text,bb-line H96566k.jpg
```

will send image file `H96566K.jpg` to all services currently supported (Amazon Rekognition and Textract, Google Cloud Vision, and Microsoft Azure Computer Vision). The output will be a file named `H96566k.handprint-all.png` with the contents shown in Figure 1.

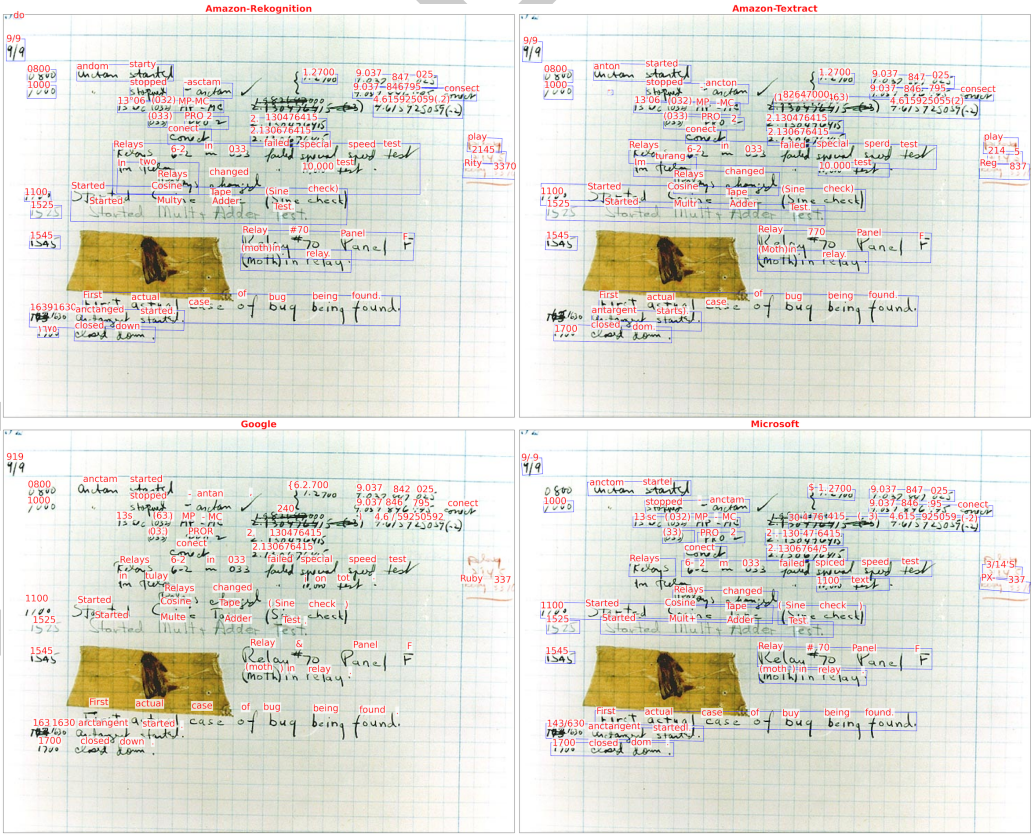

Figure 1: Sample output from Handprint. (Source image: Wikipedia (Wikipedia contributors, 2012).)

Users can can opt to skip the creation of the overview grid image if they only need the other types of outputs that Handprint can produce.

The Journal of Open Source Software

Advanced features

Handprint also includes additional, more advanced features, including the following:

- *Controlling the style and placement annotations overlaid on input images*. Users control whether to display recognized text, bounding boxes, or both, and which types of bounding boxes (word, line, and/or paragraph—although not all services provide all types).
- *Filtering the results by confidence scores*. This allows users to see which words or other components have confidence values that meet or exceed a chosen threshold.
- *Comparing text results to expected (ground truth) text*. Users can supply a text file containing the expected text for a given image, and Handprint will calculate the number of errors and the character error rate for each line. The comparison algorithm has some novel capabilities, notably in how it can treat missing, extra, or transposed lines of text from the HTR results (a common difference between the outputs of different services).

Documentation

A detailed user manual is available at https://caltechlibrary.github.io/handprint/. Handprint also prints usage information to the terminal when given the command-line option `--help`.

Acknowledgments

Handprint benefited from feedback from several people, notably Tommy Keswick, Mariella Soprano, Peter Collopy and Stephen Davison of the Caltech Library, and from bug reports by GitHub users "braytac" (actual name unknown), Tommaso Bendinelli, Tom Morrell, and "syed-9909" (actual name unknown).

References

Amazon, Inc. (2022a). *Amazon Rekognition [Computer software]*. https://aws.amazon.com/rekognition/

Amazon, Inc. (2022b). *Amazon Textract [Computer software]*. https://aws.amazon.com/textract/

Berkeley, E. C. (1956). Magnetic Ink Character Recognition. *Computers and Automation*, *5*(10).

Dimond, T. L. (1957). Devices for reading handwritten characters. *Papers and Discussions Presented at the December 9-13, 1957, Eastern Joint Computer Conference: Computers with Deadlines to Meet*, 232–237. https://doi.org/10.1145/1457720.1457765

Google, Inc. (2022). *Google Cloud Vision API [Computer software]*. https://cloud.google.com/vision/docs/handwriting

Microsoft, Inc. (2022). *Microsoft Azure Computer Vision [Computer software]*. https://azure.microsoft.com/en-us/services/cognitive-services/computer-vision/

Muehlberger, G., Seaward, L., Terras, M., Ares Oliveira, S., Bosch, V., Bryan, M., Colutto, S., Déjean, H., Diem, M., Fiel, S., Gatos, B., Greinoecker, A., Grüning, T., Hackl, G., Haukkovaara, V., Heyer, G., Hirvonen, L., Hodel, T., Jokinen, M., … Zagoris, K. (2019). Transforming scholarship in the archives through handwritten text recognition: Transkribus as a case study. *Journal of Documentation*, *75*(5), 954–976. https://doi.org/10.1108/JD-07-2018-0114

Wikipedia contributors. (2012). *First Computer Bug*. https://commons.wikimedia.org/wiki/File:First_Computer_Bug,_1945.jpg